Sensor Fused Scene Reconstruction and Surface Inspection

Daniel Thien-An Moodie

Thesis submitted to the faculty of the Virginia Polytechnic Institute and State University in partial fulfillment of the requirements for the degree of

Master of Science

In Mechanical Engineering

Alfred L. Wicks

Kathleen Meehan

John P. Bird

Feb 20, 2014

Blacksburg, Virginia

Keywords:
Sensor Fusion, 3D Scene Reconstruction, Computer Vision, Robotic Perception, Surface Defect Characterization

Sensor Fused Scene Reconstruction and Surface Inspection

Daniel Thien-An Moodie

Abstract

Optical three dimensional (3D) mapping routines are used in inspection robots to detect faults by creating 3D reconstructions of environments. To detect surface faults, sub millimeter depth resolution is required to determine minute differences caused by coating loss and pitting. Sensors that can detect these small depth differences cannot quickly create contextual maps of large environments.

To solve the 3D mapping problem, a sensor fused approach is proposed that can gather contextual information about large environments with one depth sensor and a SLAM routine; while local surface defects can be measured with an actuated optical profilometer. The depth sensor uses a modified Kinect Fusion to create a contextual map of the environment. A custom actuated optical profilometer is created and then calibrated. The two systems are then registered to each other to place local surface scans from the profilometer into a scene context created by Kinect Fusion.

The resulting system can create a contextual map of large scale features (0.4 m) with less than 10% error while the optical profilometer can create surface reconstructions with sub millimeter resolution. The combination of the two allows for the detection and quantification of surface faults with the profilometer placed in a contextual reconstruction.

Acknowledgements

I would like to thank God for blessing me with the desire to learn and the determination to survive this thesis. I would like to thank my wife for loving and supporting me through many neglectful nights while collecting and processing data. I would like to thank my parents for giving me advice and support in continuing this project.

I would like to thank my Virginia Tech advisors: associate professor of mechanical engineering Dr. Alfred Wicks, associate professor of electrical and computer engineering (retired) Dr. Kathleen Meehan, and research scientist Dr. John Bird for their support and guidance. If not for their influence in undergraduate and graduate school, I would not have found my passion for sensing, robotics, and programming.
Specifically, Dr. Meehan taught me that even mechanical engineers can enjoy electronics in my sophomore Electrical Circuit Analysis class. Dr. Wicks and Dr. Bird both taught me to enjoy robotics and programming through Mechatronics and Dr. Wicks taught me to view sensing in a new light in Advanced Instrumentation.

Lastly, I would like to mention the great support from my friends who helped me along this path. Ben Goldman who listened to my many questions when first learning about cameras and image processing. Michael Teresi has always been a great resource and an ear to talk to when programming is not going well. James Burns and Ryan Brown who have been by my side during many days of programming, and Alex Cope who built the platform for the original project from which this thesis was developed.

Table of Contents

1 – Introduction 1
1.1 – Part Inspection 1
1.1.1 – Optical Non-Contact Inspection 2
1.2 – Problem Statement 4
1.2.1 – Thesis 5
1.2.2 – Justification 5
1.3 – System Overview 5
1.4 – Thesis Overview 6
2 - Background 7
2.1 - Optical Measurement Techniques 7
2.2 - Structured Light – 2D 13
2.3 – Two Dimensional Structured Light Correspondence 14
2.4 - Structured Light – 3D from Actuated 2D Scans 16
2.5 - Structured Light – 3D 21
2.6 - Microsoft Kinect – Improved 3D Structured Light 22
2.7 - Camera Models 23
2.8 - Light Source Calibration 25
2.9 - Motion Calibration 27
2.10 – Point Cloud Simultaneous Localization And Mapping 29
2.11 – Registering Profilometer Scans to Context Scans 30
2.12 - Point Cloud Combination 30
2.13 – Three Dimensional Reconstruction 31
2.13.1 – Algorithmic Improvements towards 3D Reconstruction 31
2.13.2 – Sensor Fused Approaches for 3D Reconstruction 32
3 - Theory 33
3.1 – Swept Plane Structured light 33
3.1.1 – Profilometry via Triangulation 33
3.1.2 – Structured Light Parameter Tuning 36
3.1.3 - Camera Calibration 38
3.1.4 – Camera Lens Distortion Correction 39
3.1.5 - Structured Light Calibration 40
3.1.5 - Actuator Calibration 40
3.1.6 – Line Striped Structured Light Correspondence 42

3.1.7 – Qualitative Correspondence Results 45
3.2 – Six Degree of Freedom SLAM 48
3.2.1 – Real-Time 3D Depth Camera Correspondence 48
3.2.2 – Real-Time Simultaneous Localization And Mapping 49
3.3 – Sensor Fusion 51
3.3.1 – Sensor Characteristics – Actuated Profilometer 51
3.3.2 – Sensor Characteristics – Prime Sense Depth Camera 51
3.3.3 – Sensor Registration 52
3.3.4 – Data Fusion 52
4 – Hardware Platform and Software 53
4.1 – Sensor Array 53
4.2 – Lighting Control Scheme 55
4.3 – Structured Light Correspondence Training and Verification 56
4.4 – Computational Setup 58
5 – Calibration Procedure and Results 60
5.1 – Profilometer Calibration Verification 60
5.2 – Camera Calibration 60
5.2.1 – Checkerboard Pose Uncertainty 61
5.3 – Structured Light Calibration and Verification 63
5.4 – Motor Encoder Validation 65
5.5 – Camera to HS322-HD Homogeneous Transformation Calculation and Validation 67
5.6 – Camera to Dynamixel Homogeneous Transformation Calibration and Verification 68
5.7 – Calibration Results 69
6 – Reconstruction Procedure and Results 70
6.1 – Ground Truth Production and Comparison Techniques 70
6.1.1 – Comparison with Manual Measurement 71
6.1.2 – CAD Reconstruction 72
6.2 – SLAM Accuracy Analysis 73
6.2.1 – Kinect SLAM Reconstruction Accuracy 73
6.2.2 – Kinect SLAM Localization Accuracy 75
6.3 – Profilometer Accuracy Evaluation 76
6.3.1 – Effects of Exposure Control on Accuracy 80
6.3.2 – Flat Plane Profilometer Reconstruction 81
6.3.3 - Object Local Feature Reconstruction 82

6.4 – Sensor Fusion Results 85
7 – Conclusion and Future Work 87
7.1 – Conclusion 87
7.1.1 – Actuated Profilometer Calibration 87
7.1.2 – Real Time Full 3D Simultaneous Localization And Mapping 87
7.1.3 – Sensor Fusion 87
7.1.4 – Notes on Improving Profilometer Performance 87
7.2 – Future work 89
7.2.1 – Component Testing and Algorithm Improvements 89
7.2.2 – Calibration Algorithm and Methodology Improvements 90
7.2.4 – Depth SLAM 90
7.2.5 – Integration with Mobile Robotics Platform 91
References 92

Figure 1: Inspection tools for measuring part dimensions.. 2
Figure 2: Different inspection environments from structured to unstructured. 3
Figure 3: Two types of industrial LiDAR sensors, Flash and scanning 8
Figure 4: Single wavelength interferometer, white light interferometer. 9
Figure 5: Example of stereo vision. 10
Figure 6: Shape from focus. 10
Figure 7: Profilometer imaging a surface. 13
Figure 8: Romer Absolute Arm 14
Figure 9: Support Vector Machines. [45][46]. Used under fair use 2014. 16
Figure 10: Coordinate systems and coordinate system transforms. 17
Figure 11: Rotation of an object in sensor coordinate system. 19
Figure 12: Actuated profilometer reconstruction. 20
Figure 13: Time, color, and spatial encoding of 3D structured light. 21
Figure 14: Depth from focus via astigmatic lens 22
Figure 15: Modified pinhole camera model 23
Figure 16: Traditional pinhole camera model 25
Figure 17: Calibration of structured light laser and projector 27
Figure 18: Basic 2D model of triangulation by structured light. 34
Figure 19: Single plane structured light scanning visualized 35
Figure 20: Effect of occlusion on triangulation. 36
Figure 21: Resolution vs. distance and baseline distance. 37
Figure 22: Working distance vs. laser angle for different baseline distances. 38
Figure 23: Pseudo masking to find red laser line. 43
Figure 24: Select images from processing pipeline for laser correspondence. 43
Figure 25: Effect of saturated pixels due to over exposure on line detection. 44
Figure 26: Profilometer correspondence on red surface. 46
Figure 27: Profilometer correspondence while viewing light source. 47
Figure 28: Cross section of a TSDF volume 50
Figure 29: Simulation robot sensor rig 53
Figure 30: Laser width vs. distance for short and medium range lasers. 54
Figure 31: Exposure control validation plots. 55
Figure 32: SVM accuracy vs. sigma for a sweep of sigma from 10^-5 to 10^5. 58
Figure 33: Two checkerboard patterns used to calibrate camera 61
Figure 34: Checkerboard positioning uncertainty with distance to camera. 62
Figure 35: Standard deviation of checkerboard pose estimation vs distance between camera and checkerboard. 63
Figure 36: Corner extraction and resultant laser line in 3D space. 64
Figure 37: Estimated laser plane from collected data 64
Figure 38: Error of the estimated plane model. 65
Figure 39: Calculated and Measured displacements for HS322HD. 67
Figure 40: Calculated and Measured displacements Dynamixel. 69
Figure 41: Three-dimensional reconstruction of the global test scene. 71
Figure 42: Simulated local features, from view of camera. 71
Figure 43: Colorized point cloud based on inconsistencies with original mesh. 72
Figure 44: SLAM reconstruction accuracy for several different objects, for features in all dimensions. 74

Figure 45: Error in feature measurement vs. feature size through SLAM reconstruction..... 74
Figure 46: SLAM localization accuracy..... 75
Figure 47: SLAM localization accuracy for different voxel sizes..... 76
Figure 48: Profilometer residual error for a single line at various distances..... 77
Figure 49: Profilometer absolute distance compared to hand measured distance..... 78
Figure 50: Profilometer test setup..... 79
Figure 51: Typical profilometer image..... 79
Figure 52: Single frame reconstruction of a piece of paper..... 79
Figure 53: Estimated step height vs distance..... 79
Figure 54: Profilometer residual error and recall for a single line at various exposures..... 80
Figure 55: Average residual errors for one hundred scans at various distances with the HS-322HD digital servo..... 81
Figure 56: Reconstruction of one set of local features..... 82
Figure 57: Estimated distance between parallel planes. Full 3D reconstruction with each laser. 83
Figure 58: Profilometer point cloud overlaid with Kinect point cloud..... 85
Figure 59: Raw point cloud overlay..... 86

Table 1: Depth sensor comparisons. 12
Table 2: Digital servo motor comparison. 55
Table 3: Classification accuracy of an SVM with different features. 57
Table 4: Camera parameters after calibration 61
Table 5: HS322HD repeatability results. 66
Table 6: Dynamixel repeatability results. 66
Table 7: Rigid body transformation between camera and actuator. 67
Table 8: Dynamixel mounting homogeneous transformation. 68
Table 9: Transformation between profilometer and Kinect coordinate systems. 85

List of Acronyms and Abbreviations

2D	Two Dimensional
3D	Three Dimensional
COTS	Consumer Off The Shelf
CPU	Central Processing Unit
GPU	Graphical Processing Unit
HD	High Definition
ICP	Iterative Closest Point
PCA	Principal Component Analysis
RGB-D	Red, Green, Blue, Depth
SLAM	Simultaneous Localization And Mapping
SVM	Support Vector Machine
TSDF	Truncated Surface Distance Function

1 – Introduction

Mobile robotic inspection is poised to increase efficiency and improve the safety of regularly scheduled preventative maintenance inspection of large facilities. It is necessary for safety in accident inspection such as in the Fukishima reactor. Routine inspection is critical in nuclear power plants, chemical processing facilities, and oil refineries to prevent catastrophes such as the explosion at NDK's manufacturing plant in 2009 [1], [2]. Routine inspections can improve the efficiency and uptime of power plants and jet engines [3]. These inspections are often carried out by a complete shutdown of the subject component and an operator crawling into the enclosure to view the environment [4]. Defects are then recorded via pictures and paper notes and in rare cases advanced tools are used to measure important defects. It is a labor- and time-intensive process that may also place the operator in some danger.

Mobile robotic inspection could improve this process by creating holistic reconstructions of the environment, thus providing visual context; as well as providing detailed analysis of local defects. An interest in these tools has already been expressed by the Department of Defense [4]. These tools can be an improvement over human inspection techniques in several ways. Safety can be improved by keeping the human operator well away from the inspection environment. Repeatability can be improved by creating a consistent reference frame for inspection. Facility downtime can be reduced since a robot can enter an area that may typically require several hours to exhaust or cool down before a human can enter; and, in extreme cases, a robot could inspect hostile environments without shutting down the facility.

These advancements come at little added cost to the inspection process. A flexible mobile robot could be made for less than $40,000 while it could considerably reduce inspection operation costs. The limiting factor in using these tools is the dissemination of knowledge about their power, their limited availability and the amount of customization required for specific inspection environments. To advance this field, it is critical that there be tried-and-true guidelines to building inspection robots as well as the creation of custom sensors for specific applications.

1.1 – Part Inspection

Part inspection is a fundamental part of manufacturing process control and maintenance routines. In manufacturing plants, it is common to inspect widgets frequently during fabrication, to prevent defectives components from reaching the customer. In order to do this, it is often necessary to measure the physical characteristics of the widget. Dimensional measurements can be manually taken with inspection tools such as Calipers (*Figure 1a*), micrometer (*Figure 1b*), and height gages (*Figure 1c*). As well, these measurements can be automatically done via coordinate measurement machines (*Figure 1d*).

a b c d

Figure 1: Inspection tools for measuring part dimensions. *"Super Caliper-Solar Powered- Series 500-No battery or origin reset needed for IP67 Digital Caliper." [Online]. Available: http://ecatalog.mitutoyo.com/Super-Caliper-Solar-Powered-Series-500-No-battery-or-origin-reset-needed-for-IP67-Digital-Caliper-C1587.aspx. [Accessed: 03-Mar-2014]. "Coolant Proof Micrometer Series 293-with Dust/Water Protection Conforming to IP65 Level." [Online]. Available: http://ecatalog.mitutoyo.com/Coolant-Proof-Micrometer-Series-293-with-DustWater-Protection-Conforming-to-IP65-Level-C1089.aspx. [Accessed: 03-Mar-2014]. "ABSOLUTE Digimatic Height Gage Series 570-with ABSOLUTE Linear Encoder." [Online]. Available: http://ecatalog.mitutoyo.com/ABSOLUTE-Digimatic-Height-Gage-Series-570-with-ABSOLUTE-Linear-Encoder-C1270.aspx. [Accessed: 03-Mar-2014]. "Crysta-Apex S Series-191- Standard CNC CMM." [Online]. Available: http://ecatalog.mitutoyo.com/Crysta-Apex-S-Series-191-Standard-CNC-CMM-C1812.aspx. [Accessed: 03-Mar-2014].* Used with permission of Todd Himes, Mitutoyo 2014.

The results of these inspections can lead to process improvements and valuable preventative maintenance. In high risk applications it is sometimes necessary to inspect a part after every use to find early signs of failure; thus, inspections must be done in the field [4]. With facility maintenance it is common to shut down a facility to inspect pipes, turbines, and pressure vessels.

1.1.1 – Optical Non-Contact Inspection

Optical non-contact inspection is useful to prevent potential damage of delicate surface finishes, as well, optical techniques can be more flexible to measure crevices where probes cannot reach. Most current optical inspection tools are designed to be used in manufacturing environments during fabrication. There are many categories of these tools due to the different types of environments that they are designed for, as illustrated in *Figure 2* where the categories are defined based on the degree of designed environmental control. These categories of control will be described as follows: structured manual part inspection (*Figure 2a*), structured robotic part inspection (*Figure 2b*), semi-unstructured manual part inspection (*Figure 2c*), and unstructured robotic part inspection (*Figure 2d*).

a

b

c

d

Figure 2: Different inspection environments from structured to unstructured. Romer Absolute Arm *E. Metrology, "ROMER ABSOLUTE ARM SI product page." [Online]. Available: http://www.exactmetrology.com/products/romer/romer-absolute-si-arm/. [Accessed: 01-Sep-2013]*. Used with permission of Hexagon Metrology 2014. Mitutoyo CMM *"Crysta-Apex S Series-191- Standard CNC CMM." [Online]. Available: http://ecatalog.mitutoyo.com/Crysta-Apex-S-Series-191-Standard-CNC-CMM-C1812.aspx. [Accessed: 03-Mar-2014]*. Used with permission of Todd Himes, Mitutoyo 2014. Creaform MetraScan 210 *Creaform, "Optical CMM 3D scanner for scanning and probing | MetraSCAN by Creaform." [Online]. Available: http://www.creaform3d.com/en/metrology-solutions/optical-3d-scanner-metrascan. [Accessed: 19-Feb-2014]*. Used with permission of Creaform 3D 2014. Inspector Systems eddy current pipe inspection robot *I. Systems, "INSPECTOR SYSTEMS, eddy current test robot, complete pipe inspection," 2008. [Online]. Available: http://www.inspector-systems.com/Eddy_current_test_robot.html. [Accessed: 29-Oct-2013]*.Used with permission of Inspector Systems 2014.

Manual optical part inspection consists of a fixed platform where the inspector uses optical techniques to take measurements without contacting the widget, an example of this can be seen in *Figure 2a*. The most common case of this is a bench-top structured light measurement station in a manufacturing process line. In such a case, the work environment is fixed and the widget is placed in the work environment. Thus, the environment is controlled by the fact that the instrumentation needs to be fixed in a static reference frame relative to the part. This would not be suitable in an environment where the system must traverse terrain to observe more parts of the environment.

An improvement on manual part inspection is to program a robot to perform an inspection without intervention, as seen in *Figure 2b*. In such a case, the environment would again be controlled and the widget would be placed in a known location. Then a robot would take measurements of known points relative to each other, and output an analysis of the widget. This has the same limitation in the sense that the robot is a fixed reference frame and that the part must be placed in a set volume for measurements. This would not be suitable in an inspection environment where the robot must enter the environment.

It is often difficult to create a structured environment around large parts while maintaining manufacturing line flexibility. Thus, there are measurement systems designed to be flexible and use-able in semi-unstructured environments. Such systems create a common reference frame and track the inspection tool throughout the inspection frame. The tool can then be moved by the user to inspect points of interest. This allows for the widget to be moved with imprecise industrial moving equipment while still allowing for accurate measurements. This can be seen in *Figure 2c* where the sensor's coordinate system is determined based on a fixed reference device and the sensor head is manually moved and located relative to the fixed reference. This is still not suitable for inspecting some environments because the fixed reference shown in *Figure 2c* requires line of sight with the measurement tool. Thus this cannot be guaranteed in an unstructured environment.

All previous part inspection techniques required a degree of environmental control, such as a fixed reference frame, yet it is necessary to perform inspections in hostile environments. These hostile environments are not suited for environmental control; and thus, it is impossible to create a structured measurement system. Thus, a method must be developed to measure relative part features with an unknown reference frame through optical means. In *Figure 2d* a pipe inspection robot is used to perform inspections on a nuclear power plant, since the pipes cannot be removed and placed in a static reference frame, this robot creates its own reference frame during inspection. By doing this, all measurements are relative to a reference frame that is different for each set of measurements. This makes it possible to detect defects in environments where static reference frames are impossible, but it means measurements may not be consistent across multiple inspections, since the reference frame may be different.

1.2 – Problem Statement

Some characteristics of unstructured environments include the inability to create a fixed reference frame, and the large variety of surfaces that need to be measured. Many physically large or complex environments make fixed reference frames impossible due to either work volume constraints from the sensor or occlusions created by the environment. These environments also pose the difficulty of measuring a range of surfaces and measuring those surfaces at different ranges. Because of this, sensors may need to measure reflective and absorptive surfaces. In industry this is handled by thinly coating reflective surfaces but this may not be possible on a mobile robot. Unstructured robotic part inspection presents many challenges in the field of mobile robotics. Since there is no fixed reference frame, the robot must create a reference frame for all measurements while accurately localize itself within that reference frame to collect meaningful data. As well, the robot must be suitable for many measurement tasks; the surfaces it encounters may be unknown beforehand, and it needs to be

able to flexibly acquire measurements to account mobility restrictions imposed by the actuating platform. The design of such an instrumentation package is challenging due to size and flexibility constraints. This package needs to fit into a small space, while enabling multi-modal depth and visual sensing. The depth sensing needs to be flexible enough to measure a broad range of possible surfaces at different angles and distances.

1.2.1 – Thesis

The purpose of this Masters of Science thesis is to create and test a framework, and automated calibration tools for mobile robotic inspection through non-contact optical techniques. This framework will provide tools for creating a flexible mobile inspection platform tuned to a specific task of close-range depth measurement, while providing insight into how these tools can be tuned for different expected environments. The framework presented will include specifics on known algorithms and algorithms developed for this task while guiding the user towards usable implementations. This framework utilizes sensor fused structured light as a high accuracy, high flexibility measurement solution to solve all of the expected mobile robotics inspection problems. A three dimensional (3D) Simultaneous Localization And Mapping (SLAM) routine will be utilized to create a reference frame for all measurements and for localizing the sensor head in the environment. A structured light actuated profilometer will be used to create high accuracy reconstructions of detailed features. Calibration of all systems will be discussed; and equations and guidelines for parameter tuning will be provided.

The hardware used in this thesis has been chosen to present the robustness of the algorithms developed. Industrial optical non-contacts sensors are precision machined tools with high quality optics, but these industrial sensors are too large to be placed on an inspection robot. Thus small consumer products will be used that then can be calibrated to create high quality inspection tools. The advantages of this are twofold, primarily this allows for a great deal of flexibility in creating a system for a specific environment; but, this approach also greatly reduces the cost of the system. This cost reduction does come at a reduction in sensing quality, but this thesis will discuss the most useful areas for improvement.

1.2.2 – Justification

Currently, there is research into each individual component of this thesis; but these components come from a variety of fields that tends to inhibit knowledge acquisition. As well, there is little work into the use of profilometers in complex unstructured inspection environments. Thus the computer vision work for robust structured light correspondence is a unique contribution to the field. As well, there are no readily available tools for automated calibration of all parts of this thesis. All tools required for creating such a system have been developed for this thesis to work completely automatically, thus allowing for the processing of large amounts of data to create statistically backed calibrated models.

1.3 – System Overview

The inspection system developed for this thesis attempts to simulate a sensor rig that could be placed on a mobile robotic platform. The sensor rig includes one depth camera, in the form of

Microsoft Kinect, to gather contextual information about the whole scene. One visual camera, in the form of a Logitech HD920, which can be used for robot navigation, visual inspection and high accuracy structured light depth scanning. One servo motor will be used to simulate a robotic arm for positioning of the camera. One micro controller will be used to control the servo motor and control the lasers. Two red lasers are used in conjunction with the camera to perform depth scans.

The system needs to be able to acquire high accuracy depth measurements of surfaces to detect defects; as well as place those measurements into some kind of context so that inspectors can visualize the measurements. To do this the depth camera will perform Simultaneous Localization And Mapping (SLAM) to create a global reconstruction of the environment, while the visual camera will be used as a swept profilometer for high accuracy surface defect detection and characterization. The goal of this is to quickly create contextual maps and then improve those maps with small local scans with the swept profilometer as necessary.

As a further requirement, due to space constraints, the cameras used must be as multi-purpose as possible. The visual camera used for navigation must be usable as the inspection camera and the profilometer camera. To prevent disorientation due to tunnel vision, the visual camera should have a sufficiently wide field of view. It is assumed that tunnel vision can be prevented by using a camera with approximately the same field of view as human vision. This places a constraint that can be contrary to high resolution profilometry since a high zoom factor can be used to improve profilometer resolution.

1.4 – Thesis Overview

The paper will be structured as follows: a background review will be performed to identify the state of the art techniques used to create this system; as well as other potential solutions. Following that, Section 3 will discuss in detail the theory and implementation of the selected methods. Then, Section 4 will discuss the platform created as well as its characteristics and limitations. Section 5 will include a discussion on the testing methods and evaluation metrics which will be followed by section 6 discussing the results of various tests. The conclusion will be presented in section 7 where there will be a discussion on the benefits and drawbacks of the system as well as a proposal on future work.

2 - Background

A summary of the current state-of-the-art in 3D scanning and robotic inspection will be presented in this section. Important achievements in the field will be discussed as well as currently available solutions. Optical non-contact measurement techniques will be reviewed with respect to robotic unstructured environmental inspection. From this, an optimal technique will be identified and a detailed discussion of that technique will follow. Robotics and computer vision will also be discussed as necessary for the desired implementation. As a result, the reader should have an understanding of the topics necessary to reproduce this system, and sufficient knowledge to modify the implementation as needed.

2.1 - Optical Measurement Techniques

Optical non-contact depth measurement techniques are useful for robotic measurement because they guarantee that the robot will not damage the surface being measured. As well, they are particularly useful since the robot would need less actuation to view a defect than what would be required to make contact with a defect.

These optical measurement techniques include time of flight Light Detection And Ranging (LiDAR), optical interferometry, triangulation from stereo vision, depth from focus, shape from silhouette and triangulation from structured light. Each of these techniques has specific advantages with respect to cost, accuracy, working volume, reliability, and usability in unstructured environments.

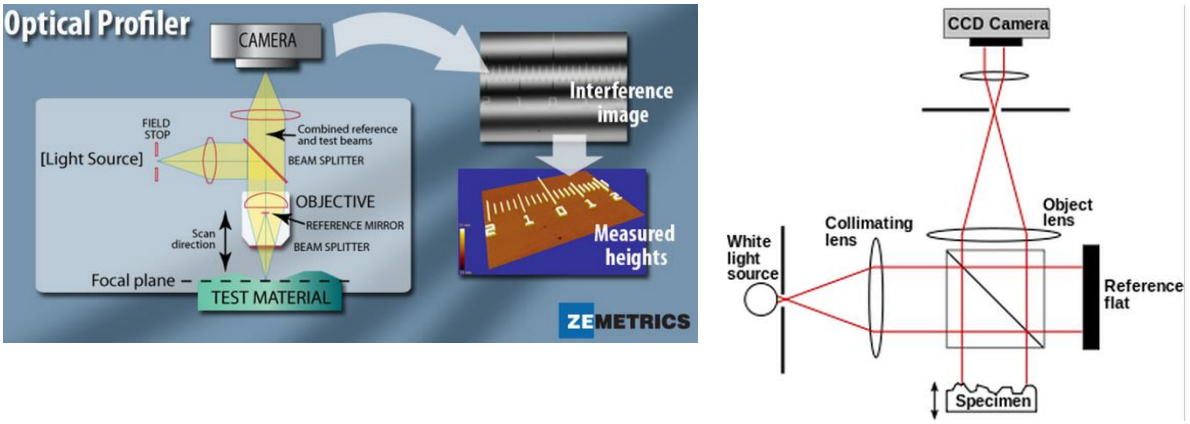
Optical Time of Flight sensors use concepts similar to Radio Detection and Ranging (RaDAR) to determine the distance to an object, thus they are generally referred to as Light Detection and Ranging (LiDAR) [12]. These sensors emit a short pulse of light and then wait to detect the return of the light from it reflecting off of a surface. The amount of time between emitting the light and perceiving the return is the total trip time of the flight that can be used to calculate distance via *Equation 1* [13]. This concept was first used in RaDAR as early as 1904 by Christian Hulsmeyer, who bounced radio waves off of ships to determine their distance [13].

$$D = \frac{C * T}{2} \qquad \textit{Equation 1}$$

Figure 3: Two types of industrial LiDAR sensors, Flash and scanning. Imaging, M. (n.d.). SR4000 Data Sheet. Retrieved January 01, 2014, from http://www.robotever.com/SR4000_Data_Sheet.pdf. Used with permission of MESA Imaging 2014. SiCK, "LMS221 outdoor LiDAR specifications," 2013. [Online]. Available: https://www.mysick.com/eCat.aspx?go=FinderSearch&Cat=Gus&At=Fa&Cult=English&FamilyID=344&Category=Produktfinder&Selections=34284,34258. [Accessed: 29-Oct-2013]. Used under fair use 2014.

There are two primary types of LiDAR, scanning and flash. Scanning LiDAR [15] sweeps a collimated beam of light into the scene. Since the beam is collimated, the intensity of the light does not significantly reduce with distances as the width of the beam does not expand significantly with distance. This allows for a relatively strong return at long distances while maintaining eye-safe output levels. However this means that to gather data, a scanning LiDAR must sweep a beam or multiple beams through the scene. Flash LiDAR [16] overcome this issue by emitting un-collimated flashes of light which can be measured by independently timed pixels. Since each pixel is independently timed relative to the initial flash, the time of flight for each pixel can be measured and thus the distance along the ray created by each pixel. However, the range of flash LiDAR is severely limited since the eye-safe output energy rapidly diverges with such a broad beam[14].

Optical interferometry is a high accuracy method of measuring small changes by looking at interference patterns of coherent light [17] [18], all forms of optical interferometry use the mixing of light through different optical paths to form these interference patterns. In the case of monochromatic interferometry, a single wavelength of light is projected onto a test material. The reflected light from that material is mixed with a reference beam. The interference between these two mixed beams is then viewed as constructive or destructive interference, which can be used to determine the focus of the test object [18]. The point of optimal focus is known as the zero Optical Path Difference (OPD) point. The mixed beam is input into a camera to obtain intensity values at each pixel. These intensity values can be analyzed to determine if a pixel is in focus or not. The test specimen or objective is then translated to determine optimal focus for each pixel that can be used to determine depth at each pixel [18][19]. A variation of this replaces mechanical translation with wavelength modulation to achieve the same effect [18]. Since monochromatic interferometry uses a single wavelength, the generated fringes are constrained to that which can be created by that wavelength. Because of this, steep gradients cannot be resolved due to high fringe density [18].



a b

Figure 4: Single wavelength interferometer, white light interferometer. Zemetrics, "Optical Profilers - how optical profilers work." [Online]. Available: http://zemetrics.com/optical-profilers-about.shtml. [Accessed: 29-Oct-2013]. Used under fair use 2014. Wikipedia. (n.d.). Interferometry. *Wikipedia*. Retrieved February 08, 2014, from http://en.wikipedia.org/wiki/Interferometry. Used under fair use 2014.

Multiple wavelength techniques have been introduced to overcome the fringe density issues with high gradient specimen. Multiple wavelengths can be combined together to create synthetic wavelengths of a much longer effective length, this process is similar to the formation of beats in audible sound waves. The fringes of this synthetic wavelength can then be used to resolve high gradient areas [18]. Another form of interferometry is the White Light Interferometer (WLI), this uses a large continuous band of light to a similar effect to previously discussed methods. The light source in this method can have low temporal and spatial coherence, which is accounted for in the design of the system. The zero OPD point can be determined by the maxima of fringes, since this is the only point where all components of the light source will have constructive interference. The design of such a system has the advantage of flexible light source selection, since a high coherence singular wavelength source is no longer necessary; whereas since the light source cannot be modulated, it requires translation of either the specimen or the objective [18]. Many other forms of interferometry exist and new forms are constantly in development, but their large size and limited working range [18] make them unsuited for mobile robotic applications.

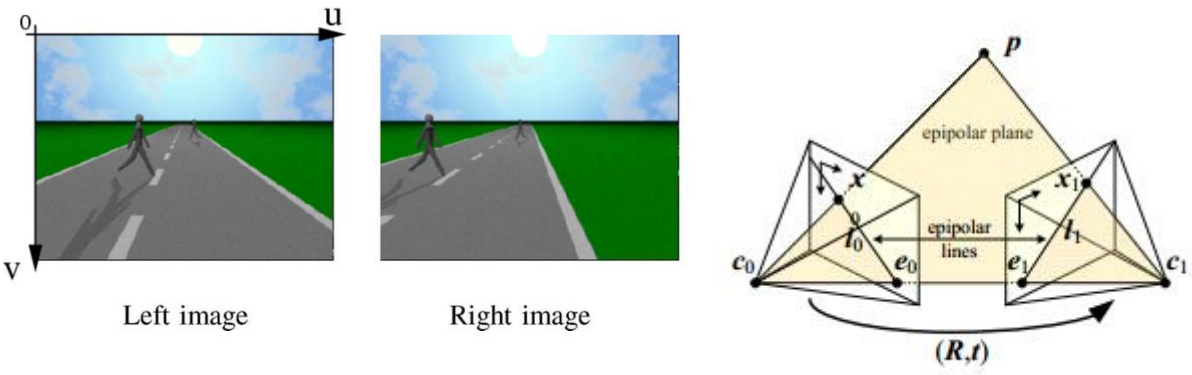


Figure 5: Example of stereo vision. *C. Caraffi, S. Cattani, and P. Grisleri, "Off-Road Path and Obstacle Detection Using Decision Networks and Stereo Vision," IEEE Trans. Intell. Transp. Syst., vol. 8, 2007.* Used under fair use 2014. *R. Szeliski, Computer vision: algorithms and applications, 1st ed. Springer, 2011, p. 979.* Used under fair use 2014.

Triangulation based stereo vision measures depth by placing two cameras next to each other and then measuring the triangle created by a common point in each image. This method of triangulation is similar to that of structured light except that it is a passive method that relies on point correspondences [22][23]. Since stereo vision uses triangulation, it is similarly powerful to structured light with respect to tunable working volumes and depth resolution; yet its passive nature makes it problematic in inspection environments with uniform texture.

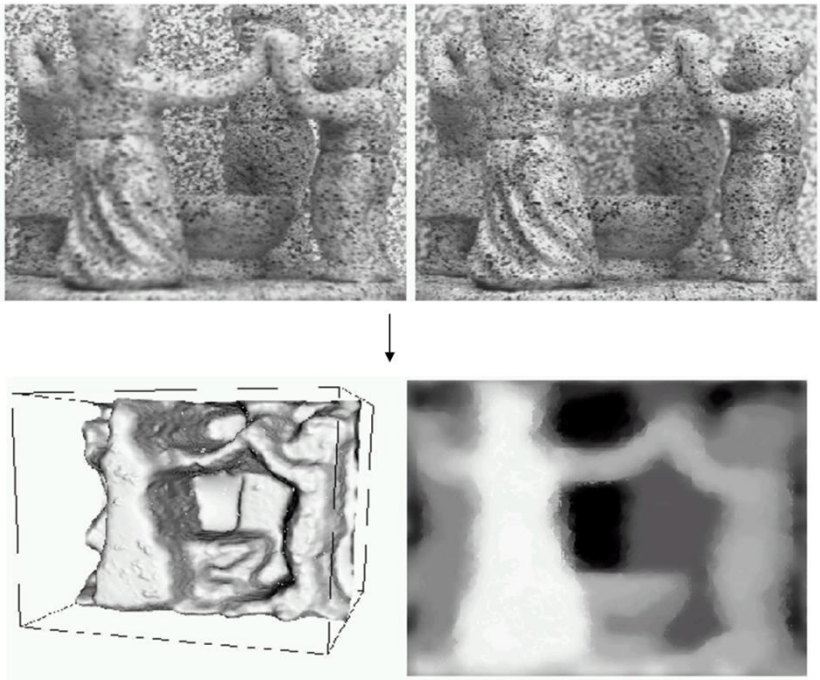


Figure 6: Shape from focus *P. Favaro, "Depth from focus/defocus," 2002. [Online]. Available: http://homepages.inf.ed.ac.uk/rbf/CVonline/LOCAL_COPIES/FAVARO1/dfdtutorial.html. [Accessed: 07-Nov-2013].* Used under fair use 2014.

Depth from focus relies on changing the focus of the camera and observing the scene as it moves in and out of focus [25][26]. As the focus is swept through its range, different parts of the scene can transition into and out of focus. Thus this can be seen as slicing through the scene to create a reconstruction. This method is limited by the parameters of the lens that determine the depth field, and the ability to algorithmically determine the focus of a pixel. As such, surface characteristics play an important role that is un-controllable. Methods for measuring focus have been developed by looking at texture [27], but these methods do rely on texture, which is not guaranteed in inspection environments.

Shape from silhouette utilizes a uniform background light to image just the silhouette of an object. The object is then rotated on a stage and multiple silhouettes can be combined together to create a profile of the object [28]. This method requires a set environment for scanning and, thus, is also not suitable in-situ inspection.

Structured light utilizes triangulation to measure the point of an object with respect to the camera by emitting a known light pattern into the scene and observing the returned reflected light. The camera can determine the angle of return and, thus, a triangle can be created to determine the point's location [29]. Since structured light is the main focus of this thesis, further discussion will follow.

Technique	Requirements	Advertised Resolution	Min working range	Max working range
Triangulation from Stereo Vision (Point Grey Bumblebee XB3) [30]	Corresponding points in environment	1.6 mm @ 0.5 m	92 mm	29.5 m
Depth from focus	Textured specimen	Depends on optics and auto focusing mechanism	Tunable	Tunable. Cannot measure past infinity focus point
Shape from silhouette	Controlled environment	High. Controlled by table encoders and camera.	Profile of object needs to be in field of view	Lower resolution of large parts far away from camera.
LiDAR				
Flash LiDAR (SR4000)[14]	Non-reflective surface if at angle	4 mm. Constant resolution over working range.	0.8 m	5.0 m
Scanning LiDAR (HDL-64E) [31]	Non-reflective surface if at angle	2 cm. Constant resolution over working range.		> 100 m
Interferometers				
White light interferometer (ZeGage 2.5x objective) [32]	Specimen placed on precision stage	3.5 nm	10.3 mm	33.3 mm
Structured Light 3D				
Kinect [33]	Non-reflective surface if at angle	2 mm @ 0.5 m	0.5 m	6 m
Structured Light Profilometer				
Mitutoyo Optical CMM [34]	Specimen placed in CMM workspace	12 μm	30 mm	60 mm
LMI RoLine 1130 [35]	Laser is normal to surface	10 μm	200 mm	400 mm

Table 1: Depth sensor comparisons.

Passive techniques are inadequate in inspection environments since they rely on scene texture for corresponding points. Such techniques include passive triangulation from stereo vision and depth from focus. In inspection environments, scene texture is likely available, but not guaranteed which is why these methods should not be used. Thus among the active methods the prime contenders are structured light, Time-of-Flight, and optical interferometry. A brief look at time of flight sensors indicate that they are generally designed for longer ranges than most inspection environments. As well, they have limited accuracy due to limitations in timing circuits. And lastly, the Time-of-Flight depth cameras have relatively low resolution (176x144 for the SR4000), which makes SLAM difficult. Optical interferometry is on the opposite end of

the accuracy and working volume spectrum, since it is extremely accurate, but has an extremely small work volume. It is this small work volume that makes optical interferometry impossible to use in inspection robots, since environmental restrictions will likely prevent use of an interferometer in the required position. As a note, since the interferometer images only a small volume, it would take the place of the profilometer; thus, it would need to be coupled with a depth camera for SLAM.

2.2 - Structured Light – 2D

Structured light is the main focus of this thesis due to the low cost of this solution along with its flexibility [36]. It is low cost, since it only requires a camera and a light source; and, it is flexible because the work volume and resolution can be tuned to provide excellent performance for specific applications [36]. Structured light works by actively emitting light into a scene, then observing the return of that light by a sensor that can measure the angle of return. By placing the emitter and receiver at an offset, the receiver can measure an angle of return. This angle can be used to construct a triangle when calculating the 3D point of an object in the scene. This is generally done with a planar light source and a camera, as seen in *Figure 7*.

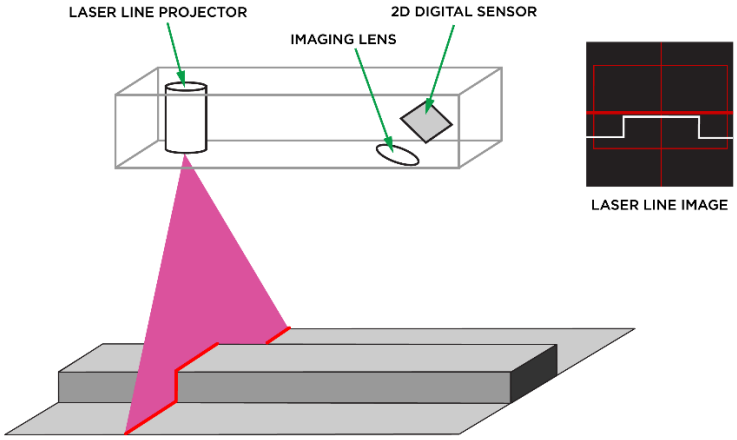
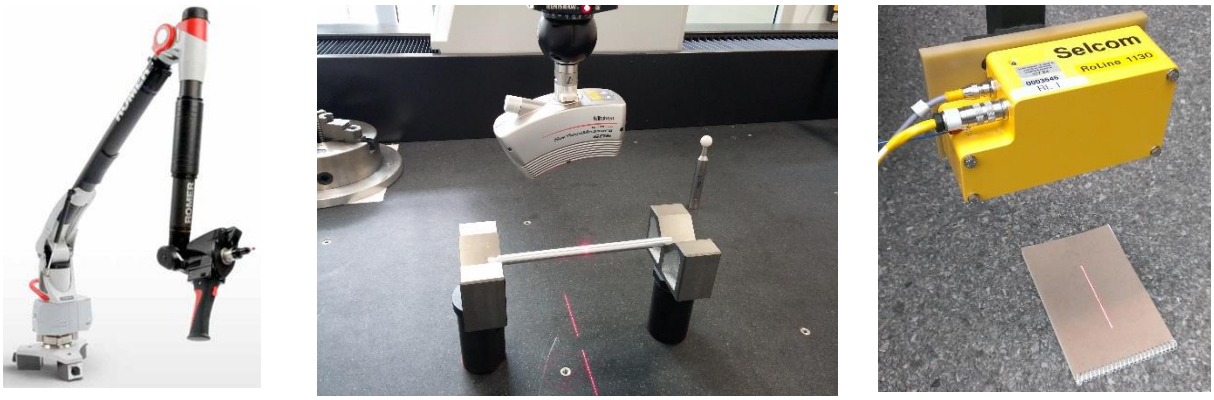


Figure 7: Profilometer imaging a surface. Laser line is projected onto surface and camera images reflected light. Used with permission of Glenn Hennin from LMI 2014.

A fundamental task in structured light is to determine which return is from the emitter and thus to reject any returns from ambient lighting, this task is known as correspondence. Correspondence can be a rather simple task with trivial light sources, a high power laser plane can be found by thresholding out all other light sources, and using a line finding algorithm [38]. Correspondence however becomes much more difficult with the use of more complicated light patterns; or when ambient light causes a noisy return [29].

Structured light is used by Mitutoyo to create their 3D scanning line of products to create digitized 3D reconstructions as an attachment to their CMMs [34]. Exact Metrology uses it in their Romer absolute arm SI and Perceptron Scanworks lines of products for part inspection and reverse engineering [39]. Selcom (purchased by LMI technologies) uses it in their RoLine family of products to measure surface defects and cracks in roads [37]. Mitutoyo achieves an advertised 12 µm depth resolution with a working range of 30 mm for $30,000 with the

SurfaceMeasure 606 attachment for their CMMs [34]. Exact Metrology's Romer absolute arm SI scanner achieves 0.03 mm accuracy with a working range of 100 mm [40]. LMI technologies' RoLine 1130 achieves an accuracy of 0.01 mm with a working range of 200 mm and at a cost of $30,000 [35]. As can be seen, there is a significant range of price points, applications, and working volumes.



a b c

Figure 8: Romer Absolute Arm *H. Metrology, "ROMER Absolute Arm," Product Brochure, 2012. [Online]. Available: http://www.hexagonmetrology.us/images/pdf/AbsoluteArmBrochure2012vFinal-web2.pdf. [Accessed: 09-Oct-2013].* used with permission of Hexagon Metrology 2014. Mitutoyo Surface Measure 606, personal photo. RoLine 1130, personal photo.

The Romer Absolute Arm SI is particularly interesting because the scanner itself has a relatively small working volume to achieve high depth accuracy; yet it maintains a large overall volume by attaching the sensor to a seven degree of freedom non-actuated arm. Each degree of freedom in the arm is sensed by an absolute encoder that allows for the precise calculation of the position of the sensor at the end of the arm.

All three of these solutions use a swept plane method of 3D scanning, ie: a single plane of light is projected into the scene and a plane of 3D coordinates is extracted. The plane is then swept through the scene to create a 3D reconstruction. These all require a form of precise motion sensing, Mitutoyo uses linear motion systems with encoder feedback to automate the process; Scanworks uses an articulated arm with encoders at each joint; and LMI uses wheel encoders on a vehicle to infer motion. Due to this, they all require extensive additional software, calibration and processing.

2.3 – Two Dimensional Structured Light Correspondence

An important topic is the use of the visual camera as the structured light profilometer. This allows for the reduction in sensors since a visual camera is necessary for navigation and visual inspection for the operator; although this adds the complexity of using a general purpose color camera as a structured light camera. The use of a visual camera as a structured light camera requires a form of software filtering to correctly select the laser line. In the industrial structured

light profilers discussed previously narrow bandpass filters and lasers with a very specific wavelength of light are used, thus allowing for a high rejection of unimportant light sources. This obviously is impossible to implement while maintaining the visual camera aspects required for an inspection robot. Instead of using physical correspondence techniques, it is necessary to use mathematic correspondence techniques.

The idea of correspondence for 2D structured light system, is to locate in the image the pixels that correspond to the return from the laser. Simple thresholding techniques can be used to look for the specific color, or hue of the laser but these techniques can produce a significant number of false positives. In order to improve upon this, advanced perception techniques have been applied such as Support Vector Machines, to correctly classify pixels as being part of the laser line [38].

Machine learning classification algorithms are extremely useful for imparting intuition into a computer program. Such algorithms can take complex data and form meaningful outputs with relative ease given appropriate learning algorithms. Classification is the task of taking input data and automatically selecting a user generated label for that data [41]. In this case, the input data is a set of features derived from image information and the output is whether or not that data represents the laser line. In terms of machine learning, there are several classification algorithms; some of them are as follows: Naïve Bayes, Logistic regression, Support Vector Machines, and K-nearest neighbors.

Naïve Bayes is a probabilistic counting algorithm. To train this algorithm, an input set of data is discretized into bins. A probability matrix is then created such that each column is a feature and each row is a bin of that feature. Then, the probability of a new object being of a certain class is determined by extracting probabilities out of the probability matrix given the discretized bins [42]. The downside of this is the need for input data. If a given dataset is not representative, then a new entry may not be represented by the probability matrix, and thus the algorithm would return a zero probability [42].

Logistic regression is a classification algorithm very similar to linear regression. It relies on calculating a weight vector to be applied to the feature vector to calculate an output. Instead of returning the dot product of the weight vector and feature vector, this algorithm finds the result and then converts it to a probability via the sigmoid function [43].

Support Vector Machines attempt to create a decision boundary which segregates classes. This boundary attempts to segregate all of one class on one side and all of the other class on another side [44]. This decision boundary is trained to maximize the margin the two classes. The decision boundary is determined by finding ‘Support Vectors’ that define the margin. This can be seen in *Figure 9*, where a decision boundary (black line) is chosen such that the margin space between the decision boundary and both classes is as large as possible.

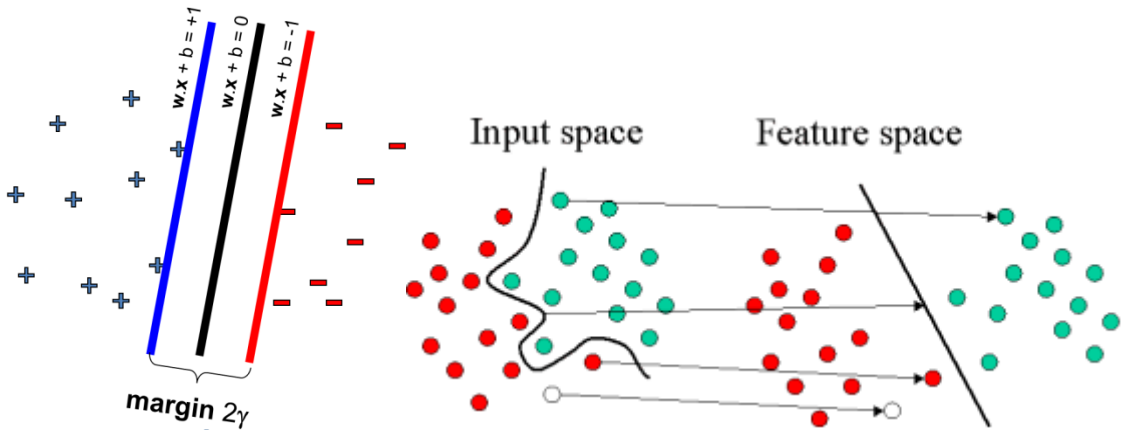


Figure 9: Support Vector Machines. *C. Guestrin, "Support Vector Machines," 2005. [Online]. Available: http://www.cs.cmu.edu/~guestrin/Class/10701-S06/Slides/svms-s06.pdf. [Accessed: 26-Feb-2014].*Used under fair use 2014. *StatSoft, "Support Vector Machines (SVM)." [Online]. Available: http://www.statsoft.com/textbook/support-vector-machines. [Accessed: 21-Nov-2013].* Used under fair use 2014.

Support vector machines (SVM) are very powerful learning algorithms, due to their ability to learn complex decision boundaries by moving to complex feature spaces for the decision process [46]. Support vector machines make use of complex feature spaces with little performance penalty, due to what is known as the kernel trick [46].

K-nearest neighbors is one of the simplest to implement Machine Learning algorithms. Training data is stored and then given new input data, the closest neighbor to that data is selected from the training data set, and the label is copied. An extension of that is to select the most found label from the K-nearest neighbors to the new data point, where K is a user selected variable [47].

2.4 - Structured Light – 3D from Actuated 2D Scans

Two dimensional structured light can create a single profile of an object at the intersection of the laser beam and the object. In order to create meaningful 3D reconstructions, it is necessary to gather multiple profiles. To do this, the 2D structured light sensor is actuated in a known manor that allows for the concatenation of multiple profiles into a 3D reconstruction. The process behind this requires a discussion on homogeneous transformations and robot kinematics.

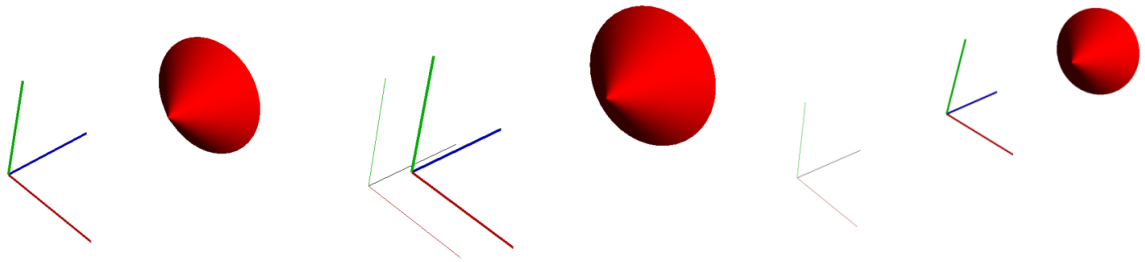
Suppose there is a sensor with its own of coordinate system, it would have its own X, Y, and Z axis and units associated with the sensor. This local coordinate system is for the sensor and anything that it measures will be relative to the sensor. Thus, it is often necessary to relate what a sensor sees to another coordinate system to allow a robot knowledge of what it sees. In order to do this, it is convenient to transform information from one coordinate system to another, such as transforming sensor data from the camera coordinate system to the robots coordinate system to allow path planning. This transformation from one coordinate system to the other is a fundamental component of 3D reconstruction and robotics as a whole [48]. Let us defined the vector notation convention used in this paper as follows:

- Lower case, standard font – scalar
- Upper case, italicized – Vector
- Upper case, bold – Matrix
- Superscript, original coordinate system
- Subscript, new coordinate system
- Superscript T, transpose of matrix or vector

Let us look at an example of such a transformation, the vector *P* is a 3D Cartesian coordinate in the camera coordinate system.

$$P^{camera} = \begin{bmatrix} x \\ y \\ z \end{bmatrix} \qquad \textit{Equation 2}$$

This coordinate is represented by the cone in *Figure Figure* 10*a*. The origin of the camera coordinate system can be seen in this frame as the red, green, and blue lines. Now, another coordinate system can be defined external to the camera coordinate system. This can be seen in *Figure 10b*, as the thinner lines. In this case, the camera coordinate system can be moved relative to the global coordinate system, such that objects seen in the camera coordinate system also move relative to the global coordinate system.



a b c

Figure 10: Coordinate systems and coordinate system transforms.

This is achieved by a linear transformation. A linear transformation is as simple as adding the x, y, and z values of the sensor coordinate system relative to the global coordinate system, to the values of the object in the sensor coordinate system, as per *Equation 3* [49] [50]. In *Equation 3*, T is a column vector of the same size as P that describes the camera coordinate system with respect to the global coordinate system. This transformation is illustrated in *Figure 10* where there is a camera coordinate system relative to a global coordinate system in *Figure 10b*, and then the camera undergoes a translation as seen in *Figure 10c*.

$$P^{Global} = P^{Sensor} + T^{Sensor}_{Global} \qquad \textit{Equation 3}$$

A rotational transformation can be trickier, due to the fact that the final coordinate along an axis of the new coordinate system is dependent on the projection of all of the coordinates from the previous coordinate system. An example of this is the 2D case where a point P's new coordinates are dependent on both the x and y values in the previous coordinates, illustrated by *Equation 4* [51].

$$Px = Px * \cos(\Theta) - Py * \sin(\Theta)$$
$$Py = Px * \sin(\Theta) + Py * \cos(\Theta) \qquad \textit{Equation 4}$$

The natural extension of this is to include all three axes; this is conveniently done with a rotation matrix [52]. A rotation matrix can simplify representation of a transformation by representing a rotation by a single matrix operation. Rotation matrices can take three general forms, one for each axis of rotation, as seen in Equation 5, Equation 6, and Equation 7 for the x, y, and z axis respectively.

$$\boldsymbol{R} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\Theta) & -\sin(\Theta) \\ 0 & \sin(\Theta) & \cos(\Theta) \end{bmatrix} \qquad \textit{Equation 5}$$

$$\boldsymbol{R} = \begin{bmatrix} \cos(\Phi) & 0 & \sin(\Phi) \\ 0 & 1 & 0 \\ -\sin(\Phi) & 0 & \cos(\Phi) \end{bmatrix} \qquad \textit{Equation 6}$$

$$\boldsymbol{R} = \begin{bmatrix} \cos(\alpha) & -\sin(\alpha) & 0 \\ \sin(\alpha) & \cos(\alpha) & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad \textit{Equation 7}$$

If there are two coordinate systems with a rotation between them, forward multiplication of a vector in one coordinate system will transform it into the equivalent vector in a new coordinate system, as seen in *Equation 8* [52]. This rotation is illustrated in *Figure 11*.

$$P^{Global} = \boldsymbol{R}^{Sensor}_{Global} * P^{Sensor} \qquad \textit{Equation 8}$$

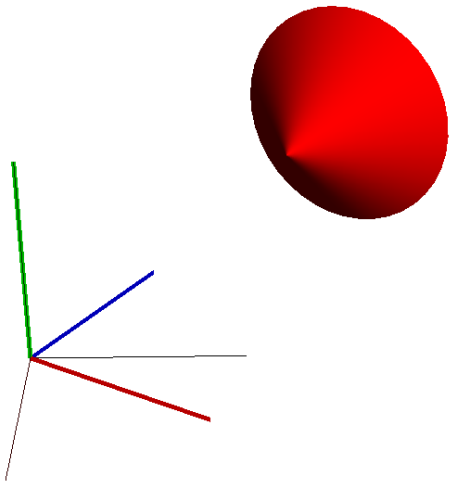


Figure 11: Rotation of an object in sensor coordinate system.

In general, two coordinate systems will not be parallel and will not lie on top of each other, thus a transformation between two coordinate systems will usually require a rotation and translation. This operation can be done as seen in *Equation 9* as a rotation of the coordinate systems, until they are parallel, and then a translation [53].

$$P^{Global} = \boldsymbol{R}^{Sensor}_{Global} * P^{Sensor} + T^{Sensor}_{Global} \qquad \textit{Equation 9}$$

This operation is so common that a common performance optimization is to append a one onto the P vector and combine the translation and rotation to allow Equation 9 to be executed in one matrix operation [54].

$$P^{camera} = \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \qquad \textit{Equation 10}$$

$$\boldsymbol{H}^{Camera}_{Global} = \begin{bmatrix} & R & & T \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad \textit{Equation 11}$$

$$P^{Global} = H^{Camera}_{Global} * P^{Camera} \qquad \textit{Equation 12}$$

As can be seen in *Equation 11*, the new **H** matrix contains the rotation component and the translation component of the transformation, this matrix is generally referred to as a homogeneous transformation matrix [54], thus why it is denoted **H**. A very convenient aspect of this is its ability to combine many transformations together into a stack of matrices, and representing all of those transformations as one new homogeneous matrix.

$$\boldsymbol{H}^{Sensor}_{Global} = \boldsymbol{H}^{Joint\ 1}_{Global}\boldsymbol{H}^{Joint\ 2}_{Joint\ 1}\boldsymbol{H}^{Sensor}_{Joint\ 2} \qquad \textit{Equation 13}$$

If someone wants to know the location of an object measured by the sensor, with respect to the global coordinate system, one could calculate the homogeneous transformation matrices for each joint, multiply all of the joint matrices together and create one transformation for the location of the sensor relative to the global coordinate system. This becomes extremely useful due to another useful feature of the homogeneous transformation matrix. To calculate the reverse transformation, from global to sensor, one only needs to calculate the sensor to global matrix and then calculate the inverse of that matrix [55]. This is incredibly useful for certain post processing techniques that require the sensor position. Such as mapping an image from a camera back onto a global point cloud for texturization and surface area measurement [56].

Three dimensional point cloud reconstruction from a 2D structured light scan plus motion should be relatively straight forward, given known joint positions and the power of homogeneous transformations. At any given image, a 2D structured light setup can return a point cloud of data on the laser plane. If that structured light system is attached to an actuator, the laser plane can be swept through the scene. If the joint positions of the actuator are known, the locations of the sensor relative to the actuator, and the actuator relative to the global coordinate system are known, then the point cloud from the structured light system can be calculated relative to the global coordinate system. A 3D reconstruction can thus be created by concatenating point clouds from the structured light sensor at a series of joint positions. An example of this can be seen in *Figure 12*, where a vertical plane was swept across a sugar box. The power of this system can be seen in the fact that the SD card attached to the front of the sugar box can be clearly seen in the point cloud, and can be measured accurately. The system in question could measure the depth of a 3.3mm thick SD card with less than 10% error.

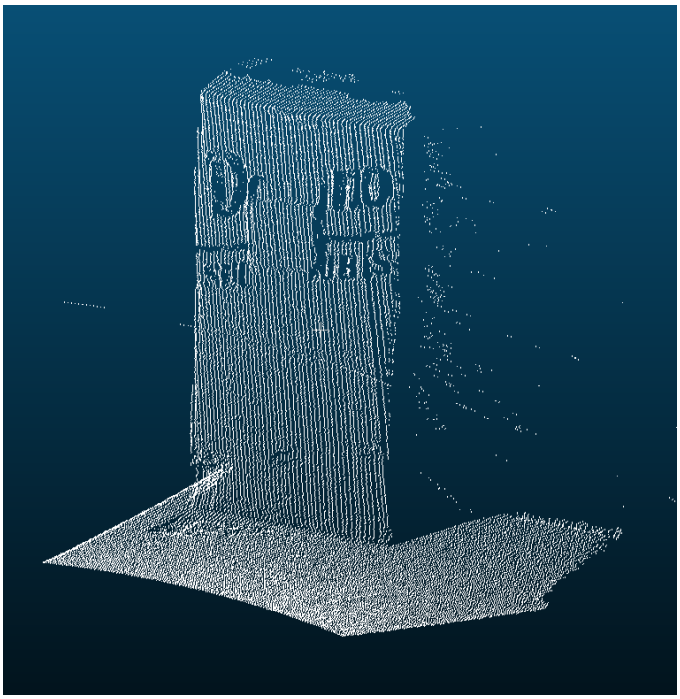
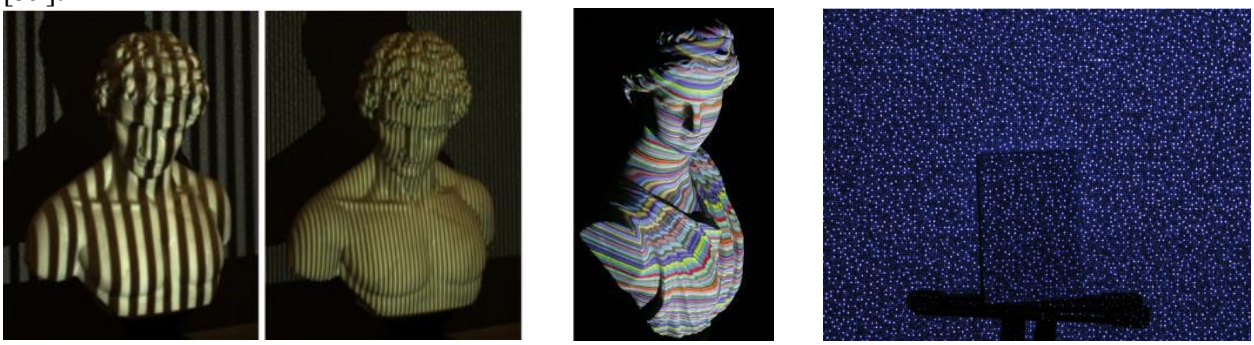


Figure 12: Actuated profilometer reconstruction.

2.5 - Structured Light – 3D

Three dimensional structured light is useful because it allows for contextual depth information by gathering depth from the whole scene simultaneously. This contextual information is very useful for localizing the robot, as well as providing a visual context for the operator to understand where a defect is.

Three dimensional structured light improves on 2D structured light reconstructions techniques by removing the use of actuators, and thus, potentially has higher accuracy. But 3D structured light also introduces more difficult correspondence than in 2D structured light. This arises from the fact that the software must be able to associate a pixel from the image with the correct ray from the light source. The state of the art methods for this include, time encoding of the light pattern to allow correspondence matching of target pixels based on observing how the pattern changes over time [57]. Pixel color encoding allows for encoding of regions based on color, assuming uniform scene color and reflectance [58]. Pattern based light encoding emits a specific pattern of light, since the original pattern is known, the deformation of the pattern can be calculated. The deformation of the pattern can then be used to calculate the depth of the scene at any given point [59].



a b c

Figure 13: Time, color, and spatial encoding of 3D structured light *D. Lanman and G. Taubin, "Build Your Own 3D Scanner : 3D Photography for Beginners," Siggraph, 2009. "Temporal Dithering of Illumination for Fast Active Vision." [Online]. Available: http://staff.aist.go.jp/shun-yamazaki/research/dlp/. D. Hoiem, "How the Kinect Works Kinect Device," Computational Photography, 2010. [Online]. Available: http://courses.engr.illinois.edu/cs498dh/fa2011/lectures/Lecture 25 - How the Kinect Works - CP Fall 2011.pdf*. Used under fair use 2014.

Time encoding has the advantage of high correspondence likelihood and high scene noise rejection; yet, it has the disadvantage of taking more time to complete a scan, and requires a projector and time syncing of the projector output with the camera input [57]. Color encoding is promising, since it allows correspondence of the entire scene at any given time, thus allowing motion capture. Yet, color encoding is susceptible to scene noise, as different objects may reflect different wavelengths of light more strongly and thus change the perceived correspondence. As well, color based encoding requires an expensive and large projector, which is undesirable in mobile robotic platforms [58]. Pattern based encoding is promising, since it creates a set pattern

that does not change over time, and does not rely on the color of the object being uniform. It only relies on relatively similar scene reflectivity. Pattern based encoding does not require a projector; thus, a static light source and optics can be used for a significant cost and size reduction [59].

2.6 - Microsoft Kinect – Improved 3D Structured Light

The Microsoft Kinect improves on the 3D structured light techniques previously discussed by using custom hardware and intelligent algorithms to improve the computational speed of 3D structured light algorithms. This allows for real time structured light. The Microsoft Kinect uses a pattern based encoding scheme combined with a patented astigmatic lens that allows for depth from focus [62]. This allows it to use the depth from focus to obtain an approximate depth of an area, and then use that to speed up the correspondence estimation. This approach is the key behind the Kinect's real time depth mapping. The Kinect's pattern projector outputs a light pattern of perfectly round dots, the astigmatic lens in-front of the IR camera causes a stretching of the dot into an ellipse with orientation associated with the depth from the sensor as seen in *Figure 14*. This ellipse orientation allows for an approximate depth to be determined, and thus can be used to set the bounds on the correspondence calculation.

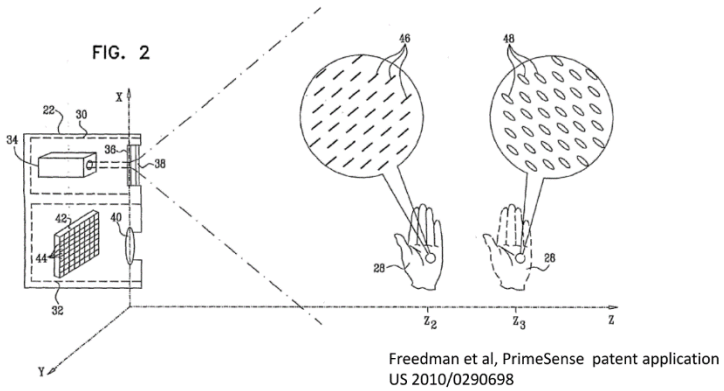


Figure 14: Depth from focus via astigmatic lens [62] Y. A. Barak Freedman, Alexander Shpunt, "Distance-Varying Illumination and Imaging Techniques for Depth Mapping," US20100290698 A118-Nov-2010. Used under fair use 2014.

The quality of the Kinect point cloud was found to not contain significant systemic error, when properly calibrated [64]. Random error starts small at less than half a centimeter at its closest range and it increases quadratically until it reaches its maximum random error of 4 cm at the sensor's maximum range of 5 m [64]. The Kinect's point clouds suffer from two corruptions, the first is occlusion caused by the IR sensor not seeing what is illuminated by the projector, or the projector not illuminating what the IR sensor sees; this corruption can be seen as holes in the resultant point cloud [65]. The second corruption is caused by over exposure of the IR sensor, if an object is too close or too reflective, parts of the IR sensor can be over exposed, and a correspondence cannot be determined [64].

2.7 - Camera Models

As a starting block for all camera based structured light problems, it is necessary to determine the camera parameters for the model being used. The most used and studied model is the pin hole camera model that was developed by Leonardo Da Vinci [66]. This camera model is known as a pin hole focusing element that projects light from a scene onto a surface. The parameters describing the basic pinhole camera include the following:

- Focal Length (f)
- Principal point (Cx, Cy), u_0, v_0 in *Figure 15*

These parameters are based off of the model seen in *Figure 15* which is a modified more general pin hole camera model.

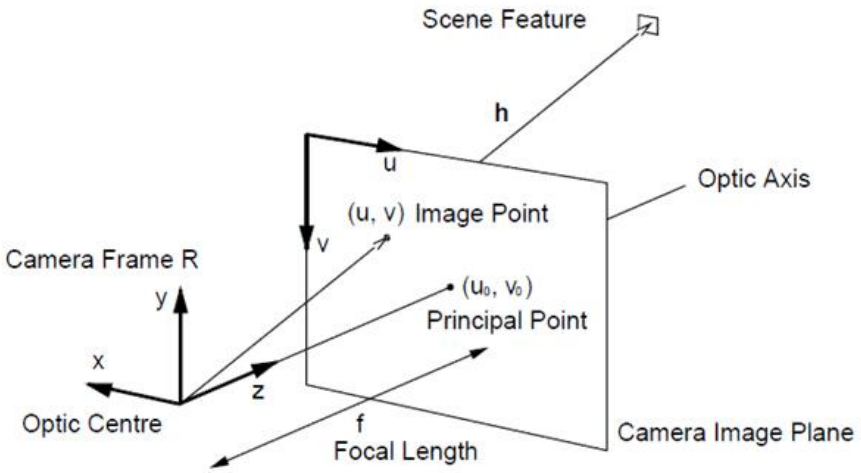


Figure 15: Modified pinhole camera model *G. B. S. D. A. Murguet, "MonoSLAM," 2012. [Online]. Available: http://www.ensta-paristech.fr/~filliat/Courses/2011_projets_C10-2/BRUNEAU_DUBRAY_MURGUET/monoSLAM_bruneau_dubray_murguet_en.html. [Accessed: 19-Feb-2014].* Used under fair use 2014.

In this camera model the object M in 3D space has coordinates m in camera space due to a projection onto the imaging plane. The coordinate system of the camera is defined with X going to the left and Y going downwards, through the optical axis. The optical axis is the axis where the central ray of the camera extends outwards from the projection center through the central point.

The traditional pinhole camera model can be seen in *Figure 16*, where the focal point is placed in-front of the image plane. This corresponds to traditional models where the image on the sensor is flipped and mirrored. This corresponds to the physical manifestation of the camera. The model seen above is still a pinhole camera model, accept it places the focal point behind the image plane, thus the scene is not mirrored or inverted. This represents what our eyes see and the typical output of a camera, as the camera automatically inverts and mirrors the image to be

the expected orientation. In this model, the focal length defines the distance between the image plane and the point where the rays converge. The principal points defined the location of the focal point relative to the image sensor origin.

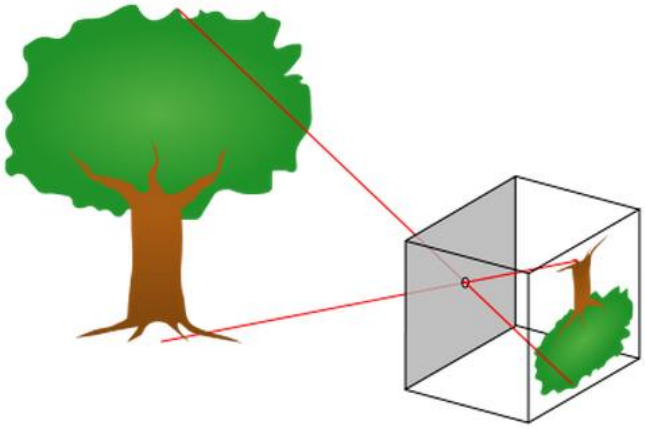


Figure 16: Traditional pinhole camera model *"File:Pinhole-camera.svg - Wikipedia, the free encyclopedia." [Online]. Available: http://en.wikipedia.org/wiki/File:Pinhole-camera.svg. [Accessed: 03-Mar-2014]*. Used under fair use 2014.

The pinhole camera model is frequently used in structured light applications because it is a linear projection, which thus makes it easily invertible for the reconstruction of a 3D point given the points observed on the image sensor. Another camera model is the two plane model. The two plane model represents the camera sensor as a set of two different planes, where the image coordinates found on the sensor are put into two equations which output coordinates on the two planes. This allows the creation of a ray through those two points, associated with the observed pixel on the image sensor. This model is inherently more complex as it introduces parameters for two different planes, but it is also more accurate due to its improved ability to model lens distortion.

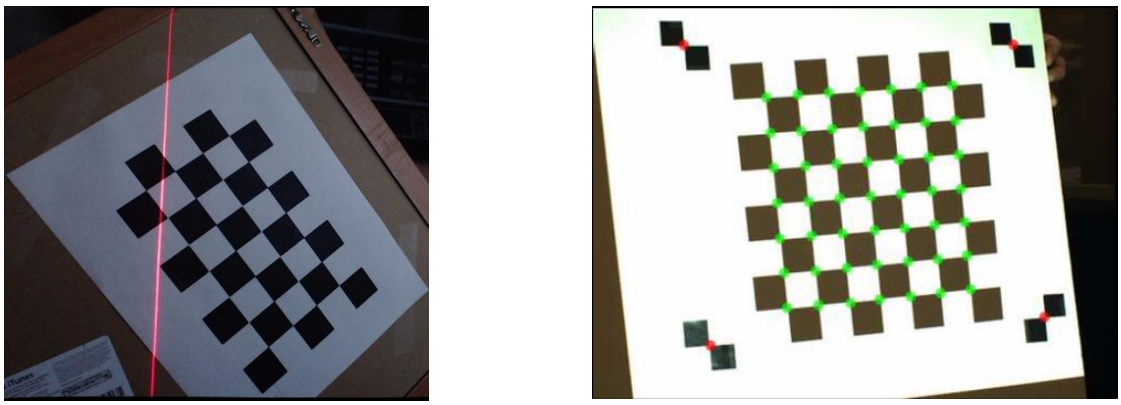
Lens distortion can be modeled with the pin hole camera model by including radial and tangential distortion coefficients. These coefficients are generally labeled k_1, k_2, p_1, and p_2 where the k terms are the coefficients to the second and fourth order radial terms respectively; and the p terms are associated with the first and second order tangential terms. These terms can be used to create an undistorted image, otherwise known as an image of what the image sensor would see if there was no lens distortion.

2.8 - Light Source Calibration

In order to gather measurements from a structured light camera, the light source has to be calibrated so that the light source orientation is known relative to the camera. Calibration of the light source varies based on the type of source and pattern used. As previously discussed, there are different types of light sources, and they all have varied calibration techniques. A profilometer only needs to estimate the parameters of the planar light source with respect to the camera. If the plane of light is structured implicitly, it is defined by a point on the plane and the normal vector to the plane, as such [69]:

$$Plane: n^t * (p - q) = 0 \qquad \textit{Equation 14}$$

In *Equation 14*, n is a vector pointing in the direction normal to the surface of the plane, p is a point on the plane and q is another point on the plane. Several techniques have been proposed to finding the proper values to this plane. It is possible to project the light source onto a well know 3D object, and optimize the values of the plane based on known features of that object. This method requires a precision object, as well as a good measurement of where the object is with respect to the camera [70]. Another proposed technique is to use other known features in the scene to obtain the pose of a known plane and then the intersection of the known plane and the unknown light source can be used to calculate the parameters of the light source [71] [72].



a b

Figure 17: Calibration of structured light laser and projector *D. Lanman and G. Taubin, "Build Your Own 3D Scanner : 3D Photography for Beginners," Siggraph, 2009.* Used under fair use 2014.

Once a camera is calibrated, a well-defined planar object can be found using vision processing techniques, and then the coordinates in 3D space of the object can be calculated based on the pixel locations and the known parameters of the object, as can be seen in *Figure 17*. When using a checkerboard calibration pattern, the 3D points of the plane can be found relative to the camera. Three dimensional points from the laser plane can be extracted relative to the camera at the intersection of the laser plane and the checkerboard plane [71]. Three dimensional points from multiple plane-plane intersections can then be used to statistically estimate the laser plane. This technique allows for a significantly cheaper calibration pattern, as a printed checkerboard pattern can be used on a rigid planar object for a cost of less than $5.

2.9 - Motion Calibration

Previously 3D scene reconstruction was discussed through using a 2D scanner and motion. In order to create an accurate reconstruction, it is necessary to know where the camera is as the actuator moves it. Since the camera coordinate system will not align with the actuator coordinate system, there will be a constant offset between the two. Calibrating this offset is known as Eye-in-hand calibration, as it originates from the idea of having an 'eye' meaning camera or other sensor, at the end effector of a 'hand' meaning robotic manipulator [73].

Previously homogeneous transformation matrices were discussed in great detail. The joint position inputs to those matrices are relatively easy to retrieve, a joint can have an encoder placed on it and that encoder can output an absolute angle value. A much more difficult to obtain parameter is the x, y, and z translation between the camera coordinate system and the robotic manipulator. In most cases, a simple measurement can be made, the distance between the estimated camera origin and the axis of rotation of a motor can be used to estimate the lever arm length; yet the accuracy of a manual measurement is not sufficient for measurement grade structured light scanners. Manual measurements are particularly in-accurate due to the difficulty

in locating the camera coordinate system and measuring that location relative to the motor's axis of rotation. This can be due to the camera coordinate system being misaligned inside of its housing; as well the camera housing could be mounted at a difficult to measure angle relative to the motor. An ideal technique would utilize the values of the sensor output along with known input values to create an accurate calibration, thus removing the need for physical access to the coordinate systems in question.

One such technique involves using a known calibration object and performing calibration based on minimizing the error between the known point cloud and the sensor output [70]. A similar technique is to use a small target object, such that the centroid of the object can be calculated in 3D space at various locations, and then multiple sets of data can be used to iteratively solve for the joint parameters [74]. These techniques rely on an accurate knowledge of the target point, with either an absolute accuracy relative to the global coordinate system [70], or many measurements and an estimated location [74]. Another downside to the previously mentioned techniques is the solution of multiple non-linear equations, which is time consuming and prone to local optima.

A more sophisticated technique is to use a known static calibration pattern and observe the motion of that pattern in the camera frame as the actuator moves the camera, since the pattern is static, computer vision algorithms can be used to determine the camera's six degree of freedom pose relative to the pattern, thus each image can result in six data points for solving joint equations [75]. With this approach, there exists enough knowledge about the camera coordinate system relative to the target pattern to create a matrix representation of the transformation between the two. A matrix representation of all the joints before the camera can also be created. The change in motion between two frames can be seen as the homogeneous transformation matrix at frame *i* multiplied by the inverse of the homogeneous transformation matrix at frame *i – 1*.

$$\Delta \boldsymbol{H} = \boldsymbol{H}_{i-1}^{-1} * \boldsymbol{H}_i \qquad \textit{Equation 15}$$

This delta **H** term represents the transformation from frame i-1 to i, thus if a calibration pattern is viewed at two points a transformation can be extracted at each point, and a delta can be calculated. The same goes for the servo's actuating the camera. The end effector can be at a known position, move to another known position, and the delta can be calculated. The use in this is calculating the unknown location of the camera relative to the end effector. Such a case would be a camera mounted onto a robotic arm, the end effector would be the motor directly above the camera in the kinematic chain; the camera location relative to the motor is another unknown transformation which may include a constant rotation and translation. This problem can be formulated as seen in *Equation 16* [76].

$$\Delta \boldsymbol{A} * \boldsymbol{X} = \boldsymbol{X}\, \Delta \boldsymbol{B} \qquad \textit{Equation 16}$$

Where delta **A** is the known transformation of the actuator, and delta **B** is the known transformation of the camera relative to the target pattern. It is now essential to calculate **X**, the unknown transformation between the two. This formulation was developed by Shiu and Ahmad [76] to remove any non-linear equations from the calculation, although this formulation does

double the number of variables. An improvement on Shiu's method has been proposed by Tsai to replace the rotation matrix component with an eigenvector of the rotation matrix [77]. This formulation maintains the linearity of Shiu's method without increasing the number of equations to solve.

2.10 – Point Cloud Simultaneous Localization And Mapping

Precise motorized motion over a large range is expensive; this is one of the reasons why systems from Cyberware cost hundreds of thousands of dollars. Precise motorized motion is also impossible when inspection robots need to use tracks or wheels to navigate. Thus it is necessary to locate the sensor relative to the global coordinate system with optical techniques.

In a purely vision based system, this can be done by placing checkerboard patterns in the environment to allow localization based on key reference points. Another technique is to not use checkerboard patterns, but instead to use feature point matching to determine the location of the camera at each frame [78]. The first of these techniques requires modification to the environment, while the second technique is prone to errors and false matches due to the reliance on good lighting conditions. The Kinect provides a dense 3D point cloud of an environment based on active sensing, thus it contains the information needed for feature based localization while using an active sensor. This improves robustness in environments where the lighting conditions cannot be controlled. This procedure is referred to as Simultaneous Localization And Mapping (SLAM). It is referred to as such due to the fact that the device is creating a map of the environment while it is localizing itself in that map.

Point cloud slam consists of creating a 3D reconstruction at two different locations, then registering those point cloud together. Registering point clouds is the process of finding the homogeneous transformation matrix between the coordinate systems of two point clouds. Common registration routines include: Iterative Closest Point (ICP), Sample Consensus Initial Alignment, and Sample Consensus Prerejective. These routines fall into one of two categories, iterative brute force methods and feature detection based. Iterative brute force methods include ICP and all derivations of ICP, these methods involve transforming the input point cloud and then calculating an error score between the input point cloud and the source point cloud. This can often be very expensive, but it also can create very accurate results. This iterative approach has many variations to reduce the computational expense and increase performance. The generalized ICP algorithm calculates a fitness score between two point clouds. This score is calculated by finding the distance between a point in the target point cloud and the estimated surface in the source point cloud [79]. An improvement of this is the non-linear ICP developed by Radu Rusu that uses a Levenberg-Marquardt optimization to increase the speed of convergence [80]. A simplification of the algorithm has been developed to allow for massive parallelization and fast implementation by Brian Curless. His implementation utilizes a Truncated Surface Distance Function volume to remove the need for corresponding point lookup and distance calculation [81]. This method has been implemented on graphics processing units to take advantage of the massive parallelization potential [82].

Feature based approaches attempt to identify corresponding features between two point clouds, and then calculate the transformation based on known corresponding points. This approach is

less computationally intensive than ICP but it also is less parallelizable and error prone in situations with noisy data. Sample Consensus Initial Alignment is an algorithm developed to find key points between two point clouds and then estimate an initial alignment to be later refined with ICP. This is done through estimating corresponding points through Fast Point Feature Histograms (FPFH) [83].

Sample Consensus Pre-rejective is an algorithm specific to RGB-D (Red, Green, Blue, Depth) data, where image features can be combined with 3D features to create a more robust point descriptor for correspondence [84]. These feature based algorithms are dependent on large feature vectors that describe unique points, which are time consuming to calculate, but also sensitive to noisy depth data. This sensitivity is due to many features depending on estimating a normal to an underlying surface, and that normal is dependent on neighboring points, thus small errors in point location can lead to large normal estimation errors, which propagates to creating a poor descriptor.

2.11 – Registering Profilometer Scans to Context Scans

Thus far we have discussed creating a point cloud from a swept plane structured light scanner, and creating a point cloud from a depth camera. Now it is necessary to discuss accurately and repeatedly combining point clouds from these two different sources. Point clouds from these two different sources will clearly have two different coordinate systems; the depth camera's origin will be at the sensor origin and the swept profilometer will be at the highest motor joint origin. Thus the transformation between these two origins must be found. A simple physical measurement can be done between the depth camera sensor origin and the swept plane sensor origin, but it is generally difficult to achieve an accurate transformation due to the sensor and motor enclosure.

There are two techniques to explore to create this transformation. One such method uses optics and a checkerboard pattern to determine the location of the camera and depth sensor relative to each other, and then calculating the motor origin relative to the depth sensor; much like what is used for stereo calibration. And the other is a point cloud registration technique, which involves creating a point cloud of an object from the two separate sources and using one of the previously discussed point cloud registration techniques to find the unknown transformation [85]. Each of these techniques has its own merits. Optical checkerboard based registration is a well-studied topic that can give a very accurate calculation of the position of the camera relative to the Kinect sensor, yet it highly depends on the camera actuator transformation. The end goal is to create a consistent point cloud between both scanners. This lends to the point cloud alignment technique which may account for inaccuracies not accounted for with the optical registration technique.

2.12 - Point Cloud Combination

Once the ideal transformation is obtained a combination of the point clouds must occur. This combination can be a simple addition of the two since each sensor has characteristic strengths and weaknesses that are apparent in the resultant point cloud. For example, the depth camera takes a large data set very quickly at a wide working range, but it has difficulty sensing depth and

complicated shapes. The points found around these features are noisy at best and completely inaccurate at worst. The swept plane structured light rig however takes relatively accurate data at these features; but it takes data slowly, thus large non-intricate objects take time to scan. Ideally, the depth camera could be used to measure the bulk of an object and the swept plane scanner can be used to measure detailed crevices and cracks.

Sensor fused depth data has been studied by Davis [86] and Kuhnert [87] in the form of combining stereo vision with a time of flight depth camera to create higher quality depth maps. These two methods are used for improving accuracy at run time and are somewhat incompatible with point cloud processing. In contrast Curless and Levoy use a point cloud post processing technique to merge data from a single plane laser scanner and a silhouette based reconstruction [88]. Yemez accomplishes this by placing the aligned point clouds into an octree along with point specific information such as surface normal and camera origin. By analyzing the surface created by silhouette scanning he was able to create a point cloud free of specular errors as is common with active methods and with less occlusion errors due to the nature of silhouette scanning. Yet this point cloud lacks the ability to find a convex surface due the nature of silhouette scanning. By analyzing the data from both sources, a complete model can be created with advantages from both systems.

2.13 – Three Dimensional Reconstruction

The topic of general 3D reconstruction should be discussed separately from SLAM due to the many different possible sensors, algorithms, and goals. The discussion that follows will primarily discuss environmental reconstruction using various optical depth measuring techniques. What is common to all of the following is the desire to create accurate dense 3D reconstructions. Accuracy and density in this case will be relative to the size of the environment being measured. The two categories that will be discussed are algorithmic improvements utilizing a single sensor and sensor fused approaches.

2.13.1 – Algorithmic Improvements towards 3D Reconstruction

This discussion will focus on the work being done on improving algorithms for performing 3D reconstructions with a single depth sensing device. This device may be an actuated LiDAR, depth camera, actuated profilometer, stereo vision, or any other single modality package.

The simplest single modality approach is the use of a single visual camera in what is known as monocular SLAM [78], monocular SLAM uses a single camera to create a reconstruction by finding key points in image pairs. This approach finds key points, with a descriptor such as SIFT or ORB, in multiple images. These key points can then be used to estimate the pose of the sensor in 3D space as well as the location of these points. This method is completely passive and depends on detecting unique points in the environment. A web based implementation of this has been created by [89] to be used by the masses with any camera.

An improvement over monocular SLAM is stereo SLAM as proposed by [90]. In this approach monocular SLAM is used to estimate the position of the stereo camera over the course of several frames, while stereo vision is used for gathering dense 3D data and range information. Like monocular SLAM, this approach is passive which has advantages in some environments, yet this presents issues in some unstructured inspection environments where texture is not guaranteed. Some closer related work can be found in [91] where a Microsoft Kinect is used with the initial

version of Kinect Fusion to perform 3D reconstruction with only depth data. This approach is fast but error prone due to potential tracking issues with insufficient computing power.
A computationally more efficient method is RGB-D SLAM [92] which takes advantage of visual and depth data from the Kinect or stereo vision. This approach uses monocular slam to estimate the pose of the camera over several frames, which is then used in conjunction with the depth data to create a dense 3D map. The map is then refined using graphical approaches to make a consistent dense 3D reconstruction.
For high quality reconstructions [93] uses a RGB-D SLAM to create an initial reconstruction via frame to model registration. Then in post processing the model is refined using a global optimization technique to create impressively accurate reconstructions of interest points. The goal being to utilize all data to create a globally consistent reconstruction while optimizing the use of dense regions to protect local geometry. This approach allows for fast initial mapping but requires significant post processing. The author lists an approximate one hour of post processing required for every minute of data capture.

2.13.2 – Sensor Fused Approaches for 3D Reconstruction

This section will discuss some relevant work in utilizing multiple sensors or sensing modalities to create accurate 3D reconstructions. These works are particularly related to this thesis since the goal of this thesis is to fuse data from a depth camera utilizing SLAM routine with data from an actuated profilometer.
One such potential combination of sensors is the use of a visual camera for monocular SLAM as described previously with a Time of Flight flash LiDAR as discussed in [94]. This approach is successful because it allows for the positioning of the LiDAR to be determined by the visual camera, while the LiDAR can measure textureless surfaces that the camera cannot measure. In general flash LiDAR does not return sufficiently dense 3D data to be used with a point cloud based SLAM algorithm but it does return quality measurements in large and outdoor environments. Because of this, this approach is well suited for environments where a structured light depth camera would not work.
Iocchi et all [95] propose the use of an Inertial Measurement Unit (IMU), stereo vision and a 2D LiDAR to create reconstructions. This has the advantage of dense, accurate stereo data at close ranges, and sparse but accurate LiDAR data at long ranges. The IMU combined with the LiDAR and stereo data allows for registration of point cloud data via IMU instead of algorithmic means.

3 - Theory

This section will discuss in detail the theory behind the specific implementation of the system discussed. It will discuss the mathematical models and statistics used in triangulation, camera calibration, structured light laser calibration, structured light laser correspondence, actuator calibration, Kinect 3D structured light, and point cloud SLAM.

3.1 – Swept Plane Structured light

A swept plane structured light system is a system that creates a 3D reconstruction of an environment via sweeping a laser plane through the environment and observing the deformation of the laser beam. To do so, one needs a device for performing triangulation and a device for actuation. The device used for triangulation is commonly referred to as a profilometer, and the actuator can provide linear motion in the form of a linear stage or translation of the part on a conveyor belt or rotational movement in the form of a servo. The result of this is a high accuracy 3D reconstruction from reconstructing the profile of the laser beam intersecting with the environment.

3.1.1 – Profilometry via Triangulation

Triangulation, in two dimensions, is the calculation of distance through similar triangles, as seen in *Figure 18* [36]. The laser creates the ray 'd' to the object which reflects off of the object and creates a return ray perceived by imager. The imager perceives that incoming light at location 'x' along the image sensor which then creates a similar triangle. This similar triangle simply states that the ratio of 'q' to 's' should be the same as the ratio of 'f' to 'x'. This can be expressed as seen in *Equation 17* [36].

$$\frac{q}{s} = \frac{f}{x} \qquad \textit{Equation 17}$$

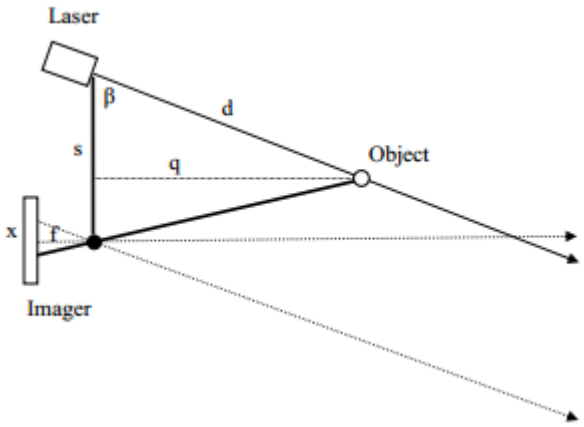


Figure 18: Basic 2D model of triangulation by structured light K. Konolige, J. Augenbraun, N. Donaldson, C. Fiebig, and P. Shah, "A low-cost laser distance sensor," 2008 IEEE Int. Conf. Robot. Autom., pp. 3002–3008, May 2008. Used under fair use 2014.

A convenient extension of this is to move to vector form, thus allowing for an easy extension into three dimensional space. The parametric vector form of a point on a line is seen in Equation 18 [69], which describes a point *P* given a base point *Q* and a vector *V* scaled by a constant lambda; thus given two rays the point P is ideally found by solving Equation 19 for λ_1 and λ_2 [69]. Where the first ray is created by the light source and the second ray is determined by the focal point and the intersection of the ray with the image plane [69].

$$P = Q + \lambda * V \qquad \textit{Equation 18}$$

$$Q_1 + \lambda_1 * V_1 = Q_2 + \lambda_2 * V_2 \qquad \textit{Equation 19}$$

Another form of this is used with line plane intersection to allow for the calculation of a point on a plane created by a line generator [69]. A line generator is an optical element which creates a plane of light, which when projected on a surface, creates a line [96] thus the name. This other form of structured light involves an implicit representation of the plane with the parametric form of the ray, as seen in *Figure 19.*

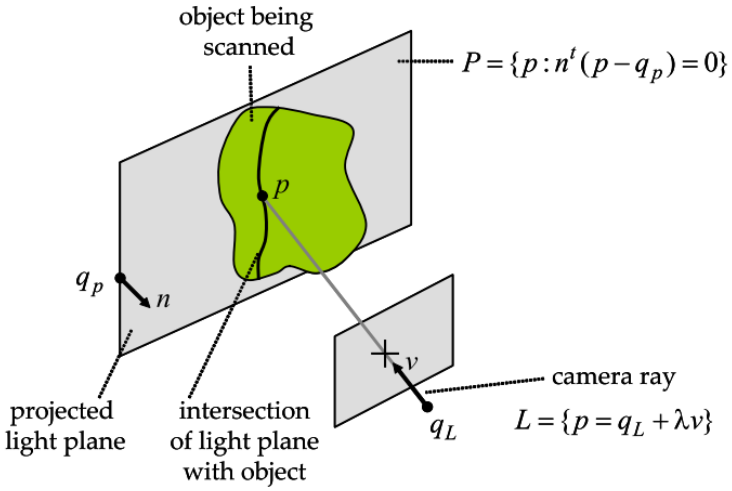


Figure 19: Single plane structured light scanning visualized *D. Lanman and G. Taubin, "Build Your Own 3D Scanner : 3D Photography for Beginners," Siggraph, 2009.* Used under fair use 2014.

The implicit representation of a plane can be seen in *Equation 20* where *N* is the vector normal to the plane (*n* in the figure), P is an intersection point of the plane and the surface being scanned, and Q_{Plane} is a point on the plane (q_p in the figure). Solving for a point on a ray and on the plane is done by inputting *Equation 18* into *Equation 20* to create *Equation 21*. *Equation 21* is valid because the intersection point is mutual between the ray and the light plane [69]. Reorganizing *Equation 21* gives *Equation 22* which calculates the scaling factor, this can then be used with *Equation 18* to find the point *P* [69].

$$N^T(P - Q_{Plane}) = 0 \qquad \textit{Equation 20}$$

$$N^T(Q_{Line} + \lambda_{Line} * V_{Line} - Q_{Plane}) = 0 \qquad \textit{Equation 21}$$

$$\lambda_{Line} = \frac{N^T(Q_{Plane} - Q_{Line})}{N^T * V_{Line}} \qquad \textit{Equation 22}$$

A downside of triangulation can be occlusion. Occlusion occurs when the camera's view of the light is obstructed. Likewise, the opposite is true when the camera can view a surface but the light is obstructed. The effect of this can be seen in *Figure 20*: Effect of occlusion on triangulation. *Figure 20* where the height of the pegs cause shadows on both sides due to occlusion. This can be particularly problematic when measuring the depth of negative features with high aspect ratios.

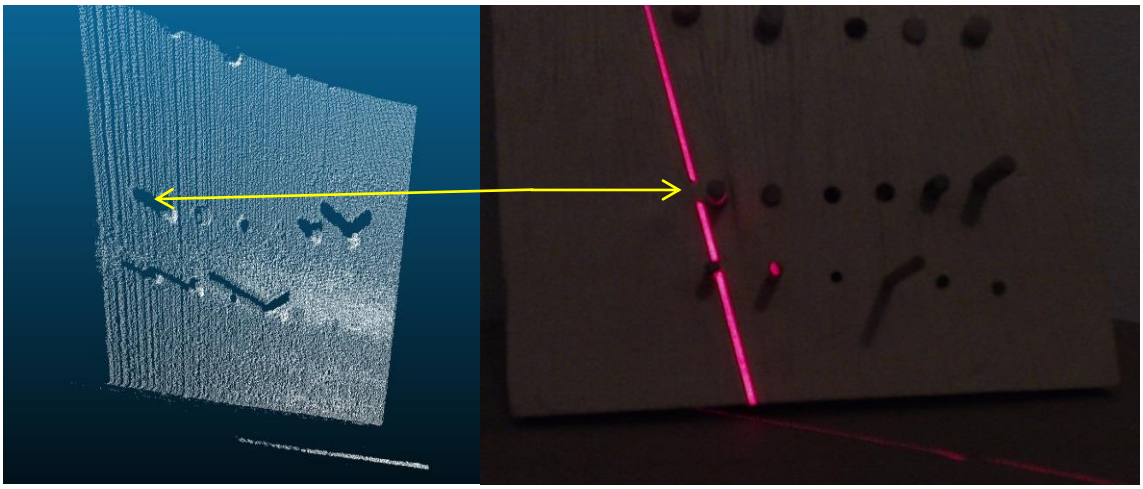


Figure 20: Effect of occlusion on triangulation.

3.1.2 – Structured Light Parameter Tuning

The parameters for creating a 2D structured light system can be tuned for the desired application.

The depth resolution of the system depends on the following factors: base line distance between camera and light source, camera focal length, pixel density, and sub-pixel interpolation. The camera base line distance can be optimized for a specific application. The camera focal length can be changed with different lensing. The sensor can be selected to achieve the necessary pixel density. Lastly, there are methods to improve sub-pixel interpolation for higher accuracy. The parameters of the system will be simplified for analysis by analyzing the middle row or column of the sensor that is perpendicular to the laser plane. This affectively projects the problem back into 2D for analysis. *Figure 18* shows this projected structured light configuration.

The most important parameters for a structured light depth scanner are depth resolution, minimum working range, and depth accuracy. Depth resolution describes the smallest feature that can be perceived, minimum working range describes the closest feature that can be perceived, and depth accuracy describes the perceived absolute location of a point in space.

Depth resolution can be seen as the change in 'q' for the smallest measurable change in 'x', where 'x' is the position on the image sensor, since both 'f' and 's' are constant within a set system with a non-variable zoom [36]. The smallest measurable change in 'x' is related to the pixel size and correspondence algorithm's sub pixel interpolation. For example, an algorithm that can find the return light's beam to within 0.1 pixels, with a typical pixel size of 0.004 mm, can determine 'x' with a resolution of 0.0004 mm. To analytically determine the expected resolution, one can take the derivative of *Equation 17* to find *Equation 23* which describes the resolution with respect to the absolute 'x' position [36]. Although it is known from *Equation 17* that the absolute 'x' position also describes the absolute distance between the object and the sensor. Thus we can input *Equation 17* into *Equation 23* to find the resolution with respect to distance from the sensor, as this is a more relatable term as seen in *Equation 24* [36].

$$\frac{dq}{dx} = -\frac{f * s}{x^2} \qquad \textit{Equation 23}$$

$$\frac{dq}{dx} = -\frac{q^2}{f * s} \qquad \textit{Equation 24}$$

Now that the terms for the depth resolution have been determined, it is convenient to plot these terms for a given sensor and expected sub pixel interpolation to see how resolution changes with distance 'q' from the sensor and baseline distance 's', as seen in *Figure 21*.

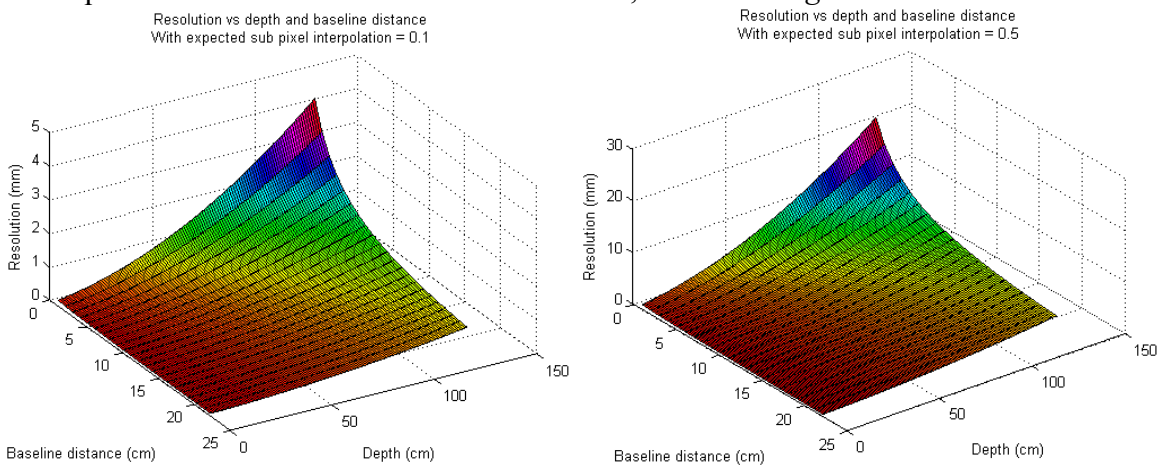


Figure 21: Resolution vs. distance and baseline distance.

As seen in *Figure 21*, there is a hyperbolic effect on resolution as the object's distance increases; this is intuitively due to the image sensor not being able to resolve small changes from far away. Increased baseline distance has a significant influence on improving depth resolution at further distances. Lastly, sub pixel interpolation has a linear effect on resolution (given a constant pixel size) and can be improved without changing hardware.

Opposing factors to increasing the baseline distance to infinity are the minimum and maximum working distances; which are factors controlled by the field of view of the camera, the laser position, and the angle that the laser is placed at. The minimum working distance is governed by *Equation 25* while the maximum working distance is governed by *Equation 26*.

$$\min = \frac{BaseLine}{\tan(Theta) + \tan(\frac{FOV}{2})} \qquad \textit{Equation 25}$$

$$\max = \frac{BaseLine}{\tan(Theta) - \tan(\frac{FOV}{2})} \qquad \textit{Equation 26}$$

A plot of *Equation 25* can be seen to illustrate the effect between baseline and Theta for a constant field of view (FOV).

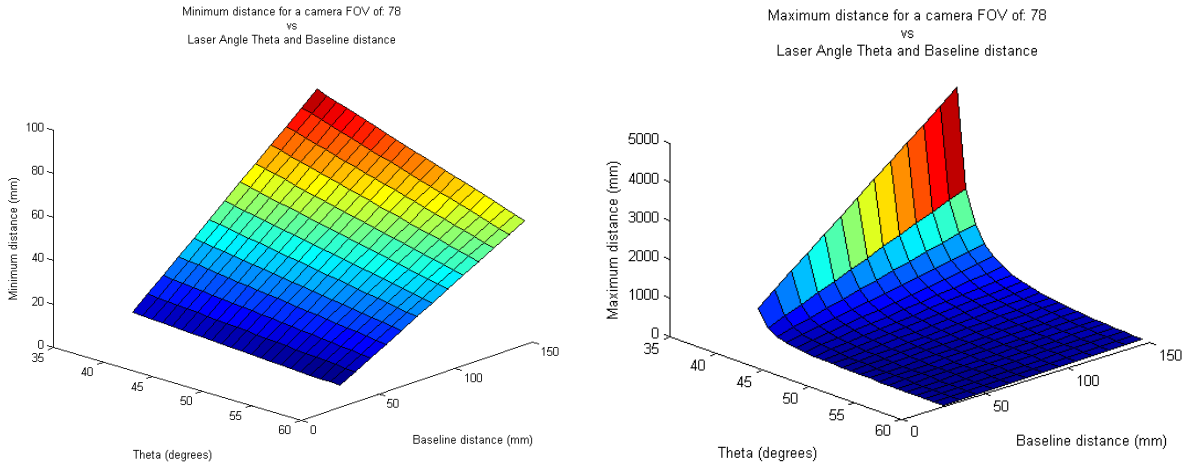


Figure 22: Working distance vs. laser angle for different baseline distances.

Figure 22 shows that the baseline distance has a strong effect on the minimum working distance whereas Theta has a strong effect on the maximum working distance. As can be seen, the maximum working distance approaches infinity when Theta is equal to half of the field of view, this is due to the laser beam being parallel to the FOV limit and thus the parallel rays will never intersect, which was predicted in *Equation 26*. It is undesirable to have such a large maximum range in a measurement grade sensor since the expected resolution at higher ranges is significantly worse than desired. Thus one could calculate the worst acceptable resolution and then calculate the angle Theta accordingly.

3.1.3 - Camera Calibration

The camera model that was discussed in section 2.7 contains several parameters describing the camera image sensor and optics. These parameters are generally given with the specifications of the camera, but these given parameters are hardly ever sufficient due to variation in the manufacturing process. A calibration process for finding more exact values of these parameters is necessary to perform accurate measurements.

Camera calibration consists of analyzing images of known structures to determine camera parameters. These calibration structures can consist of a 1D stick, a 2D plane, or a 3D structure. Two dimensional planes are the most widely used calibration patterns because they are cheaply and easily made and they create generally accepted results. A planar calibration object can be used to calibrate a camera by finding key correspondence points to reconstruct a known plane. The corresponding points can be used to estimate the location of the camera relative to the checkerboard plane, as well as the internal camera parameters. Calibration via a 2D plane is most common, and thus most optimized. The process is generally as follows [97]:

1. Detect a known pattern in a given image via image processing or manual selection. This gives the coordinates on the image plane associated with known geometric coordinates in real space.
2. Given the known points on the image plane and known real world geometric parameters,

solve for the 3D orientation of the object plane with respect to the camera coordinate system.

3. Solve the closed form camera parameter equations for ideal camera parameters, such as focal length, and principal point, assuming no projection error, no measurement error, and no lens distortion.
4. Perform a most likely estimator optimization routine on sets of image data to minimize the projection error of multiple known objects onto the camera plane
5. Model radial lens distortion to account for the rest of the projection errors
6. Perform linear least squares on the lens distortion parameters to minimized the projection error

3.1.4 – Camera Lens Distortion Correction

Radial and tangential lens distortion is corrected for by the method proposed by Zhang [97]. Lens distortion is modeled with second and fourth order radial terms as well as tangential terms. To calculate such a distortion, a suspect point is converted to a ray that is projected into space from the origin of the camera [56]. This is done by multiplying the point in homogeneous camera coordinates by the inverse camera matrix. The ray created by this projection is scaled such that Z is unity and the X and Y coordinates in 3D space are extracted. *Equation 27* through *Equation 30* adjust a (X, Y) coordinate in 3D space relative to the camera, as viewed with lens distortion, to a (X', Y') point in 3D space relative to the camera, as if viewed with no lens distortion [56]. The points can then be projected back onto the image sensor via the camera matrix [56].

$$R^2 = X^2 + Y^2 \qquad \textit{Equation 27}$$

$$C = 1 + k_1 * R^2 + k_2 * R^4 \qquad \textit{Equation 28}$$

$$X' = X * C + 2 * p_1 * X * Y + p_2(R^2 + 2 * X^2) \qquad \textit{Equation 29}$$

$$Y' = Y * C + p_1(R^2 + 2 * Y^2) + 2 * p_2 * X * Y \qquad \textit{Equation 30}$$

Equation 27 is the radial location of the point relative to the optical axis. *Equation 28* is the coefficient of radial lens distortion that determines the amount of warping caused from moving away from the optical center. *Equation 29* and *Equation 30* determine the lens distortion in the X and Y directions with respect to the combined radial and tangential components. The final image sensor coordinate can then be calculated via projecting the (X', Y') ray onto the image sensor via the following equations:

$$u' = f_x * X' + c_x \qquad \textit{Equation 31}$$

$$v' = f_y * Y' + c_y \qquad \textit{Equation 32}$$

Equation 32 and *Equation 31* are simplified projections of the adjusted points onto the image plane. The simplifications stem from the fact that Z is set to unity by the ray projection method

previously discussed. The (u', v') coordinates found can thus be used as if there was no lens distortion present.

3.1.5 - Structured Light Calibration

Once camera calibration is complete, the camera can be used as a precise angle measuring device due to its ability to measure the angle of an incident ray to within sub pixel accuracy. To perform measurements it is necessary to calibrate the emitter with respect to the camera. Ideally this is done through an optical method, as this would create a model for the emission source with respect to the camera by using the camera.

The method used in this thesis follows the method presented by [71] with some improvements. Given a known checkerboard pattern presented to the camera, the plane of that pattern can be determined. The checkerboard plane is represented by its normal vector, Nc, and a point on the plane, Qc; both of which are with respect to the camera [71]. These parameters can be extracted from the homogeneous transformation between the plane's coordinate system and the camera coordinate system (found via the same techniques used for camera calibration). The expected laser line is then extracted from each row of the image via the correspondence algorithm discussed later and then projected onto the checkerboard plane. Each row of the imager thus creates a single ray extending from the focal point of the camera towards the checkerboard plane. The intersection of that ray and the checkerboard plane is a point in 3D space that represents a point that lies on the laser plane and the checkerboard plane with respect to the camera coordinate system. Furthermore, a set of these points can be gathered from multiple images; and, that set can be used to determine the laser plane parameters.

Laser plane estimation from a set of data points can be done through two methods; least squares fitting of the data to a plane, and principal component analysis. Least squares fitting involves attempting to minimize the sum of the normal distance between all points and the estimated plane. Principal Component Analysis (PCA) attempts to determine the two directions of most variation in the dataset, which would thus be two vectors on the laser plane. The cross product of the two vectors from PCA can then be used to determine the plane normal while the average location of all points provides a good estimate of a point on the plane. These two parameters are all that are required to define the laser plane relative to the camera.

3.1.5 - Actuator Calibration

Once a single plane can be reconstructed, it is necessary to sweep that plane through the scene to gather more useful data. To do this, at least one actuator is required. To create high quality 3D reconstructions the motion of the camera must be calibrated to the motion of the actuator. The state of the art in actuator calibration was previously discussed in section 2.8. We will now continue with a discussion of the method developed by Tsai in [98]. Let us start by defining the naming convention of each coordinate system.

1. Camera: Sensor origin, defined as seen in *Figure 15*.
2. World: An arbitrary static location in the world. Often defined by the origin of a static

checkerboard pattern placed in the scene.

3. Base: Fixed coordinate frame of the actuator base. All joints, and thus camera, move relative to this coordinate frame.

Thus, the following kinematic chain describes a point P in the camera coordinate system relative to the base coordinate system.

$$P^{\text{Base}} = \boldsymbol{H}^{\text{Base}}_{Joint1}\boldsymbol{H}^{Joint1}_{Camera}P^{Camera} \quad \text{Equation 33}$$

Finding the transformation between the base and the first joint linkage is straightforward due to encoder feedback from servos. The encoder outputs an assumed reliable position value which can be entered into *Equation 5* through *Equation 7*, based on the axis of rotation, to determine the transformation. The topic of actuator calibration focuses on finding the transformation, between the camera and the first joint. This rigid body transformation describes the offset between the camera and the actuator, as well as any skew in the mounting configuration.

To find this unknown transformation we can state that since the Base coordinate system and the World coordinate system are static, that any transformation of the camera relative to the base coordinate system should induce a similar transformation of the camera relative to the world coordinate system [98]. Ideally this could be solved by also knowing the transformation between the Base and the World, which would make the solution as trivial as *Equation 34*.

$$\boldsymbol{H}^{Base}_{World} = \boldsymbol{H}^{Base}_{Joint1}\boldsymbol{H}^{Joint1}_{Camera}\boldsymbol{H}^{Camera}_{World} \quad \textit{Equation 34}$$

Unfortunately the transformation between the world and base are not known in this situation and thus this will not work. Instead we must analyze the displacement between two positions, as seen in *Equation 35* and *Equation 36*. These two equations describe the displacement of the camera and the displacement of the actuator between frame i and j.

$$\Delta\boldsymbol{H}^{\boldsymbol{Camera}} = \boldsymbol{H}^{Camera}_{World}(i) * (\boldsymbol{H}^{Camera}_{World}(j))^{-1} \quad \textit{Equation 35}$$

$$\Delta\boldsymbol{H}^{\boldsymbol{Actuator}} = (\boldsymbol{H}^{Base}_{Actuator}(i))^{-1} * \boldsymbol{H}^{Base}_{Actuator}(j) \quad \textit{Equation 36}$$

Since the camera is rigidly mounted to the actuator, it is reasonable to use *Equation 16* to model this motion. This model contains a total of sixteen unknowns due to the formulation of it as a 4x4 matrix, yet in reality the transformation between the camera and actuator has six degrees of freedom. Tsai's formulation significantly reduces the number of unknowns by properly formatting it to account for these constraints. Four of these unknowns are actually known as they are the bottom row of the transformation that is added for convenience to allow rotation and translation at the same time, as described in *Equation 11*. Thus Tsai splits the solution into a rotation and translation part [98].

The 3x3 rotation matrix describes a rotation about three different axis thus it could also be described as an angle of rotation about a vector around a principal axis. This principal axis is the eigenvector of the rotation matrix [98]. This compresses the nine value rotation matrix into a four value vector describing the rotation axis and the angle of rotation. The advantage of this is a linear mapping of values that reduces the number of unknowns, while maintaining the linearity of the system. Since a linear mapping was used, it is possible to use a linear least squares

technique to solve for the most likely rotation.

The linear transformation between the camera and actuator can then be found using a system of linear equations containing *Equation 37* for each pair of i-j poses.

$$(\boldsymbol{R}_{Actuator}^{Base} - \boldsymbol{I}) * \boldsymbol{T}_{Camera}^{Actuator} = \boldsymbol{R}_{Camera}^{Actuator} \Delta \boldsymbol{T}^{Camera} - \Delta \boldsymbol{T}^{Actuator} \qquad \textit{Equation 37}$$

In *Equation 37*, **R** is the rotation matrix component of the associated homogeneous transformation, the same goes for **T** as the translation component. Intuitively this states that displacement of the camera that is not accounted for by the rigid rotation (right side of the equation) should be accounted for by a rigid translation.

The result of this calibration is a homogeneous transformation matrix describing the position of the camera relative to the actuator coordinate system; thus this can be used to precisely locate the camera through the entire motion of the actuator. An added advantage of this is that it can account for errors in actuator alignment and camera mounting alignment that could be in-perceivable and immeasurable otherwise.

3.1.6 – Line Striped Structured Light Correspondence

This section will discuss the mathematical formulation and training process for creating an algorithm that can perform structured light correspondence in unstructured environments. The result of which is the ability to use profilometry in any environment without using optical filters to remove ambient lighting. Furthermore this section will discuss sub pixel estimation methods used to improve perceived sensor resolution.

Once an image is acquired, a pseudo mask is created from the visual image by subtracting the green plane from the red plane; thus highlighting the strongly red values while removing areas that have a combined strong response in the red plane, such as white areas.

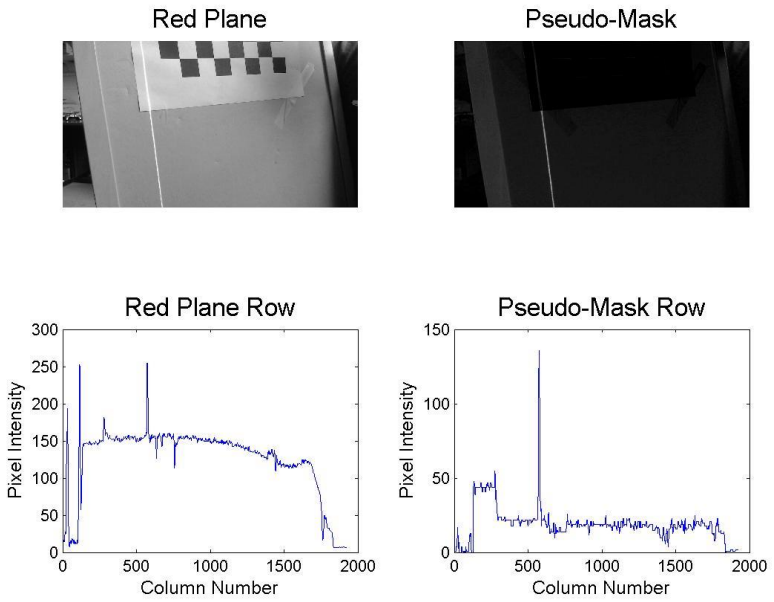


Figure 23: Pseudo masking to find red laser line.

As can be seen in the plots above, the laser line is barely distinguishable in just the red plane but it is extremely easy to extract from the pseudo red minus green plane. The horizontal gradient is then calculated based on the mask via a 3x3 Sobel filter convolved over the pseudo mask image.

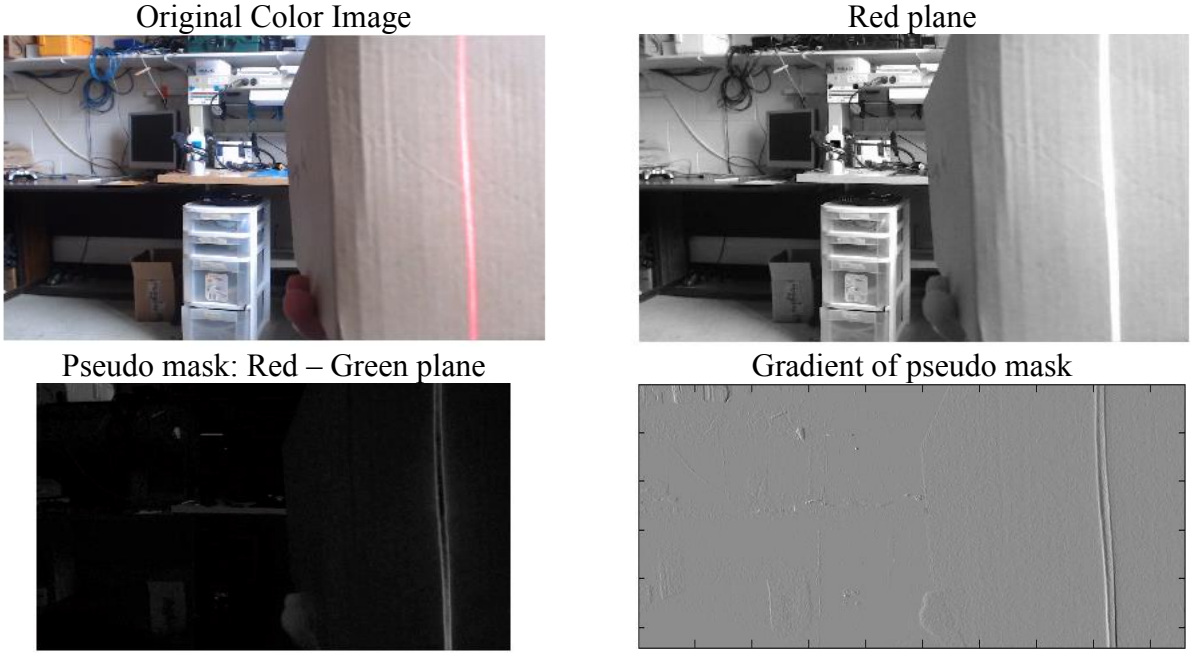


Figure 24: Select images from processing pipeline for laser correspondence.

The pseudo mask is then used to estimate local maxima. For each row, the algorithm sweeps the

column from left to right and looks for a high point in the pseudo mask image. A high point is determined by a maximum value followed by a set number of values significantly lower than that maximum. Since the maximum value estimation is performed on the pseudo mask, it can have issues with over-exposed beam centers.

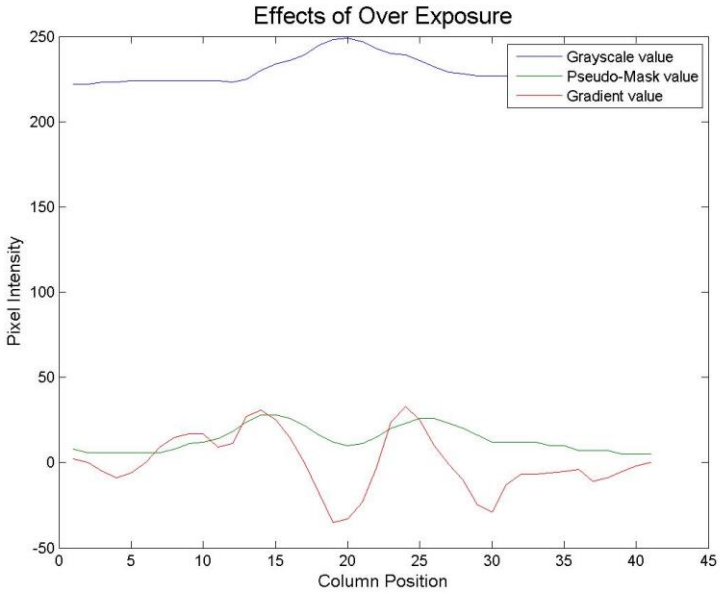


Figure 25: Effect of saturated pixels due to over exposure on line detection.

As can be seen above, the mask image has a dip in the middle, due to the over exposed pixels appearing as white instead of red. Thus, to find the true beam center, the algorithm uses the estimated local max as a starting point for a centroid search algorithm, based on the red plane. This centroid estimation improves the estimate of the center, and is useful for machine learning algorithms. Machine learning algorithms are extremely useful tools for adding intuition to computer programs. In this case, machine learning algorithms are used to teach the computer how to differentiate between the laser beam line and the many false positives that could be present. To do this, several features are selected that describe useful information about local pixel values in a compact form. The features selected for testing include the red plane intensity at center, pseudo mask intensity at center, left gradient sum, right gradient sum, sum of left gradient sum and right gradient sum squared, estimated beam width, X position, red plane intensity * X position, red plane intensity / width, and width * X position.

The red plane value at center is an obvious choice since it correlates to the beam strength. The pseudo mask value at center is useful, because if it is close to the red plane value, then the laser beam is not over exposed; whereas if it is extremely low, then the beam is over exposed. This helps the algorithm know what to expect in other features. The sum of intensities to the left and right, inside of the search window, are similarly intuitive, since they should indicate a low to high transition and then a high to low transition, assuming a constant background color. The sum of the summed gradients should intuitively be close to zero, since the low to high and then high to low transition should be approximately equal; and the square of the sum negates any small difference in sign. The X position in the image is the distance from the origin to the estimated beam center, thus this should correlate to the position of the beam in space. Beam width should

be strongly correlated to beam intensity and X position, since the X position on the image plane determines the distance to an object. The combination of these features should create a correlation based on known interactions.

These local features were selected, because they are local features and they are fairly consistent across many different angles and curvatures. More complicated features could be selected such as Laplacian of Gaussian, which is a popular line detecting feature but Laplacian of Gaussian fails to give a consistent response over many laser beam orientations and widths.

The machine learning algorithms considered for this application include Naïve Bayes, Logistic regression, K-nearest Neighbors, and Support Vector Machines. Each algorithm has been described in the literature review section. Support Vector Machines were chosen due to their discriminant nature, high performance, and well optimized training routines.

Support Vector Machines (SVMs) are trained by optimizing the decision boundary between two classes of data. Data is segregated as either a positive or negative example with associated feature values. These feature values are then used to determine a hyper plane that optimizes the margin between positive and negative examples. The plane can be formulated as seen in *Equation 38* where w is a weight vector, x is a feature vector for a query point and b is a real value, which is equivalently the distance between a query point x and the hyper plane. In this linear formulation the <w,x> term represents the inner product of the w and x vectors. Thus for positive classes this distance is positive and for negative classes this distance is negative. Solving for this plane is done through *Equation 39* by solving for a w and b such that all points in the data set satisfy *Equation 39*.

$$< w, x > +b = 0 \qquad \textit{Equation 38}$$

$$y_i(< w, x_i > +b) \geq 0 \qquad \textit{Equation 39}$$

The above formulation is for a simple linear hard margin SVM. That means that the feature space is directly used for creating the hyper plane and outliers are not elegantly handled. Soft margin SVMs utilize a slack variable to allow outlier points to violate the margin, this significantly increases solution complexity since it is no longer linearly separable data. Solution of these SVMs involves quadratic programming because now the slack needs to be minimized simultaneously with the hyper plane parameters. Non-linear SVMs can further improve the complexity of functions that can be fit by moving the SVM decision space into higher features spaces. This is done by modifying the inner product found in the above equations to some other type of product that may internally include movement of a feature vector to a higher space to calculate this product. This can be done since the only thing that matters is that the product produces a distance metric between the two vectors, the formulation of this distance metric need not matter, and thus the distance doesn't need to be Euclidean distance in the feature space. Instead this distance can be a Euclidean distance in infinite dimensional space defined by any number of parameters to create the transformation between feature space and decision space.

3.1.7 – Qualitative Correspondence Results

The previous section described training a support vector machine to find the laser line in unstructured images. This section will show some example cases of this algorithm working correctly and some failure cases. Some concerns with using the features previously described

are when the surface being viewed is red, and when the camera views direct lighting. The first set of images will be an example set of images of a red surface at different exposures; the exposure will be manually controlled in this case. On the left is the raw image collected, and on the right will be a mask of where the laser was detected. Red will indicate positive laser beam classification and green will indicate a rejected false positive.

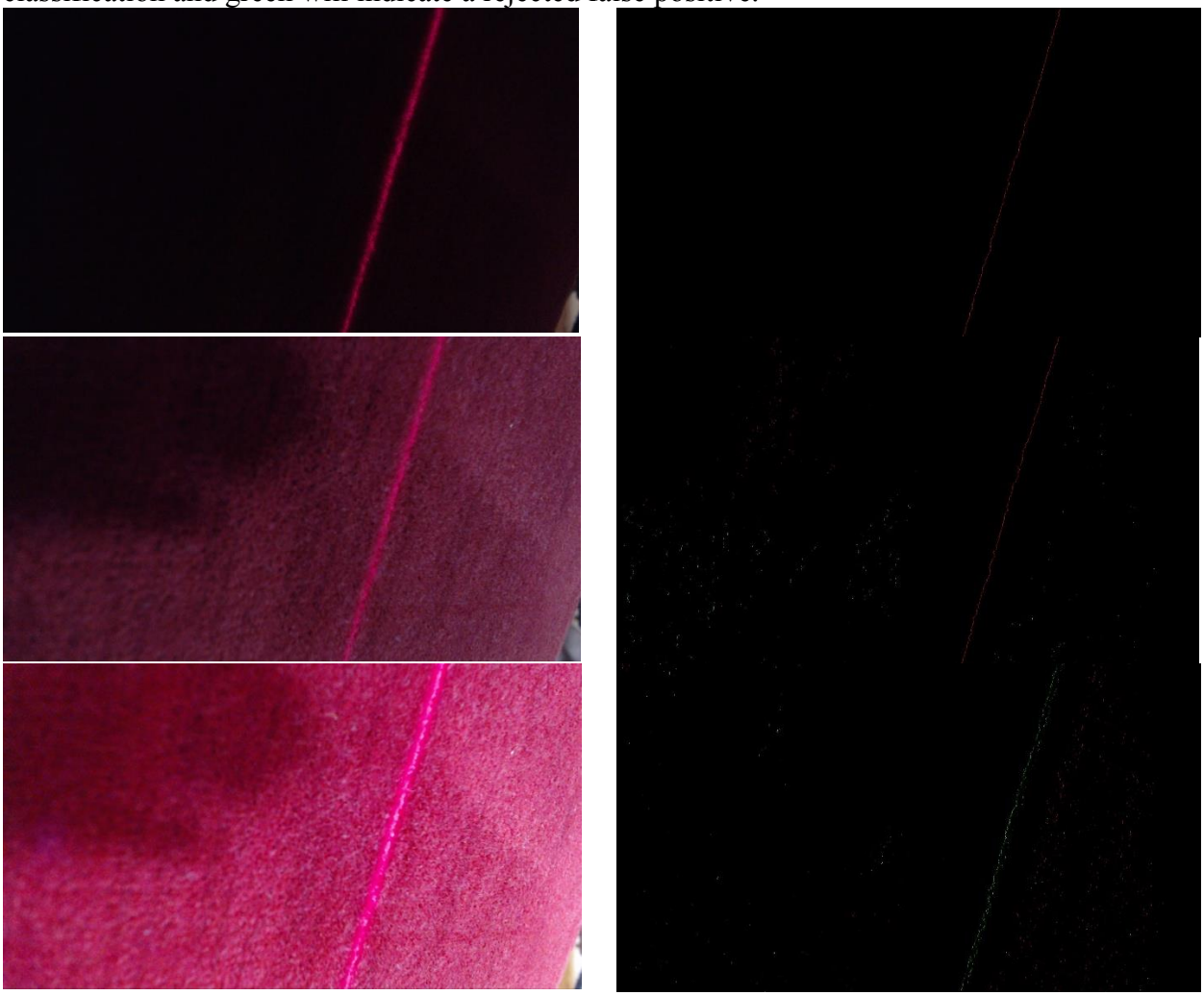


Figure 26: Profilometer correspondence on red surface.

As can be seen above, as the camera exposure increases so does the false positive rate. That is because features on the surface with ambient lighting being to look more and more like the laser line. This could be fixed with two approaches, filtering the data to look for continuity in the extracted line, assuming a continuous surface or automatically adjusting the exposure. The exposure control loop of this system would have maintain the exposure seen in the first image, so this would not have been an issue. If the system was naively started at the exposure found in the second set of images, then it would have brought the exposure down to what was found in the first set of images. The true failure case would be if the system started at the exposure found in the last set of images in *Figure 26*, since the algorithm could not find the laser beam, it would not adjust exposure properly.

Another concern is proper correspondence when a light source is in direct view because such an

over powering light source could cause blooming and bleeding in the resulting image. Once more, this was qualitatively studied by testing various exposures with a light source placed in view of the camera.

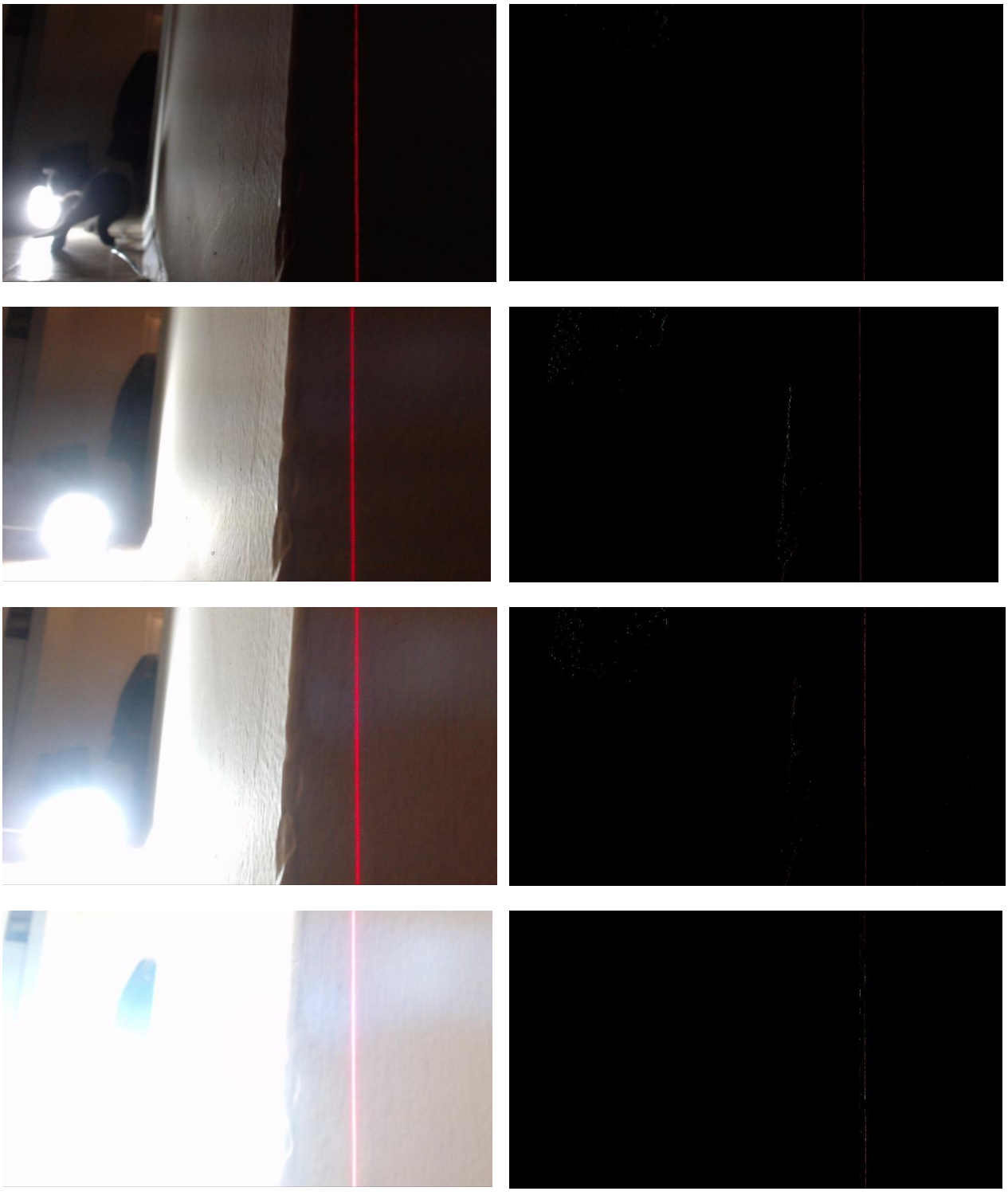


Figure 27: Profilometer correspondence while viewing light source.

As can be seen in *Figure 27*, the correspondence algorithm does adequately well at detecting the laser line at proper exposure levels. At increased exposure levels, the laser beam is identified sufficiently that the exposure control algorithm could correctly adjust exposure. There is a small rate of false positives at the lowest exposure which could be cleaned up with some continuity filtering. As well, there are point cloud post processing filtering steps that could be applied for such small amounts of false positives. At the highest exposure, the light source is over powering and the laser beam itself is over exposed. Ideally this case would not happen due to the exposure control algorithm, but in a naïve situation, the algorithm does sufficiently well in selecting the laser beam to properly adjust the exposure to what is needed.

3.2 – Six Degree of Freedom SLAM

In order to create a contextual reconstruction a Simultaneous Localization And Mapping algorithm needs to be performed on data with sufficient information to determine pose in all six possible degrees of freedom of motion. In order to do this, point data needs to be calculated in a full volume of space instead of on a plane in space, as in actuated profilometry. The chosen way of doing this is by using a depth camera which can collect three dimensional position data in a volume of space. Depth cameras return a depth map which allow for the perception of an environment with respect to distances instead of the color intensities that a normal camera sees. This section will discuss some of the theory and uses of depth cameras to create contextual reconstructions.

3.2.1 – Real-Time 3D Depth Camera Correspondence

In order to create a triangulation based structured light depth camera, one must replace the 2D plane used in swept plan scanning with a 3D light pattern as discussed in section 2.5. The entire pattern can be viewed at each frame, thus correspondence needs to be performed between the projected pattern and the observed image. It is believed that Prime Sense solves this problem by using an algorithm known as Region-Growing Random Dot Matching, with a depth from defocus lookup for speed improvements; although all of the following is speculation based on reverse engineering from patent searches [62]. The algorithm is as follows:

1. Detect dots and label them as unknown.
2. Select an unknown dot randomly and label it as a region anchor.
3. Determine the orientation of the ellipse created by the dot being skewed by the astigmatic lens.
4. Estimate the starting point for the search window based on ellipse orientation.
5. If the search window finds a match, change label to known. Otherwise set as invalid and go back to step 2.
6. Add neighboring points to processing queue.
7. Use known location as anchor for neighboring points then go to step 3.
8. Repeat process until all points are known or invalid.

There are many optimizations used to increase the performance of this algorithm over typical region searching algorithms. First, the camera uses an astigmatic lens that causes skewing of the dots based on distance; this alone would be sufficient for determining some level of depth, and thus is used to limit the correspondence search window. The second major speedup over typical window correspondence search is the use of anchor points and neighboring points to constrict the

search window, thus preventing excessive calculations for each point. The last speed up is Prime Sense's chip level implementation, meaning custom manufactured micro-processors that are designed for this specific task. These processors are finely tuned and very fast. The result of this is a system that can create a 640x480 pixel depth image from a projected pattern at 30 frames per second with cheap components.

3.2.2 – Real-Time Simultaneous Localization And Mapping

In order to increase the effective sensor volume, a depth camera is used in parallel with the previously described swept profilometer to allow localization of the system throughout the process. Since this system is supposed to work in an unstructured environment simple localization based on known key-points or a previously defined map is insufficient; therefore, a Simultaneous Localization And Mapping (SLAM) routine needs to be used. It would be possible to cache all of the data for a run until the end and then run a SLAM routine in post processing to create the reconstruction; but this would prevent any real time feedback for the user, which could be very useful to see where data is insufficient. To do that, a real time SLAM algorithm would be necessary.

One real time SLAM algorithm is the Kinect Fusion Graphics Processing Unit (GPU) accelerated implementation of ICP [91][82]. This algorithm takes advantage of a GPU's highly parallelized processing powers to accelerate ICP to a real time state. A sufficiently powerful GPU can run this implementation of ICP at the frame rate of the sensor. Thus, so long as the sensor isn't moved significantly between frames, the ICP algorithm can build a map and locate itself within its environment. The specifics of the algorithm are outlined as follows for each frame [63][73]:

1. A frame is captured from the depth sensor and sent to the GPU as a raw depth map.
2. A bi-linear filter is applied to smooth noisy depth measurements.
3. The depth map is converted to a 3D point cloud on the GPU.
4. A meshed point cloud is created from the point cloud.
5. The meshed point cloud is compared to the global Truncated Surface Distance Function (TSDF) volume and, a correspondence is calculated. (If this is the first point cloud, it is simply used to create the initial TSDF volume.)
6. Transform the point cloud and calculate a new correspondence.
7. Iteratively execute step six for a set number of iterations.
8. Pick the highest correspondence of all iterations.
9. Add selected point cloud to the TSDF volume.
10. Adjust the pose of the sensor based on the transformation used by the selected point cloud.

Graphics Processing Unit (GPU) computing is incredibly fast but can often be memory bandwidth limited when transferring data between the CPU and GPU. This is due to the limited bandwidth between the CPU and GPU relative to the massive processing power of the GPU and the large memory bandwidth on GPU itself. In order to help alleviate this issue the depth camera's point cloud is sent to the GPU in the most raw and compressed form possible; which is as a depth map. A depth map looks much like an image; except that each pixel corresponds to the distance of a point in the scene to the sensor. Thus instead of a point requiring twelve bytes

of data (one floating point value for each dimension), it only requires two bytes of data (This is an 11 bit depth map packed into 16 bits). The projection of the depth map into a point cloud is handled on the GPU based constant camera parameters, thus allowing for the parameters to be loaded at startup.

Point cloud meshing is the process of linking discrete points together to create a solid surface based on knowing which points are close to each other and on the same surface. This is done through connecting adjacent points from the depth image in an eight connectedness fashion and a distance threshold. The meshed surface is then converted into a Truncated Surface Distance Function Volume.

The Truncated Surface Distance Function (TSDF) volume is a convenient and incredibly fast method of encoding surface distance information [81]. Typically ICP requires searching for closest points between two point clouds and then calculating the distance between the two points, the sum of all distances can then be used as an alignment error metric between the two point clouds. A TSDF volume greatly increases the speed of this process by caching the distance between a surface and nearby voxels, thus instead of searching for closest points and calculating a distance, correspondence can be calculated by summing the voxels of the new input volume.

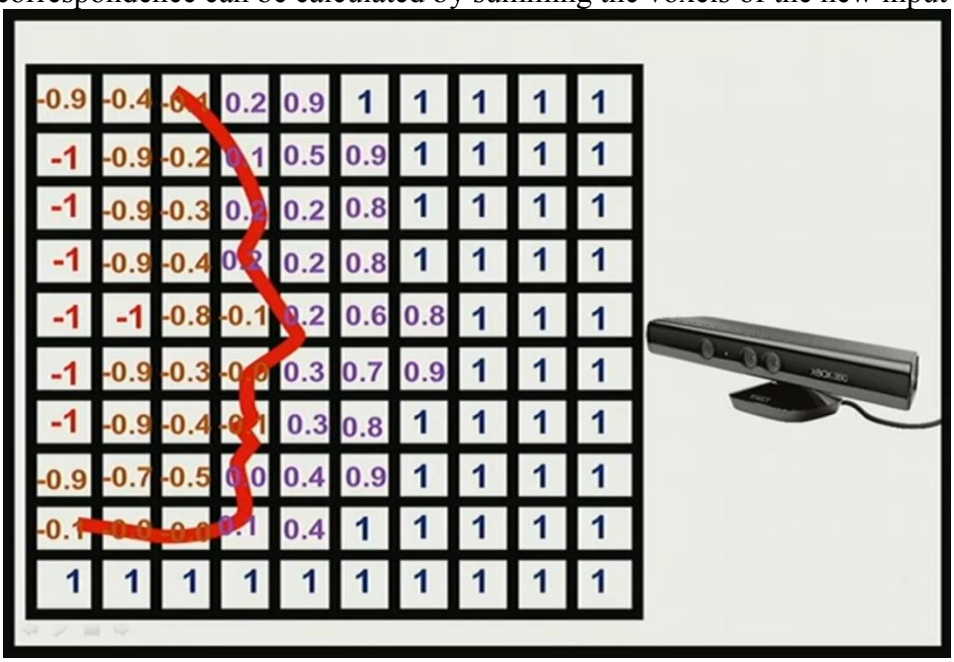


Figure 28: Cross section of a TSDF volume. *"Using Kinfu Large Scale to generate a textured mesh," PCL docu, 2013. [Online]. Available: http://pointclouds.org/documentation/tutorials/using_kinfu_large_scale.php. [Accessed: 06-Feb-2014].*

As can be seen the surface perceived is the meshed point cloud. The values in each voxel indicate the distance between that voxel and the surface. For small perturbations of the sensor, the input cloud will shift by a few voxels. The distance between the input cloud surface and global surface can be calculated by summing all of the values in the TSDF volume corresponding to the input cloud. The input cloud can be transformed by a small amount and, then a new

distance can be calculated. This is a significantly faster method of calculating distance that can be greatly sped up via parallel processing.

3.3 – Sensor Fusion

In order to make sense of the data from these two sensors, the data must be fused in a meaningful and understandable way. This involves understanding the strengths and weaknesses of each sensor, and creating a fusing function based on the output data and the sensor characteristics.

3.3.1 – Sensor Characteristics – Actuated Profilometer

Swept plane structured light scanners are characterized by the components used to create them; but, in general, they can be characterized as follows:

1. High accuracy
2. High resolution
3. Able to perceive intricate features
4. Relatively slow data acquisition
5. Unable to self-localize in unstructured environments

The first three characteristics of a swept plane scanner come from the geometric model used in triangulation and the correspondence search method. The geometric model used in triangulation refers to the usage of a plane ray intersection to calculate 3D position. Because of this the plane is a continuous object that allows for nearly continuous surface measurement thus, it gracefully handles discontinuities. This comes at a huge measurement speed disadvantage and the loss of real time scene awareness, since only a slice of the scene can be measured at a single frame.

3.3.2 – Sensor Characteristics – Prime Sense Depth Camera

The specific depth camera chosen is a Prime Sense depth camera that uses a 3D structured light pattern to create a 3D point cloud of the entire scene at each frame. This improves the downfalls of the 2D pattern used in swept plane scanners at the cost of the strengths of the actuated profilometer. The improvements come from the increased pattern complexity, since a 3D pattern is being projected into the scene, correspondence can be calculated for discrete sets of points in the pattern. It is this exact scheme that is also the downfall of this sensor. Since a 3D pattern is used, triangulation must occur through ray to ray intersection. Unfortunately, in 3D space, it is unlikely for two rays to perfect match with each other, thus a most likely point estimate is used to determine the intersection of the rays [69]. This estimate is the most likely actual point in 3D space where the ray from the projector and the ray from the camera intersect, and it is based on the closest intersection points of the two. Inherently, this is more susceptible to pattern laser calibration errors, as well as camera calibration errors. As well, correspondence becomes a pivotal factor in resolution. The use of a 3D pattern instead of a 2D plane makes correspondence significantly more difficult. Previously correspondence consisted of finding the laser line, now correspondence requires matching an observed ray to a projected ray. To simply this, 3D correspondence often involves searching for matching windows. Thus the likelihood of a group of rays can be determined instead of a single ray. This however creates issues with intricate features when only a few rays can be observed.

3.3.3 – Sensor Registration

The first step in fusing sensor data is to align that data to a common reference frame. This allows the alignment and comparison of features. Sensor registration was briefly discussed in section 2.10, where two methods were proposed, an image measurement centric methodology and a point cloud centric methodology. Each method has its own merits; an image based measurement between the two sensors has the advantages of known statistical optimization techniques; whereas, a data centric method has the advantage of accounting for un-previously accounted for errors in the measuring process.

Measurement centric camera calibration is consistent with the techniques used for stereo camera calibration. This process is well studied by Bouguet in [100]. The process consists of creating sets of image data from both cameras viewing the same checkerboard pattern. The pattern can be determined in 3D; space and thus, each camera's pose can be calculated. With the poses relative to the checkerboard known, it is easy to calculate the pose relative to each other through homogeneous transformation matrices. Consequently, an optimization routine can be run to further improve the accuracy through multiple poses.

Data centric methods for registration involve analyzing similar point clouds from the two sensors and determining the registration between them. To do this, the sensors need to be able to measure the same object to within reasonably similar resolutions.

To perform this registration, two point clouds can be collected from the same scene, the point clouds can then be manually aligned and ICP would be used to refine the manual alignment. The resulting transformation between the two point clouds is the transformation between the two sensors. The two sensors used for this project are on the boarder for compatibility with this method and thus this method will be attempted and results of such a registration will be discussed in the results section.

3.3.4 – Data Fusion

The task for this thesis is to provide contextual information for the swept profilometer scans such that a defect can be localized within a 10 cm area. Thus the most important feature of sensor fusion is that all local scans are in the same global coordinate system as the contextual scans. This can be done by accurate coordinate system calibration. Display of this data can then be performed with any point cloud display tool, the preferred tool for this thesis is Cloud Compare.

4 – Hardware Platform and Software

This section discusses the specific platform created for this thesis, as seen in *Figure 29*, as well as the software created to perform reconstructions.

Figure 29: Simulation robot sensor rig

The sensor rig shown above has the minimum necessary components for a mobile robotic inspection sensor rig. This sensor rig contains one depth camera and one profilometer, mounted in a rigid orientation. The profilometer contains one actuator, one laser and one camera. Such a system could be used as the end effector of a mobile robot platform with no need for feedback from any higher up actuators. The actuator used in this system could also be replaced with an actuator external to the sensor rig and higher in the actuator chain.

4.1 – Sensor Array

The depth sensor for this platform is a Microsoft Kinect for Xbox 360. This device returns an 11 bit depth map of the scene at a resolution of 640 x 480 at 30 frames per second. This device is rigidly attached to a baseplate that attaches to the actuated profilometer.

The actuated profilometer contains one digital servo motor which rotates the camera and laser assembly. The profilometer assembly contains one Logitech C920 HD webcam and two 5 mW, 650 nm LED laser diodes. One laser is a set focus line generating laser while the other is an adjustable focus line generating laser. The set focus laser achieves optimal focus at medium to long range, while the adjustable focus laser is optimally focused for very close range.

The laser width can be plotted against distance to create an estimate for the optimal working range for each laser, which is useful for determining when to switch between short range to medium range. The adjustable focus laser width can be seen in *Figure 30*, which indicates that the near focus laser should be used between 100 and 300 mm, while the medium focus laser should be used after 300 mm.

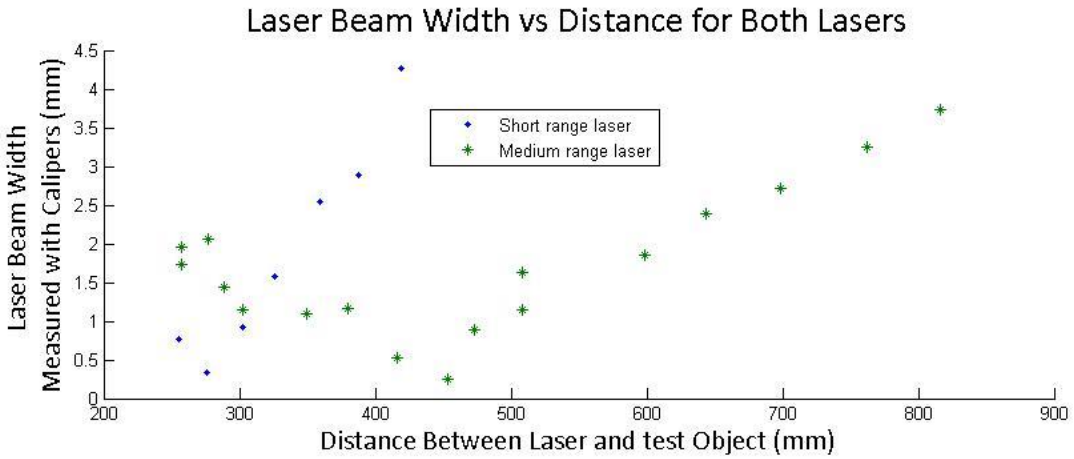


Figure 30: Laser width vs. distance for short and medium range lasers.

Two configurations of the structured light assembly will be tested, one using an HS-322HD digital servo, and one using a Dynamixel MX-28R digital servo. The HS-322HD digital servo is an internally closed loop digital servo, which is controlled by a PWM input signal. Since the servo has no feedback to the control software, the profilometer software must effectively work in open loop. This means that a command is sent to the servo to go to a position; then the command software waits an appropriate amount of time and then acquires an image, assuming that the servo is at the correct position. The Dynamixel MX-28R digital servo has an internal close loop PID controller with two way communication to the host controller. This means that for any given image, the digital servo can report an absolute position value. The comparison between the two servos can be seen in Table 2.

Feature	Dynamixel MX-28R	HS-322HD
Cost	$225.90	$9.99
Communication	RS-485	PWM
Positional Resolution	0.088 degrees	0.1 degree input resolution
Internal Feedback	PID via absolute magnetic encoder	PID via potentiometer
External Feedback	Absolute Position reported	Nothing reported
Mounting Hardware	Metal Internal Bearings	Resin Bushing

Table 2: Digital servo motor comparison.

As can be seen, the two motors vary significantly in cost and complexity. The HS-322HD uses an internal potentiometer and feedback loop to seek a value based on the input PWM signal, with an advertised resolution of 0.1 degrees. The Dynamixel instead uses a high quality magnetic absolute encoder for digital control and host controller feedback. Something that cannot be derived from datasheets is the build quality of each device. The Dynamixel servo has no apparent backlash, whereas the HS-322HD has a noticeable amount of backlash. As well, the rigidity of the of the Dynamixel's shaft interface is far superior.

4.2 – Lighting Control Scheme

Since this system is designed to work in unstructured spaces, the focus of lighting control is to maintain proper exposure. Maintaining proper exposure of the laser line is critical for proper correspondence since over exposure or under exposure can jeopardize the line finding algorithm. In order to account for scene differences due to distance and spectral response, a feedback control loop has been implemented to attempt to seek the best exposure in real time. This control loop looks at the average red plane intensity value for the peak of the laser beam. It then attempts to keep the average between 110 and 170, for an 8 bit analog to digital converter (ADC). These values were experimentally determined to be optimal in section 6.3.1.

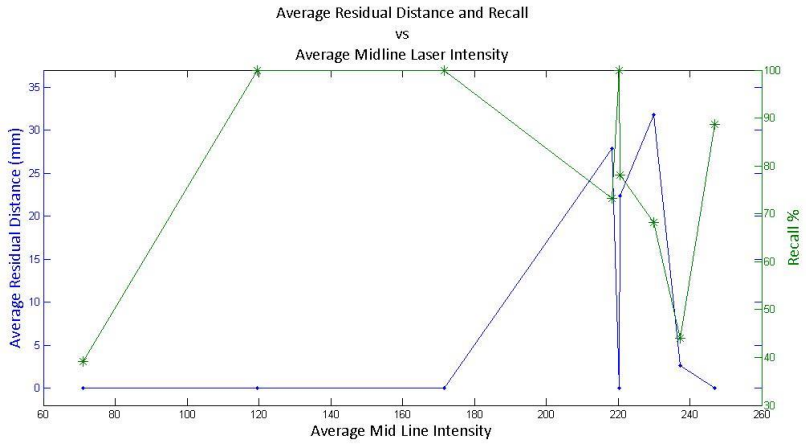


Figure 31: Exposure control validation plots.

The procedure for generating this data follows that of section 6.3.1, a single profile of a flat object is reconstructed, a line is estimated from the profile, and the residuals from the line estimation are used to estimate error. The standard deviation of these residuals is indicative of how noisy the measurement is. Further details on this process are found in section 6.3.1, and these plots are simply added here to provide evidence for the need and improvement found with proper exposure control.

As seen above, the optimal exposure creates an average laser beam intensity value between 120 and 170 to minimize residual error and maximize recall. Specifics about this reasoning is in section 6.2.

4.3 – Structured Light Correspondence Training and Verification

The features and the algorithm used were both previously discussed in the Theory section, thus this section will discuss the parameters used to train the selected machine learning algorithm and the results of validation. For each set of features, 51,000 training points were used from 34 hand labeled images. The data is randomly split into 80% for training and 20% for testing. The features for further reference are:

1. Red plane intensity at center.
2. Pseudo mask intensity at center
3. Sum of left gradient sum and right gradient sum squared
4. Left gradient sum
5. Right gradient sum
6. Estimated beam width
7. X position
8. Red plane intensity * X position
9. Red plane intensity / width
10. Width * X position

Support Vector Machines were trained using a Radial Bias Function kernel with a sigma value of one. A sequential Minimal Optimization algorithm was used with a termination convergence criteria of 0.001 and max iterations of one million. On average, the optimization routine converged after eight thousand iterations. Each feature was tested and the associated test errors are shown in *Table 3*. *Table 3* indicates which features are used, the percentage of correctly labeled laser beam points, the percentage of incorrectly labeled laser beam points, the percentage of false positives selected by the algorithm, the percentage of false negatives selected by the algorithm, and lastly the average improvement compared to the previous set of features used. The average improvement is the average difference between the correctly labeled laser line and correctly rejected negative points between the current set of features and the previous set of features.

Features used	Correctly labeled laser line	Correctly Rejected points	False positive	False negative	Avg % Improvement
Naive selection with no SVM	100	0	77.17	0	
1	72.35	65.96	34.05	27.65	46.81
1,2	83.89	89.54	10.46	16.11	17.29
1,2,3	86.99	87.12	12.88	12.01	0.34
1,2,3,4	91.40	95.83	4.17	8.60	6.56
1,2,3,4,5	92.51	96.16	3.84	7.49	0.72
1,2,3,4,5,6	94.37	96.73	3.27	5.63	1.215
1,2,3,4,5,6,7	98.44	98.42	1.58	1.56	2.72
1,2,3,4,5,6,7,8	98.51	98.53	1.47	1.49	0.09
1,2,3,4,5,6,7,9	98.22	95.52	1.43	1.49	-1.65
1,2,3,4,5,6,7,10	98.25	95.50	1.45	1.46	0.002
1,2,3,4,6	93.92	97.48	2.52	6.08	
1,2,3,4,6,7	98.15	97.65	2.35	1.85	

Table 3: Classification accuracy of an SVM with different features.

As can be seen in *Table 3*, most features improve classification accuracy. The most useful features appear to be feature one, feature two, feature four, feature six, and feature seven. This is somewhat intuitive since feature one gives an idea to the intensity of the beam, feature two indicates whether or not the laser beam is over exposed. Feature four gives some indication as to the magnitude of the gradient transition. Feature six adds an estimate of the width of the beam which should be strongly related to the intensity of the beam and depth of the scene. Feature seven adds location information which should tie into feature one, two, and six; since the location of the beam in the camera image indicates the distance between the beam and the camera which would indicate the expected intensity of return. As can be seen, feature eight and ten add an insignificant amount of useful information and feature nine actually adds confusion to the algorithm.

The strongest features are then used in another set of training where the SVM parameters are varied. The goal of this test is to determine the best sigma based on test set performance. The test set was again chosen to be 20% of the total dataset not used for training. Thus all data points will be completely unique to the algorithm. The best performing set of features from *Table 3* were chosen and then Sigma was varied. Sigma varied from 10^-5 to 10^5, increasing by a factor of two every attempt.

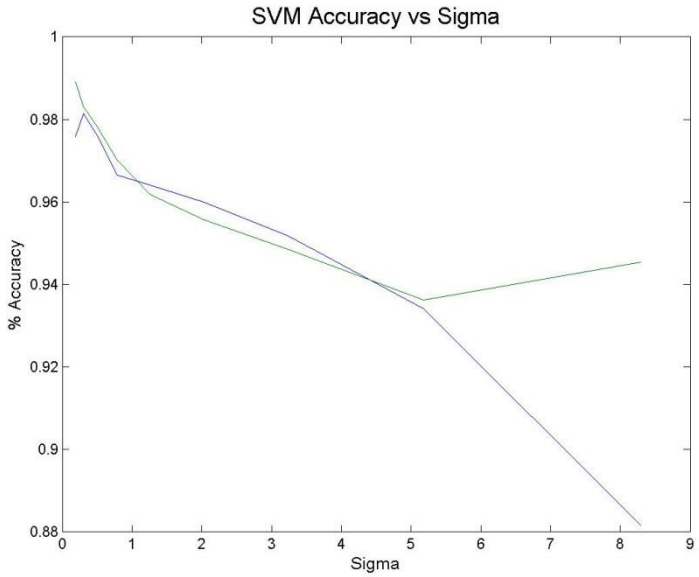


Figure 32: SVM accuracy vs. sigma for a sweep of sigma from 10^-5 to 10^5.

In many cases the training algorithm failed to converge within one million iterations; thus only parameters with successful convergence were shown. The data in *Figure 32* indicates a preference towards a small Sigma value of 0.3089 to achieve a correct classification accuracy of 98.9% and a correct rejection accuracy of 98.78%. Further training could be done with smaller increments around the peak sigma value; but any marginal improvement is likely not worth the computational time.

4.4 – Computational Setup

An important aspect of creating real time structured light system is the computational architecture and performance enhancements used to achieve real time results. Thus this section will discuss the hardware and software architecture.

The computational hardware required would be considered midrange by current standards, with graphics processing unit performance being most important for real time operation. The real time depth camera SLAM algorithm requires a significant amount of GPU memory to maintain a single TSDF volume. A single voxel is stored as a 32bit floating point value, thus a single 512x512x512 volume contains 134,217728 voxels which corresponds to about 512MB of video card memory. To put this into real world units, that means that a 3 meter volume can only be discretized 512 times, so points are discretized into 5.85mm^3 cubes. A video card with more memory can obviously have a higher resolution volume; but that comes at a significant computational tradeoff. The system used uses an Nvidia GTX 770m with 3GB of video memory, but it is only computationally powerful enough to use a volume of 512 x 512 x 512 at 15 frames per second. High end computers can easily use larger volumes at up to 30 frames per second at significantly higher costs.

Profilometry correspondence and scene reconstruction is completely done on the CPU; and, while being CPU intensive, it is memory speed limited due to the excessive amount of per pixel

accessing used in the line finding algorithm. Thus to maintain real-time speeds software optimizations are necessary. A significant speed increase is to cache the location of the line from a previous image and use that location to create a search window for the next iteration. This assumes that there is little perturbation of the line between images which is valid for smooth continuous surfaces. In the case of a discontinuity, it is likely that the line will jump outside of the window. Thus upon not finding the line in a new image, when the line was present in a previous image, it is apparent that a new global search should be done.

Continuous data collection is reliant on acquiring, responding to, and processing data faster than the camera's frame rate, this currently is not possible while using a SVM for correspondence due to the large computational load. Instead of continuous data collection, the system described caches data into a processing queue, since profilometer data collection is assumed to be used only when high accuracy is desired.

The selected camera for swept plane scanning is a Logitech HD 920, gathers data at 1920 x 1080 resolution at a rate of 30 frames per second. To be able to process this much data in real time, it is necessary to use multiple threads and another caching scheme. This scheme utilizes a slave thread to gather camera data which is then queued for a processing thread. Processing a single image usually takes about 300 ms which is equivalent to 3 frames per second. Thus caching each image into a queue allows for temporary real time data collection. Since the system is designed to only use the profilometer for specific interesting features, it is reasonable to assume that it will not operate continuously. This wait between scans allows the processing thread to catch up on queued items.

5 – Calibration Procedure and Results

This section will discuss methods used for calibrating and validating the operation and performance of the swept profilometer, as well as a discussion of the effects of component selection on creating a swept profilometer. Profilometer calibration consists of camera calibration, laser to camera calibration, and actuator calibration. In this section each component will be calibrated and then the results of those calibrations will be discussed.

5.1 – Profilometer Calibration Verification

Profilometer calibration relies on several constituent components, first of which is camera calibration. Camera calibration determines camera parameters such as focal length, the optical center and lens distortion which are necessary parameters for all future profilometer calibrations. Laser calibration is then performed to determine the exact orientation of the laser beam relative to the camera which completes the profilometer calibration. Lastly to create a swept profilometer, the motion of the camera must be calibrated relative to the motor. So the motor encoder's repeatability will be tested and then the camera's motion will be calibrated.

Verification of profilometer data will rely on minimizing deviation from the calibrated models to the collected data. This deviation can often be analyzed in the form of millimeter difference between the collected data and the model. This millimeter difference metric is convenient since the system's goal is to measure 3D models in this unit, thus later performed reconstruction accuracy analysis can consistently be compared to calibration data.

5.2 – Camera Calibration

Camera calibration is the most fundamental calibration required for all structured light calibrations since both motor calibration and laser plane calibration require accurately knowing the camera parameters. These parameters should be close to the manufacturers' specifications, yet closeness to manufacturers' specifications is an insufficient benchmark for the quality of a calibration. Thus, the best benchmark for a quality calibration is the well-known pixel projection error. This error stems from having a known plane with known points on it, such as a checkerboard pattern. The ground truth projection of these points onto the image plane can be calculated based on known camera parameters and then the error between the found image points and the ground truth can be calculated. Obviously, this gives rise to the issue of calculating the model plane using image data points, and then to project those points back onto the image plane to calculate error. This error metric is the best estimate to how well the estimated camera parameters fit the real camera parameters.

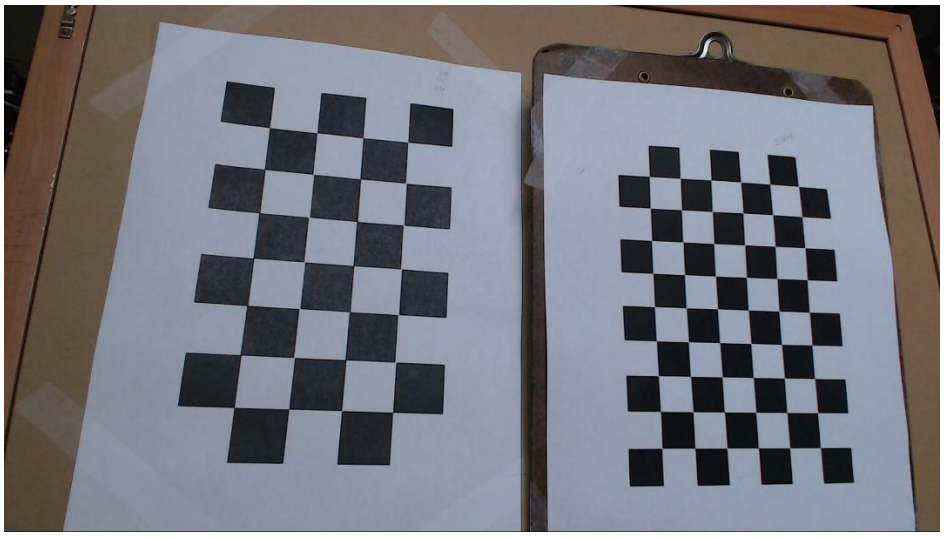


Figure 33: Two checkerboard patterns used to calibrate camera

Camera calibration was achieved with a set of seventy two images of two checkerboard patterns using GML's camera calibration toolbox, as seen in *Figure 33*. GML's toolbox uses two different sized patterns with different numbers of rows and columns which allows for a more accurate estimation of camera parameters. The results were as follows:

Parameter Name	Parameter value	+/- Uncertainty
Focal Length X:	1396.1 (pixels)	2.6
Focal Length Y:	1404.0 (pixels)	2.7
Principal Point X:	996.2 (pixels)	1.2
Principal Point Y:	575.0 (pixels)	1.7
Distortion K1	0.122714	0.003075
Distortion K2	-0.1940	0.16716
Distortion P1	0.004658	0.000416
Distortion P2	0.003433	0.000230
Pixel Error X:	0.65	
Pixel Error Y:	0.42	

Table 4: Camera parameters after calibration

As can be seen, the final pixel error is less than a single pixel in both X and Y. Thus indicating that after re-projecting a checkerboard pattern onto the image plane, the expected pixel value and the true pixel value varied by on average less than a single pixel.

5.2.1 – Checkerboard Pose Uncertainty

Many of the later calibrations rely on an estimation of the checkerboard plane relative to the camera. The position of this checkerboard plane is calculated based off of the calibrated camera parameters found in the previous section. Because there is uncertainty associated with the camera parameters, there is uncertainty in the checkerboard location. This section will analyze

the expected uncertainty to aid in determining the uncertainty in later calculations.

The checkerboard plane is determined by extracting corner points, projecting those corner points as rays into 3D space, and calculating the location of the checkerboard given the known measurements of the checkerboard. Uncertainty in this measurement comes from the camera parameters and an estimation of the corner pixel location on the image sensor. An uncertainty analysis of the projected ray can be performed with the camera parameters found above and an assumed 0.5 pixel uncertainty in estimating the checkerboard corner point.

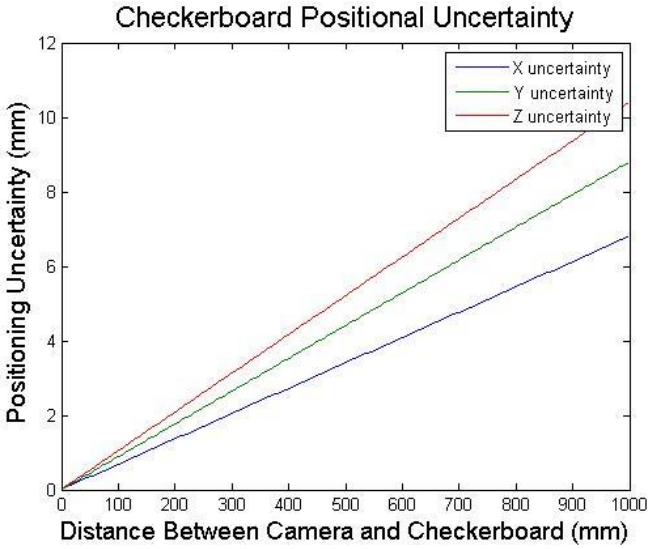


Figure 34: Checkerboard positioning uncertainty with distance to camera.

As can be seen, the theoretical uncertainty of a single point increases with distance, which is expected as the camera can no longer resolve small changes in checkerboard positioning. In practice, this uncertainty is a worst case scenario since it is an estimation of uncertainty of a single point whereas the actual checkerboard estimation involves minimizing error of all corners of the checkerboard.

Experiments were performed to determine the effect of this optimization technique on improving the checkerboard location estimation. To do this, the camera and checkerboard were placed in a static location for a set of twenty images. For each image, the checkerboard orientation is estimated. The standard deviation in checkerboard orientation among a set of images at a given location is estimated and then the camera is moved. Below are plots showing the standard deviation of checkerboard pose estimation at static camera and checkerboard positions.

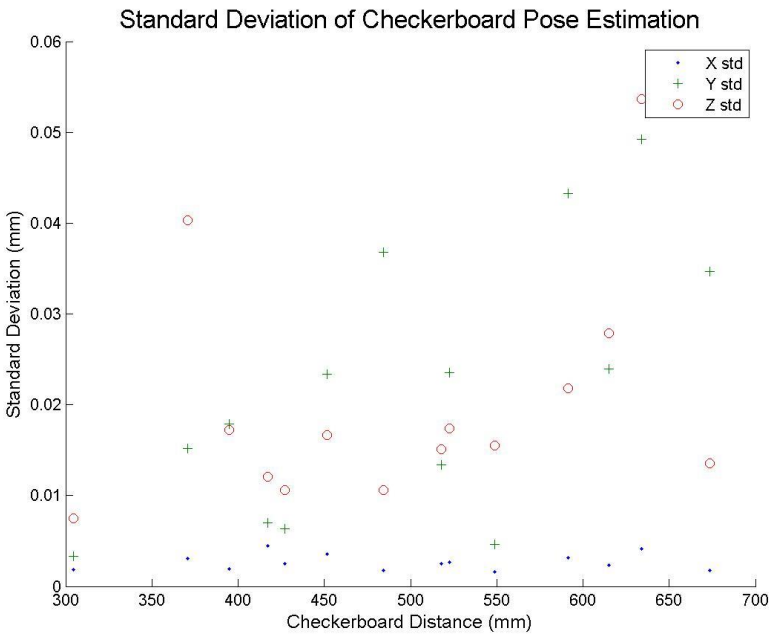


Figure 35: Standard deviation of checkerboard pose estimation vs distance between camera and checkerboard.

As can be seen in *Figure 35*, the lowest standard deviation is in the estimation of the X position. This is due to the fact that the checkerboard pattern used has the most detectable features in this direction. As well, this is the direction in which the camera coordinate system has the least uncertainty with regards to measuring camera parameters, as seen in *Table 4*. Checkerboard position estimation in the Y direction is higher due to fewer features, and estimation in the Z direction is highest since it is derived from other measurements and parameters with high variance, namely the focal length. In all counts, the standard deviation at any given distance is less than 0.06 mm, which would indicate a high confidence in checkerboard plane estimation.

5.3 – Structured Light Calibration and Verification

Structured light calibration was performed after camera calibration on a set of thirty images. In each image, the laser plane was projected onto a planar checkerboard. The images are post-processed to remove lens distortion. Then the checkerboard corners were extracted and the plane of the checkerboard was calculated with MatLab's camera calibration toolbox. The methods discussed in section 3.1.4 were used.

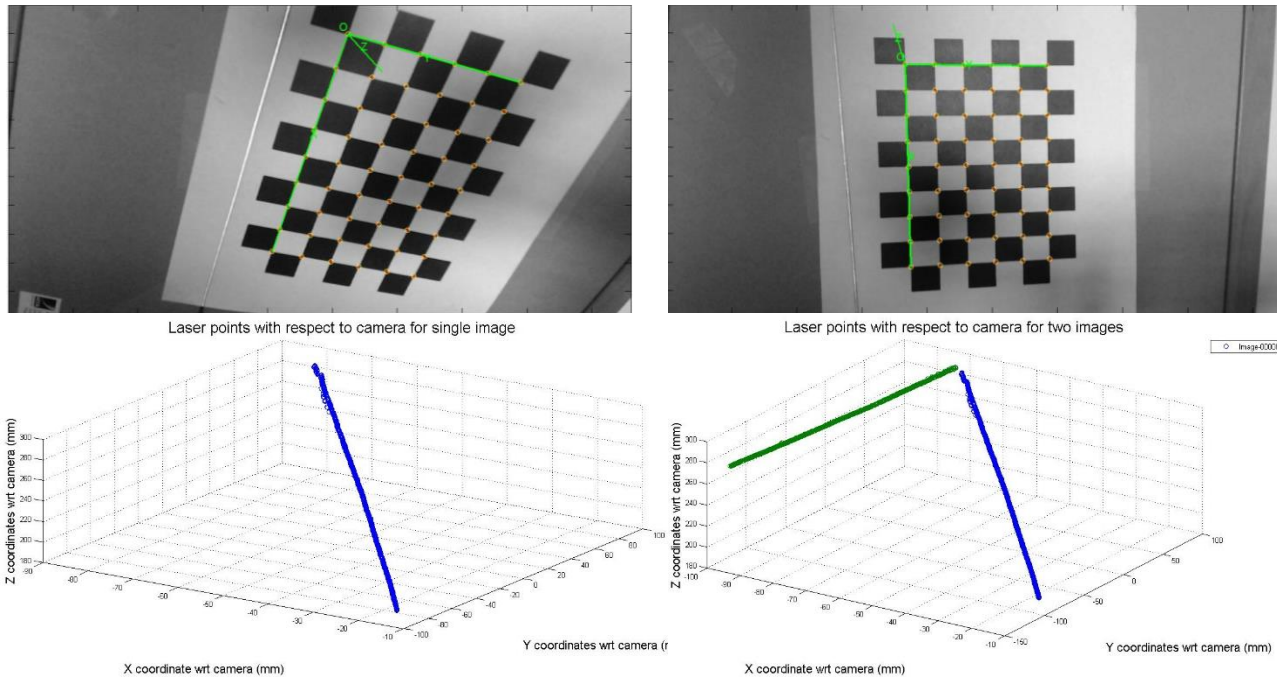


Figure 36: Corner extraction and resultant laser line in 3D space.

As can be seen in *Figure 36* the line of intersection of the laser plane with the checkerboard plane can be extracted in 3D space relative to the camera. The concatenation of these lines can then begin to form a plane in 3D that defines the plane of light that the laser creates, as seen in *Figure 37*.

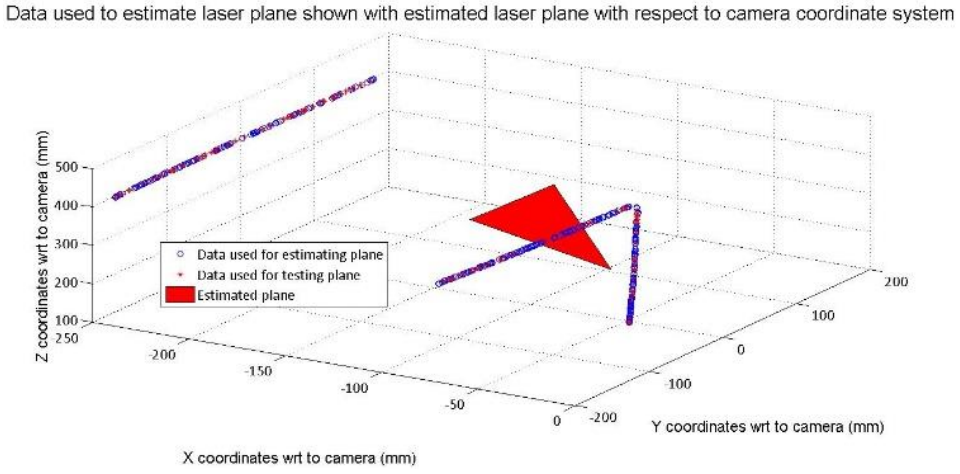


Figure 37: Estimated laser plane from collected data

The Least Squares (LS) plane fitting and Principal Component Analysis (PCA) techniques can be applied to the concatenated intersection line data to estimate plane parameters. Both techniques have merit, Least Squares is computationally complex as it involves minimizing error across all points. Principal component analysis determines the direction of most variance of the covariance matrix, which is a well optimized technique in the form of solving for the Eigen Vectors of the covariance matrix. In Matlab both techniques have been implemented, PCA generally found a solution faster than LS fitting; as well, PCA can be used for fitting a straight line to data, which is useful for other analyses. The residual error from this plane fitting is calculated and a histogram is plotted in *Figure 38*.

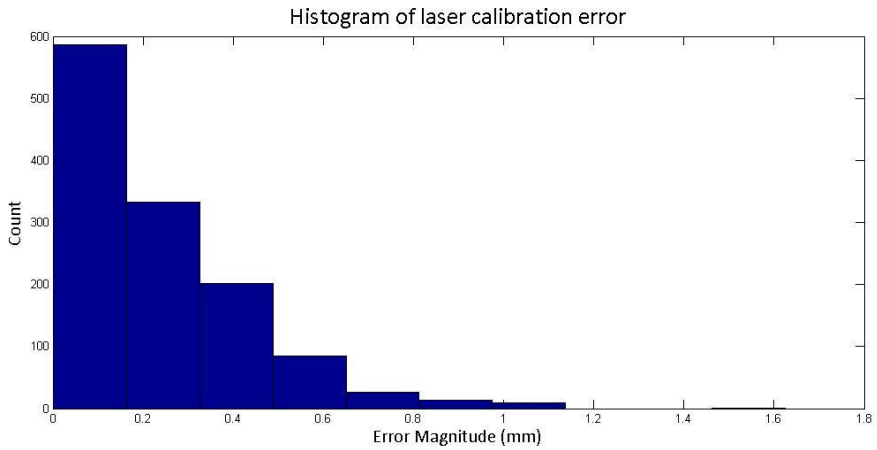


Figure 38: Error of the estimated plane model.

The mean residual was 0.23 mm with a standard deviation of 0.2 mm. The residuals in this case are intuitively related to the distance between a point and the estimated plane. Thus an average residual of 0.28 mm indicates that on average the points in the dataset vary by 0.28 mm from the estimated plane. The primary source of error in this calibration is the improper exposure of the laser line in images with proper exposure of the checkerboard pattern.

5.4 – Motor Encoder Validation

All camera position values are calculated based off of the assumed correct motor encoder position values. Thus the first motor validation step is that of the motor encoder's positional repeatability. Repeatability is of primary importance since if positional accuracy is skewed by a constant value, then the calculated camera to servo ridged body transformation can account for that skew. Repeatability will be measured as follows:

1) Acquire an image of a checkerboard pattern at a set encoder position value
2) Move the camera to a different encoder position value
3) Move the camera back to the original encoder position value and re-acquire an image

At each frame the position of the checkerboard pattern relative to the camera is estimated, thus with a static checkerboard pattern, the estimated positions should be the same between frames given the same encoder position. The positional error of the encoder is quantified as the Euclidian distance between two estimated positions given the same encoder position value.

Both servo configurations are tested. In the case of the Dynamixel, the motor is sent to a specific position; upon reaching that value an image is acquired. Since the HS322HD effectively runs in open loop with respect to the controller the motor is sent to a specific position then the host controller waits 250 milliseconds before acquiring an image. The error for seven sets of data is presented in *Table 5* and *Table 6* for both motors:

Encoder position	Average Error (mm)	Standard Deviation (mm)
-15	0.46	0.20
-10	0.49	0.25
-5	0.49	0.19
0	0.42	0.11
5	0.47	0.12
10	2.46 (0.47)	2.04 (0.12)
15	0.34	0.14
Average:	0.73 (0.45)	0.43 (0.19)

Table 5: HS322HD repeatability results.

Encoder position	Average Error (mm)	Standard Deviation (mm)
-15	0.19	0.08
-10	0.15	0.07
-5	0.09	0.08
0	0.22	0.17
5	0.15	0.07
10	0.16	0.08
15	0.16	0.09
Average:	0.16	0.09

Table 6: Dynamixel repeatability results.

The above tables contain the average placement error for six images at each encoder position. Each image is compared to all other images in the set to create this average error, thus the error is between six pairs of data.

For the HS322HD the standard deviation and average error are generally relatively low with the exception of the set associated with a position of 10 degrees. This position has one outlier that could be explained by the image being out of focus. If that outlier is removed then the average and standard deviation at that encoder position fall into the expected range of all other positions, as seen by the values in parenthesis. This evaluation indicates that the HS-322HD can only be trusted to position the camera to within 0.73 mm, the outcome of this will be explored later.

The Dynamixel MX-28R digital servo performs generally much better than the HS-322HD, achieving more than three times higher positioning accuracy than the HS-322HD. In general, the positioning of the Dynamixel is quite accurate and varies very little between motor positions. The standard deviation of these measurements is also within an order of magnitude of the standard deviation of measurements from checkerboard pose estimation, which would indicate that the Dynamixel does not add a significant amount of variance onto the pose estimation variance.

5.5 – Camera to HS322-HD Homogeneous Transformation Calculation and Validation

Motion calibration was performed on a set of four hundred and fifty images for each digital servo. Each image taken is saved with an associated encoder position which is used to calculate a homogeneous transformation for the motor. An automated extrinsic calibration script is then run to extract the transformation between the checkerboard pattern and the camera. Thus the transformation of the motor and the camera are known at each frame and the resulting rigid body transformation between the motor and camera needs to be calculated. This transformation is calculated using Tsai's method as previously discussed using open source code provided by Zoran Lazarevic. The resultant transformation for this system is displayed in *Table 7* for the transformation calculated with the HS-322HD digital servo.

0.99	0.00	0.14	-106.84
-0.02	0.99	0.09	0
-0.14	-0.1	0.98	18.75
0	0	0	1

Table 7: Rigid body transformation between camera and actuator.

The above values accurately describe the location of the camera coordinate system relative to the axis of rotation created by the servo motors used and can be used in *Equation 12*, as the 'H' matrix to transform a point from the camera coordinate system to the axis of rotation of the servo.

The actual displacement, as measured by optical methods, can be compared to the calculated displacement, which is calculated through the motor encoder position and homogeneous transformation. The histogram shown below is for the full set of data used to calculate the homogeneous transform.

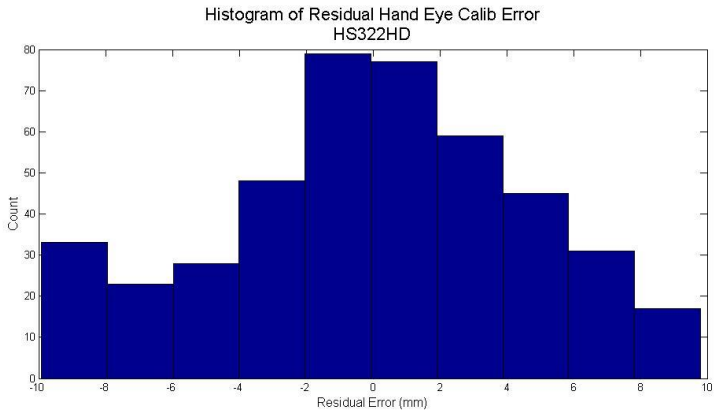


Figure 39: Calculated and Measured displacements for HS322HD.

Figure 39 shows the histogram of error between the measured displacement and the calculated displacement for the HS-322HD digital servo. The calculated movement is based on the calculated position of the camera from the motor encoder position output and the homogeneous transformation found in *Table 7*. The actual camera displacement values are derived from the homogeneous transformation between the camera and the static world coordinate system. Each point is the Euclidean distance between a frame and the previous frame. As can be seen the error

has a mean of -0.22 mm with a standard deviation of 4.86 mm.

As can be seen, there is a significant amount of variation in the actual positioning of the camera compared to the calculated positioning, with the HS-322HD digital servo, this is either due to poor estimation of the position of the camera relative to the checkerboard, or due to poor actual positioning of the camera by the motor. The later of the two options is the most likely case since while running the system, there is a perceivable alternating lag and jump in the image when there should be a continuous movement between frames.

The cause of such a positioning failure is either from a mechanical gear slip in the motor or from a lack of digital servo resolution. In the first case, the motor is told to go to a position, and attempts to do such; but, the slack in the gears combined with motor dynamics, causes the motor to overshoot the desired position. This problem could potentially eliminated by slowing the servo down to reduce the dynamic effects; unfortunately the HS-322HD digital servo does not allow speed adjustment. In the second case, the motor is told to go to another position but the motor does not perceive the difference in input signal; thus, it does not attempt to move to the new position.

5.6 – Camera to Dynamixel Homogeneous Transformation Calibration and Verification

This section will mirror the previous section except with data collected for the Dynamixel MX-28R. This data will be useful for comparing the quality of the two motors, and quantifying the performance gain achieved through a higher precision digital servo.

The dynamixel digital servo portrays significantly different characteristics to the HS322-HD. The results of the rigid body transformation given a similar dataset collected with the dynamixel digital servo indicate similar values, but the error histogram paints a significantly different picture.

0.99	0.04	0.03	-79.81
-0.03	1.00	0.05	0
0.02	-0.05	1.00	45.87
0	0	0	1.0000

Table 8: Dynamixel mounting homogeneous transformation.

As can be seen, the Dynamixel mounting location places the camera closer to the rotational axis of the motor. A similar displacement plot to *Figure 39* can be found in *Figure 40*.

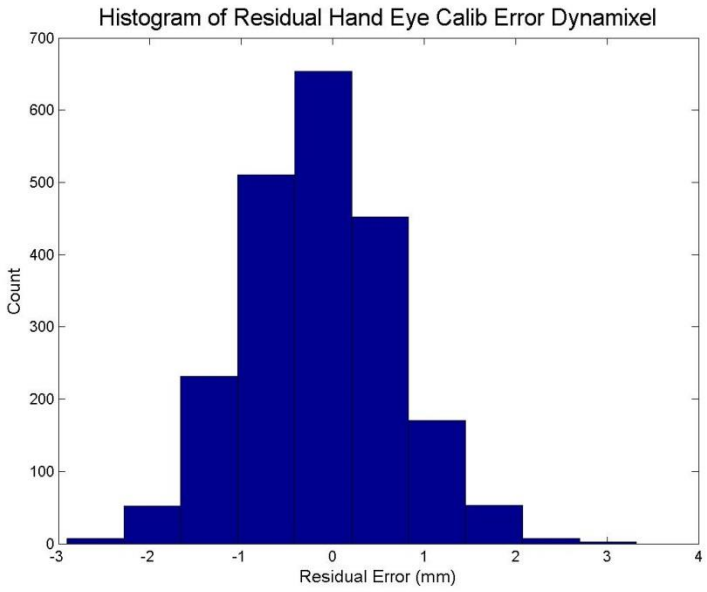


Figure 40: Calculated and Measured displacements Dynamixel.

As can be seen, the hand-eye calibration for the Dynamixel results in a function that can very closely match the actual movement of the camera. The average of absolute differences in the displacements vary by 0.03 mm with a standard deviation of 0.72 mm. This is indicative of a much better fit to the motor motion than found with the HS322-HD. This is indicative of more than forty times improved positional accuracy over the HS-322HD; which is likely due to the higher quality of the Dynamixel servo.

5.7 – Calibration Results

The result of the previous section is a calibrated actuated profilometer. The practices presented were the result of countless attempts to determine the best procedure for creating a quality calibration. As a result of this, the model for the actuated profilometer contains error on the order of 0.28 mm with respect to the deviation of the laser beam model, and 0.0315 mm with respect to the deviation from the motor actuation model. As well, it is now statistically proven that the Dynamixel digital servo performs significantly better than the HS322HD in camera positioning. The result of these calibration procedures on real world reconstructions will be discussed in the following section.

6 – Reconstruction Procedure and Results

This section will discuss the procedure for gathering data with the described system as well as the results of those reconstructions. First SLAM will be analyzed to determine accuracy of reconstructions and then localization accuracy will be studied. Then profilometer based reconstructions will be analyzed to determine the accuracy of local defect detection and quantification.

6.1 – Ground Truth Production and Comparison Techniques

All measurements gathered by this system need be validated against a ground truth to verify accuracy. There are several ways of producing this comparison model. The simplest is to take measurements of known features and then directly compare those measurements to the gathered data. A more complicated method involves creating a reference reconstruction of the test model and then using point cloud statistics to determine the error between the models. While being more data intensive and more representative of holistic scene reconstruction, the second approach is susceptible to the approximation algorithms used to create a comparison. That being said, a manual comparison also involves user input and thus the final accuracy results are dependent on the user's ability to collect data.

Since mobile robotic inspection is generally used in post-production inspection, features of interest in these tests will be chosen to represent wear and damage from use, rather than to maintain tolerances in a manufacturing environment. Interesting local features include chipping of a coating, surface pitting, cracks, and material buildup. Thus positive and negative features will be selected with small depth differences with respect to the base surface. Thermal barrier coatings are typically 0.1 ~ 0.2 mm thick [101], thus the system should be able to resolve such a small thickness. Pipe pitting is characterized as problematic when it reaches 20% of the pipe thickness and critical at 45% [102]. Thus for large nuclear power plant pipes this type of defect can reach several millimeters in depth. Cracks are a complicated problem because of their high aspect ratio and unpredictable geometry, but similar systems require cracks to be at least 1 mm deep for detection [103]. To simulate these defects, the ground truth object will have positive and negative features in the 0.1 to 0.5 mm range.

Global features should be representative of what could be found in large gas turbines, pumps, and power plants. Such features include turbine blades, flame holders, and valves. These objects can range in size drastically due to the application thus for the application of power plant gas turbine inspection a global feature size between 10 cm and 100 cm will be used. The global feature test scene can be seen in *Figure 41* where several books are placed such that key points can be selected and measured easily on a table. These objects were chosen due to their rigidity, such that measurements would not change between scans, and their reflectivity, since their covers represent a wide range of possible materials and colors.

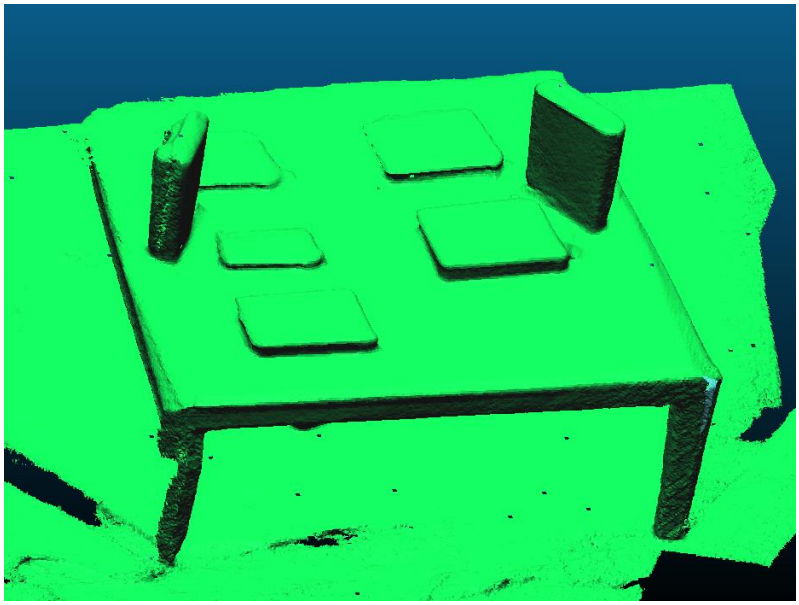


Figure 41: Three-dimensional reconstruction of the global test scene.

Local features are measured off of a wooden board with small paper cutouts glued to the surface to represent positive features, as seen in *Figure 42*. The difference in height between the base plane and the top of the cutout can be used to analyze the depth resolution of the system.

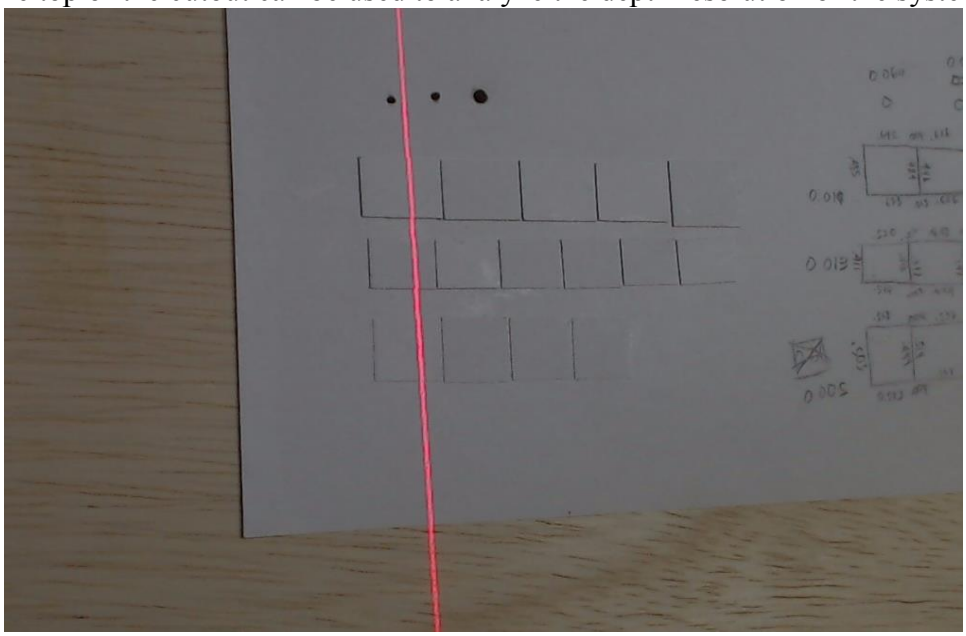


Figure 42: Simulated local features, from view of camera.

6.1.1 – Comparison with Manual Measurement

Manual measurement and comparison involves physically measuring the test object with industry standard contact based measurement tools, and then comparing those values to values hand selected from the scanned point cloud. To do this, one must follow best practices for

acquiring accurate physical measurements, and then manually select corresponding points in the point cloud. The distance between corresponding points should thus then be comparable to the manually gathered measurements.

6.1.2 – CAD Reconstruction

Computer Aided Design (CAD) reconstruction is possible via manual measurement of key points and then reconstruction based on those points. With simple scenes or under ideal manufacturing conditions this could be used to create a very accurate ground truth comparison. But in the created environment this can be very difficult due to unaccounted for variations in the specimen compared to a created CAD model. In the case of the test object, it is impossible to create an accurate CAD reconstruction since in the global case the features may be placed at unknown angles relative to each other and in the local case the features are too minute and difficult to measure to create a proper reconstruction from hand measurements and the tools available. However discussion will be presented for doing this comparison in cases where a CAD model does exist.

Given a point cloud and a CAD model, their units and coordinate system must be consistent. To ensure this the CAD model should be exported into a compatible format in the correct units of the reconstruction software. The coordinate systems can then be manually aligned to a rough estimate and then ICP can be used to finalize the alignment. The residuals from the final ICP alignment can then be used to determine model inconsistencies. Iterative Closest Point calculates the normal distance between each point in the point cloud and the underlying surface from the CAD model; thus each point in the point cloud has a distance which can be used to colorize the point cloud based on inconsistencies as seen in *Figure 43*.

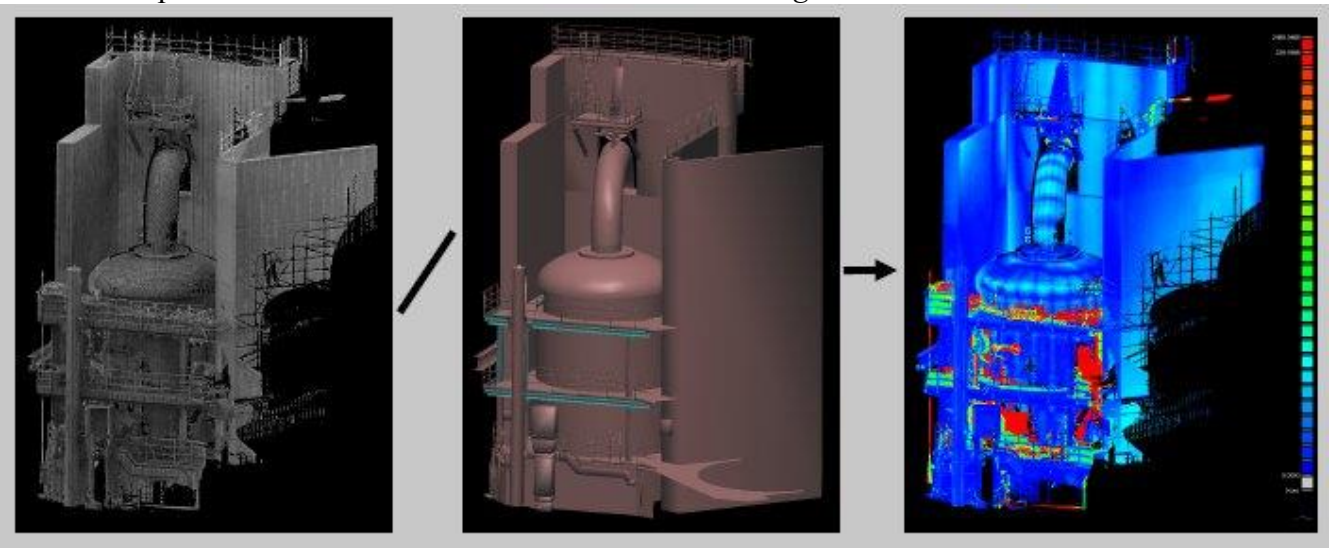


Figure 43: Colorized point cloud based on inconsistencies with original mesh. *D. Girardeau-Montaut, "Distances Computation - CloudCompareWiki," Cloud Compare Documentation, 2012. [Online]. Available: http://www.danielgm.net/cc/doc/wiki/index.php5?title=Distances_Computation. [Accessed: 26-Feb-2014]*. Used under fair use 2014.

6.2 – SLAM Accuracy Analysis

Simultaneous Localization And Mapping is important for two main reasons, it provides a reference frame for the profilometer scans, and it provides relative pose information between multiple profilometer scans. Simply put this means that the reconstructed environment gives a context in which a profilometer scan can be interpreted, as well as placing those scans in accurate locations relative to each other. Contextual features are considered large features that are visually recognizable such that an inspector can identify an important part. Such features can be large pipe intersections, turbine blades, flame holders, nuclear control rods, and other uniquely identifiable landmarks. When analyzing the results of these reconstructions, it is important to remember the context. These features should be accurate enough to uniquely identify a location in the environment, but they need not be accurate enough for defect measurements. This is primarily due to the fact that a system that can gather data in a large volume cannot gather detailed measurements in a small volume, thus the need for sensor fusion.

6.2.1 – Kinect SLAM Reconstruction Accuracy

The SLAM algorithm used in this thesis uses several approximations in order to perform at real time frame rates. Because of these approximations there is some accuracy loss. That accuracy loss will be analyzed in this section with respect to scene reconstruction. This will provide a basis for how accurate a global reconstruction can be without including information from the swept plane scanner. To do this a scene is reconstructed, then global features are compared to hand measurements. Of critical importance is the comparison of global features since local features can be accurately scanned with the swept plane scanner.

Global reconstruction accuracy will be evaluated by creating a reconstruction of a reference scene and then measuring global features from that scene. The error of these measurements can be seen in *Figure 44*.

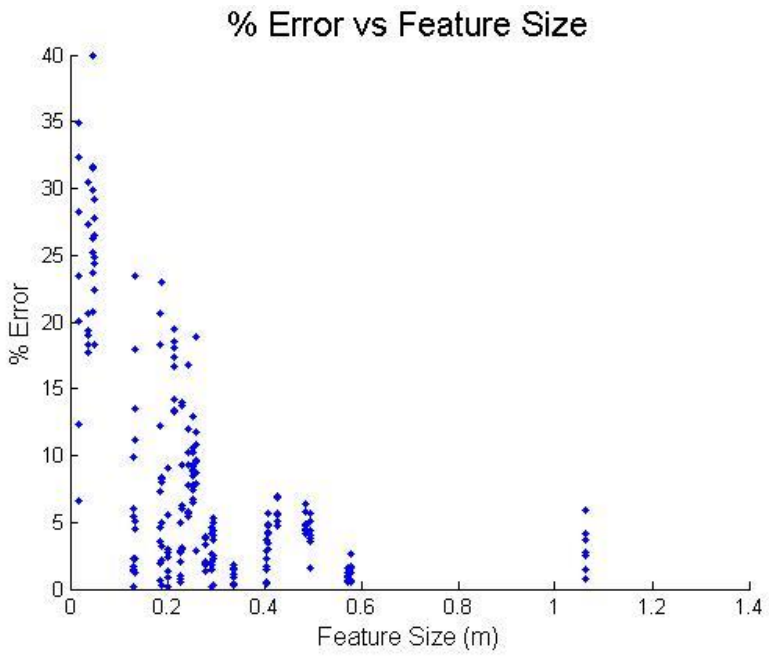


Figure 44: SLAM reconstruction accuracy for several different objects, for features in all dimensions.

Figure 44 shows a trend where there is significantly higher error when trying to measure small objects. This provides an idea of the scale of what features can be measured using the SLAM algorithm and this specific depth camera.

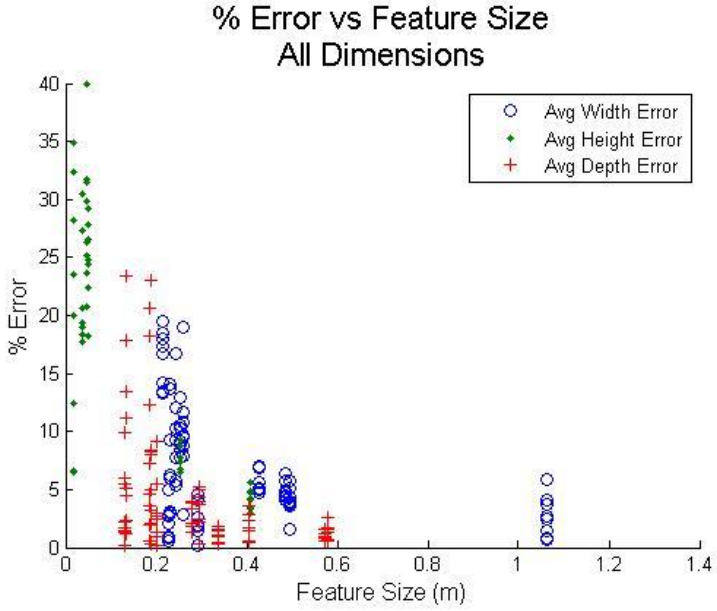


Figure 45: Error in feature measurement vs. feature size through SLAM reconstruction

Again this plot is indicative of the measurements previously discussed. There is no significant trend when measuring objects with respect to height, width or depth. But it is again clear that the trend of poor measurement accuracy is related to the size of the feature.

6.2.2 – Kinect SLAM Localization Accuracy

Previously SLAM reconstruction algorithm accuracy was discussed with respect to global scene reconstruction, now it is important to discuss the accuracy of the SLAM algorithm with respect to locating the sensor head. To do this, the sensor rig is moved by a known translation and then the known translation is compared to the pose given by the SLAM algorithm. The linear translation is measured by a set of calipers in increments of 12.5 mm, while the SLAM algorithm gives a homogeneous transformation matrix for the pose of the sensor head. Assuming that the motion is purely translation, the SLAM algorithm's estimated transformation can be extracted from the translation vector in the homogeneous transformation matrix. The Euclidean distance between the pose at the initial position and the pose at a later position can then be compared to the distance measured by the calipers to estimate the accuracy of the SLAM algorithm.

Data will be collected for several configurations of voxel size as well as with contextual information and without. Contextual information will be defined as a prior information from the scene before translational measurement begins. Context will be gathered by viewing the scene from several viewpoints before beginning any measurements. The average error for all datasets is displayed below.

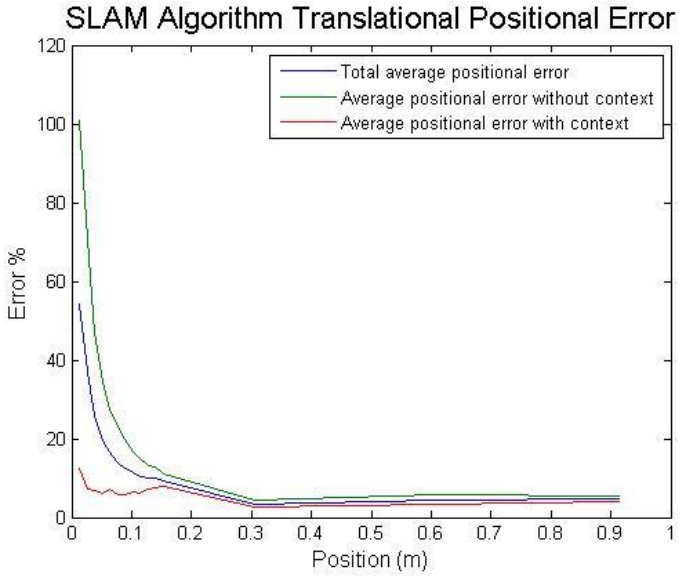


Figure 46: SLAM localization accuracy

The above plot shows the average errors for each measured position across all configurations used in blue. The green and red lines indicated datasets with no prior contextual information and prior contextual information respectively. It is apparent that the added information from context greatly improves localization accuracy. As well, it should be noted that position measurements have the least context at the beginning of the measurement and the most at the end, and thus as the X-axis increases so does the amount of scene information.

Accuracy can theoretically be improved by increasing the number of voxels, or the voxel grid size. The number of voxels is a number defining the size of the dimensions of the TSDF volume.

Increasing the number of voxels while maintaining a constant volume size, would reduce voxel size and the quantization error of converting a 3D point to a TSDF point. The errors for several voxel grid sizes are plotted below.

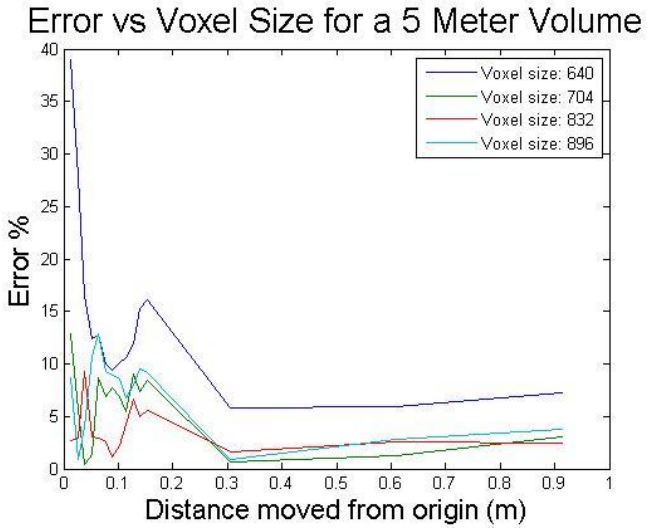


Figure 47: SLAM localization accuracy for different voxel sizes.

As can be seen, the lowest voxel size does have the highest average error, but all larger voxel sizes have very similar errors. This could be due to the fact that a 5 meter volume is discretized into 7.1 mm cubes at a voxel size of 704, which is a smaller voxel than the random error of the sensor used [64]; thus any reduction of quantization error is dwarfed by the Kinect's sensor error.

Higher accuracy applications could always use a higher performance depth camera or fusion with another localization technique such as dead reckoning or odometry to achieve higher localization accuracy. In general the localization accuracy is sufficient to locate local scans relative to global features with sufficient accuracy to understand the context of the scene.

6.3 – Profilometer Accuracy Evaluation

In order to characterize the swept profilometer's resolution without errors caused by motor encoder precision and motor to camera calibration, a single frame will be reconstructed such that there is no actuation involved. This single frame will reconstruct two objects, a flat plane, and two parallel flat planes with a known distance between.

The reconstruction of a flat plane via a single frame will give a line in space on that plane. The parameters of the line can be extracted from the 3D point cloud data and the residual error of that estimation can be used to analyze the accuracy of the profilometer. The reconstruction of two parallel planes can be used to estimate the smallest discontinuity that can be measured. The actual discontinuity difference will be measured by hand and then those values will be compared to values measured through the optical profilometer.

The line estimate will be done via singular value decomposition of the point cloud from a single frame reconstruction. The vector associated with highest variance is the directional component

to the line, while the mean value in all three dimensions is a point on that line. The residuals for each point are calculated via the point to line distance which is given by *Equation 40* [105], where V is the vector in the direction of the line, P_L is a point on the line, and P is the query point.

$$D = \|(P - P_L) - ((P - P_L)^T * V) * V\| \qquad \textit{Equation 40}$$

The average distance between a point and the line are given in *Figure 55* for various distances between the sensor and the test object.

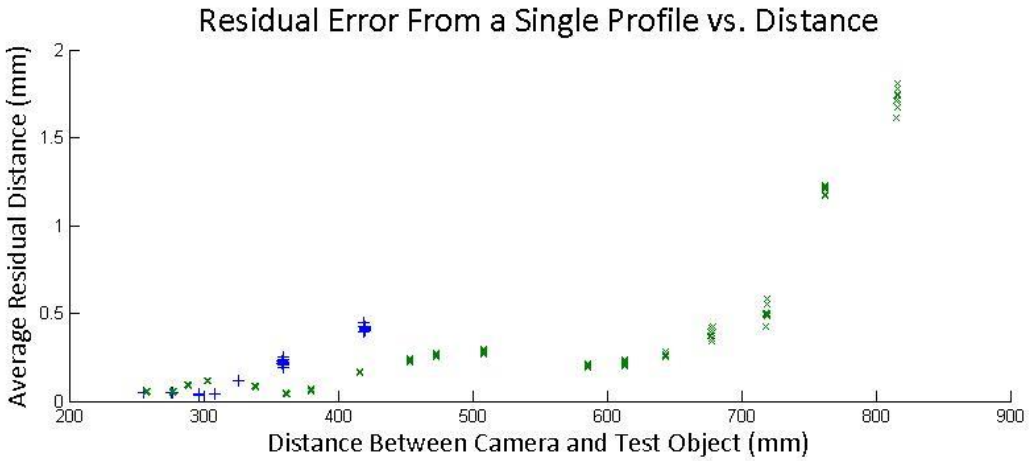


Figure 48: Profilometer residual error for a single line at various distances.

As can be seen, there is a general increase in average error with distance to the test object. This was predicted in section 3.1.2 in the discussion of theoretical sensor resolution. The error flattens out at extremely close ranges, which is believed to be due to the camera's auto-focus not focusing on the laser, or due to the laser beam being too close to focus on. The minimum error is at 133.65 mm between the camera and test object with a value of 0.06 mm average residual error.

The absolute positional accuracy can also be analyzed with respect to the profilometer, since each location of the test object can easily be located relative to the profilometer coordinate system. By plotting profilometer based distance measurements verses handmade distance measurements, one can determine the accuracy and repeatability of absolute distance measurements.

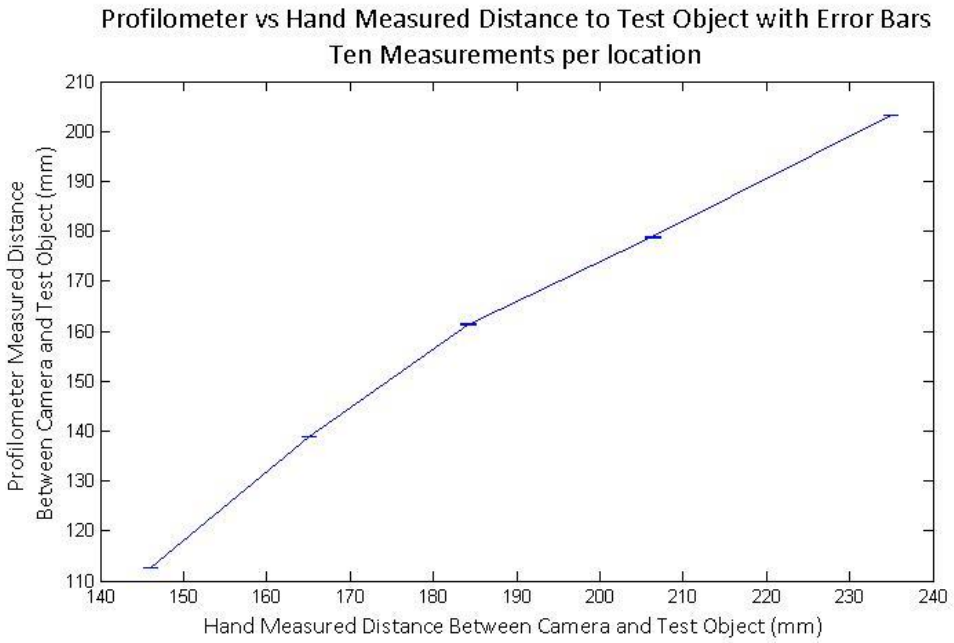


Figure 49: Profilometer absolute distance compared to hand measured distance.

As can be seen, the data is highly linear with very consistent profilometer measurements. The standard deviation of each measurement is less than 0.07 mm, which would indicate a highly repeatable distance measurement. The linear fit to this data gives a line with an intercept at -28.38 mm and a slope of 1.00 mm/mm. This would indicate that the profilometer is accurately calculating distances and that the manual measurements were not to the correct profilometer coordinate system which is the reason for the constant offset of 28.38 mm.

Reconstruction of known parallel planes will allow for estimation of the smallest discontinuity that can be measured. This test setup is created by gluing paper squares onto a wood backing at various stacked heights. The height of each stack is manually measured via a set of calipers. The test setup can be seen in Figure 50: Profilometer *Figure 50* and the typical image gathered from the camera can be seen in *Figure 51*.

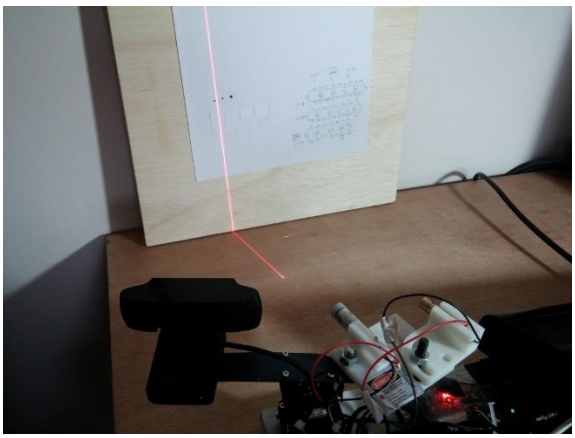


Figure 50: Profilometer test setup

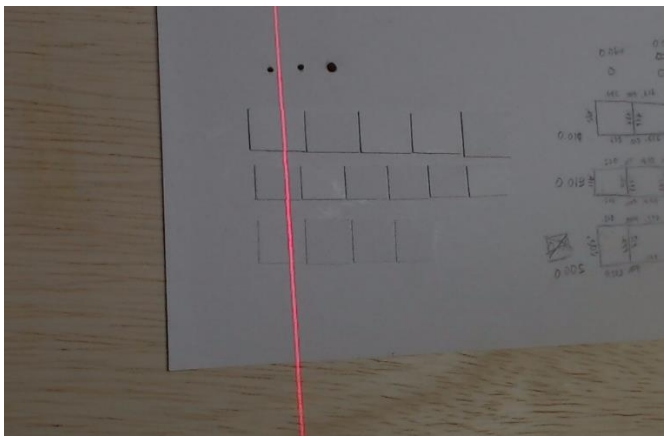


Figure 51: Typical profilometer image.

The typical output from processing the profilometer image is a point cloud representing the intersection of the laser beam with this surface, as seen in *Figure 52*.

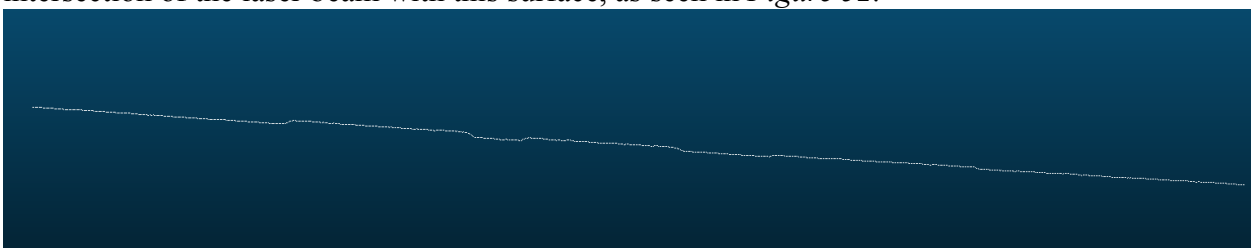


Figure 52: Single frame reconstruction of a piece of paper.

As can be seen, there is a perceivable height difference between the underlying surface and the piece of paper. Automatically extracting this difference in the presence of other noise sources is a non-trivial task and outside of the scope of this thesis, thus the data will be manually extracted. To do this, two points will be selected on both sides of the step and a third point will be selected on the step. The two lower points will be used to construct a line and the distance between the line and the third point will be calculated via *Equation 40*. The results of this are plotted in *Figure 53* for several different distances to the sensor.

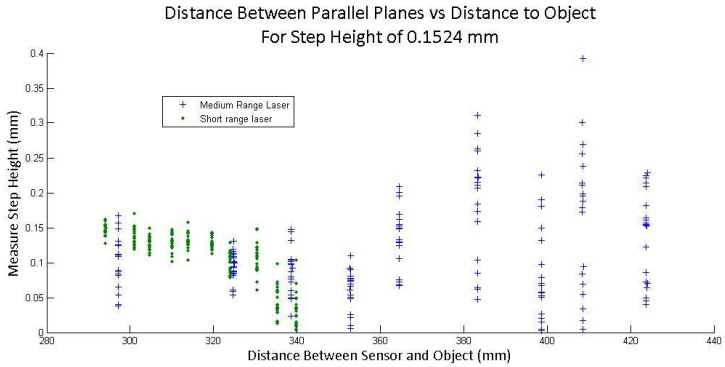


Figure 53: Estimated step height vs distance.

As can be seen, such a small step height can only accurately be measured with the close range laser at its optimal distance. The medium range laser creates too noisy of a measurement to accurately measure this feature at nearly all distances. It does show though that the medium range laser does have sufficient sensing power to detect the defect as statistically significant relative to the noise floor and thus it can adequately detect that the defect exists. The downward trend of the close range laser is indicative of too weak of a return to measure the surface at long distances.

6.3.1 – Effects of Exposure Control on Accuracy

As previously discussed, exposure control is done to maintain optimal accuracy by exposing enough of the beam to create an accurate estimate, while preventing over exposure. Section 4.2 discussed the need and methodology for exposure control as a primer for this section. This section will discuss the experiment performed to determine the optimal exposure values as well as a more detailed explanation of the results.

The optimal exposure is found by reconstructing a single profile of a flat plane. This single profile should be a flat line in 3D space. Thus the methods in section 6.3 will be used for line estimation and error calculation. Instead of estimating error versus distance to object, this experiment will attempt to quantify error versus average beam intensity as controlled by adjusting the exposure. To do so, the sensor is placed at approximately 133.65 mm from the test object, the location of optimal sensor performance as determined in section 6.3. The sensor will then perform a set of ten reconstructions at a full sweep of available exposure values. The line is extracted and residual error is calculated as discussed in section 6.3.

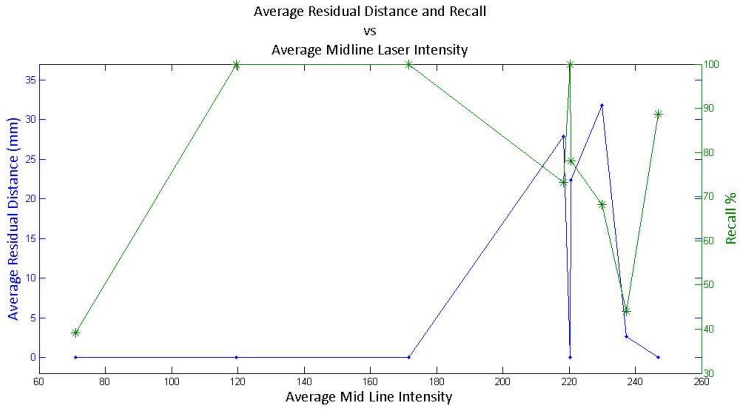


Figure 54: Profilometer residual error and recall for a single line at various exposures.

Figure 54 shows the residual error from fitting a line to the data collected from a flat surface. The blue axis is the residual error from this fitting process, while the green line is the recall. Recall is defined as the number of points extracted divided by the number of points that it should have extracted (one point for each row of the image). Average mid line intensity is the average pixel response at the center of the laser beam for each row of the image sensor. Data was collected at nine different exposures, which is all that is available on this camera. As can be seen in *Figure 54*, error and recall become erratic at higher laser beam intensities due to the sensor

being un-able to estimate the laser plane center of an over exposed beam. At lower intensities, the primary issue is correct correspondence, since the return is so faint that it is incredibly difficult to consistently find the laser beam, thus causing very low recall. Thus the optimal intensity from this graph is between 120 and 170. Finer collection of laser beam data could be accomplished by dimming the emitter or moving the sensor rig at a set exposure; but the effects of these parameters on Residual Error would lead to poor conclusions about the results.

The exposure control values used were determined from *Figure 54* to balance accuracy with the coarse exposure control available with this camera. The only parameter controlled is the exposure value on the camera. There are many more parameters for laser exposure control that could play a role in future systems. Such parameters include camera gain and laser power level that could help maintain a tight window on laser intensity. Unfortunately such a multi-parameter controller is out of the scope of this current thesis and should be studied as future work.

6.3.2 – Flat Plane Profilometer Reconstruction

Previously calibration errors were used to estimate the sources of error in the swept profilometer. It was found that for the local scans, the motor to camera calibration measurement for the HS-322HD digital servo was most error prone, at a standard deviation of 4.86 mm for the HS-322HD digital servo, while the Dynamixel has a standard deviation of 0.72 mm. While the laser point estimation error had a standard deviation of 0.2 mm. These error estimates hope to provide insight into what can be improved for better measurement accuracy, but they cannot be translated into the final measurement accuracy.

Final expected measurement resolution is related to the standard deviation of the residual error, since it is necessary for an environmental difference to be perceivable as more than background noise. Ten local reconstructions of a flat surface were performed at ten different distances. The standard deviation of the residual error between each point and the fitted plane is reported for each reconstruction in *Figure 55*.

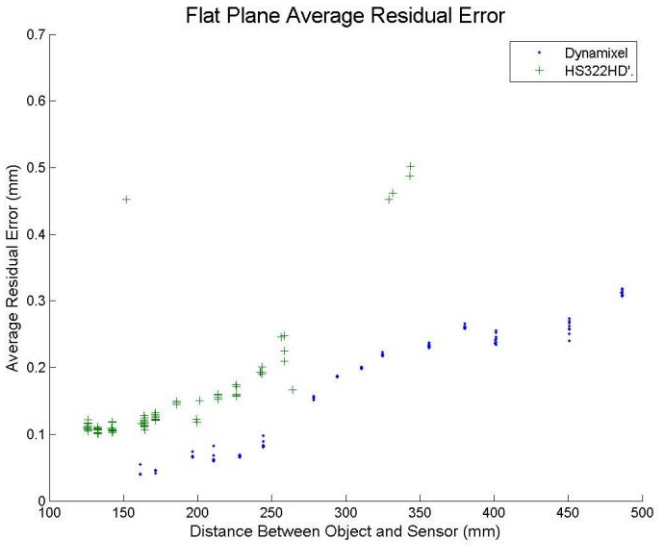


Figure 55: Average residual errors for one hundred scans at various distances with the HS-322HD digital servo.

As can be seen in *Figure 55,* the standard deviation for the HS322HD increases quadratically with distance from the reference object; which is to be expected from the sensor characteristics discussed in section 3.1.2. This plot varies from the expected values calculated in 3.1.2 for sensor resolution because this includes errors due to the motor encoder and camera to motor calibration. As well, this represents real world results from a non-perfect laser plane produced with real optics. In contrast *Figure 55* shows a similar trend for the Dynamixel with smaller slope, this indicates similar sensor characteristics with less error attributed from the motor.

6.3.3 - Object Local Feature Reconstruction

Local feature reconstruction with the actuated profilometer is performed to determine the depth reconstruction resolution of the actuated system. This differs from previous single profile analysis and flat plane reconstructions in that it will look at combined depth disparities over actuated sets of data.

Since prior results indicated that the Dynamixel configuration performed significantly better, this configuration will be used for all further testing.

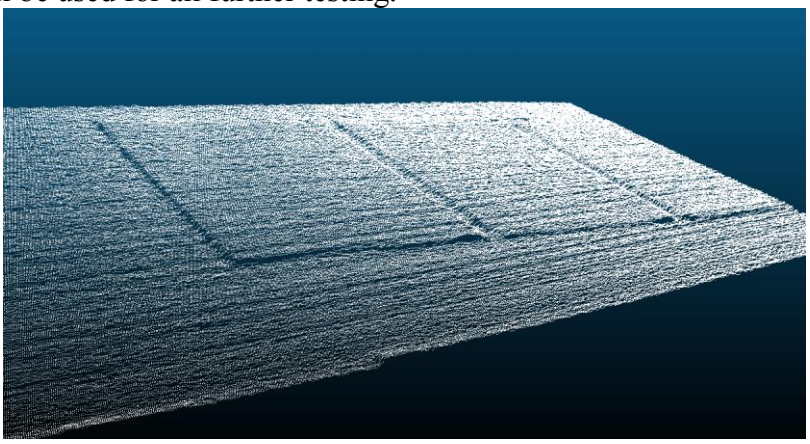


Figure 56: Reconstruction of one set of local features.

Reconstructions will be performed at distances between 55 mm and 320 mm, at intervals of approximately 20 mm. Each reconstruction will sweep the servo the required angle to reconstruct the whole area, thus larger angles will be swept at close distances and smaller angles will be swept further away. Exposure values will be maintained as best as possible, but the coarse exposure control available prevents perfect feedback control. The result of estimating a single step from the reconstructed surface can be seen below.

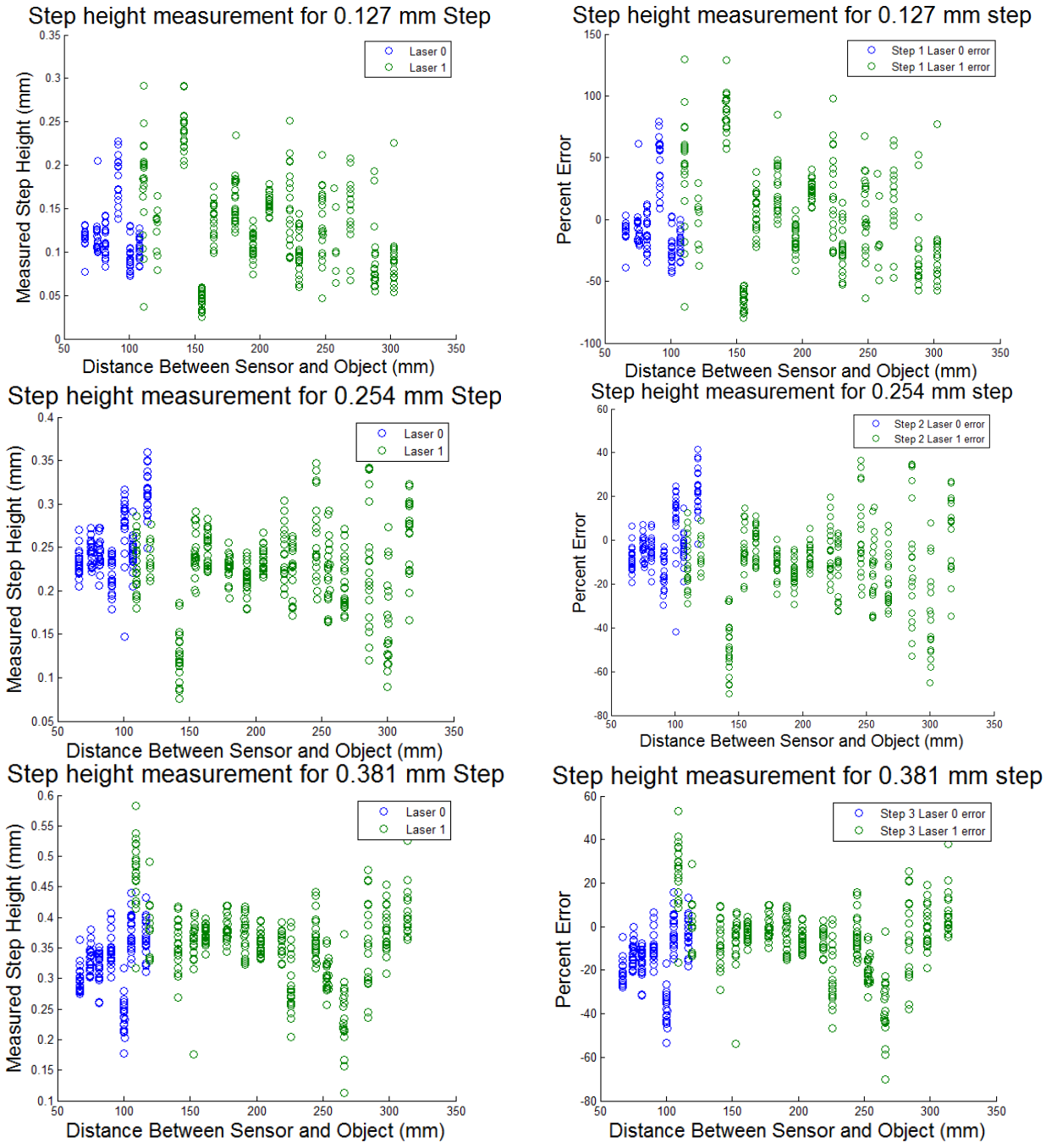


Figure 57: Estimated distance between parallel planes. Full 3D reconstruction with each laser.

As can be seen in *Figure 57*, the error of the reconstruction is effected by sensor noise, and as sensor noise increases, so does the apparent error. With respect to the matter of detection, the 0.13 mm thick piece of paper is human perceivable in the point cloud, as seen in *Figure 56*, at all distances displayed, thus indicating that this system would be sufficient for detecting defects of this order of magnitude. As a few notes with regards to the measurement accuracy found in *Figure 57*, the step height was hand measured with a pair of calipers, thus there is variability in

those measurements that isn't shown above. Thus a constant offset in the error could be due to poor ground truth measurement. As well, the points used to determine the step were handpicked and thus there is some variability in that.

As can be seen in the plots, there is significant variability in measuring the smallest step height of 0.13 mm with both lasers. This is due to the feature size being so close to the noise floor of the sensor that the noise is overpowering the signal from the feature. For the second step height of 0.25 mm, it is apparent that the there is a linear trend to the error of laser zero where the feature size is first under estimated and then over estimated. This would indicate that the laser either needs further calibration or that the camera focusing element changes the focal length as it adjusts focus. It is known that this camera's lens changes focal length based on focus; since there is a perceivable zooming effect at the focus is swept, so the latter of these explanations is likely. With laser one there is an initial high error due to the laser not achieving focus yet, the error then flattens as the laser achieves focus and then increases uniformly. This uniform increase follows the trend predicted in section 3.1.2. The high variability of the system can best be explained by two factors. Primarily the camera's automatic focusing element is poorly designed so that the focal length changes slightly with the focus of the camera, this changing of focal length changes everything that the camera perceives. Secondarily the exposure of the laser beam is very coarse, as previously determined, so optimal values cannot be achieved with this camera. The camera selected does have some faults but it should now be clear what can be achieved with what is readily available, thus the end application should drive the selected equipment. In this case, the resulting point clouds were sufficiently dense and detailed to detect these faults, which is the primary goal of this thesis.

6.4 – Sensor Fusion Results

Sensor fusion is achieved through manual registration of point clouds to determine the homogeneous transformation between the profilometer coordinate system and the Kinect coordinate system. This manual registration was done by selecting corresponding points between each point cloud then solving for a homogeneous transformation between the two. This process was done in Cloud Compare via the "Align Two Point Clouds" function. The result of this process is a homography matrix much like the one found in section 5.5 and 5.6. The matrix found by point cloud alignment is as follows:

1.00	0.01	-0.08	-232
-0.01	1.00	-0.02	-60
0.08	0.02	1.00	-50
0	0	0	1

Table 9: Transformation between profilometer and Kinect coordinate systems.

This indicates that the profilometer is to the left of the Kinect by 232 mm, above it by 60 mm and behind it by 50 mm, with almost negligible rotation difference. The result of using this transformation on a profilometer point cloud, is the alignment of the two clouds in 3D space as seen in *Figure 58*.

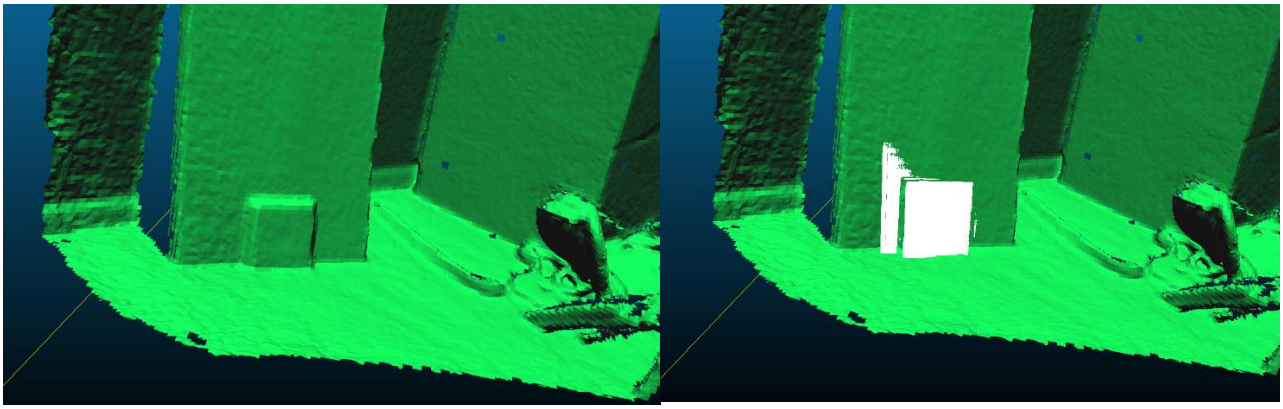


Figure 58: Profilometer point cloud overlaid with Kinect point cloud

Figure 58 shows the global reconstruction in green as created by the Kinect SLAM algorithm, before and after overlay of the profilometer point cloud. This overlay grants the contextual information necessary for general feature localization while allowing for detection of flaws at the sub millimeter scale, as proven in section 6.3. The overlay of data from the two sensors is not only useful for detecting local depth features, but it also grants a significantly increased point density, as can be seen in *Figure 59*.

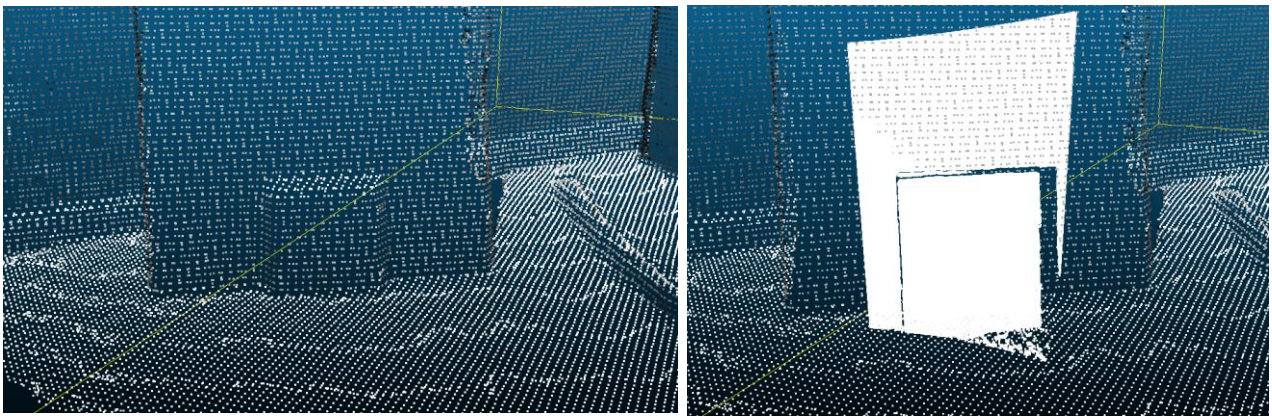


Figure 59: Raw point cloud overlay

This increased point density is useful for detecting faults such as cracks and pitting that may be too small for the Kinect to detect. This is due to the fact that Kinect's correspondence algorithm requires a sufficient number of matching points in an area to match correspondence, whereas with the profilometer a continuous beam can fundamentally detect smaller features.

7 – Conclusion and Future Work

This section will discuss the advantages and disadvantages of the proposed system found in the production of this work as well as areas of further improvement.

7.1 – Conclusion

This thesis' goal was to create a mobile robotic sensing package that can gather contextual scene information in real time while maintaining an accurate way of measuring and detecting defects, with the aim being to allow for defect detection and an approximate defect location to determine severity. To do this, this thesis created tools for creating a sensor fusion platform with widely available hardware. It has achieved detection of height differences on the order of 0.127 mm while collecting holistic scene data simultaneously for a cost of less than $500 in sensing hardware. The primary value of this thesis is a useable procedure for creating a capable structured light sensing tool with the necessary calibration routines for such a tool, as well as efficient algorithms for using the tool in real time.

7.1.1 – Actuated Profilometer Calibration

A significant portion of this thesis was the creation of algorithms for calibrating a system for acquiring highly detailed measurements with an actuated profilometer. The calibration of this system is algorithmically based such that no high cost external hardware is required. The camera model is determined by observing a checkerboard plane of known size. This camera model is then used as a basis for calibrating the laser with respect to the camera, by observing the laser beam in a known scene. The motor is then calibrated with respect to the camera by observing the motion of the camera as it is actuated. The result of this is a calibration routine that can accurately and quickly calibrate an actuated profilometer with just a laser, camera, servo motor and checkerboard pattern.

7.1.2 – Real Time Full 3D Simultaneous Localization And Mapping

Contextual information is gathered via a depth camera and a Simultaneous Localization And Mapping algorithm. Because of the intelligent nature of this algorithm, no other sensors are required such as accelerometers or motor encoders. Yet this algorithm is extremely computationally intensive, thus requiring GPU acceleration. The accuracy of this algorithm has been analyzed and determined to be acceptable for contextual information. This SLAM routine allows for the holistic reconstruction of infinitely large environments, through multiple TSDF volumes. But it is limited in small spaces due to the depth camera's standoff distance.

7.1.3 – Sensor Fusion

Sensor fusion was achieved through calibrating the two sensors relative to each other such that all profilometer data is placed in the same coordinate system as the SLAM reconstruction. This results in profilometer scans registered to global features as well as relative to each other.

7.1.4 – Notes on Improving Profilometer Performance

The following items have been discussed to some extent concerning the performance of a

profilometer with a given camera:

1. Effects of baseline found in section 3.1.2,
2. Effects of sub-pixel interpolation of the beam centroid found in section 3.1.2,
3. Effects of lighting control found in section 6.3.1
4. Effects of laser focus found in section 4.1
5. Effects of motor selection in section 6.3.2, 5.4, 5.5, and 5.6

Item one was discussed purely theoretically since the baseline of the manufactured rig was maximized to the available width sensor rig. Item two was discussed in theory since the discussion on accuracy of such an interpolation algorithm is outside of the scope of this work. Item three was empirically explored and found to have a strong influence on performance. Item four was discussed anecdotally since the configuration used cannot adjust laser focus while maintaining the same laser calibration. Item five was discussed in many forms and the superiority of the high end digital servo was determined. From that it was apparent that selection of new actuators should focus on positional repeatability, encoder resolution and mounting rigidity.

Some further anecdotal notes will follow which should help guide development of a similar system. These notes will focus on component selection to optimize the performance of such a system while minimizing cost.

Lens selection should be considered highly important to profilometer performance. As seen in *Equation 23* and *Equation 24* in section 3.1.2, increasing lens zoom factor linearly increases expected resolution. This should be intuitive since the camera will be able to resolve smaller differences at further distances than with a low zoom factor lens. With this in mind, it may be beneficial to explore variable zoom lenses to allow tuning of zoom factor for the needed task or a higher zoom lens to increase accuracy. The problem with a variable zoom lens is availability and calibration. At the time of this writing, there are no electronically controlled variable zoom lenses with appropriate working ranges for the discussed application. As well, calibration routines for such a system would need to be developed to incorporate feedback control of a changing focal point. Higher constant zoom factor lenses may be possible, but this imposes the possible issue of operator tunnel vision.

Camera selection is second most important in creating an optical profilometer. The two things to consider with selecting a camera include low sensor noise and high resolution. Random sensor noise has the potential effect of shifting the laser centroid calculation. Resolution should be intuitive since a higher resolution camera can resolve smaller changes in the laser beam. The tradeoff with such a high resolution sensor is the expected lower framer rate associated with it. Some systems allow selecting a region of interest in the image, this approach would allow for a high framerate at an effectively high resolution.

Laser selection is the least and most important component in creating an optical profilometer. It is the most important because a focused beam is very important for creating accurate reconstructions. It is potentially the least important component to invest in since a $20 laser diode module can perform very well in its field of focus. The system proposed successfully uses two lasers with different focus points to improve sensing accuracy, and any further systems could also use such an approach to keep costs low. Laser manufacturers advertise machine vision

lasers for profilometery from $200 to $1000 depending on configuration, with modules quoting the ability to focus a beam into the μm thickness range. Yet in a mobile robotics platform, the depth of focus is equally important to the minimum thickness, since it is unlikely to measure a surface at the exact optimal measuring distance. Thus the benefit of these more expensive modules needs to be closely analyzed with respect to the working environment. To build a flexible system it would be beneficial to create an array of calibrated lasers focused at different depths. Such a system could be made with many cheap $20 adjustable focus line lasers while potentially achieving better focus over the entire working range than a single expensive industrial line laser.

7.2 – Future work

This section will discuss areas of future improvements to this field given the groundwork provided in this current work.

7.2.1 – Component Testing and Algorithm Improvements

The components selected were chosen because they fit the requirements without significant modification, cost, and size. Because of this there may be more ideal components for specific tasks. Characterization of more components could improve future selection tasks. Some important areas for testing include automated camera optic characterization, camera sensor characterization, the effect of different lasers sources, the effects of multiple joints along with the associated calibration challenges and laser specific camera auto focusing. The specific of each will be discussed in the following.

Camera optics perform a critical role in 3D reconstruction, as such camera parameters must be stable in all operating conditions. A critical assumption is that the focal length does not change based on other parameters, such as focus. This assumption may or may not be valid given the quality of the optics and thus is an interested area of concern for the design of future systems. A method for evaluating optics should be established with respect to picking optimal optics for profilometer applications.

Camera sensor characterization is important because un-accounted for variation in the intensity of the laser return can cause for corrupted output of the intensity weighted centroid finding algorithm. Such a problem could stem from non-uniform sensor spectral response or electrical noise sources in the sensor itself. Characterization of these noise sources and their effects on the final system accuracy would prove valuable for further system design.

The system created assumes an infinitesimally thin plane, this assumption is most violated in situations where the laser beam is out of focus. Thus different laser sources could prove to be better approximations of flat planes, which would in theory provide for a higher accuracy system. This theory is backed up by an immediate improvement in accuracy found from transitioning from a pre-focused line laser to a manually focused line laser at close ranges. Validation of this theory and quantification would be desirable, as well as an evaluation of different laser sources.

The camera used in this thesis has built in autofocus which is not optimized for profilometer

tasks. Since the laser line is found in real time, and its location is known in 3D space relative to the camera, then it is feasible that an autofocusing system could be created to track the laser beam. This would alleviate any focus related accuracy issues at extremely close ranges.

7.2.2 – Calibration Algorithm and Methodology Improvements

Multiple joint calibration is important to many robotic operations since it is likely that a sensor head will require at least two degrees of freedom to provide sufficient flexibility to the operator. Thus to create accurate scans within the entire range of operation, all joints must be fully calibrated. The procedure discussed in this thesis is for one actuator, and while it is extensible to multiple joints, the potential for compounding error is present. Thus tests should be done to quantify if this procedure is still feasible for multiple joints.

Laser calibration is particularly problematic because proper exposure of the checkerboard often violates optimal laser exposure values. The effects of which are unknown but it is likely possible to reduce laser calibration error through an improved test setup. This improved test setup would gather pairs of images, instead of single images for calibration. Each image pair would have one image with the checkerboard exposed properly and one image with the laser exposed properly. It would be rather trivial to setup this procedure in software, and adjust the calibration routine. The difficulty lies in maintaining a rigid orientation of the profilometer and checkerboard between the two images, which is not even guaranteed with precision actuators.

The Graphics Media Lab of Lomonosov Moscow State Univeristy has developed a camera calibration technique that takes advantage of multiple checkerboard patterns to further improve calibration accuracy [106]. This technique works well for camera calibration, but is not well adapted for laser and motor calibration. It would be beneficial to adapt this technique to these other optical calibration tasks.

7.2.4 – Depth SLAM

The depth SLAM algorithm used naive iterative attempts at different poses to determine the registration between frames. Incorporating external location feedback could improve SLAM by having an estimated initial movement, but that task is outside of the scope of this thesis since it would require building a robot to actuate the whole platform. This however would be a trivial task to include in the development of a mobile robot. Fusion of SLAM localization data and odometry or dead reckoning has been discussed in section 2.13.2 and has been shown to be beneficial in other applications.

The SLAM algorithm also used uses a TSDF volume to quantize the continuous 3D space of the depth sensor. Another improvement could be to use a continuous space instead of a TSDF volume. Such a change may be infeasible due to the increased computational load, unless estimated transformations are provided from external sources.

7.2.5 – Integration with Mobile Robotics Platform

The system described in this thesis is designed to be completely self-containing such that integration with a robotics platform is not necessary. As discussed in section 7.2.3, such integration could be useful for reducing computational load through other forms of sensor fusion. The physical form factor of this sensor package can also be significantly reduced by using other depth sensors or by using different mounting configurations.

In general, this thesis has determined the feasibility of using COTS products in producing a high accuracy scene reconstruction device. While this work is sufficient for the task at hand, it is far from the end solution for all tasks. Refining this design for a specific application should be relatively trivial given the information provided.

References

[1] "CSB Finds Fatal 2009 NDK Explosion Resulted from Stress Corrosion Cracking of High-Pressure Vessel; Crystal Maker Failed to Inspect or Test Vessels and to Heed Earlier Warning about Risk to Public - General News - News | the U.S. Chemical Safety Board." [Online]. Available: http://www.csb.gov/csb-finds-fatal-2009-ndk-explosion-resulted-from-stress-corrosion-cracking-of-high-pressure-vessel-crystal-maker-failed-to-inspect-or-test-vessels-and-to-heed-earlier-warning-about-risk-to-public-/. [Accessed: 06-Feb-2014].

[2] "NDK Explosion Resulted from Stress Corrosion Cracking of High Pressure Vessel | 2013-11-14 | Process Heating." [Online]. Available: http://www.process-heating.com/articles/89773-ndk-explosion-resulted-from-stress-corrosion-cracking-of-high-pressure-vessel. [Accessed: 06-Feb-2014].

[3] "Boeing 737 Accident Reports." [Online]. Available: http://www.b737.org.uk/accident_reports.htm. [Accessed: 06-Feb-2014].

[4] "SBIR/STTR Interactive Topic Information System (SITIS)." [Online]. Available: http://webcache.googleusercontent.com/search?q=cache:V-r8_OBuqTkJ:www.dodsbir.net/sitis/display_topic.asp?Bookmark=45371+&cd=6&hl=en&ct=clnk&gl=us. [Accessed: 06-Feb-2014].

[5] "Super Caliper-Solar Powered- Series 500-No battery or origin reset needed for IP67 Digital Caliper." [Online]. Available: http://ecatalog.mitutoyo.com/Super-Caliper-Solar-Powered-Series-500-No-battery-or-origin-reset-needed-for-IP67-Digital-Caliper-C1587.aspx. [Accessed: 03-Mar-2014].

[6] "Coolant Proof Micrometer Series 293-with Dust/Water Protection Conforming to IP65 Level." [Online]. Available: http://ecatalog.mitutoyo.com/Coolant-Proof-Micrometer-Series-293-with-DustWater-Protection-Conforming-to-IP65-Level-C1089.aspx. [Accessed: 03-Mar-2014].

[7] "ABSOLUTE Digimatic Height Gage Series 570-with ABSOLUTE Linear Encoder." [Online]. Available: http://ecatalog.mitutoyo.com/ABSOLUTE-Digimatic-Height-Gage-Series-570-with-ABSOLUTE-Linear-Encoder-C1270.aspx. [Accessed: 03-Mar-2014].

[8] "Crysta-Apex S Series-191- Standard CNC CMM." [Online]. Available: http://ecatalog.mitutoyo.com/Crysta-Apex-S-Series-191-Standard-CNC-CMM-C1812.aspx. [Accessed: 03-Mar-2014].

[9] E. Metrology, "ROMER ABSOLUTE ARM SI product page." [Online]. Available: http://www.exactmetrology.com/products/romer/romer-absolute-si-arm/. [Accessed: 01-Sep-2013].

[10] Creaform, “Optical CMM 3D scanner for scanning and probing | MetraSCAN by Creaform.” [Online]. Available: http://www.creaform3d.com/en/metrology-solutions/optical-3d-scanner-metrascan. [Accessed: 19-Feb-2014].

[11] I. Systems, “INSPECTOR SYSTEMS, eddy current test robot, complete pipe inspection,” 2008. [Online]. Available: http://www.inspector-systems.com/Eddy_current_test_robot.html. [Accessed: 29-Oct-2013].

[12] *LIDAR—Light Detection and Ranging—is a remote sensing method used to examine the surface of the Earth*. NOAA.

[13] christian hulsmeyer, “Radio apparatus for determining distance and direction.,” 2,151,3231939.

[14] S. R. D. Sheet, “SR4000 Data Sheet.”

[15] SiCK, “LMS221 outdoor LiDAR specifications,” 2013. [Online]. Available: https://www.mysick.com/eCat.aspx?go=FinderSearch&Cat=Gus&At=Fa&Cult=English&FamilyID=344&Category=Produktfinder&Selections=34284,34258. [Accessed: 29-Oct-2013].

[16] Medina, Antonio and A. Medina, *Three Dimensional Camera and Rangefinder*. United States Patent 5081530.

[17] P. Hariharan, *Optical Interferometry*, 2nd ed. Academic Press, 2003, p. 351.

[18] T. U. of A. Veeco Instruments Inc. Optineering. Colllege of Optical Science, “Surface Profilers, Multiple Wavelength, and White Light Intereferometry,” in *Optical Shop Testing*, 3rd ed., M. Centro de Investigaciones en Optica, A. C., Loma del Bosque # 115 Col. Lomas del Campestre, P.O. Box 1-948, Postal Code 7 150, León, Gto., Ed. Wiley, 2006, pp. 674–764.

[19] Zemetrics, “Optical Profilers - how optical profilers work.” [Online]. Available: http://zemetrics.com/optical-profilers-about.shtml. [Accessed: 29-Oct-2013].

[20] Zygo, “Optical Profilers - How Optical Profilers Work.” [Online]. Available: http://www.zygo.com/?/met/profilers/opticalprofilersabout.htm. [Accessed: 19-Feb-2014].

[21] Wikipedia, “Interferometry,” *Wikipedia*. .

[22] C. Caraffi, S. Cattani, and P. Grisleri, “Off-Road Path and Obstacle Detection Using Decision Networks and Stereo Vision,” *IEEE Trans. Intell. Transp. Syst.*, vol. 8, 2007.

[23] R. Szeliski, *Computer vision: algorithms and applications*, 1st ed. Springer, 2011, p. 979.

[24] P. Favaro, “Depth from focus/defocus,” 2002. [Online]. Available: http://homepages.inf.ed.ac.uk/rbf/CVonline/LOCAL_COPIES/FAVARO1/dfdtutorial.html. [Accessed: 07-Nov-2013].

[25] S. K. Nayar and Y. Nakagawa, “Shape from focus,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 16, 1994.

[26] S. K. Nayar and Y. Nakagawa, “Shape from focus: an effective approach for rough surfaces,” *Proceedings., IEEE Int. Conf. Robot. Autom.*, 1990.

[27] M. Watanabe and S. K. Nayar, “Rational Filters for Passive Depth from Defocus,” *Int. J. Comput. Vis.*, vol. 27, pp. 203–225, 1998.

[28] A. Laurentini, “The Visual Hull Concept for Silhouette Based Image Understanding,” *IEEE PAMI*, vol. 16, pp. 150–162, 1994.

[29] D. Lanman and G. Taubin, “Build Your Own 3D Scanner : 3D Photography for Beginners,” *Siggraph*, 2009.

[30] P. Grey, “Point Grey Bumblebee® XB3.” [Online]. Available: http://ww2.ptgrey.com/stereo-vision/bumblebee-xb3. [Accessed: 25-Feb-2014].

[31] Velodyne, “HDL-64E S2 datasheet.” [Online]. Available: http://velodynelidar.com/lidar/products/brochure/HDL-64E S2 datasheet_2010_lowres.pdf. [Accessed: 25-Feb-2014].

[32] Zygo, “ZeGage datsheet.” [Online]. Available: http://www.zygo.com/met/profilers/zegage/zegage_specs.pdf. [Accessed: 25-Feb-2014].

[33] “KINECT DEPTH SENSOR EVALUATION FOR COMPUTER .” [Online]. Available: http://eng.au.dk/fileadmin/DJF/ENG/PDF-filer/Tekniske_rapporter/Technical_Report_ECE-TR-6-samlet.pdf.

[34] Mitutoyo, “Surfacemeasure 606.” [Online]. Available: http://www.mitutoyo.co.jp/eng/support/service/catalog/01/E16000.pdf. [Accessed: 25-Feb-2014].

[35] LMI, “RoLine 1130 datasheet.” [Online]. Available: http://downloads.lmi3d.com/sites/default/files/support/roline_1130.pdf. [Accessed: 10-Oct-2013].

[36] K. Konolige, J. Augenbraun, N. Donaldson, C. Fiebig, and P. Shah, “A low-cost laser distance sensor,” *2008 IEEE Int. Conf. Robot. Autom.*, pp. 3002–3008, May 2008.

[37] LMI, “RoLine 11X0 User’s Manual.” LMI, p. 53, 2010.

[38] T. Botterill, R. Green, and S. Mills, “DETECTING STRUCTURED LIGHT PATTERNS IN COLOUR IMAGES USING A SUPPORT VECTOR MACHINE Department of Computer Science , University of Canterbury , Christchurch , New Zealand Department of Computer Science , University of Otago , Dunedin , New Zealand.”

[39] “ROMER Absolute Arm with Integrated Scanner.” [Online]. Available: http://www.hexagonmetrology.us/products/portable-measuring-arms/romer-absolute-arm-with-integrated-scanner#technical-specifications. [Accessed: 25-Feb-2014].

[40] H. Metrology, “ROMER Absolute Arm,” *Product Brochure*, 2012. [Online]. Available: http://www.hexagonmetrology.us/images/pdf/AbsoluteArmBrochure2012vFinal-web2.pdf. [Accessed: 09-Oct-2013].

[41] K. P. Murphy, “Classification,” in *Machine Learning: A probabilistic Perspective (Adaptive Computation and Machine Learning series)*, The MIT Press, 2012, pp. 3 – 8.

[42] K. P. Murphy, “Naive Bayes classifiers,” in *Machine Learning: A Probabilistic Perspective (Adaptive Computation and Machine Learning series)*, The MIT press, 2012, pp. 82 – 87.

[43] K. P. Murphy, “Logistic regression,” in *Machine Learning: A probabilistic Perspective (Adaptive Computation and Machine Learning series)*, The MIT Press, 2012, pp. 245 – 271.

[44] K. P. Murphy, “Support vector machines (SVMs),” in *Machine Learning: A probabilistic Perspective (Adaptive Computation and Machine Learning series)*, The MIT Press, 2012, pp. 496 – 505.

[45] C. Guestrin, “Support Vector Machines,” 2005. [Online]. Available: http://www.cs.cmu.edu/~guestrin/Class/10701-S06/Slides/svms-s06.pdf. [Accessed: 26-Feb-2014].

[46] StatSoft, “Support Vector Machines (SVM).” [Online]. Available: http://www.statsoft.com/textbook/support-vector-machines. [Accessed: 21-Nov-2013].

[47] K. P. Murphy, “A simple non-parametric classifier: K-nearest neighbors,” in *Machine Learning: A probabilistic Perspective (Adaptive Computation and Machine Learning series)*, The MIT Press, 2012, pp. 16 – 19.

[48] S. M. LaValle, “Geometric Representations and Transformations,” in *Planning Algorithms*, Cambridge University Press, 2006, p. 81.

[49] Steven Michael LaValle, “Translation,” in *Planning Algorithms*, Cambridge University Press, 2006, pp. 77 – 78.

[50] Steven Michael LaValle, “3D translation,” in *Planning Algorithms*, Cambridge University Press, 2006, pp. 80 – 81.

[51] Steven Michael LaValle, “Rotation,” in *Planning Algorithms*, Cambridge University Press, 2006, pp. 78 – 79.

[52] Steven Michael LaValle, “Yaw, pitch, and roll rotations,” in *2Planning Algorithms*, Cambridge University Press, 2006, pp. 80 – 81.

[53] Steven Michael LaValle, “Combining translation and rotation,” in *Planning Algorithms*, Cambridge University Press, 2006, p. 79.

[54] Steven Michael LaValle, “The homogeneous transformation matrix for 3D bodies,” in *Planning Algorithms*, Cambridge University Press, 2006, p. 82.

[55] “Transformations in Homogeneous Coordinates.” [Online]. Available: http://www.cs.iastate.edu/~cs577/handouts/homogeneous-transform.pdf.

[56] “Camera Calibration and 3D Reconstruction — OpenCV 2.4.6.0 documentation.” [Online]. Available: http://docs.opencv.org/modules/calib3d/doc/camera_calibration_and_3d_reconstruction.html.

[57] O. Hall-Holt and S. Rusinkiewicz, “Stripe boundary codes for real-time structured-light range scanning of moving objects,” *Proc. Eighth IEEE Int. Conf. Comput. Vis. ICCV 2001*, vol. 2, no. C, pp. 359–366, 2001.

[58] J. Salvi, “Pattern codification strategies in structured light systems,” *Pattern Recognit.*, vol. 37, pp. 827–849, 2004.

[59] J. Salvi, J. Pagès, and J. Batlle, “Pattern codification strategies in structured light systems,” *Pattern Recognit.*, vol. 37, no. 4, pp. 827–849, Apr. 2004.

[60] “Temporal Dithering of Illumination for Fast Active Vision.” [Online]. Available: http://staff.aist.go.jp/shun-yamazaki/research/dlp/.

[61] D. Hoiem, “How the Kinect Works Kinect Device,” *Computational Photography*, 2010. [Online]. Available: http://courses.engr.illinois.edu/cs498dh/fa2011/lectures/Lecture 25 - How the Kinect Works - CP Fall 2011.pdf.

[62] J. Maccormick, “How does the Kinect work?”

[63] Y. A. Barak Freedman, Alexander Shpunt, “Distance-Varying Illumination and Imaging Techniques for Depth Mapping,” US20100290698 A118-Nov-2010.

[64] K. Khoshelham, “ACCURACY ANALYSIS OF KINECT DEPTH DATA,” 2010.

[65] B. Bockstege, A. Huus, and N. Dame, “Measuring the Performance of the Microsoft Kinect,” *netscalecsendedu*, pp. 1–3, 2010.

[66] E. B. Thro, “Leonardo da Vinci’s solution to the problem of the pinhole camera,” *Archive for History of Exact Sciences*, vol. 48. pp. 343–371, 1994.

[67] G. B. S. D. A. Murguet, “MonoSLAM,” 2012. [Online]. Available: http://www.ensta-paristech.fr/~filliat/Courses/2011_projets_C10-2/BRUNEAU_DUBRAY_MURGUET/monoSLAM_bruneau_dubray_murguet_en.html. [Accessed: 19-Feb-2014].

[68] “File:Pinhole-camera.svg - Wikipedia, the free encyclopedia.” [Online]. Available: http://en.wikipedia.org/wiki/File:Pinhole-camera.svg. [Accessed: 03-Mar-2014].

[69] D. Lanman and G. Taubin, “Build your own 3D scanner: optical triangulation for beginners,” *SIGGRAPH ASIA 09 ACM SIGGRAPH ASIA 2009 Courses*, pp. 1–94, 2009.

[70] J. Santolaria, J. J. Pastor, F. J. Brosed, and J. J. Aguilar, “A one-step intrinsic and extrinsic calibration method for laser line scanner operation in coordinate measuring machines,” *Meas. Sci. Technol.*, vol. 20, no. 4, p. 045107, Apr. 2009.

[71] F. Z. F. Zheng and B. K. Bin Kong, “Calibration of linear structured light system by planar checkerboard,” *Int. Conf. Inf. Acquis. 2004 Proc.*, 2004.

[72] D. K. D. Kim, S. L. S. Lee, H. K. H. Kim, and S. L. S. Lee, *Wide-angle laser structured light system calibration with a planar object*. 2010.

[73] “HandEyeCalibration.” [Online]. Available: http://campar.in.tum.de/Chair/HandEyeCalibration.

[74] G. Puskorius and L. Feldkamp, “Global calibration of a robot/vision system,” *IEEE Conf. Robot. Autom.*, vol. 1, pp. 190–195, 1987.

[75] C. Chen, “Robotic Eye-in-Hand Calibration by Calibrating Optical Axis and Target Pattern,” pp. 155–173, 1995.

[76] Y. Shiu and S. Ahmad, “Finding the mounting position of a sensor by solving a homogeneous transform equation of the form AX = XB,” *Proc. 1987 IEEE Int. Conf. Robot. Autom.*, vol. 4, pp. 1666–1671, 1987.

[77] R. Y. Tsai and R. K. Lenz, “Real time versatile robotics hand/eye calibration using 3D machine vision,” in *Proceedings. 1988 IEEE International Conference on Robotics and Automation*, 1988, pp. 554–561.

[78] G. Klein and D. Murray, “Parallel Tracking and Mapping for Small AR Workspaces,” *2007 6th IEEE ACM Int. Symp. Mix. Augment. Real.*, 2007.

[79] A. V Segal, D. Haehnel, and S. Thrun, “Generalized-ICP.”

[80] M. D. Radu Bogdan Rusu, “pcl::IterativeClosestPointNonLinear< PointSource, PointTarget > Class Template Reference,” *PCL Documentation*, 2011. [Online]. Available: http://docs.pointclouds.org/1.0.1/classpcl_1_1_iterative_closest_point_non_linear.html. [Accessed: 09-Oct-2013].

[81] B. Curless and M. Levoy, “A volumetric method for building complex models from range images,” *Proc. 23rd Annu. Conf. Comput. Graph. Interact. Tech. - SIGGRAPH ’96*, pp. 303–312, 1996.

[82] S. Izadi, D. Kim, O. Hilliges, D. Molyneaux, R. Newcombe, P. Kohli, J. Shotton, S. Hodges, D. Freeman, A. Davison, and A. Fitzgibbon, “KinectFusion : Real-time 3D Reconstruction and Interaction Using a Moving Depth Camera *.”

[83] R. B. Rusu, N. Blodow, and M. Beetz, “Fast Point Feature Histograms (FPFH) for 3D registration,” *2009 IEEE Int. Conf. Robot. Autom.*, pp. 3212–3217, May 2009.

[84] A. G. Buch, D. Kraft, J. Kamarainen, H. G. Petersen, and N. Kr, “Pose Estimation using Local Structure-Specific Shape and Appearance Context.”

[85] Y. Huang, X. Li, and P. Chen, “Calibration method for line-structured light multi-vision sensor based on combined target,” *EURASIP J. Wirel. Commun. Netw.*, vol. 2013, no. 1, p. 92, 2013.

[86] J. Davis, “Fusion of time-of-flight depth and stereo for high accuracy depth maps,” *2008 IEEE Conf. Comput. Vis. Pattern Recognit.*, pp. 1–8, Jun. 2008.

[87] K. Kuhnert and M. Stommel, “Fusion of Stereo-Camera and PMD-Camera Data for Real-Time Suited Precise 3D Environment Reconstruction,” *2006 IEEE/RSJ Int. Conf. Intell. Robot. Syst.*, pp. 4780–4785, Oct. 2006.

[88] Y. Yemez and C. J. Wetherilt, “A volumetric fusion technique for surface reconstruction from silhouettes and range data,” *Comput. Vis. Image Underst.*, vol. 105, no. 1, pp. 30–41, Jan. 2007.

[89] M. Vergauwen and L. Gool, “Web-based 3D Reconstruction Service,” *Machine Vision and Applications*, vol. 17. pp. 411–426, 2006.

[90] P. Elinas and J. J. Little, “Stereo vision SLAM : Near real-time learning of 3D point-landmark and 2D occupancy-grid maps using particle filters .,” *Comput. Intell.*, 2007.

[91] R. A. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. J. Davison, P. Kohli, J. Shotton, S. Hodges, and A. Fitzgibbon, “KinectFusion: Real-time dense surface mapping and tracking,” *2011 10th IEEE Int. Symp. Mix. Augment. Real.*, vol. 7, pp. 127–136, 2011.

[92] F. Endres, J. Hess, D. Cremers, and N. Engelhard, “An Evaluation of the RGB-D SLAM System,” *Perception*, vol. 3, pp. 1691–1696, 2012.

[93] Q.-Y. Zhou and V. Koltun, “Dense scene reconstruction with points of interest,” *ACM Trans. Graph.*, vol. 32, p. 1, 2013.

[94] Y. M. Kim, C. Theobalt, J. Diebel, J. Kosecka, B. Miscusik, and S. Thrun, “Multi-view image and ToF sensor fusion for dense 3D reconstruction,” *2009 IEEE 12th Int. Conf. Comput. Vis. Work. ICCV Work.*, pp. 1542–1549, 2009.

[95] L. Iocchi and S. Pellegrini, “Building 3d Maps With Semantic Elements Integrating 2d Laser, Stereo Vision And IMU On A Mobile Robot,” in *Proc. 2nd ISPRS Intl. Workshop 3D-ARCH*, 2007.

[96] “Laser Line Generator Lenses | Edmund Optics.” [Online]. Available: http://www.edmundoptics.com/optics/optical-lenses/cylinder-lenses/laser-line-generator-lenses/1910.

[97] Z. Z. Z. Zhang, “Flexible camera calibration by viewing a plane from unknown orientations,” *Proc. Seventh IEEE Int. Conf. Comput. Vis.*, vol. 1, pp. 666–673 vol.1, 1999.

[98] R. Y. Tsai and R. K. Lenz, “A new technique for fully autonomous and efficient 3D robotics hand/eye calibration,” *IEEE Trans. Robot. Autom.*, vol. 5, pp. 345–358, 1989.

[99] “Using Kinfu Large Scale to generate a textured mesh,” *PCL docu*, 2013. [Online]. Available: http://pointclouds.org/documentation/tutorials/using_kinfu_large_scale.php. [Accessed: 06-Feb-2014].

[100] Bouguet, “Calibrating a stereo system, stereo image rectification and 3D stereo triangulation,” 2010. [Online]. Available: http://www.vision.caltech.edu/bouguetj/calib_doc/htmls/example5.html. [Accessed: 24-Sep-2013].

[101] “Nondestructive evaluation of superalloy specimens with a thermal barrier coating .” [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.157.1223&rep=rep1&type=pdf.

[102] “Boiler Generating Bank Inspection – Pitting Detected – using RFET :: TesTex-NDT.” [Online]. Available: http://testex-ndt.com/from-the-field/boiler-generating-bank-inspection-pitting-detected-using-rfet/.

[103] GE, “Crack Management.” [Online]. Available: 1. [Accessed: 04-Mar-2014].

[104] D. Girardeau-Montaut, “Distances Computation - CloudCompareWiki,” *Cloud Compare Documentation*, 2012. [Online]. Available:

http://www.danielgm.net/cc/doc/wiki/index.php5?title=Distances_Computation. [Accessed: 26-Feb-2014].

[105] Wikipedia, “Distance from a point to a line.” [Online]. Available: http://en.wikipedia.org/wiki/Distance_from_a_point_to_a_line. [Accessed: 04-Mar-2014].

[106] “GML C++ Camera Calibration Toolbox | Graphics and Media Lab.” [Online]. Available: http://graphics.cs.msu.ru/en/node/909.