

# Multistage Localization for High Precision Mobile Manipulation Tasks

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

# Multistage Localization for High Precision Mobile Manipulation Tasks

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  $\pm 2$  mm based on a multifaceted computer vision approach with only a USB webcam.

# Acknowledgments

Insert Acknowledgement

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>Related Work</b>	<b>3</b>
<b>3</b>	<b>Fundamentals of Autonomous Robotics</b>	<b>4</b>
3.1	ROS Concepts . . . . .	4
3.1.1	ROS Communication . . . . .	4
3.1.2	Unified Robot Description Format and Transformations . . . . .	5
3.2	Simultaneous Localization and Mapping Concepts . . . . .	5
3.3	Localization and Path Planning Concepts . . . . .	7
3.4	Manipulation Concepts . . . . .	11
3.5	State Machines Concepts . . . . .	14
3.6	Camera Concepts . . . . .	14
3.7	Computer Vision Concepts . . . . .	17
3.7.1	Color Spaces . . . . .	17
3.7.2	Linear and Non-Linear Filters . . . . .	18
3.7.3	Binary Operations . . . . .	20
3.7.4	Hough Circle Detection . . . . .	21
3.7.5	Good Feature To Track . . . . .	23
3.7.6	Optical Flow . . . . .	26
3.7.7	Pose Estimation and Tracking Through Augmented Reality Tag Detection, . . . . .	27

<b>4 Multistage Localization for High Precision Mobile Manipulation Tasks</b>	<b>29</b>
4.1 Approach Overview . . . . .	29
4.2 Global Map Creation and Task Location Specification . . . . .	31
4.3 Autonomous Localization and Navigation . . . . .	31
4.4 Task Association and A Priori Knowledge . . . . .	32
4.5 Generic Framework for Multi-Stage Computer Vision Algorithm . . . . .	32
4.5.1 Initial Feature Location Prediction . . . . .	32
4.5.2 Corrected Feature Locations . . . . .	32
4.6 Approach Implementation . . . . .	33
4.6.1 System Overview . . . . .	33
4.6.2 Drilling Framework . . . . .	34
4.6.3 Sealant Application Framework . . . . .	35
<b>5 Experiments and Results</b>	<b>37</b>
5.1 Hardware Architecture . . . . .	37
5.2 Software Architecture . . . . .	38
5.3 Experiments . . . . .	38
5.3.1 Camera Calibration Setup . . . . .	38
5.3.2 Navigation System Experimental Setup . . . . .	41
5.3.3 Drilling Operation Experimental Setup . . . . .	41
5.3.4 Sealant Application Experimental Setup . . . . .	42
5.4 Results . . . . .	43
5.4.1 Camera Calibration Accuracy Achieved . . . . .	43
5.4.2 Navigation System Accuracy Achieved . . . . .	43
5.4.3 Discussion of Drilling Operation Accuracy . . . . .	44
5.4.4 Discussion of Sealant Application Accuracy . . . . .	44
<b>6 Conclusion and Future Work</b>	<b>46</b>
<b>Bibliography</b>	<b>47</b>

# List of Figures

1.1	Vision for future transformable production systems . . . . .	2
3.1	Simple ROS Node Flowchart . . . . .	4
3.2	Robot Model with TFs . . . . .	5
3.3	Overview of SLAM Framework. [Riisgaard et al. 2005] . . . . .	5
3.4	Example of 2-D Occupancy Grid Map Produced. . . . .	6
3.5	Real World Performance Analysis of ROS Available SLAM Algorithms. [Santos et al. 2013] . . . . .	6
3.6	ROS Navigation Stack setup. . . . .	7
3.7	1D Particle Filter Example . . . . .	8
3.8	Particle Filter Resampling Example . . . . .	9
3.9	Visual of AMCL in RVIZ . . . . .	10
3.10	Trajectory Rollout Path Planning Framework . . . . .	11
3.11	Moveit!'s System Architecture. . . . .	12
3.12	Forward and Inverse Kinematics Example . . . . .	13
3.13	Graphical View of State Machine using SMACH . . . . .	14
3.14	Digital Image Formation. . . . .	15
3.15	Ground Sample Distance Effects on Image Quality. . . . .	16
3.16	Shutter Type Effects on Image Quality. . . . .	17
3.17	RGB and HSV color space models . . . . .	18
3.18	Effects of common linear filters. . . . .	19
3.19	Histogram equalization depiction. . . . .	20

3.20	Effect of histogram equalization on an image.	20
3.21	Effects of common binary operations.	21
3.22	Summary of preprocessing operation performed by Hough Circle detector.	21
3.23	Summary of Hough Circle detector with known radius.	22
3.24	Summary of Hough Circle detector with unknown radius.	22
3.25	Summary of Good Feature detector.	23
3.26	Image Gradient Example.	24
3.27	Difference Between Good Feature and Harris Corner Scoring Functions.	25
3.28	Example of optical flow.	26
3.29	Summary of Kanade-Lucas-Tomasi feature tracker.	27
3.30	Summary of ALVAR AR Tag Detection Framework.	28
4.1	Multistage Localization Approach Overview.	30
4.2	2-D map environment with specified start locations.	31
4.3	Initial Feature Localization Framework.	32
4.4	Feature Detection and Tracking Pipeline.	33
4.5	Manipulator Correction Control Loop.	33
4.6	General system overview.	34
4.7	Drilling operation framework.	35
4.8	Sealant Application Framework.	36
5.1	Clearpath Robotics' Husky and Fetch Robotics' Fetch Mobile Manipulator	38
5.2	Camera Calibration Setup.	39
5.3	Extracted Corners of Calibration Pattern.	40
5.4	Extracted Corners and Laser Center	40
5.5	Navigation System Experimental Setup.	41
5.6	Drilling Operations Experimental Setup.	42
5.7	Sealant Application Experimental Setup.	43

# List of Tables

3.1	Real World Error Estimation for ROS Available SLAM Algorithm. [Santos et al. 2013] . . . . .	6
3.2	Comparison Between ROS Available Moveit! Inverse Kinematic Plugins. [Beeson et al. 2015] . . . . .	13
5.1	Global Localization Accuracy Comparison Given Distance From Global Location. . . . .	44
5.2	Drilling Operation Accuracy Comparison Given Distance From Work Surface. . . . .	44
5.3	Sealant Application Accuracy Comparison Given Distance From Work Surface. . . . .	45

# 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.

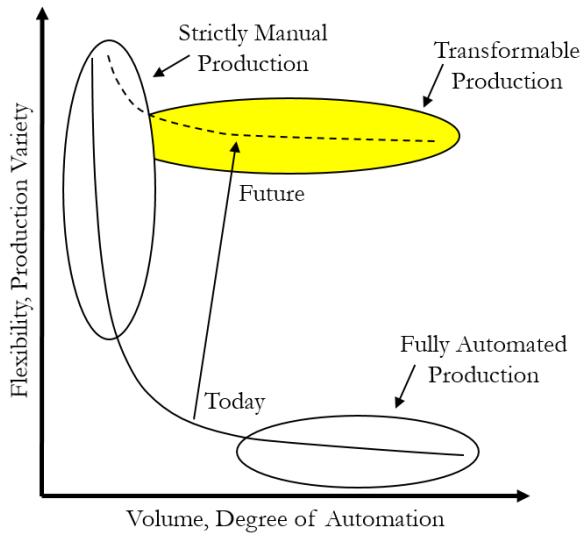


Figure 1.1: Vision for future transformable production systems.

# **Chapter 2**

## **Related Work**

This section will discuss related work.

# Chapter 3

## Fundamentals of Autonomous Robotics

Insert text outlining upcoming sections.

### 3.1 ROS Concepts

Insert Text Defining ROS.

#### 3.1.1 ROS Communication

Insert Text Regarding Node/Topic/Service Communication.

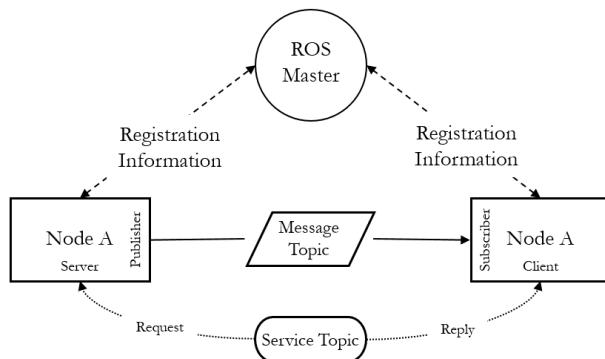


Figure 3.1: Simple ROS Node Flowchart

### 3.1.2 Unified Robot Description Format and Transformations

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

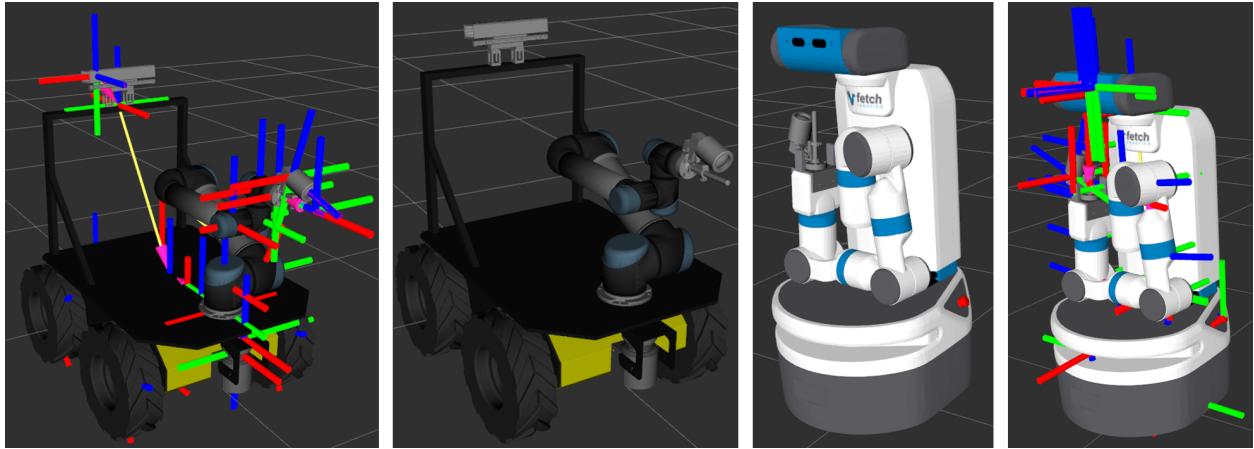


Figure 3.2: Robot Model with TFs

## 3.2 Simultaneous Localization and Mapping Concepts

Insert Text Outlining the Basic Intuition Behind SLAM.

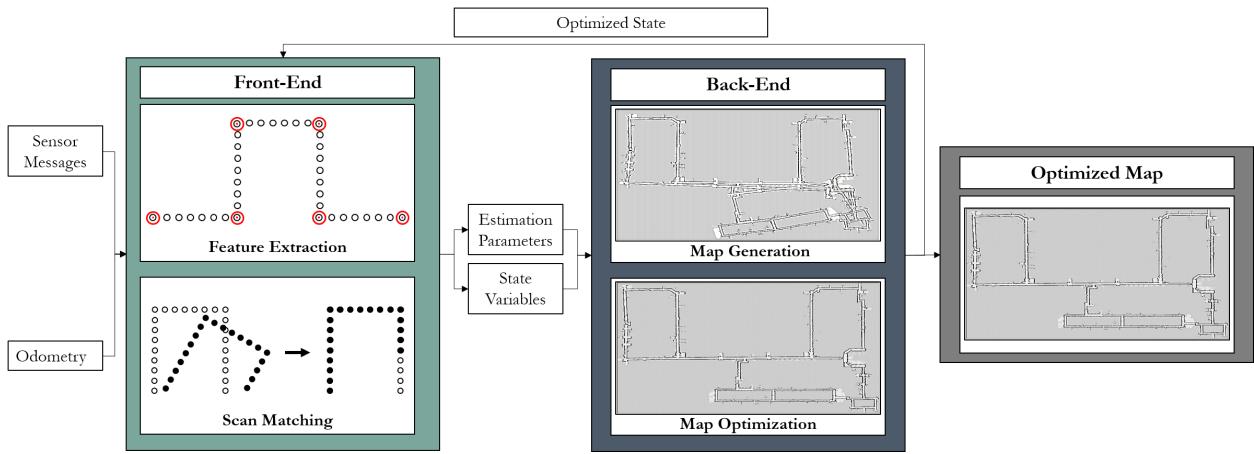


Figure 3.3: Overview of SLAM Framework. [Riisgaard et al. 2005]

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

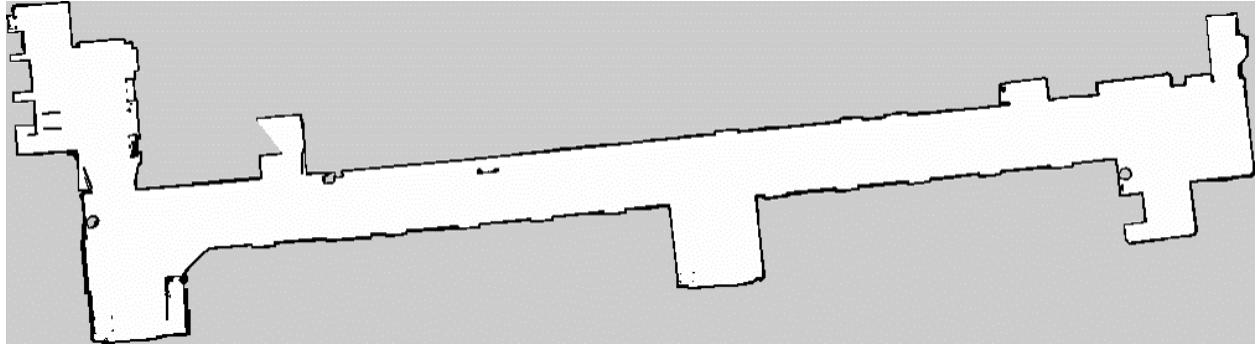


Figure 3.4: Example of 2-D Occupancy Grid Map Produced.

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

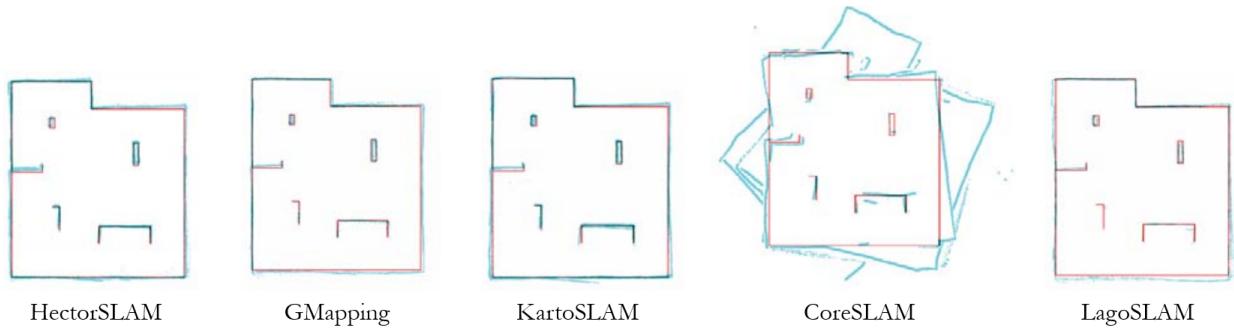


Figure 3.5: Real World Performance Analysis of ROS Available SLAM Algorithms. [Santos et al. 2013]

Table 3.1: 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

### 3.3 Localization and Path Planning Concepts

Insert Text Outlining The Navigation Stack's Framework.

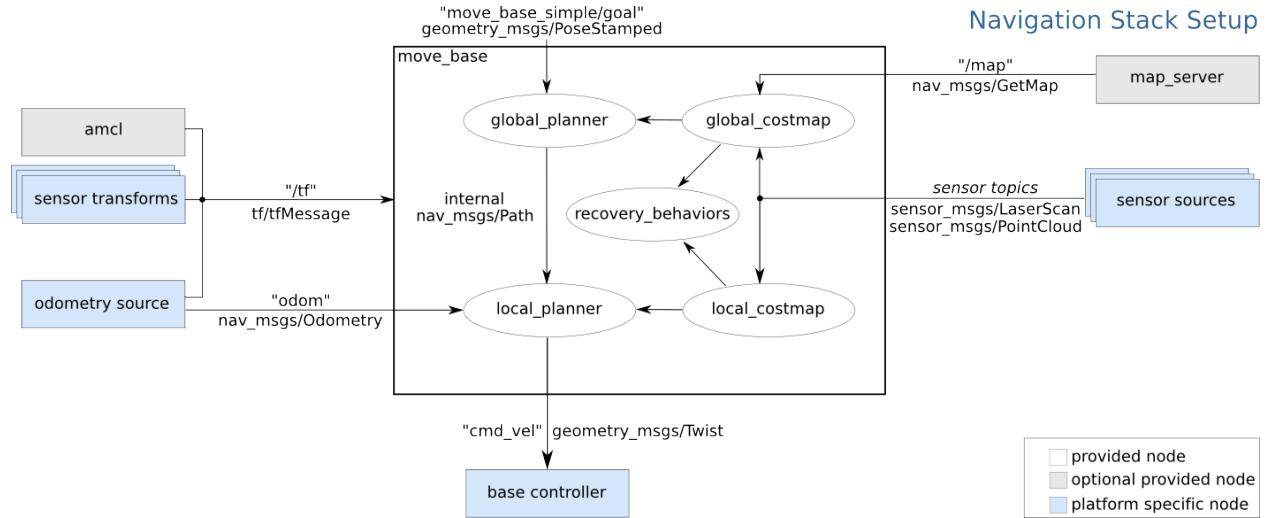


Figure 3.6: ROS Navigation Stack setup.

Insert Text Explaining How Adaptive Monte Carlo Localization Works. Specifically Start With An Overall Review.

Explain the Basics of a Particle Filter in 1D.

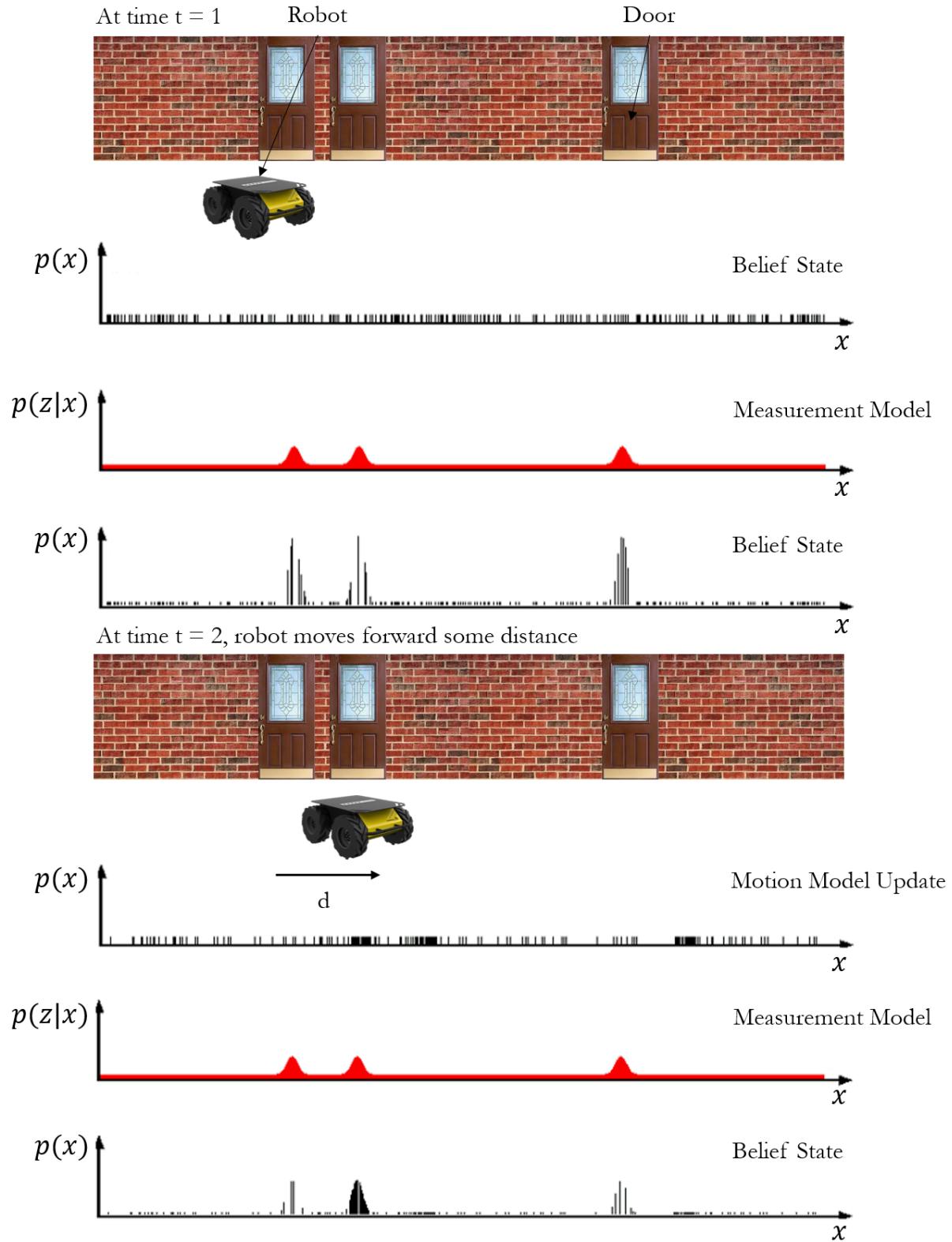


Figure 3.7: 1D Particle Filter Example

Explain How the Particle Filter works in 2D. Refer to 2D Map Figure. The Equations Below are Used to Calculate Scan Correlation and Then How to Weight Each Particle.

$$s = \frac{\sum_m \sum_n (A_{mn} - \bar{A})(B_{mn} - \bar{B})}{\sqrt{(\sum_m \sum_n (A_{mn} - \bar{A})^2)(\sum_m \sum_n (B_{mn} - \bar{B})^2)}} \quad (3.1)$$

$$W_k \leftarrow W_{k-1}s \quad (3.2)$$

Explain the Basics of Resampling in a Particle Filter and Why It is Done.

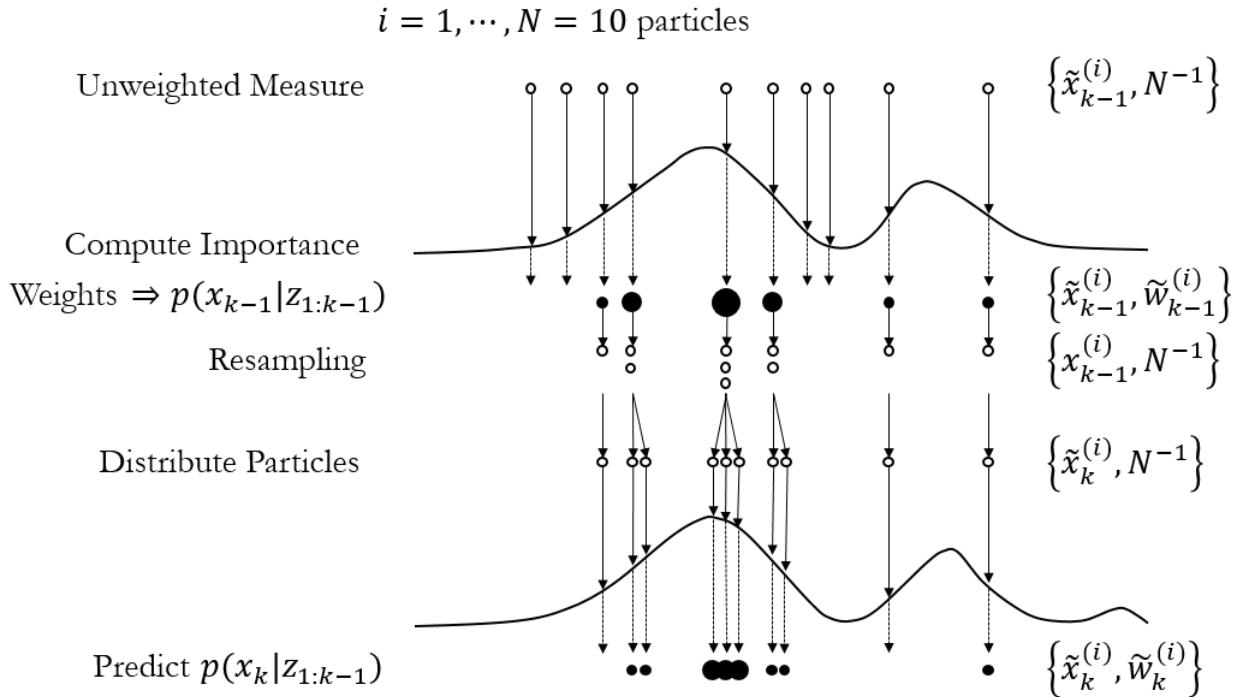


Figure 3.8: Particle Filter Resampling Example

Explain KLD Sampling and Why it is Used.

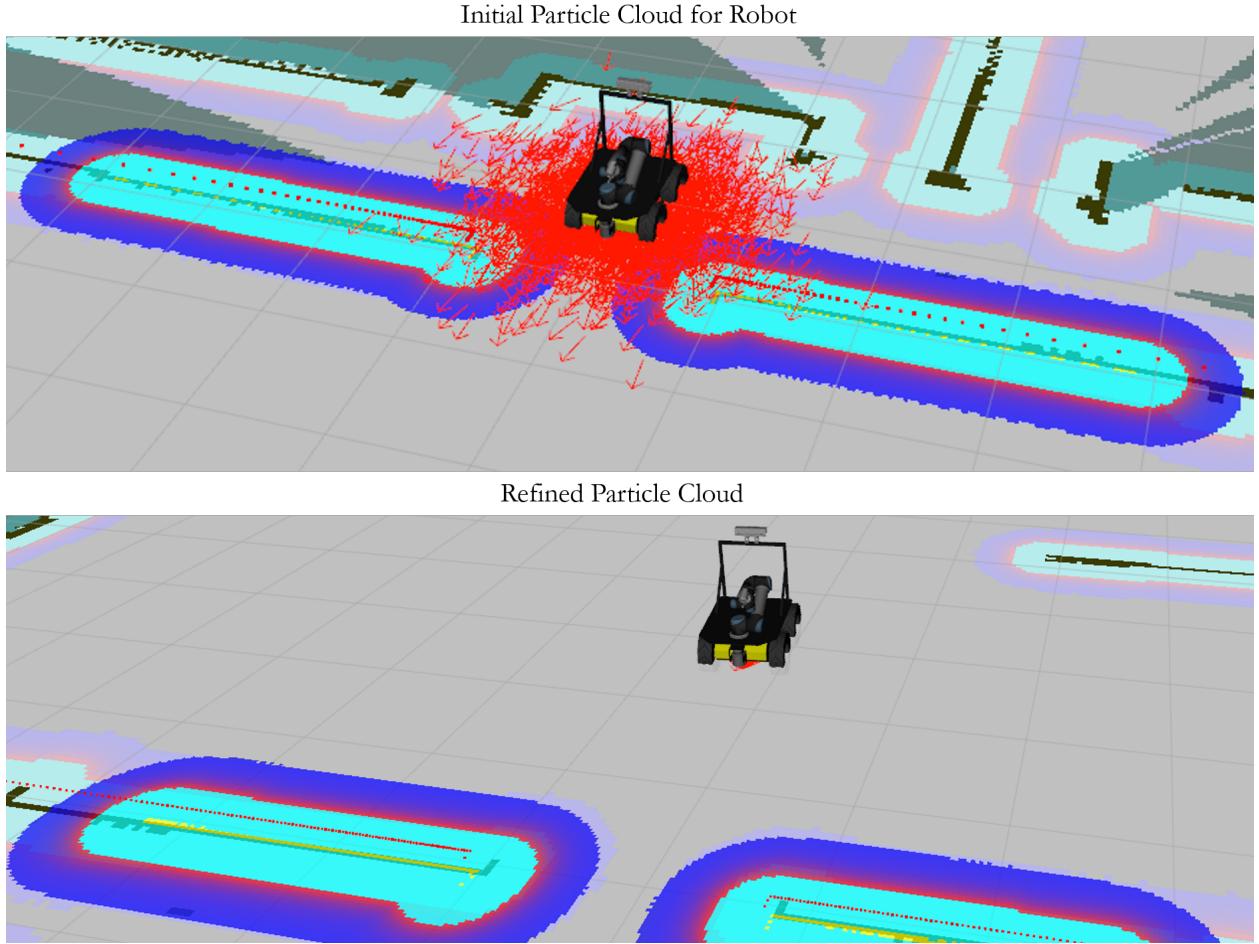


Figure 3.9: Visual of AMCL in RVIZ

Insert Text Explaining How Trajectory Rollout Works for Path Planning.

$$C(k) = \alpha Obs + \beta Gdist + \gamma Pdist + \delta \frac{1}{\dot{x}^2} \quad (3.3)$$

Where  $Obs$  is the sum of grid cell cost through which the trajectory passes (taking account of the robot's actual footprint in the grid);  $Gdist$  and  $Pdist$  are the estimated shortest distance from the endpoint of the trajectory to the goal and the optimal path, respectively; and  $\dot{x}$  is the translation component of the velocity command that produces the trajectory.

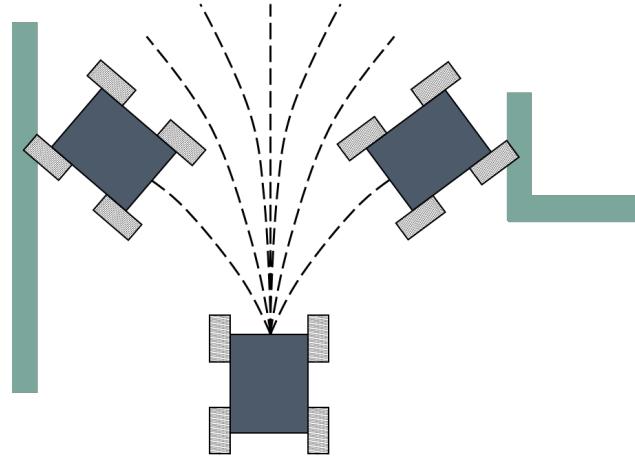


Figure 3.10: Trajectory Rollout Path Planning Framework

### 3.4 Manipulation Concepts

Insert Text Outlining Moveit!'s System Architecture.

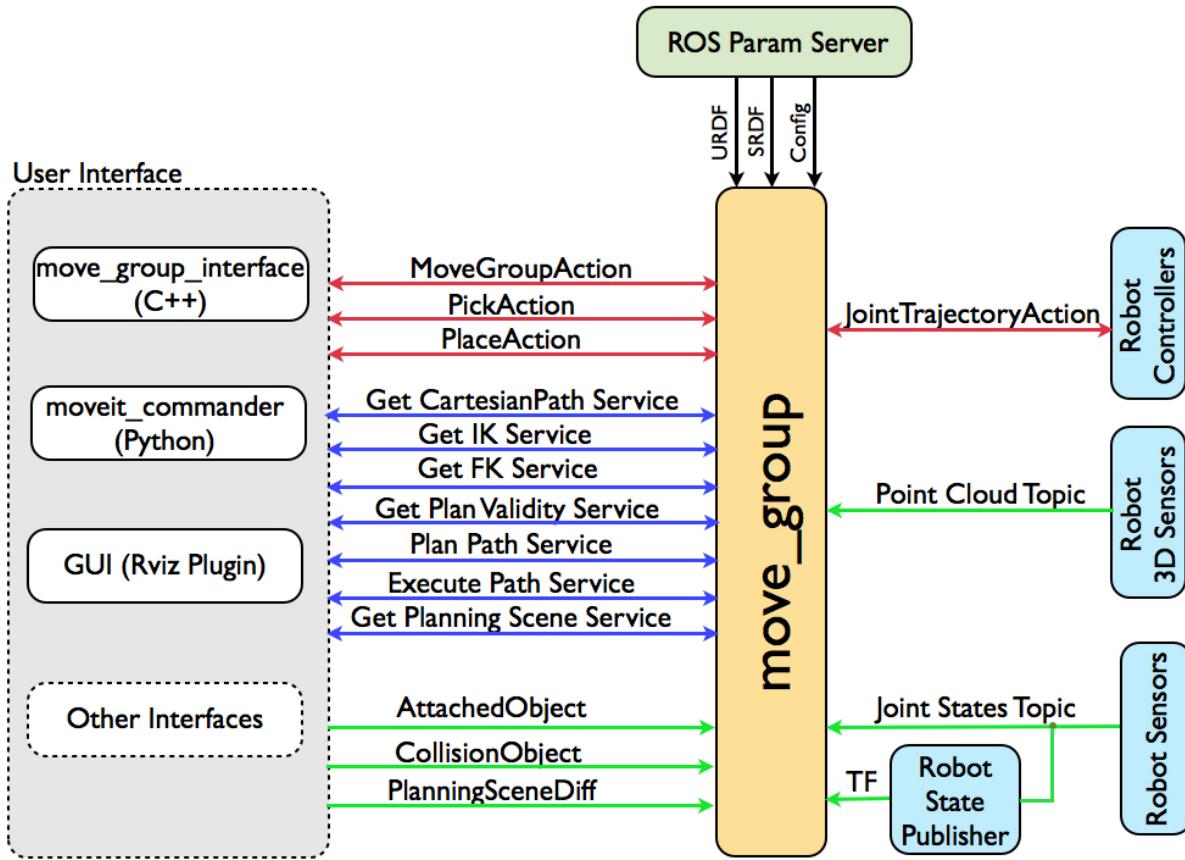


Figure 3.11: Moveit!'s System Architecture.

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

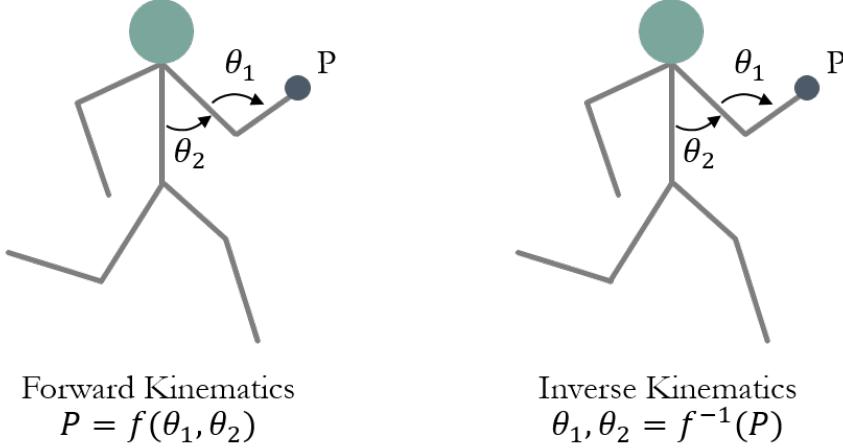


Figure 3.12: Forward and Inverse Kinematics Example

Insert Test Explaining the Difference Between The Numerical Solvers available in ROS (KDL, KDL-RR and Trac\_IK) and Why Trac\_IK was chosen to be used.

Table 3.2: Comparison Between ROS Available Moveit! Inverse Kinematic Plugins. [Beeson et al. 2015]

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

### 3.5 State Machines Concepts

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

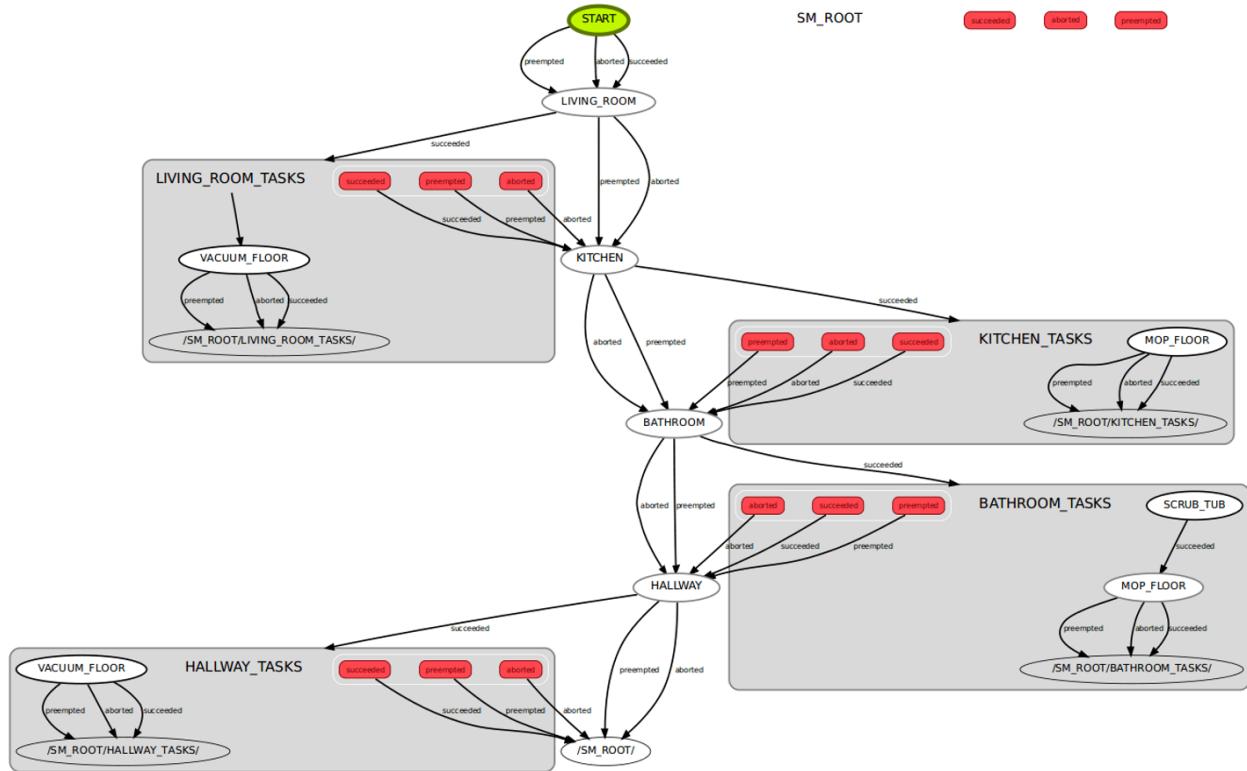


Figure 3.13: Graphical View of State Machine using SMACH

### 3.6 Camera Concepts

Insert Text Explaining How Digital Image Are Recorded and Stored.

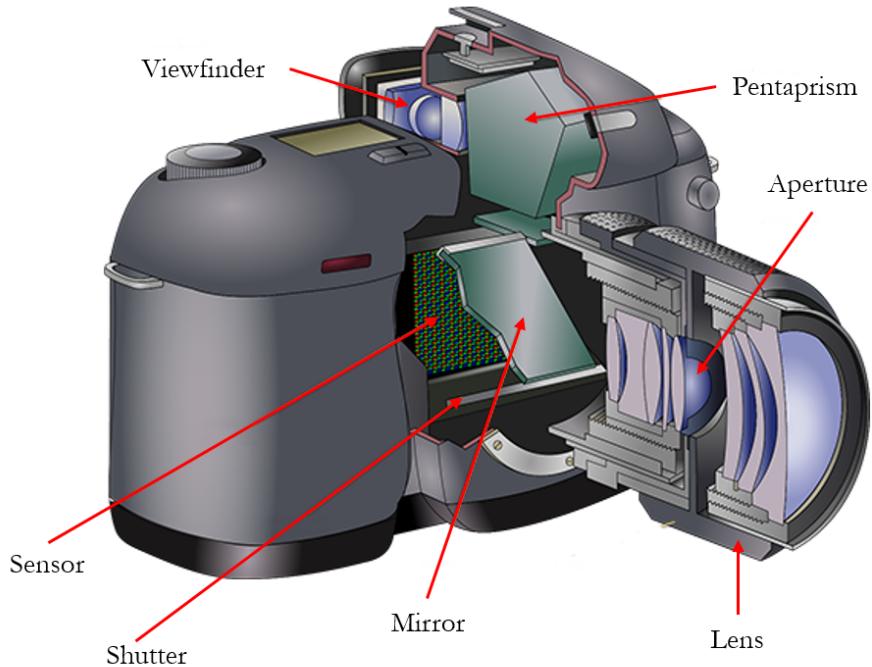


Figure 3.14: Digital Image Formation.

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

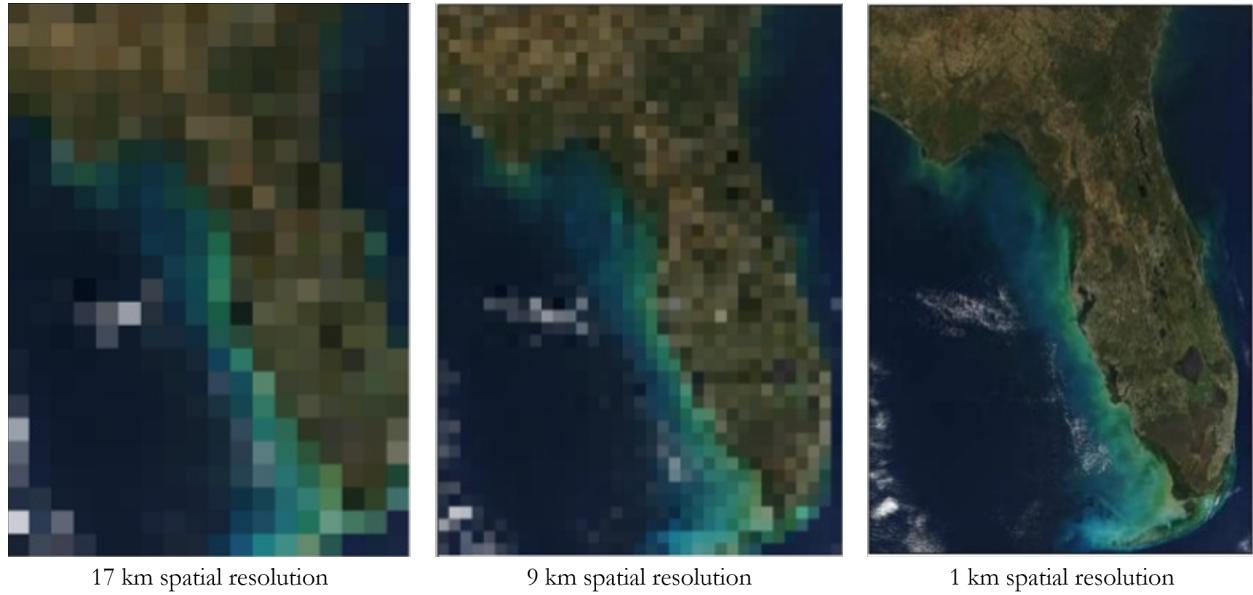


Figure 3.15: Ground Sample Distance Effects on Image Quality.

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

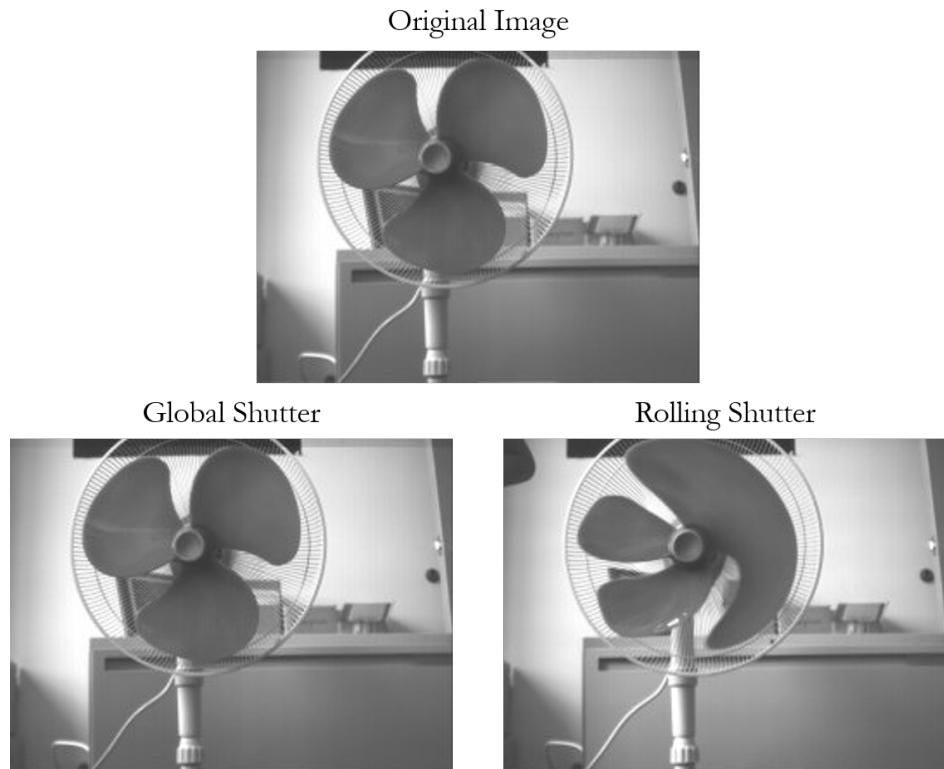


Figure 3.16: Shutter Type Effects on Image Quality.

## 3.7 Computer Vision Concepts

Insert Text Regarding Upcoming Sections.

### 3.7.1 Color Spaces

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

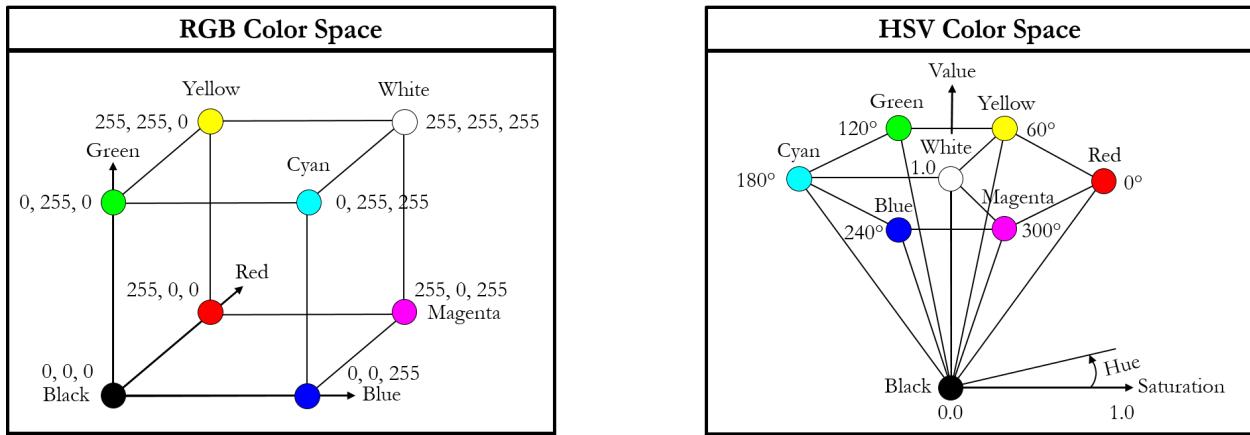


Figure 3.17: RGB and HSV color space models

### 3.7.2 Linear and Non-Linear Filters

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



Figure 3.18: Effects of common linear filters.

Insert Text Explaining the Basics behind Histogram Equalization.

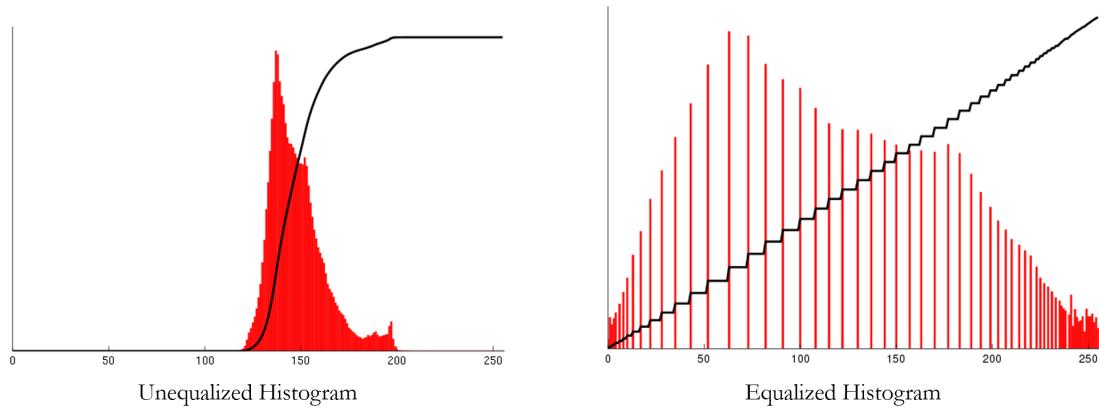


Figure 3.19: Histogram equalization depiction.

Insert Text Explaining The Effect of Histogram Equalization on Images.



Figure 3.20: Effect of histogram equalization on an image.

### 3.7.3 Binary Operations

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

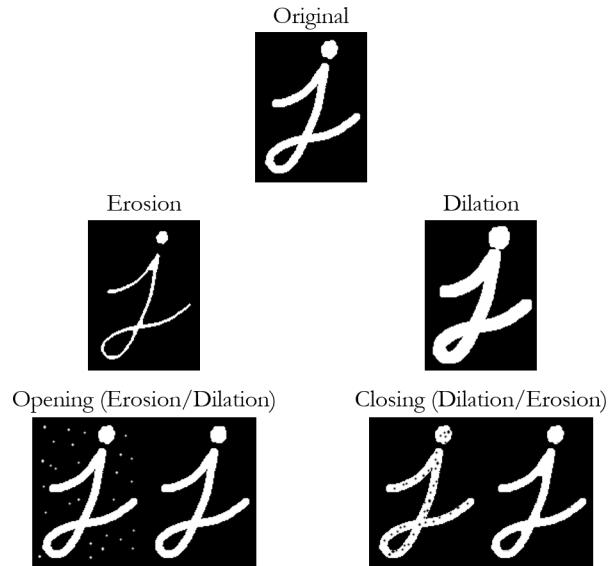


Figure 3.21: Effects of common binary operations.

### 3.7.4 Hough Circle Detection

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

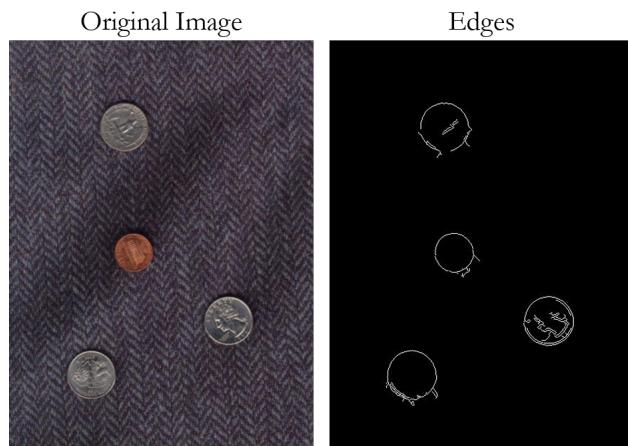


Figure 3.22: Summary of preprocessing operation performed by Hough Circle detector.

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

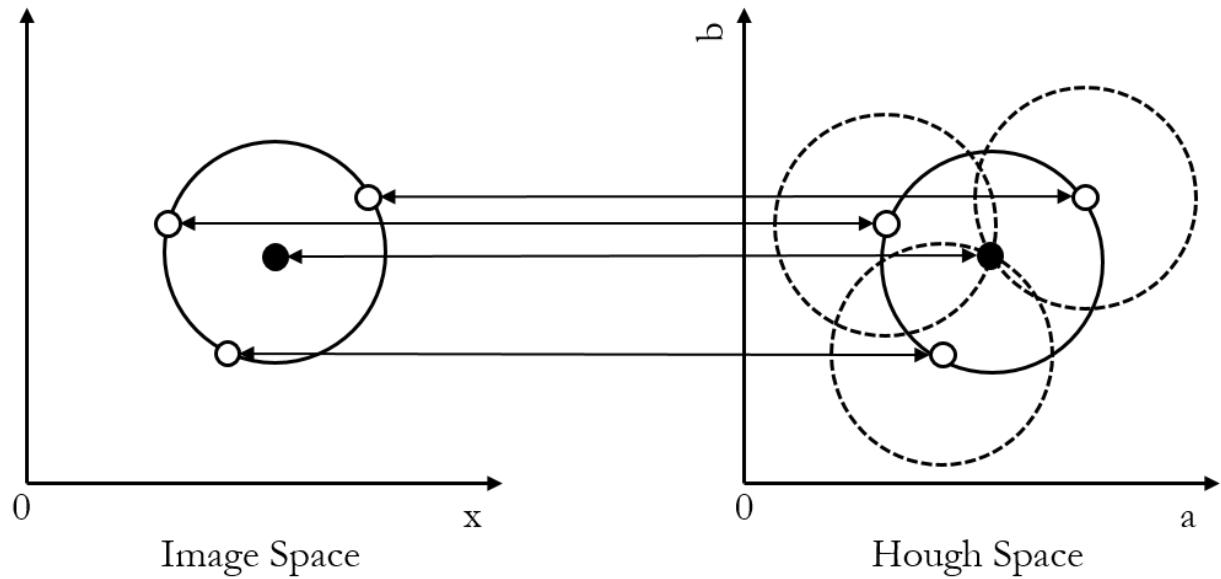


Figure 3.23: Summary of Hough Circle detector with known radius.

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

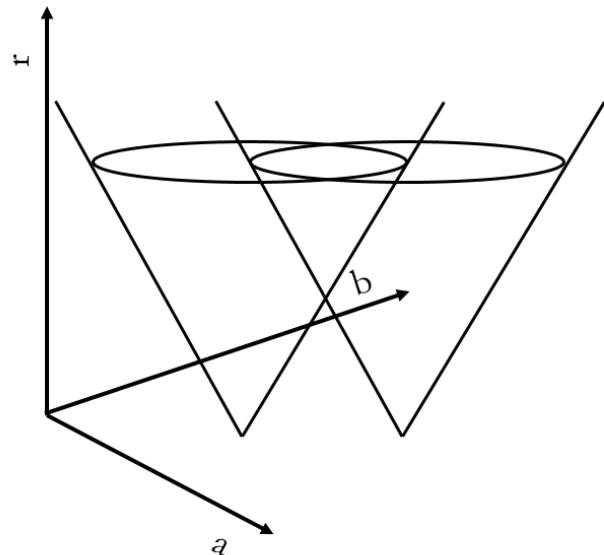


Figure 3.24: Summary of Hough Circle detector with unknown radius.

### 3.7.5 Good Feature To Track

Insert Text Explaining the Basic Intuition Behind Good Feature.

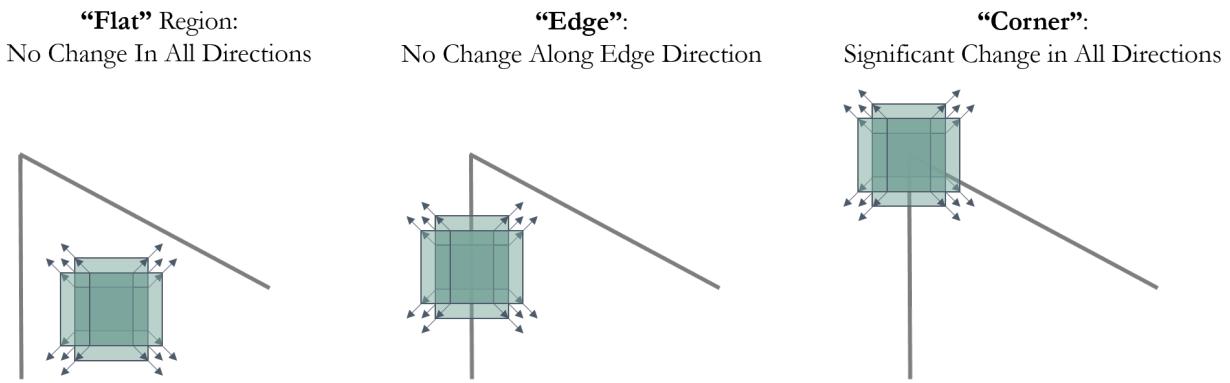


Figure 3.25: Summary of Good Feature detector.

Insert Text Explaining an Image Gradient/Derivative and Why We Look for Them.

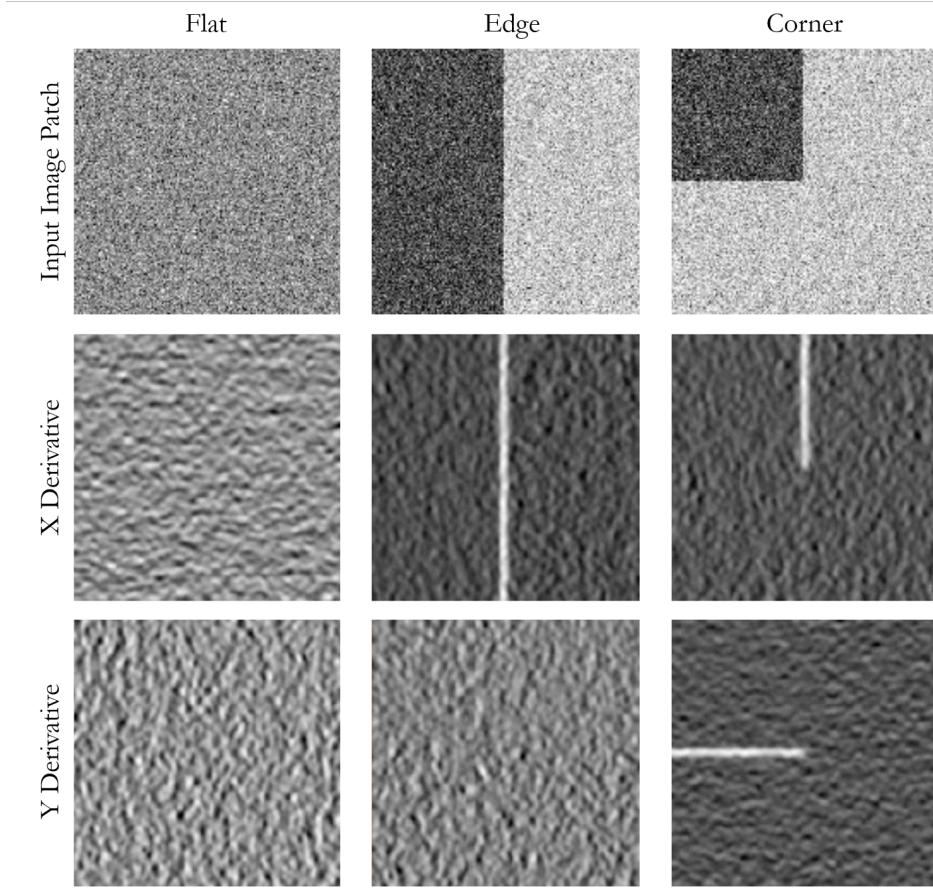


Figure 3.26: Image Gradient Example.

Insert Text Explaining Math.

$$E(u, v) = \sum_{x,y} w(x, y)[I(x + u, y + v) - I(x, y)]^2 \quad (3.4)$$

$$\sum_{x,y} [I(x + u, y + v) - I(x, y)]^2 \quad (3.5)$$

$$E(u, v) \approx \sum_{x,y} [I(x, y) + uI_x + vI_y - I(x, y)]^2 \quad (3.6)$$

$$E(u, v) \approx \sum_{x,y} u^2 I_x^2 + 2uv I_x I_y + v^2 I_y^2 \quad (3.7)$$

$$E(u, v) \approx [u \ v] \left( \sum_{x,y} w(x, y) \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix} \right) \begin{bmatrix} u \\ v \end{bmatrix} \quad (3.8)$$

$$E(u, v) \cong [u \ v] M \begin{bmatrix} u \\ v \end{bmatrix} \quad (3.9)$$

Where

$$M = \sum_{x,y} w(x, y) \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix}$$

Insert Text Explaining How Both Harris Corner and Good Feature Scoring Functions Work and Why Good Feature Performs Slightly Higher.

$$R = \det(M) - k(\text{trace}(M))^2 \quad (3.10)$$

Where  $\det(M) = \lambda_1 \lambda_2$  and  $\text{trace}(M) = \lambda_1 + \lambda_2$ . So,

$$R = \lambda_1 \lambda_2 - k(\lambda_1 + \lambda_2)^2 \quad (3.11)$$

$$R = \min(\lambda_1, \lambda_2) \quad (3.12)$$

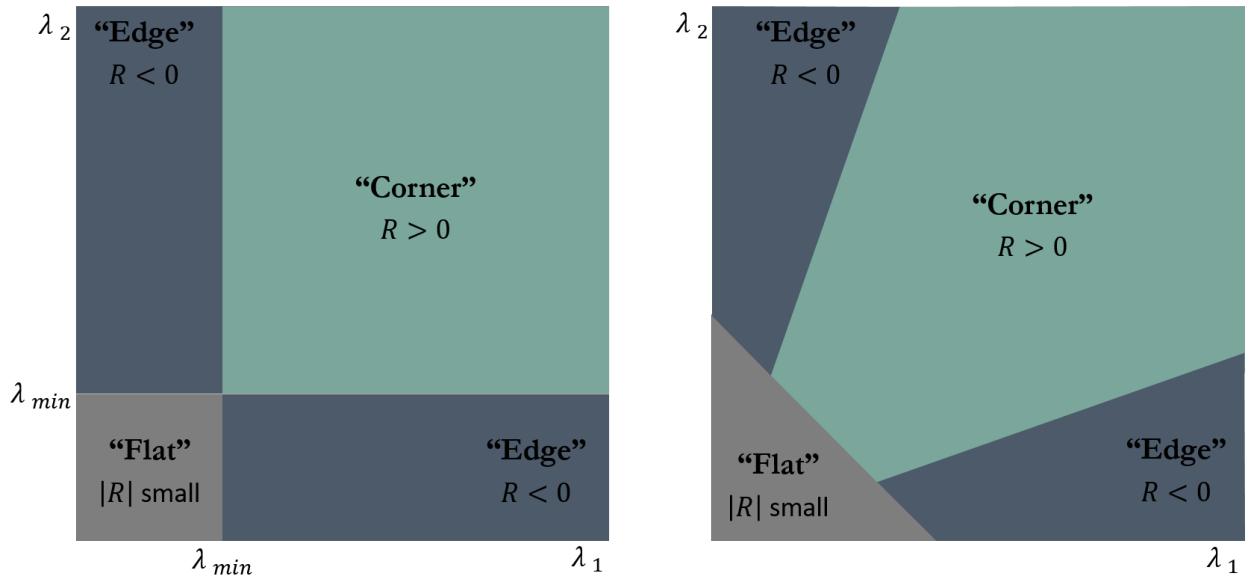


Figure 3.27: Difference Between Good Feature and Harris Corner Scoring Functions.

### 3.7.6 Optical Flow

Insert Text Explaining the Basic Intuition Behind Optical Flow.



Figure 3.28: Example of optical flow.

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

$$\sum_x [T(\mathbf{W}(\mathbf{x}; \Delta \mathbf{p})) - I(\mathbf{W}(\mathbf{x}; \mathbf{p}))]^2 \quad (3.13)$$

$$\sum_x \left[ T(\mathbf{W}(\mathbf{x}; \mathbf{0})) + \nabla T \frac{\partial \mathbf{W}}{\partial \mathbf{p}} \Delta \mathbf{p} - I(\mathbf{W}(\mathbf{x}; \mathbf{p})) \right]^2 \quad (3.14)$$

$$\Delta \mathbf{p} = H^{-1} \sum_x \left[ \nabla T \frac{\partial \mathbf{W}}{\partial \mathbf{p}} \right]^T [I(\mathbf{W}(\mathbf{x}; \mathbf{p})) - T(\mathbf{x})] \quad (3.15)$$

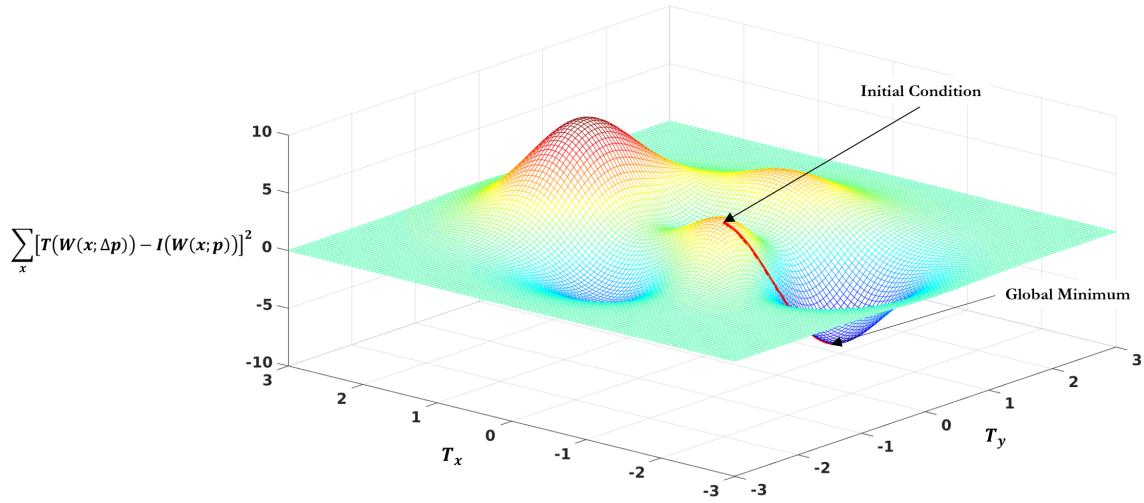


Figure 3.29: Summary of Kanade-Lucas-Tomasi feature tracker.

### 3.7.7 Pose Estimation and Tracking Through Augmented Reality Tag Detection,

Insert Text Describing How AR Tag Detection, Pose Estimation, and Tracking Works with AR\_Track\_Alvar.

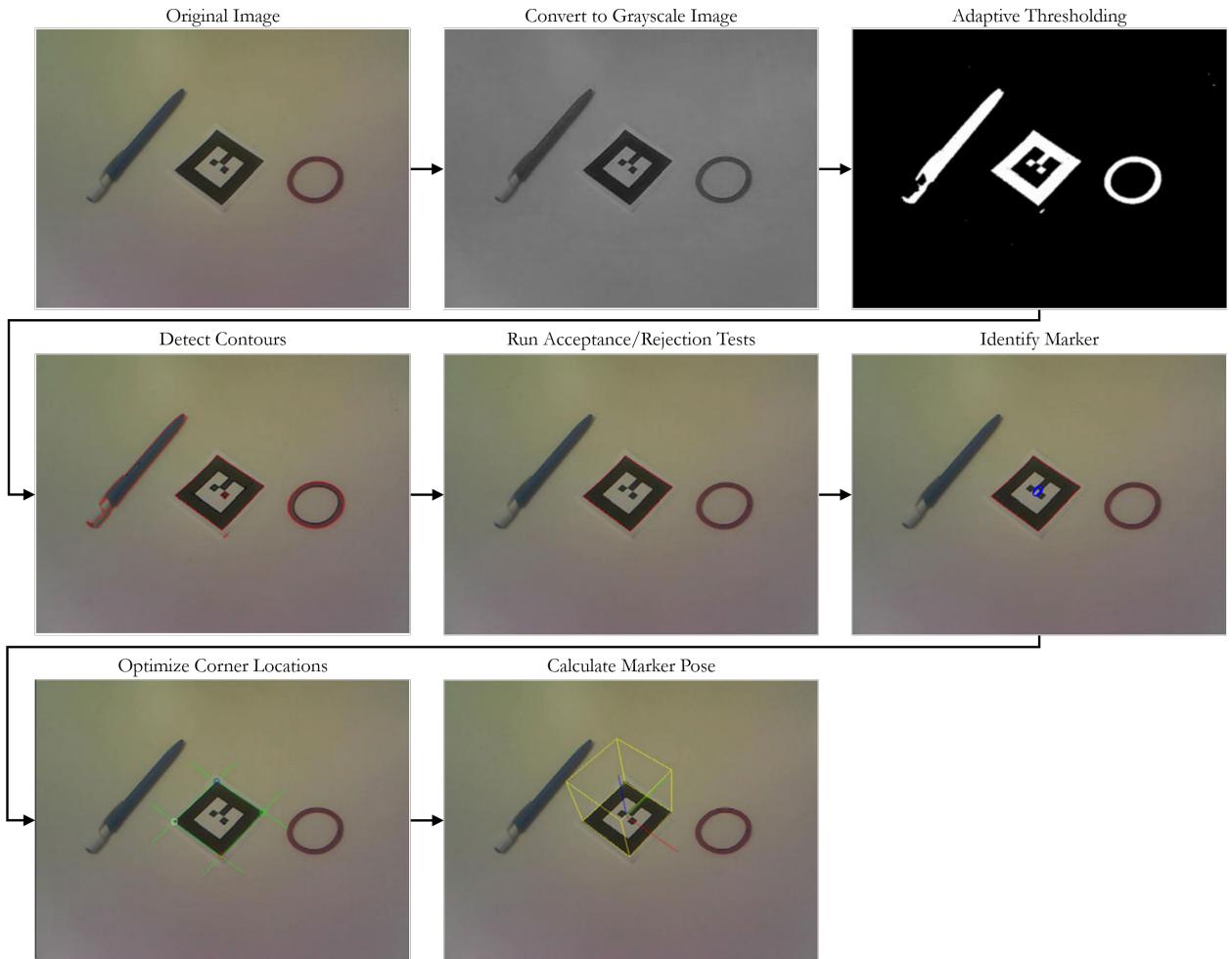


Figure 3.30: Summary of ALVAR AR Tag Detection Framework.

# **Chapter 4**

## **Multistage Localization for High Precision Mobile Manipulation Tasks**

Insert Text Outlining the Upcoming Sections.

### **4.1 Approach Overview**

Insert Text Outlining the Steps of the Approach.



Figure 4.1: Multistage Localization Approach Overview.

## 4.2 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 4.2.

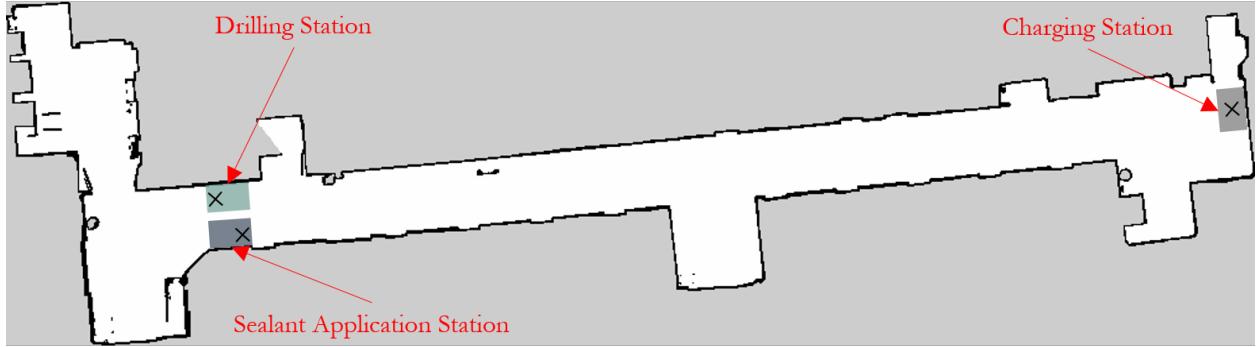


Figure 4.2: 2-D map environment with specified start locations.

## 4.3 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.

## 4.4 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 ??.

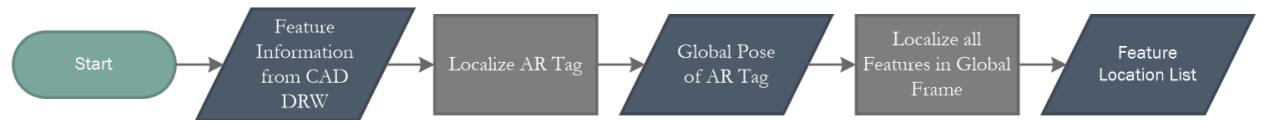


Figure 4.3: Initial Feature Localization Framework.

## 4.5 Generic Framework for Multi-Stage Computer Vision Algorithm

Insert Text Outlining the Upcoming Sections.

### 4.5.1 Initial Feature Location Prediction

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

### 4.5.2 Corrected Feature Locations

Currently the sponsor of this work uses a hole template on the object to be drilled to ensure accuracy within  $\pm 0.3$  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.

Insert Text Regarding Computer Vision Detection and Tracking System.



Figure 4.4: Feature Detection and Tracking Pipeline.

Insert Text Regarding Control Loop.

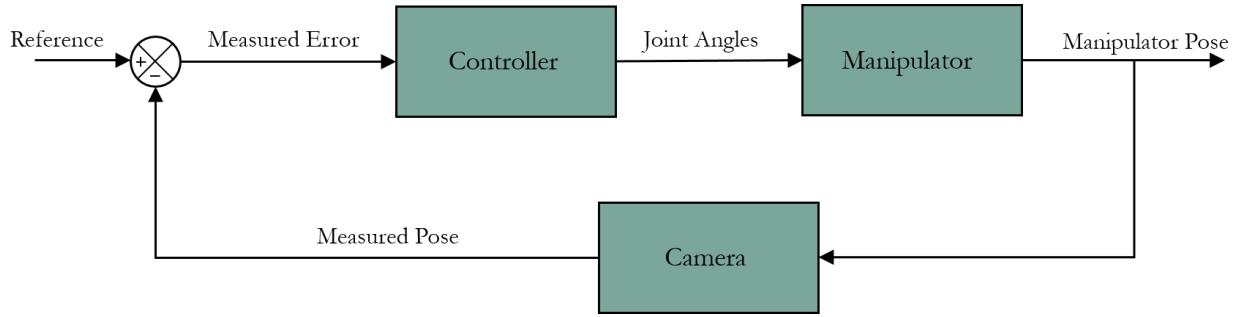


Figure 4.5: Manipulator Correction Control Loop.

## 4.6 Approach Implementation

Outline the upcoming sections.

### 4.6.1 System Overview

The generic framework of the system is depicted in Figure 4.6. The system allows the user to input predefined tasks. Given a priori knowledge of each tasks 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.

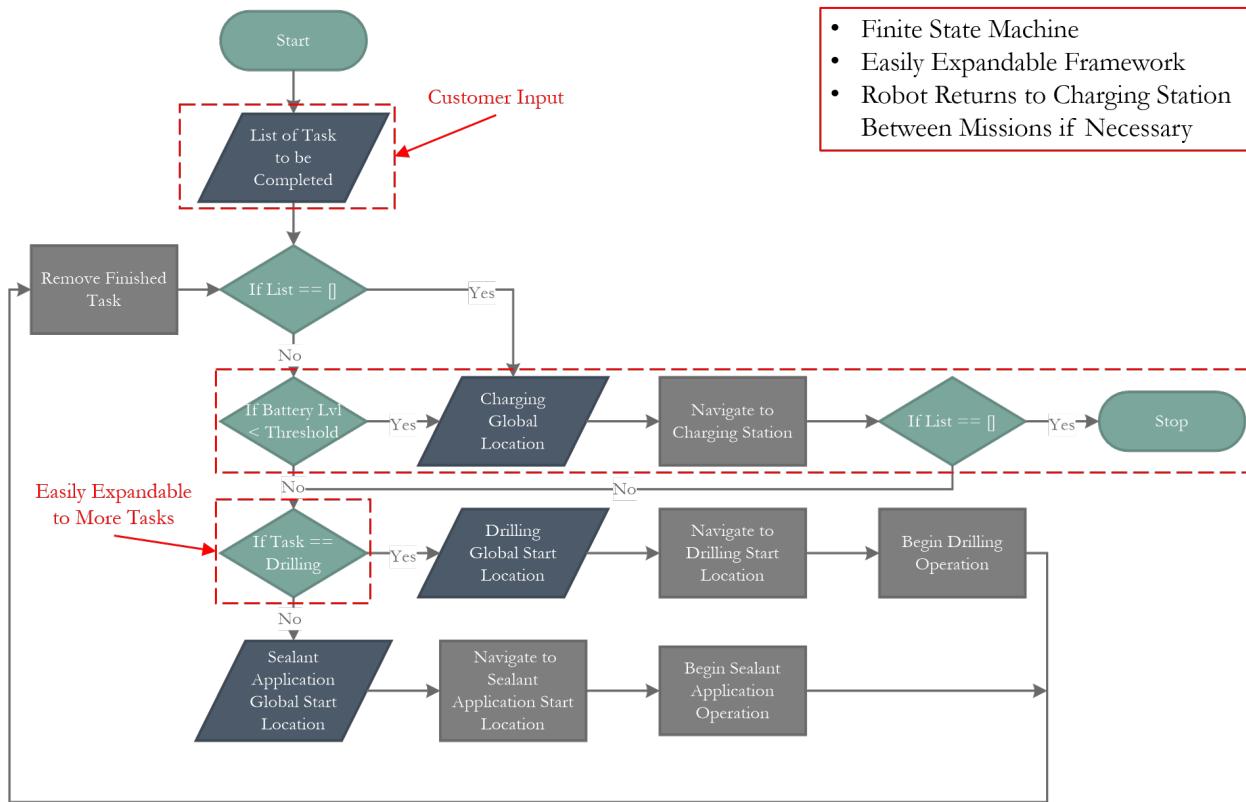


Figure 4.6: General system overview.

#### 4.6.2 Drilling Framework

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

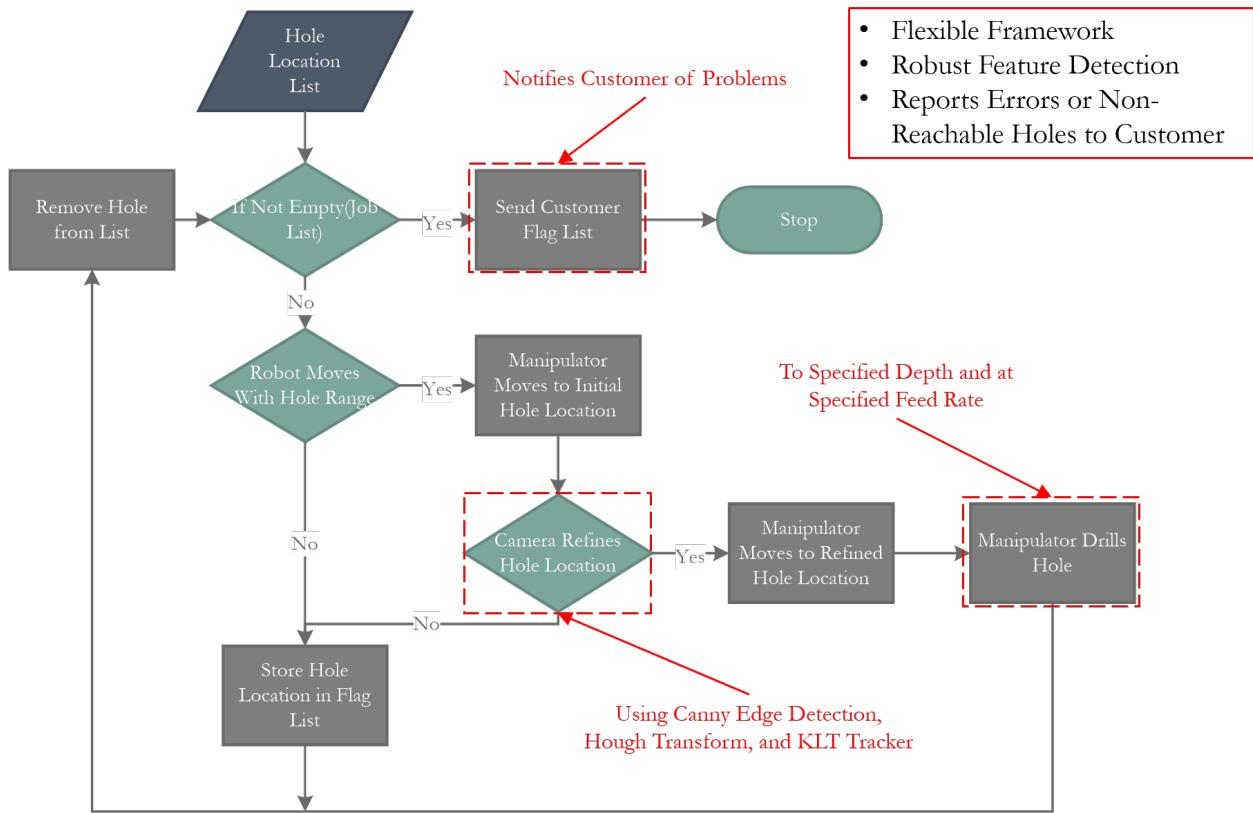


Figure 4.7: Drilling operation framework.

#### 4.6.3 Sealant Application Framework

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

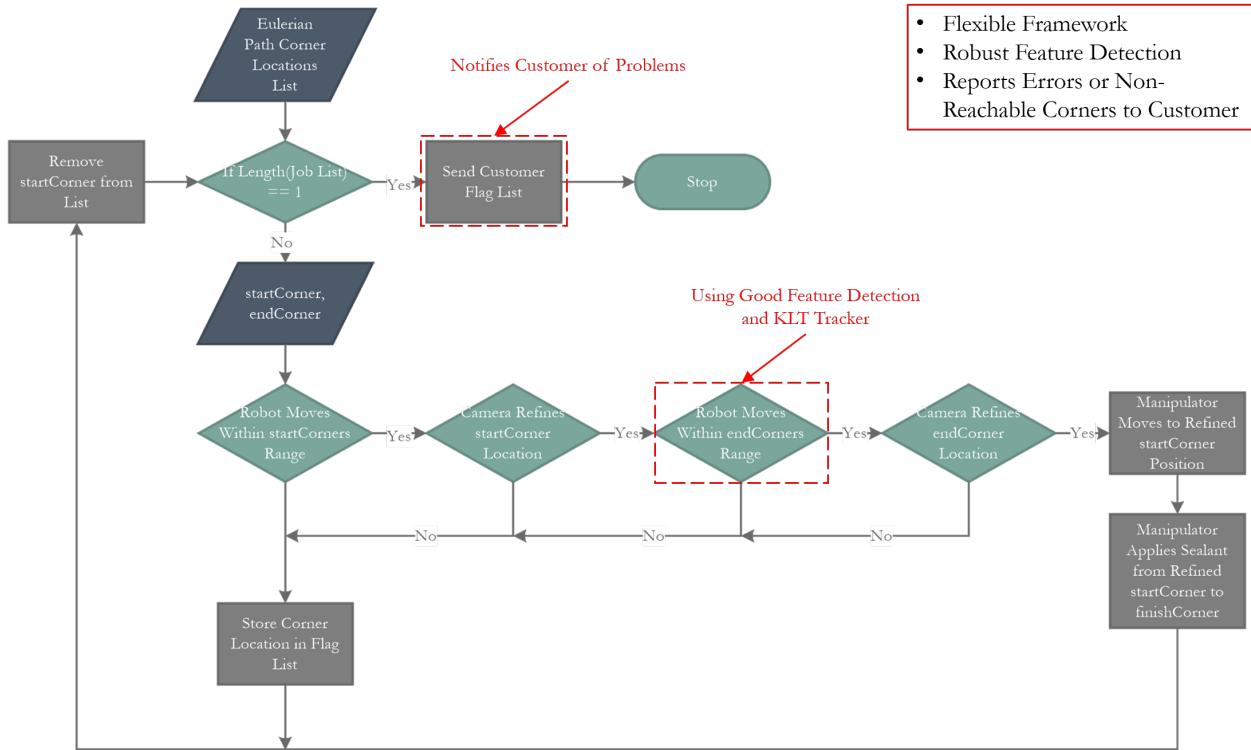


Figure 4.8: Sealant Application Framework.

# **Chapter 5**

## **Experiments and Results**

Outline the upcoming sections.

### **5.1 Hardware Architecture**

Insert Text Regarding Robot, Sensors, etc.

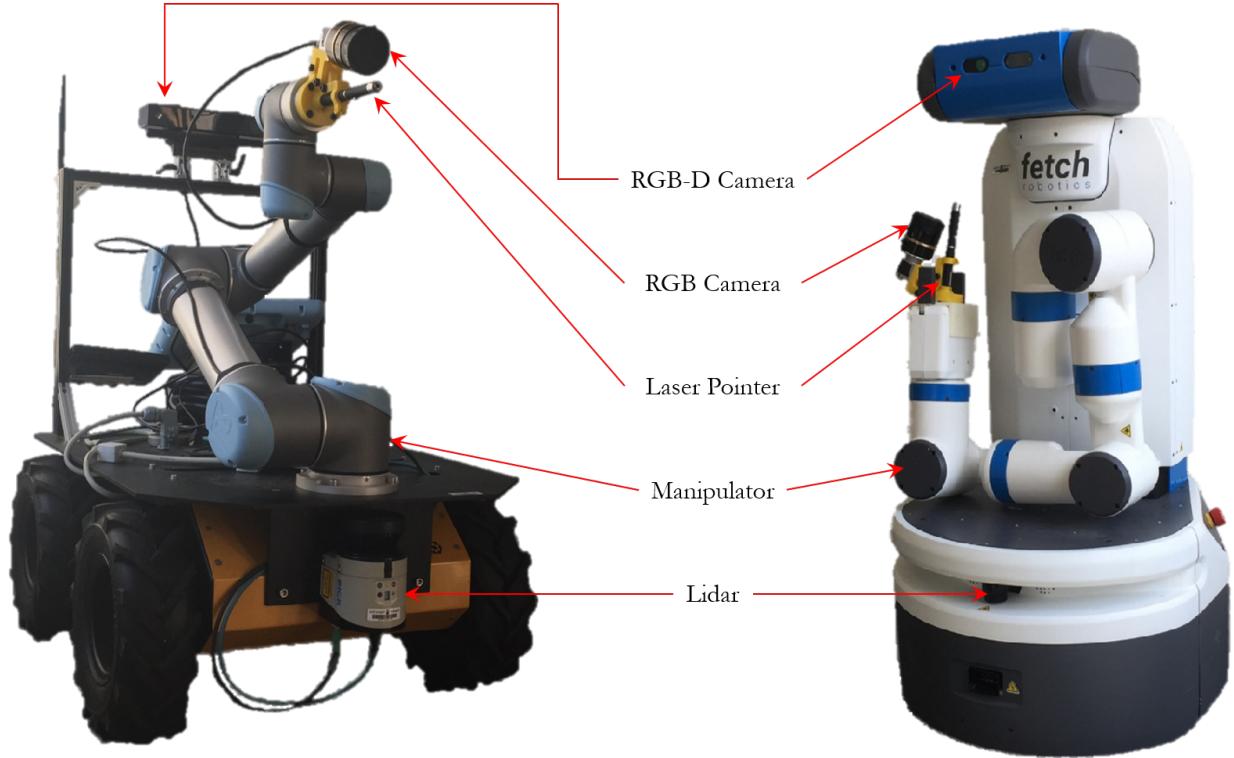


Figure 5.1: Clearpath Robotics' Husky and Fetch Robotics' Fetch Mobile Manipulator

## 5.2 Software Architecture

Insert Text Regarding The Software Architecture.

## 5.3 Experiments

Insert Text Regarding Upcoming Sections.

### 5.3.1 Camera Calibration Setup

Insert text setup for camera calibration.



Figure 5.2: Camera Calibration Setup.

Insert text regarding methods used for camera calibration (citing paper)

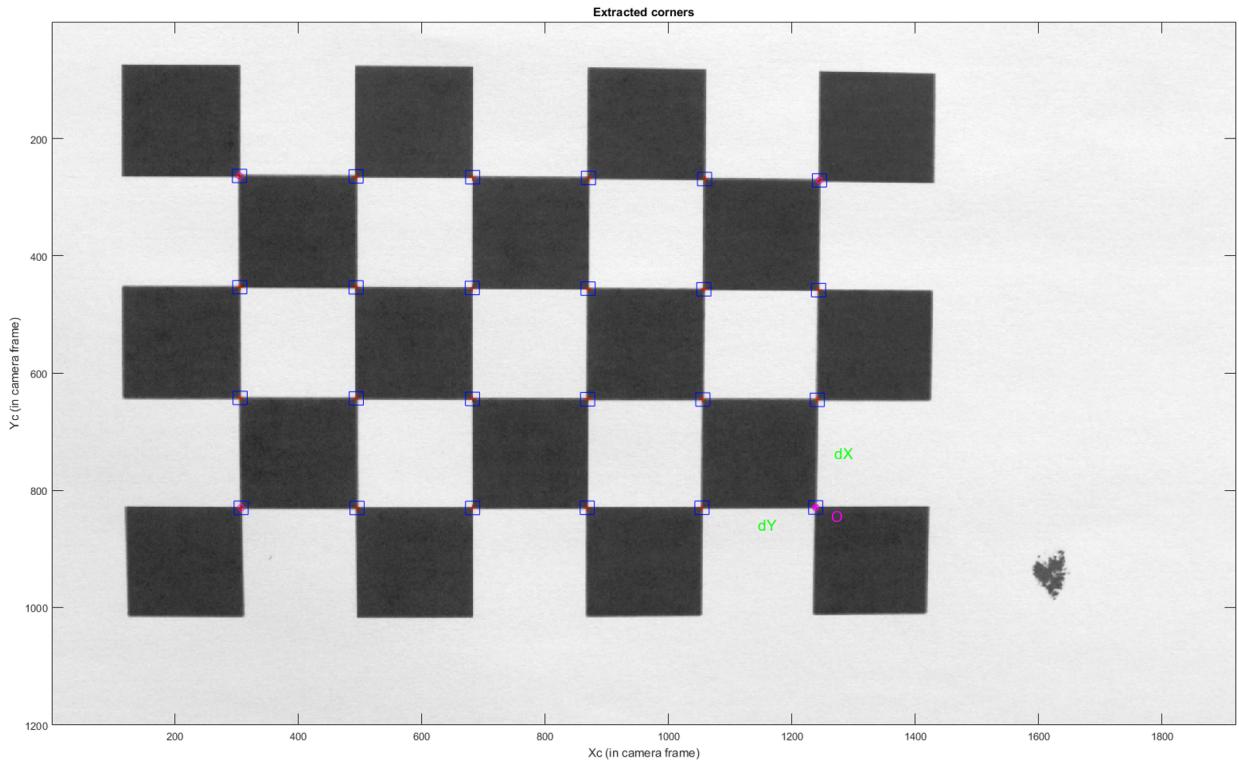


Figure 5.3: Extracted Corners of Calibration Pattern.

Insert Text and Math Regarding Calibration of Laser.

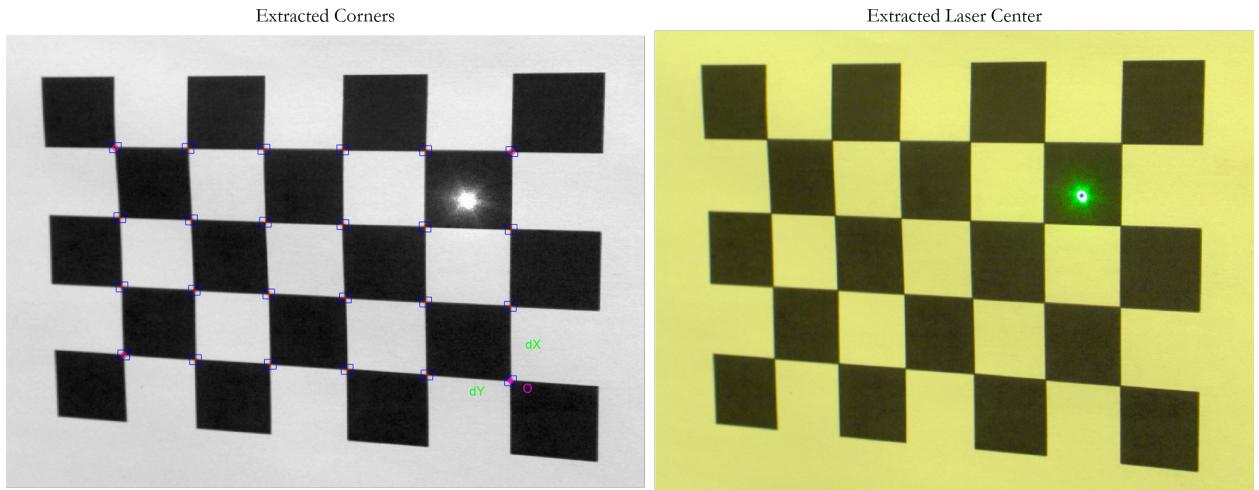


Figure 5.4: Extracted Corners and Laser Center

### 5.3.2 Navigation System Experimental Setup

Insert Text Regarding the Navigation System Experimental Setup.

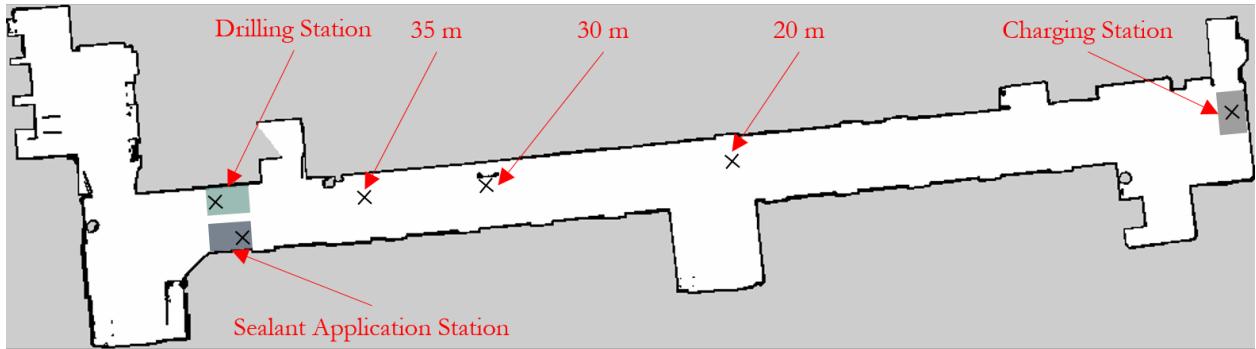


Figure 5.5: Navigation System Experimental Setup.

### 5.3.3 Drilling Operation Experimental Setup

Insert Text Regarding the Drilling Operations Experimental Setup.

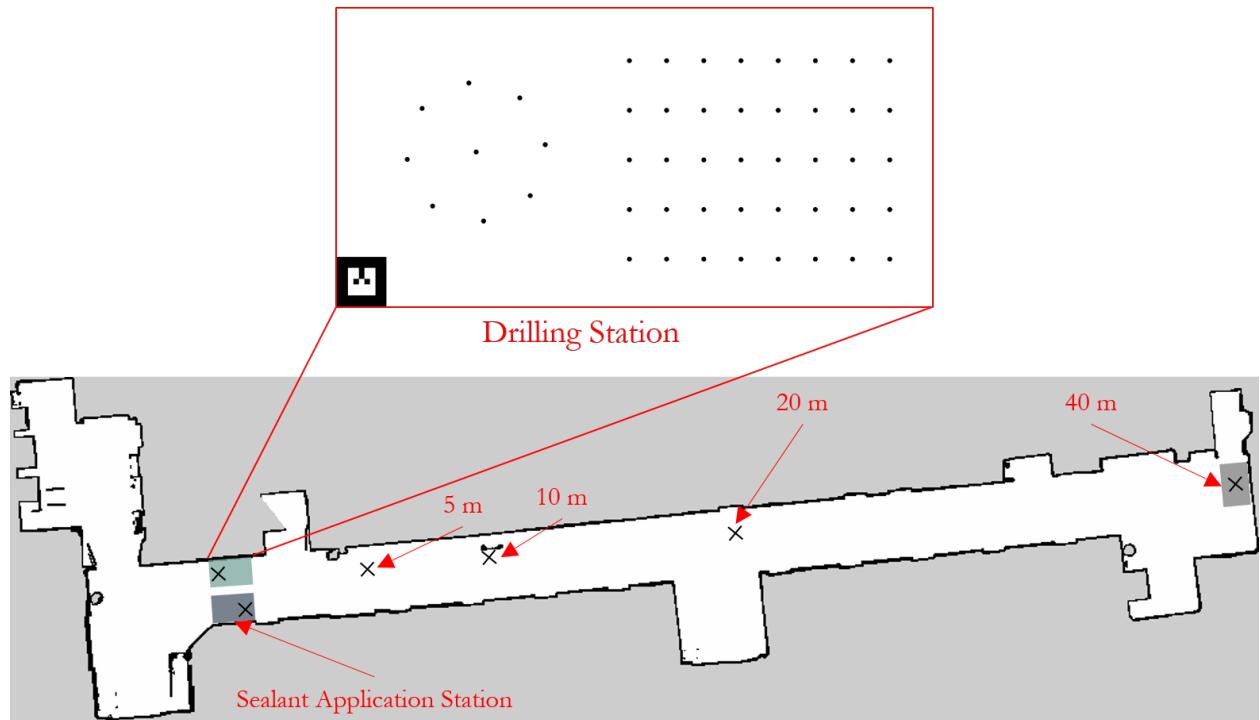


Figure 5.6: Drilling Operations Experimental Setup.

### 5.3.4 Sealant Application Experimental Setup

Insert Text Regarding the Sealant Application Experimental Setup.

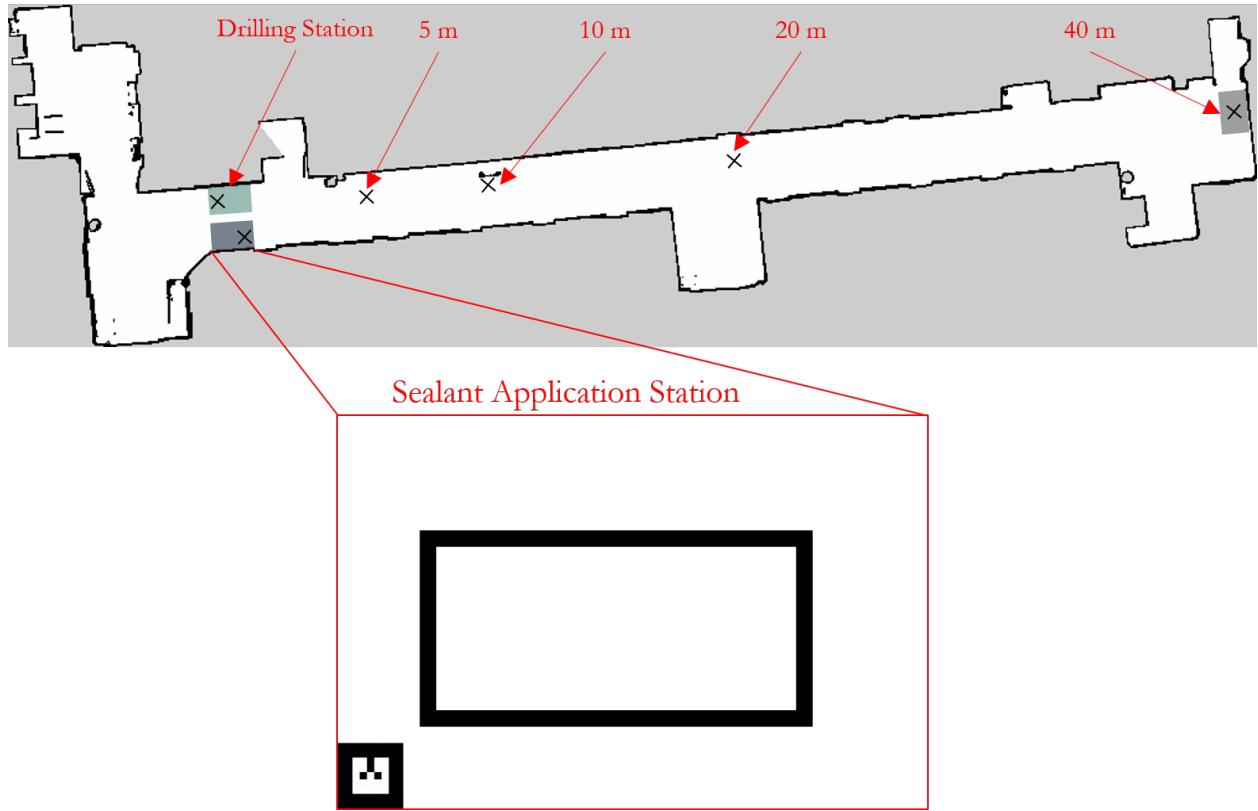


Figure 5.7: Sealant Application Experimental Setup.

## 5.4 Results

### 5.4.1 Camera Calibration Accuracy Achieved

Insert Text Regarding Pixel Error of Kinect V2.

Insert Text Regarding Pixel Error of Basler ACA150-UC.

Insert Text Regarding Pixel Error of Laser Calibration.

### 5.4.2 Navigation System Accuracy Achieved

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

Table 5.1: Global Localization Accuracy Comparison Given Distance From Global Location.

	Distance from Global Position	x (mm)	y (mm)	$\psi$ (rad)
Global Position Localization	5	0	0	0
	10	0	0	0
	20	0	0	0
	40	0	0	0

### 5.4.3 Discussion of Drilling Operation Accuracy

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

Table 5.2: Drilling Operation Accuracy Comparison Given Distance From Work Surface.

	Distance from Global Position (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

### 5.4.4 Discussion of Sealant Application Accuracy

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

Table 5.3: Sealant Application Accuracy Comparison Given Distance From Work Surface.

	Distance from Global Position (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 6

## 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  $\pm 2$  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).