RGBD Camera Pose Estimation Techniques, Slip Detection, and Occluded Object Search Strategies for Deformable Linear Object Features in Autonomous Robotic Space Task Execution

Hwang, Lawrence Jason

2024

RGBD Camera Pose Estimation Techniques, Slip Detection, and Occluded Object Search Strategies for Deformable Linear Object Features in Autonomous Robotic Space Task Execution

By

LAWRENCE JASON HWANG
THESIS

Submitted in partial satisfaction of the requirements for the degree of

MASTER OF SCIENCE

in

Computer Science

in the

OFFICE OF GRADUATE STUDIES

of the

UNIVERSITY OF CALIFORNIA

DAVIS

Approved:

Bahram Ravani, Chair

Stephen Robinson

Julian Panetta

Committee in Charge

2024

Abstract

This thesis studies Robotic handling of Deformable Linear Objects (DLO). Many habitats used for space exploration include panels with multiple wires and connections which can be easily reconfigured by humans but very difficult to be handled autonomously by robotic systems due to the flexible nature of the wires. In some situations, the wires can come loose and get separated from their connections resulting in malfunctioning of some onboard systems. This thesis develops methods for autonomous handling of flexible wires (deformable linear objects) involving the unplugging and re-plugging or stowing of one end of the wire from a connection point. An anomaly situation may arise when the end of a gripped DLO slips away from the robotic end effector into the environment while being maneuvered, entering the object into an unknown state. The objective of the research presented herein was to use purely visual sensing to detect this DLO slip locating the loose connector end, estimating its pose, and autonomously developing a motion plan for retrieval and delivery of the connector end to its originally intended destination. Three pose estimation methods are implemented: employing fiducial markers, RGBD image processing, and machine learning algorithms to generate the pose of the end of the DLO being manipulated.

Experiments are performed using two cooperating robotic arms that show identification rates of 48.1%, 100.0%, and 77.8% and arm retrieval grasp rates of 48.1%, 74.1%, and 64.0% respectively among 27 trials. The identification rate varied based on the level of occlusion of the DLO end within the workspace. Slip detection is accomplished by comparing this estimated position's distance to the manipulating arm's end effector against a threshold quantifying a slip, producing a success rate of 77.2% from 18 slip trials. In the event that the loose connector settles out of the camera's view, a spiral search pattern was designed to maneuver the secondary camera for further workspace inspection, with a search identification rate of 91.7% in 36 trials. The effectiveness of the overall system as a solution for anomaly detection and resolution is exhibited through three demonstrations with varying environmental configurations.

Acknowledgements

My journey thus far as a student, scientist, and engineer would not have at all been possible without the guidance and mentorship of Professors Bahram Ravani and Stephen Robinson. The ideas and discussions shared challenged me to critically think, shifting how I resolved problems and viewed the world around me. I wished to thank Professor Ravani for giving me chance and allowing me to represent him in the Habitats Optimized for Missions of Exploration (HOME) research lab. With this opportunity, I not only found a love for robotics, but was able to learn, grow, and contribute to a cause I was truly passionate about. I would also like to express the utmost gratitude to Professor Robinson for being my direct advisor through each stage of this research. It was an absolute honor engaging in this topic under his tutelage, where the support he gave made put every difficulty and obstacle in reach. Professors Ravani and Robinson are the problem solvers, tinkerers, and most importantly, engineers, I strive to be as I progress in my own career.

I also wanted to thank each and every professor I had the privilege to study under, deepening my knowledge and appreciation for the field of computer science. I wished to thank Professor Julian Panetta in particular for his role and contributions in my committee, as well as the efforts he made to ensure I succeeded in both my thesis and his course.

I would last like to honor my immediate family. The boundless love and support of my mother, father, and grandmother made me the person I am today. I am beyond thankful for the countless hours and financial sacrifices each of them made to provide me with a cherished upbringing and invaluable education. I hope I can continue making each of you proud with every passing day.

我真是愛你們，謝謝你們為我所做的一切。

Contents

Abstract ii

Acknowledgements........ iv

List of Equations ix

List of Figures........ x

List of Tables xiv

1 Introduction 1

1.1 Motivation 1

1.2 Project Goals and Scope 1

1.3 Thesis Structure 4

2 Background 5

2.1 Cameras and Image Processing 5

2.1.1 Sensor and Camera Types 5

2.1.2 Camera Placement in the Robotic Environment 8

2.1.3 Morphological Operations 9

2.1.4 RGB and Depth Alignment 11

2.2 Identification Techniques 13

2.2.1 Fiducial Markers 13

2.2.2 Machine Learning Image Recognition 15

2.3 Robotic Task Execution 17

2.3.1 Sampling-Based Motion Planning 17

2.3.2 Coordinate Frames and Transformations 19

2.4 Related Work 21

2.4.1 Image Sensor Based Object Recognition 21

2.4.2 Occlusion Search 23

2.4.3 Summary 26

3 Methods 27

3.1 System Overview 27

3.1.1 Configuration 27

3.1.2 Software Overview 31

3.1.3 System Requirements and Assumptions 32

3.2 Identification and State Estimation of Connector 34

3.2.1 Pose Estimation and Techniques Overview 34

3.2.2 Fiducial Markers 35

3.2.2.1 Camera Calibration 35

3.2.2.2 Fiducial Marker Registration 37

3.2.2.3 Fiducial Marker Identification 38

3.2.2.4 Pose Estimation 39

3.2.2.5 Representative Frame Adjustments 40

3.2.2.6 Fiducial Marker Visual Servoing Pipeline 41

3.2.3 RGBD Image Processing 42

3.2.3.1 RGBD Image Segmentation for Position 42

3.2.3.2 Point Cloud Processing for Orientation 47

3.2.3.3 Pose Estimation 49

3.2.3.4 RGBD Image Processing Visual Servoing Pipeline 50

3.2.4 Machine Learning 51

3.2.4.1 Image Dataset Configuration 51

3.2.4.2 Convolutional Neural Network Training 53

3.2.4.3 Pose Estimation 55

3.2.4.4 Machine Learning Visual Servoing Pipeline 57

3.3 Management of Robotic Agent 59

3.3.1 Motion Planning and Execution Architecture 59

3.3.2 Robotic Arm Task Allocation 59

3.4 Slip Anomaly Detection and Response 60

3.4.1 Anomaly Monitoring, Quantification, and Detection 60

3.4.2 System Control Flow for Off-nominal Slip Response 62

3.4.3 Environment Spiral Search Pattern 67

3.4.3.1 Search Area Optimization 67

3.4.3.2 Spiral Search Strategy 69

3.4.3.3 Software Implementation 72

4 Results **74**

4.1 Techniques Validation 74

4.1.1 Identification and State Estimation 74

4.1.1.1 Experiment Overview 74

4.1.1.2 Fiducial Markers Results 79

4.1.1.3 RGBD Image Processing Results 81

4.1.1.4 Machine Learning Results 83

4.1.1.5 Identification and State Estimation Results Summary 85

4.1.2 Slip Detection 89

4.1.2.1 Experiment Overview 89

4.1.2.2 Slip Detection Results 91

4.1.3 Environment Spiral Search Pattern 93

4.1.3.1 Experiment Overview 93

4.1.3.2 Search Pattern Results 95

4.2 Full Scenario Demonstration 99

4.2.1 Demonstration Overview 99

4.2.2 Results and Summary 103

4.3 Discussion 103

4.3.1 Identification and State Estimation 103

4.3.1.1 Fiducial Markers 102

4.3.1.2 RGBD Image Processing 105

4.3.1.3 Machine Learning 108

4.3.1.4 Identification and State Estimation Summary 109

4.3.2 Slip Detection 111

4.3.3 Environment Spiral Search Pattern 113

4.3.4 Full Scenario Demonstration 116

5 Conclusion and Future Work 117

5.1 Conclusion 117

5.2 Future Work 120

5.2.1 Identification and State Estimation 120

5.2.2 Slip Detection 122

5.2.3 Environment Spiral Search Pattern 123

List of Equations

1 Depth from image disparities [11] 8

2 Moving window threshold [17] 10

3 Convolution over a binary image [16] 10

4 2D point in 3D given camera parameters [19] 12

5 Transformation between corresponding 2D points to shared 3D point [7] 13

6 Three-dimensional Euclidean distance formula [55] 61

List of Figures

1 Visualization of global and rolling shutter approaches [9] 7

2 Example structuring elements with respective matrix implementations [7] 10

3 Morphological operations depicting dilation (top) and erosion (bottom) [15] 11

4 Alignment relationship for a point P between an RGB and depth image [19] 12

5 Examples of fiducial markers [21] 14

6 Visualization of relationship between FRE, FLE, and TRE with respect to known and measured points. f_0, f_1 and d_0, d_1 represent the RMS distance of the fiducials and the distance of the observation from the principal axis of the fiducial configuration [25] 15

7 Convolutional neural network architecture featuring input, convolutional, pooling, fully-connected, and softmax layers [27] 17

8 Sampling-based roadmap featuring k = 2 edges for each node [30] 18

9 Sampling-based Rapidly-exploring Random Tree graph output featuring a root node and thirteen sample nodes [30] 19

10 Example of a pure translation between a world, parent, and child frame [35] 21

11 Example of fiducial marker guided visual servoing; insertion of a wooden panel at pose estimated from ArUco tag [39] 23

12 Estimation of deformable linear object position under partial occlusion via the method developed by Chi et al. (2019) [41] 24

13 Actual (left) and simulation (right) views of the robotic system in environment, with two visual sensors mounted behind and on the left arm's wrist joint 28

14 Designed short (left) and tall (right) foam environment obstacles with dimensions 29

15 Mock DLO cable (top) and battery ORU (bottom) with dimensions 30

16 Primary types of camera lens distortions: (a) No distortion (b) Barrel distortion (c) Pincushion distortion (d) Mustache distortion [46] 35

17 Checkerboard with OpenCV calibration lines drawn over the RGB image 37

18 Pose estimation of two fiducial markers in workspace, generated by OpenCV 39

19 Rviz simulation view of fiducial marker frames and offset grasping frame 41

20 Fiducial marker guided visual servoing pipeline 42

21 Loose cable connector point cloud (white) in Rviz simulation view of environment 45

22 Generated frame for connector with position given by averaged center point 46

23 Direction vector (red) between the averaged minimum and maximum points of the connector point cloud 48

24 Connector frame generated from HSV segmented point cloud 50

25 RGBD image segmentation point cloud guided visual servoing pipeline 51

26 Bounding box annotated (yellow) RGB images of cable with connector end in environment (left) and its augmented copy (right) 52

27 Augmentations applied to the second dataset for training YoloV5 model 54

28 Generated frame (left) for bounding box over identified connector (right) 57

29 Machine learning and depth image guided visual servoing pipeline 58

30 System architecture for robotic agent motion planning and execution 59

31 3D Euclidean distance measurement (green) between connector end and grasping arm's end effector in nominal (left) and off-nominal (right) operation 62

32 Visualization of noise affecting connector end pose estimation (RGBD image processing method), set one second apart from the same viewing angle 64

33 ROS transformation tree depicting parent child relationships between entities 65

34 System architecture for slip anomaly response with fiducial markers 66

35 Behavior tree representing states and actions of this robotic system 67

36 Cone representing the most probable region a loose connector end may settle 69

37 Spiral search path with primary stopping points (dots), initiated from the connector's last known location (star) 70

38 Spiral search path with eight stopping subpoints (green dots), and direction vectors (green arrows) denoting end effector orientation at each subpoint 71

39 Searching arm at two stages of the spiral search pattern: primary point with flat orientation (left) and subpoint oriented towards the central primary point (right) 72

40 Search frame at its initial pose, positioned at the connector end’s last known location with zeroed orientation 73

41 Pose estimation configurations for technique validation experiment 76

42 Measurement procedure for determining connector end ground truth X, Y, Z relative to system world frame 77

43 RViz simulation view of the generated ground truth for DLO Configuration 3 78

44 RViz simulation view of the configuration 3 ground truth (left), fiducial marker pose estimation frame (center), and both frames together (right) 81

45 RViz simulation view of the configuration 3 ground truth (left), RGBD image pose estimation frame (center), and both frames together (right) 83

46 RViz simulation view of the configuration 3 ground truth (left), machine learning pose estimation frame (center), and both frames together (right) 85

47 Identification success and failure count per trial by pose estimation technique 86

48 Grasp success and failure count per trial by pose estimation technique 86

49 Compiled grasp success rates between fiducial markers and the shared RGBD and ML orientation estimation approach 87

50 Ground truth distance error per trial by pose estimation technique 88

51 Right manipulating robotic arm grasping DLO end at each of the six slipping points 90

52 Slip detection success and failure count by trial drop height 92

53 Cause of failure breakdown for slip detection failed trials 93

54 Spiral search initiated from the DLO connector's last known position (left) and workspace origin (right) 94

55 Spiral search trial success and failure count, by initial position 97

56 Elapsed time to find DLO connector end per subset trial, by initial position 98

57 Distribution of successful identifications by spiral search pattern subpoint 99

58 Outline of all stages composing the full system demonstration 102

59 Noisy segmented point clouds for a nearly horizontal connector pose, captured 3 seconds apart 108

60 Grasped (left) and empty (right) readings from a force sensor on the end effector of a UR5e robotic arm 123

List of Tables

1 Robotic Sensor Variety Summary [8] 6

2 Evaluation of model version performances under mAP, precision, and recall 55

3 ArUco fiducial marker pose estimation identification/grasp success and ground truth error per trial, by DLO configuration 80

4 RGBD image processing pose estimation identification/grasp success and ground truth error per trial, by DLO configuration 82

5 Machine learning pose estimation identification/grasp success and ground truth error per trial, by DLO configuration 84

6 Ground truth mean distance error, variance, and standard deviation per pose estimation technique 88

7 Slip detection success per trial, by DLO configuration and slip position 91

8 Spiral search success, time, and point per trial, by occluding obstacle and start position . 95

1 Introduction

1.1 Motivation

As humanity continues to expand its presence and exploration of space, self-reliance in spacecraft design becomes exponentially vital. NASA's current objectives detail aspirations of human spaceflight into deep space, of which the Moon and Mars are included destinations [1]. Journeying into deep space demands heightened consideration in how crews are to be supported, as they entail missions longer in duration increasingly further from planet Earth. Historically, as seen in the operation of the International Space Station in Low Earth Orbit, teams aboard are highly dependent on resupplies, live support, and communication from mission control located on Earth's surface [2], [3], [4]. Meanwhile, necessary onboard tasks, such as repair, maintenance, and inspection operations, are conducted continuously by onboard crews. For long-duration expeditions into deep space, these affordances are made null as increasingly greater distances from Earth delay real-time communication and ground support. Spacecraft may also operate (at least for some period of time) without any crew members, resulting in no viable manpower to perform necessary tasks for extended periods of time. Therefore, missions into deep space require a self-sufficient system comparatively free of dependence on Earth.

A proposed solution is the SmartHab, a self-aware and self-sufficient habitat capable of supporting a crew when present and sustaining itself when not [2]. At any time, these spacecrafts may be in a number of situations, including in orbit, on a surface, or in transit between destinations. Despite its present situation, the SmartHab must be able to meet the biological needs of astronauts onboard, as well as needs of the spacecraft itself, such as repair or maintenance procedures. An integrated Environmental Control and Life Support System (ECLSS), capable of providing clean water and air to space crews, assists in accommodating the former [5]. The latter, traditionally handled by humans aboard a spacecraft, may be viably supplemented by a or multiple robotic agents. Menial tasks, not worth an astronaut's invaluable time, are prime for such a substitution, especially if those demands are within the scope of a robotic arm's capabilities. Assuming the interior of such a SmartHab borrows in design from the ISS, necessary work may need to be performed on arrays of tightly compacted subsystems, organized in such a way that maximizes the utilization of volume available in the spacecraft [6], [7]. These subsystems may also feature an assortment of DLOs that are necessary to that module's functionality. In the event that a maintenance or repair task for such a subsystem is necessary, a robotic system onboard the SmartHab must be able to manipulate the DLOs. In order to manipulate such objects, however, the system first needs to have the ability to track and monitor them, informing the agent of the entity's present state. Additionally, anomaly events may occur during the execution of a task, further requiring a robotic agent detect and appropriately respond to any off-nominal situation. A SmartHab, meeting self-sufficiency standards through the utilization of autonomous systems, requires the ability to satisfy each of these specifications.

1.2 Project Goals and Scope

Among the range of subsystems on the SmartHab are Battery Orbital Replacement Units (ORUs), sources of distributable energy. Over the course of their operation, these battery ORUs become degraded and require substitution. Before the unit can undergo replacement, the DLO connected to its face must be safely unplugged and stowed. Assuming a robotic agent, composed of two arms bearing gripper end effectors, onboard the SmartHab possesses this capacity, an anomaly event may occur where that recently unplugged DLO end slips from the manipulating arm's grasp enroute to a designated stowing point. At any point, the system requires a reliable and robust method of estimating the connector end's pose, as to inform itself of the object's state. The autonomous system must then recognize that the DLO, an electrical cable in this scenario, which is no longer held by the grasping arm's gripper. Once flagging the off-nominal slipped status of the cable, its unknown location must be found via onboard system technology. A plan for retrieval of the cable connector end is calculated and executed before completing a final trajectory that restores the cable to its original destination.

The research presented in this thesis aims to devise, implement, and demonstrate a zero-gravity robotic solution to resolving this DLO slip anomaly for the purposes of SmartHab autonomy and self-sufficiency. In order to do so, capabilities in pose estimation, object state monitoring, gripper end effector slip detection, and workspace search are developed such that a DLO under manipulation is restored to a known state if entering this off-nominal situation. By nature of the robotic agent's intended use case, the developed solution targets autonomy, requiring no user input or initial conditions except basic attributes of the DLO connector end supplied to the system.

With the capabilities achieved, attention is next dedicated to system robustness. Maximizing the repeatability and success of task execution in the presence of low accuracy and noise is crucial in a live workspace, where spacecraft misalignments or other environmental conditions may severely alter performance compared to a sterile testing environment. The robotic agent must be capable of performing its task without human assistance or correction, no matter the circumstances of its workspace. Variations in lighting, primarily, introduces noise in the form of reduced RGB and depth image quality, posing challenges to a system employing visual sensors. Additionally, measurements of entities may not be as accurate as the system's knowledge, rendering finely tuned robotic maneuvers unreliable in interacting with the workspace. The degree to which requirements and tolerances may be relaxed therefore contributes significantly to robustness in system capabilities, further cementing the system's qualifications in autonomy.

1.3 Thesis Structure

Following this introduction, the proceeding thesis is divided into the following sections:

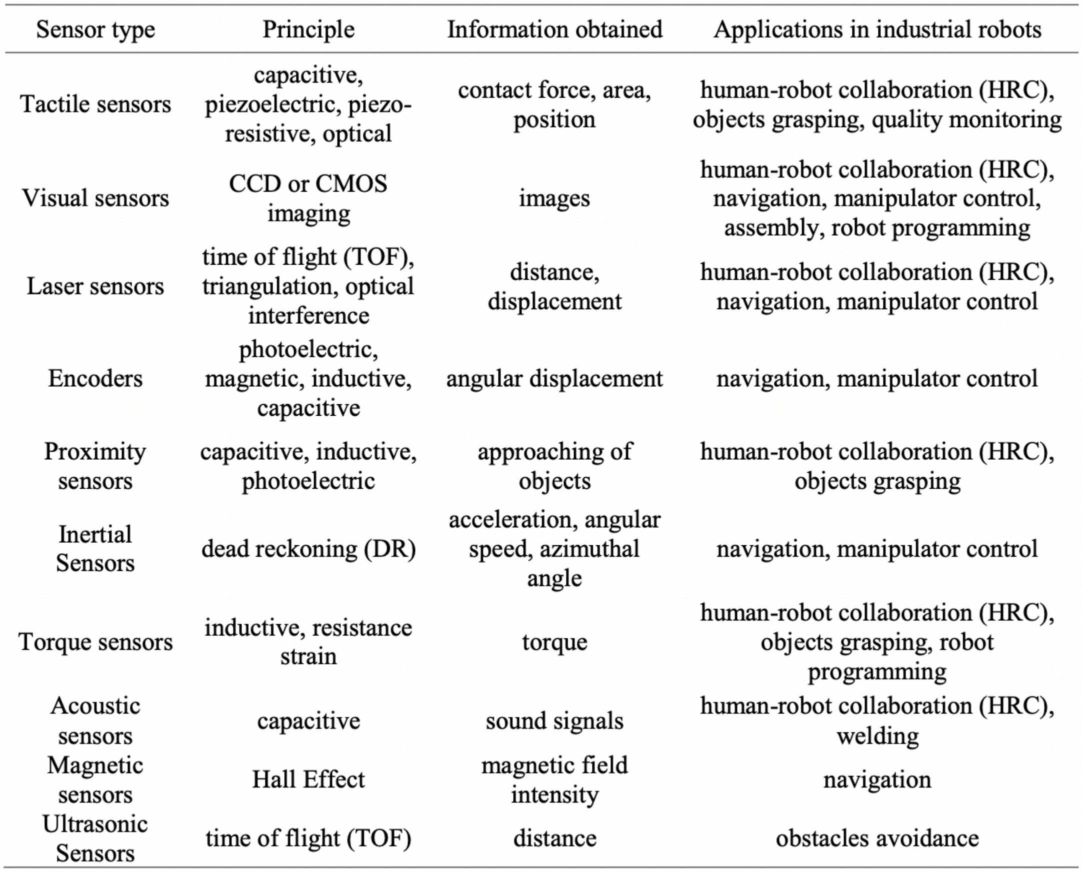
- Background: providing critical explanations of the theories and concepts underlying the capabilities developed in this work. A literature review of related work is also provided, showcasing the state of similar research. Approaches in solving related problems are highlighted to demonstrate resolutions to adjacent problems, as well as their advantages and disadvantages.
- Methods: detailing this work's implemented solutions to both connector end pose estimation and anomaly response, composed of slip detection and occluded environment search. This section delves into the underlying architecture and technologies behind the capabilities enabling SmartHab autonomy.
- Results: a summary of the experimental results, demonstrations, and findings made in this thesis. The process of experimentation, accuracy of pose estimation methods, and the developed system's proficiency in resolving the proposed research problem are presented and discussed.
- Conclusion and Future Work: an overview of the research conducted as well as avenues for where this work could be expanded upon or improved.

2 Background

2.1 Cameras and Image Processing

2.1.1 Sensor and Camera Types

The major senses, such as sight or touch, serve as the human body's bridge to perceiving and understanding the physical world. Much in the same vein, robotic systems have a varied suite of sensor options that enable them to absorb information about their environment. Four of the major categories of sensing systems employed in modern robotics include tactile, visual, laser, and encoder, while other miscellaneous types of external sensors used include proximity, inertial, force/torque, acoustic, magnetic, and ultrasonic sensors [8]. Each sensor collects data in various forms for the system, which must be processed to glean valuable information. In Table 1 below, these varieties are listed and broken down by type, principle, information obtained, and common utilizations in robotic applications.



Sensor type	Principle	Information obtained	Applications in industrial robots
Tactile sensors	capacitive, piezoelectric, piezo-resistive, optical	contact force, area, position	human-robot collaboration (HRC), objects grasping, quality monitoring
Visual sensors	CCD or CMOS imaging	images	human-robot collaboration (HRC), navigation, manipulator control, assembly, robot programming
Laser sensors	time of flight (TOF), triangulation, optical interference	distance, displacement	human-robot collaboration (HRC), navigation, manipulator control
Encoders	photoelectric, magnetic, inductive, capacitive	angular displacement	navigation, manipulator control
Proximity sensors	capacitive, inductive, photoelectric	approaching of objects	human-robot collaboration (HRC), objects grasping
Inertial Sensors	dead reckoning (DR)	acceleration, angular speed, azimuthal angle	navigation, manipulator control
Torque sensors	inductive, resistance strain	torque	human-robot collaboration (HRC), objects grasping, robot programming
Acoustic sensors	capacitive	sound signals	human-robot collaboration (HRC), welding
Magnetic sensors	Hall Effect	magnetic field intensity	navigation
Ultrasonic Sensors	time of flight (TOF)	distance	obstacles avoidance

Table 1: Robotic Sensor Variety Summary [8]

For the purposes of object detection in this body of work, visual sensors in the form of stereo cameras, offering RGB color image and depth distance data, are utilized. Other varieties of sensors were considered, but either yielded unnecessary information (e.g. acceleration via inertial sensors) or were not plausible due to hardware constraints (e.g. lack of tactile sensors for arm end effectors).

While several techniques for capturing RGB images exist, many modern cameras utilize the rolling shutter approach. Rolling shutter cameras compose images not by taking a single snapshot of a scene, as done in the more traditional pinhole camera model, but rather by scanning along directions of the scene (i.e. vertically, horizontally, or rotationally) and building a composite image from each scan's returned data [9]. Figure 1 illustrates a comparison of the approaches, depicting the stage of capture in relation to time for both global and rolling shutter. Assuming proper distortion correction and image compilation, the primary advantage of rolling shutter over other conventional capture is improved accuracy via rapid updates. The continual scanning of this technique introduces a constant influx of new scene data, making completed images available for processing.

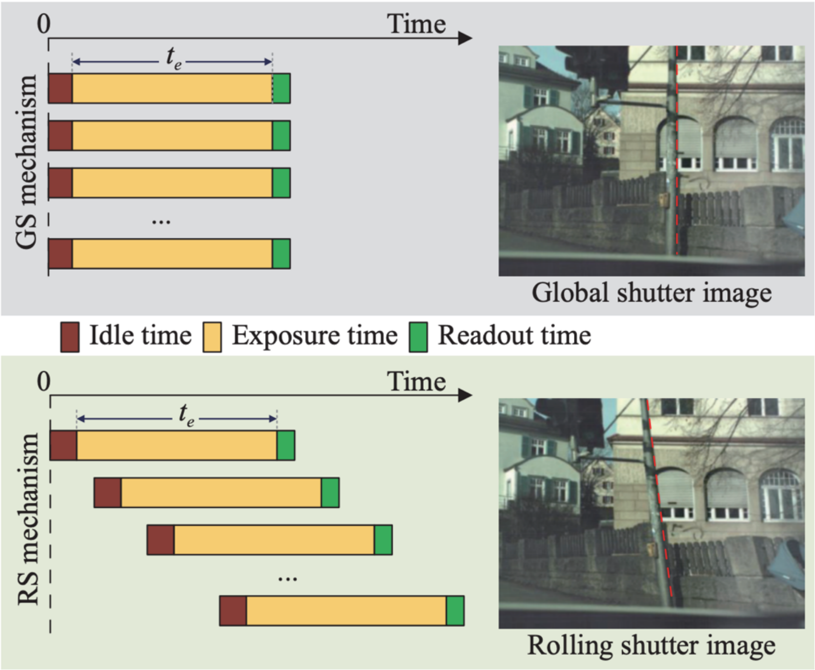


Figure 1: Visualization of global and rolling shutter approaches [9]

Depth cameras offer spatial distance information between the position of the sensor and any objects located in its field of view (FOV). Stereoscopic depth imaging is one approach that determines depth by capturing two, slightly offset two-dimensional RGB images [10]. Positional differences between features in the two images are measured and processed to produce a final depth image. The relationship of depth (z) to measured disparities (d) is represented in Equation 1, with respect to focal length of the imaging sensor (f) in pixels and baseline between the offset capturing lenses (B) in the desired depth units, typically meters or millimeters [11].

$$z = \frac{f \cdot B}{d} \tag{1}$$

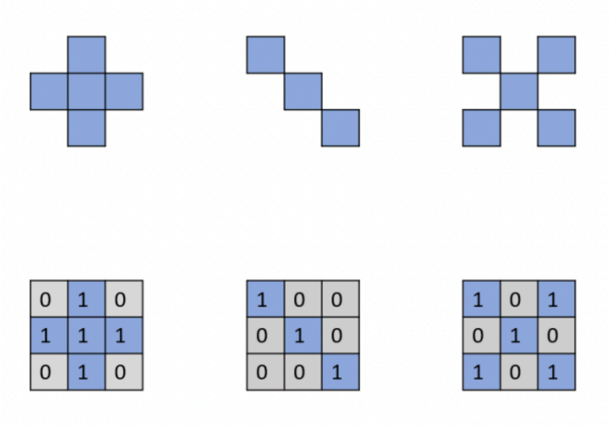
2.1.2 Camera Placement in the Robotic Environment

When employing visual cameras as the primary sensing mechanism for a robotic system, consideration must be placed in determining their optimal placement that yields the most rapid and efficient collection of information from the workspace. Cameras have options not only in how they are oriented, but where they are positioned in the operational environment. Cameras can be mounted on the arm itself, along any of the joint parts or on the end effector, opening up the possibility of dynamic image capture alongside the robotic agent. Alternatively, a camera or sensor can be placed in a static pose (position and orientation) somewhere in the environment. For each camera used in a system, its respective pose should either maximize coverage of the environment or provide effective time monitoring of significant entities.

For static camera sensors, one approach to determining placements is by repeating a desired task and varying sensor locations for each trial such that observability is maximized [12]. Defining a heuristic in this iterative approach offers a systematic method of determining the most optimal camera configuration, demarcated by the highest scoring sensor configuration. This heuristic could be optimized to maximize view of either scene or an object of interest. The physical environment in which the robotic system operates in, constrains potential sensor configurations, where tangible surfaces must be available for cameras to be placed. Pose options would, for instance, be a lot more limited in open environments such as outer space extravehicular activities, underwater ocean tasks, or wide outdoor fields in industrial farming applications. In such instances, determining valid “optimal camera sites” over an area of interest and maximizing their coverage is desirable [13]. Dynamic camera sensors, such as ones placed on a mobile robotic agent, comparatively offer greater flexibility in placement. Due to the degrees of freedom afforded by a robot, the camera can be simply moved to the most optimal view of the work environment [14]. It is critical, however, to capitalize on this fact by placing the sensor in a pose that benefits from the movement of the robotic agent, such as on an arm's end effector.

2.1.3 Morphological Operations

For a robotic agent to utilize information from a visual sensor, camera output in the form of images must first be processed in order for relevant information extraction. Due to natural noise introduced by imperfect sensors, these processes enhance images to better extract information from collected data [15]. Morphological operations are computer vision techniques that processes images by applying a defined shape, in the form of a structuring element, over a source image [16]. The two (the shape template and the image) are combined via convolution, a mathematical operation that receives as inputs two matrices of compatible dimensionality and outputs a resulting matrix [16]. The two matrices are pictures p, which can be represented as matrices in the form of binary images, and a structuring element as matrix convolution kernel t [7]. Illustrated in Figure 2 are these convolution kernels in matrix format, and their corresponding shapes.



$$\begin{bmatrix} 0 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 0 \end{bmatrix} \quad \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix}$$

Figure 2: Example structuring elements with respective matrix implementations [7]

For any submatrix within the image *p*, a thresholding operation can be applied as follows in Equation 2 [17]:

$$threshold(p,t) = \begin{cases} 1 & p \geq t \\ 0 & else \end{cases} \tag{2}$$

Morphological operations are commonly applied to data in the form of binary images as, by nature, their numerical structure lends itself to processing with this method [18]. The structuring element *s*, calculated for each cell by the threshold, acts as a moving window over the binary image during convolution *c*, calculating the number of ones *S* that overlay with the structuring (size of the structuring) presented in Equation 3 [7], [16].

$$c = f \circledast s \tag{3}$$

Dilation and Erosion are two such morphological operations that can be applied to a binary image [17]. Dilation expands a binary image by extending its shape, defined at a window as whether the number of ones in the binary image that overlap with the structuring element is greater than or equal to the number of ones, then the pixel window's corresponding pixel in the image that overlaps with the center of the structuring element will be set to one [16]. Erosion alternatively shrinks a binary image by reducing its shape, turning the image pixel corresponding to the structuring element's origin to zero if all ones in that window of the image do not overlap with the structuring element [16]. The effects of both morphological operations are depicted in Figure 3.

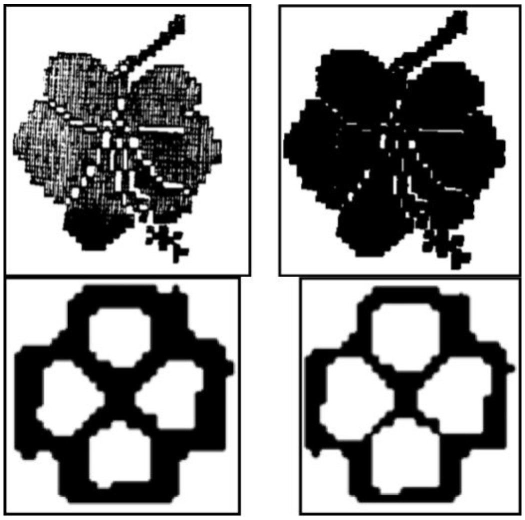


Figure 3: Morphological operations depicting dilation (top) and erosion (bottom) [15]

2.1.4 RGB and Depth Alignment

A stereo camera system, capable of capturing both the RGB and depth image from two separate RGB and infrared sensors respectively, are inherently located at different positions due to the physical space occupied by each lens [11]. In the context of a robotic system, this results in different frames for each lens, as well as different source points of reference between the RGB and depth image views of the same scene. To remedy this hardware constraint, the images taken from the separate depth and RGB lenses can be aligned via a mapping calculation to place them in a shared coordinate system [19]. For each sensor in the stereo camera, a 3D point from the scene can be mapped to a corresponding point on a planar, 2D pixel coordinate system, with respect to each camera's internal parameters [7][19]. A point *P* in either the RGB or depth image is specified as *(u, v)*, with *(u,v)* representing the pixel coordinates of point *P*. Its corresponding point is represented as *(X, Y, Z)*, in 3D space with *(X, Y, Z)* representing pixel coordinates. The transformation between these two representations is given in Equation 4 as follows [19]:

$$Z\begin{bmatrix}u\\v\\1\end{bmatrix}=\begin{bmatrix}f_x & 0 & u_0\\0 & f_y & u_1\\0 & 0 & 1\end{bmatrix}\begin{bmatrix}X\\Y\\Z\end{bmatrix} \tag{4}$$

In this equation, f_x and f_y are the focal lengths of the respective lens in pixel coordinate units, u_0 and u_1, the principal point of the image, and *Z* is the transpose between the two coordinate systems [7][19]. With external parameters stored in a matrix *M* determined for a specific camera sensor, an alignment relationship between the RGB and depth lenses of that sensor can be formed as illustrated in Figure 4.

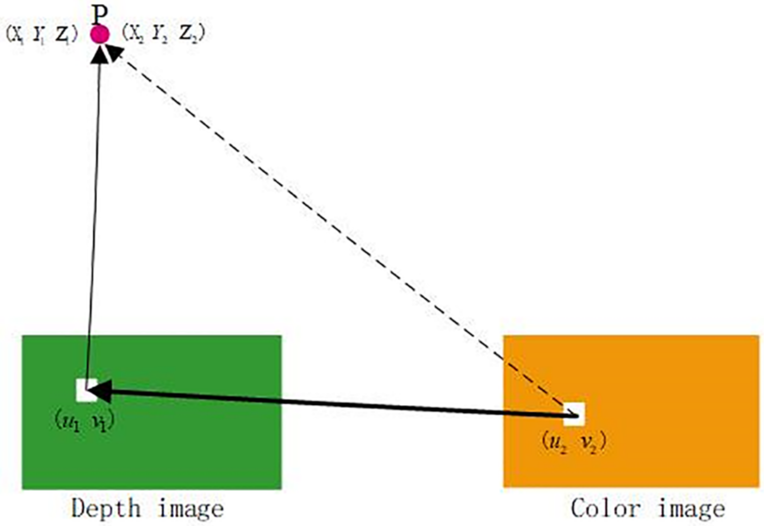


Figure 4: Alignment relationship for a point P between an RGB and depth image [19]

For two points (X_1, Y_1, Z_1) and (X_2, Y_2, Z_2), the calculated 3D points for the corresponding points in the depth and color images, the full transformation between the two 2D coordinate system points can be found using Equation 5 by solving for *R*, the rotation matrix, and *t*, the translation vector [7]:

$$\begin{bmatrix} X1 \\ Y1 \\ Z1 \\ 1 \end{bmatrix} = M \begin{bmatrix} X2 \\ Y2 \\ Z2 \\ 1 \end{bmatrix} = \begin{bmatrix} R & t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} X2 \\ Y2 \\ Z2 \\ 1 \end{bmatrix} \tag{5}$$

2.2 Identification Techniques

2.2.1 Fiducial Markers

Visual sensors enable a robotic system to optically gather information about its environment in the form of images. If these images are two-dimensional RGBs, the system must translate what is captured in the image to usable information in the 3D space. The use of fiducial markers is one such approach. Fiducial markers are shapes bearing unique patterns that once identified in an image by a scanning algorithm, offer position and orientation information about the marker [20]. These can be affixed to entities in the scene, enabling pose estimation by serving as visual landmarks when processing input images. Figure 5 depicts several examples of commonly used fiducial markers.

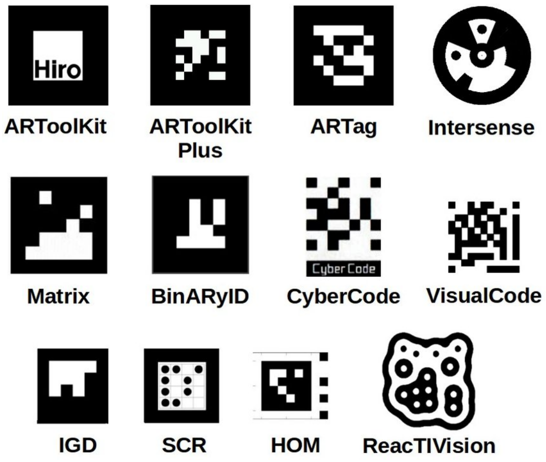


Figure 5: Examples of fiducial markers [21]

Pose estimation with a camera and fiducial markers is achieved by determining the spatial relationship between the two. A system is first given information about the marker to be searched for. Such identification include pattern, size, and units of measurement [22]. By analyzing the size and pattern of a fiducial marker identified in an image, a scanning algorithm is able to estimate the relative distance, orientation, and distortions with respect to the camera [23]. This process is a form of optical tracking and can be categorized as a registration problem between homologous measurements of the marker and its pattern, also known as an Orthogonal Procrustes Analysis problem (OPA) [24].

OPA facilitates pose estimation of fiducial markers by conducting a point-to-point registration between the known pattern and shape geometry of the marker and those measured by the visual sensor [25]. Pose estimation is given by the best alignment between the known x_i and observed y_i sets of points, found by solving the point-to-point registration problem for the rotation matrix R and translation vector t [25]. Measuring errors in the form of fiducial localization error (FLE), fiducial registration error (FRE), and target registration error (TRE) refine the pose estimation FLE represents the distance between an observed point and the unknown true location of the point prior to computing the registration transformation [25]. FRE is simply the root-mean-squared (RMS) distance between x_i and y_i following registration [25]. Lastly, the calculated distance between a point, not used in the registration calculation, to its corresponding point when the registration is applied is the TRE [25]. The relationship between the points, error varieties, rotations, and translations following the completed point registration is visualized in Figure 6.

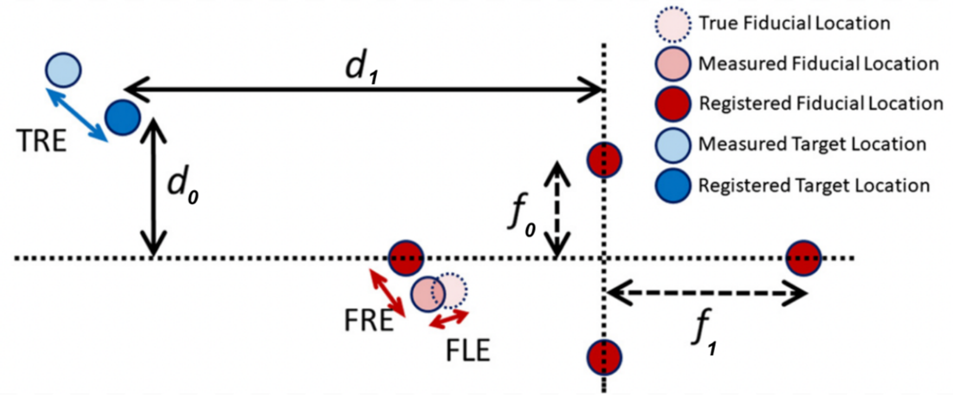

True Fiducial Location
Measured Fiducial Location
Registered Fiducial Location
Measured Target Location
Registered Target Location
d_1
TRE
d_0
f_0
FRE
FLE
f_1

Figure 6: Visualization of relationship between FRE, FLE, and TRE with respect to known and measured points. f_0, f_1 and d_0, d_1 represent the RMS distance of the fiducials and the distance of the observation from the principal axis of the fiducial configuration [25]

2.2.2 Machine Learning Image Recognition

Given a digital RGB image, image recognition by a machine learning (ML) algorithm is the process in which said algorithm successfully identifies a subject of interest in the image by detecting common features attributed to that entity. A common approach to implementing ML object detection is by reducing it to a classification problem and training a convolutional neural network (CNN) to perform the task [26]. A CNN is a derivation of the standard deep neural network (DNN), a structured neural architecture consisting of a series of interconnected layers, wherein each layer is composed of neuron nodes that accept an input and return some output [27]. Convolution, or cross-correlation, simply improves filtering and requires fewer nodes for classification by regularizing weights over those nodes [28].

Beginning from a CNN's initial input layer, sections of the image are filtered by a convolution layer for lines, edges, curves, colors, or any other identifying pattern of pixels that could be correlated to what is to be identified [26]. Often between the convolutional layers are supporting pooling layers, responsible for reducing the height and width of an input, resulting in the reduction of computation and avoidance of overfitting [27]. One or more rounds of convolution and pooling complete feature extraction, passing those on to fully-connected layers for classification. A fully-connected layer receives the features in a the form of a flattened one-dimensional vector, where a set of neuron nodes bearing full connection to all nodes in the previous layer, begin making decisions [27]. These decisions are attempts at classification, made via weights and biases, which are initially randomized if not pretrained. Estimates by the model are compared to solutions provided in the training dataset, demarcating the location of a feature if present in the image. A loss value is calculated depending on the accuracy of the model's guess to the correct solution, beginning the fine-tuning process of back propagation in which the model adjusts weights and biases of layers to better predict on subsequent images [29]. A softmax layer (or function) is finally introduced following the fully-connected layers to produce a probability distribution classifying the likelihood that a feature belongs to a given class label [27]. These estimates in the form of probabilities denotes the algorithm's perceived percent likelihood that a feature is both present and located at the estimated position. Illustrated in Figure 7 is the architecture of a standard CNN, featuring the input, convolutional, pooling, fully-connected, and softmax layers resulting in a CNN's estimate.

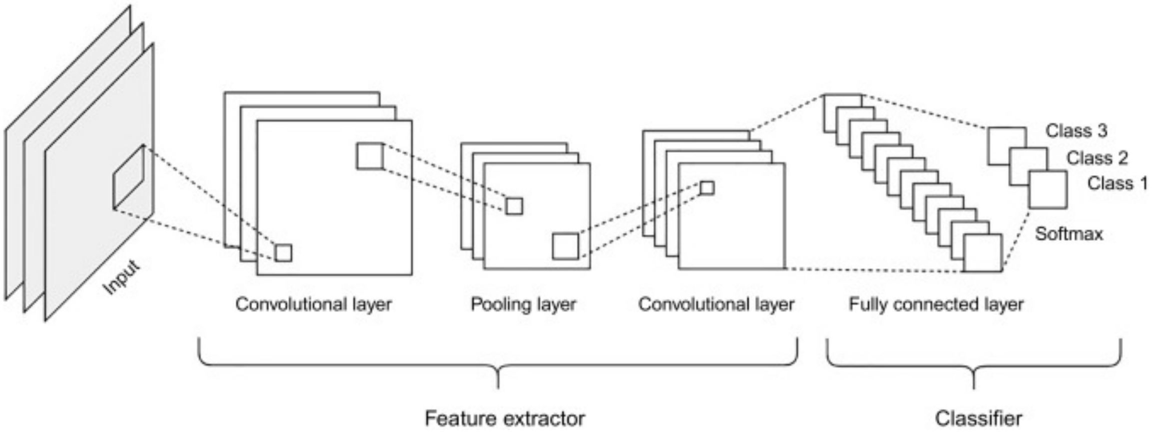


Figure 7: Convolutional neural network architecture featuring input, convolutional, pooling, fully-connected, and softmax layers [27]

2.3 Robotic Task Execution

2.3.1 Sampling-Based Motion Planning

As entities in a three-dimensional world, robotic agents have the capability of enacting physical change in its environment. To make any meaningful impact, a robot must move to target configurations from an initial starting state. A configuration is a complete specification of the positions of every point on the robot, with all possible configurations making up a robot's configuration space, or all achievable movement states [30]. Deciding what sequence of movements to execute to continuously traverse between configurations is the responsibility of motion planners. In robotics, motion planning is the process of dividing a desired movement trajectory into individual, discrete motions while satisfying any given constraints on that movement, such as avoiding obstacles in its path [31]. Motion planners employ a variety of techniques in deciding which specific movements to make. The standard categories of motion planning are sampling, optimal node, mathematic model, bioinspired, and multifusion based algorithms [32]. The Open Motion Planning Library (OMPL) is the motion planning software library utilized in this body of work, which utilizes the first category of algorithm.

Sampling-based algorithms are classically split into two categories: roadmap and tree-based planners. In the first approach, a probabilistic roadmap in the form of a graph is formulated during an initial learning phase [33]. Each node in that graph is sampled, or randomly generated, and represent an accessible robot configuration, free from collisions [33]. Next begins the query phase, where edges are iteratively added by attempting to link configuration nodes to their neighbors, successful in the event that a collision free trajectory exists between any two neighboring vertices [33]. Given an initial and target end configuration, a motion plan can be found by first adding them as nodes in the roadmap. Once edges are drawn for each node to their *k* nearest neighbors, a graph search algorithm is called on the roadmap to find an optimal traversal between the start and goal nodes [30]. Figure 8 visualizes a potential roadmap for a configuration space, where an attempted *k* = 2 edges are drawn for each vertex.

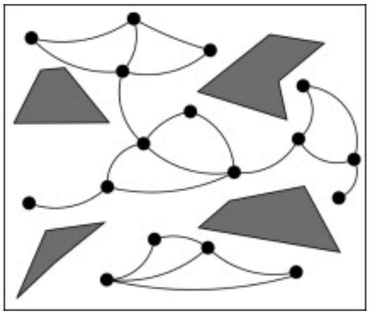


Figure 8: Sampling-based roadmap featuring k = 2 edges for each node [30]

In the second sampling-based motion planning strategy, a classical tree graph is utilized to represent the total configuration space. The robot's initial configuration is defined as the root of the tree, and the structure branches by generating new sample configurations from the previous node, added only if collision free trajectories exist between the parent and child [30]. This process iteratively samples achievable movement options until the goal configuration is accessible in a leaf node. Searching the tree for the path between starting and goal configurations is most commonly done with the Rapidly-Exploring Random Trees (RRT) graph planning algorithm [30]. RRT offers flexibility in tree graph search as it expands based on control inputs (achievable configurations), avoiding the probabilistic roadmap's requirement of point-to-point convergence [34]. A sample RRT output consisting of a root and thirteen sampled nodes is depicted in Figure 9.

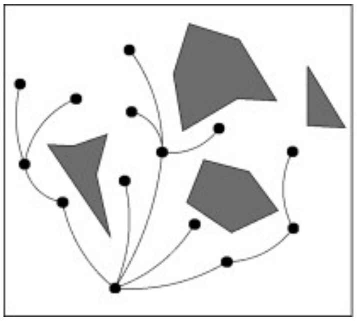


Figure 9: Sampling-based Rapidly-exploring Random Tree graph output featuring a root node and thirteen sample nodes [30]

2.3.2 Coordinate Frames and Transformations

A robotic agent operates in a three-dimensional world. Robotic systems commonly represent their knowledge of their operational environment by mapping a Cartesian coordinate system, or frames, along three axes, *x*, *y*, and *z*. A universal world frame is first defined, in which all other entities, including the robot, exists. The robot further refines its own positional specification within the world frame via three additional coordinate frames: the world reference frame, detailing the robot's physical relationship to its environment, the joint reference frame, describing individual joint movements, and lastly the tool reference frame, specifying movements of the robot’s end effector relative to a frame attached to the hand (and therefore along the rest of the robot’s body) [35]. Significant entities in the scene are also described by reference frames to define their positional and orientational relationships to other entities, the robotic agent, and the world environment. All frames in a scene relate and interact with one another through relationships called transformations, which define a frame’s position and orientation. This information is relative to a parent frame, which is either the universal world frame or an already existing frame in the world. Frames are expected to dynamically change over the course of a task, therefore giving transformations three primary forms [35]. The first is a pure translation, wherein a frame moves in the environment without any adjustment to its orientation [35]. Next are pure rotations, where no positional change occurs between a parent and child frame, but orientation changes along any of the three axes [35]. Lastly, a frame could undergo a combinational transformation that maneuvers positionally and orientationally relative to its parent frame [35]. Illustrated in Figure 10 is a pure translation operation, depicting a child positionally changed with respect to its parent frame. Relationships between frames and their respective coordinate systems are mathematically represented by matrices, which describe current translational and rotational information between a parent and a child [36]. Depending on the desired transformation, a series of matrix operations can describe the relationship between two entities in a robotic workspace in a structured, numerical representation.

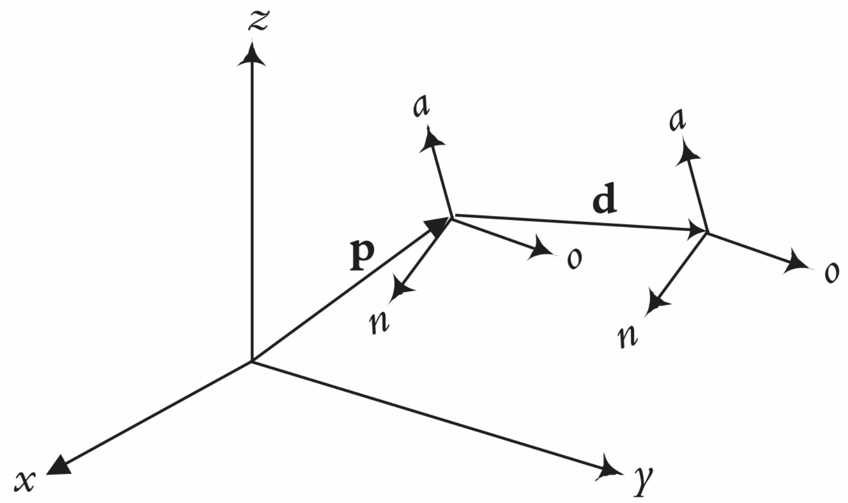


Figure 10: Example of a pure translation between a world, parent, and child frame [35]

2.4 Related Work

2.4.1 Image Sensor Based Object Recognition

Visual sensors, such as RGB and depth lens equipped stereo cameras, are capable of capturing both color and depth information from its FOV. These images, used either independently or in tandem, offer valuable information about a 3D environment despite being two-dimensional.

To estimate pose with a visual sensor, a significant entity must first be differentiated from its surrounding environment. With recent advances in the efficiency and accessibility of machine learning (ML) algorithms, artificial intelligence offers a method of identification via RGB images alone. In a typical use case of this approach, Ryu et al. (2021) surveyed the potential of machine learning in the application of livestock disease monitoring by testing with human subjects' faces [37]. The team trained artificial intelligence models to draw 3D bounding boxes given images taken from an RGBD camera. They found that given enough training data, these models were successful in determining label, distance, and position of a subject [37]. ML algorithms are similarly employed by Rahul et al. (2018), where an RGB and depth capable stereo camera trains a CNN to identify and classify objects, while also calculating distances from the camera [38]. Their model tags suspected regions of images with bounding boxes to get 2D positional data, finally determining the depth dimension via triangulation of a constructed 3D point cloud of the scene [38]. Much of the work in ML object detection aligns with these works, training varying algorithms to identify and classify objects in an image, and offering 3D distances provided capable visual sensor hardware. However, in the realm of robotics, further must be done to relate those bounding boxes and 3D coordinates of the objects to representative poses in the real world.

In maintaining the use of only RGB images, fiducial markers are widely used as a means to classify objects and estimate pose, enabling the guidance of one or more robotic agents via visual servoing. Because these markers only require seeing a pattern and performing point-based registration to obtain the full position and orientation [24], they serve as an incredibly quick and efficient method of pose estimation once affixed to a significant entity. In the work of Rogeau et al. (2020), ArUco tags were utilized in the 6-degree of freedom robotic arm assembly of timber panels requiring precise insertion maneuvers [39]. To accomplish this task, a visual feedback loop was developed beginning with a camera capturing an image of ArUco equipped panels [39]. An algorithm proceeds to calculate the full pose of the fiducial marker, later using this information in the robot controller to fine tune the movement trajectory reaching a target within <5mm of imprecision [39]. This is demonstrated in Figure 11, where a robotic arm inserts a wooden panel at the pinpointed pose estimated from an ArUco tag [39]. Fiducial guided visual servoing is seen again in the work by Yu et al. (2019), where AprilTags, a class of markers, were utilized in guiding a 4 degree of freedom arm in object grasping [40]. Images were captured on an OpenMV camera, responsible for recognizing significant entities with affixed fiducial tags offering full positional and orientational information [32]. These AprilTags were utilized in tandem with force-compliant end effectors to precisely tune the final grasp of targets [40].

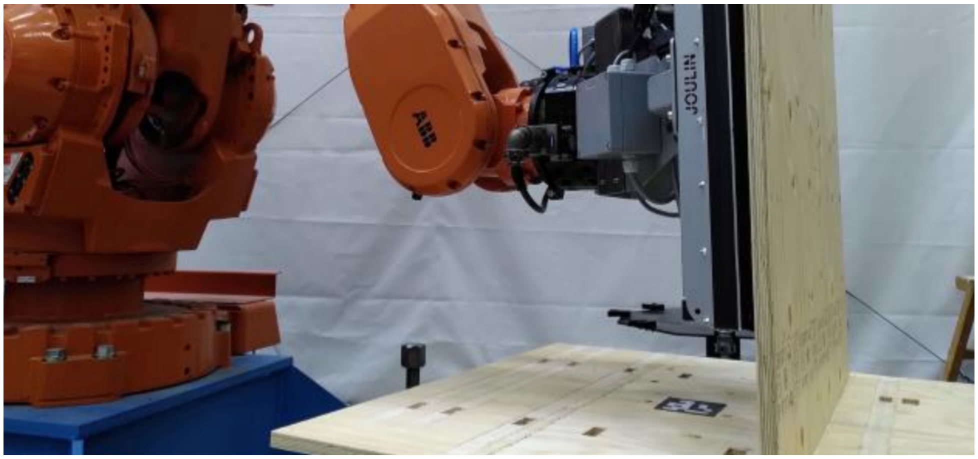


Figure 11: Example of fiducial marker guided visual servoing; insertion of a wooden panel at pose estimated from ArUco tag [39]

2.4.2 Occlusion Search

A robotic system may encounter potential obstacles that impede, or even damage the agent, in a live working environment. While some hazards pose physical harm, others might simply hinder robotic performance or halt execution of a task. Occluding obstacles are one such class of environmental hazard, typically occurring by preventing a robotic agent from reaching a goal configuration or obstructing sensors from perceiving the surrounding environment. In a robotic system operating with or depending solely on visual sensors, such occluding obstacles present significant challenges in gathering information about the workspace. To negate the interferences brought forth by occlusions, predictive estimation, active visual search, or a combination of the two are viable counter strategies.

Beginning with the former, Chi et al. (2019) propose a tracking framework of deformable linear objects (DLO), utilizing RGBD data and Coherent Point Drift (CPD) to estimate occluded regions of said objects [41]. Their approach begins with regularization of CPD output via locally linear embedding and constrained optimization, offering topological consistency of the object view when occlusions obscure a DLO [41]. Next, the team's developed algorithm considers free space visible in the scene to pinpoint areas of occlusion, and to improve already made estimations of on point locations [41]. Finally, shape descriptors are employed to fully estimate and reason the most probable positions of an object of interest under occlusion [41]. Given partial view of a DLO, this method seeks not to return an object to view, but rather assumes its position based on expected approximations of where it should be behind an obstacle. Figure 12 illustrates the results of this work, where the shape of the deformable linear object is estimated with occlusions partially obscuring regions of the target [41].

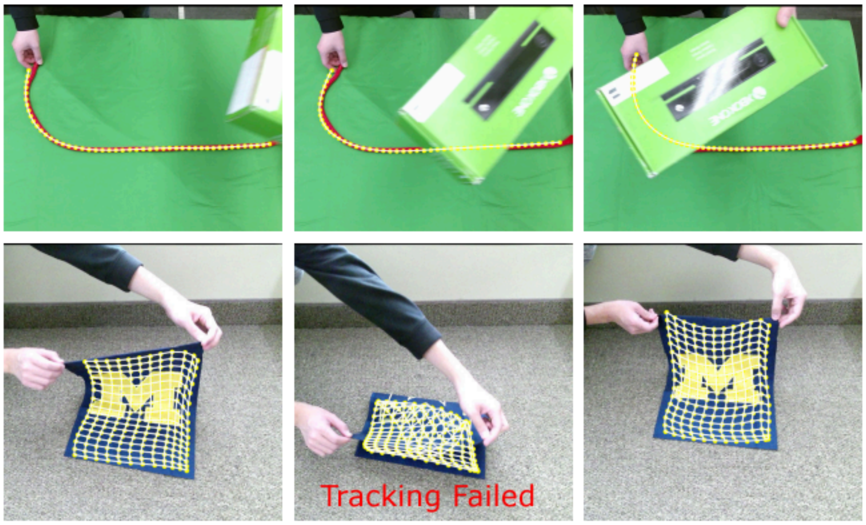


Figure 12: Estimation of deformable linear object position under partial occlusion via the method developed by Chi et al. (2019) [41]

Lin et al. (2015) proposed an implementation consisting purely of active search, requiring no estimation. The team conducts robotic search of an environment based on the selection of potential agent configurations [42]. An A* search tree algorithm was developed with leaf nodes as sampled, available poses that lead to, or enable the grasping of an occluded target [42]. The robot in this scenario is able to move either itself or obstacles in the environment to improve its view, selecting actions that minimize cost of future action cost [42]. Nodes in the tree are connected to each other if transitions between the two are available to the robot, and a greedy plan is applied such that minimum actions are taken to reveal remaining hidden target poses that lead to revealing the goal [42].

Applying active visual search, in tandem with predictive estimations calculated by the system, significantly enhances the effectiveness of occlusion search compared to using a single approach alone. An ideal joint effort may consider an object's likely position behind occlusion and initiating a search at the estimated position to search an entire scene more efficiently. Radmard et al. (2018) developed a system wherein a robotic agent repositions its visual sensor to achieve a view around an obstacle by estimating its most likely location [43]. The team developed a particle filter responsible for continuously updating a guess, an estimated position of the object found by their algorithm [43]. A map of the occlusion's boundaries is first produced to compute potential obstacle clearing motions, while a cost function optimizes between information gain of the subject or obstacle and cost of maneuvering the sensor [43]. A planner uses this cost to balance information gain and search to prevent a complete mapping of the scene, until a view of the target object is acquired [43]. Similarly in the work of Wong et al. (2013), a robotic agent explores specific areas depending on most probable target locations [44]. In this cupboard scenario, obstructing entities in the form of containers hold goal objects within. A generative model of the scene was developed, where, co-occurrence information and spatial constraints guide a robotic agent to searching specific containers for the target [44]. This similarity-based approach estimates target location by matching the target to occlusions with similar attributes, much in the same sense as a coffee mug would likely be located near other cups and not with bowls or plates [44].

2.4.3 Summary

The key objective of this thesis is the utilization of various techniques to extend the information gained about an environment through visual sensors alone. Fed RGB or depth images from equipped stereo cameras, a robotic system must first detect and recognize an object of interest among its surroundings. Machine learning algorithms for object detection, as presented in the works of [37][38], demonstrate capability in identification of an object and estimating its position by detecting features learned from training data. Although successful in recognition, it lacks the component of defining a full pose for the object, namely orientation, from the images alone. Fiducial markers remedy this by offering unique identifications discernible by their patterns. Robotic visual servoing as performed in [39][40] is made possible by the full pose estimations calculated via point registration [24] of marker geometries observed in the RGB images. Fiducial tags, however, are not readable unless the entire pattern is visible or a computationally expensive backup algorithm estimates the remaining pattern blocked by occlusions. This thesis work therefore aims to fully estimate the pose of an object, including position and orientation, utilizing only RGBD stereo cameras without the need of fiducial markers. Additionally, partial occlusions should not impede pose estimation, creating a robust system capable of visual servoing without searching unless a target is heavily obscured.

In the presence of occlusions, three approaches to searching behind obstacles are presented. A system could fully estimate the position of occluded objects as done in [41], or initiate a comprehensive search around obstacles by minimizing action costs between maneuvers in the robot's total configuration space [42]. Active search can also be guided by estimations, as seen in the combined approaches of [43][44]. Here, robotic agents prioritize search in locations with a high chance of containing the target. This thesis aims to search around occlusions efficiently and robustly, characterized by not processing the entire configuration space while successfully locating a target. Additionally, the search algorithm proposed by this thesis should be lightweight, refraining from heavier predictive estimations using factors considered in the presented related works.

3 Methods

3.1 System Overview

3.1.1 Configuration

The method presented in this section describes aspects of an end-to-end solution to the DLO connector end slip anomaly detection and response problem. The overall system is designed to emulate the interior of a SmartHab, dimensionally appropriate for the anticipated volumetric constrained dimensions of this human-crewed spacecraft. It is configured and composed of two parts: the robotic agent and its workspace. The robotic agent is comprised of two Trossen Robotics ViperX 300 S six degree of freedom (DOF) robotic arms equipped with gripper end effectors. Two cameras, mounted at different locations in the environment, provide visual information about the scene through RGB and depth images. The cameras consisting of the Intel RealSense d435i and d415, are affixed, respectively, one to the left arm's wrist joint and the second to a rear camera stand behind both arms. The entire setup, including the environment and robotic agent, is depicted in both real world (left) and RViz simulation (right) views in Figure 13. The real-world view displays the two arms of this system's robotic agent, with two visual sensors one mounted behind and the other mounted on the left arm's wrist joint. In front of the pair are the mock work environment with attached DLO. The simulation view displays the ROS (Robotic Operating System [45]) frames for the robotic arms, cameras, and world. A central world frame is defined on the plane of and directly between the robotic arms. The environment is composed of a floor and two plywood panels of dimensions 36 x 24 inch^2 and 12 x 24 inch^2 attached to one another. These are placed 20 inches (+20 along the *x*-axis) in front of the world frame. The two robotic arms are mounted along the world frame's *y*-axis, offset seven inches along the positive and negative directions. The rear camera is mounted 14 inches behind (-14 along the *x*-axis) and 16 inches (+16 along the *z*-axis) above the world frame, and the arm camera is mounted 1 inch (+1 along the *z*-axis) above the left arm's wrist joint.

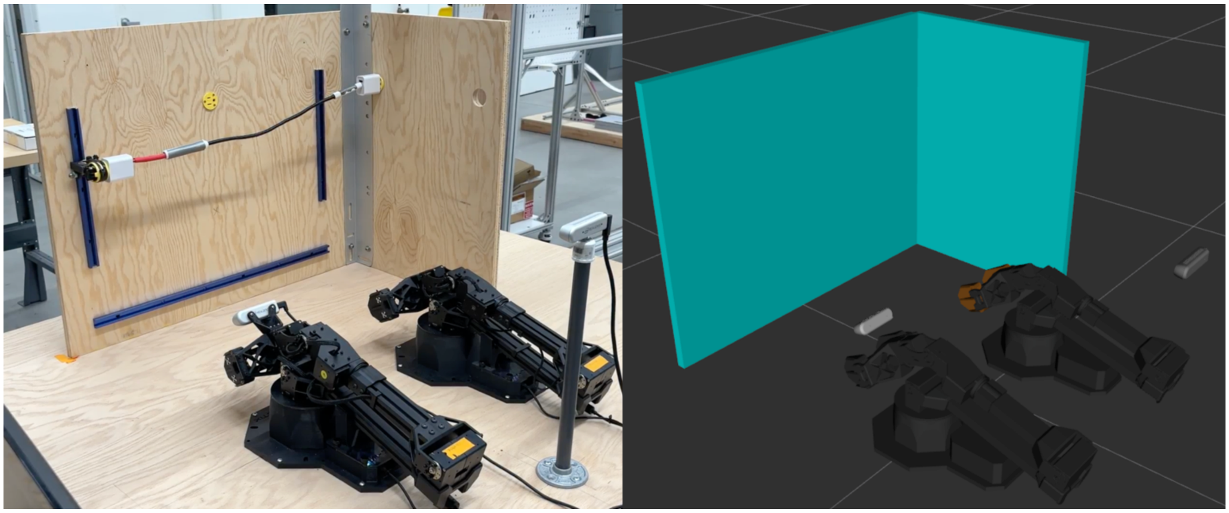


Figure 13: Actual (left) and simulation (right) views of the robotic system in environment, with two visual sensors mounted behind and on the left arm's wrist joint

Two different obstructions were also created to challenge the system's ability to locate the end of the loose connector. These are made of purple polystyrene foam and are displayed in Figure 14 with labels specifying their dimensions. Both of these foams were sized relative to the volumetric constraints of the environment. They were also shaped such that they could block the end of the DLO connector from the rear camera view without completely preventing the robotic agent from searching around them with the arm camera.

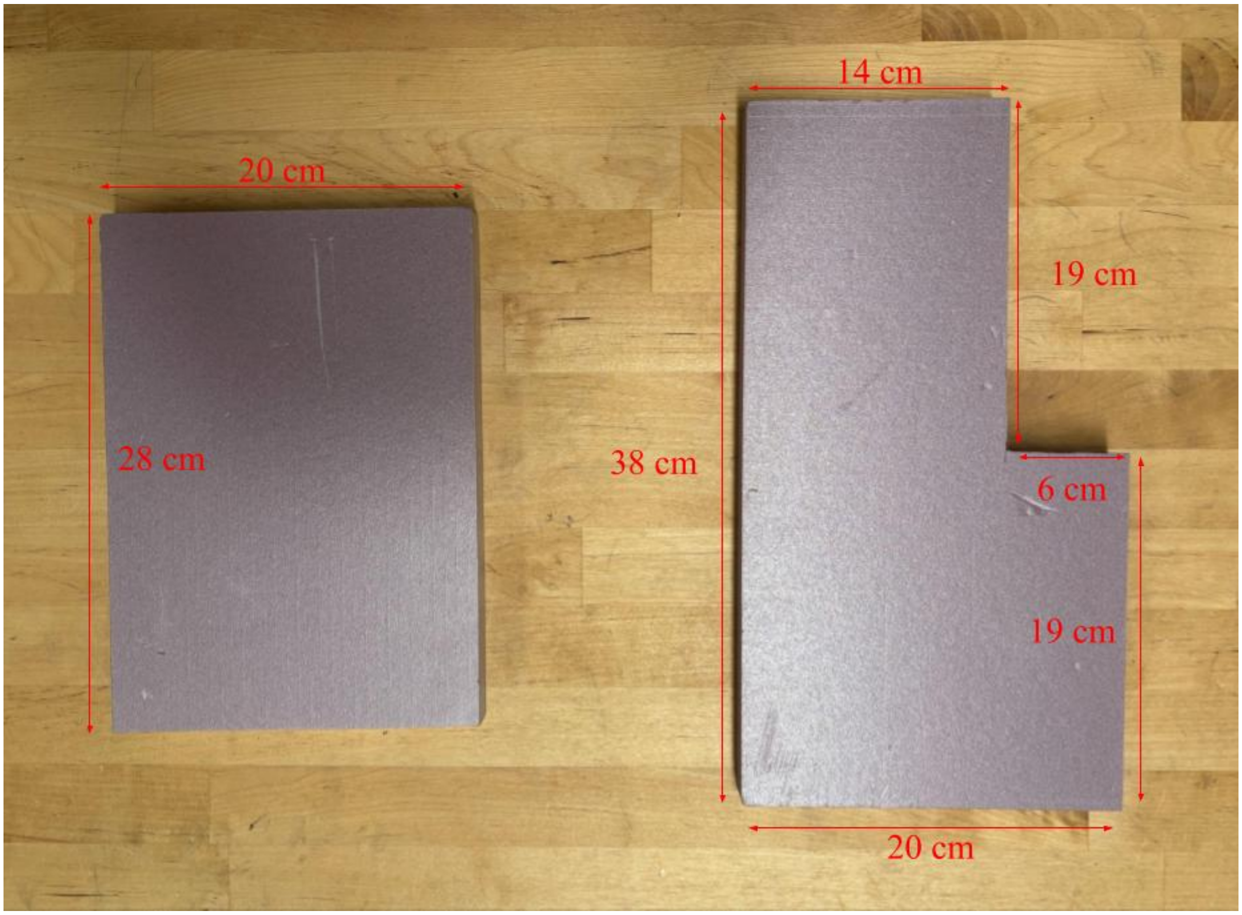


Figure 14: Designed short (left) and tall (right) foam environment obstacles with dimensions

Affixed to the back panel are four blue racks, providing a sliding connection point for the DLOs. These attachment points are outlet socket USB-C power blocks, roughly emulating the shape and connection face of a battery ORU. The DLO is a red USB-C to USB-C cable, with a bolstered

red plastic wrapped around the end of the connector. As a method to approximately emulate near-zero gravity conditions, the length of the cable is reinforced with a black plastic wrapping as to stiffen the entirety of the DLO. The stiffening would allow the DLO to hang in space rather than falling due to gravity. Figure 15 depicts the DLO and mock battery ORU connector blocks utilized in this work, with respective dimensions labeled.

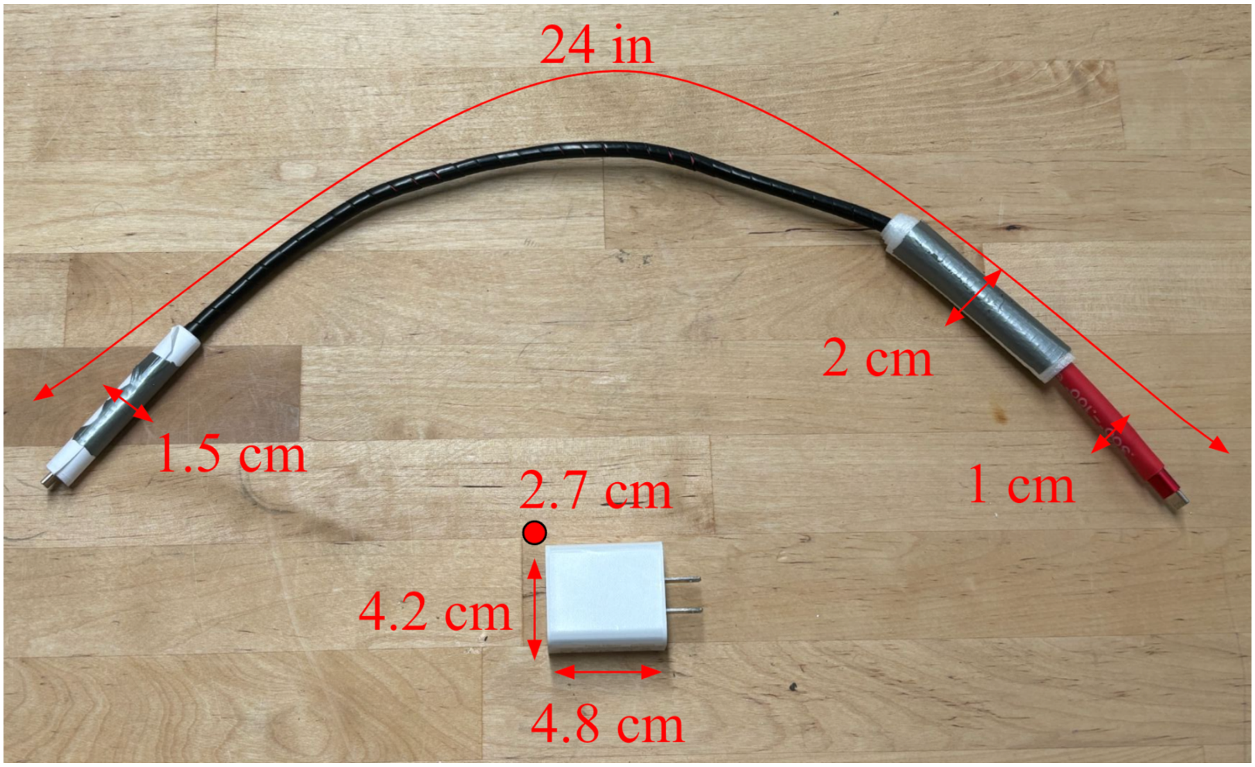


Figure 15: Mock DLO cable (top) and battery ORU (bottom) with dimensions

The starting configuration of the environment consists of the arms in their sleep state and the DLO plugged into both connection points on the wooden panels. The system, employing its rear mounted camera, continuously monitors the position of the DLO connector end. The right robotic arm, tasked with manipulating the connector end, grasps and unplugs it from the USB-C power block. It maneuvers to a temporary stowing point before an engineered slip occurs dropping the unplugged end into the workspace. The system, having lost the DLO, consults the rear camera for sight of the connector. If unsuccessful, a search pattern over the workspace is initiated. Once the connector end has been identified, pose estimation for the loose connector end is conducted by either camera. This is accomplished using fiducial markers, machine learning, or RGBD image processing. Following successful pose estimation, the pose of the connector end is transformed to the coordinate system of the right manipulating robotic arm. Motion plans are then generated and executed for retrieval and restoration of the connector end to its original destination.

3.1.2 Software Overview

The codebase developed for this thesis was written in Python 3 and built under the Noetic distribution of ROS 1 running on a Linux Ubuntu 20.04 operating system. Software interfaces with the robotic arm hardware via the Robotic Operating System (ROS) framework, an open-source structured communications layer facilitating transfer of information between robotic agents and written code [45].

The software presented in this thesis follows standard ROS practices using an object-oriented approach of information transmission between class-like nodes. This messaging infrastructure supports transmission of data between nodes via a publish and subscribe message pattern [45]. Several custom nodes were implemented in this thesis, including responsibility for intaking sensor data, processing it for relevant information, calculating frame pose estimations, and deciding on appropriate robotic responses depending on perception of the environment.

As much of this work concerns applications of computer vision to guide visual servoing, OpenCV, a software library containing computer vision algorithms and functions, is employed to process output from the system's cameras. OpenCV is typically accessible via the standard

Python 3 distribution but receives more direct support from ROS via a framework bridge. Not only is the library fully accessible to ROS nodes, but efficient conversions between ROS and OpenCV datatypes are available. Control of the robotic arms is handled by MoveIt, the ROS motion planning framework which, once configured for this two-arm setup, offers motion planning and trajectory execution capabilities. OMPL, a sampling-based motion planner available in MoveIt, handles planning and movement of the arms throughout this work [32]. A series of demonstration scripts leverage MoveIt motion planning and ROS node communication to test the robotic skills developed in this body of research.

3.1.3 System Requirements and Assumptions

The requirements for the robotic system developed for this thesis are as follows:

1. The robotic system demonstrates the ability of pose estimation, including position along the *X*, *Y*, *Z* axes and orientation, in quaternions, along the *X*, *Y*, *Z, W* axes, for the connector end of a deformable linear object (DLO). Pose estimation is calculated directly from output RGB and depth images published by the visual sensors, with no prior knowledge pertaining to the DLO or the environment.
2. The cameras within the robotic system demonstrate the ability to track an object of interest by monitoring the state of the DLO. By continuously tracking the position of the DLO, an unknown state is flagged in the event that its current position differs from what is expected by the system.
3. The robotic agent demonstrates the ability to perform its task despite the impeding impact of occluding obstacles. In the presence of occlusions, a search of the workspace is

conducted using a robotic arm equipped with a camera if the DLO connector end is no longer in a known state.

4. The robotic agent develops a motion plan and executes retrieval plans for a connector in an unknown state by locating the object, calculating its full pose estimation, and sending a robotic arm for grasping it.

The assumptions used for the robotic agent developed for this thesis are as follows:

- The location and geometries of the ground and panels are known to the system.
- Initially, the two ends of the cable are connected to two connection points on the back and right panels. The locations of these connection points are known to the system.
- One end of the cable is always fixed, plugged into the right panel's connection point. The other, left end, along with the rest of the wire, may be free if slipped from the robotic arm's gripper.
- The system has no information about the physics or properties of the cable, except limited information about the left connector end to be unplugged. Depending on the pose estimation technique employed, the system is only aware of one of the following:
    i. The fiducial marker pattern affixed to the connector end that the system is to search, identify, and register.
    ii. Prior learned pattern features (through neural network training) about the connector for use in machine learning object recognition.
    iii. The color of the connector end it is to identify and estimate pose for.
- The process of unplugging the cable and maneuvering it to a new location is known and organized by the system, and not handled by this work.

- At any point, the pose of the connector end is assumed unknown to the system. Identifying, tracking, and monitoring the status of the connector end requires pose estimation by the system to determine position and orientation.
- The work presented in this thesis is intended to be utilized in zero gravity conditions. This is emulated under standard gravity conditions via the use of a stiff cable, reinforced with semi-malleable plastic. This would allow the cable to hang in the air rather than falling under gravity

3.2 Identification and State Estimation of Connector Cable End

3.2.1 Pose Estimation Techniques Overview

The research presented in this thesis concerns manipulation of a deformable linear object at its end extremity, where a point of connection exists for plugging into an appropriate outlet. In order to successfully manipulate the cable, the system must first be able to identify, monitor, track, and search for the end of a cable using its position and orientation. Pose estimation of the cable's connector end is critical as it informs the system of the cable's state, enabling all other capabilities presented in Methods (**Section 3**). This information can be utilized to update the robotic agent on task progress and status, or be converted to individual ROS frames which guide the robotic arms in visual servoing based manipulation of the DLO's.

Three pose estimation techniques are presented in this work, all based purely on visual feedback and requiring only the RGB and depth image outputs from a pair of RGBD capable stereo cameras. Although each approach receives the same input options, they require specific setup conditions that result in unique identification and pose estimation procedures. The three strategies for pose estimation demonstrated in this thesis are as follows:

1. Fiducial markers
2. Machine learning object detection
3. RBG segmentation and depth point cloud processing

3.2.2 Fiducial Markers

3.2.2.1 Camera Calibration

For a system to perform pose estimation with fiducial markers, two steps must first occur. A 2D image is first produced which is a flat view of the camera's observed scene. Cameras, by the nature of their construction and how images are captured by their internal sensors, may introduce levels of distortion that do not realistically depict a view of the environment. Figure 16 illustrates the various types of potential camera lens distortions, which vary depending on the type of camera capturing the image. The second step is to correct any distortions so that the returned image accurately represents the scene.

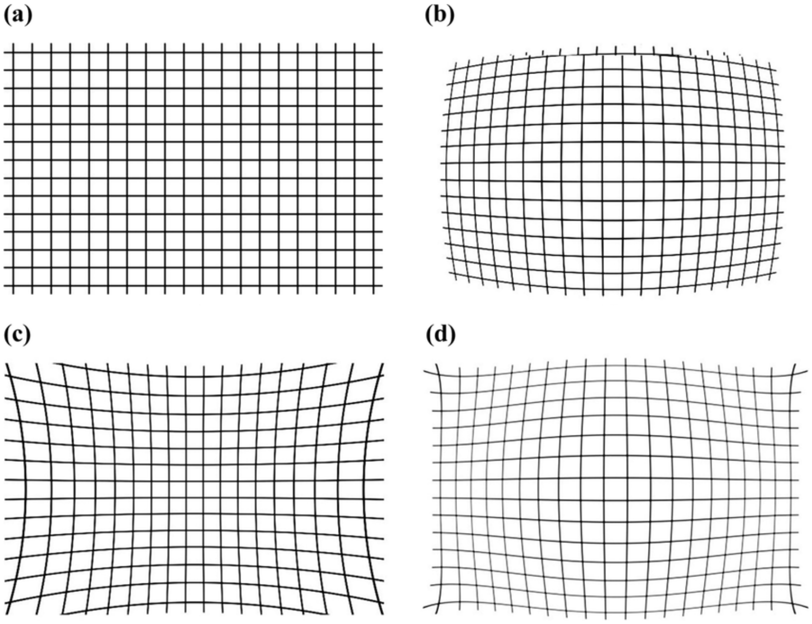
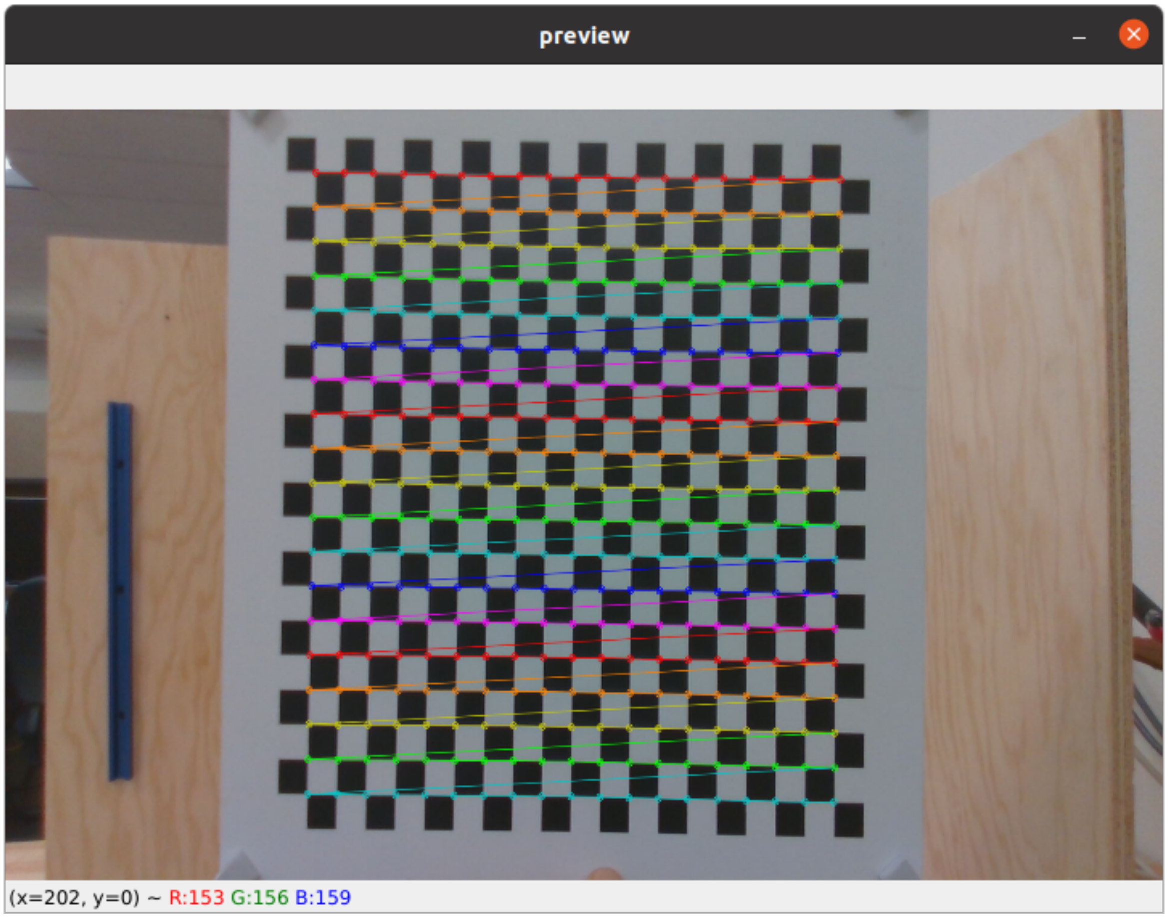


Figure 16: Primary types of camera lens distortions: (a) No distortion (b) Barrel distortion (c) Pincushion distortion (d) Mustache distortion [46]

Because fiducial marker pose estimation is heavily dependent on the curves, contours, and relation of points observed, distortion heavily skews this calculation. Depending on the specific camera used, the manufacturer defaults may already correct for distortions in the sensor's intrinsic parameters [11]. Additional distortion correction is possible through OpenCV, which offers a suite of functions for camera calibration using a flat, reference checkerboard image. The dimensions of the pattern are passed to the functions, which are used to recognize distortions based on the checkerboard's appearance in relation to those dimensions in the camera's images [47]. The camera's view of the checkerboard, with calibration lines drawn by OpenCV, is depicted in Figure 17. Distortions are recorded in the form of image coefficients, which are then passed to fiducial identifying functions to correct for distortions in images scanned for present markers.



(x=202, y=0) ~ R:153 G:156 B:159

Figure 17: Checkerboard with OpenCV calibration lines drawn over the RGB image

3.2.2.2 Fiducial Marker Registration

With calibration complete, fiducial markers to be placed in the environment must be registered with the system, essentially informing the algorithm of what specific patterns to search for. For the fiducial marker work completed in this thesis, ArUco markers were chosen since they are well supported in OpenCV. They are also robust with high detection rates across various environments, as found in [20]. For differentiating tags in a scene, each unique pattern included in the set of default ArUco dictionary are numbered. Identification of one of these default patterns therefore returns a numerical value, which can be recalled to not only gather information about which patterns were found, but also to identify the entities those tags represented in the environment.

3.2.2.3 Fiducial Marker Identification

Once a system is calibrated with distortion coefficients, and informed of fiducial markers to identify, the OpenCV ArUco identification algorithm can be called for an image. The camera publishes images at a set framerate, received by a ROS node with an appropriate callback for processing the picture. Images are passed to the node in the form of ROS images, which are converted via an OpenCV bridge to format the files into OpenCV images. This formatting step enables the application of OpenCV computer vision processing functions on images returned by the camera.

OpenCV detects ArUco markers in a two-step process over a compatible image type. The function first searches for ArUco tags by applying an adaptive thresholding to segment potential markers from their surroundings. Any plausible candidates have their contours extracted, and any candidate bearing no resemblance to the tags are discarded [22]. Next, valid ArUco patterns are identified by analysis of their inner codification. A perspective transformation and thresholding prepare prospective pattern bits for verification of whether the pattern belongs to a default ArUco dictionary [22]. Assuming valid markers are identified, outputs required for pose estimation are calculated alongside any image coefficients to correct for distortions. These outputs, **tvec** and **rvec**, are vectors representing the 3D positional difference between the camera and the marker, and the Rodrigues axis of rotation and rotation angle about that axis between the camera and marker center respectively. These vectors supply the marker's position and orientation information relative to the camera and are determined by the features of the tag's pattern via a point-to-point registration between a known base view of the pattern and the pattern observed in the image [25]. In the context of this thesis, two ArUco tags are present in the scene, affixed to the left cable connector end and an eventual stowing point. Figure 18 depicts the rear camera view of the workspace, with pose estimations generated for these two fiducial markers by OpenCV.

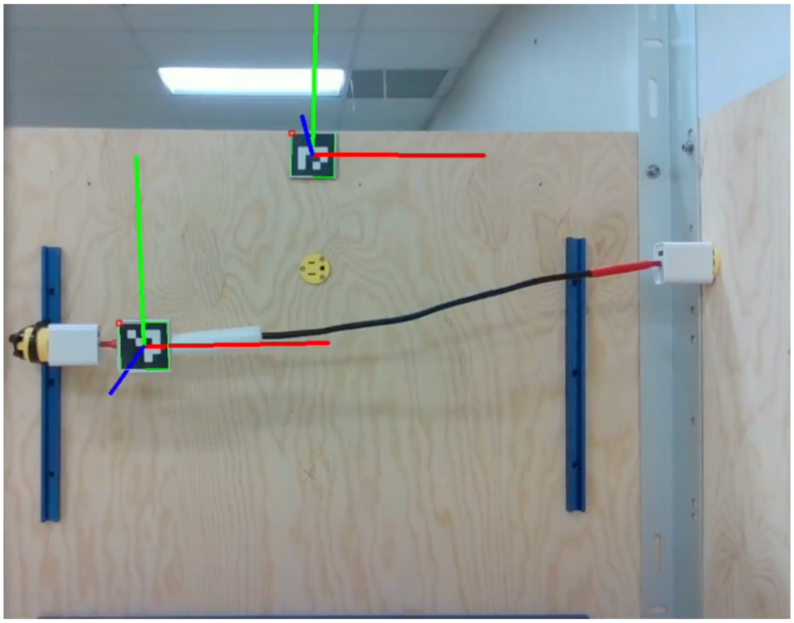


Figure 18: Pose estimation of two fiducial markers in workspace, generated by OpenCV

3.2.2.4 Pose Estimation

A frame for an ArUco marker can be drawn using the vectors following identification. The translational vector is applied directly as the new ArUco child frame's position, functioning as a transformation from the position of the parent camera frame's location. The rotational vector requires minor processing before it can be used in representing frame orientation. The Rodrigues values are given in the form of a float vector but can be represented as a rotational matrix. The ROS tf library, capable of tracking multiple coordinate frames over time, can transform this matrix of Rodrigues angles to Euler angles, and finally from Euler to quaternions. Quaternion values can then be applied to calculate the orientation portion for the ArUco frame. With position and orientation set, a full pose is defined enabling the generation of this new frame representing the fiducial marker, relative to the camera's frame.

3.2.2.5 Representative Frame Adjustments

Generated fiducial marker frames exist in the ROS system's world coordinate system, serving as reference points for entities where the tags are affixed to in the workspace. An ArUco tag placed on the left connector end, for instance, essentially represents that entity by offering numerical identification and offering pose estimation. For visual servoing, these frames, if positionally reachable and orientationally aligned to the robotic agent, act as goal poses for the arm's end effector. Under these conditions, a motion planner is capable of calculating and executing a trajectory to maneuver the robotic arm from its initial position to the pose specified by the fiducial marker pose, as similarly accomplished in [39][40].

Suppose, however, that the pose generated by a fiducial marker frame is not exactly where a frame of the robot should traverse to. The ArUco affixed to the left cable end is at the extremity of the cable, and grasping of the marker itself could result in damage over repeated manipulation. In this instance, a simple offset transformation along the ArUco frame's red *x* axis remedies this issue by attempting a grasp point slightly right of the marker's physical position. The motion planner can now generate a trajectory for the arm by sending it to the ArUco frame offset, accurately grasping via pose estimation and safely handling the marker with this minor positional transformation. Figure 19 depicts the RViz simulated view of the environment featuring the two original ArUco marker frames, and the third offset frame used for grasping.

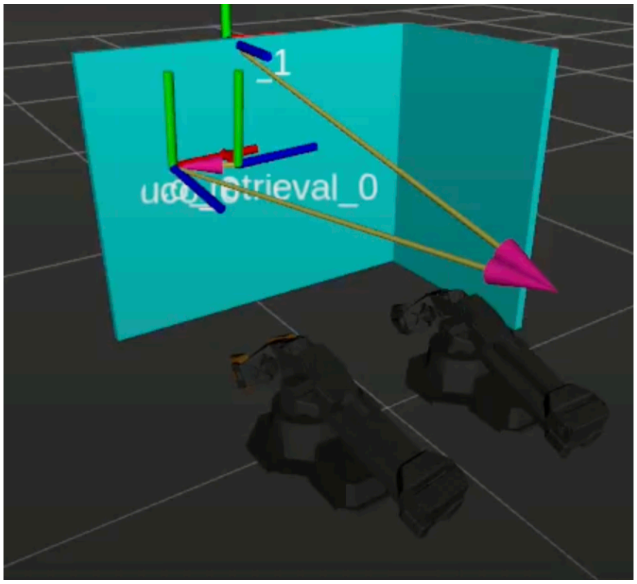


Figure 19: RViz simulation view of fiducial marker frames and offset grasping frame

3.2.2.6 Fiducial Marker Visual Servoing Pipeline

Figure 20 illustrates the full pipeline this system follows in order to visually maneuver the robotic agent using fiducial markers. The selected camera constantly publishes its intrinsic parameters and captured RGBD images. The former is subscribed to by a camera calibration node that calculates any coefficients for correcting lens distortions, improving accuracy of fiducial marker readings. A central node that performs the Aruco tracking subscribes to the camera for its RGB images and receives any distortion correction coefficients from the camera calibration node. It passes this information to cv2, which detects and returns the **rvec** and **tvec** matrices of detected Aruco markers. The node generates a frame based on the returned matrices, making final adjustments before publishing this information in the form of a frame, which the system may then use to guide the robotic agent in visual servoing.

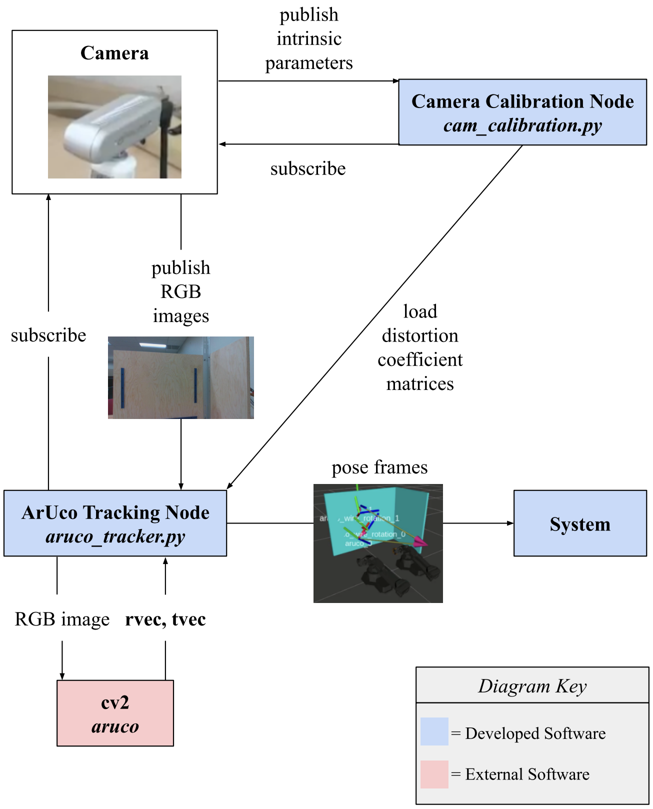


Figure 20: Fiducial marker guided visual servoing pipeline

3.2.3 RGBD Image Processing

3.2.3.1 RGBD Image Segmentation for Position

Obtaining position of the cable connector is achievable by processing the RGB and depth images published by the system's visual sensors. For this approach, this thesis follows the work of [7] in which this sensory information from the camera is processed to generate a segmented depth image by leveraging concepts from computer vision [7]. The purpose of the segmentation is to filter the cable connector end from its environment based on its red color, represented by a Hue Saturation Value (HSV) range [48]. Once filtered, the system will have a binary image representation of the connector to further process for pose estimation, ignoring irrelevant information from the rest of the environment. The HSV filtering process for image segmentation, as implemented by [7] is as follows:

1. An RGB image from the camera is passed to the OpenCV library's *cvtColor()* function to convert it from the RGB color model to HSV.
2. An HSV color range is defined for the object of interest, in this case red for the color of the cable connector end.
3. A mask is generated of all RGB image pixels falling within the defined HSV color range using the OpenCV library's *inRange()* function.
4. The original image is segmented to create a new depth image by applying the bitwise AND operation to the raw image with the masks defined in step 3.
5. The result is converted to a binary image.

The HSV filtered segmented binary image represents the cable connector, but still bears noise from the environment. To remedy this, further isolation of the connector from its environment is achieved using three morphological operations [7], [17]:

1. Dilation: performed by passing the segmented image to the OpenCV library's *dilate()* function, enlarges the boundary of the isolated connector end. It corrects for errors such as holes in the segmented object, missing boundaries, or disconnected portions of the object.
2. Contour detection: extracts only the pixels associated with the curve of the connector by first applying the OpenCV *findContours()* function to the binary image, returning a list of all contours found. The OpenCV contourArea() function is next called to find the contour with the largest area, which corresponds to the connector. Dilation is therefore a

significant preprocessing step as filtering for the largest contour would fail if the segmented object of the connector was broken up by errors fixed during that step.

3. Erosion: counteracts the effects of the dilation step, removing added boundaries to the segmented object. This is done by passing the binary image following contour detection to OpenCV's *erode()* function.

Following this extraction and refinement procedure, the cable connector is isolated from its environment in the format of a binary image. A ROS nodelet manager, defined in the system launch file, converts from the depth binary image format to an XYZ point cloud. The system now has a point cloud representing the cable connector in a coordinate system with respect to the chosen camera's frame of reference. Figure 21 depicts the loose cable's connector point cloud, following HSV segmentation and conversion from a binary image, in the RViz simulation view of the environment.

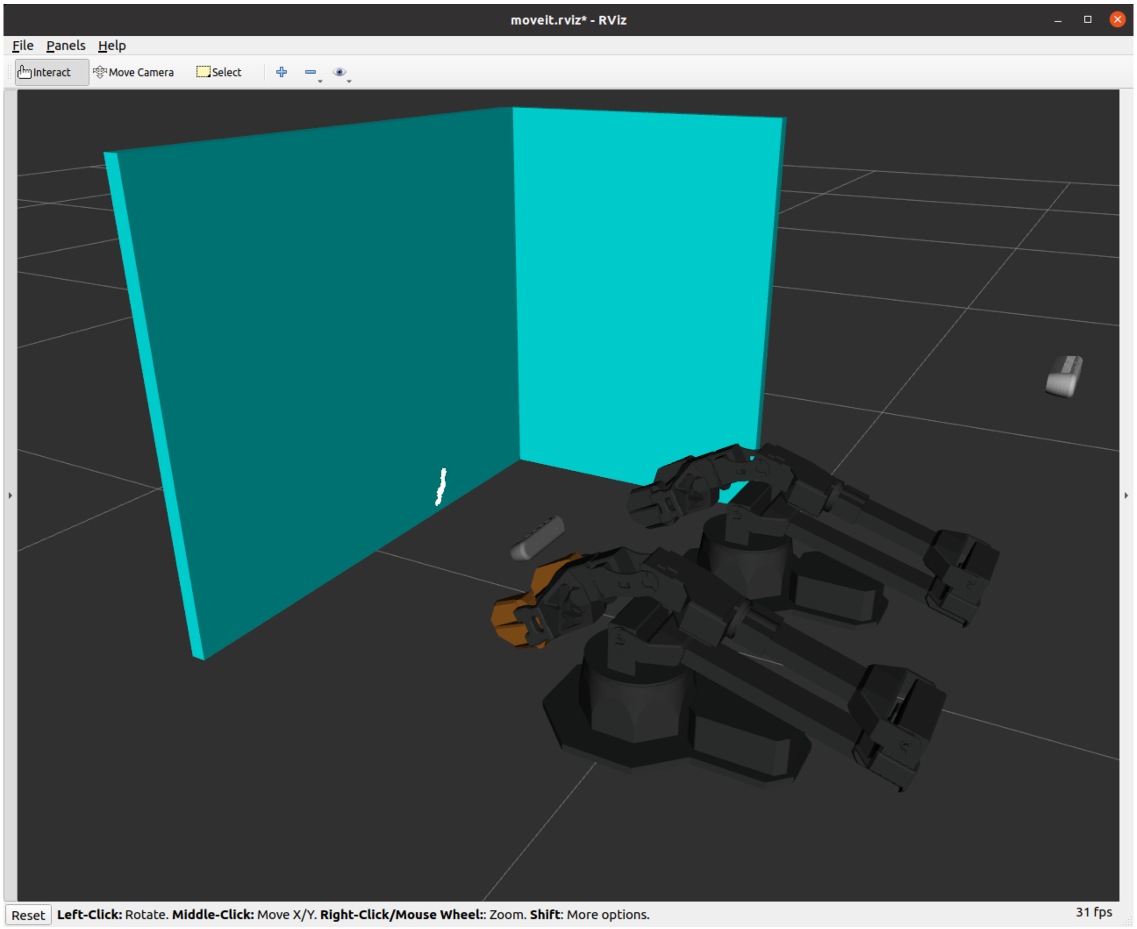

moveit.rviz* - RViz
File
Panels
Help
Interact
Move Camera
Select
Reset
Left-Click: Rotate. Middle-Click: Move X/Y. Right-Click/Mouse Wheel:: Zoom. Shift: More options.
31 fps

Figure 21: Loose cable connector point cloud (white) in RViz simulation view of environment

Position can next be calculated via processing of this connector point cloud. The absolute center pixel along all three axes of this point cloud is a prime candidate for the position portion of pose estimation. To make this approach more robust (avoiding the influence on positioning any single pixel would have) outliers are first filtered before *n* points along each of the *X*, *Y*, and Z axes are averaged to generate an *n*-mean 3D point instead. The steps for calculating the center *n*-mean point are as follows:

1. Define a variable *n* representing the total number of points to average for the calculation of the point cloud's new 3D mean center point
2. Sort all points in the point cloud according to a dimension (*X, Y,* or *Z axis*)

3. Extract the *n* middle points from the point cloud along the axis selected in step 2, and average their values
4. Repeat steps 1-3 for all three axes and input their value to the corresponding position in the new *n*-mean center point frame

Performing this procedure yields an optimal center of the connector's point cloud, which can then be supplied to the position portion of the connector's pose estimation. Figure 22 depicts a generated frame, relative to the camera parent frame, for the cable connector with this *n*-mean averaged center point as its position with no orientation values set.

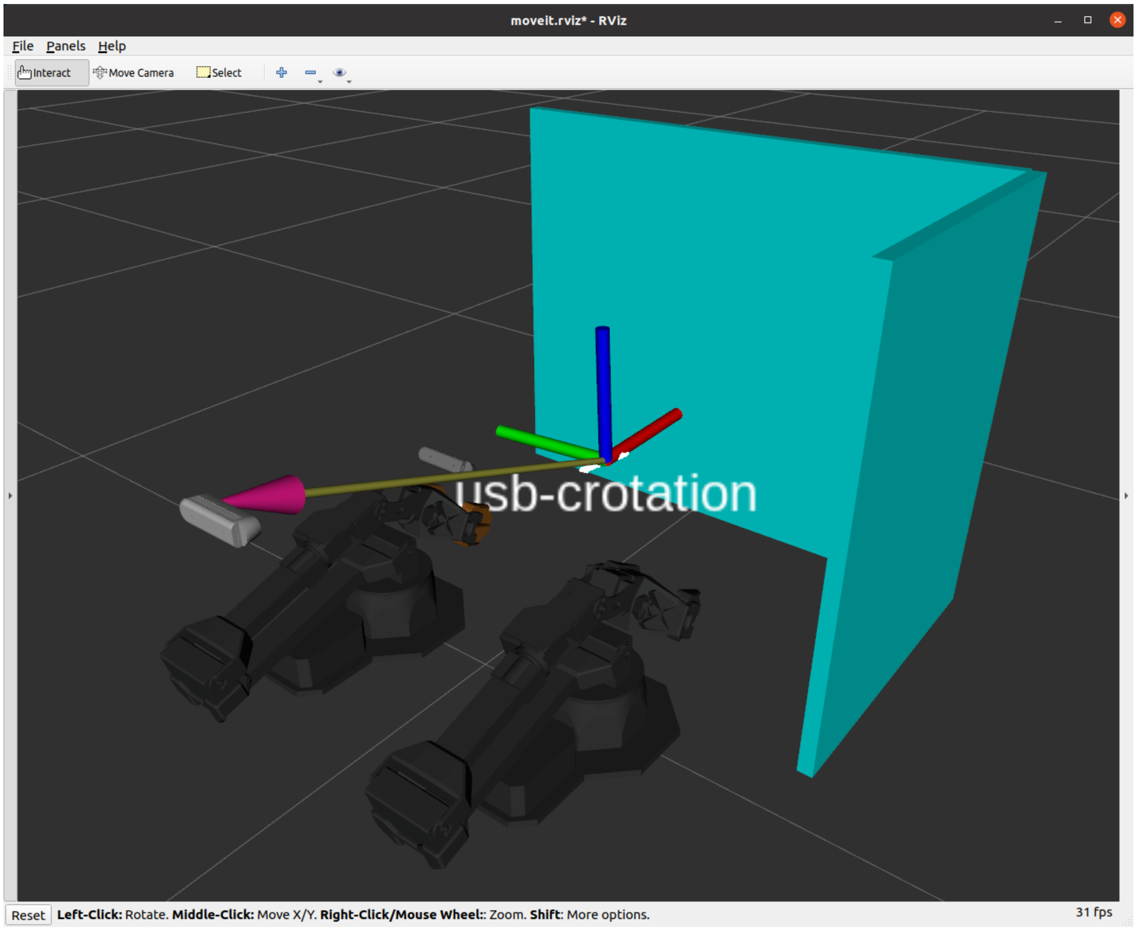

moveit.rviz* - RViz
File
Panels
Help
Interact
Move Camera
Select
usb-crotation
Reset
Left-Click: Rotate. Middle-Click: Move X/Y. Right-Click/Mouse Wheel:: Zoom. Shift: More options.
31 fps

Figure 22: Generated frame for connector with position given by averaged center point

3.2.3.2 Point Cloud Processing for Orientation

The same segmented binary image's point cloud used to find position is leveraged to find the orientation of the cable connector. A point cloud, by nature of it being three dimensional, inherently contains information about how it is oriented in the environment. Orientation for a cylindrical connector end, abstracted as a geometric line in 3D space, can therefore be calculated by the vector connecting its start and end points [49]. Rotational information about each axis, offered by the directional component of the vector, is defined by its head and tail points. A ROS conversion of this vector to a quaternion rotation is then assigned to the rotation component of the frame representing the pose estimation of the connector end.

Calculating orientation begins by determining the minimum and maximum points of the point cloud, representing the tip and the base of the connector end. The *n* minimum and maximum points of the point cloud are similarly averaged for finding orientation. Because the point cloud is already well isolated by the segmenting process described in *Section 2.3.3.2*, few outlier points remain. These are first removed via a mean of all points along an axis, where any points outside a standard deviation threshold are deleted. Consideration is then placed on which axes' minimum and maximum points should be utilized, depending on the actual orientation of the connector in the environment. If the connector is principally horizontal, the points with the *n* minimum and *n* maximum *X* values are averaged for the starting and ending points. Similarly, the *n* minimum and *n* maximum *Y* values are averaged if the connector is principally vertical. The decision on which direction and axes to move forward with is made by comparing the distance between the minimum and maximum averaged points along the *X* versus *Y* axes, with the connector's principal orientation given by the greater distance. The direction of the line is the

3D difference vector between the minimum and maximum averaged points along the selected axis. This is depicted in a bold red line for legibility in Figure 23's RViz simulation view.

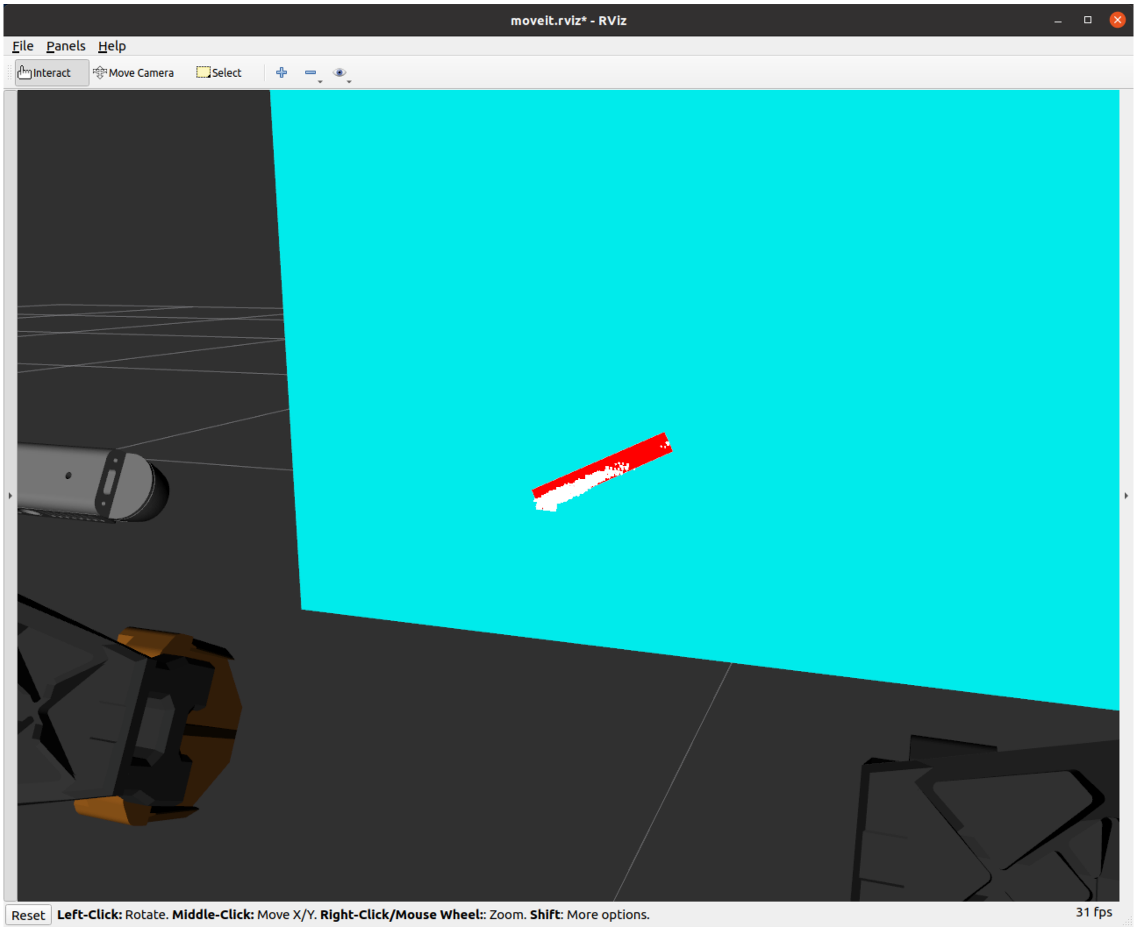

moveit.rviz* - RViz
File Panels Help
Interact
Move Camera
Select
Reset
Left-Click: Rotate. Middle-Click: Move X/Y. Right-Click/Mouse Wheel:: Zoom. Shift: More options.
31 fps

Figure 23: Direction vector (red) between the averaged minimum and maximum points of the connector point cloud

The direction vector is then normalized to have a length of one unit. This step ensures the vector represents only the direction, and not the magnitude of the line as well [49]. Each component of the direction vector, the difference between the average minimum and maximum for each axis, is divided by the total length of the vector. Finally, the vector is converted to quaternions to match the orientation component of a new ROS transform frame. It is important to note that the rotations calculated via this method only offer suitable rotation values about the Y and Z axes.

Due to the point cloud’s cylindrical nature, this vector approach results in a non-unique solution, and therefore unstable, orientation values about the *X* axis. Therefore, the *X* axis rotation is copied from the parent camera frame. By matching the *X* axis of rotation to the typically upright camera frame, grasping of the connector is simplified as a result of the relatively more stable orientation of the camera.

3.2.3.3 Pose Estimation

Having calculated both position and orientation from the connector point cloud, a new frame representing the cable end can be generated by supplying positional coordinates and rotational values from *3.2.3.1* and *3.2.3.2* respectively. This frame is positioned at the center of the point cloud, and oriented by both the camera’s frame and the directional vector between the *n* averaged minimum and maximum points. Figure 24 depicts the frame generated for the connector, with appropriate position and orientation for the point cloud’s pose in the environment.

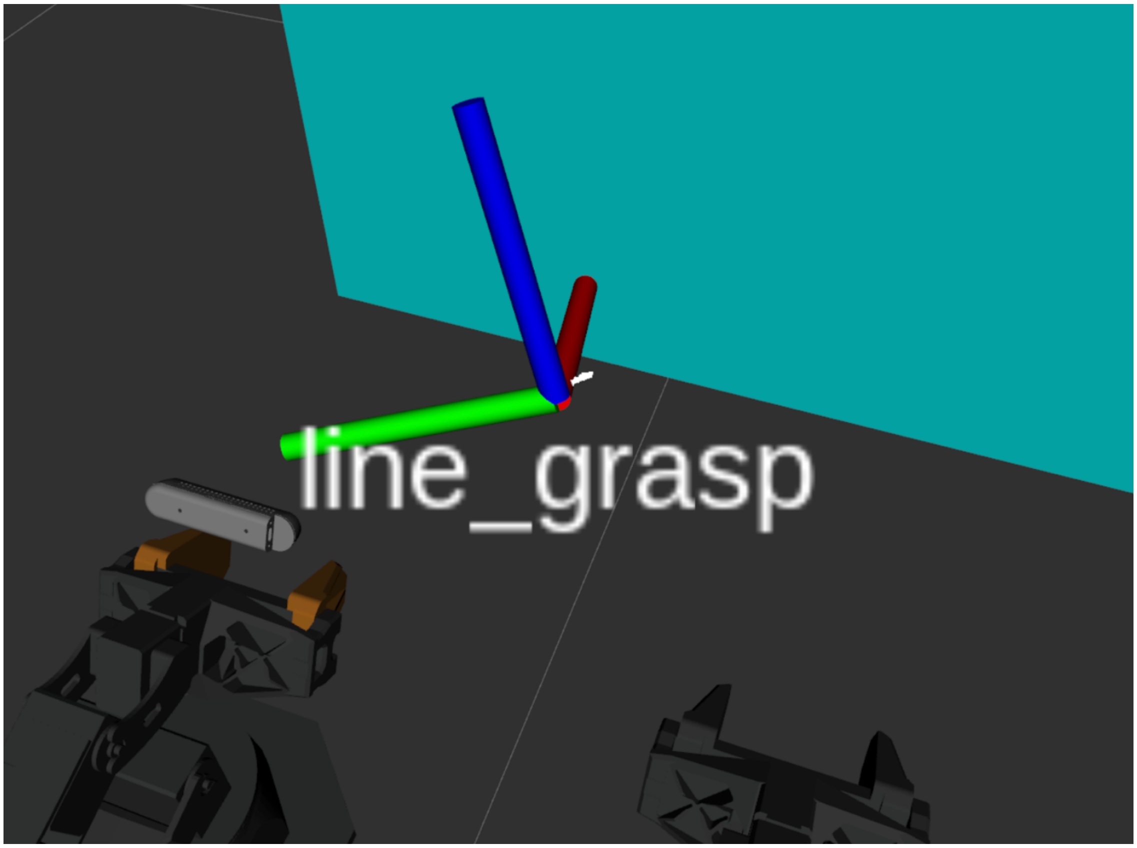


Figure 24: Connector frame generated from HSV segmented point cloud

3.2.3.4 RGBD Image Processing Visual Servoing Pipeline

Visual servoing of the robotic arm, guided by HSV segmentation and point cloud processing, is outlined in the Figure 25 pipeline. The approach begins with the visual sensor publishing both RGB and depth images to a node that performs the HSV segmentation. It calls on the OpenCV library to assist in computer vision processing tasks of image masking and morphological/bitwise operations. These segmented images are remapped by a ROS nodelet manager, which converts the bitwise images to point clouds. The point clouds are then published to two separate nodes, handling position and orientation calculation separately. The latter also adjusts its *X* axis of rotation from the camera's transform, stored in ROS's transform library *tf2*. It also receives the calculated position values through a transform frame and publishes the full completed frame representing the connector end to the system.

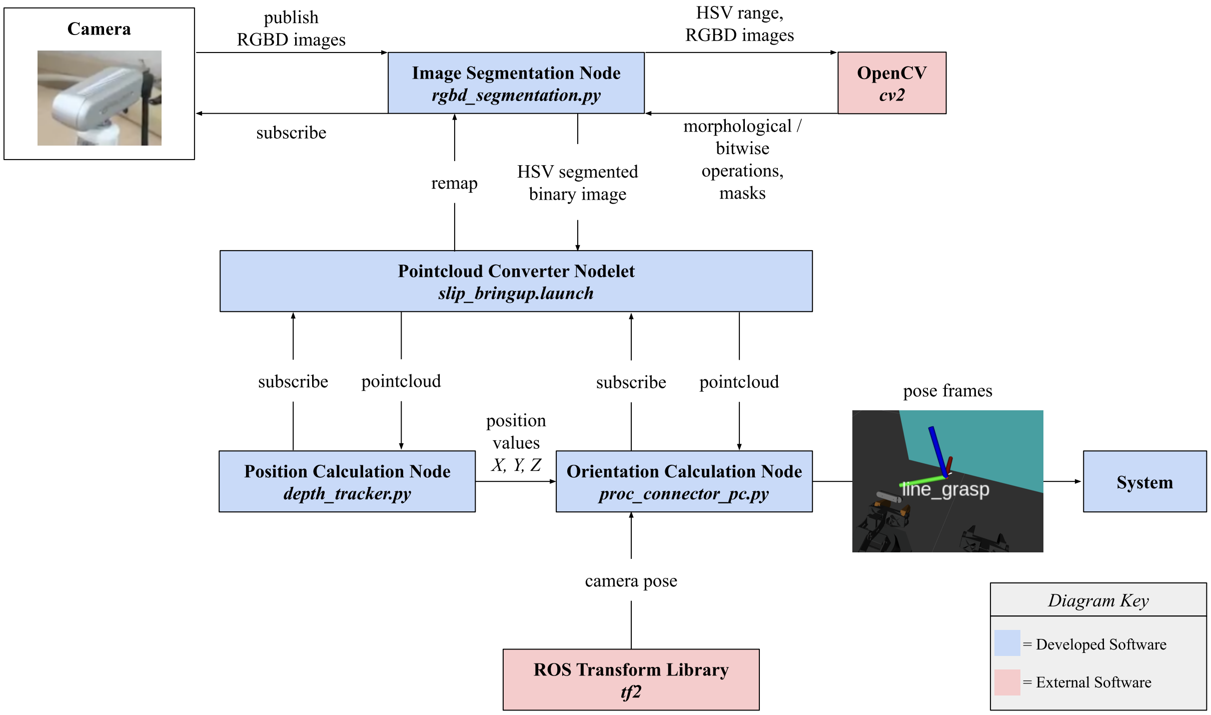


Figure 25: RGBD image segmentation point cloud guided visual servoing pipeline

3.2.4 Machine Learning

3.2.4.1 Image Dataset Configuration

Machine learning can be leveraged for pose estimation through the utilization of a commonly employed task for object detection. Training of any neural network requires an input dataset consisting of a collection of samples which are all belonging to a uniform medium the algorithm is to operate on. Because information about the world, in this work, is captured by cameras, the form of data for this body of work is image files. Alongside the images, the training process requires annotations. This is typically provided in a text document specifying where the objects of interest are located in each image [50]. These annotations assist the neural network in learning what features to recognize by providing the correct locations of entities, adjusting its weights to correlate certain pixel colors and patterns to the classes it must identify [50]. Only one class label is used in this work for representing the cable's connector end. The annotations are drawn manually using the open-source labelling tool CVAT (Computer Vision Annotation Tool) and are defined by rectangular bounding boxes capturing the region of images where objects of interest are located.

To further expand the dataset and intensify the training process for more robust predictions, copies of samples from the dataset can be augmented and added [51]. Augmentations, done by adding noise, transformations, shaders, etcetera, enable the CNN to accurately identify features despite changes to the environment, such as lighting changes or image deformations [51]. The augmentations are designed to emulate the variations in the view that the sensors may encounter. Figure 26 depicts two JPG images from the final dataset, both with bounding boxes included: an unaltered sample from this work's dataset, with connector end in view, as well as a corresponding copy with augmentations applied.

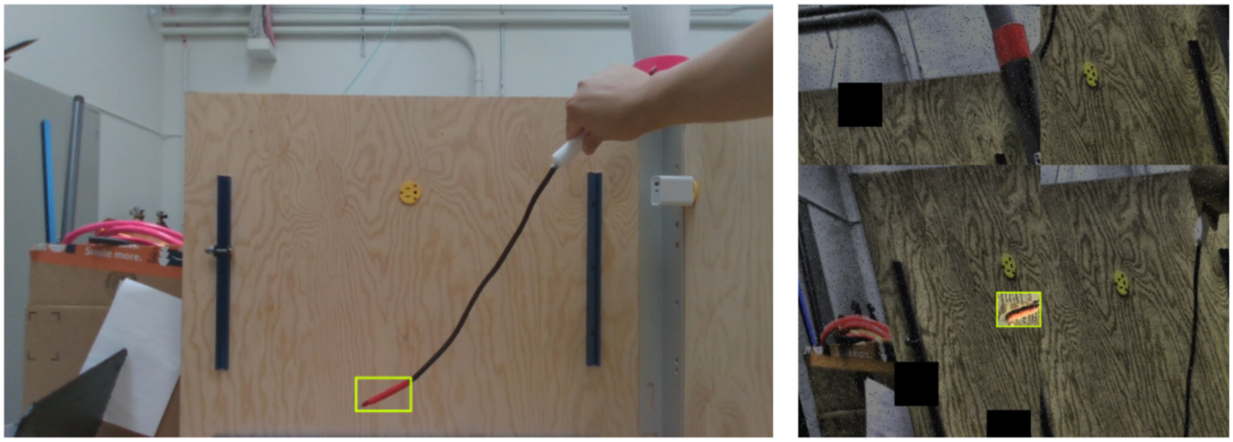
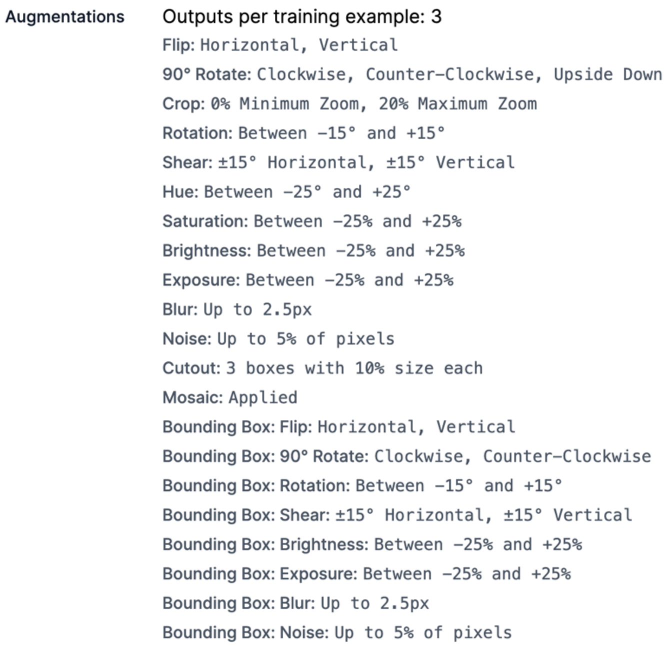


Figure 26: Bounding box annotated (yellow) RGB images of cable with connector end in environment (left) and its augmented copy (right)

3.2.4.2 Convolutional Neural Network Training

Given a dataset of JPG images and corresponding class annotations, a CNN can learn to recognize specific features within these samples to detect and estimate the locations of those same features in new images. The CNN selected for object detection in this work is YOLOv5, an evolution of the You Only Look Once (YOLO) family of real-time object detection models [52]. Training and hosting of the model is accomplished with Roboflow, a computer vision and artificial intelligence tool offering rapid deployment of object detection development with their computer vision services [53]. Two versions of the model exist as a result of training twice under two separate datasets. The first, Dataset A, consists of 122 images all taken under the same lighting conditions of the cable connector in various positions and orientations. This dataset was split such that 85, 25, and finally 12 images were a part of the train, validation, and test sets respectively. The second, Dataset B, similarly captures the connector in diverse poses, but additionally contains some images with lighting variations, introduced by a floodlight pointed at the environment. Also added are augmentations on those captured images, producing a dataset of 292 total images. 255, 25, and 12 images were respectively placed in the train, validation, and test sets. The augmentations applied to the second dataset are fully listed in Figure 27 and were selected based on anticipated conditions that the robotic agent may encounter during live operation, such as lighting changes or unpredictable poses of the connector end introduced by the cable’s free fall or the pose of the camera itself.



Augmentations

Outputs per training example: 3

Flip: Horizontal, Vertical

90° Rotate: Clockwise, Counter-Clockwise, Upside Down

Crop: 0% Minimum Zoom, 20% Maximum Zoom

Rotation: Between -15° and +15°

Shear: ±15° Horizontal, ±15° Vertical

Hue: Between -25° and +25°

Saturation: Between -25% and +25%

Brightness: Between -25% and +25%

Exposure: Between -25% and +25%

Blur: Up to 2.5px

Noise: Up to 5% of pixels

Cutout: 3 boxes with 10% size each

Mosaic: Applied

Bounding Box: Flip: Horizontal, Vertical

Bounding Box: 90° Rotate: Clockwise, Counter-Clockwise

Bounding Box: Rotation: Between -15° and +15°

Bounding Box: Shear: ±15° Horizontal, ±15° Vertical

Bounding Box: Brightness: Between -25% and +25%

Bounding Box: Exposure: Between -25% and +25%

Bounding Box: Blur: Up to 2.5px

Bounding Box: Noise: Up to 5% of pixels

Figure 27: Augmentations applied to the second dataset for training YoloV5 model

The two versions of the model naturally yielded differing metrics as a result of the uploaded datasets. Dataset A produced higher accuracies across all categories over Dataset B, likely due to the controlled lighting and un-augmented conditions under which this version of the model was trained under, therefore making prediction highly accurate. The evaluations of these two models under mAP (average of the Average Precision metric across all classes in a model), precision (rate of correct model's predictions), and recall (percentage of relevant labels successfully identified) are listed in Table 2 [53].

	mAP	**Precision**	**Recall**
Dataset A (unagumented)	99.5%	99.8%	100%
Dataset B (augmented)	84.0%	72.1%	64.0%

Table 2: Evaluation of model version performances under mAP, precision, and recall

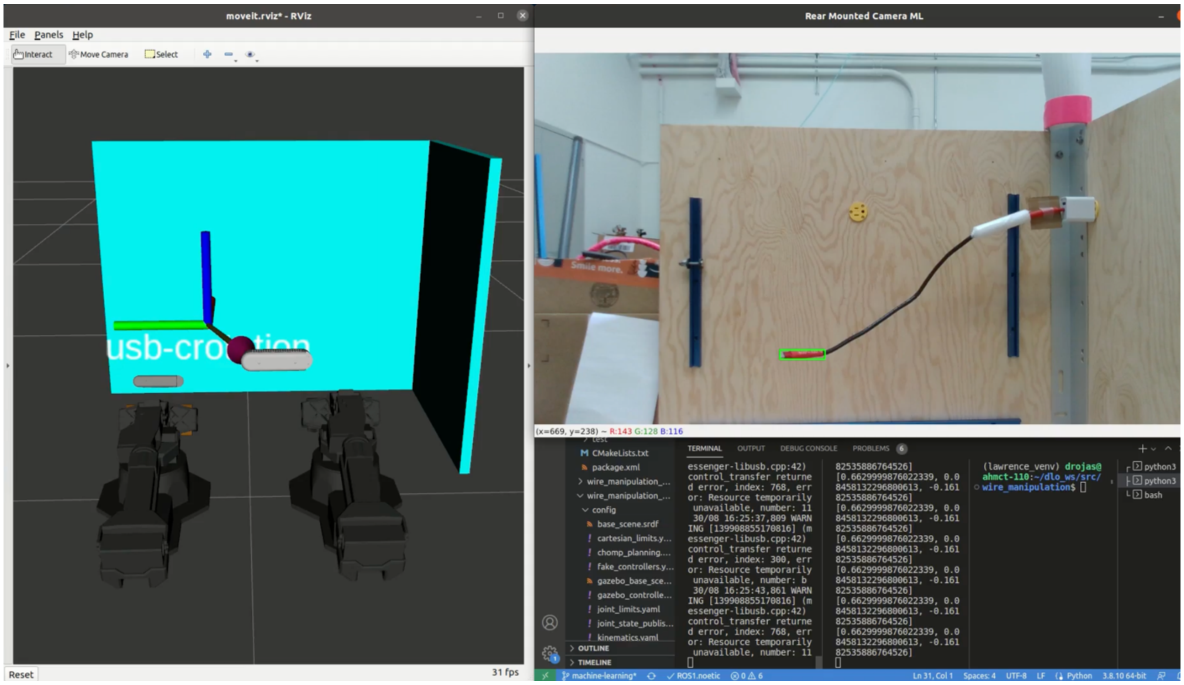
3.2.4.3 Pose Estimation

The trained YoloV5 model is capable of drawing a rectangular bounding box over the region of an image which it estimates the connector end to be. The accuracy will be subject to the presented metric evaluation of each version. A bounding box not only predicts the position of the cable connector relative to the visual sensor, but also confirms the presence of the entity in view of the camera. Leveraging this information enables the identification and pose estimation of the cable connector, using only RGBD images from the camera and the machine learning model.

To visual servo under this technique, the position of the guiding frame must first be calculated. Because rectangular bounding boxes are utilized instead of closer-fitting, oriented bounding boxes, the estimated region captures the entirety of an identified cable connector. All four edges of the bounding box tightly enclose the connector with minimal to no padding, guaranteeing that the center point of the box will represent the center of the connector despite the orientation of the connector within. Therefore, the X and Y coordinates are defined as the center point of the 2D bounding box. Z can then be found by simply looking up the value of this X, Y

coordinate in the corresponding aligned depth image. This approach in obtaining the position is naïve; however, in that it depends on a singular pixel value to determine the value along each axis. Assuming orientation is approximately correct, a single pixel is, in practice, acceptable for determining *X* and *Y* coordinate values due to the width of the gripper. The gripper's grasping zone lowers tolerances and leaves margin for error given how much wider it is than the thin connector end. Considering depth however, the consequences of a single pixel's inaccuracy may result in an under or over shooting of the target grasp. To make the depth value more robust, a cluster of *Z* values from the aligned color-to-depth image can be averaged, similar to the point cloud averaging performed in **Section 3.2.3**.

Although *X*, *Y*, and *Z* are now defined, these are in the coordinate system of the camera images, and not in the camera frame relative to the world environment. The corresponding 3D point in the environment is calculated using the camera's provided developer kit (Intel Realsense SDK) for Python 3, *pyrealsense2*, where the function *rs2_deproject_pixel_to_point* handles conversion with an *X*, *Y* position, depth value *Z*, and intrinsic parameter values [54]. Figure 28 depicts the result of this position calculation, where a bounding box is drawn over the connector present in the camera's view (right) and a corresponding frame (without orientation values) is generated in the RViz view (left).



```
essenger-libusb.cpp:42)
control_transfer returne
d error, index: 768, err
or: Resource temporarily
 unavailable, number: 11
 30/08 16:25:37,809 WARN
ING [139908855170816] (m
essenger-libusb.cpp:42)
control_transfer returne
d error, index: 300, err
or: Resource temporarily
 unavailable, number: b
 30/08 16:25:43,861 WARN
ING [139908855170816] (m
essenger-libusb.cpp:42)
control_transfer returne
d error, index: 768, err
or: Resource temporarily
 unavailable, number: 11
▯
```

```
82535886764526]
[0.6629999876022339, 0.0
8458132296800613, -0.161
82535886764526]
[0.6629999876022339, 0.0
8458132296800613, -0.161
82535886764526]
[0.6629999876022339, 0.0
8458132296800613, -0.161
82535886764526]
[0.6629999876022339, 0.0
8458132296800613, -0.161
82535886764526]
[0.6629999876022339, 0.0
8458132296800613, -0.161
82535886764526]
[0.6629999876022339, 0.0
8458132296800613, -0.161
82535886764526]
▯
```

```
(lawrence_venv) drojas@
ahmct-110:~/dlo_ws/src/
wire_manipulation$ ▯
```

Figure 28: Generated frame (left) for bounding box over identified connector (right)

An orientation must next be determined to completely specify the connector's pose. Fully calculating the three principal rotations about the *X*, *Y*, and *Z* axes proves challenging given only 2D RGB images for a model to review. Although several approaches were considered, a completely machine learning driven pose estimation strategy was abandoned due to overarching research direction and complexity in obtaining orientation values. These proposed techniques are however elaborated upon in the Future Work portion. The machine learning pose estimation approach presented in this work borrows the orientation estimation technique using point clouds converted from segmented images, described in *Section 3.2.3.2*.

3.2.4.4 Machine Learning Visual Servoing Pipeline

The full pipeline of this system employs involving visually servoing the robotic arm using the machine learning approach is illustrated in Figure 29. This procedure begins with the selected camera constantly publishing its intrinsic parameters and captured RGBD images. The depth images are provided to an orientation calculation node, which is responsible solely for calculating *X, Y, Z,* and *W* for the connector. These values, along with RGBD images and intrinsic parameters from the camera, are sent to a central connector tracking node responsible compiling the information and estimating the full pose. This node first sends the RGB images and intrinsic parameters to Roboflow via an API call, in return receiving predictions made on that image and their coordinate positions in 2D. The prediction coordinates are then forwarded to the camera's developer kit, where it is converted to a 3D coordinate position relative to the world frame. With position and orientation, the tracking node generates a full pose for the connector and publishes it to the system for use with the robotic arm.

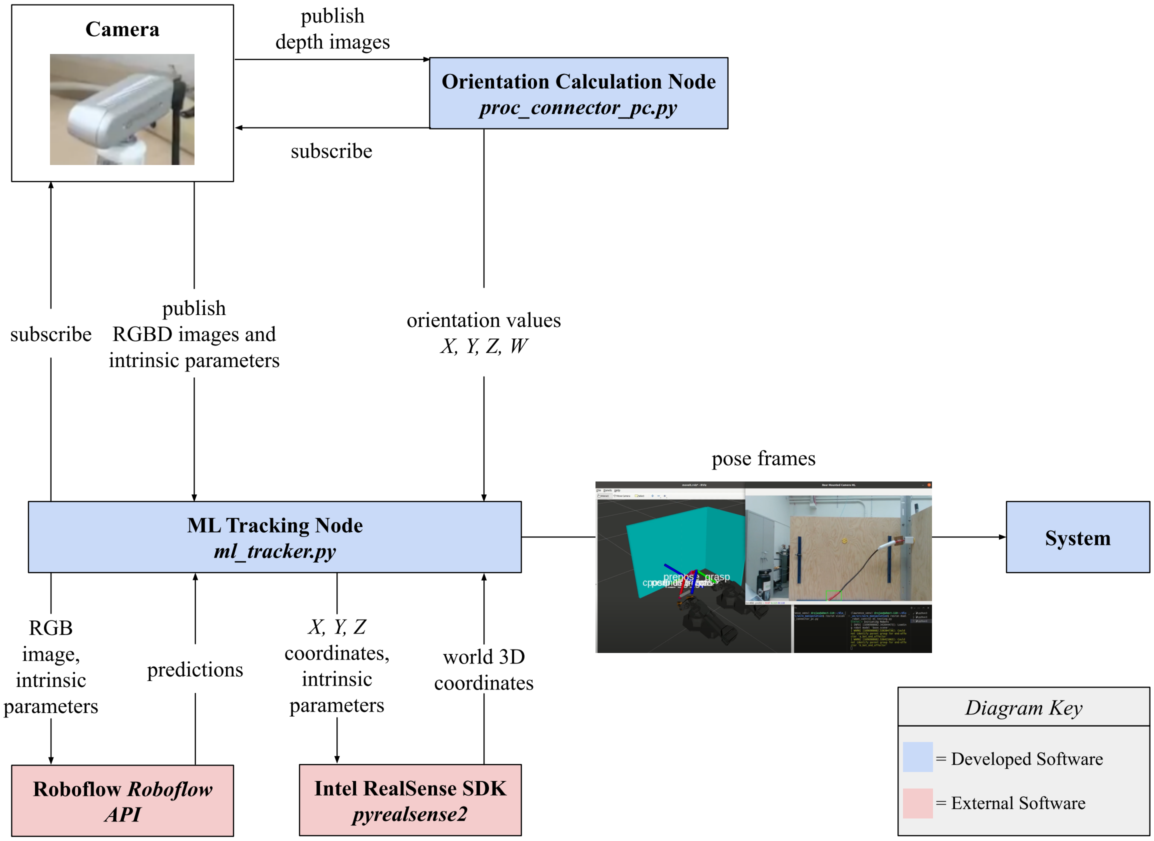


Figure 29: Machine learning and depth image guided visual servoing pipeline

3.3 Management of Robotic Agent

3.3.1 Motion Planning and Execution Architecture

In order to complete its assigned objectives, the robotic agent must be commanded to meaningful poses in its workspace. As this system employs only visual sensors, visual servoing is the principal guiding strategy for robot motion control. A simple visual servoing strategy is employed wherein the system defines a target frame that the selected arm's end effector is sent to, handled by MoveIt's motion planning and execution. This strategy, portrayed through system architecture in Figure 30, is utilized for both cable connector end grasping (**Section 3.4.2**) and search pattern maneuvers (**Section 3.4.3**).

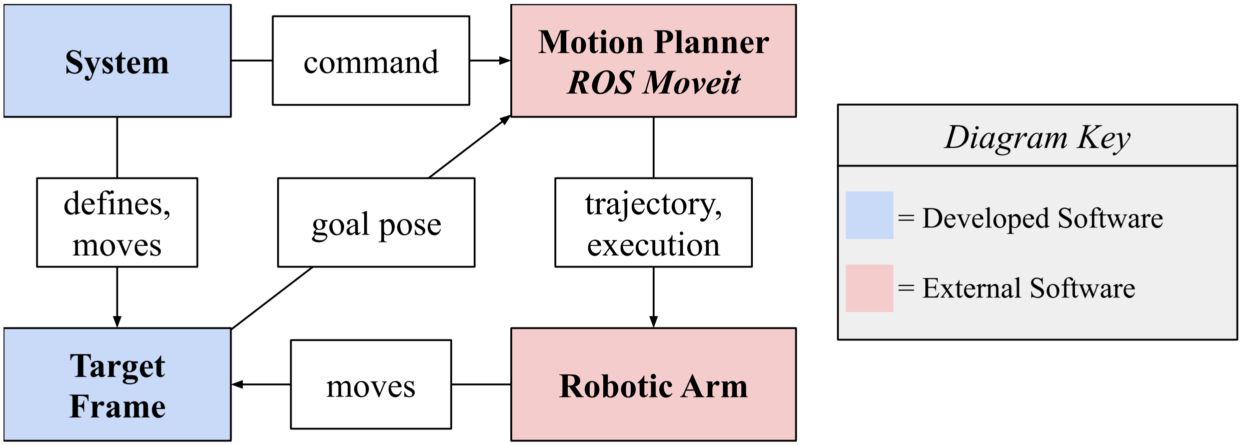


Figure 30: System architecture for robotic agent motion planning and execution

3.3.2 Robotic Arm Task Allocation

The robotic agent in this system is composed of two separate arms capable of individual commands. In order to complete the system's objectives, only one arm is necessary for manipulation of the DLO. In the event that a cable slip anomaly results in a complete loss of vision of the DLO’s connector end, the second arm with an arm mounted camera conducts a search of the environment in an effort to relocate it behind occlusions.

In this thesis, the left and right arms are assigned to camera search and DLO manipulation respectively. Depending on the location of the cable in the environment, it may be more optimal to have a third camera affixed to the right arm such that both arms are capable of performing either role. In a more traditional robotic system, this decision is typically made by a higher-level task planner, which is neither developed nor used in this system. In a more flexible application of the technology presented in this thesis, the roles of the arms may switch interchangeably depending on the current status of the environment. The intention of this research is to ultimately integrate the capabilities developed here into a larger, more intelligent system. Therefore, a predetermined allocation of roles for the arms are set and maintained throughout this work, as the focus is to demonstrate the capabilities of the visual sensor guided arms themselves.

3.4 Slip Anomaly Detection and Response

3.4.1 Anomaly Monitoring, Quantification, and Detection

As the robotic agent is executing its assigned tasks, a DLO in the environment is nominal if its state, characterized by its position, is known to the system. This nominal state can be verified by monitoring the cable through the system’s visual sensors. In this work, the connector end of a cable is tracked instead of the entirety of the DLO’s length. Assuming the cameras have an unobstructed view of the connector, the system is capable of estimating its pose using the three approaches described in **Section 3.2**: fiducial markers, machine learning, and RGBD image processing.

Pose estimation grants live positional information, which is compared against where the system expects the connector to be at that point in time. This idea serves as the foundation of how a slip is detected. In the presence of a robotic arm's working environment, a nominal DLO is either in one of two modes. The first is at rest, not moving in the environment and expected to remain at its untouched position. The second is under manipulation by the robotic agent, wherein the connector should be in motion relative to the end effector managing it. Within the grasp of the end effector, it should be expected that the connector end maintains a consistent distance (within some margin) from the robotic arm. Because both the end-effector, p, and the cable connector end, q, exist in 3D space, the distance between them is represented by the three-dimensional Euclidean distance formula, given by Equation 6 [55]:

$$d(p,q) = \sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2 + (p_3 - q_3)^2} \tag{6}$$

Using only the visual sensors, a slip of the cable from the robotic arm's end effector can be defined when this distance is surpassed by a set threshold value. The connector surpassing the slip distance threshold is equivalent to a connector existing outside the possible positional range to still be considered within grasp by the end effector, thus denoting a slip anomaly. Constant monitoring of the cable by the visual sensor allows for tracking of the connector's current location and flagging a slip to the system if the set limit is exceeded. Figure 31 illustrates the 3D Euclidean distance measurement (green) between connector and grasping arm's end effector. This is visualized in both nominal manipulation (left) and off-nominal slip (right) operation.

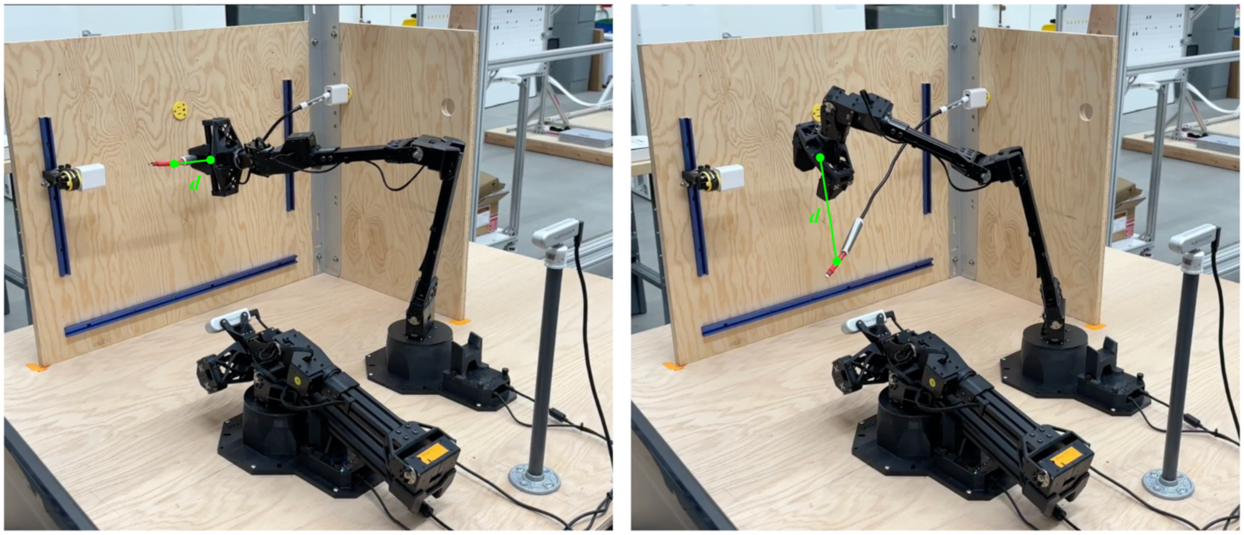


Figure 31: 3D Euclidean distance measurement (green) between connector end and grasping arm's end effector in nominal (left) and off-nominal (right) operation

3.4.2 System Control Flow for Off-nominal Slip Response

Once a slip of the cable connector from the grasping arm's end effector is identified by the visual sensors, the system must respond to the slip anomaly in order to reestablish the nominal state. A return to the nominal state is achieved by restoring the DLO to its position just prior to its slip, thus requiring the robotic agent to conduct a three-step recovery response consisting of locating, retrieving, and maneuvering the cable.

Anomaly response begins with relocating the slipped connector end. Following a slip, the cable end could potentially settle anywhere in the environment. The primary camera mounted behind the robotic arms is consulted first, due to its panoramic view over the entire workspace, for relocating the connector end [13]. Failure to identify the DLO suggests the loose cable end has landed behind an entity obscuring it from view of the rear camera. Whether the obstruction is another DLO, obstacle in the environment, or even the robotic arms themselves, the connector end still requires relocating. Instead of sending the robotic agent to move probable blocks in the scene [42], this thesis proposes conducting a search for the connector end with an auxiliary camera mounted to the wrist of the second robotic arm which is not utilized in grasping tasks. The search algorithm developed is further described in **Section 3.4.3** and is only deployed if at least one of the following three conditions fail. These conditions ensure the connector end is truly obstructed and lost from field of view in a live monitoring of the scene, thus avoiding initiation of a search on behalf of a system or sensor error. These three conditions are as follows:

1. The frame for the connector end exists. This condition is met if the frame exists in ROS transform library tree.
2. The last active frame for the connector end is within a preset duration of the current system time. The timestamp of the frame is compared against the current system time and is considered active if the time difference is within an active threshold.
3. The frame for the connector end has been active over a preset duration. This condition is checked by exploiting the inherent noise from the camera sensors. This noise stems from the effects of light intensities, viewing angles, and reflections from the object or scene [7]. The noise results in minor pose estimation differences, minute changes to position or orientation that are still accurate reflections of the connector end's real location in the environment. Figure 32 illustrates this noise as a result of the minor difference in pose between two captures of the RViz simulation view. It is taken only one second apart but from the same viewing angle. The connector frame is deemed active if this noise has resulted in minor pose estimation changes over the course of a set interval.

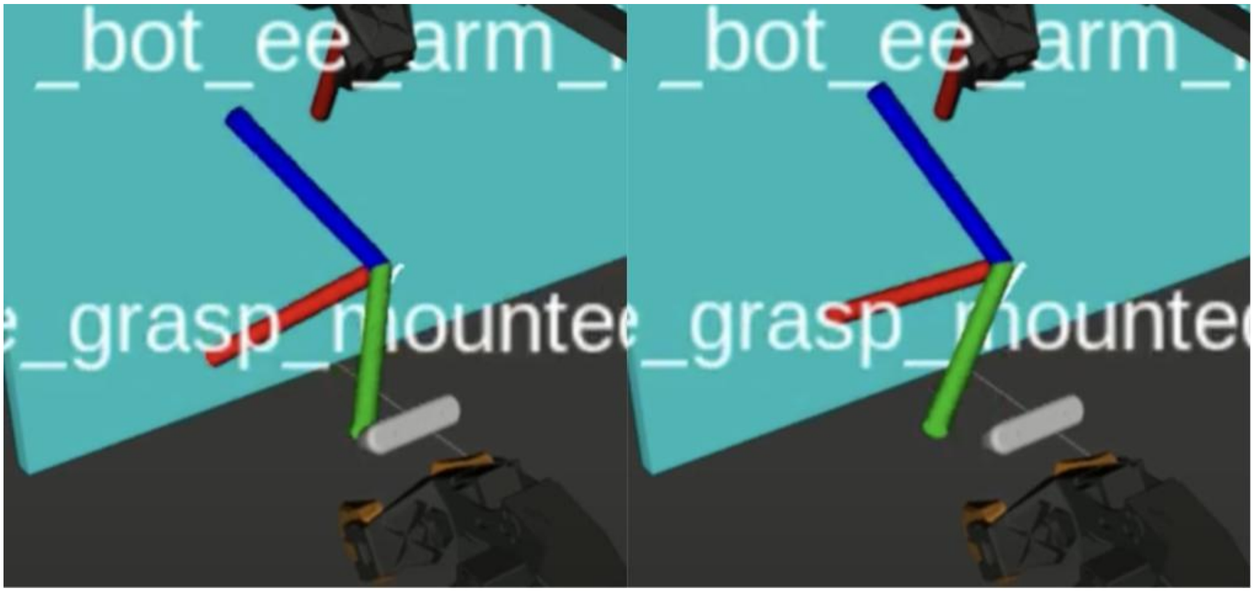


Figure 32: Visualization of noise affecting connector end pose estimation (RGBD image processing method), set one second apart from the same viewing angle

Regardless of how the system locates the connector end, identification and pose estimation are performed via the selected method described in **Section 3.2**. Once the connector end is relocated by the system, it must next retrieve the DLO by regrasping the connector end. Assuming successful pose estimation in view of the system's visual sensors, the connector end is now represented in the system by its own frame. The ROS transformation library *tf2* is capable of translating the coordinate system of the connector frame to that of the robotic arm so that the agent can maneuver to it [35], [36]. An abstracted view of this relationship is depicted in Figure 33, where the connector frame is first transformed to its parent camera frame of reference, then the world reference frame the robot's frames exist.

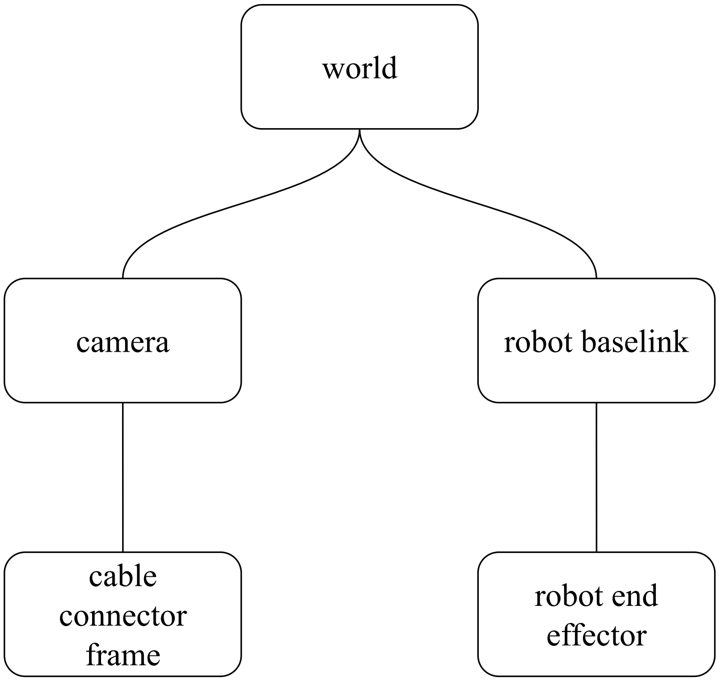
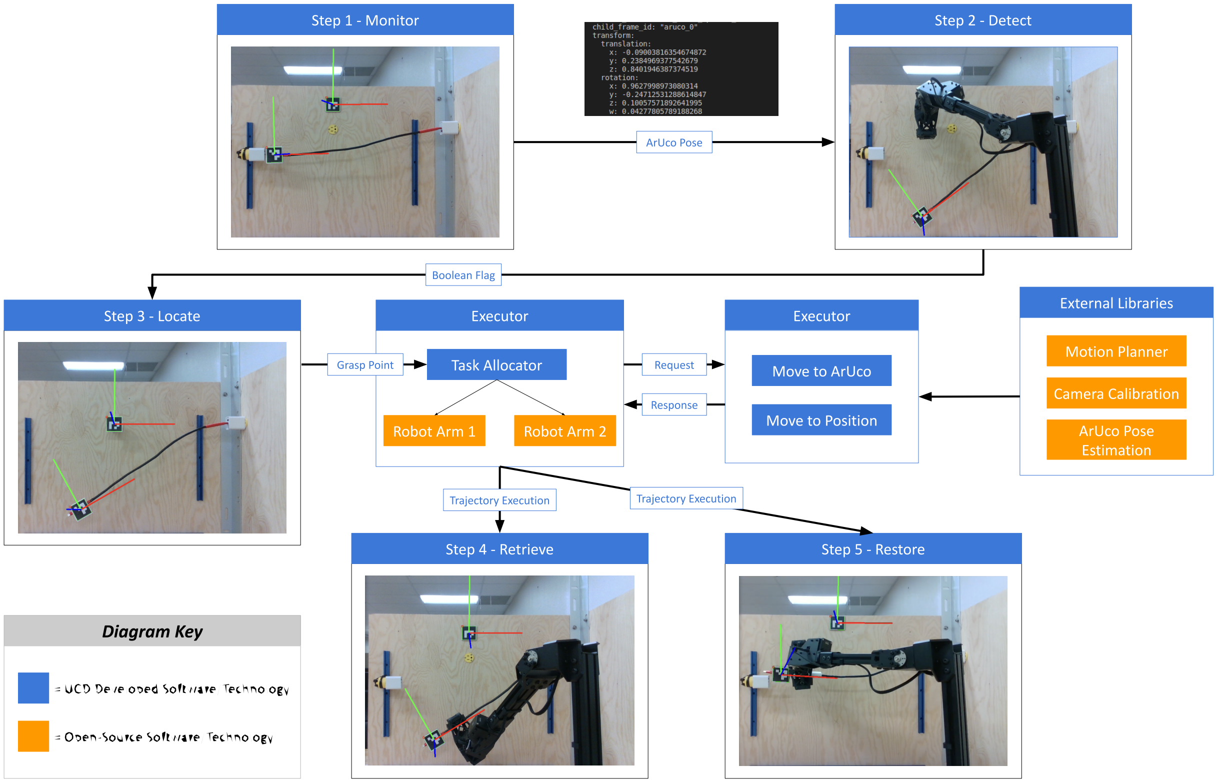


Figure 33: ROS transformation tree depicting parent child relationships between entities

Following the transformations conducted by *tf2*, the pose of the connector end is in a format comprehensible to the robotic agent. A motion plan is next made and executed by MoveIt, commanding the robotic agent to maneuver to the connector end's pose for grasping. This work assumes no obstructions are present between the grasping arm and connector end during the retrieval process. In the presence of further occlusions, more robust motion planners with avoidance strategies may be employed. Once the grasping arm reaches the pose of the connector, a grasp attempt is made, restoring the loose DLO end to a known state. Another motion plan is finally planned and executed to restore the grasped end to its original position. The Euclidean distance measurement monitoring (**Section 3.4.1**) between the connector end and arm end effector is deployed again here to ensure another slip does not occur enroute to the cable's destination. The underlying system architecture of this monitoring and retrieval process, using fiducial markers as the pose estimation strategy, is summarized in Figure 34.



```
child_frame_id: "aruco_0"
transform:
  translation:
    x: -0.09003816354674872
    y: 0.2384969377542679
    z: 0.8401946387374519
  rotation:
    x: 0.9627998973080314
    y: -0.24712531288614847
    z: 0.10057571892641995
    w: 0.04277805789188268
```

Figure 34: System architecture for slip anomaly response with fiducial markers

A full robotic behavior tree diagram mapping the sequence of actions taken by this system is depicted in Figure 35.

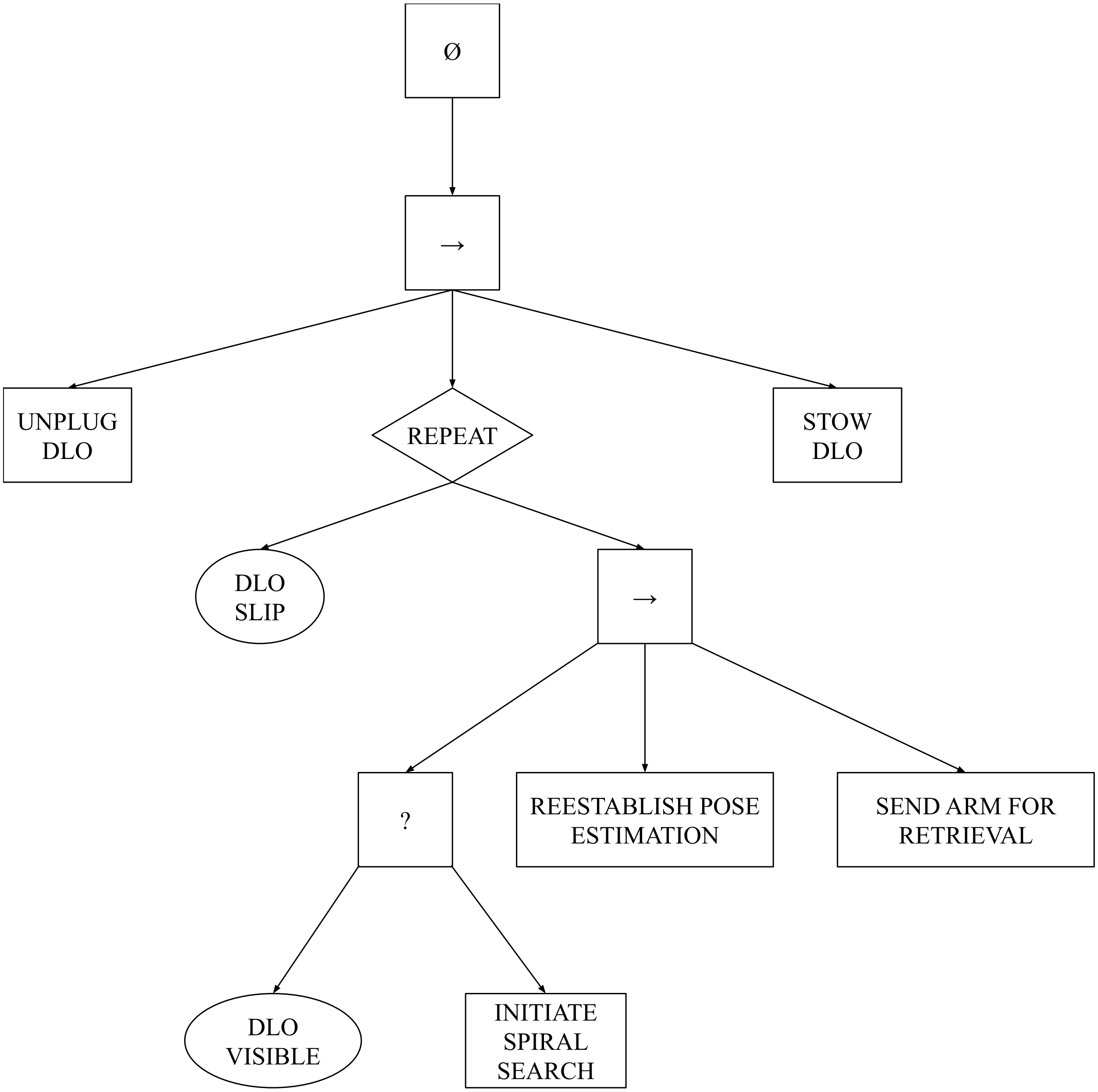


Figure 35: Behavior tree representing states and actions of this robotic system.

3.4.3 Environment Spiral Search Pattern

3.4.3.1 Search Area Optimization

In the event that the primary rear mounted camera fails to locate the loose connector end and the connector frame is deemed inactive, the second robotic arm which is not utilized in DLO manipulation, with a secondary camera affixed to its wrist, is deployed in a search of the environment. Moving makes this robot mounted visual sensor dynamic, opening up views to the system previously not observed by the static visual sensor alone [14].

The search's objective is to identify the connector end regardless of its location in the workspace, and conduct pose estimation with its arm mounted camera. Searching the entirety of the workspace, while thorough, proves highly inefficient when properties/characteristics of the cable may assist in narrowing the search to specific regions of the scene. These cable properties/characteristics are as follows:

1. The connector end's movement is restricted by the length and stiffness of the cable. Certain regions of the scene can therefore be ruled out as the cable cannot physically reach those positions. This area of search interest can be represented by a cone, illustrated in Figure 36. Regions of this cone vary in probability in terms of containing the loose connector end.
2. Search of the probability cone can be optimized by first inspecting areas where the loose connector end was last observed. Doing so minimizes the time in finding the connector, as it is more probable the entity has settled near its last location prior to entering an unknown state [43], [44].

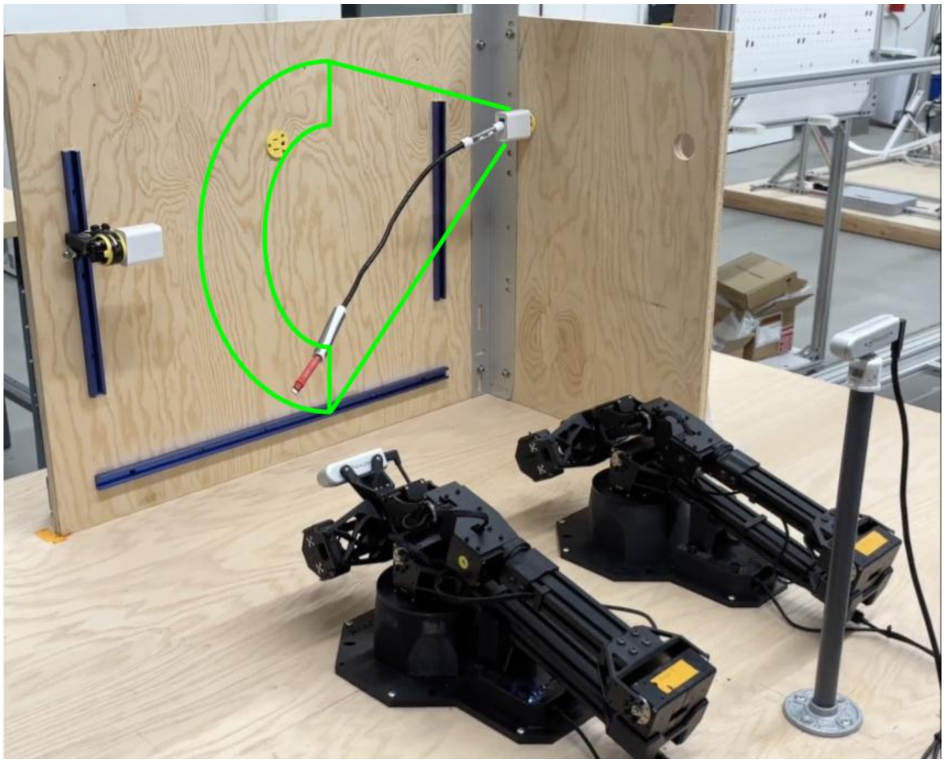


Figure 36: Cone representing the most probable region a loose connector end may settle

3.4.3.2 Spiral Search Strategy

A spiral shaped search pattern ("Spiral Search") was therefore developed to scan the loose DLO's most probable settling regions The search included the entire workspace in the event the lose DLO is not located in the most probable regions. The pattern is initiated where the connector end was last observed. The robotic arm's end effector with the camera is then sent in a circular path imitating an algorithmic breadth-first search, concentrically moving outwards as the search expands – hence spiral. Figure 37 maps a typical course (red arrows) the arm follows, as viewed from the rear mounted camera. The search begins where the connector was last observed (red star). Pauses are made at set checkpoints (red dots) along the route to scan the scene for the connector end. When the search pattern traverses out of bounds (blue), outside the camera's Field of View (FOV), the spiral path resumes at the next valid point. The end effector is flatly oriented at each valid point, giving the camera a view parallel to the back wooden panel.

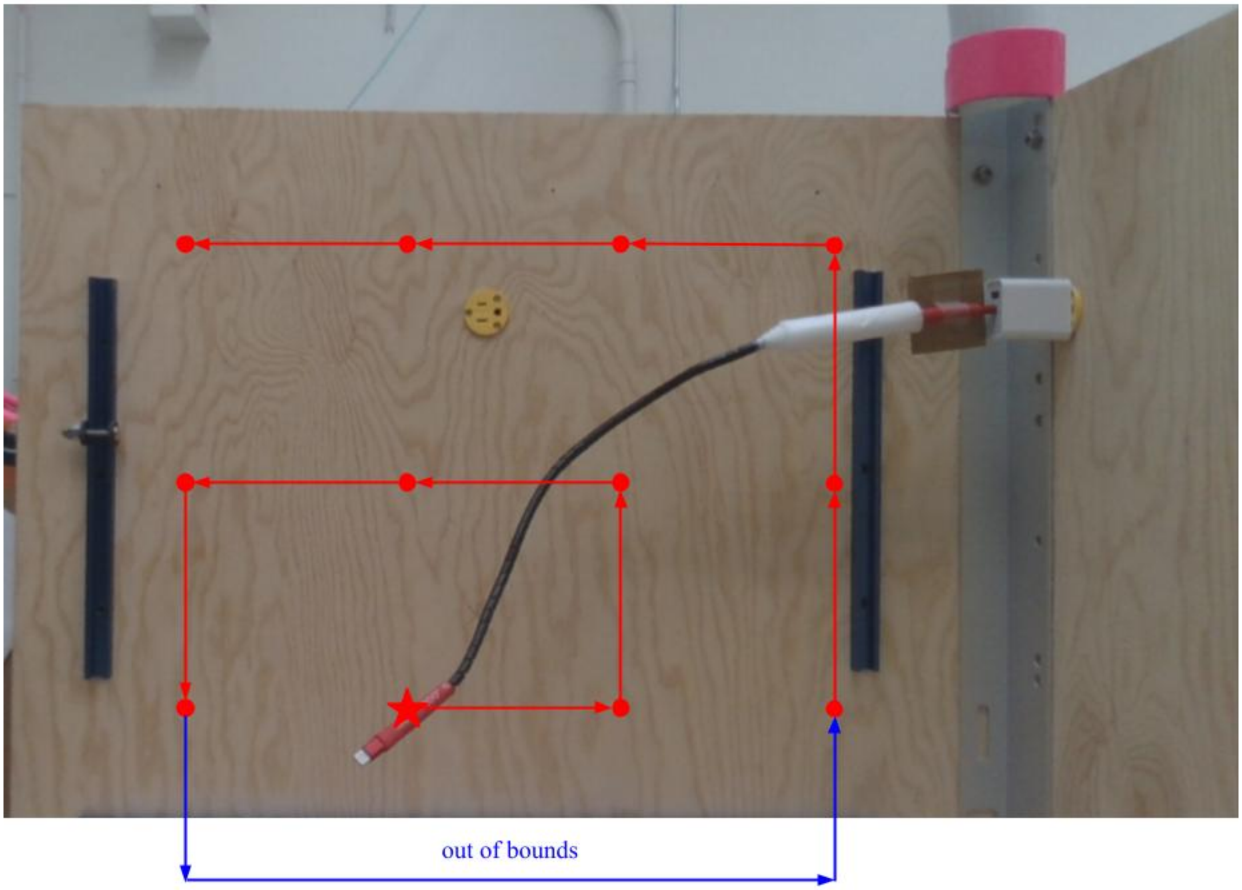


Figure 37: Spiral search path with primary stopping points (dots), initiated from the connector's last known location (star)

At each checkpoint, subpoints (green dots) are also defined and inspected to ensure an area is thoroughly scanned prior to moving onto a different portion of the search route, as illustrated in Figure 38. Eight subpoints encircling each primary point are scanned, with the end effector oriented toward that area's central point (green arrows) as a means of searching around any obstacles present in that subregion. The distances between the red primary points along the *X* and *Y* axes is 0.25 meters. The first green sub point is placed 0.125 meters right of its corresponding primary point, with the following 7 placed 0.125 meters horizontally and vertically from the previous. At each sub point, the searching robotic arm faces the primary point at an angle of pi/6. These distances were chosen based on the dimensions of the connector end relative to the total area of the environment. The rotation was selected following testing of views around the obstacles designed in **Section 3.1.1**, resulting in the choice of the value which will provide a general angle for searching.

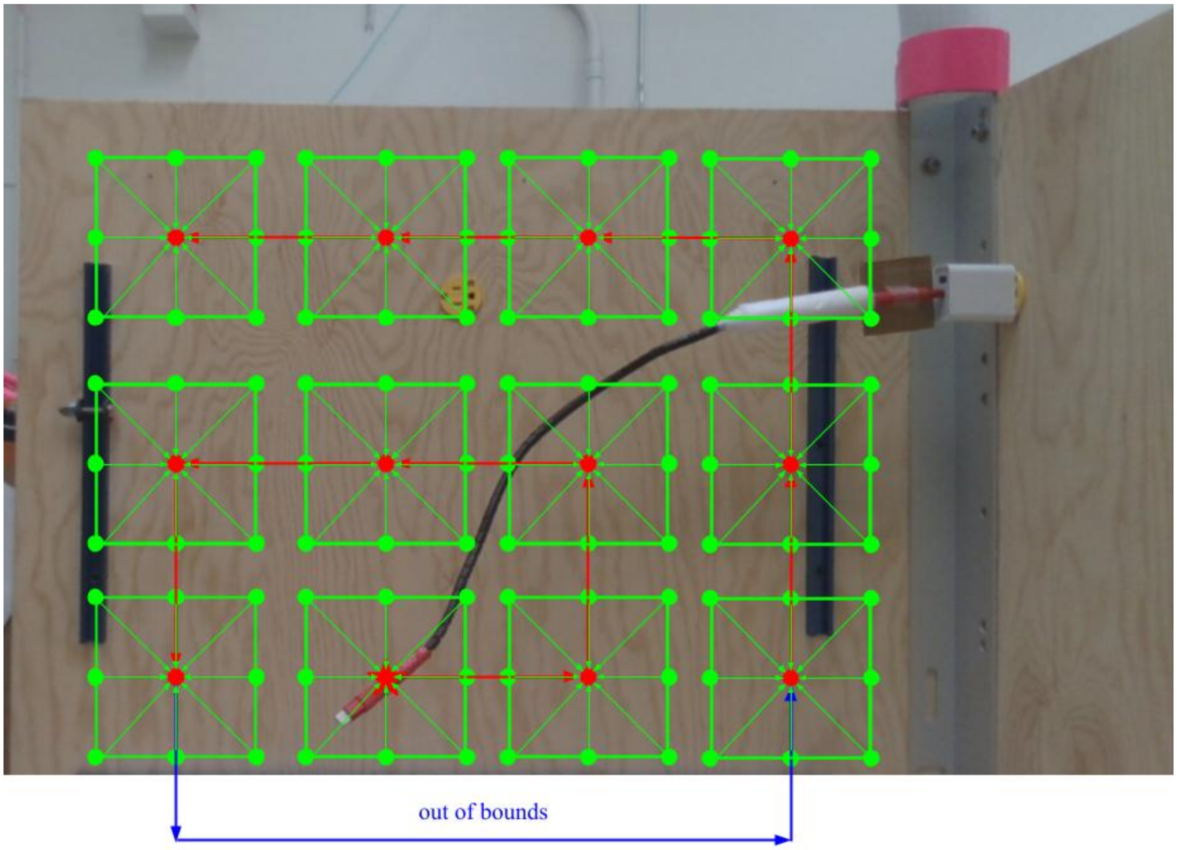


Figure 38: Spiral search path with eight stopping subpoints (green dots), and direction vectors (green arrows) denoting end effector orientation at each subpoint

Search in this manner is both robust and efficient through this systematic approach to scanning the scene. Efficiency is achieved by exploiting information about the connector, beginning the search where the end is most likely to settle [43], [44]. Robustness is achieved by comprehensively scanning each portion of the spiral search path (subpoints) and carrying out the search over the entire scene in this systemic manner until the connector is relocated. In a simulation view of the robotic agent, Figure 39 depicts the searching arm at two stages of the spiral search: primary point with flat orientation parallel to the back wooden panel (left) and subpoint oriented towards the central primary point (right).

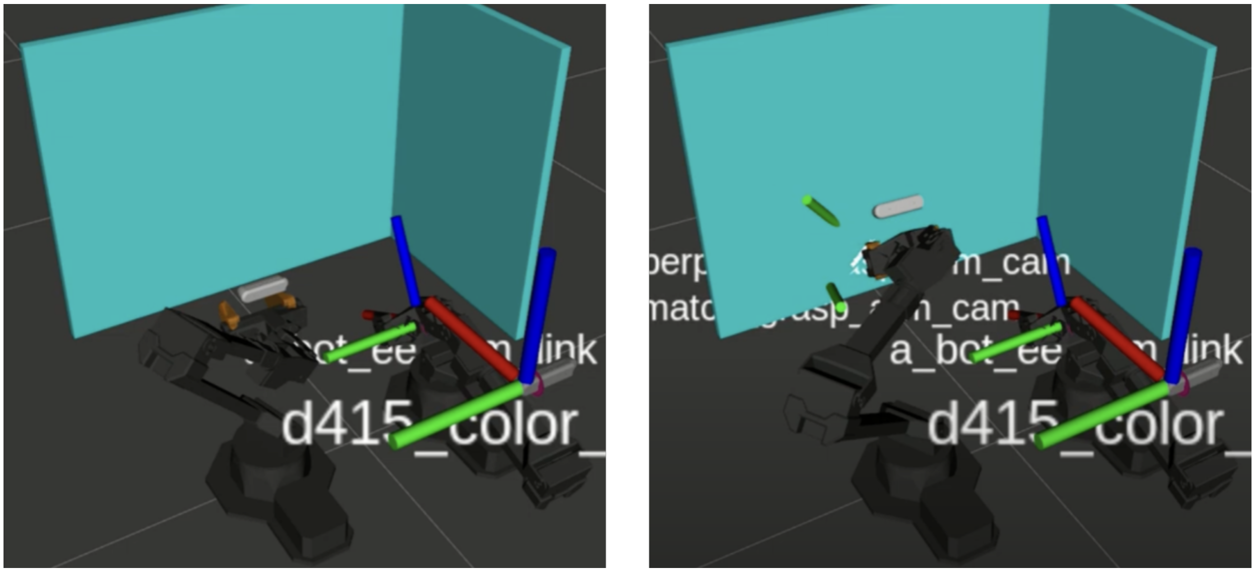


Figure 39: Searching arm at two stages of the spiral search pattern: primary point with flat orientation (left) and subpoint oriented towards the central primary point (right)

3.4.3.3 Software Implementation

The spiral search pattern is implemented as a single moving frame that iteratively traverses through the main points and the sub points in the path. The frame, representing a point to be scanned, begins at the connector end's last known position. MoveIt then motion plans and executes a trajectory that commands the searching arm and camera to that frame's pose. A scan is initiated at that point, and if the connector end is not located, the system updates the search frame for the next point. The algorithm makes the necessary adjustments to the search frame by editing the position and orientation of where the end effector must next go. This process is repeated until the connector end is found or the search terminates. Figure 40 depicts a simulation view of the search frame at its initial pose, positioned at the connector end's last known location with no orientation.

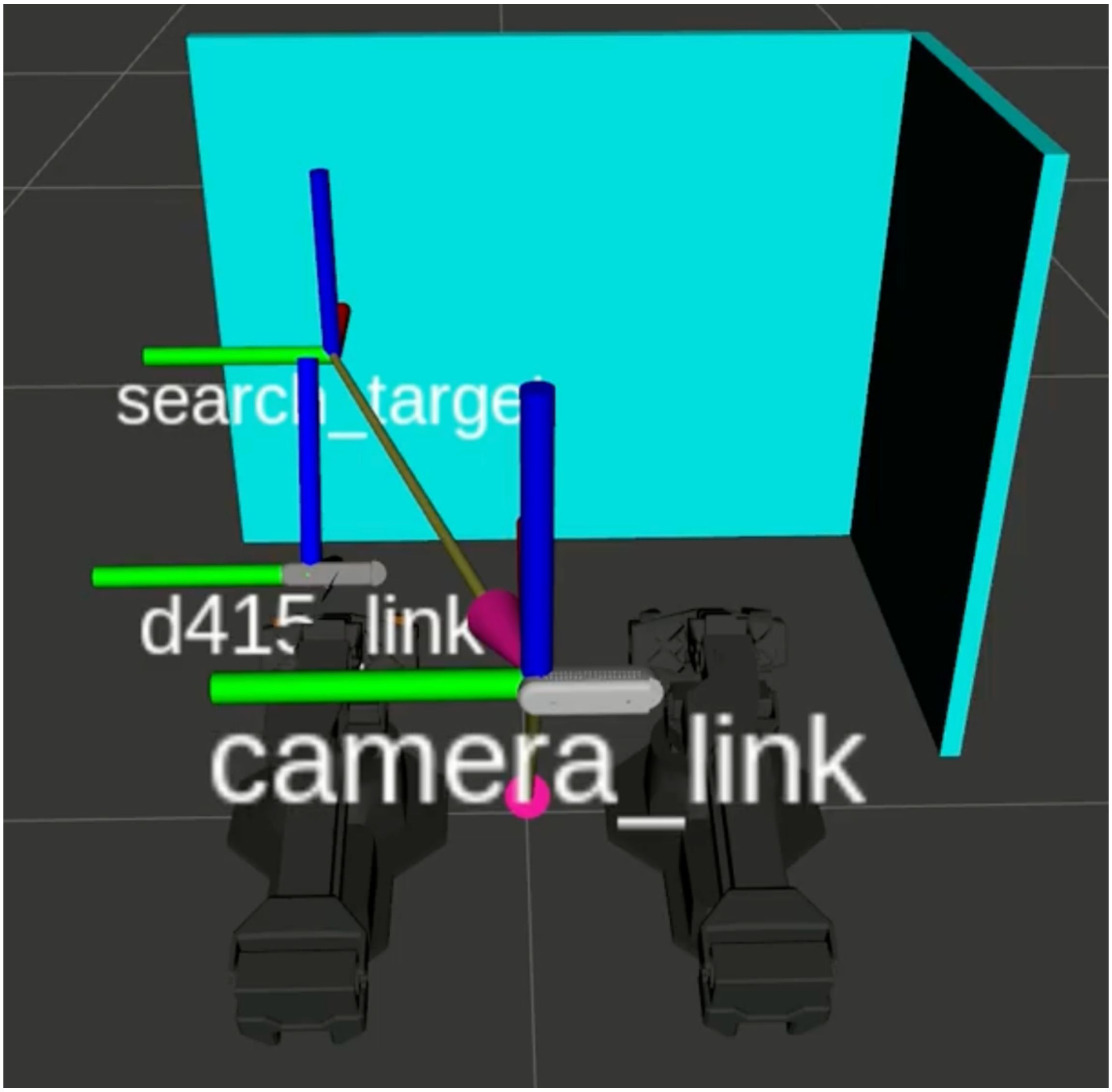


Figure 40: Search frame at its initial pose, positioned at the connector end's last known location with zeroed orientation

4 Results

In this thesis, a complete robotic system employing a selection of methods is proposed to resolve the research problem of DLO slip anomaly identification and response. The solution, requiring a robotic system perform pose estimation, monitoring slip detection, and performing retrieval of a DLO connector end. The system developed employs two Trossen ViperX 300 S robotic arms and Intel RealSense d415 and d435i cameras. To examine the effectiveness of the solution offered by this system, a testing environment was developed to perform a series of three technique validation experiments to evaluate the capabilities of the developed pose estimation approaches, slip detection strategy, and spiral search algorithm. These individual parametric studies emphasize repeatability, where validity of the solution is reinforced through consistent outcomes instead of relying solely on value-based comparisons to determine accuracy. Given low noise and stable environmental conditions, each technique is statistically expected to succeed based on consistent results obtained from many repeated trials. Following technique validation, these newly acquired system capabilities are assembled into an end-to-end demonstration of slip anomaly detection and response, addressing the central research problem of this thesis.

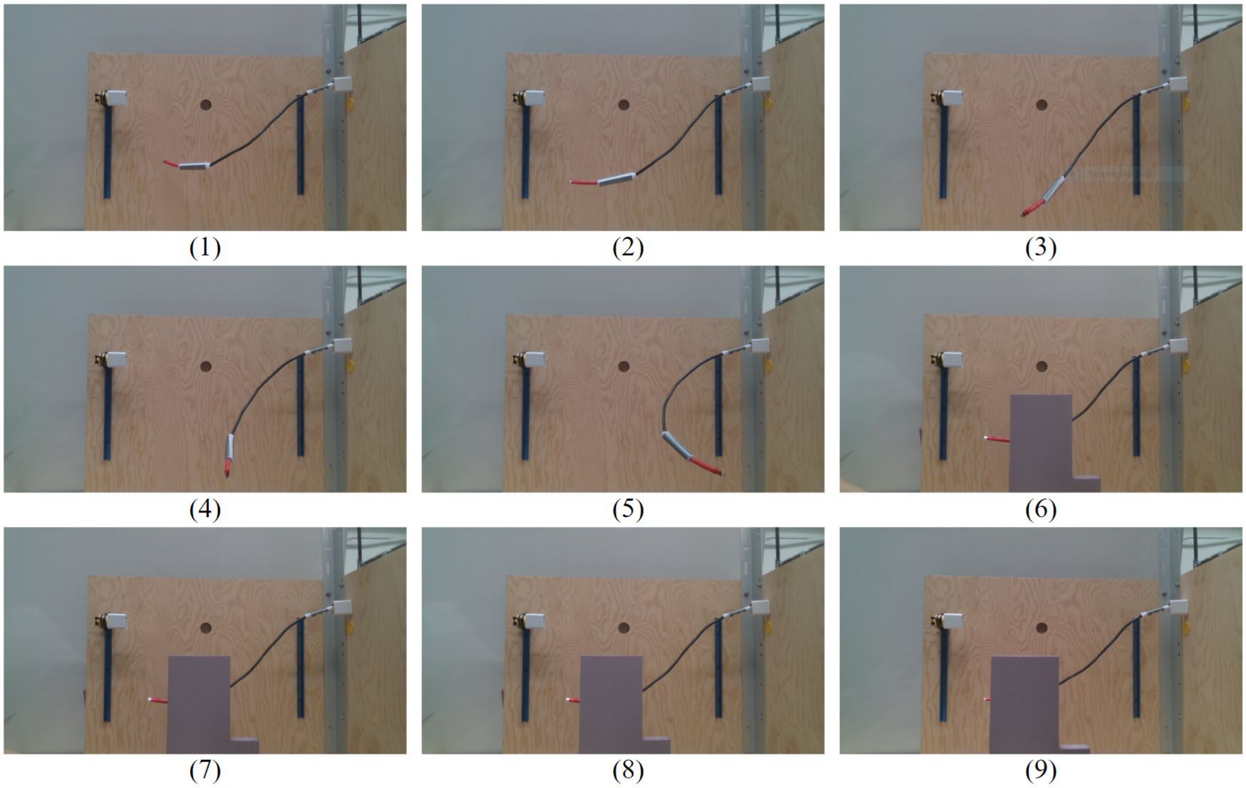
4.1 Techniques Validation

4.1.1 Identification and State Estimation

4.1.1.1 Experiment Overview

Three pose estimation techniques were developed and presented in this thesis for the purposes of identification, state estimation, and monitoring of a DLO connector end. To validate the practicality and accuracy of each approach's ability to estimate position and orientation, the following experiment was formulated. One end of the DLO described in **Section 3.1.1** is connected to the right panel, while the other is free in the workspace.

Over each trial of the experiment for each pose estimation technique, the length of the cable is set in different configurations that vary the connector end's pose. These configurations were chosen based on realistic settled DLO poses following a slip in zero gravity, factoring in droop as a result of the physical constraints posed by testing the weighted end in a standard Earth gravity environment. The system then attempts to identify and estimate the connector end's position and orientation. A total of 9 configurations are defined: 5 of which are unobstructed and a final 4 that are partially obscured from the view of the rear camera. For stability in repeated testing over all trials, only the rear-mounted camera was employed in this experiment as pose estimation is performed in the same manner between both visual sensors. The connector end under configuration 3, chosen for its high identification and grasp success rate across all configurations' metrics, is utilized in the partial occlusion configurations by the taller obstacle described in **Section 3.1.1**. These four trials partially occlude the entire connector end by roughly 10%, 30%, 50%, and 70%. In doing so, each approach's ability to work properly, in the face of obstacles, without supplemental algorithms, is also tested. In summary, for each of the 3 pose estimation methods, 9 configurations are tested in which 5 are obstruction free and 4 are occluded in increasingly more coverage of the connector end. All 9 configurations utilized in this experiment are displayed in Figure 41 as seen by the rear mounted camera. Each trial is attempted three times, resulting in a total of 27 trials per pose estimation method and 81 trials in total for this technique validation experiment.



(1) (2) (3) (4) (5) (6) (7) (8) (9)

Figure 41: Pose estimation configurations for technique validation experiment

Three metrics are collected per trial to assess each pose estimation approach. First, the capacity to identify the connector end is recorded as either a success or failure, determined by the system's generation of a frame for the connector end within 5 seconds. If identification was successful, position estimation is gauged next by measuring the Euclidean distance between the position of the connector end in the actual environment and its estimated position in the system. This is accomplished by first producing a ground truth, a measurement of the actual connector end's position relative to the environment's origin which would be the world frame origin. This ground truth is found by manually measuring the distance along the *X*, *Y*, and *Z* axes of the connector end from the world frame's pose in the environment. The process for determining the ground truth for configuration 3 is depicted in Figure 42, where the *X* (left), *Y (left)*, and *Z* (right) coordinate values of the connector end position (green) is measured relative to the world frame's pose (red).

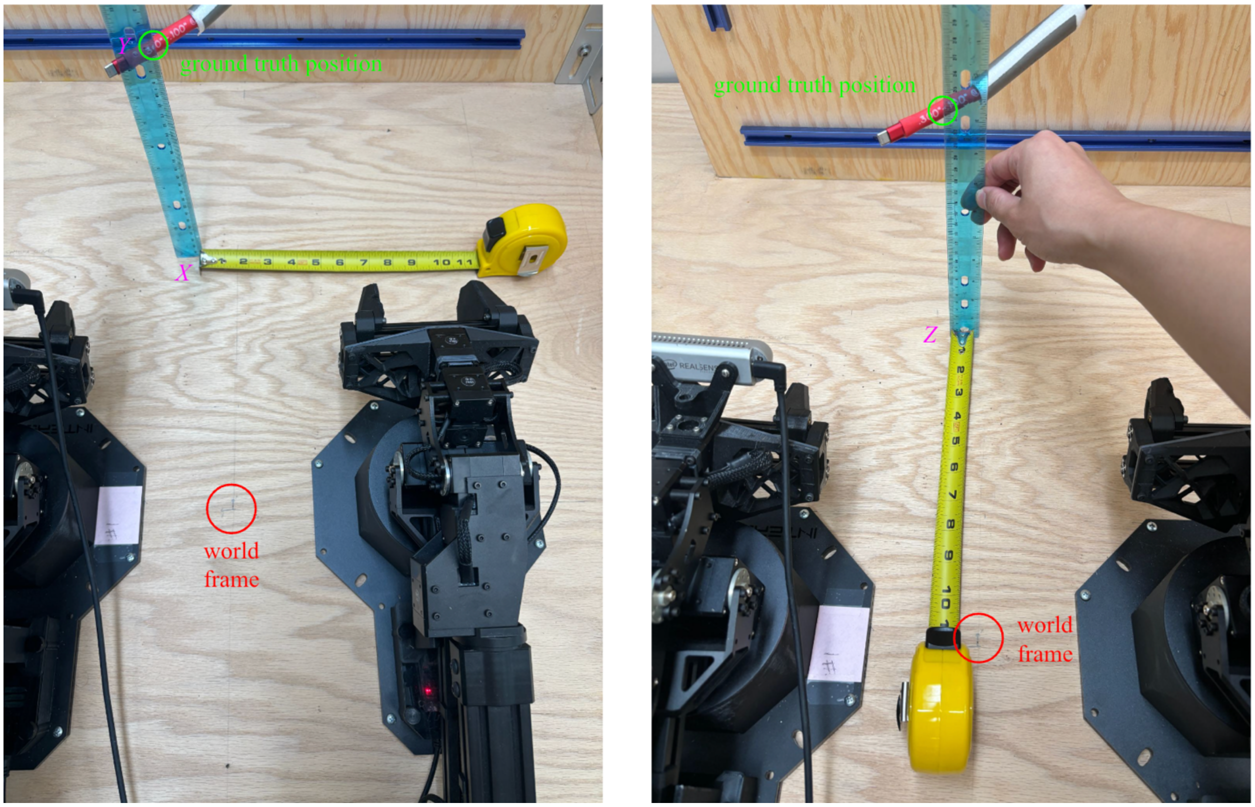


Figure 42: Measurement procedure for determining connector end ground truth *X*, *Y*, *Z* relative to system world frame

In the system's simulation view of the environment, a testing frame for the ground truth is created using these measurements. Figure 43 visualizes the ground truth frame for DLO configuration 3, generated from the measurement procedure depicted in Figure 42, in the RViz system simulation.

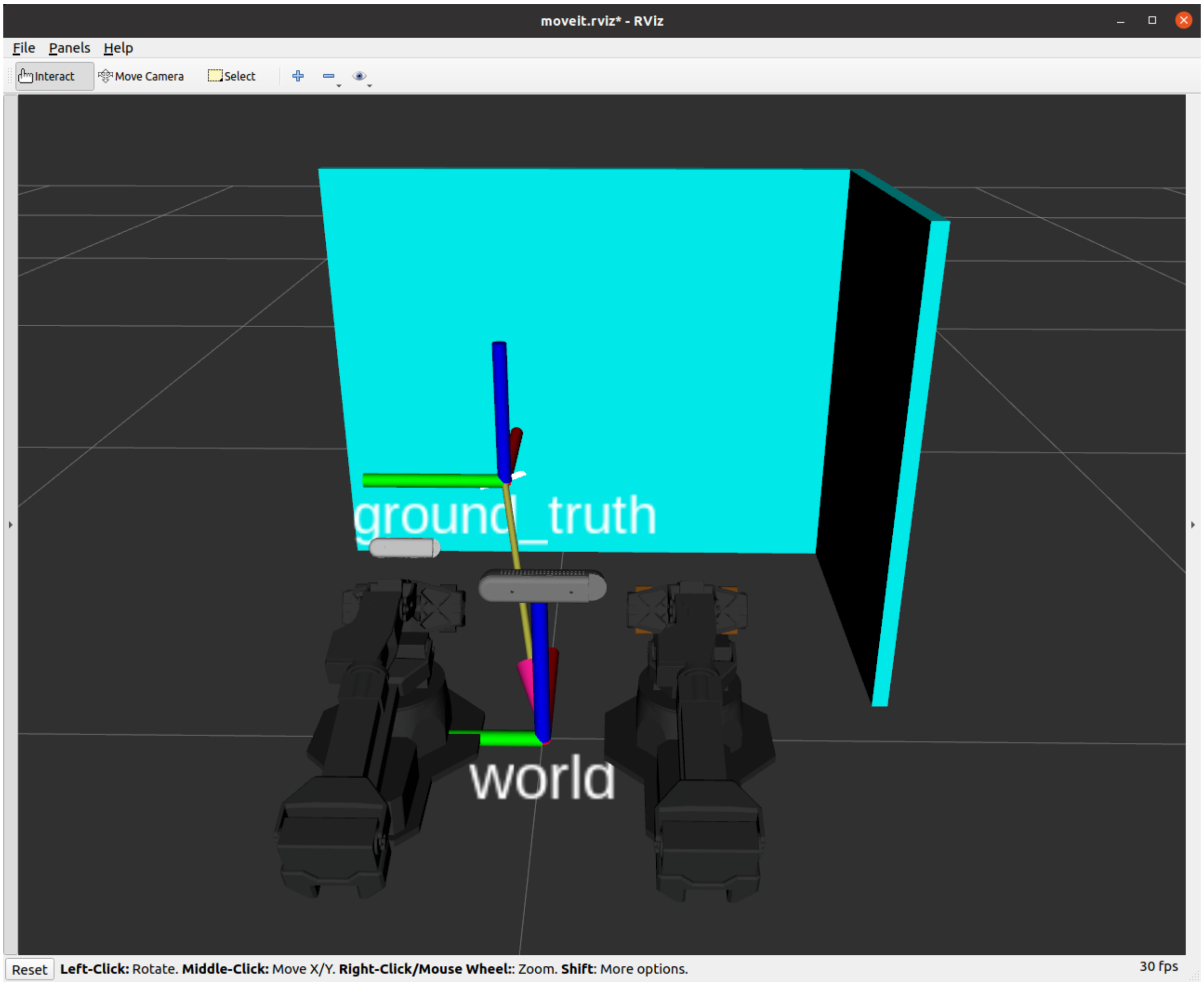

moveit.rviz* - RViz
File
Panels
Help
Interact
Move Camera
Select
ground_truth
world
Reset
Left-Click: Rotate. Middle-Click: Move X/Y. Right-Click/Mouse Wheel:: Zoom. Shift: More options.
30 fps

Figure 43: RViz simulation view of the generated ground truth for DLO Configuration 3

The distance between the connector end's ground truth testing frame and the pose estimation frame is then computed by calculating the 3D distance between them. Due to inherent noise from the sensors affecting the accuracy of pose estimation, the position used for comparison against the ground truth is the position averaged over 3 seconds. A lower distance equates to a more accurate estimation of position as the difference between the frames, or error in position estimation, is less significant. Each approach's ability to estimate orientation is lastly gauged by verifying whether the connector end is graspable by the robotic arm guided by the estimation frame generated. Because a theme explored in this thesis is maximizing robustness in task execution with the minimum level of accuracy required, success of the system is determined by a valid grasp of the end and not necessarily a perfect estimation of its orientation. Therefore, orientation estimation results are quantified again as a success or failure depending on whether the robotic arm is able to grasp the connector end with the provided calculation. The DLO is configured such that grasp is certainly achievable if the system performs pose estimation properly. The results of each technique's pose estimation along the metrics of identification, positional difference, and orientation graspability is described in the following sections.

4.1.1.2 Fiducial Markers Results

With 9 trials in this experiment, the use of fiducial markers for pose estimation resulted in an identification success rate of 48.1% (13/27 successful, 14/27 failure), grasp success rate of 48.1% (13/27 successful, 14/27 failure), and average ground truth distance error of 0.02881 meters. The data collected from each trial is listed in Table 3.

Trial	DLO Configuration	Identification Success	Ground Truth Error (m)	Grasp Success
1	1	success	0.02891	success
2	1	success	0.02888	success
3	1	success	0.02889	success
4	2	success	0.02881	success
5	2	success	0.02879	success
6	2	success	0.02881	success
7	3	success	0.02874	success
8	3	success	0.02873	success
9	3	success	0.02875	success
10	4	failure	-	-
11	4	failure	-	-
12	4	success	0.02879	success
13	5	failure	-	-
14	5	failure	-	-
15	5	failure	-	-
16	6 (10%)	success	0.02885	success
17	6 (10%)	failure	-	-
18	6 (10%)	success	0.02887	success
19	7 (30%)	failure	-	-
20	7 (30%)	failure	-	-
21	7 (30%)	success	0.02877	success
22	8 (50%)	failure	-	-
23	8 (50%)	failure	-	-
24	8 (50%)	failure	-	-
25	9 (70%)	failure	-	-
26	9 (70%)	failure	-	-
27	9 (70%)	failure	-	-

Table 3: ArUco fiducial marker pose estimation identification/grasp success and ground truth error per trial, by DLO configuration

From the RViz simulation, Figure 44 demonstrates the utilization of fiducial markers to estimate the connector end pose in configuration 3, depicting the ground truth frame (left), estimated pose frame (center), and the two frames together (right).

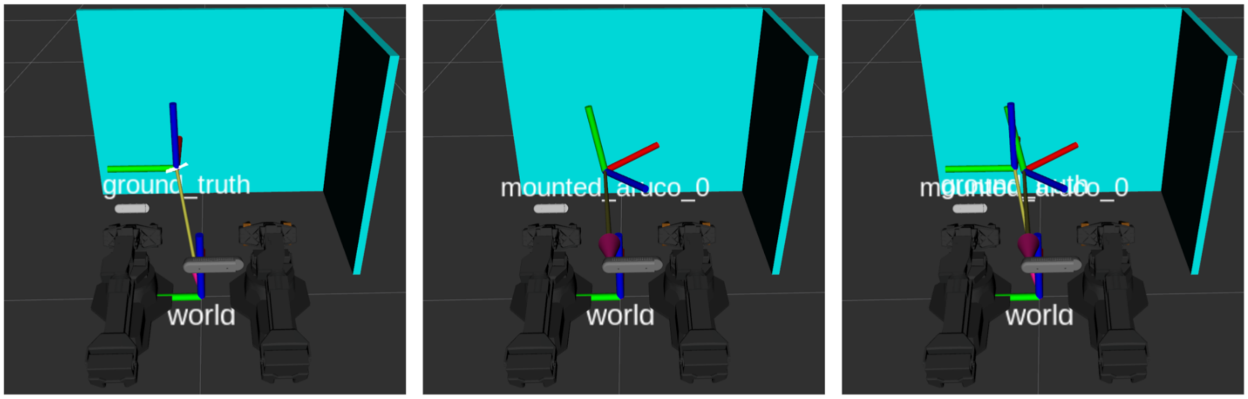


Figure 44: RViz simulation view of the configuration 3 ground truth (left), fiducial marker pose estimation frame (center), and both frames together (right)

4.1.1.3 RGBD Image Processing Results

The utilization of RGB and depth image processing for pose estimation yielded an identification success rate of 100.0% (27/27 successful, 0/27 failure), grasp success rate of 74.1% (20/27 successful, 7/27 failure), and average ground truth distance error of 0.01081 meters. The data obtained per trial is presented in Table 4.

Trial	DLO Configuration	Identification Success	Ground Truth Error (m)	Grasp Success
1	1	success	0.00443	success
2	1	success	0.00448	success
3	1	success	0.00446	success
4	2	success	0.00432	failure
5	2	success	0.00437	success
6	2	success	0.00436	failure
7	3	success	0.00447	success
8	3	success	0.00443	success
9	3	success	0.00439	success
10	4	success	0.00446	success
11	4	success	0.00451	success
12	4	success	0.00448	failure
13	5	success	0.00439	success
14	5	success	0.00442	success
15	5	success	0.0044	success
16	6 (10%)	success	0.00881	success
17	6 (10%)	success	0.00889	success
18	6 (10%)	success	0.00883	success
19	7 (30%)	success	0.01359	success
20	7 (30%)	success	0.01365	failure
21	7 (30%)	success	0.01358	success
22	8 (50%)	success	0.01699	success
23	8 (50%)	success	0.01695	success
24	8 (50%)	success	0.01694	failure
25	9 (70%)	success	0.0357	failure
26	9 (70%)	success	0.03575	success
27	9 (70%)	success	0.03572	failure

Table 4: RGBD image processing pose estimation identification/grasp success and ground truth error per trial, by DLO configuration

The configuration 3 ground truth frame (left), RGBD image pose estimation frame (center), and the two frames together (right) is visualized in Figure 45’s simulation view.

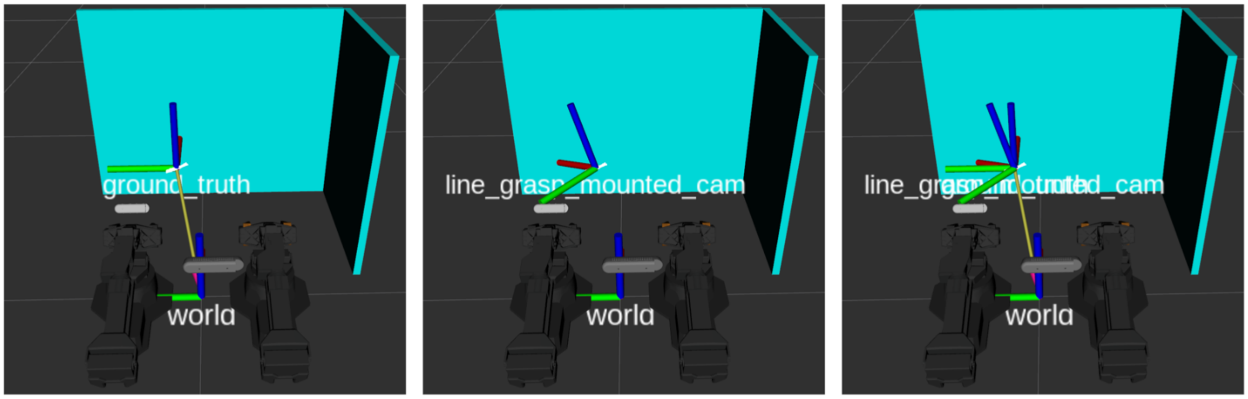


Figure 45: RViz simulation view of the configuration 3 ground truth (left), RGBD image pose estimation frame (center), and both frames together (right)

4.1.1.4 Machine Learning Results

Pose estimation conducted by machine learning object detection for position and depth image processing for orientation produced an identification success rate of 77.8% (21/27 successful, 6/27 failure), grasp success rate of 63.0% (17/27 successful, 10/27 failure), and average ground truth distance error of 0.02913 meters. Table 5 details the results per trial.

Trial	DLO Configuration	Identification Success	Ground Truth Error (m)	Grasp Success
1	1	success	0.03091	success
2	1	success	0.03099	success
3	1	success	0.03097	success
4	2	success	0.01877	failure
5	2	success	0.01879	failure
6	2	success	0.01876	failure
7	3	success	0.03065	success
8	3	success	0.03068	success
9	3	success	0.03059	success
10	4	success	0.01471	failure
11	4	success	0.01472	success
12	4	success	0.01469	success
13	5	success	0.03129	success
14	5	success	0.03136	success
15	5	success	0.03134	success
16	6 (10%)	success	0.03521	success
17	6 (10%)	success	0.03522	success
18	6 (10%)	success	0.03517	success
19	7 (30%)	success	0.04002	success
20	7 (30%)	failure	-	-
21	7 (30%)	success	0.04006	success
22	8 (50%)	failure	-	-
23	8 (50%)	failure	-	-
24	8 (50%)	success	0.04677	success
25	9 (70%)	failure	-	-
26	9 (70%)	failure	-	-
27	9 (70%)	failure	-	-

Table 5: Machine learning pose estimation identification/grasp success and ground truth error per trial, by DLO configuration

Figure 46 depicts the RViz simulation views of the configuration 3 connector end pose ground truth frame, the estimated pose generated via machine learning (center), and the two frames together (right).

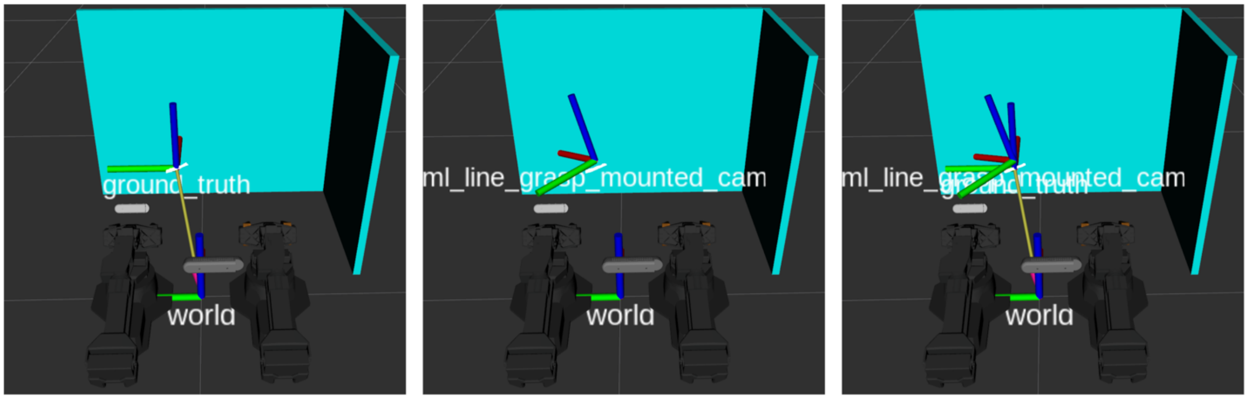


Figure 46: RViz simulation view of the configuration 3 ground truth (left), machine learning pose estimation frame (center), and both frames together (right)

4.1.1.5 Identification and State Estimation Results Summary

Figure 47 and Figure 48 charts the number of successes and failures for connector end identification and grasp respectively, per pose estimation technique. Between the three pose estimation techniques evaluated in this experiment, RGBD image processing had the highest identification success rate at 100.0% as well as the highest grasp success rates at 74.1%.

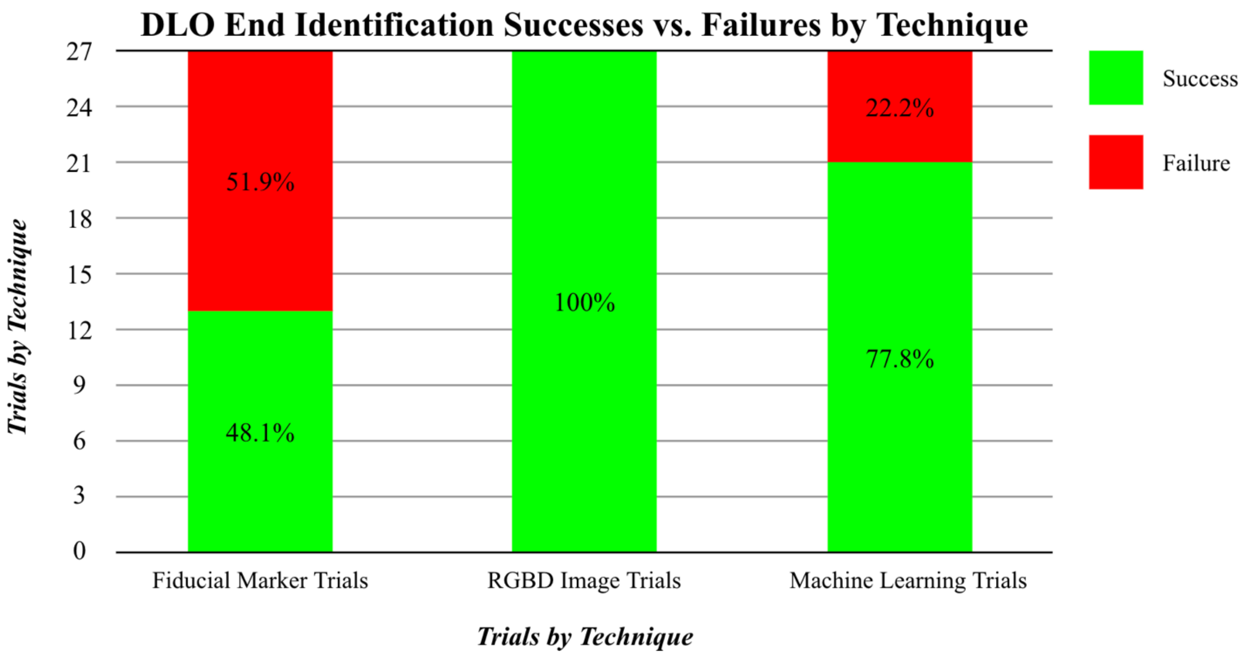


Figure 47: Identification success and failure count per trial by pose estimation technique

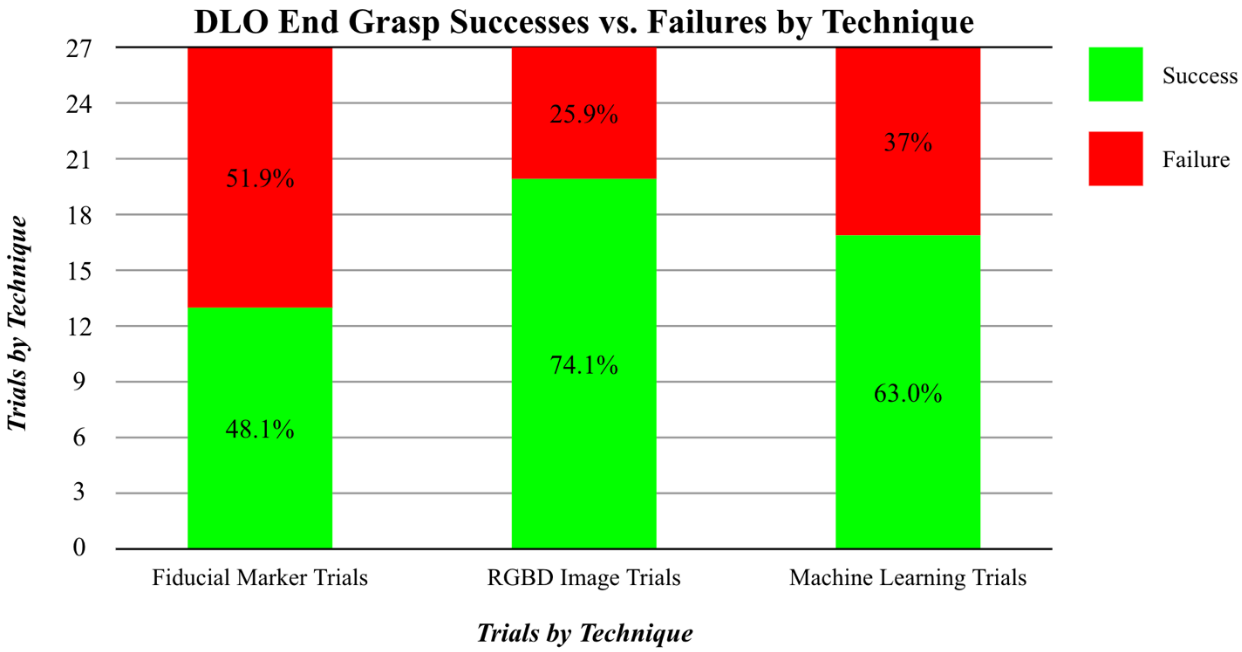


Figure 48: Grasp success and failure count per trial by pose estimation technique

Because the RGBD Image and machine learning approaches share the same orientation estimation method, their total number of grasping results can be compiled and compared to the grasp success rate of the fiducial markers, as illustrated in Figure 49.

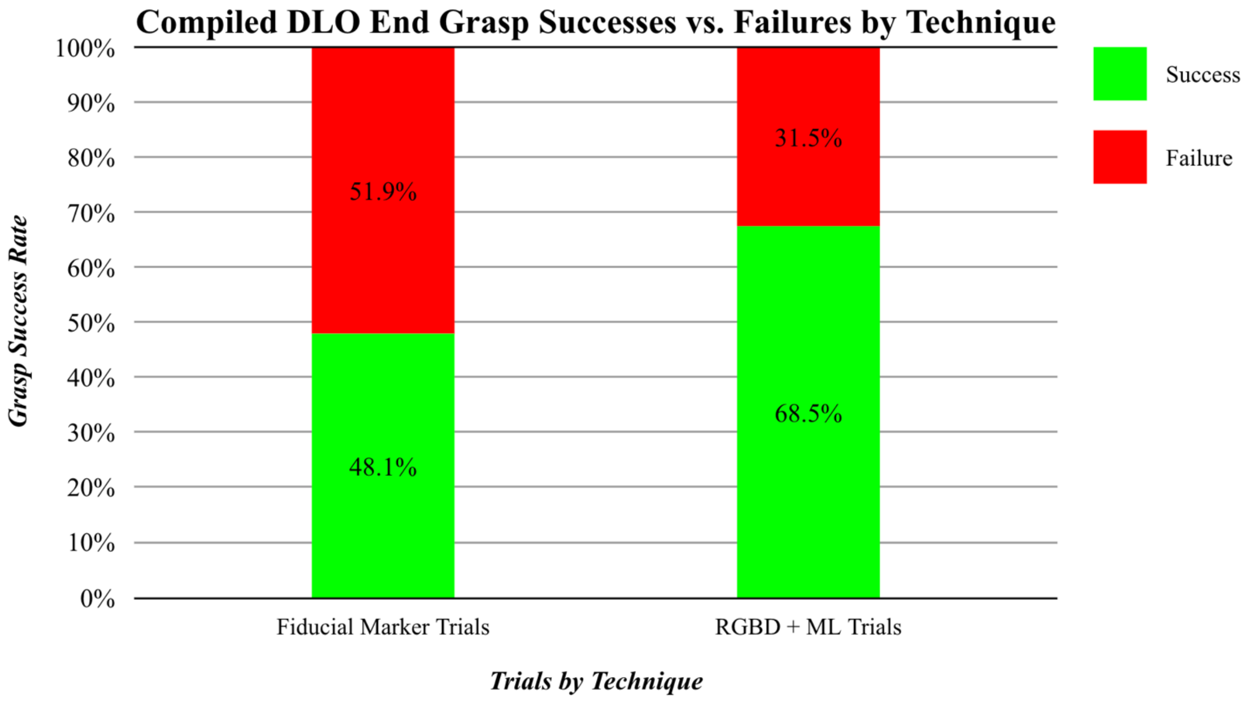


Figure 49: Compiled grasp success rates between fiducial markers and the shared RGBD and ML orientation estimation approach

The ground truth distance error per trial for each technique is plotted in Figure 50. RGBD image processing had the lowest average ground truth error at an average distance of 0.01081 meters, followed by machine learning at 0.02881 meters, and lastly fiducial markers at 0.02913 meters.

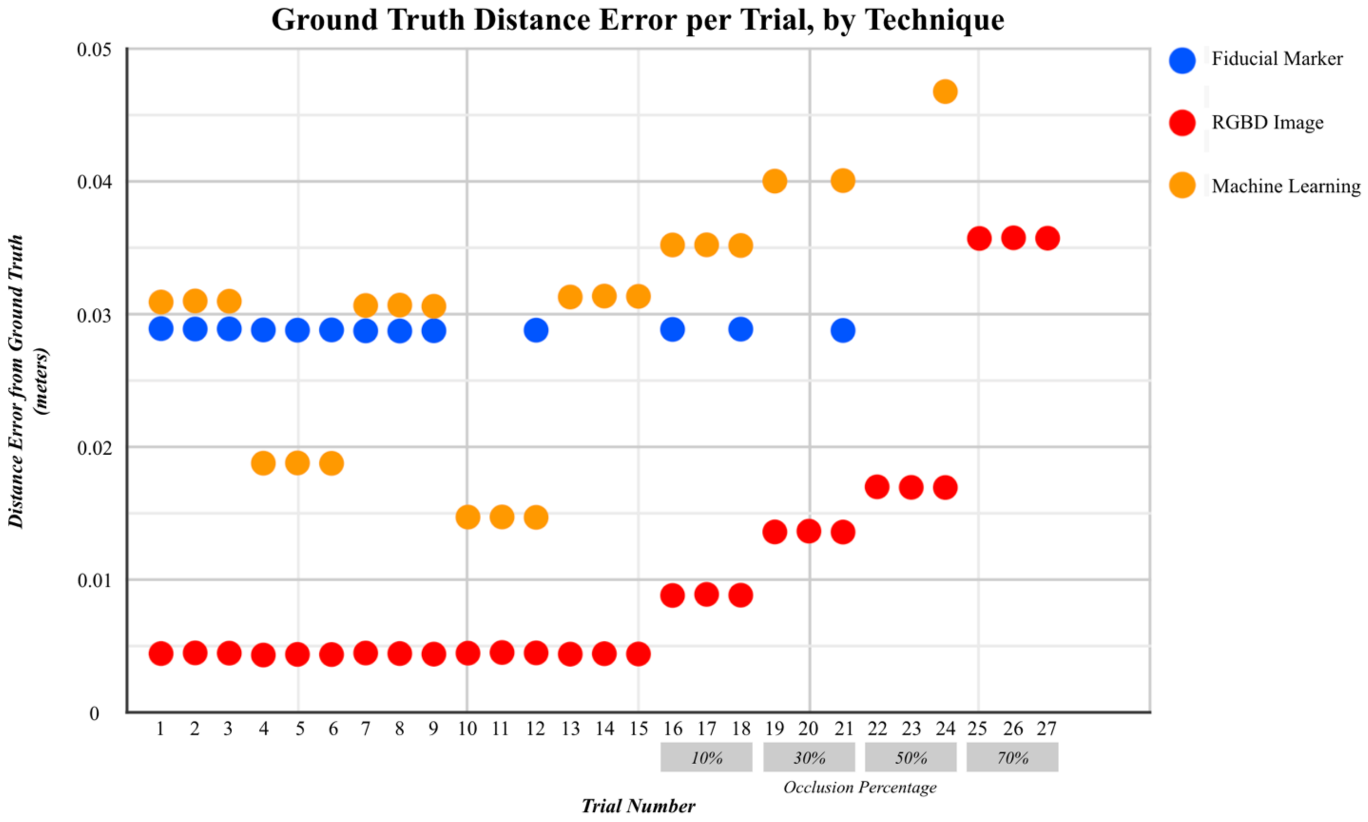


Figure 50: Ground truth distance error per trial by pose estimation technique

For each pose estimation technique, the average ground truth distance error, variance, and standard deviation are presented in Table 6.

Pose Estimation Technique	Ground Truth Mean Distance Error (meters)	Variance	Standard Deviation
Fiducial Marker	0.02881	3.627e-9	6.022e-5
RGBD Image	0.01081	1.009e-4	1.004e-2
Machine Learning	0.02913	8.185e-5	9.047e-3

Table 6: Ground truth mean distance error, variance, and standard deviation per pose estimation technique

4.1.2 Slip Detection

4.1.2.1 Experiment Overview

Presented in this thesis is an approach to recognizing the slip of the DLO connector end from the manipulating robotic arm's end effector utilizing only visual data from camera sensors. An identified connector end is constantly monitored throughout the execution of a task though pose estimation. The positional aspect of the estimated state is compared against the current position of the end effector and is flagged a slip anomaly in the case that the distance between the two surpasses a preset threshold. To assess this technique's robustness in detecting a slip, and assess where shortcomings and failures may present themselves, the following experiment was devised.

Each trial of this experiment begins with both ends of the DLO described in **Section 3.1.1** connected to the right and back panel mock battery ORU blocks. The manipulating robotic arm proceeds to unplug the left end of the DLO and carries it in its gripper to one of three positions across the workspace's horizontal plane. Once the arm maneuvers the connector end to that point, the arm purposefully releases the end connector from its grip in a controlled, engineered slip from two different heights. Due to spatial constraints of the environment, location of the connector, and available reach of the manipulating arm, these locations were selected as they evenly space slipping points laterally at heights suitable for producing low- and high-distance slips. Figure 51 depicts the manipulating robotic arm with DLO connector in hand at each of the six slipping points.

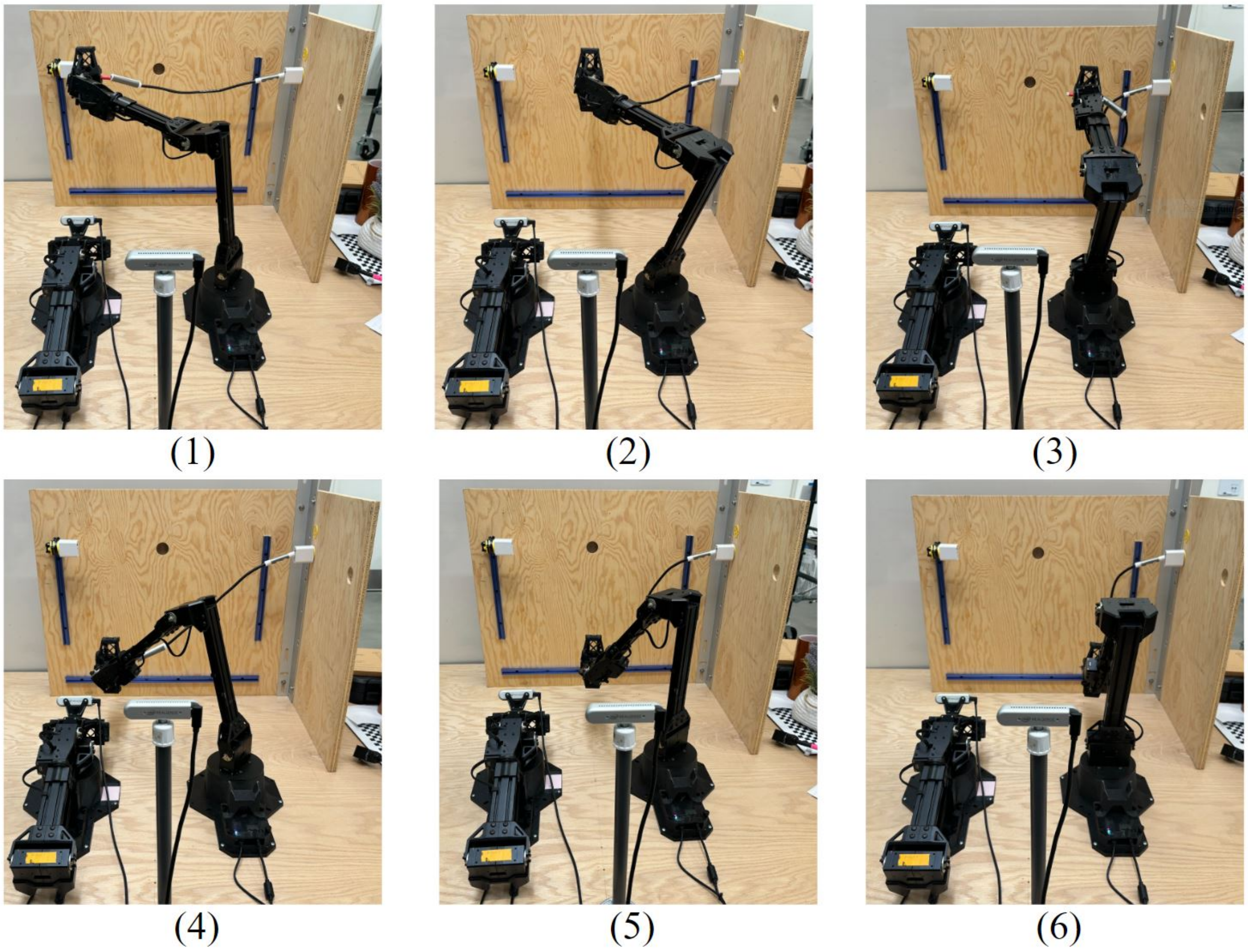


Figure 51: Right manipulating robotic arm grasping DLO end at each of the six slipping points

Across all 6 trials, a slip threshold of 0.2 meters is set, meaning a slip is flagged by the system if the distance between the estimated position of the DLO connector end and center of the manipulating arm's end effector surpasses this value. Following a slip, the system is given 10 seconds to identify the slip and estimate the loose DLO end's new pose. For pose estimation, the RGB and depth image processing method was selected due to its robust performance across metrics presented in **Section 4.1.1**. As slips are difficult to reproduce exactly for a controlled experiment, three slip trials are conducted at each of the six slipping points, resulting in a total of 18 trials for this experiment.

4.1.2.2 Slip Detection Results

Over the 18 trials performed across the six slipping point DLO configurations, the system had a success rate of 72.2%, composed of 13 successful and 5 failed detections within the allotted ten seconds. This data is presented in Table 7, detailing the trial number, position configuration of the DLO, height describing whether the slip point was classified as "high" or "low", identification success, and reason for failure if the slip was not detected.

Trial	DLO Configuration	Height	Identification Success	Cause of Failure
1	1	high	success	-
2	1	high	success	-
3	1	high	failure	loss of vision
4	4	low	failure	threshold not met
5	4	low	success	-
6	4	low	failure	loss of vision
7	2	high	success	-
8	2	high	success	-
9	2	high	success	-
10	5	low	success	-
11	5	low	failure	threshold not met
12	5	low	failure	threshold not met
13	3	high	success	-
14	3	high	success	-
15	3	high	success	-
16	6	low	success	-
17	6	low	success	-
18	6	low	success	-

Table 7: Slip detection success per trial, by DLO configuration and slip position

The failures occurred under configurations 1, 4, and 5, having 1, 2, and 2 unsuccessful detections respectively. To offer insight into the trials with failed detections, these can be viewed with respect to the height of each trial's slip position. Figure 52 splits the total number of trials by whether each experienced a high or low height slip and shows the breakdown of successes according to that slip height. Out of 9 trials per set of high and low drop trials, success rates of 88.9% and 55.6% were achieved by the system.

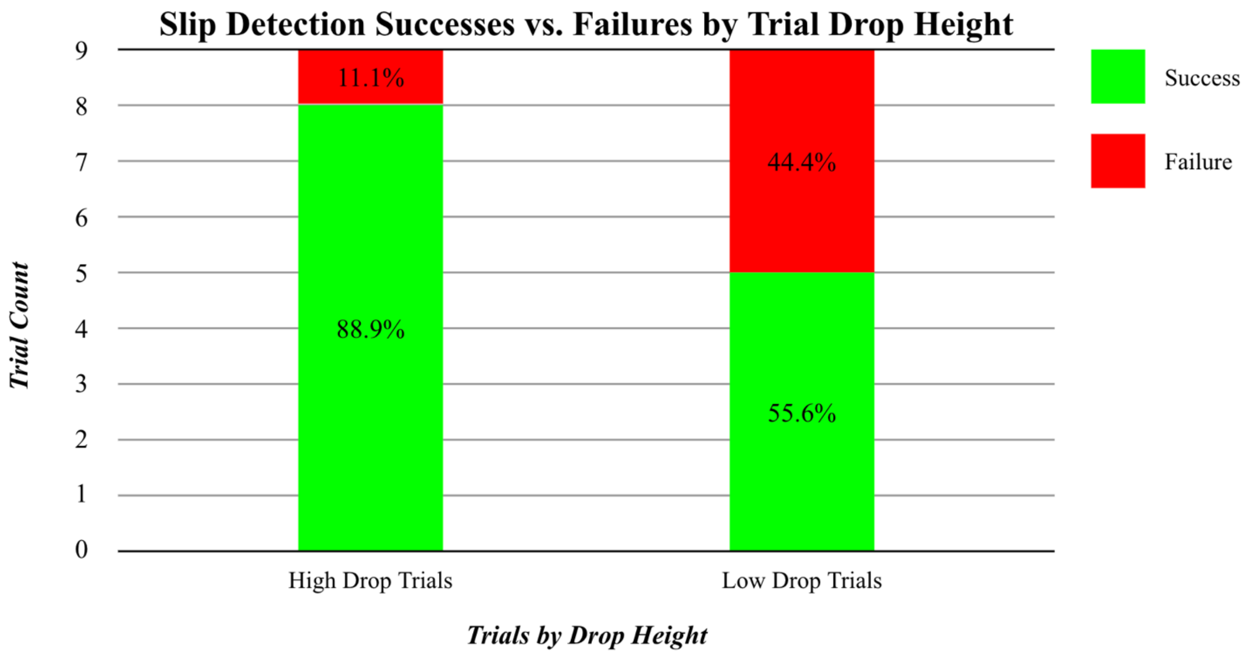


Figure 52: Slip detection success and failure count by trial drop height

Furthermore, Figure 53 charts only the failed trials by drop height, delineated by the cause of that trial's respective failure. Among high drop trials, 100.0% of failures were the result of vision failure. For low drop trials, 25.0% of failures were due to vision failure while 75.0% were the result of the slip threshold not being met.

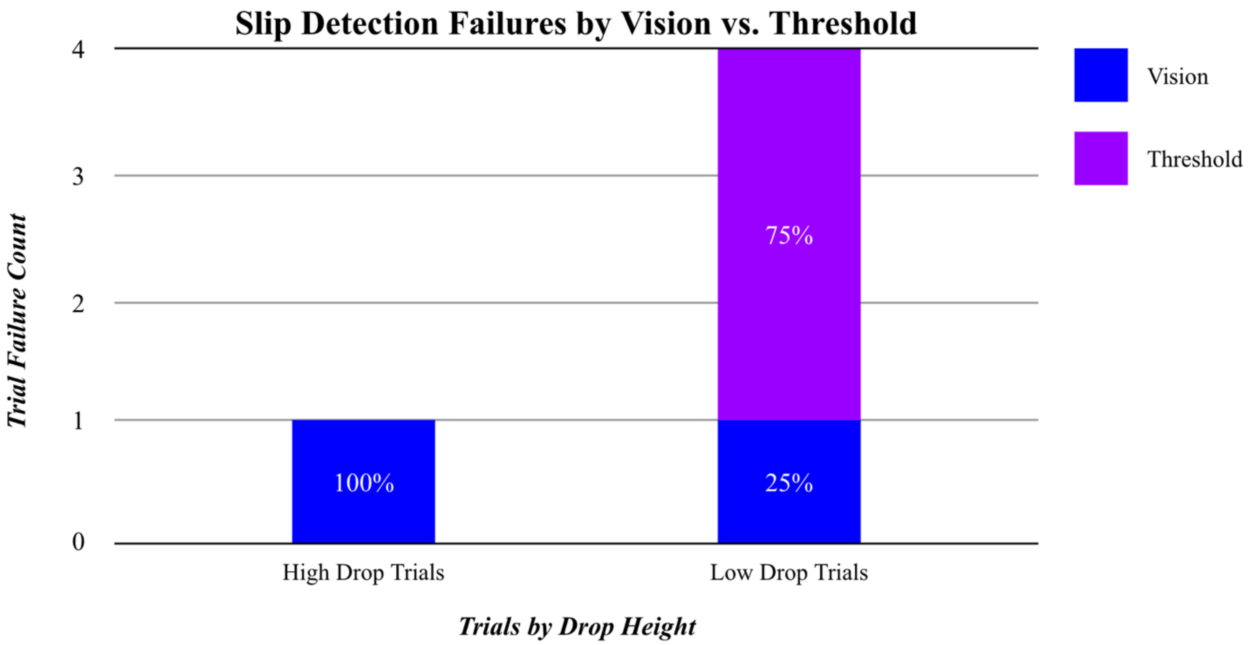


Figure 53: Cause of failure breakdown for slip detection failed trials

4.1.3 Environment Spiral Search Pattern

4.1.3.1 Experiment Overview

Once the system determines the loss of the DLO connector end behind an obstructing environmental obstacle, this thesis proposes sending the second arm which is not utilized for DLO manipulation in a spiral pattern search of the environment as described in **Section 3.4.3**. The following experiment was designed to evaluate this technique's robustness in relocating the DLO connector end and locating the blind spots within the pattern.

The spiral search pattern experiment begins with one end of the DLO connected to the right panel block, while the left is disconnected and loose in the environment. The DLO is configured into one of three configurations, again chosen for emulating realistic settled DLO poses after a slip in zero gravity with natural droop due to testing the weighted end under Earth gravity. Each configuration is then covered by either the smaller or larger obstacle described in

Section 3.1.1. Under this setup, spiral search is then initiated twice: once in which the system is informed of the DLO end's last known location, and another where that location is unknown, requiring the system begin searching from the center of the back panel, the plane parallel to the camera view. This is defined as the exact center along both the *X* and *Y* axis of the rear-mounted camera's view. Figure 54 depicts the left robotic arm initiating a search of the workspace from both the connector's last known location, as well as the camera's view of the workspace origin, with the smaller obstacle obscuring the connector end.

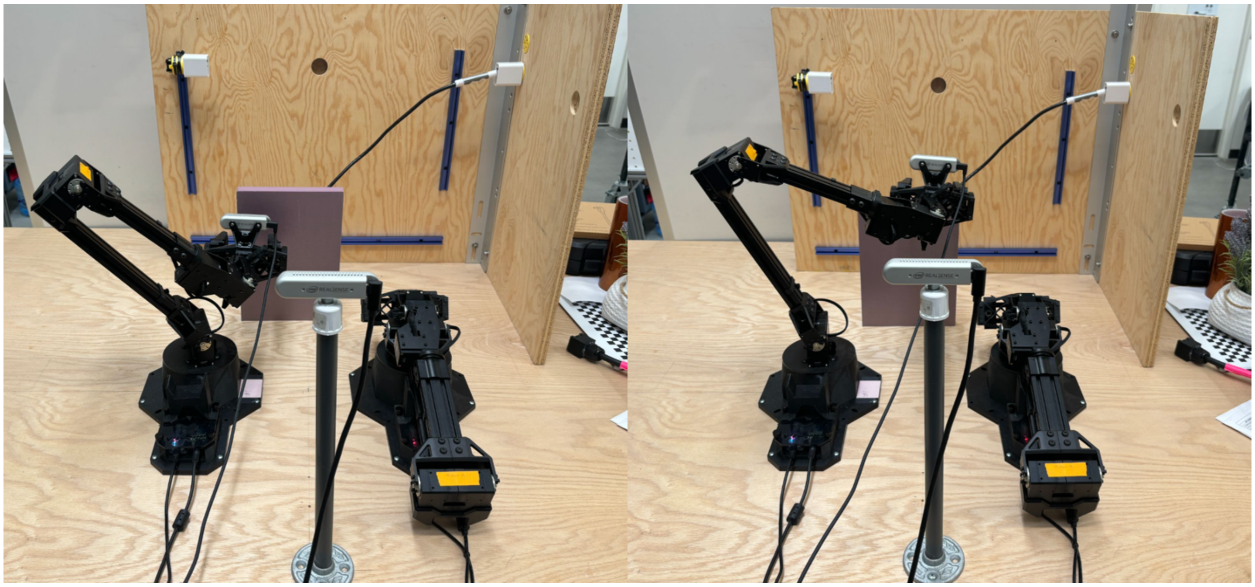


Figure 54: Spiral search initiated from the DLO connector's last known position (left) and workspace origin (right)

Each trial is performed three times per configuration of DLO, selected obstacle, and starting search position. The experiment is therefore composed of 36 trials across these three variables, wherein three metrics are recorded per attempt. Whether the system is able to successfully locate the DLO connector end with the search pattern within 5 minutes is first recorded, denoted as either a success or failure. If the search was successful, the time to find the connector end is also recorded. This is defined as the total time, in seconds, the search pattern required between initiation of the search procedure to the pose estimation of the DLO end. Lastly, the main point or sub point along the spiral search pattern in which the connector was identified is also recorded.

4.1.3.2 Search Pattern Results

In the 36 trials testing the developed spiral search pattern, the system achieved a success rate of 91.7%, composed of 33 successful and 3 failed identifications within the 5-minute search period. Table 8 outlines these results, listing the trial number, DLO configuration, occluding obstacle, search pattern's starting position, success of the search. If the identification was successful, the time (in seconds) to find the DLO and the point of identification are also presented.

Trial	DLO Configuration	Obstacle	Start	Identification Success	Time to Find (sec)	Point of Identification
1	1	short	last known	success	29.64	main point 1, subpoint 1
2	1	short	last known	success	30.55	main point 1, subpoint 1
3	1	short	last known	success	29.91	main point 1, subpoint 1
4	1	short	origin	success	141.09	main point 1, subpoint 8
5	1	short	origin	success	144.23	main point 1, subpoint 8
6	1	short	origin	success	142.7	main point 1, subpoint 8
7	1	tall	last known	success	46.11	main point 1, subpoint 2
8	1	tall	last known	success	43.39	main point 1, subpoint 2
9	1	tall	last known	success	44.47	main point 1, subpoint 2
10	1	tall	origin	success	267.16	main point 2, subpoint 7
11	1	tall	origin	success	271.25	main point 2, subpoint 7
12	1	tall	origin	success	252.68	main point 2, subpoint 6
13	2	short	last known	success	25.1	main point 1, subpoint 1

14	2	short	last known	success	26.71	main point 1, subpoint 1
15	2	short	last known	success	25.49	main point 1, subpoint 1
16	2	short	origin	success	139.26	main point 1, subpoint 8
17	2	short	origin	success	143.07	main point 1, subpoint 8
18	2	short	origin	success	142.38	main point 1, subpoint 8
19	2	tall	last known	success	43.39	main point 1, subpoint 2
20	2	tall	last known	success	44.92	main point 1, subpoint 2
21	2	tall	last known	success	44.51	main point 1, subpoint 2
22	2	tall	origin	failure	timeout	-
23	2	tall	origin	failure	timeout	-
24	2	tall	origin	success	141.66	main point 1, subpoint 8
25	3	short	last known	success	29.02	main point 1, subpoint 1
26	3	short	last known	success	26.8	main point 1, subpoint 1
27	3	short	last known	success	27.95	main point 1, subpoint 1
28	3	short	origin	success	144.58	main point 1, subpoint 8
29	3	short	origin	success	140.6	main point 1, subpoint 8
30	3	short	origin	success	144.14	main point 1, subpoint 8
31	3	tall	last known	success	45.9	main point 1, subpoint 2
32	3	tall	last known	success	43.69	main point 1, subpoint 2
33	3	tall	last known	success	47.11	main point 1, subpoint 2
34	3	tall	origin	success	141.78	main point 1, subpoint 8
35	3	tall	origin	failure	timeout	-
36	3	tall	origin	success	142.21	main point 1, subpoint 8

Table 8: Spiral search success, time, and point per trial, by occluding obstacle and start position

Of the 36 total trials, half of them began their search at either the DLO connector's last known location or at the origin of the workspace. Two subsets of trials can thus be defined depending on the trial's initial search position, with their success and failure counts visualized in Figure 55.

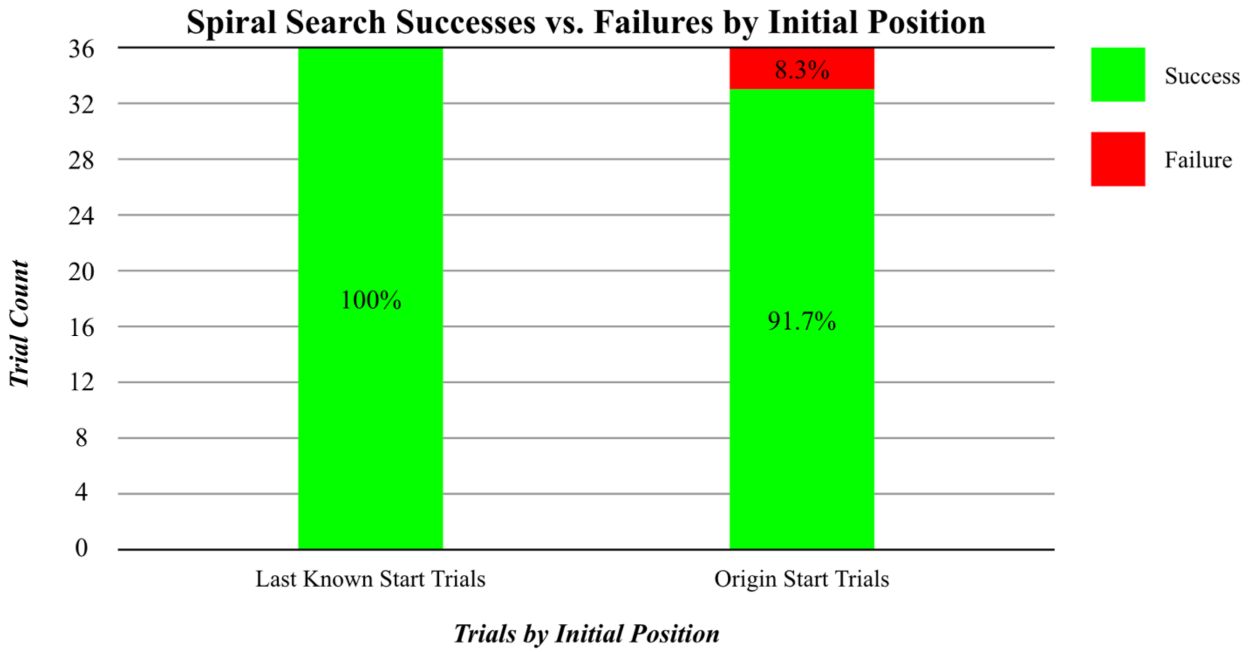


Figure 55: Spiral search trial success and failure count, by initial position

All successful trials, separated into these two start position subsets, are plotted in Figure 56 according to the total elapsed time to locate the DLO end connector.

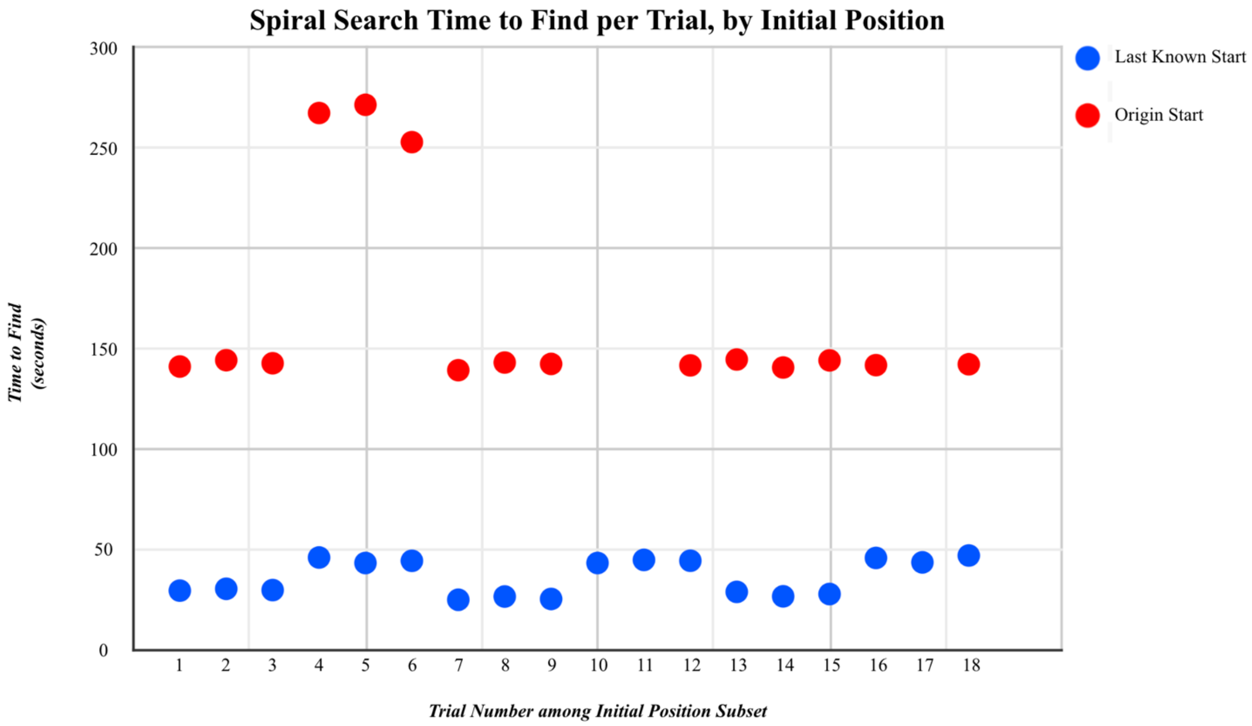


Figure 56: Elapsed time to find DLO connector end per subset trial, by initial position

Among the 33 successful trials, the connector end was most often found within the subpoints of the first main point, which accounted for 30/33 of the identifications. The remaining 3 successful trials were found within the main point's second subpoints. Among subpoints, identification was most common in subpoint 8, totaling 12 successful identifications. Subpoints 1 and 2 both had 9 successful trials, followed by subpoints 7 and 6 with 2 and 1 identifications, respectively. This data is illustrated in Figure 57's pie chart, visualizing the distribution of identifications between subpoint poses.

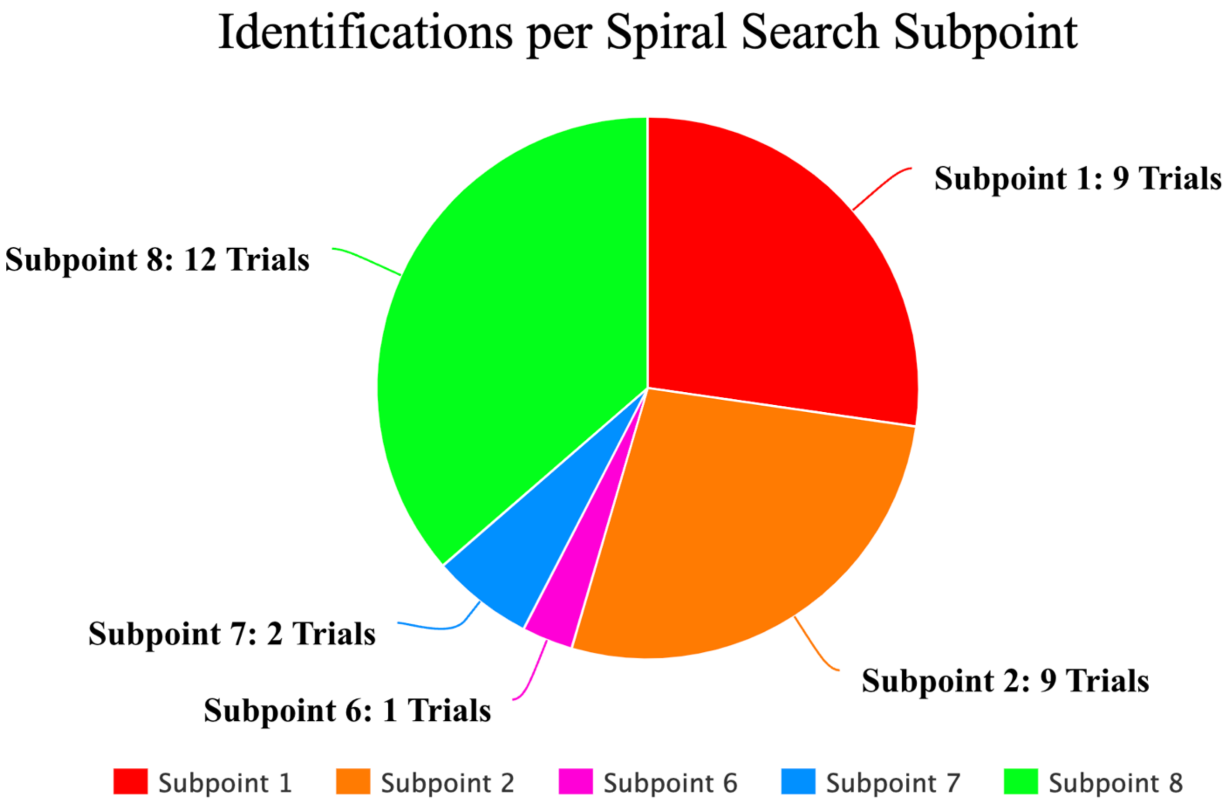


Figure 57: Distribution of successful identifications by spiral search pattern subpoint

4.2 Full Scenario Demonstration

4.2.1 Demonstration Overview

Having individually evaluated and proven the robustness and efficacy of each capability presented in the Methods section of this thesis, these approaches can be organized into a complete solution demonstration to prove the validity of these techniques in resolving slip detection and response. The demonstration designed to assess the system directly mirrors the anomaly scenario described in **Section 1.2**, and is conducted as follows.

The demonstration begins under the condition that the battery ORU is degraded and requires replacement but no action has been taken by the system. The environment is arranged to reflect this, with the right end of the DLO plugged into the right panel, and the left end connected to the mock degraded battery ORU unit. As no actions have been executed, the two robotic arms are in their sleep state. The system then sends the right manipulating arm to unplug the left connector end of the DLO. After disconnecting the DLO from the degraded battery ORU, the arm, with unplugged connector end in its end effector, maneuvers towards an imaginary stowing point. While en route, the arm purposefully drops the connector, emulating a slip of the DLO into the workspace. The motion planning of the robotic agent for unplugging and slipping is preset to maintain consistency and repeatability among the set of demonstrations. Additionally, the objective of this demonstration, and thesis as a whole, aims to address the slip detection and resolution aspect of the scenario, rather than fine tuning DLO manipulation. The DLO connector end has thus entered an unknown state and must be relocated by the system to return to nominal functionality. For this demonstration, identification and pose estimation are done via the RGB and depth image processing approach as it was proven to be the most robust across its technique validation metrics (*Section 4.1.1.3*). This demonstration seeks only to offer the entire system as a solution for the slip anomaly problem, therefore the other pose estimation methods are not explored. Employing this technique, the system must first recognize the slip of the DLO connector end from the manipulating arm's end effector, recorded as either a success or failure. Next, the system must locate and estimate pose for the loose connector end. The procedure to accomplish this, consisting of selecting which camera to use and whether to initiate a spiral search of the environment, is left to the discretion of the system operating with the methods described in **Section 3.4**. Three trials of the proposed full system solution are conducted, varying in workspace configuration. The first contains no obstacles in the environment, while the second and third contain the shorter and taller foam obstacles, respectively. These objects are placed such that they completely obscure the line of sight that the rear mounted camera has on the anticipated settled position of the loose connector end (post slip). The system's ability to locate the connector end and perform pose estimation, again, is noted as either a success or failure. The robotic arm must actually physically retrieve the relocated connector end to restore it to a known state, and the pose calculated by the system should offer a valid position and orientation for grasping. Whether the system's pose estimation results in a solid grasp of the connector end is also recorded as a success or failure. Assuming a successful retrieval is made, the connector end is finally moved to a mock stowing point for demonstration purposes, concluding the demonstration. Figure 58 details each step of the full demonstration, with accompanying labels and visuals of the respective stage.

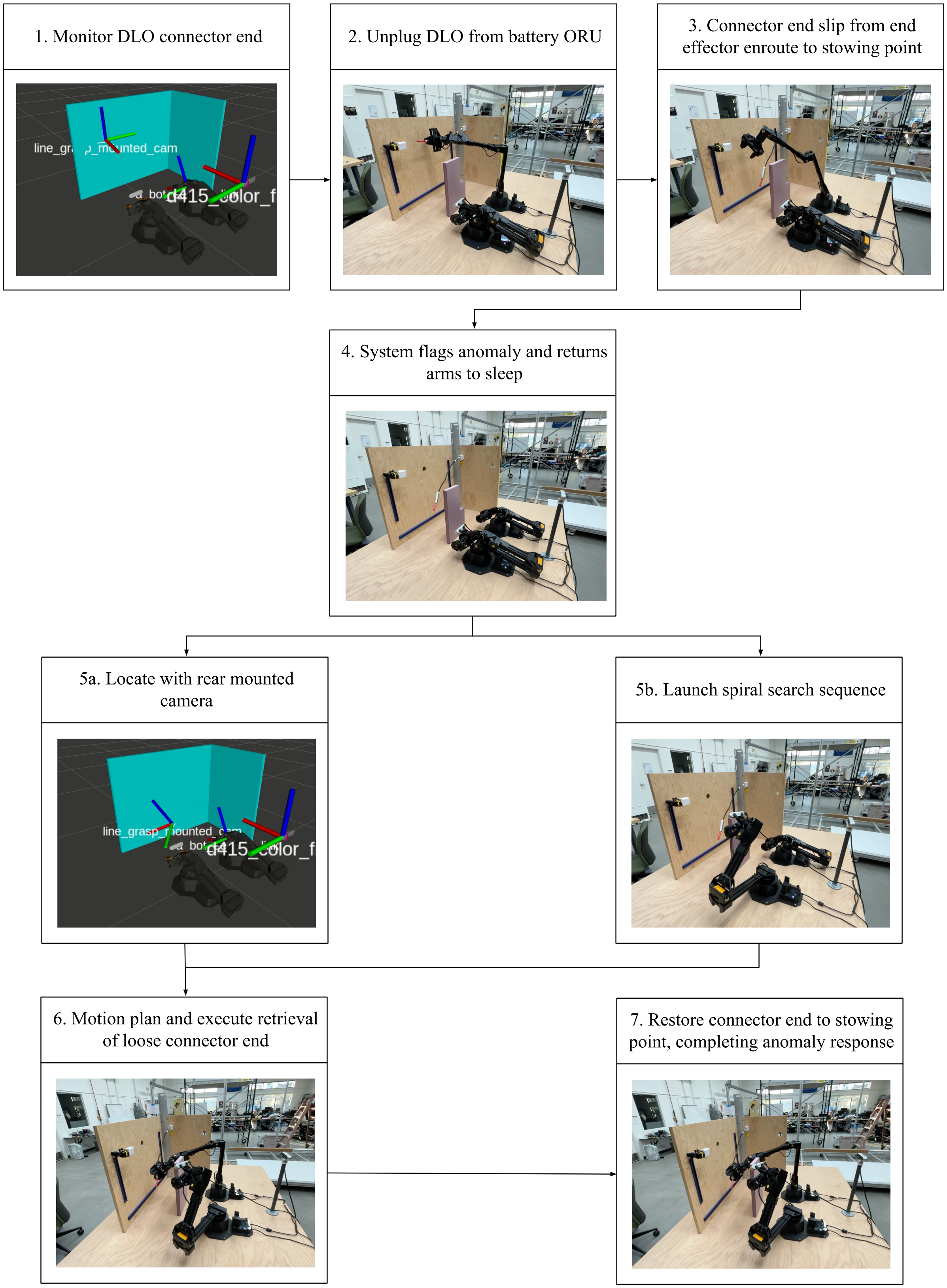

1. Monitor DLO connector end
2. Unplug DLO from battery ORU
3. Connector end slip from end effector enroute to stowing point
4. System flags anomaly and returns arms to sleep
5a. Locate with rear mounted camera
5b. Launch spiral search sequence
6. Motion plan and execute retrieval of loose connector end
7. Restore connector end to stowing point, completing anomaly response

Figure 58: Outline of all stages composing the full system demonstration

4.2.2 Results and Summary

The complete system demonstration, evaluating the end-to-end solution proposed by this work, was successfully conducted over three iterations. The three versions consisting of featuring no obstacles, using the short foam obstacle, and using the long foam obstacle respectively, all detected the slip of the DLO from the manipulating arm's end effector. They all also relocated the loose connector end via either the rear mounted camera or a deployment of the arm camera with spiral search. They all performed pose estimation of the DLO end, performed motion planning and executed a trajectory to retrieve the DLO end, and restored the DLO end to its originally intended position. Full videos of each of the three demonstrations is provided in the following links:

1. Demonstration 1 (no obstacles): https://youtu.be/C0trQMqRs6c
2. Demonstration 2 (short obstacle): https://youtu.be/aMunUT4PcUU
3. Demonstration 3 (tall obstacle): https://youtu.be/16bP1cBrb_E

4.3 Discussion

4.3.1 Identification and State Estimation

Each pose estimation technique, implemented in this robotic system for the purposes of identification and state estimation of a DLO connector end, presents unique advantages and disadvantages in its use. The data collected verify this notion, as trends observed from the technique validation experiment provide insight into the situations in which a particular approach is most appropriate according to measurements of accuracy or robustness. The three pose estimation techniques are further analyzed as follows.

4.3.1.1 Fiducial Markers

A rudimentary analysis of the fiducial marker's results in the validation experiments suggest that the method of using fiducial markers is an inferior pose estimation technique, netting the lowest identification rate, grasp rate, and largest average ground-truth distance error. The cause of these failures, however, lies in how fiducial markers are to be used and the environmental context in which this experiment placed them in. In order to function at maximum capacity, fiducial markers should fully be visible to a calibrated camera, entailing adequate lighting and an unobstructed view of the entire marker pattern. Several of the testing configurations, though realistic to the slip anomaly problem, violated the conditions that these tags depend on. Under DLO configuration 4, the connector end is near perpendicular to the camera, while configuration 5 leaves the fiducial marker pattern completely out of sight as the end is twisted towards the right of the workspace. Similarly in configurations 7, 8, and 9 (30%, 50%, and 70% occlusion respectively), high failure rates were the product of increasing occlusion of the connector end, blocking the crucial marker pattern from the camera's perspective.

When circumstances allow for an unobstructed view of the fiducial marker's entirety, the ArUco tags proved to be potent tools for pose estimation. Where robustness was lacking, fiducial markers compensated with highly precise positional data. Among the three pose estimation techniques explored, fiducial markers yielded the lowest variance and standard deviation values in its set of average ground truth error distances, both by a considerable margin. Despite its reported mean distance error, accuracy of the fiducial markers is also deceivingly exact. Even when askew, as long as the full pattern was visible, successful identification always led to a valid grasp of the connector. Furthermore, unlike the other two methods that generated a pose estimation from the connector directly, this technique required physically affixing an ArUco tag to the connector end. The tag itself has a thickness which is not accounted for in the ground truth measurement. Although the distance error indicates inaccuracy, this is only relative to the ground truth frame. Fiducial markers maintain high levels of precision, as evidenced by the deltas between their measurements. It is for this fact that the related visual servoing works seen in of [39], [40] chose to employ fiducial markers for boosting accuracy.

In this work, fiducial markers are affixed to the ends of DLOs that, due to frequent manipulation and their cylindrical nature, may twist in various directions unbefitting of ideal recognition conditions. This is clearly not the optimal use case for this tool. The marker's pattern, moving in direct tandem with the DLO it is affixed to, may not always be facing the visual sensors, or may be obstructed from that camera's view. Therefore, although fiducial markers offered a quick setup of live pose estimation, the conditions they require to operate do not align with the pose estimation demands required for this body of research.

4.3.1.2 RGBD Image Processing

Across all metrics, pose estimation accomplished with the RGB and depth image processing technique resulted in the highest success rates for identification and grasping, as well as the lowest mean ground-truth distance error. This approach's success can largely be attributed to the computer vision techniques employed, where the segmentation step of isolating the connector from the environment grants the system an accurate representation of the connector's actual pose. To begin, once any amount of the matching red hue is visible, the system is able to construct a point cloud representation for the connector. The approach's flawless identification rate is due to this fact. Even if only a minute fragment of red is visible, segmentation remains feasible, and a point cloud with at least two points can establish both a position and orientation estimation.

Assuming full view of the connector, the point cloud is automatically positioned correctly in the world frame given its generation from the camera frame, with accuracy in placement subject to the accuracy of said camera's depth image capture. Therefore, defining the center point of this point cloud as the position of the pose frame results in a correspondingly accurate estimation, evidenced by the lowest mean ground-truth distance error. The estimated pose frame aligns exactly with where the ground-truth frame is measured and defined, as they represent the same point in 3D space. The distance error, however, does increase when occlusion is present. Since the position is determined by the center of the point cloud, its coordinate value is biased toward the part of the red connector that the system can view and segment. As a result, while the system can identify the connector, the calculated position might not be entirely accurate. This discrepancy is evident in the increasingly poor results from trials 16 through 27. This supports the subtheme of accomplishing retrieval and maximizing robustness with the minimum level of accuracy necessary, as a grasp may still be possible despite partial segmentation of the connector end.

Segmenting the visible parts of the connector is highly reliable for identification, while still providing reasonably accurate positional estimations. However, when only a small portion of the connector is visible, orientation estimation suffers dramatically. A smaller segmented point cloud implies a shorter distance between the maximum and minimum points used to draw the direction vector. Conversely, a longer distance obtained from the full length of the connector allows for greater orientation accuracy, as it captures and reflects more of the true direction. This concept is illustrated by the rise in grasping failures as the occlusion of the connector end increases between configurations 7 to 9. Because orientation is calculated in the same way under the machine learning pose estimation approach, the same pattern emerges with increasing grasp failures among its trials using these configurations. Orientation accuracy also tends to decline when estimating the pose of a connector end that is almost perfectly parallel (configuration 2) or perpendicular (configuration 4) to the viewing camera, evidenced by their corresponding grasp failures. Due to the nearly flat horizontal and vertical views of the connector in these configurations, the system seems to struggle in grasping its depth dimension. However, this may be attributed to the limitations of the visual sensor, particularly the depth imaging capabilities of the camera. Even under optimal conditions and lighting, the depth images returned by the Intel RealSense cameras introduced certain degrees of noise, affecting both orientation and position estimation. This also helps account for why this has the highest variance and standard deviation among approaches. Segmented point clouds for connector end configurations that exhibit clear 3D dimensionality are therefore only generally accurate, limited by the hardware of the visual sensor. However, when faced with a flat view, the segmented point clouds tend to severely misrepresent the object's depth. Figure 59 depicts two captures of the segmented point cloud and the corresponding calculated orientation for a connector that is nearly parallel to the camera, taken 3 seconds apart. This issue could be resolved by acquiring additional depth images of the object or replacing the visual sensor with one offering higher resolution depth imaging. Alternatively, the entire camera could be repositioned, such as by the arm camera, as to afford it a more optimal view that perceives dimensionality of the connector [14].

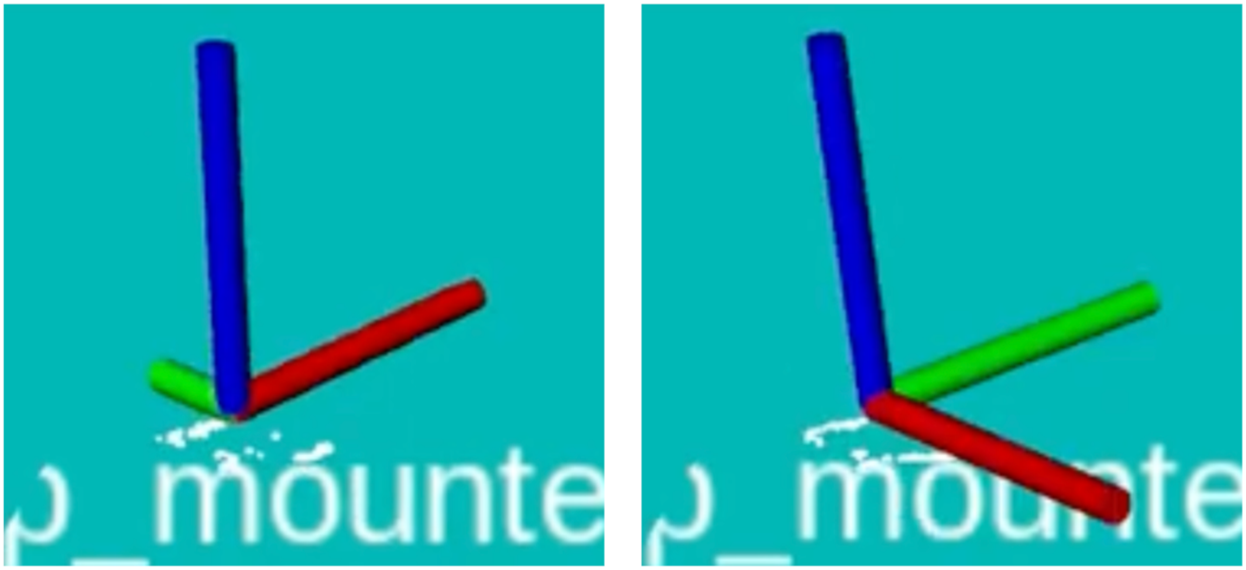


Figure 59: Noisy segmented point clouds for a nearly horizontal connector pose, captured 3 seconds apart

4.3.1.3 Machine Learning

Because the machine learning pose estimation technique shares its orientation calculation with RGBD image processing (as detailed in *Section 4.3.1.2*), this approach is primarily a position estimation technique rather than a complete offering of state estimation. This is akin to the work of [38], where depth related aspects of state estimation did not rely on machine learning principles.

For the calculation of position, the machine learning approach was relatively imprecise and had the largest average distance error, as evidenced by its large variations in mean ground-truth distance error between configurations (Figure 49). Despite this, the positioning of this approach did result in variance and standard deviation results similar to those of the RGBD approach. The ground-truth distance error of the machine learning approach may not necessarily be the fault of the machine learning algorithm, but rather how the generated pose estimation frame’s position is defined. The approach utilizes the center of the bounding box as the supplied positional coordinate, which introduces inaccuracy as it often does not tightly bind the actual connector’s pose. If there's additional padding between the edges of the bounding box and the connector, or if the orientation of the connector causes its center to deviate from that of the bounding box, the ground truth distance error increases. In configurations 2 and 4, the DLO is set such that it is horizontal and vertical to the camera’s view. With little rotation about each axis, the machine learning approach accurately calculated position as the bounding box origin more closely aligned with the connector.

The machine learning approach also performed poorly in detecting the connector end behind occlusions, as evidenced by its failures to identify and grasp under DLO configurations 7 through 9. Neural networks gain the ability to accomplish a task by learning from supplied data. Although the training set employed a series of augmentations, none of the images directly showed the connector behind obstacles. This issue could be resolved by retraining the model with more images of the connector under different occlusion scenarios.

4.3.1.4 Identification and State Estimation Summary

In evaluating the performances of each implemented pose estimation technique, the most significant takeaway is that the approach selected for the system’s needs should consider the context in which it will be used. The pose estimation technique should first and foremost identify and pose estimate for the connector end within the environment. While all three approaches are capable of achieving this, some are better suited to the specific conditions of this workspace’s anticipated conditions. The most prevalent are the infinite number of possible poses the DLO may take from manipulation, as well as any number of obstacles that may be present in the scene.

Additionally, the system must be independent, and therefore employ robust methods as the task must be performed in an environment expecting low to zero gravity and little to no human intervention.

Supported by the experiment results, the RGBD image processing approach best answers the system's needs for pose estimation. This approach proved highly reliable in identification in the presence of obstacles, as segmentation confirms the presence of the connector upon seeing any amount of the matching red color. From the segmented point cloud, a position and orientation can be calculated. Figure 50 supports the approach's consistent and accurate pose estimations, relative to the ground truth and other approaches. Although orientation estimation especially suffered given a sparse point cloud, the other techniques also failed to accomplish this task. This can be resolved by repositioning the camera sensor to get a better view of the connector, only feasible if the object's general location is identified. Across all configurations, this approach's orientation estimation technique resulted in a higher number and percentage of successful grasps made, as seen in Figure 49.

Besides pose estimation capability, another consideration in selection is the conditions the approach requires to operate. RGBD image processing requires only the RGB and depth images published by the system's camera, as well as information about the connector, specifically its color. Fiducial markers require the production of a physical tag and mounting it to the object of interest. The machine learning approach, though also only requiring RGB and depth images, is expensive to run. Running the model through the API, as done in this thesis, requires an internet connection. If running the model locally on the system, heavy computing power is necessary to support the hundreds of nodes. Furthermore, the training process for the model is uncertain, as there is no definitive answer for how many images are enough or when the training is considered complete. RGBD image processing is relatively cheap in computing cost, requires no modification to the target DLO, and needs only image files to accomplish pose estimation. Its ability to accomplish the task of pose estimation is also verifiable through the testing of its unchanging, underlying algorithms. Therefore, although three pose estimations were developed and presented, only the RGBD image processing approach seems suitable for the application of DLO manipulation in a busy and compact zero gravity environment.

4.3.2 Slip Detection

Equipped with only camera sensors, the system must rely solely on visual data to detect a slip from its robotic arm end effectors. This work presented a method in which a threshold is imposed on the Euclidean distance between the position of the arm's gripper and estimated position of the connector end. Through repeated slipping trials of this technique validation experiment, it was observed that the system was most successful in detection when the connector slipped by a significant distance and far from the manipulating arm. This corresponds to the two causes of failure described in the experiment's results: loss of vision monitoring the connector end and not surpassing the distance threshold despite a slip.

Among the tested configurations, the horizontal *X* coordinate of the slip point mattered significantly less than the vertical *Y* position in detection success rate. Slip points 1, 2, and 3 all dropped the connector end from a higher vertical position, allowing for gravity to carry the DLO further downwards and result in higher slip detection rates (Figure 52). This increased distance easily surpassed the set threshold, clearly defining a slip. Although slip point 6 constituted a lower slipping height, it was still easily recognized as the connector sprung back to an extended configuration, moving far from its original slipping point. Slip points 4 and 5 dropped the connector end from a low height, leaving little height for it to fall and settle. Despite a slip occurring, the shorter vertical distance led to less movement, making it harder to exceed the slip threshold resulting in its low slip detection rate. A majority of total failures (Figure 53), especially under low height slip points, were caused by this failure to surpass the distance threshold. To address this issue, the threshold limit could be raised or lowered. However, this introduces its own set of issues: a low threshold might generate false positives during normal manipulation, while a high limit could overlook slips, especially if the connector end moves only slightly after a slip incident.

The system must constantly monitor the state of the connector end, and compare that position with where the system expects it to be to make judgements concerning its status. Because this technique is severely dependent on visual data, any obstructions to the camera's view also interrupted its ability to detect slips. Slipping by a considerable distance, as seen in slip points 1, 2, 3, and 6, caused the connector end to spring away from the manipulating arm, resulting in high detection rates. No obstacles were introduced into the environment for this test, but the robotic arm manipulating the DLO served as an inherent obstruction. If the cable moves only slightly or comes to rest in a position where it's obstructed by the arm or other environmental objects, this approach is rendered ineffective. However, granted that this system is only equipped with visual sensors, slip detection by comparing Euclidean distance against a threshold is proven to be a viable strategy if supplied with constant vision data of the connector end.

4.3.3 Environment Spiral Search Pattern

The Spiral Search algorithm is designed to inspect areas of the workspace that are obstructed by environmental obstacles, blocked from view of the primary rear mounted camera. Out of the 36 trials, varying in selected obstacle and starting position, the Spiral Search pattern successfully located the connector end 33 times. This high success rate can be attributed to two factors: the arm camera's broad FOV and the use of information regarding the connector's last known location.

The environment in which this system was tested is dimensionally small, simulating the compact volumetric constraints of an actual robotic workspace aboard a SmartHab vehicle. The selected cameras, limited only by their own hardware, are capable of capturing wide views over the workspace. As a result, successful identification often came from viewing large portions of the environment rather than from a specific placement location. This rendered the selection of an obstacle that meaningfully impeded identification futile as the expansive views had little trouble looking around them. Identification was further boosted by the robust pose estimation technique provided by RGBD image processing, capable of discerning any fraction of DLO from the environment through color segmentation. It is important to note that this experiment assessed whether identification was possible through the initiation of a spiral search, and not whether the resulting pose estimation was suitable for grasping. Thus, locating the connector end at a given main or subpoint might ultimately be pointless, despite identification afforded by the camera’s wide FOV.

All of this does not imply, however, that optimal camera placement was not beneficial. Initiating a search from the connector’s last known location clearly accelerated the search, as evidenced by Figure 56’s mapping of each trial’s completion times, sorted by that search’s initial starting position. A search conducted from the connector's last known location, as opposed to the origin of the workspace, yielded shorter total search time durations, often by a wide margin. Among runs with identical trial variables, the minimal differences in time likely stem from the system's pose estimation calculations since the search itself was consistent throughout. These results are in line with similar occlusion search approaches that target specific areas depending on the target's most probable locations [44]. Starting a search from the workspace's origin can be inefficient, as it involves examining areas where the connector is less likely to be found. All 3 unsuccessful trials from the experiment began their search from the origin, ultimately failing due to timing out of the allotted trial time. Although it might have been possible to eventually locate the connector after exploring all points along the pattern, this would be too time consuming and therefore, not feasible.

Another contribution to the failures lies in the settings used in the algorithm of the search pattern. Across all trials, the distances traveled between two points and the viewing angles at those points remained the same no matter what the obstacle was or the starting position. This strategy is extremely rigid, with inflexibility leading to a series of issues and potential areas of improvement. From an optimization standpoint, the wide FOV rendered many subpoints redundant, particularly 3 and 7. The perspectives they offered overlapped substantially with other segments of the pattern, squandering valuable search time by reexamining already covered areas. The results presented in Table 8 and Figure 57 support this notion, wherein most identifications were accomplished by a few subpoint poses. Subpoints 1, 2, and 8 accounted for 90.9% of all successful trials (30/33), with the remaining 8.1% of identifications accomplished by subpoints 6 and 7. These likely experienced the greatest identification success rates due to gravity pulling the connector into the bottom half of the workspace, making views that look downward over obstacles perform better. Subpoints 1 and 2 are also the first to be checked. This design choice is intentional because they are ideally located to find the connector, so they should be reviewed first, and then skipping over the successive subpoints. In this experiment, the main point, along with subpoints 3, 4, and 5, are almost entirely ineffective because they yielded no successful identifications. The system wastes time positioning the camera arm and examining the view in these areas already covered, or that will be better covered by a later subpoint. This might not hold in a genuine zero-gravity environment. If unaffected by gravity, the DLO would not tend to drift or settle downwards as seen in this experiment.

Although the subpoints might have too much overlap in their views of the workspace, the camera arm's viewing angles also conversely create blind spots in its search. Success in these cases depends on the hope that another subpoint will cover these blind spots, but this is certainly not guaranteed. These blind spots most often occurred because the connector end settled too close to an obstacle, or behind a part of the object that the camera arm has difficulty seeing behind. At no point during the search does the arm look entirely parallel behind an object, which could have resulted in earlier identification, or covered scenarios in which the connector end was completely missed.

Despite these limitations, the system still achieved a high identification rate using the spiral search pattern. With a wide field of view, precise camera placement was found to be less critical, allowing for added flexibility in the positions to which the camera arm is directed. To accelerate the search and cover difficult blind spots missed by the current setup, the search pattern can be optimized by adjusting the points to check, the distances to move, and the orientations from which to view the workspace.

4.3.4 Full Scenario Demonstration

Three in-depth demonstrations are presented, illustrating the potential of this system to resolve the slip anomaly issue from end to end with the developed technologies described in this thesis. Covered in the solution demonstration are capabilities for monitoring the state of the DLO, detecting any slip of its connector end during manipulation, searching the environment to locate it in the event of a slip, pose estimation once found, and finally calculation and execution of a grasp plan for retrieval to bring the DLO back to a known, nominal state. Although successful, the system remains inaccurate to its intended use case as all technique validation experiments and demonstrations were performed in a standard gravity environment. Though mimicked by a stiffened cable, the system remains untested against DLO behavior in zero gravity. Additionally, only one DLO was designed and used throughout this body of work. The system's ability to identify and pose estimate for DLOs of various colors and sizes also remains unproven. Finally, while notable progress was achieved in robotic capabilities despite the constraint of relying solely on visual sensors, incorporating additional types of, or more advanced sensors could make the system much more robust and efficient. Visual sensors with enhanced resolution or processing power could address some of the noise-related issues mentioned, while a broader range of sensors could compensate for the limitations of depending on cameras alone. However, even with just two cameras, the system accomplished a great deal, demonstrating that it is a viable solution to DLO slip detection and response.

5 Conclusion and Future Work

5.1 Conclusion

In this thesis, a selection of individual robotic capabilities is developed and integrated to address the need for DLO slip anomaly detection and response handling in an autonomous space habitat. A system, composed of two 6-DOF robotic arms and two RGB and depth capable camera sensors, is proposed to adequately respond to these needs.

At the root of this system's functionality is a dependable method for determining DLO pose, as state estimation of the object to be manipulated is necessary throughout task execution. Rather than estimating the state of the entire DLO, only the connector end under manipulation is considered. Obstacles may be present in the workspace blocking camera lines of sight, and the DLO itself may hold unpredictable configurations from robotic manipulation, making robustness in the approach a key factor. Three methods are presented for use by either camera, each with their own unique set of prerequisites, advantages, and disadvantages in utilization. Fiducial markers, requiring physical tags and RGB images, proved to be both extremely accurate and precise. This required consideration in placement to avoid damage of the tag, as well as positional offsetting for proper representation of the entity it stood for. Under situations where the tags were occluded or not directly facing the camera, however, fiducial markers faltered as their patterns were not available for view. This hindered the pose estimation and grasping process, resulting in success rates of 48.1% during its trials in the technique validation experiment. Although fiducial markers are unfit for the rigors of constant DLO movement and manipulation, they show promise for use in representing less busy entities, such as the battery ORU block or stowing point. The next approach involved processing both RGB and depth images such that a point cloud representation of the connector end is isolated from its environment, achieved through color segmentation. This approach yielded highly encouraging results, reaching identification and grasp success rates of 100.0% and 74.1% respectively. Robustness is achieved through the point cloud formulation process, as only a sliver of red color seen by the camera is necessary to identify and generate a rough position. Though orientation calculation success decreased under heavy occlusion, it generated enough information for launching other methods, such as an arm camera search for an improved viewing angle. The final approach explored was the utilization of machine learning algorithms to detect the connector in an RGB image and determining a frame position from the pixel values. A trained model proved relatively successful but encountered difficulty under occlusions. Additionally, no orientation calculation methods were proposed, requiring the approach to borrow from RGBD image processing. This approach achieved success rates of 77.8% and 64.0% for identification and grasping trials, respectively. Although further remedial training of the algorithm could resolve these issues, it calls into question when enough training is achieved. Additionally, the model is computationally expensive to run compared to the inexpensive processing of the other two approaches. For these reasons, as well as its robust attributes proven through experimentation, the RGBD image processing approach is deemed most fit for DLO connector end pose estimation.

With a pose estimation technique validated through repeated trials of the designed parametric study, the system must next detect slips of the connector end from the right manipulating arm's end effector. Limited to only visual sensors, an approach was developed that measures the live three-dimensional Euclidean distance between the estimated position of the connector end and the arm's gripper. If within grasp, this distance should be within a distance range. Therefore, a slip can be quantified by surpassing this set threshold. While this method saw a detection success rate of 77.2%, its failures are attributed to the loss of vision or failure to surpass the threshold despite a slip occurring. These are the limitations due to depending on visual sensors only and can be improved by equipping the system with more varieties of sensors, namely tactile.

Once a slip is detected, the system may not be able to immediately acquire sight of the connector if it were to settle behind an obstacle, blocking it from view of the primary rear camera. To remedy this, a spiral search pattern of the workspace was developed for maneuvering a secondary camera affixed to a second arm around the environment. Search of the loose connector end employing this technique resulted in 91.7% successful identifications among the total trials. This success is attributed to the utilization of the connector end's most probable location, in which the cable's last known position is supplied to the system to initiate searching from. Launching a search in this manner produced overall lower weight times and a higher success rate. Success also stemmed from the camera's wide field of view, which made even non-optimal segments of the search pattern suitable for identification, as evidenced by the breakdown of which pattern points were credited with the most success.

Connector end pose estimation, calculated distance-based slip detection, and environment spiral search are all individual capabilities designed to solve their own confined problems. In tandem, the system takes advantage of each to offer an end-to-end solution for the overall slip anomaly problem. Monitoring, slip detection, search identification, and pose estimation are accomplished via the three state estimation techniques. The Euclidean threshold approach to quantifying a slip flags the anomaly with only visual data. In the event that the loose connector is blocked from view, a search protocol can be initiated to further inspect the environment. The entire solution, compiled into three separate demonstrations, establishes the system's validity in autonomously monitoring, detecting, and resolving the DLO manipulation slip anomaly problem.

5.2 Future Work

The state of the system, as described at the conclusion of this work, would benefit significantly from future efforts aimed at increasing its robustness or further enhancing its newly developed capabilities. The first step is to address the shortcomings of the pose estimation, slip detection, and active search strategies identified in the **Section 4**. These areas for improvement can then inspire new avenues in extending the system to better fulfill its current task, or to handle related DLO manipulation tasks more effectively. Ideas for the future building and expansion of this research are next presented.

5.2.1 Identification and State Estimation

Each of the three pose estimation techniques was found to have their own set of unique advantages and disadvantages. Rather than relying solely on one technique, the system could benefit from using them in tandem, or at least selecting between approaches based on the task at hand.

Fiducial markers, though unreliable if not facing the camera or under occlusion, did prove highly accurate and precise when estimating the pose of its own tag. In situations where an entity is not expected to move unpredictably, or remain constantly in a camera's line of sight, visual servoing with fiducial markers is a promising approach.

When the movement of an entity, such as the DLOs explored in this work, is unstable, the RGBD image processing approach is best used for its robust identification and pose estimation.

Endeavors could be made to improve its accuracy, either through computer vision techniques or improved depth images. This accuracy should further extend into orientation calculation, and ensuring rotations computed accurately reflect the actual pose of an item no matter how miniscule its corresponding segmented point cloud is. Orientation calculation could, alternatively, be entirely overhauled using the more robust, tested, and efficient principal component analysis (PCA) data processing technique. Akin to this work's approach of fitting a vector between the length of the connector end, PCA could be used to fit a cylindrical ellipsoid to the segmented point cloud, enabling generation of the principal axes to calculate orientation.

Another area of improvement could be replacing the current color segmentation approach, as it is highly subject to lighting conditions, and requires bold outstanding colors to discern from the environment. Objects of interest must be of that color, and the system currently has no applicable method to discern separate objects sharing the segmentation color. Also, any future objects of interest, and their chosen colors, must be registered to the system. One possible remedy for differentiation is the application of the Iterative Closest Point (ICP) algorithm, which can be utilized to correlate the shape of a point cloud to a computer-aided design model, and therefore distinguish connectors based on this matching. This approach was implemented, but ultimately abandoned due to low depth image resolutions returned by the system's visual sensors between two distinct connectors. ICP alone would also still fail at discriminating between multiple cables with the same connector head. Moving forward, further testing of ICP's application in this context, and fusion with other identification methods may resolve the issues described.

Lastly, much work could be done for the machine learning approach. The most obvious next step would be training the model to recognize an object under occlusion, which can be accomplished by supplying images of an object partially occluded behind obstacles. Because the machine learning approach simply borrowed its orientation estimation technique, an interesting future direction could be reliably estimating depth and orientation from the RGB images alone. Depth could be calculated by creating labels for the connector at different distances from the camera and using pixel comparisons of size to estimate this *Z* values. A set of class labels could also be defined for the object in various orientations, and the model could be trained to predict an orientation from a given image. These class labels could be mapped to a library of preset poses, which the robotic agent could use as it maneuvers to the given position. Position estimation could also be improved by using oriented bounding boxes, a more advanced form of prediction unsupported by this system's model host Roboflow.

5.2.2 Slip detection

The system's ability to detect slips was severely limited by only having access to visual data. Loss of vision severely hindered monitoring of the connector end and therefore detection of slips. Future work could see the addition of more cameras around the environment when limiting its sensors to strictly a visual based system. However, more varieties of sensors, such as the addition of tactile force sensors on the robotic arm's end effectors, could greatly bolster the system's awareness of an object's grasp status. If vision is lost, or if the set slip threshold is not surpassed, the force data from these sensors could still reveal a slip. An object in the grasp of a connector would result in a force reading, while an empty set of grippers would show near-zero values, as depicted in Figure 60. A combination of these approaches would greatly enhance the system's ability to detect slips as cameras, alone, are not solely responsible for recognizing the anomaly.

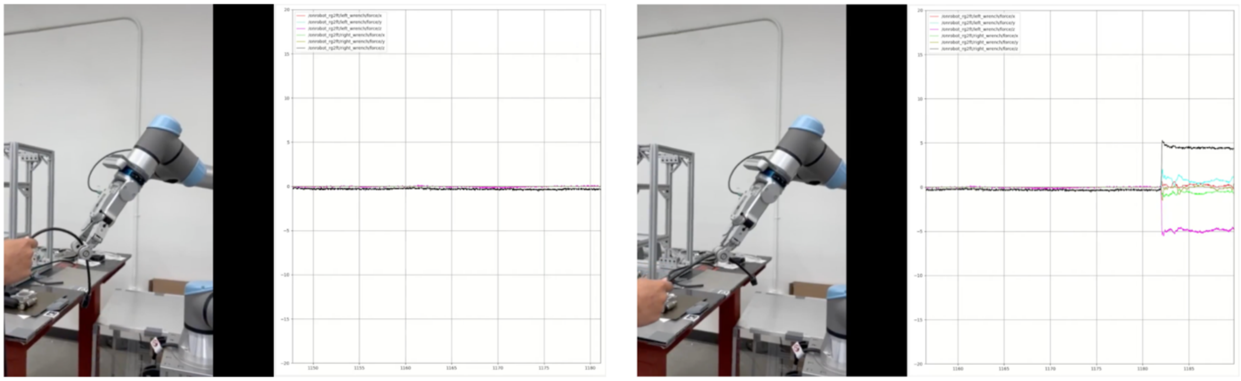


Figure 60: Grasped (left) and empty (right) readings from a force sensor on the end effector of a UR5e robotic arm

5.2.3 Environmental Spiral Search Pattern

Though the spiral search pattern presented in this work yielded high identification rates, much of that success stems from the wide, and therefore camera-placement forgiving FOV afforded by the Intel RealSense cameras utilized. The locations checked by the system were not intelligently chosen. Instead, one can start with robustly checking the entire workspace. A future direction could be enhancing the spiral search by intelligently tuning the angles it views the environment, and the distances it should move to locate an object of interest most efficiently.

Alternatively, for the purpose of relocating a loose connector end, search could be avoided entirely if any portion of the entire DLO is visible. Akin to the work of [41], the system could generate, depending on the length and direction of cable it can see, where the occluded portion may have settled. Predictive capabilities could be combined with active search to further improve search speed and accuracy. Alternatively, if equipped with the tactile force sensors described in 5.2.2, the lost connector end could also be retrieved by simply tracing the end effector down the length of the DLO. The total length of the DLO could be supplied to the system, and as long as a nonzero force is read, tracing continues until the total DLO’s length is traveled, thus reaching the connector end.

References

[1] A. E. Rollock and D. Klaus, “A Historical Analysis of Earth-Independence in Human Spaceflight Missions,” in *AIAA SCITECH 2023 Forum*, National Harbor, MD & Online: American Institute of Aeronautics and Astronautics, Jan. 2023. doi: 10.2514/6.2023-0267.

[2] H. Rozas, B. Basciftci, and N. Gebraeel, “Data-driven joint optimization of maintenance and spare parts provisioning for deep space habitats: A stochastic programming approach,” *Acta Astronaut.*, vol. 214, pp. 167–181, Jan. 2024, doi: 10.1016/j.actaastro.2023.10.028.

[3] B. N. Griffin, “Why Deep Space Habitats Should be Different from the International Space Station,” in *AIAA SPACE 2016*, Long Beach, California: American Institute of Aeronautics and Astronautics, Sep. 2016. doi: 10.2514/6.2016-5278.

[4] A. E. Rollock and D. M. Klaus, “Defining and characterizing self-awareness and self-sufficiency for deep space habitats,” *Acta Astronaut.*, vol. 198, pp. 366–375, Sep. 2022, doi: 10.1016/j.actaastro.2022.06.002.

[5] P. Henson *et al.*, “An Environmental Control and Life Support System (ECLSS) for Deep Space and Commercial Habitats,” in *ICES 2020: 50th International Conference on Environmental Systems*, Lisbon, Portugal: Honeywell International Inc., Jul. 2021.

[6] J. F. Russell and D. M. Klaus, “Maintenance, reliability and policies for orbital space station life support systems,” *Reliab. Eng. Syst. Saf.*, vol. 92, no. 6, pp. 808–820, Jun. 2007, doi: 10.1016/j.ress.2006.04.020.

[7] D. Rojas, “Autonomous Robotic Manipulation of Deformable Linear Objects During Deep Space Maintenance and Repair Procedures,” UC Davis, Davis, CA, 2022.

[8] P. Li and X. Liu, “Common Sensors in Industrial Robots: A Review,” *J. Phys. Conf. Ser.*, vol. 1267, no. 1, p. 012036, Jul. 2019, doi: 10.1088/1742-6596/1267/1/012036.

[9] B. Fan, Y. Dai, and M. He, “Rolling Shutter Camera: Modeling, Optimization and Learning,” *Mach. Intell. Res.*, vol. 20, no. 6, pp. 783–798, Dec. 2023, doi: 10.1007/s11633-022-1399-z.

[10] H. Xu, J. Xu, and W. Xu, “Survey of 3D modeling using depth cameras,” *Virtual Real. Intell. Hardw.*, vol. 1, no. 5, pp. 483–499, Oct. 2019, doi: 10.1016/j.vrih.2019.09.003.

[11] L. Keselman, J. I. Woodfill, A. Grunnet-Jepsen, and A. Bhowmik, “Intel RealSense Stereoscopic Depth Cameras,” 2017, doi: 10.48550/ARXIV.1705.05548.

[12] R. Bodor, A. Drenner, P. Schrater, and N. Papanikolopoulos, “Optimal Camera Placement for Automated Surveillance Tasks,” *J. Intell. Robot. Syst.*, vol. 50, no. 3, pp. 257–295, Oct. 2007, doi: 10.1007/s10846-007-9164-7.

[13] A. M. Heyns, “Optimisation of surveillance camera site locations and viewing angles using a novel multi-attribute, multi-objective genetic algorithm: A day/night anti-poaching application,” *Comput. Environ. Urban Syst.*, vol. 88, p. 101638, Jul. 2021, doi: 10.1016/j.compenvurbsys.2021.101638.

[14] F. Kececi, M. Tonko, H.-H. Nagel, and V. Gengenbach, “Improving visually servoed disassembly operations by automatic camera placement,” in *Proceedings. 1998 IEEE International Conference on Robotics and Automation (Cat. No.98CH36146)*, Leuven, Belgium: IEEE, 1998, pp. 2947–2952. doi: 10.1109/ROBOT.1998.680742.

[15] N. Jamil, T. M. T. Sembok, and Z. A. Bakar, “Noise removal and enhancement of binary images using morphological operations,” in *2008 International Symposium on Information Technology*, Kuala Lumpur, Malaysia: IEEE, 2008, pp. 1–6. doi: 10.1109/ITSIM.2008.4631954.

[16] R. Szeliski, *Computer vision: algorithms and applications*. in Texts in computer science. London ; New York: Springer, 2011.

[17] Y. Li, “Object Detection and Instance Segmentation of Cables,” KTH ROYAL INSTITUTE OF TECHNOLOGY, STOCKHOLM, SWEDEN, 2019.

[18] M. L. Comer, “Morphological operations for color image processing,” *J. Electron. Imaging*, vol. 8, no. 3, p. 279, Jul. 1999, doi: 10.1117/1.482677.

[19] C. Jia, T. Yang, C. Wang, B. Fan, and F. He, “A new fast filtering algorithm for a 3D point cloud based on RGB-D information,” *PLOS ONE*, vol. 14, no. 8, p. e0220253, Aug. 2019, doi: 10.1371/journal.pone.0220253.

[20] M. Kalaitzakis, B. Cain, S. Carroll, A. Ambrosi, C. Whitehead, and N. Vitzilaios, “Fiducial Markers for Pose Estimation: Overview, Applications and Experimental Comparison of the ARTag, AprilTag, ArUco and STag Markers,” *J. Intell. Robot. Syst.*, vol. 101, no. 4, p. 71, Apr. 2021, doi: 10.1007/s10846-020-01307-9.

[21] A. Sagitov, K. Shabalina, R. Lavrenov, and E. Magid, “Comparing fiducial marker systems in the presence of occlusion,” in *2017 International Conference on Mechanical, System and Control Engineering (ICMSC)*, St.Petersburg, Russia: IEEE, May 2017, pp. 377–382. doi: 10.1109/ICMSC.2017.7959505.

[22] “Detection of ArUco Markers.” Accessed: Mar. 12, 2024. [Online]. Available: https://docs.opencv.org/4.x/d5/dae/tutorial_aruco_detection.html

[23] A. B. Craig, *Understanding augmented reality: concepts and applications*. Amsterdam: Morgan Kaufmann, 2013.

[24] J. Gower and G. Dijksterhuis, *Procrustes problems*, Reprint. in Oxford statistical science series, no. 30. Oxford: Oxford Univ. Press, 2009.

[25] S. K. Zhou, D. Rueckert, and G. Fichtinger, Eds., *Handbook of medical image computing and computer assisted intervention*. in The Elsevier and MICCAI Society book series.

London, United Kingdom ; San Diego, CA, United States: Academic Press is an imprint of Elsevier, 2020.

[26] V. Piuri, S. Raj, A. Genovese, and R. Srivastava, Eds., *Trends in deep learning methodologies: algorithms, applications, and systems*. in Hybrid computational intelligence for pattern analysis and understanding series. London, United Kingdom ; San Diego, CA, United States: Academic Press, 2021.

[27] S.-C. Huang and T.-H. Le, Eds., *Principles and labs for deep learning*. Walthum: Elsevier, 2021.

[28] R. Venkatesan and B. Li, *Convolutional neural networks in visual computing: a concise guide*. Boca Raton, FL: CRC Press, 2018.

[29] J. Ren and H. Wang, *Mathematical methods in data science*. Amsterdam: Elsevier, 2023.

[30] K. I. Tsianos, I. A. Sucan, and L. E. Kavraki, “Sampling-based robot motion planning: Towards realistic applications,” *Comput. Sci. Rev.*, vol. 1, no. 1, pp. 2–11, Aug. 2007, doi: 10.1016/j.cosrev.2007.08.002.

[31] S. Li *et al.*, “Proactive human–robot collaboration: Mutual-cognitive, predictable, and self-organising perspectives,” *Robot. Comput.-Integr. Manuf.*, vol. 81, p. 102510, Jun. 2023, doi: 10.1016/j.rcim.2022.102510.

[32] L. Yang, J. Qi, D. Song, J. Xiao, J. Han, and Y. Xia, “Survey of Robot 3D Path Planning Algorithms,” *J. Control Sci. Eng.*, vol. 2016, pp. 1–22, 2016, doi: 10.1155/2016/7426913.

[33] L. E. Kavraki, P. Svestka, J.-C. Latombe, and M. H. Overmars, “Probabilistic roadmaps for path planning in high-dimensional configuration spaces,” *IEEE Trans. Robot. Autom.*, vol. 12, no. 4, pp. 566–580, Aug. 1996, doi: 10.1109/70.508439.

[34] S. M. LaValle, "Rapidly-exploring random trees: A new tool for path planning," Computer Science Department, Iowa State University, Oct. 1998.

[35] S. B. Niku, *An introduction to robotics analysis, systems, applications*. Upper Saddle River, N.J: Prentice Hall, 2001.

[36] J. R. Mühlbacher and F. X. Steinparz, "Transformation of co-ordinates for robot control," *J. Microcomput. Appl.*, vol. 7, no. 1, pp. 1–17, Jan. 1984, doi: 10.1016/0745-7138(84)90084-8.

[37] H. W. Ryu and J. H. Tai, "Object detection and tracking using a high-performance artificial intelligence-based 3D depth camera: towards early detection of African swine fever," *J. Vet. Sci.*, vol. 23, no. 1, p. e17, 2022, doi: 10.4142/jvs.21252.

[38] Rahul and B. B. Nair, "Camera-Based Object Detection, Identification and Distance Estimation," in *2018 2nd International Conference on Micro-Electronics and Telecommunication Engineering (ICMETE)*, Ghaziabad, India: IEEE, Sep. 2018, pp. 203–205. doi: 10.1109/ICMETE.2018.00052.

[39] N. Rogeau, V. Tiberghien, P. Latteur, and Y. Weinand, "Robotic Insertion of Timber Joints using Visual Detection of Fiducial Markers," presented at the 37th International Symposium on Automation and Robotics in Construction, Kitakyushu, Japan, Oct. 2020. doi: 10.22260/ISARC2020/0068.

[40] G. Yu, Y. Liu, X. Han, and C. Zhang, "Objects Grasping of Robotic Arm with Compliant Grasper Based on Vision," in *Proceedings of the 2019 4th International Conference on Automation, Control and Robotics Engineering*, Shenzhen China: ACM, Jul. 2019, pp. 1–6. doi: 10.1145/3351917.3351958.

[41] C. Chi and D. Berenson, “Occlusion-robust Deformable Object Tracking without Physics Simulation,” in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Macau, China: IEEE, Nov. 2019, pp. 6443–6450. doi: 10.1109/IROS40897.2019.8967827.

[42] Yu-Chi Lin, Shao-Ting Wei, Shih-An Yang, and L.-C. Fu, “Planning on searching occluded target object with a mobile robot manipulator,” in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, Seattle, WA, USA: IEEE, May 2015, pp. 3110–3115. doi: 10.1109/ICRA.2015.7139626.

[43] S. Radmard, D. Meger, J. J. Little, and E. A. Croft, “Resolving Occlusion in Active Visual Target Search of High-Dimensional Robotic Systems,” *IEEE Trans. Robot.*, vol. 34, no. 3, pp. 616–629, Jun. 2018, doi: 10.1109/TRO.2018.2796577.

[44] L. L. S. Wong, L. P. Kaelbling, and T. Lozano-Perez, “Manipulation-based active search for occluded objects,” in *2013 IEEE International Conference on Robotics and Automation*, Karlsruhe, Germany: IEEE, May 2013, pp. 2814–2819. doi: 10.1109/ICRA.2013.6630966.

[45] M. Quigley *et al.*, “ROS: an open-source Robot Operating System,” presented at the ICRA workshop on open source software, in 3.2, vol. 3. Kobe, Japan, 2009, p. 5.

[46] L. R. Ramírez-Hernández *et al.*, “Improve three-dimensional point localization accuracy in stereo vision systems using a novel camera calibration method,” *Int. J. Adv. Robot. Syst.*, vol. 17, no. 1, p. 172988141989671, Jan. 2020, doi: 10.1177/1729881419896717.

[47] opencv dev team, “Camera calibration With OpenCV,” OpenCV. [Online]. Available: https://docs.opencv.org/2.4/doc/tutorials/calib3d/camera_calibration/camera_calibration.html

[48] D. J. Bora, “A Novel Approach for Color Image Edge Detection Using Multidirectional Sobel Filter on HSV Color Space,” *Int. J. Comput. Sci. Eng.*, vol. 5, no. 2, pp. 154–159, Feb. 2017.

[49] L. Dorst, D. Fontijne, and S. Mann, *Geometric algebra for computer science: an object-oriented approach to geometry*. in Morgan Kaufmann series in computer graphics. Amsterdam : San Francisco: Elsevier ; Morgan Kaufmann, 2007.

[50] “Bounding Boxes in Computer Vision: Uses, Best Practices for Labeling, and More,” Aya Data. [Online]. Available: https://www.ayadata.ai/blog-posts/bounding-boxes-in-computer-vision-uses-best-practices-for-labeling-and-more/

[51] C. Shorten and T. M. Khoshgoftaar, “A survey on Image Data Augmentation for Deep Learning,” *J. Big Data*, vol. 6, no. 1, p. 60, Dec. 2019, doi: 10.1186/s40537-019-0197-0.

[52] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You Only Look Once: Unified, Real-Time Object Detection,” 2015, doi: 10.48550/ARXIV.1506.02640.

[53] J. Solawetz, “How to Train a YOLOv5 Model On a Custom Dataset,” Roboflow Blog. [Online]. Available: https://blog.roboflow.com/how-to-train-yolov5-on-a-custom-dataset/

[54] Y. Chiba, “Converting 2D image coordinates to 3D coordinates using ROS + Intel Realsense D435/Kinect,” Medium. [Online]. Available: https://medium.com/@yasuhirachiba/converting-2d-image-coordinates-to-3d-coordinates-using-ros-intel-realsense-d435-kinect-88621e8e733a

[55] J. Tabak, *Geometry*. New York: Chelsea, 2008.