

TITLE OF THESIS

Christopher James Mobley

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

Masters of Science
in
Mechanical Engineering

Tomonari Furukawa
Brian Lattimer
Kevin Kochersberger

Blacksburg, Virginia

Keywords: Autonomous Navigation, SLAM, Visual Servoing, Mobile Manipulation,
Computer Vision, State Machine
Copyright 2016, Christopher James Mobley

TITLE OF THESIS

Christopher James Mobley

(ABSTRACT)

This paper will show the general framework necessary in order to solve two main problems preventing the development and widespread adoption of automation and robotics in construction (ARC). These problems include the fact that typical construction sites tend to be unstructured and are continuously evolving versus the highly controlled environments found in manufacturing. Also, the relationship between the part and manipulator has been reversed, causing increased complexity not seen in manufacturing environments where the part arrives at a fixed manipulator. The techniques that will be presented allow systems to create a 2-D map of their environment, localize themselves and complete the task(s) assigned. After localizing an augmented reality (AR) tag at the work site, the system is able to use a priori knowledge to localize points of interest (POIs) and complete several different types of operations achieving an accuracy of approximately ± 2 mm based on a multifaceted computer vision approach with only a USB webcam.

Acknowledgments

Insert Acknowledgement

Contents

1	Introduction	1
2	Background	3
2.1	ROS Concepts	3
2.1.1	ROS Communication	3
2.1.2	URDF and TF	4
2.1.3	SLAM	4
2.1.4	Navigation Stack	6
2.1.5	Moveit!	8
2.1.6	SMACH	10
2.2	Camera Concepts	10
2.3	Computer Vision Concepts	12
2.3.1	Color Spaces	12
2.3.2	Linear and Non-Linear Filters	13
2.3.3	Binary Operations	14
2.3.4	Hough Circle	15
2.3.5	Good Feature To Track	16
2.3.6	Optical Flow	16
3	Related Work	18
4	System Overview	19

5	System Implementation	21
5.1	Global Map Creation and Task Location Specification	21
5.2	Autonomous Localization and Navigation	22
5.3	Task Association and A Priori Knowledge	22
5.4	Generic Framework for Multi-Stage Computer Vision Algorithm	23
5.4.1	Sensor Calibration	23
5.4.2	Initial Feature Location Prediction	24
5.4.3	Corrected Feature Locations	25
5.5	Specific Task Implementation	25
5.5.1	Drilling Framework	25
5.5.2	Sealant Application Framework	26
6	Experiments and Results	28
6.1	Hardware Architecture	28
6.2	Software Architecture	29
6.3	Experiments	29
6.3.1	Navigation System Experimental Setup	29
6.3.2	Drilling Operation Experimental Setup	30
6.3.3	Sealant Application Experimental Setup	31
6.4	Results	31
6.4.1	Navigation System Accuracy Achieved	31
6.4.2	Discussion of Drilling Operation Accuracy	32
6.4.3	Discussion of Sealant Application Accuracy	32
7	Conclusion and Future Work	33
	Bibliography	34

List of Figures

1.1	Vision for future transformable production systems.	2
2.1	Simple ROS Node Flowchart	3
2.2	Robot Model with TFs	4
2.3	Overview of SLAM Process. [Riisgaard et al. 2005]	5
2.4	Example of 2-D Occupancy Grid Map Produced.	5
2.5	Real World Performance Analysis of ROS Available SLAM Algorithms. [Santos et al. 2013]	6
2.6	ROS Navigation Stack setup.	7
2.7	Adaptive Monte Carlo Localization Flowchart.	7
2.8	Trajectory Rollout Path Planning Framework	8
2.9	Moveit!’s System Architecture.	8
2.10	Graphical View of State Machine using SMACH	10
2.11	Digital Image Formation.	11
2.12	Ground Sample Distance Effects on Image Quality.	11
2.13	Shutter Type Effects on Image Quality.	12
2.14	RGB and HSV color space models	13
2.15	Effects of common linear filters.	13
2.16	Effect of histogram equalization.	14
2.17	Effects of common binary operations.	14
2.18	Summary of preprocessing operation performed by Hough Circle detector. . .	15
2.19	Summary of Hough Circle detector with known radius.	15

2.20	Summary of Hough Circle detector with unknown radius.	16
2.21	Example of optical flow.	17
2.22	Summary of Kanade-Lucas-Tomasi feature tracker.	17
4.1	General system overview.	20
5.1	2-D map environment with specified start locations.	22
5.2	Task association and localization framework.	23
5.3	Sensor Calibration Setup and Results.	24
5.4	Initial Feature Localization Framework.	24
5.5	Feature Location General Correction Framework.	25
5.6	Drilling operation framework.	26
5.7	Sealant Application Framework.	27
6.1	Clearpath Robotics' Husky and Fetch Robotics' Fetch Mobile Manipulator .	28
6.2	Software Architecture Diagram	29
6.3	Navigation System Experimental Setup.	30
6.4	Drilling Operations Experimental Setup.	30
6.5	Sealant Application Experimental Setup.	31

List of Tables

2.1	Real World Error Estimation for ROS Available SLAM Algorithm. [Santos et al. 2013]	6
2.2	Comparison Between ROS Available Moveit! Inverse Kinematic Plugins. [Beeson et al. 2015]	9
6.1	Global Localization Accuracy Comparison Given Distance From Global Location.	31
6.2	Drilling Operation Accuracy Comparison Given Distance From Work Surface.	32
6.3	Sealant Application Accuracy Comparison Given Distance From Work Surface.	32

Chapter 1

Introduction

Unlike the substantial benefits seen in the manufacturing industry through automation and robotics, automation and robotics in construction (ARC) has lagged far behind in adoption [1]. Consequently, when compared with other industries, construction has seen a significant decrease in productivity, as well as an increase in workplace injuries/fatalities over the last several decades [2]. While several technical complexities inherent in construction have hindered the development and adoption of field construction robots [4], through the capitalization of advances made by other industries, ARC can quickly close this gap. Thereby allowing dangerous and or mundane repetitive tasks to be accomplished autonomously. Thus, causing an increase in productivity and a decrease in workplace injuries/fatalities [1]. However, ARC faces two unique challenges when compared to other industries. Unlike manufacturing environments, which are tightly controlled, typical construction sites tend to lack structure and are continuously evolving. In addition, the reversal in relationship between the part and manipulator has dramatically increased the complexity of the problem to be solved [3]. Instead of the part appearing at a fixed manipulator, the manipulator must now move to and localize itself with respect to the part. The remainder of this paper is structured as follows: In Section 2, the author's technical approach is outlined, with particular focus on problem two, and experimental results are shown. Conclusions are then drawn and future work discussed in Section 3.

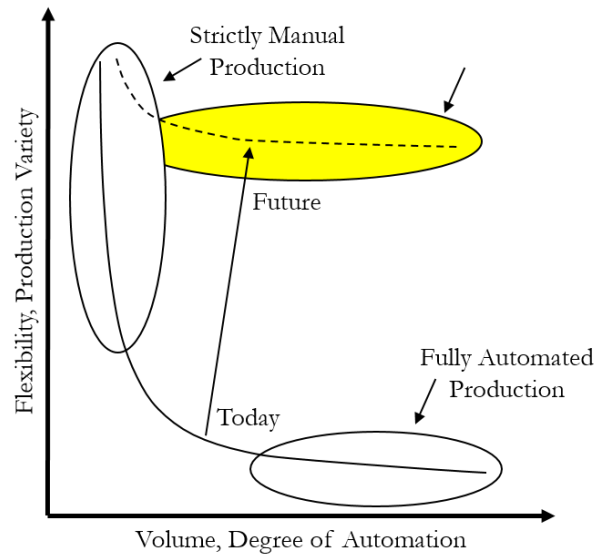


Figure 1.1: Vision for future transformable production systems.

Chapter 2

Background

Insert text outlining upcoming sections.

2.1 ROS Concepts

Insert Text Defining ROS.

2.1.1 ROS Communication

Insert Text Regarding Node/Topic/Service Communication.

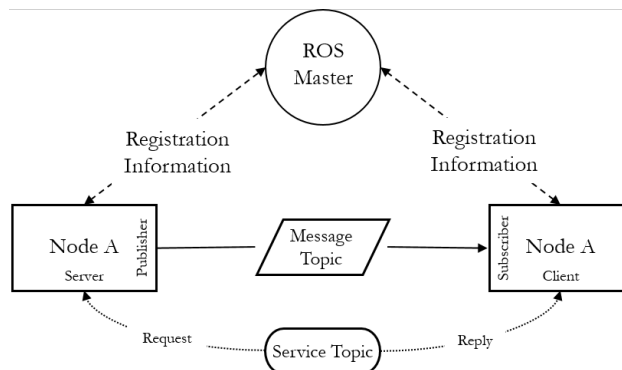


Figure 2.1: Simple ROS Node Flowchart

2.1.2 URDF and TF

Insert Regarding How To Setup Up TFs on a Robot and How TF Communication Works.



Figure 2.2: Robot Model with TFs

2.1.3 SLAM

Insert Text Outlining the Basic Intuition Behind SLAM.



Figure 2.3: Overview of SLAM Process. [Riisgaard et al. 2005]

Insert Text Explaining How ROS Uses SLAM to Create 2-D Occupancy Grid Maps and What How They Are Used.



Figure 2.4: Example of 2-D Occupancy Grid Map Produced.

Insert Text Specifying Available 2-D SLAM Techniques and Why KartoSLAM was Chosen.



Figure 2.5: Real World Performance Analysis of ROS Available SLAM Algorithms. [Santos et al. 2013]

Table 2.1: Real World Error Estimation for ROS Available SLAM Algorithm. [Santos et al. 2013]

Real World Experiments				
HectorSLAM	GMapping	KartoSLAM	CoreSLAM	LagoSLAM
1.1972	2.1716	1.0318	14.75333	3.0264
0.5094	0.6945	0.3742	7.9463	0.8181
1.0656	1.6354	0.9080	7.5824	2.5236

2.1.4 Navigation Stack

Insert Text Outlining The Navigation Stack’s Framework.



Figure 2.6: ROS Navigation Stack setup.

Insert Text Explaining How Adaptive Monte Carlo Localization Works.



Figure 2.7: Adaptive Monte Carlo Localization Flowchart.

Insert Text Explaining How Trajectory Rollout Works for Path Planning.



Figure 2.8: Trajectory Rollout Path Planning Framework

2.1.5 Moveit!

Insert Text Outlining Moveit!'s System Architecture.



Figure 2.9: Moveit!'s System Architecture.

Insert Text Explaining the Basic Intuition Behind Forward and Inverse Kinematics.

Insert Text Explaining the Difference Between KDL, KDL-RR and Trac_IK and Why Trac_IK was chosen to be used.

Table 2.2: Comparison Between ROS Available Moveit! Inverse Kinematic Plugins. [Beeson et al. 2015]

Chain	DOFs	Orocos' KDL solve rate	Orocos' KDL Avg Time	KDL-RR solve rate	KDL-RR Avg Time	TRAC-IK solve rate	TRAC-IK Avg Time
Atlas 2013 arm	6	75.54%	1.35ms	97.13%	0.39ms	99.97%	0.33ms
Atlas 2015 arm	7	75.71%	1.50ms	93.13%	0.81ms	99.18%	0.48ms
Baxter arm	7	61.07%	2.21ms	89.52%	1.02ms	99.17%	0.60ms
Denso VS-068	6	27.92%	3.69ms	98.13%	0.42ms	99.78%	0.38ms
Fanuc	5	21.07%	3.99ms	88.34%	0.92ms	99.16%	0.58ms
M-430iA/2F							
Fetch arm	7	92.49%	0.73ms	93.82%	0.72ms	99.96%	0.44ms
Jaco2	6	26.23%	3.79ms	97.66%	0.58ms	99.51%	0.58ms
KUKA LBR iiwa	7	37.71%	3.37ms	94.02%	0.73ms	99.63%	0.56ms
14 R820							
KUKA LWR 4+	7	67.80%	1.88ms	95.40%	0.62ms	99.95%	0.38ms
PR2 arm	7	83.14%	1.37ms	86.96%	1.27ms	99.84%	0.59ms
NASA	7	61.27%	2.29ms	87.57%	1.10ms	99.31%	0.67ms
Robonaut2							
‘grasping leg’							
NASA	15	97.99%	0.80ms	98.00%	0.84ms	99.86%	0.79ms
Robonaut2 ‘leg’							
+ waist + arm							
NASA	7	86.28%	1.02ms	94.26%	0.73ms	99.25%	0.50ms
Robonaut2 arm							
NASA	7	61.74%	2.44ms	99.87%	0.36ms	99.93%	0.44ms
Robosimian arm							
TRAC Labs	7	79.11%	1.35ms	95.12%	0.63ms	99.80%	0.53ms
modular arm							
UR10	6	36.16%	3.29ms	88.05%	0.82ms	99.47%	0.49ms
UR5	6	35.88%	3.30ms	88.69%	0.78ms	99.55%	0.42ms
NASA Valkyrie	7	45.18%	3.01ms	90.05%	1.29ms	99.63%	0.61ms
arm							

2.1.6 SMACH

Insert Text Explaining the Basics of State Machine and How They are Implemented in SMACH.



Figure 2.10: Graphical View of State Machine using SMACH

2.2 Camera Concepts

Insert Text Explaining How Digital Image Are Recorded and Stored.



Figure 2.11: Digital Image Formation.

Insert Text Explaining Ground Sample Distance and It's Effects on Image Quality Thereby Effecting Ones Camera Choice.



Figure 2.12: Ground Sample Distance Effects on Image Quality.

Insert Text Explaining The Effects of Different Shutter Type on Image Quality Thereby Effecting Ones Camera Choice.



Figure 2.13: Shutter Type Effects on Image Quality.

2.3 Computer Vision Concepts

Insert Text Regarding Upcoming Sections.

2.3.1 Color Spaces

Insert Text Explaining Both RGB Color Space and HSV/HSL Color Space and Why One Would Be Chosen Over The Other.

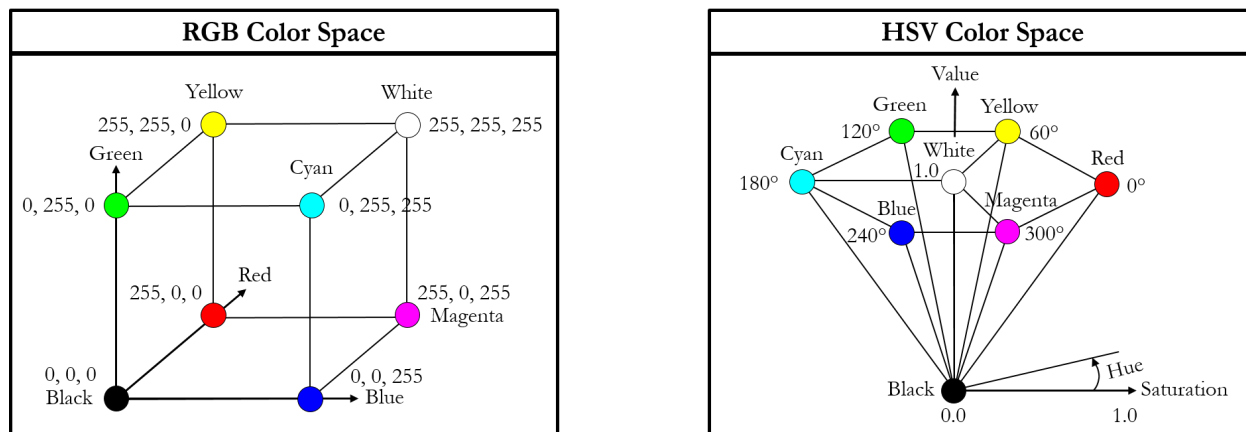


Figure 2.14: RGB and HSV color space models

2.3.2 Linear and Non-Linear Filters

Insert Text Explaining the Basics behind Linear Filters, Which Ones Were Chosen and Why.



Figure 2.15: Effects of common linear filters.

Insert Text Explaining the Basics behind Non-Linear Filters, Which Ones Were Chosen and Why.



Figure 2.16: Effect of histogram equalization.

2.3.3 Binary Operations

Insert Text Explaining the Basics behind Binary Operations, Which Ones Were Chosen and Why.

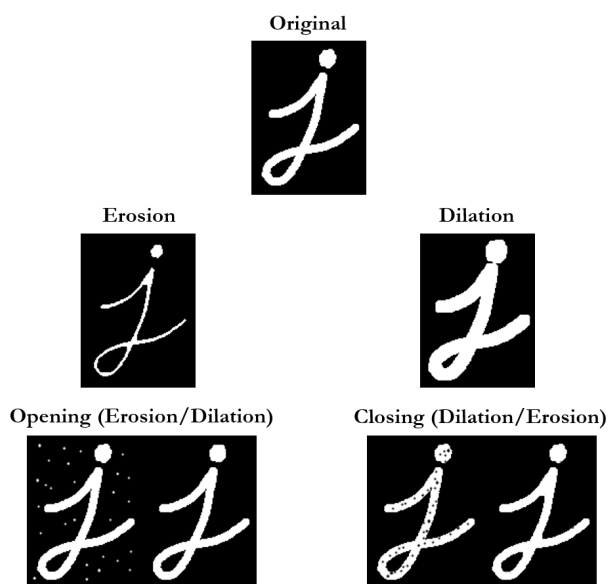


Figure 2.17: Effects of common binary operations.

2.3.4 Hough Circle

Insert Text Explaining The Basic Preprocessing Operation Performed by Hough Circle Transform In Order to Get Edges.



Figure 2.18: Summary of preprocessing operation performed by Hough Circle detector.

Insert Text Explaining the Basic Intuition Behind Hough Circle with Known Radius.

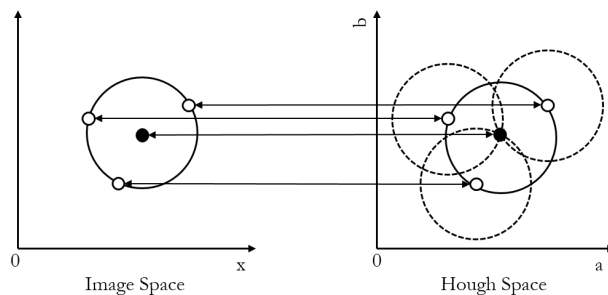


Figure 2.19: Summary of Hough Circle detector with known radius.

Insert Text Expanding Basic Intuition Behind Hough Circle with Unknown Radius.

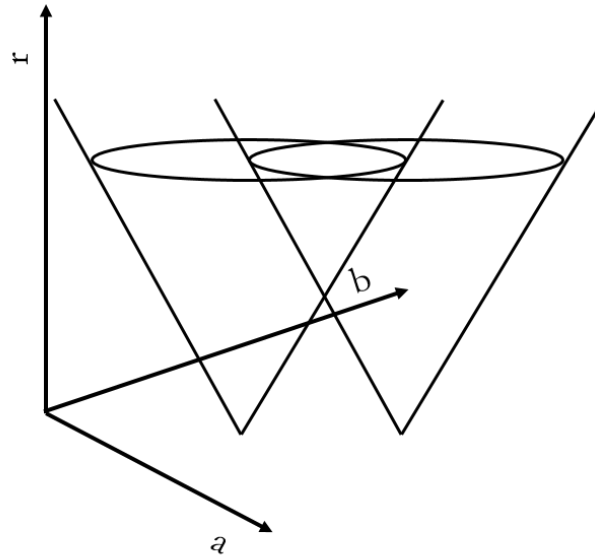


Figure 2.20: Summary of Hough Circle detector with unknown radius.

2.3.5 Good Feature To Track

Insert Text Explaining the Basic Intuition Behind Good Feature.

2.3.6 Optical Flow

Insert Text Explaining the Basic Intuition Behind Optical Flow.



Figure 2.21: Example of optical flow.

Insert Text Explaining the Basic Intuition Behind Kanade-Lucas-Tomasi Feature Tracker.



Figure 2.22: Summary of Kanade-Lucas-Tomasi feature tracker.

Chapter 3

Related Work

This section will discuss related work.

Chapter 4

System Overview

The generic framework of the system is depicted in Figure 4.1. The system allows the user to input predefined tasks. Given a priori knowledge of each task and their global start locations, the mobile system navigates to and performs the requested operation(s) while monitoring its battery level to ensure mission completion. The framework shown was implemented using a hierarchical state machine and was written in such a way as to make it compatible with the Robot Operating System (ROS) to ensure ease of use across differing robotic platforms.



Figure 4.1: General system overview.

Chapter 5

System Implementation

Insert Text Outlining the Upcoming Sections.

5.1 Global Map Creation and Task Location Specification

The purpose of this section is reiterate why KartoSLAM was chosen, as well as describe how maps are created, the map files themselves, and how global locations are specified.

Before the manipulator can localize itself with respect to the part, the system must first navigate to the general vicinity in which the work will take place. In order to achieve this, the system utilizes odometry data, given by wheel encoders and an onboard inertial measurement unit (IMU), as well as sensor data, such as laser scans from a LIDAR or point clouds from an RGB-D sensor, to output safe velocity commands that will be sent to the mobile base of the system.

First, a Simultaneous Localization and Mapping (SLAM) technique named KartoSLAM uses the system's odometry and laser scan or point cloud data, to create a 2-D map of the environment in which the operation(s) will take place. After which, the global location of specific operation(s) are defined, as shown in Figure 5.1.



Figure 5.1: 2-D map environment with specified start locations.

5.2 Autonomous Localization and Navigation

Adaptive Monte Carlo localization (amcl) is used to localize the system within the map. Subsequently, odometry data is combined with a global and local cost map, in which obstacles and a specific distance around them represent a cost. These maps are used to plan optimal and obstacle free paths through the environment. The global path is computed before the system begins moving and takes into account all known obstacles, while the local path monitors incoming sensor data to compute suitable linear and angular velocities for the system to complete the current section of the global path. The local path is typically computed at a rate of 20 Hz; however, this parameter is adjustable given the needs of the system.

5.3 Task Association and A Priori Knowledge

Once arriving in the general vicinity of the task to be accomplished, the system then locates an augmented reality (AR) tag [4], which allows it to localize and transform points of interest (POIs) associated with the AR tag into the system's frame of reference. This general localization framework is presented in Figure 5.2.



Figure 5.2: Task association and localization framework.

5.4 Generic Framework for Multi-Stage Computer Vision Algorithm

Insert Text Outlining the Upcoming Sections.

5.4.1 Sensor Calibration

This subsection will go through the method used for camera calibration (citing paper), as well as specify results in terms of error. In addition, the math behind the laser calibration will be presented.



Figure 5.3: Sensor Calibration Setup and Results.

5.4.2 Initial Feature Location Prediction

This subsection will go through AR tags and how they are used to populate point.



Figure 5.4: Initial Feature Localization Framework.

5.4.3 Corrected Feature Locations

Currently the sponsor of this work uses a hole template on the object to be drilled to ensure accuracy within ± 0.3 mm. Using the predicted hole locations, given by the AR tag, an inverse kinematic solver is used to move the manipulator to the specified Cartesian location. A camera mounted on the manipulator is then used to further correct the position of the end-effector. Canny Edge Detection, Hough Transforms, as well as the camera's intrinsic characteristics and a priori knowledge of each hole's size is used to output an adjusted Cartesian location of the circle on the templet closest to the predicted position [5]. If a hole to be drilled is not found or is outside the range of the manipulator, that hole will be added to a list and the customer notified of all such holes after the operation is completed. In addition to the high precision achieved by the above technique, it can also provide a video log of all work done for inspection.



Figure 5.5: Feature Location General Correction Framework.

5.5 Specific Task Implementation

Outline the upcoming sections.

5.5.1 Drilling Framework

This section will go in depth into how the robot simulates drilling.

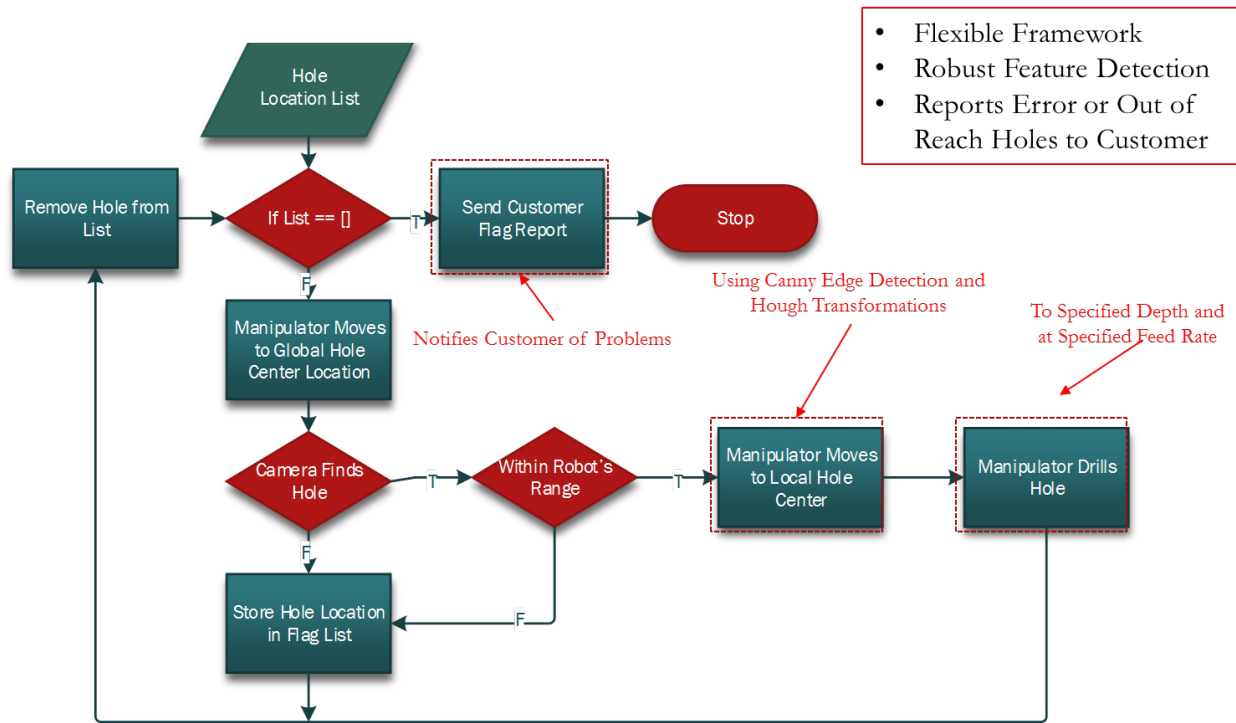


Figure 5.6: Drilling operation framework.

5.5.2 Sealant Application Framework

This section will go in depth into how the robot simulates sealant application.

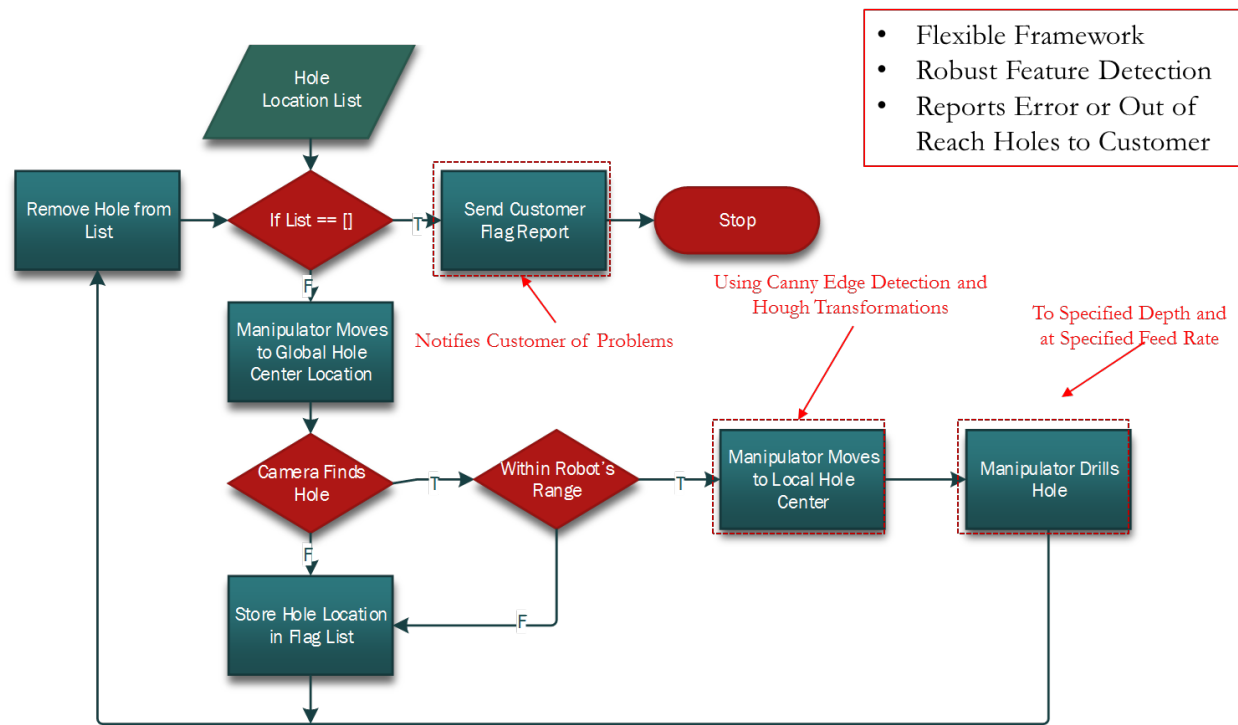


Figure 5.7: Sealant Application Framework.

Chapter 6

Experiments and Results

Outline the upcoming sections.

6.1 Hardware Architecture

Insert Text Regarding Robot, Sensors, etc.



Figure 6.1: Clearpath Robotics' Husky and Fetch Robotics' Fetch Mobile Manipulator

6.2 Software Architecture

Insert Text Regarding The Software Architecture.



Figure 6.2: Software Architecture Diagram

6.3 Experiments

Insert Text Regarding Upcoming Sections.

6.3.1 Navigation System Experimental Setup

Insert Text Regarding the Navigation System Experimental Setup.



Figure 6.3: Navigation System Experimental Setup.

6.3.2 Drilling Operation Experimental Setup

Insert Text Regarding the Drilling Operations Experimental Setup.



Figure 6.4: Drilling Operations Experimental Setup.

6.3.3 Sealant Application Experimental Setup

Insert Text Regarding the Sealant Application Experimental Setup.



Figure 6.5: Sealant Application Experimental Setup.

6.4 Results

6.4.1 Navigation System Accuracy Achieved

Insert Text Regarding Accuracy of Navigation System and Ways To Improve Accuracy.

Table 6.1: Global Localization Accuracy Comparison Given Distance From Global Location.

Distance from Global Position		x (mm)	y (mm)	z (mm)
Global Position Localization	5	0	0	0
	10	0	0	0
	20	0	0	0
	40	0	0	0

6.4.2 Discussion of Drilling Operation Accuracy

Insert Text Regarding Accuracy of Drilling Operations and Ways To Improve Accuracy.

Table 6.2: Drilling Operation Accuracy Comparison Given Distance From Work Surface.

	Distance from Work Surface (m)	x (mm)	y (mm)	z (mm)
AR Tag Localization	5	0	0	0
	10	0	0	0
	20	0	0	0
	40	0	0	0
Feature Localization	5	0	0	0
	10	0	0	0
	20	0	0	0
	40	0	0	0

6.4.3 Discussion of Sealant Application Accuracy

Insert Text Regarding Accuracy of Sealant Application and Ways To Improve Accuracy.

Table 6.3: Sealant Application Accuracy Comparison Given Distance From Work Surface.

	Distance from Work Surface (m)	x (mm)	y (mm)	z (mm)
AR Tag Localization	5	0	0	0
	10	0	0	0
	20	0	0	0
	40	0	0	0
Feature Localization	5	0	0	0
	10	0	0	0
	20	0	0	0
	40	0	0	0

Chapter 7

Conclusion and Future Work

This paper showed the general framework necessary in order to solve the two main problems preventing the development and widespread adoption of ARC. These problems have caused a decrease in productivity and increase in workplace injuries/fatalities over the past several decades in comparison to other industries [2]. These problems include the fact that typical construction sites tend to be unstructured and are continuously evolving versus the highly controlled environments found in manufacturing. Also, the relationship between the part and manipulator has been reversed, causing increased complexity not seen in manufacturing environments where the part arrives at a fixed manipulator [3]. The techniques presented allow systems to create a 2-D map of their environment, localize themselves and complete the task(s) assigned. After localizing an AR tag at the work site, the system is able to use a priori knowledge to localize POIs and complete a plethora of operations achieving an accuracy of approximately ± 2 mm based on a multifaceted computer vision approach with only a USB webcam.

Future work to be explored includes increasing the accuracy of the computer vision system to the sub-millimeter levels through the use of a machine vision camera, as well as a relatively new calibration technique developed by Feng et al [3]. In addition, automated approaches to create 3-D maps are being looked into in order to provide updated data about the robot's surroundings and task(s) automatically without human intervention. Also, human and robot collaboration over a distributed network is being explored.

Bibliography

- [1] C Balaguer. “NOWADAYS TRENDS IN ROBOTICS AND AUTOMATION IN CONSTRUCTION INDUSTRY: TRANSITION FROM HARD TO SOFT ROBOTICS”. In: ().
- [2] Eddy M Rojas, Peerapong Aramvareekul, and Chris Hendrickson. “DISCUSSION AND CLOSURES Discussion of ” Is Construction Labor Productivity Really Declining? ””. In: 129129.1 (2003). DOI: 10.1061/.
- [3] Chen Feng et al. “Vision guided autonomous robotic assembly and as-built scanning on unstructured construction sites”. In: *Automation in Construction* (2015). ISSN: 09265805. DOI: 10.1016/j.autcon.2015.06.002.
- [4] Sanni Siltanen. “Theory and applications of marker-based augmented reality”. In: ().
- [5] T D Alter. “3D Pose from 3 Corresponding Points under Weak-Perspective Projection”. In: (1992).