IMPLEMENTING ADVANCED OBJECT INTERACTION IN RESCUE ROBOTICS USING AN AUTONOMOUS MANIPULATOR ARM

THORANIN OONARIYA

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF MASTER OF ENGINEERING IN
AUTOMATION ENGINEERING
DEPARTMENT OF INSTRUMENTATION AND ELECTRONICS ENGINEERING
GRADUATE COLLEGE
KING MONGKUT'S UNIVERSITY OF TECHNOLOGY NORTH BANGKOK
ACADEMIC YEAR 2024

IMPLEMENTING ADVANCED OBJECT INTERACTION IN RESCUE ROBOTICS USING AN AUTONOMOUS MANIPULATOR ARM

THORANIN OONARIYA

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF MASTER OF ENGINEERING IN
AUTOMATION ENGINEERING
DEPARTMENT OF INSTRUMENTATION AND ELECTRONICS ENGINEERING
GRADUATE COLLEGE
KING MONGKUT'S UNIVERSITY OF TECHNOLOGY NORTH BANGKOK
ACADEMIC YEAR 2024

Thesis Certificate

The Graduate College, King Mongkut's University of Technology North Bangkok

Title Implementing Advanced Object Interaction in Rescue Robotics Using an Autonomous Manipulator Arm

By Thoranin Oonariya

Accepted by the FACULTY OF ENGINEERING, King Mongkut's University of Technology North Bangkok in Partial Fulfillment of the Requirements for the Master of Engineering in Instrumentation and Electronics Engineering

.... Dean / Head of Department

(Associate Professor KITTICHAI TANASUBSIN, Ph.D.)

Thesis Examination Committee

.... Chairperson

(Dr. Nattapon Chayopitak)

.... Advisor

(Assistant Professor Jiraphan Inthiam, Ph.D.)

.... Committee

(Assistant Professor Wisanu Jitviriya, Ph.D.)

.... External examiner

(Dr. Nattapon Chayopitak)

Name : Thoranin Oonariya
Thesis Title : Implementing Advanced Object Interaction in Rescue Robotics Using an Autonomous Manipulator Arm
Major Field : Automation Engineering
King Mongkut's University of Technology North Bangkok
Thesis Advisor : Assistant Professor Jiraphan Inthiam, Ph.D.
Academic Year : 2024

ABSTRACT

Manipulator arms in rescue robots remain a challenge in both competitions and real-life situations. In this paper, we present an autonomous in-house manipulator system specifically engineered to enrich object interaction in rescue robotics for the RoboCup Rescue Robot Competition 2024 as a contestant under the iRAP Robot Team. The manipulator is equipped with an RGB-D camera, providing pose estimation of objects by leveraging 3D models generated through prior scanning. YOLOv8 2D object detection is employed to identify objects in the scene, and cropped depth images are used to subsequently generate object point clouds. Iterative Closest Point (ICP) techniques, including point-to-plane, point-to-point, and color registration, are utilized in conjunction to provide accurate alignment between the object's point cloud and the pre-scanned 3D models. These techniques work together to minimize Root Mean Square Error (RMSE) during the alignment process, ensuring reliability of pose estimation before executing robotic movement actions. The system also uses IMU data from the RGB-D camera for manipulator impact detection. The entire system is developed within the Robot Operating System (ROS) framework, utilizing MoveIt for motion planning and control. The methodologies and innovations introduced in this work are adapted for deployment in the challenging and unexpected environments of the RoboCup Rescue Robot Competition.

(Total 48 pages)

Keywords: Object Interaction, RGB-D camera, ICP (iterative closest point),ROS (Robot Operating System)

______________________________________ Advisor

ชื่อ	:	นายธรณินทร์ อุ่นอารีย์
ชื่อวิทยานิพนธ์	:	การพัฒนาเทคนิคขั้นสูงสำหรับการโต้ตอบกับวัตถุในหุ่นยนต์กู้ภัยโดยการใช้แขนหุ่นยนต์อัตโนมัติ
สาขาวิชา	:	วิศวกรรมอัตโนมัติ มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ
อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก	:	ผศ.ดร.จิรพันธุ์ อินเทียม
ปีการศึกษา	:	2567

บทคัดย่อ

แขนกลในหุ่นยนต์กู้ภัยยังคงเป็นความท้าทายทั้งในเวทีการแข่งขันและการใช้งานในสถานการณ์จริง ในงานวิจัยนี้ ได้นำเสนอระบบแขนกลอัตโนมัติ ที่ถูกออกแบบเป็นพิเศษเพื่อเสริมความสามารถในการโต้ตอบกับวัตถุสำหรับงานกู้ภัย โดยตั้งเป้าเข้าร่วมแข่งขัน RoboCup Rescue Robot Competition 2024 ภายใต้ทีม iRAP Robot ระบบแขนกลดังกล่าวติดตั้งกล้อง RGB-D สำหรับประเมินท่าทางและตำแหน่งของวัตถุ โดยอาศัยแบบจำลอง 3 มิติที่ได้มีการเก็บข้อมูลไว้ล่วงหน้า และตรวจจับวัตถุในภาพ 2 มิติโดยใช้โมเดล YOLOv8 จากนั้นนำภาพความลึกที่ถูกตัดเฉพาะส่วนจาการตรวจจับ มาใช้สร้าง point cloud ของวัตถุ ในขั้นตอนการหาตำแหน่งและท่าทางข้อมูล 3 มิติ มีการนำเทคนิค Iterative Closest Point (ICP) ทั้ง point-to-plane, point-to-point และ color registration มาประยุกต์ร่วมกัน เพื่อลดค่า ค่ารากที่สองของความคลาดเคลื่อนเฉลี่ย Root Mean Square Error (RMSE) ในระหว่างการปรับตำแหน่งของ point cloud ของวัตถุให้ตรงกับแบบจำลอง 3 มิติที่มีอยู่ ยิ่งไปกว่านั้นระบบยังใช้ข้อมูล IMU จากกล้อง RGB-D เพื่อช่วยตรวจจับแรงกระแทกที่เกิดขึ้นกับแขนกล ระบบทั้งหมดพัฒนาภายใต้โครงสร้าง Robot Operating System (ROS) และใช้ MoveIt ในการวางแผนและควบคุมการเคลื่อนที่ ทั้งแนวทางและนวัตกรรมที่นำเสนอได้รับการปรับใช้งานเพื่อให้เหมาะสมกับสภาพแวดล้อมที่ท้าทายและคาดเดาได้ยากในการแข่งขัน RoboCup Rescue Robot Competition

(วิทยานิพนธ์มีจำนวนทั้งสิ้น 48 หน้า)

คำสำคัญ: การโต้ตอบกับวัตถุ, กล้อง RGB-D, ICP (การจับคู่จุดแบบใกล้เคียงโดยการทำซ้ำ), ROS (ระบบปฏิบัติการหุ่นยนต์)

______________________________อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก

ACKNOWLEDGEMENTS

This thesis was successfully completed thanks to the invaluable guidance and assistance from Asst. Prof. Dr. Jiraphan Inthiam, my thesis advisor, who generously provided knowledge and advice on both academic matters and everyday life. I also appreciate the constructive feedback from Asst. Prof. Dr. Noppadol Pudchuen, ensuring that this thesis reached its final form. My gratitude further extends to Asst. Prof. Dr. Wisanu Jitviriya and Asst. Prof. Dr. Chirdpong Deelertpaiboon for imparting foundational knowledge in robotics, which has significantly influenced my academic pursuits, as well as for offering guidance during the development of this thesis. I also wish to acknowledge the faculty, staff, and personnel of the Department of Instrumentation and Electronics Engineering for their kindness in providing facilities and documentation essential to the completion of this research.

I would like to express my deepest gratitude to my advisor, colleagues, and members of the iRAP robot club for their invaluable guidance, support, and contributions. Their expertise and insights have been crucial to the successful completion of this work. I am also thankful for the support provided by the Faculty of Engineering, KMUTNB, which enabled me to conduct this research.

Finally, I am profoundly grateful to my parents, relatives, and siblings for the educational opportunities, unwavering support, and encouragement that motivated me throughout my studies. I sincerely hope that this thesis will serve as a valuable resource for anyone interested and as a steppingstone for future developments in this field.

Thoranin Oonariya

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

ArUco	Augmented Reality University of Cordoba
ASTM	American Society for Testing and Materials
CAD	Computer Aided Design
CHOMP	Covariant Hamiltonian Optimization for Motion Planning
DEX	Dexterity
DH parameters	Denavit-Hartenberg parameters
DoF	Degrees of Freedom
EXP	Exploration
FPFH	Fast Point Feature Histograms
ICP	Iterative Closest Point
IK	Inverse Kinematics
IMU	Inertial Measurement Units
iRAP	Invigorating Robot Activity Project
KMUTNB	King Mongkut's University of Technology North Bangkok
kNN	K-Nearest Neighbor Algorithm
LiDAR	Light Detection and Ranging
MAN	Maneuvering
MDH	Modified Denavit-Hartenberg
MOB	Mobility
OMPL	Open Motion Planning Library
PCA	Principal Component Analysis
PRM	Probabilistic Roadmaps
RGBD	Red Green Blue Depth
RMSE	Root Mean Square Error
ROS	Robot Operating System
RRT	Rapidly exploring Random Trees
SLAM	Simultaneous Localization and Mapping
STOMP	Stochastic Trajectory Optimization for Motion Planning
SVD	Singular Value Decomposition
ToF	Time-of-Flight
USAR	Urban Search and Rescue
YOLO	You Only Look Once

CHAPTER 1
INTRODUCTION

1.1 Background

Rescue robots have become indispensable tools in emergencies and disaster areas where human involvement poses significant risks or is altogether impractical. These environments such as collapsed buildings, earthquake zones, or areas exposed to hazardous materials often present extreme conditions where uneven terrain, unstable debris, and limited visibility challenge both the mobility and sensing capabilities of autonomous systems. Consequently, the ability to detect, identify, and manipulate objects within these environments is critical to facilitate essential tasks, including locating and extracting survivors, delivering supplies, and clearing obstructed paths. At the core of effective rescue robotics lies in robust object interaction. Robots must not only navigate complex, unpredictable terrains but also reliably detect, track, and manipulate a broad variety of objects whose shapes, sizes, and positions may vary dramatically. Conventional approaches to object detection and interaction, while continually improving, can still struggle with issues such as partial distortion, poor lighting conditions, and the presence of obstacles in dynamic environments. Accurate object recognition and responsive control strategies are vital for ensuring the robot can adapt to unexpected challenges without necessitating continuous human intervention.

In recent years, advancements in sensor technologies and artificial intelligence have opened new frontiers in autonomous manipulation. By leveraging state-of-the-art 2D and 3D vision systems, rescue robots can construct richer environmental models, enabling more precise localization and control. In particular, RGB-D cameras offer both color (RGB) and depth (D) information, making it possible to generate real-time 3D reconstructions of the surrounding area. This enhanced perception facilitates more accurate object detection and pose estimation, which is essential for successful grasping and manipulation. Additionally, software frameworks like the Robot Operating System (ROS) and motion planning tools such as MoveIt further streamline the development of complicated robotic behaviors by providing modular libraries for tasks like path planning, collision avoidance, and inverse kinematics calculations.

Within the context of the RoboCup Rescue Robot competition, these cutting-edge techniques are put to the test in an environment designed to simulate real-world rescue scenarios. The competition evaluates various facets of a robot's performance from mobility in rough terrain to the dexterity required for handling critical objects under stringent guidelines inspired by established standards, such as ASTM International for Urban Search and Rescue (USAR) Robots. By integrating an advanced robotic arm capable of fully automated operations, teams aim to demonstrate how higher levels of autonomy can significantly increase the robot's versatility while minimizing demands on human operators, especially in fast-paced, high-stakes situations. Through sophisticated control algorithms and state-of-the-art sensors, such an arm can identify, grasp, and manipulate objects with minimal manual input. This higher degree of autonomy not only allows robots to take on tasks that would be hazardous or impractical for humans but also frees operators to focus on strategic decision-making rather than detailed, time-consuming controls. In emergency missions where every second counts the fully automated arm can execute rapid interventions, like closing valves or removing debris, without the constant oversight that simpler systems require. By reducing the manual burden, human operators can better direct their attention to mission-critical analyses, thereby boosting efficiency, reducing errors, and enhancing overall mission success.

Ultimately, this research seeks to further refine and validate the robotic arm's capabilities in object interaction by combining 2D and 3D vision, RGB-D sensing, robust motion planning, and competition-proven performance. Through iterative development and rigorous testing against real-world challenges, the approach aspires to push the boundaries of rescue robotics, contributing to safer and more effective emergency response strategies worldwide.

1.2 Objective

1.2.1 To develop a system capable of detecting and interacting with objects in rescue environments efficiently.

1.2.2 To apply RGBD cameras for 3D object detection and position analysis, utilizing the Iterative Closest Point (ICP) technique to match and align point cloud data between scanned objects and real-world data.

1.2.3 To integrate ROS and MoveIt for planning and controlling the robotic arm's movement in the RoboCup Rescue Robot competition.

1.3 Scope of the Study

1.3.1 Study and understand the ROS and MoveIt systems for planning and controlling robotic arm movements.

1.3.2 Design a system for detecting and interacting with objects in rescue environments using a robotic arm.

1.4 Utilization of the Study

1.4.1 To develop an automated system for object interaction in rescue environments.

1.4.2 To support the work of rescue robot operators by enhancing robotic capabilities.

CHAPTER 2
LITERATURE REVIEW

Research in rescue robotics has been growing rapidly due to the increasing frequency and severity of natural disasters worldwide. The RoboCup-Rescue initiative set an early precedent for this field by illustrating how robots can be effectively deployed in disaster mitigation scenarios. According to the RoboCup-Rescue project [1] focused on developing technology and standardized tests that enable robots to navigate hazardous environments, locate survivors, and perform critical tasks under real-world constraints. More recently, iRAP Robot showcased advancements in robot design and system integration for the RoboCup Rescue league [2], underscoring the competition's contribution to accelerating innovation and pushing the boundaries of robotics research. Based on the literature review, a broad spectrum of robotic processes can be grouped into key thematic areas that capture both foundational theories and practical applications. These areas address various technical challenges from how robots plan and execute movements, to how they detect and manipulate objects in dynamic environments. By examining these categories, researchers can better understand the interconnections among the different facets of robotic systems and pave the way for more robust, efficient, and adaptable solutions. The following sections outline five principal domains identified in the literature:

2.1 Motion Planning and Trajectory Optimization
2.2 Bin Picking, Object Detection, and Pose Estimation
2.3 3D Mapping, Localization, and Sensor Sensitivity
2.4 ICP (Iterative Closest Point) and Alignment Algorithms
2.5 Impact Sensing

2.1 Motion Planning and Trajectory Optimization

A core challenge in rescue robotics is ensuring safe and efficient motion planning, particularly in debris-filled or unpredictable terrains. Traditional path planning methods can be computationally expensive or prone to local minima when dealing with high-dimensional manipulator arms. MoveIt provides a variety of motion planners such as OMPL, CHOMP, STOMP, and the Pilz Industrial Motion Planner to calculate and plan trajectories for a manipulator. In this work, I will focus on using MoveIt to control the manipulator arm that the structure of the system is same as **FIGURE 2-1**.

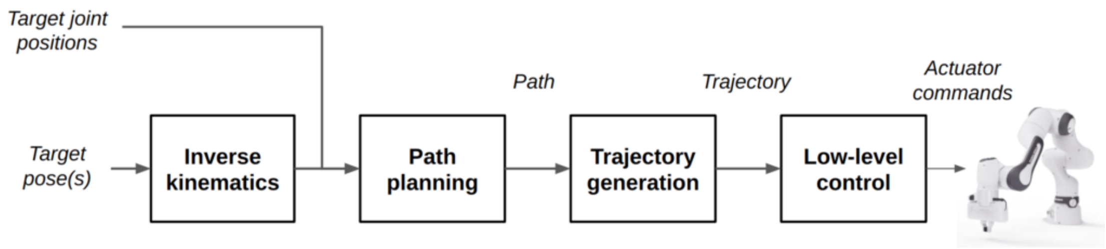

FIGURE 2-1 The Diagram of Control Manipulator Procedure

2.1.1 OMPL (Open Motion Planning Library)

In MoveIt, one of the most widely used frameworks for motion planning is the Open Motion Planning Library (OMPL). It provides a range of sampling-based algorithms, such as Rapidly exploring Random Trees (RRT) and Probabilistic Roadmaps (PRM). These algorithms work by randomly sampling the robot's configuration space to find feasible paths from a start to a goal pose, focusing primarily on collision-free feasibility. As a result, they are highly effective in high-dimensional spaces, which makes them suitable for complex robotic arms. While they excel at quickly finding feasible solutions, these paths may require additional smoothing or optimization if a more optimal path such as one minimizing travel distance or time is required.

2.1.2 CHOMP (Covariant Hamiltonian Optimization for Motion Planning)

CHOMP takes a gradient-based approach to trajectory optimization, starting with an initial trajectory (often a simple interpolation) and then iteratively refining it to reduce collision risk and enhance smoothness. By leveraging gradient information, CHOMP adjusts each segment of the path in a continuous manner, typically converging to a trajectory that balances collision avoidance and smooth kinematics. However, because it relies on local gradient information, CHOMP can become trapped in local minima if the environment is cluttered or if the initial trajectory is not well-chosen.

2.1.3 STOMP (Stochastic Trajectory Optimization for Motion Planning)

STOMP blends elements of gradient-based optimization with stochastic sampling, injecting random perturbations into the path at each iteration. This stochastic element helps the planner escape local minima more effectively than purely deterministic methods. As a result, STOMP can handle complex cost functions and high-degree-of-freedom robotic systems where standard gradient-based methods might struggle. However, the computational overhead can be higher, and thorough parameter turning such as noise levels and cost function weights may be necessary to achieve robust results.

2.1.4 Pilz Industrial Motion Planner

Designed with industrial applications in mind, the Pilz Industrial Motion Planner places a high priority on safety and deterministic behavior. Its trajectory generation strategies are optimized around predictable, smooth motion that adheres to specified velocity, acceleration, and jerk limit requirements that are often mandated in production environments. Because of this focus on reliability, the Pilz planner may be less flexible than research-oriented planners like OMPL, CHOMP, or STOMP. Nonetheless, it is a go-to option for industrial robotics tasks where compliance with strict performance and safety standards is critical.

In this thesis, STOMP (Stochastic Trajectory Optimization for Motion Planning) from MoveIt is selected for its ability to iteratively refine paths through stochastic optimization [3]. Additionally, MoveIt is widely recognized for manipulator control, supported by an active community and numerous developers dedicated to its continuous improvement. Owing to its inherent flexibility, STOMP can handle complex cost functions, making it particularly suitable for navigation in cluttered environments and for manipulation tasks where avoiding collisions is critical. Advanced planners like STOMP are essential for operations such as object manipulation, where the robot must account not only for a collision-free path but also for precise end-effector positioning. The manipulator arm in this thesis is custom-made, featuring a unique prismatic joint in the middle. Testing has shown that STOMP effectively handles the motion planning for this arm.

2.2 Bin Picking, Object Detection, and Pose Estimation

Research into bin picking a problem analogous to picking objects from disordered piles provides valuable insights for object interaction in rescue settings, where a robot might encounter random debris or items of interest. Studies have proposed both 2D and 3 D vision-based techniques, emphasizing robust pose estimation to enable successful grasping of unknown or partially occluded objects.

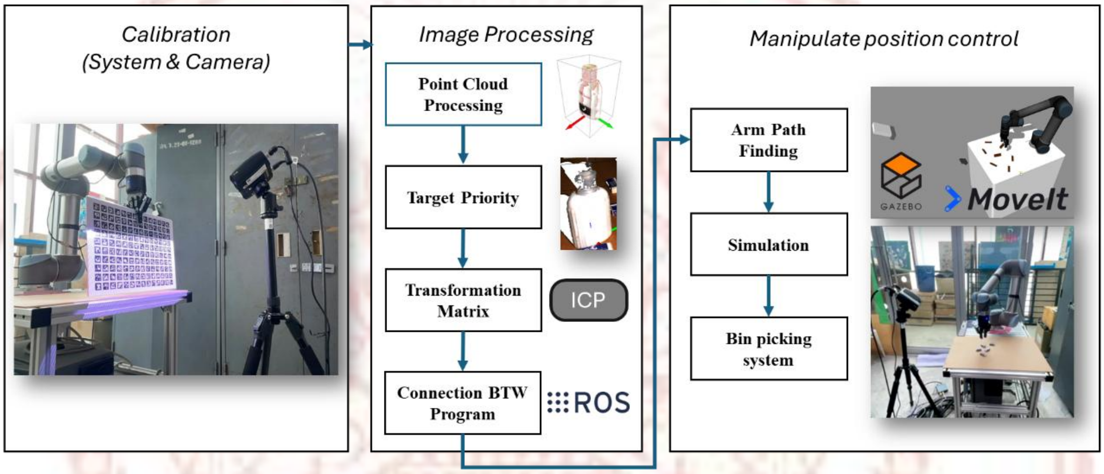

FIGURE 2-2 System Overview for Bin-Picking

In reference [4], the authors outline a complete bin picking pipeline that integrates calibration, 3D modeling, perception, trajectory planning, and simulation as in **FIGURE 2-2**. Below is a step-by-step summary of their approach, which provides insight into how stereo vision and depth sensors can be leveraged for precise object detection and manipulation in bin picking scenarios

2.2.1 Calibration (Camera and Eye-to-Hand)

The calibration process for aligning the camera and manipulator, known as eye-to-hand calibration, involves several steps to ensure accurate mapping of the camera's data to the manipulator’s coordinate system. The process begins with camera calibration, which establishes the intrinsic parameters of the camera, such as focal lengths, optical center, and lens distortion coefficients. This is achieved by capturing images of an industrial chessboard or calibration board with known dimensions from various angles and distances. The resulting data is used to compute the camera's intrinsic matrix (K), which defines how 3D points project onto the camera's 2D plane, while also correcting for lens distortions. By following this, system calibration, or extrinsic calibration, determines the transformation matrix between the camera and the robot’s base. This involves using an ArUco marker board, strategically mounted within the manipulator’s workspace. The marker is attached to the manipulator’s end-effector and moved to various predefined poses. Using the robot's forward kinematics, the position and orientation of the end-effector are calculated and aligned with the camera's observations. This alignment produces the transformation matrix, capturing the translation and rotation between the camera and the robot base.

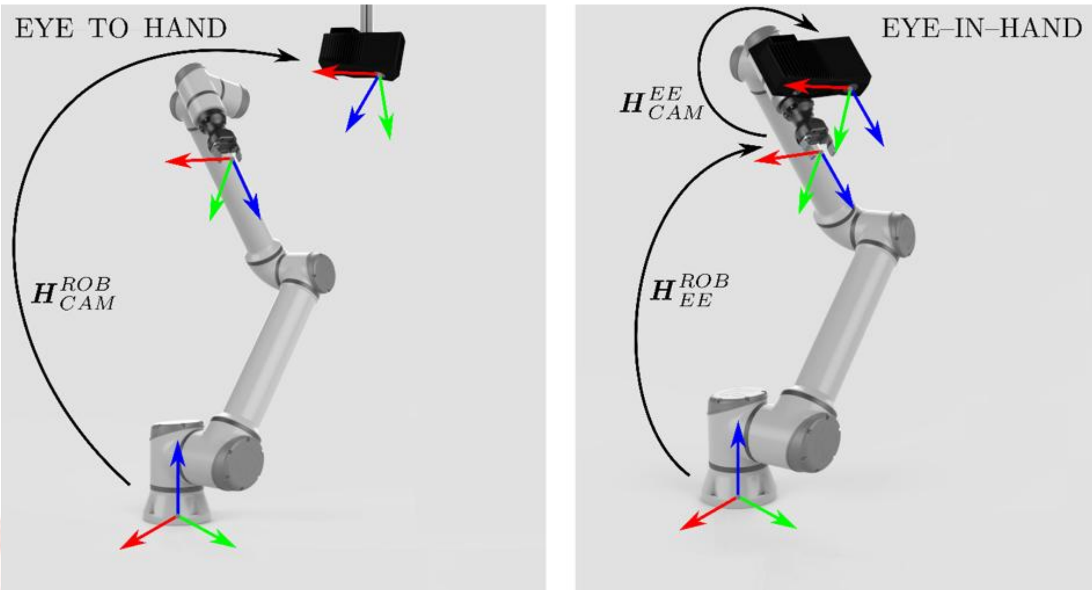

FIGURE 2-3 The Show of Eye-to-Hand and Eye-in-Hand Relative

Once the intrinsic and extrinsic parameters are established, they are combined during the eye-to-hand calibration step to create a comprehensive transformation matrix. This matrix translates object positions detected by the camera into the robot's coordinate system, ensuring spatial alignment as **FIGURE 2-3**. To validate the calibration, the system tests its ability to detect a known object in the workspace and moves the robot to the corresponding position. Any discrepancies between the detected position and the robot's actual position are adjusted iteratively to refine the accuracy. This process ensures that the manipulator can reliably interpret sensor data from the camera, enabling precise actions in tasks like object picking or interaction within the workspace.

2.2.2 3D Modeling (Reference Point Cloud for ICP)

The process of 3D modeling for creating a reference point cloud begins with generating a detailed 3D model of the target object, which is then stored as a reference for comparison during operation. In this approach, an RGB-D camera is used to capture both RGB images and depth images of the object from multiple angles as in **FIGURE 2-4**. To facilitate accurate data acquisition, the object is placed on an ArUco marker board, which provides precise positional and orientation data for each capture. The captured images are processed to create individual point clouds, which are subsequently merged into a single unified model using iterative techniques. This process involves methods like plane segmentation to remove background noise and techniques such as statistical outlier removal and voxel downsampling to enhance the quality of the point cloud.

The resulting reference model serves as a "template" for comparison with real-time sensor data acquired during system operation. When the system is running, Iterative Closest Point (ICP) algorithms are employed to align the incoming point cloud data, representing the current view of the object in the bin, with the pre-generated reference model. By iteratively refining the transformation between the two data sets, ICP ensures precise alignment even in scenarios where the target object is partially occluded or randomly oriented. This accurate and robust modeling and matching process underpins the system’s ability to reliably identify and manipulate objects in complex environments

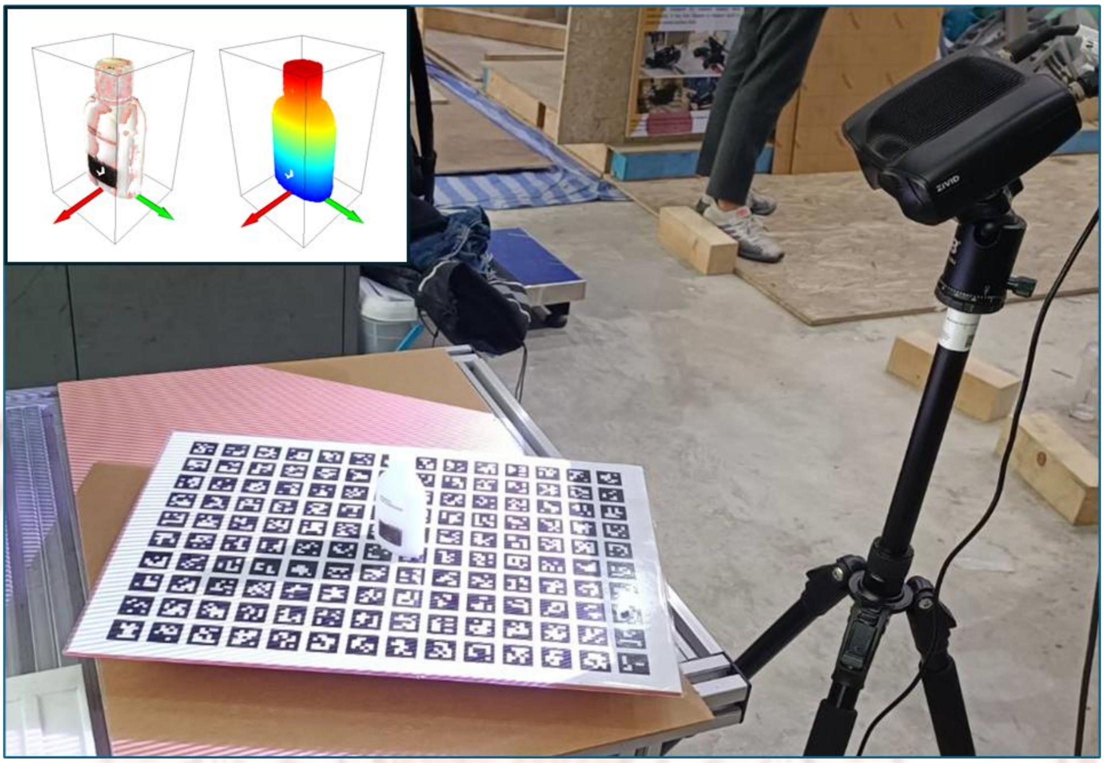

FIGURE 2-4 The 3D Modeling Process

2.2.3 Perception (Pose Estimation)

The perception process for pose estimation begins once the incoming point cloud data is aligned with the reference 3D model. Using the alignment provided by the Iterative Closest Point (ICP) algorithm, the system calculates the object's 6-DoF pose, which includes its position and orientation in space. This pose estimation is crucial for enabling the robot to determine the appropriate approach and grasping strategy for the object. The process relies on robust methods to ensure accuracy even in challenging conditions, such as cluttered or noisy environments. After aligning the object’s point cloud to the reference, techniques such as Principal Component Analysis (PCA) are used to calculate the mean and eigenvectors of the point cloud, providing a foundational understanding of the object’s orientation relative to the camera. Additionally, advanced feature-based methods like Fast Point Feature Histograms (FPFH) are applied to refine the pose estimation further, especially in cases of partial occlusion or random orientations.

To integrate this information into the manipulator's workspace, the estimated pose is transformed from the camera’s coordinate frame to the robot’s base frame using the transformation matrix derived during the calibration process. This comprehensive approach to pose estimation ensures the system can reliably and precisely identify the location and orientation of objects, facilitating successful interaction and manipulation.

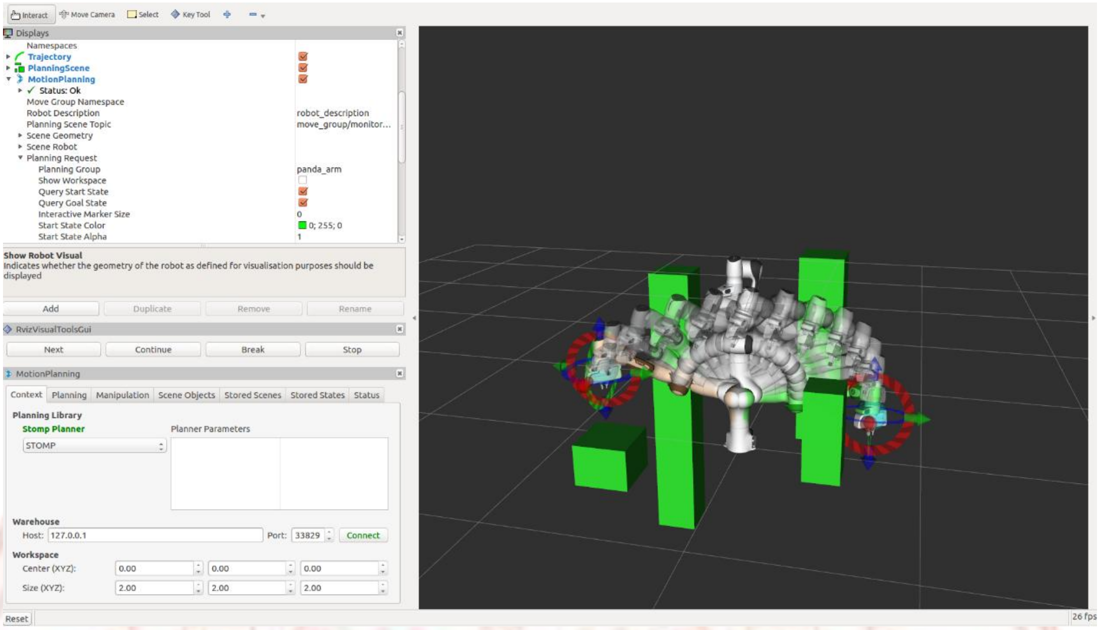

FIGURE 2-5 The MoveIt Trajectory Instance

2.2.4 Planning (MoveIt with STOMP)

The planning process for robot motion begins once the object's pose is determined, focusing on generating a collision-free trajectory for grasping the object. This process leverages the MoveIt motion planning framework in combination with the Stochastic Trajectory Optimization for Motion Planning (STOMP) planner. STOMP is a versatile tool that refines potential trajectories by introducing stochastic perturbations, enabling the planner to overcome challenges such as local minimum in the optimization landscape. In this system, the combination of MoveIt and STOMP is particularly effective for tasks like bin picking, where the manipulator must navigate through complex geometries and avoid collisions with multiple objects. By iteratively adjusting trajectories and incorporating collision avoidance constraints, STOMP ensures smooth and efficient motion planning but requires a lengthy computation process. The planner also considers additional factors like torque limits and energy optimization, ensuring that the manipulator operates within safe and efficient parameters. This robust planning mechanism allows the robot to handle cluttered environments reliably, guiding its end-effector to the object's pose with precision. The result is a well-optimized trajectory that ensures safe and effective interaction with the target object while minimizing risks of collisions as example in **FIGURE 2-5**.

2.2.5 Simulation (Gazebo)

Before executing commands on a physical robot, the entire workflow is tested in Gazebo, a robotics simulation environment. In this phase, the calibrated robot model, the 3D sensor data, the reference point cloud, and the STOMP planner all operate virtually. By simulating the pick-and-place sequence first, researchers can verify the system's performance and safety, reducing the risk of hardware damage and improving overall reliability as in **FIGURE 2-6**.

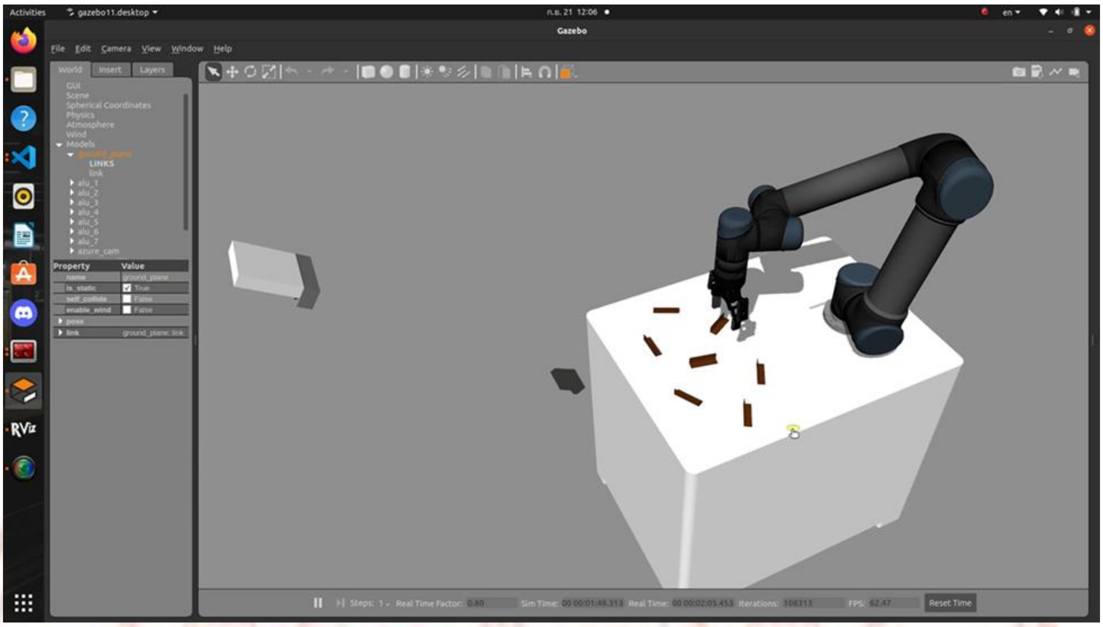

FIGURE 2-6 Gazebo Simulation for Testing

Taken together, these five components form a robust pipeline for automated bin picking, demonstrating how accurate calibration, reliable 3D modeling, effective perception, and carefully optimized motion planning can work in tandem to enable successful and efficient grasping of objects from disordered piles. This methodology translates directly into rescue robotics applications, where detecting, localizing, and safely manipulating items often under challenging conditions are key objectives. Similarly, Chen et al. [5] employed CAD-based pose estimation with multi-view image acquisition to improve success rates in cluttered environments. They employed two depth cameras to capture the 3D scene, followed by preprocessing steps such as down-sampling, noise removal, and normal estimation. The segmented data was then processed using the k-NN method to identify clusters, after which the CAD model was matched to determine a coarse pose before applying ICP for refinement. While Kuo et al. [6] and Kanso et al. [7] further explored 3D depth imaging and extract the key point of the object in the scene, CAD modeling to compare with the feature of the object, and measurement data to enhance accuracy in detecting randomly placed objects critical capabilities that also transfer directly to rescue robots attempting to locate victims or key structural elements amidst rubble.

2.3 3D Mapping, Localization, and Sensor Sensitivity

In rescue robotics, 3D mapping and localization are intertwined with 3D modeling, and both processes heavily depend on sensor quality. First, creating a 3D map of a disaster site involves gathering spatial data often through an RGB-D camera that captures depth information and color images in real time. This map not only helps the robot know where it is (localization) but also allows it to plan safe navigation paths through debris-filled or partially collapsed environments as example in **FIGURE 2-7**.

At the same time, 3D modeling builds on similar spatial data to construct a more detailed representation of individual objects or areas of interest. For instance, when a robot needs to manipulate or move specific debris, accurate 3D models provide the essential geometric information for tasks like grasp planning and collision avoidance.

However, the fidelity of these maps and models can be compromised if sensor sensitivity is poor.

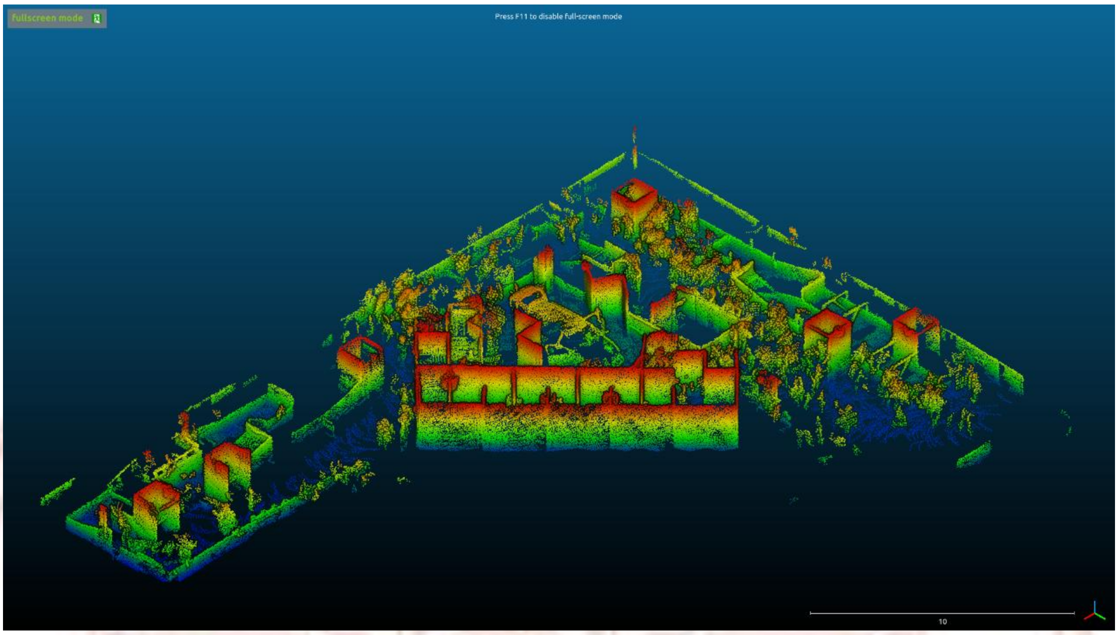

FIGURE 2-7 The 3D Map of Robocup Rescue Robot League 2023

Environmental factors such as low light, heavy dust, or partial occlusions can reduce the accuracy and reliability of depth measurements, leading to gaps or noise in the final 3D representation. If the sensor cannot consistently capture high-quality depth data, the robot risks basing its decisions on incomplete or distorted models. This can affect everything from path planning to object manipulation, increasing the likelihood of collisions or failed grasps. Robust calibration, sensor fusion (combining data from multiple sources), and algorithmic compensation (e.g., filtering or noise reduction techniques) are therefore crucial for maintaining high-quality 3D maps and models. By ensuring the sensor data is reliable, the robot can localize itself more accurately and execute complex tasks with greater success even under challenging, real-world conditions.

Accurate 3D mapping and localization are vital for rescue robots operating in dynamic, partially known disaster environments. Research has shown that low-cost RGB-D cameras can effectively generate real-time 3D maps [8][9] and enable accessible object modeling techniques, even for non-expert users. However, sensor sensitivity is affected by factors like lighting, occlusions, and dust can degrade depth data quality [14], thus reducing success rates in tasks such as bin picking or object manipulation. To overcome these challenges, robust sensor fusion and precise calibration become essential for maintaining reliable performance in uncertain conditions. Below is an overview of several common techniques used to create 3D maps or 3D models.

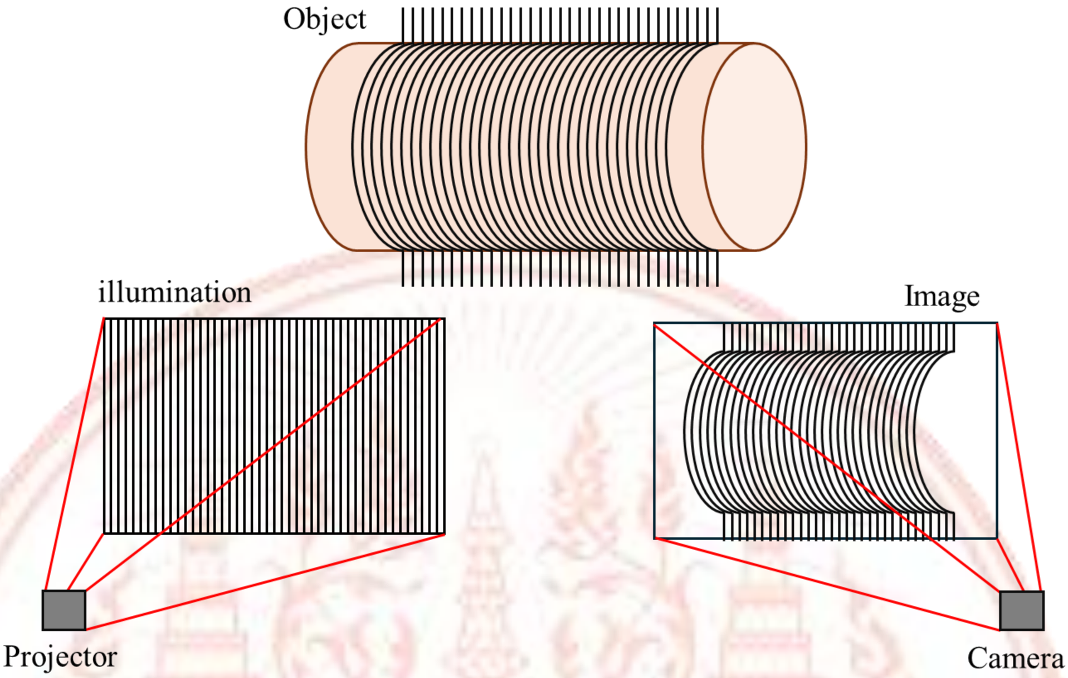

FIGURE 2-8 Structured Light Scanning Principle

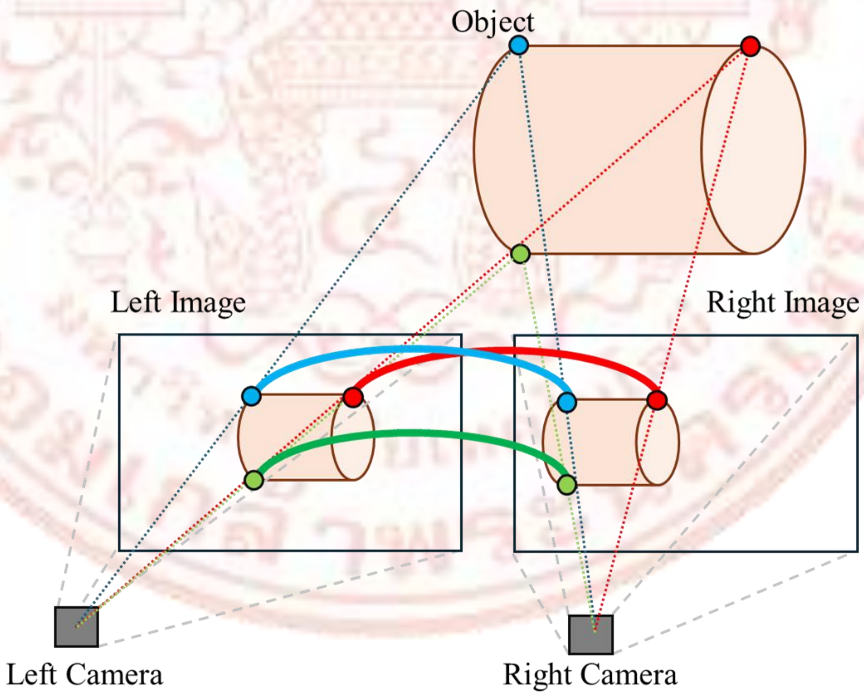

FIGURE 2-9 Stereo Vision Principle

2.3.1 Structured Light Scanning

Structured light systems project a pattern (often infrared or visible light) onto a scene. As the pattern deforms over objects, onboard cameras capture the distortion. A computer then reconstructs a 3D surface from these deformations. This method can achieve high accuracy and resolution at short to medium ranges, making it popular for tasks such as indoor scanning or precise object modeling as in **FIGURE 2-8**. However, performance may degrade in bright, outdoor environments and with reflective or transparent surfaces.

2.3.2 Stereo Vision

Stereo vision uses two (or more) cameras placed at known distances apart (a "baseline"). Each camera captures a slightly different view of the same scene. By matching corresponding points between the images, the system computes depth via triangulation. This approach closely mimics human binocular vision and can provide good depth estimates in well-lit environments as in **FIGURE 2-9**. However, it can struggle with low-contrast or texture less surfaces, and calibration between the cameras must be precise for accurate results.

2.3.3 RGB-D Sensors

RGB-D cameras combine conventional color (RGB) imaging with depth data, usually obtained via structured light or time-of-flight principles. These devices output both a color image and a corresponding depth map, enabling real-time 3D reconstructions without the need for elaborate multi-camera setups. They are widely used for robotic applications thanks to their simplicity and cost-effectiveness. However, depth accuracy can decline in bright outdoor settings or when confronted with reflective and transparent surfaces.

2.3.4 Photogrammetry

Photogrammetry reconstructs 3D models from multiple 2D images taken at different angles. Advanced algorithms (Structure-from-Motion, Multi-View Stereo) estimate camera positions and generate a dense point cloud or mesh. Photogrammetry can yield highly detailed models, especially if the images are taken carefully with well-textured scenes as in **FIGURE 2-10**. The main trade-off is computation time; reconstructing high-resolution models can be resource intensive. Lighting changes or texture less surfaces can also complicate matching between images.

2.3.5 Time-of-Flight Cameras

Time-of-Flight (ToF) cameras emit light pulses (often infrared) and measure the round-trip time of these pulses to compute depth. The principle behind this involves emitting a light pulse towards an object and then detecting the light reflected back. The time taken for the round-trip is directly proportional to the distance, as light travels at a constant speed as shown in **FIGURE 2-11**. Using the formula (2-1) the camera calculates the depth information.

$$\mathbf{d} = \left(\frac{\mathbf{c}}{\mathbf{2}}\right) \times \left(\frac{\Delta\boldsymbol{\varphi}}{\mathbf{2\pi f_{mod}}}\right) \tag{2-1}$$

Where d is the distance (m)
c is the speed of light (m/s)
φ is phase offset (rad)
f_{mod} is the modulation frequency.

This direct measurement can be faster than structured light in certain implementations, enabling robust real-time depth acquisition. While less susceptible to ambient lighting than structured light, some ToF sensors can still be affected by highly reflective or bright outdoor conditions.

2.3.6 LiDAR-Based Mapping (SLAM)

LiDAR (Light Detection and Ranging) sensors emit laser pulses and measure the time it takes for them to return after hitting an object as the technique time-of-fight. Algorithms often combined under the term SLAM (Simultaneous Localization and Mapping) use these measurements to build highly accurate 2D or 3D maps and track the sensor's position in real time. LiDAR excels in accuracy over larger distances and can work in a wide range of lighting conditions, including darkness. However, LiDAR units can be more expensive than RGB-D cameras, and complex environments with glass or highly reflective surfaces can introduce noise or partial scans.

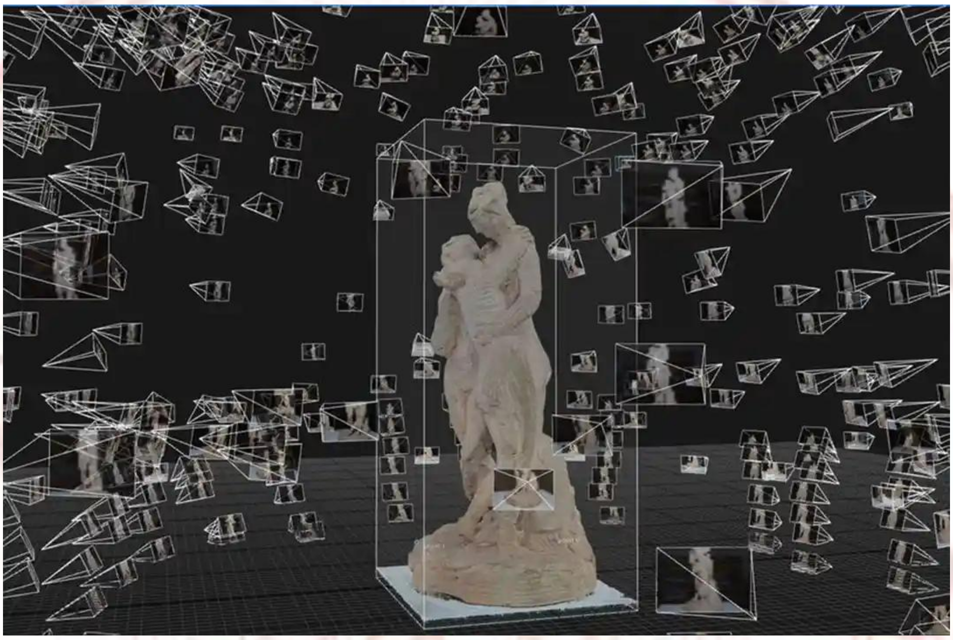

FIGURE 2-10 Example of Photogrammetry

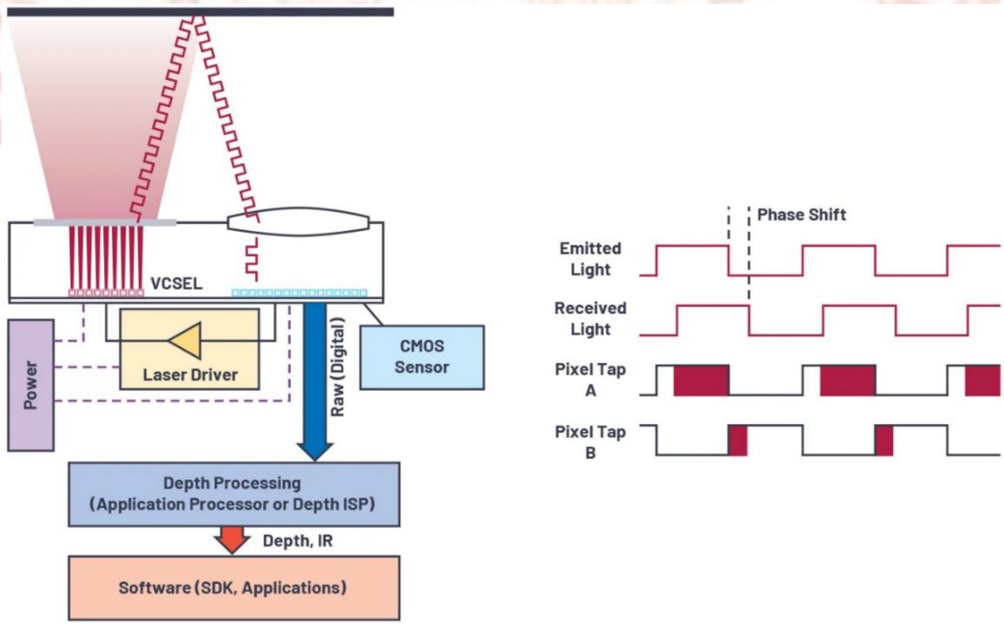

FIGURE 2-11 Overview of Continuous Wave Time of Flight Sensor Technology

2.4 ICP (Iterative Closest Point) and Alignment Algorithms

In rescue robotics and other fields requiring high precision, the Iterative Closest Point (ICP) algorithm has become a cornerstone for aligning 3D point clouds, enabling accurate object pose estimation. Its importance stems from the need to match sensor data, such as scans of an environment or object, with pre-existing models or other sensor readings. This process is critical for enabling tasks like object manipulation, navigation, or detailed mapping, especially in dynamic and unpredictable environments. The ICP algorithm works by iteratively refining the alignment between two sets of points: a source point cloud, which could be data captured from a sensor, and a target point cloud, often a pre-scanned 3D model as shown the principal in **FIGURE 2-12**. The alignment process begins with an initial transformation estimate that may be derived from methods such as feature matching or Principal Component Analysis (PCA). This initial guess, while often coarse, serves as a starting point for the algorithm to refine the pose further.

The first step in ICP involves establishing correspondences between points in the source and target clouds. For each point in the source cloud, the closest point in the target cloud is identified. This correspondence assumes that the nearest point represents the same feature or part of the object, a critical but sometimes challenging assumption when the surfaces of the two-point clouds are noisy or sparsely populated. Once correspondence is established, the algorithm computes a rigid transformation, consisting of a rotation and translation, that minimizes the distance between the matched points. This transformation is often calculated using mathematical techniques like Singular Value Decomposition (SVD) or linear least-squares optimization [10]. After calculating the transformation, it is applied to the source point cloud to bring it closer to alignment with the target. The process then repeats iteratively, refining the alignment in each step, until the change between successive iterations becomes negligible, as measured by a predefined metric such as the root mean square error (RMSE) of point distances.

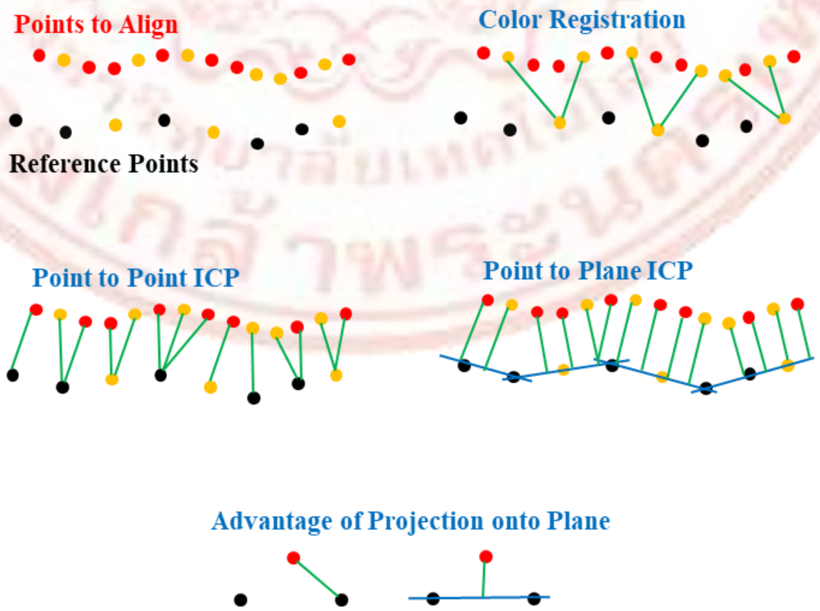

FIGURE 2-12 The Explanation of Method to Use in Iterative Closest Point

2.4.1 Point-to-Point ICP

The point-to-point ICP variant is the simplest and most computationally efficient form of the algorithm. It focuses on minimizing the Euclidean distance between corresponding points in two-point clouds: the source (e.g., the scanned data) and the target (e.g., a pre-existing model). In this method, each point in the source cloud is matched to the closest point in the target cloud, and a rigid transformation composed of rotation and translation is computed to reduce the overall distance between these correspondences. This method works well for datasets with sparse or distinct points, where the features are easily identifiable and separated. It is often used as an initial alignment step due to its computational simplicity, enabling quick rough alignment. However, its limitations become evident when applied to smooth surfaces. On such surfaces, the nearest point may not accurately represent the true correspondence, leading to misalignments. Thus, while point-to-point ICP is computationally efficient, it is best suited for datasets with clearly defined features and requires a good initial alignment to avoid errors.

2.4.2 Point-to-Plane ICP

The point-to-plane ICP variant builds on the point-to-point approach by incorporating surface normals into the alignment process. Instead of minimizing the Euclidean distance between corresponding points, it minimizes the distance between a point in the source cloud and the tangent plane at the corresponding point in the target cloud. By considering the surface geometry, point-to-plane ICP achieves more accurate and stable alignment, especially for smooth or planar surfaces. This variant is particularly effective for aligning large datasets with well-defined surface structures, such as walls, floors, or mechanical components. The inclusion of surface normals ensures that the algorithm accounts for the geometry of the object, leading to faster convergence and greater accuracy. However, the added complexity of computing and using surface normals makes this method more computationally intensive than point-to-point ICP. Additionally, it is less effective for sparse datasets or point clouds without clear surface definitions. Point-to-plane ICP is ideal for applications where planar features dominate, and precision is critical.

2.4.3 Color Registration ICP

The color registration ICP variant extends the algorithm beyond geometric alignment by incorporating color as a matching feature. In this method, point correspondences are established not only based on spatial proximity but also on similarity in RGB values as in **FIGURE 2-13**. This approach leverages additional information provided by color to refine alignments, making it particularly effective in scenarios where geometric features alone are insufficient or ambiguous. Color ICP is highly useful in environments with mixed textures or complex patterns, where color provides context that enhances the reliability of correspondences. For example, in scenes with repetitive geometric patterns, the color data can help differentiate regions that might otherwise appear identical. While this variant significantly improves alignment accuracy in such cases, it comes at the cost of increased computational demand. Additionally, the effectiveness of color ICP relies on the quality of the color data; noisy or inconsistent lighting conditions can negatively impact performance.

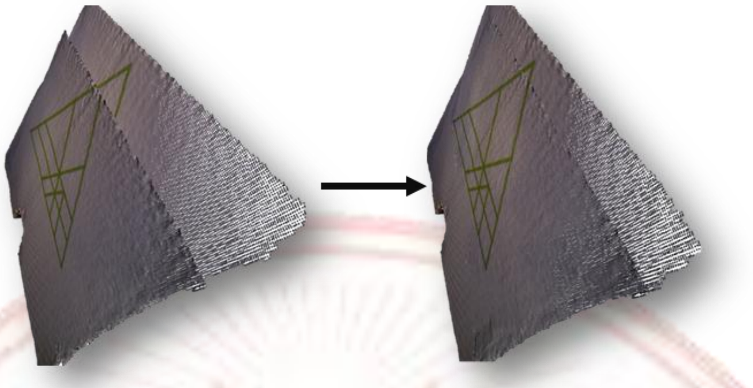

FIGURE 2-13 The Example of the Color Registration ICP

The differences between these ICP variants highlight their unique strengths and weaknesses [11]. Point-to-point ICP is computationally less intensive and ideal for scenarios where point cloud features are sparse and discrete. Point-to-plane ICP, while more computationally demanding, excels in aligning smooth or continuous surfaces by leveraging surface normals. Meanwhile, color ICP adds another dimension of data, incorporating color as a feature, which increases the computational cost but also significantly enhances accuracy in multi-modal environments. Ultimately, the success of ICP in rescue robotics depends on its ability to minimize errors during alignment, as even small inaccuracies can affect mission outcomes. By reducing alignment errors and ensuring robust pose estimation, ICP enables robots to perform critical tasks like object manipulation or navigation with precision and reliability. This capability underscores its role as a fundamental tool in environments where precise alignment can mean the difference between mission success and failure

2.5 Impact Sensing

Many rescues robotic arms must be lightweight and maneuverable, sometimes featuring flexible or prismatic joints to handle tight or cluttered spaces [15]. Research into flexible manipulator control addresses challenges including vibration control, inverse dynamics calculations, and maintaining stability under external disturbances. These factors become critical in disaster settings where collisions with debris are frequent. Additionally, impact sensing plays a pivotal role in detecting contact with objects or the environment, enabling safer and more adaptive responses. In [12] explored the time series classification of IMU data to determine impact points, which can be leveraged for immediate motion adjustments and reduced damage risks to both the robot and its surroundings.

Detecting impact in flexible manipulator systems often involves instrumenting the robot arm with inertial sensors such as accelerometers and gyroscopes collectively known as IMUs (Inertial Measurement Units) **FIGURE 2-14**. When the manipulator contacts an external object or experiences a sudden jolt (e.g., hitting debris), the IMU data displays distinctive patterns: spikes in acceleration, abrupt changes in angular velocity, or a combination of both. By classifying these time series signals, it becomes possible to precisely pinpoint when and where the impact occurred.

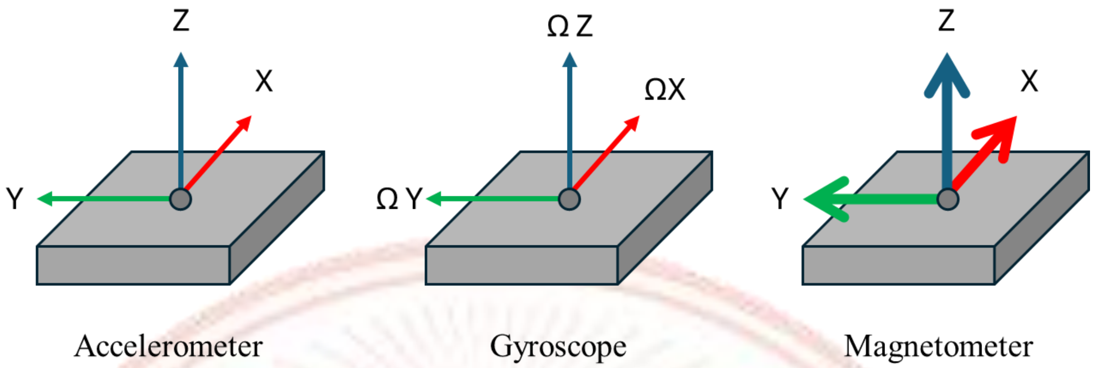

FIGURE 2-14 IMU Motion Sensor

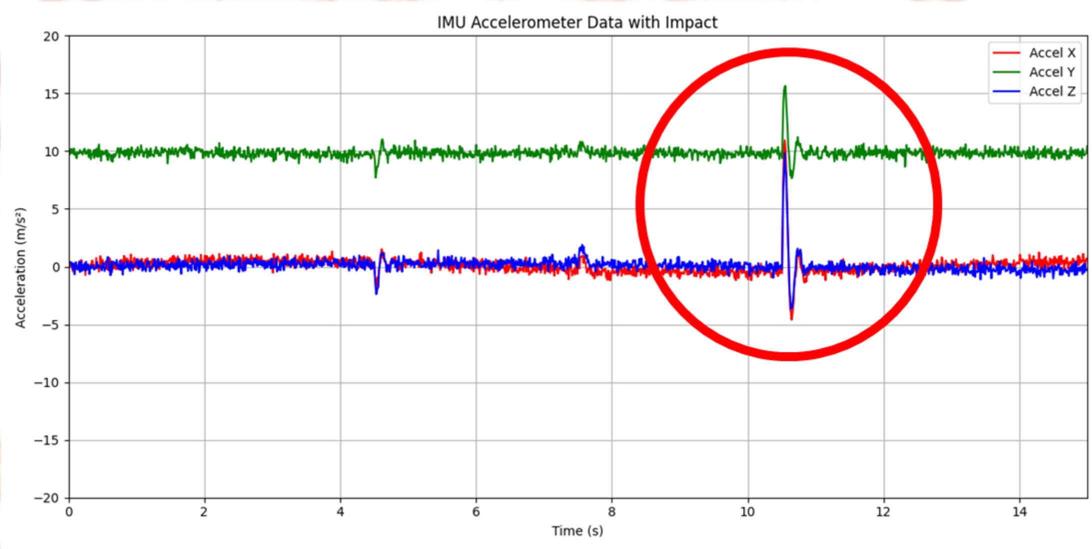

FIGURE 2-15 The Impact Detection in IMU

Once an impact is detected as see in **FIGURE 2-15**, the sensor data serves multiple purposes. First, the system can trigger immediate motion adjustments such as halting the current motion plan, retracting the arm, or switching to a compliant control mode that reduces force exerted on the environment. This responsiveness helps minimize potential damage to the robot itself, as well as to any fragile surroundings or trapped victims in rescue scenarios. Second, the recorded impact data can feed into higher-level algorithms for fault detection or anomaly recognition. These algorithms can adapt the robot's control strategy over time, improving its resilience to unstructured environments and boosting the overall safety and efficiency of the rescue operation.

CHAPTER 3
METHODOLOGY

In this thesis, it will show the procedure and the process of the object interaction system for rescue robotics which can be separate into 3 main sections. First, the calibration section will describe how to calibrate each module. Second, the pose estimation section will explain how to prepare data for the process and process. Lastly, the interactions section which to interact with the object as in **FIGURE 3-1**.

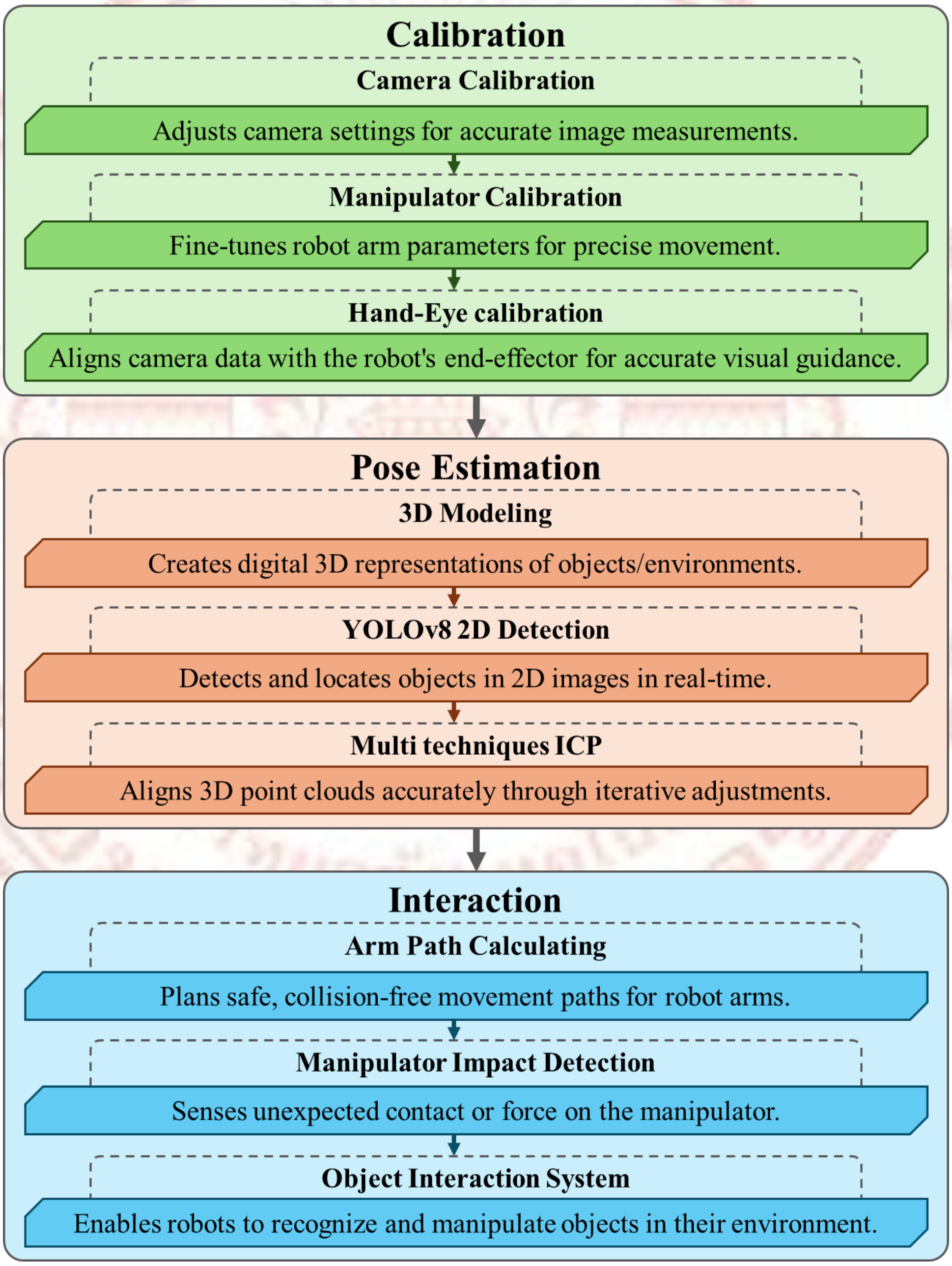

FIGURE 3-1 The Overview of The Object Interaction System

3.1 Calibration section

There are three steps of calibration to make sure that the system will operate precisely and accurately. The calibration consists of camera calibration, manipulator calibration, and hand-eye calibration. Each calibration ensures that every subsection in the system provides the appropriate tolerance.

3.1.1 The camera calibration, it's crucial to ensure precise parameter tuning as the transformation matrix is sensitive to camera settings. Below is a step-by-step expanded explanation of the calibration process:

3.1.1.1 Preparation and Setup, start by gathering all necessary equipment, including the L515 LiDAR camera and industrial-grade chessboards as **FIGURE 3-2**. Ensure the chessboard is well-lit and placed on a stable, flat surface. The camera should be securely mounted and positioned to capture the entire chessboard.

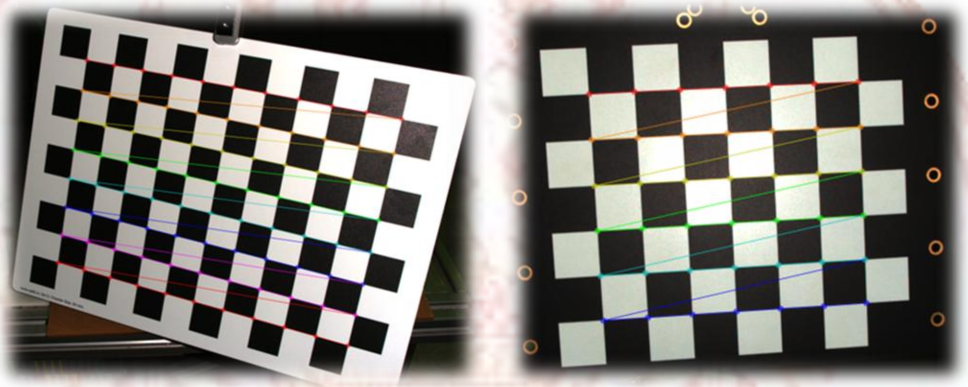

FIGURE 3-2 Zivid 7x8-30mm and calib.io 9x12-30mm

3.1.1.2 Intrinsic Calibration and Lens Distortion Correction, this step involves calibrating the camera's intrinsic parameters, which include focal lengths, optical center, and lens distortion coefficients. Using calibration software (OpenCV), capture multiple images of the chessboard from different angles and distances. Detect and extract the chessboard corners in each image. The software uses these points to compute the camera matrix Equation (3-1), where f_x, f_y are the focal lengths, and c_x, c_y are the optical centers.

$$\boldsymbol{K} = \begin{bmatrix} \boldsymbol{f_x} & \boldsymbol{0} & \boldsymbol{c_x} \\ \boldsymbol{0} & \boldsymbol{f_y} & \boldsymbol{c_y} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{1} \end{bmatrix} \tag{3-1}$$

Where f_x is focal lengths in x axis
f_y is focal lengths in y axis
c_x is the optical center coordinates in x axis
c_y is the optical center coordinates in y axis

Which will be used in transmission data between 3D point into the camera's coordinate system onto a 2D image plane or from 2D image plane into 3D point as the Equation 3-2)

$$\boldsymbol{s} \cdot \begin{bmatrix} \boldsymbol{u} \\ \boldsymbol{v} \\ \mathbf{1} \end{bmatrix} = \boldsymbol{K} \cdot \begin{bmatrix} \boldsymbol{X} \\ \boldsymbol{Y} \\ \boldsymbol{Z} \end{bmatrix} \qquad 3\text{-}2)$$

Where $\boldsymbol{s}$ is the scaling factor
$\boldsymbol{u}$ is horizontal pixel coordinate in the image
$\boldsymbol{v}$ is vertical pixel coordinate in the image
$\boldsymbol{K}$ is the intrinsic parameters
$\boldsymbol{X}$ is x axis in the 3D coordinates in the camera's frame (m)
$\boldsymbol{Y}$ is y axis in the 3D coordinates in the camera's frame (m)
$\boldsymbol{Z}$ is z axis in the 3D coordinates in the camera's frame (m)

Next, lens distortion correction, Analyze the captured images for radial and tangential distortion. These distortions often manifest as barrel or pincushion effects, where straight lines appear curved. The software calculates distortion coefficients (OpenCV), which are used to undistort the images, ensuring that measurements are accurate and free from distortion.

3.1.1.3 Post-Calibration Testing, conduct a series of tests to ensure the calibration process is effective. This includes capturing real-world scenes and analyzing the accuracy of point cloud data and object positioning. If inaccuracies persist, refine the calibration by increasing the number and diversity of chessboard images or adjusting the calibration setup.

3.1.2 The manipulator calibration process, which ensures accurate end-effector positioning and orientation, several critical steps are followed:

3.1.2.1 Preparation and Setup, begin by preparing the necessary tools, including a protractor and gyroscope sensor, which are essential for measuring joint angles accurately. Ensure that the manipulator is in its home position and all joints are accessible for measurement and adjustments. Confirm that both incremental and absolute encoders are functional to capture accurate feedback.

3.1.2.2 Understanding and Establishing DH Parameters, The Denavit-Hartenberg (DH) parameters, which define the geometry of the manipulator, are established. These parameters include link lengths, link twists, joint angles, and offsets, forming the foundation for understanding joint movements. A baseline is created by comparing the theoretical DH parameters with the physical design of the manipulator to identify any initial discrepancies due to manufacturing or assembly tolerances. In this thesis, the Modified Denavit-Hartenberg (MDH) method is utilized. There are two widely accepted techniques for representing robotic kinematics: the Classical Denavit-Hartenberg (DH) method and the Modified Denavit-Hartenberg (MDH) method. While both methods serve the same purpose describing the spatial relationship between consecutive joints of a robotic manipulator their mathematical formulation differs.

3.1.2.3 Sensor Feedback Analysis, Analyze feedback from the incremental and absolute encoders. Incremental encoders provide relative motion data, while absolute encoders offer the exact position of each joint. Identify shifts in feedback data that may result from factors such as gear backlash or slippage, which can introduce inaccuracies in joint positioning. This also includes gear ratio compensation which will adjust the encoder feedback to account for the manipulator's gear ratio. This step ensures that the actual joint motion is accurately reflected in the encoder readings.

3.1.2.4 Angle Measurement and Adjustment, Use the protractor to manually measure the angle of each joint. This provides a reference to validate the encoder readings. Complement manual measurements with data from the gyroscope sensor, which captures angular velocities and orientations with high precision. Compare the encoder feedback with the measured values from the protractor and gyroscope to detect and quantify any deviations. Based on the observed discrepancies, adjust the encoder readings by applying compensation values. This involves fine-tuning the manipulator's control algorithms to correct for misalignments or errors in joint motion.

3.1.2.5 Validation and Testing, perform static validation by moving the manipulator to predefined positions and re-measuring joint angles to ensure that the calibrated encoder readings align with the actual positions. Conduct dynamic tests, where the manipulator executes various tasks to verify that the end-effector reaches target positions and orientations accurately under operational conditions.

3.1.3 The manipulator calibration process, which aligns the RGBD camera (L515) with the manipulator's end-effector, the following steps are performed:

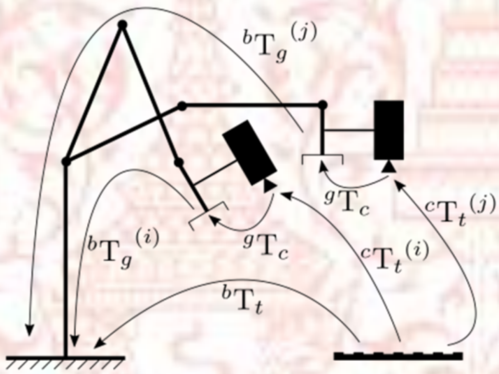

FIGURE 3-3 The Relation of Each Coordinate of The Manipulator

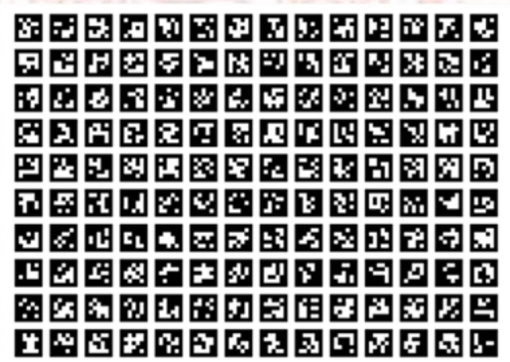

FIGURE 3-4 ArUco 10x14-21.5mm, DICT_5X5_250

3.1.3.1 Preparation and Setup, start by securely mounting the RGBD camera on the manipulator's gripper. This configuration ensures the camera moves with the end-effector during operation. Use a high-precision ArUco marker board as **FIGURE 3-4** for calibration. Position the marker board in a well-lit and stable environment within the camera's field of view. Additionally, the reason for using a 10**x**14 grid ArUco marker is to increase the precision of the coordinates that are provided by the board since the noise from the image processing is eliminated using the average of a greater number of ArUco.

3.1.3.2 Capturing Manipulator Poses and Image Processing with ArUco Markers, Moving the manipulator to various positions and orientations, ensuring the camera captures the ArUco marker from multiple angles. For each pose, record the transformation matrix that describes the relationship between the end-effector and the manipulator's base. This matrix is derived using forward kinematics. At the same time, detect the ArUco markers in the camera's image. These markers provide precise coordinates of the board relative to the camera's frame. Extract the position and orientation of the ArUco marker using image processing techniques. This information serves as a reference for aligning the camera's frame with the manipulator.

3.1.3.3 Calculating the Hand-Eye Transformation, Combine the recorded manipulator poses and the camera's marker detections to compute the hand-eye transformation matrix. This matrix represents the fixed link between the end-effector and the camera. It translates and rotates the camera's coordinate system into alignment with the manipulator's end-effector frame. Then use forward kinematics to ensure the calculated hand-eye transformation is consistent with the manipulator's movement. Validate that the transformation matrix remains stable across all recorded poses, ensuring precise alignment regardless of the manipulator's orientation as in **FIGURE 3-3**.

3.1.3.4 Validation and Testing, Test the calibration by using the camera to detect the ArUco marker from various positions and verifying that the calculated marker coordinates match the known ground truth. Simulate operations where the camera and manipulator work together, ensuring that the end-effector reaches target positions accurately when guided by the camera.

Calibration is a foundational process in robotic systems, ensuring precision, accuracy, and seamless integration between visual and mechanical components. Camera calibration corrects intrinsic parameters, such as focal lengths and lens distortions, to produce distortion-free depth and RGB data, which are critical for accurate object recognition and pose estimation. Without this step, errors in the transformation matrix could compromise downstream tasks. Manipulator calibration refines joint positions and end-effector orientations by compensating for feedback discrepancies, such as gear backlash or encoder inaccuracies, using Denavit-Hartenberg parameters. This ensures precise motion control and alignment with intended trajectories, which is essential for reliable task execution. Hand-eye calibration bridges the gap between visual perception and mechanical action by establishing a fixed transformation between the camera and the manipulator's end-effector. This alignment ensures that detected objects correspond to actionable coordinates, enabling accurate interactions like grasping and placement. Together, these calibrations enhance the system's accuracy, repeatability, and robustness, enabling effective operation in dynamic environments and making calibration a critical determinant of overall system performance.

3.2 Pose Estimation section

The pose estimation process aims to determine the position and orientation of the target object [6][7]. The process consists of multiple steps beginning with gathering the reference 3D model and a labelled 2D dataset to train the YOLOv8 2D Detection, then using the result of the detection to create the target object point cloud. Afterward, we use multi-technique Iterative Closest Point (ICP) with the reference 3D model to get pose reference from camera. The transformation of the object is calculated back to the manipulator base link which is the root of the arm reference.

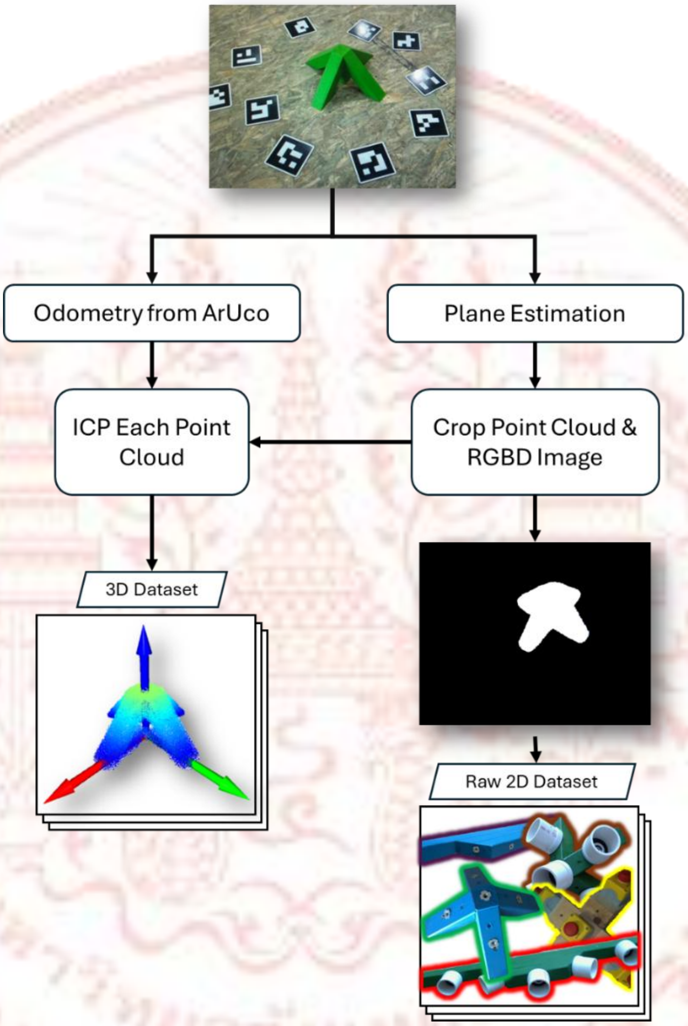

FIGURE 3-5 Diagram of 2D and 3D Dataset Collection

3.2.1 3D Modeling

The 3D modeling process begins with capturing RGB and depth images of the target object using the Intel RealSense L515 camera. Multi-ArUco markers are strategically placed in the scene to serve as spatial references, facilitating the generation of accurate odometry for the camera [8]. This odometry is crucial in determining the camera's position and orientation relative to the object during the scanning process. By combining the RGB data with depth information, the system creates a comprehensive spatial representation of the object as shown in **FIGURE 3-5**.

Once the images are captured, they are converted into point cloud data. This transformation involves mapping each pixel in the RGB image to a 3D point in space [9], utilizing depth information and the camera's intrinsic parameters. The ArUco markers enhance precision during this process by providing reliable reference points.

The resulting point cloud represents the object and its surroundings, forming the raw data for further processing.

To construct a cohesive 3D model, point clouds from multiple viewpoints are mapped and aligned. This alignment is achieved using the Iterative Closest Point (ICP) algorithm, which minimizes discrepancies between overlapping point clouds by iteratively refining their alignment. The camera's odometry data ensures an accurate initial alignment, and the ICP process enhances the model's precision by reducing Root Mean Square Error (RMSE) between the point clouds. The system removes the background by calculating a reference plane using the ArUco markers. Points below this plane, representing the ground or other flat surfaces, are cropped out. This step results in a cleaner dataset, free from unnecessary information, which streamlines subsequent processes. Further refinement involves removing noise and optimizing the point cloud through statistical outlier removal and voxel downsampling. Statistical outlier removal filters out points that deviate significantly from their local neighborhoods, while voxel downsampling reduces the resolution by grouping nearby points into voxels and replacing them with representative points. These techniques improve the quality of the point cloud and reduce computational complexity.

Finally, the processed point clouds from different viewpoints are merged to create a detailed reference model of the object. This unified model, free of noise and irrelevant data, serves as the baseline for object recognition, pose estimation, and robotic manipulation tasks. Each step in this process not only enhances the fidelity of the model but also ensures robustness and efficiency, making it ideal for complex robotic applications.

3.2.2 YOLOv8 2D Detection

The process begins with the collection of a comprehensive dataset to train the AI model. RGB images are captured using the Intel RealSense L515 camera, with each image carefully labeled to provide ground truth data. This dataset serves as the foundation for training, ensuring the model can recognize and segment objects accurately. However, to enhance its robustness and adaptability, the dataset is augmented with synthetic data. This synthetic dataset is created through various image augmentation techniques, including geometric transformations, such as rotation and scaling, color adjustments to simulate different lighting conditions, the addition of noise and blurring to mimic sensor imperfections, and cropping to emphasize specific object regions. These augmentations significantly expand the dataset, exposing the model to diverse real-world scenarios as shown in **FIGURE 3-6**.

The captured and augmented images are then preprocessed using the output from the 3D modeling step. This involves mapping 3D object data into 2D space and cropping each image to focus solely on the object region. By removing irrelevant background elements and adding diverse backgrounds to the cropped images, the model’s robustness to environmental variations is further enhanced. This preprocessing ensures that the model concentrates on key object features while eliminating noise and distractions.

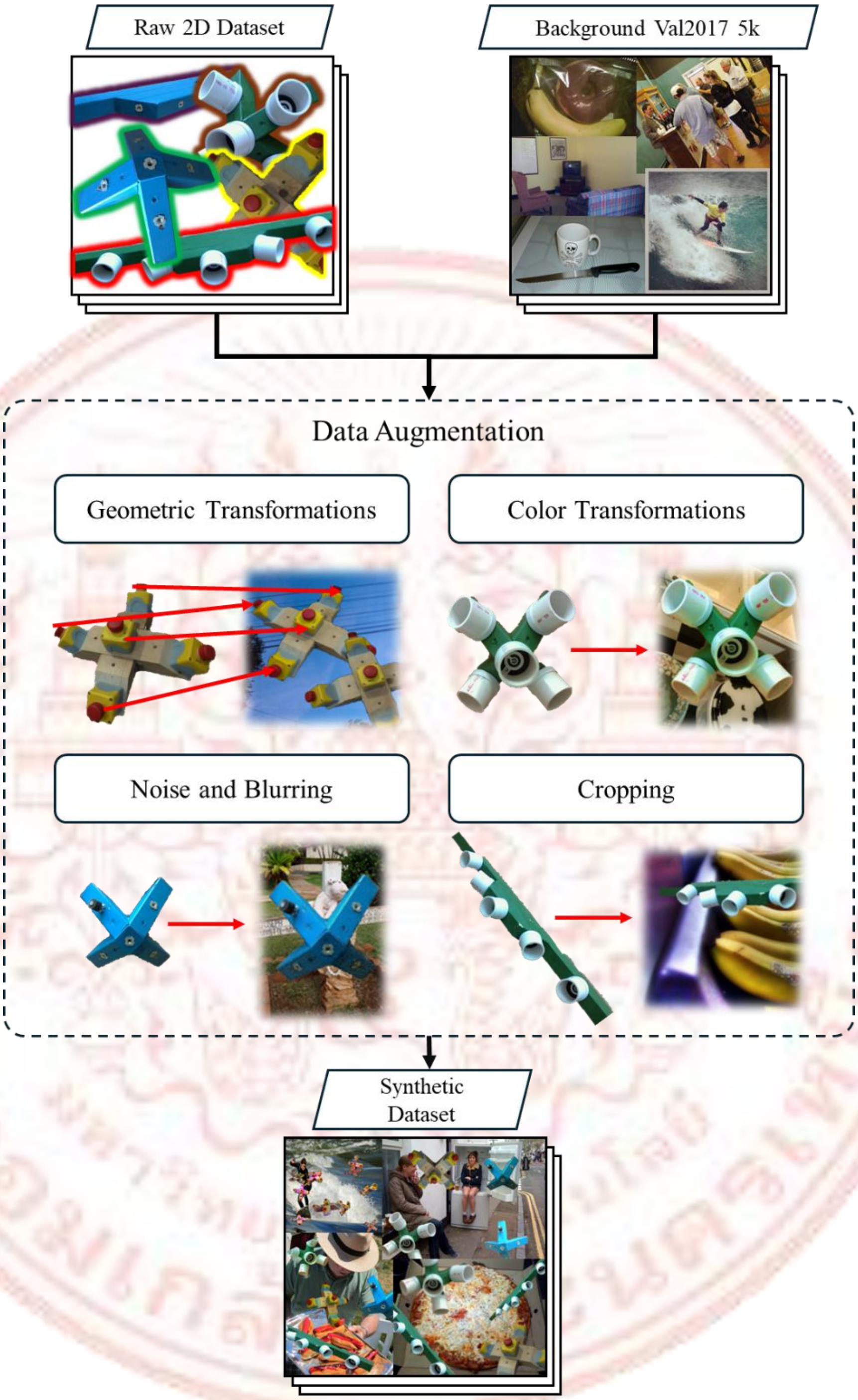

FIGURE 3-6 Diagram of Creating Synthetic Dataset

For the purposes of this study, the focus is narrowed to five key object classes relevant to dexterity tasks in a competition setting: Linear Inspection, Linear Insert, Omni Inspection, Omni Insert, and Omni Emergency Button. Each class is represented by an extensive set of augmented and labeled images. While the competition typically involves up to 50 object classes, this focused approach ensures the system's high accuracy and reliability for these critical tasks.

Once the YOLOv8 model is trained, it is deployed to detect and segment objects in real-time. The model identifies the target object in a given scene and isolates it from the background, providing a bounding box or mask for the detected object. The detected region is then combined with depth information captured by the L515 camera to create a target point cloud. This point cloud represents only the task object, with all irrelevant environmental elements removed.

This entire process of detection and point cloud generation ensures a clean and accurate 3D representation of the target object, which is essential for subsequent processes like pose estimation and robotic manipulation. The combination of a robust training dataset, diverse augmentations, and precise detection mechanisms ensures the system performs reliably under varying conditions, making it well-suited for complex dexterity tasks in competitive and real-world scenarios.

3.2.3 Multi-Technique Iterative Closest Point

To find the position and orientation of the target object to the camera, we use iterative closest point to align the relative of the target point cloud with the reference point cloud which assigned into the camera frame. In the process of iterative closest point, it has several types of process but in this system use multiple type together including point-to-point ICP for quick initial alignment, point-to-plane ICP for stable surface matching, and colored point cloud registration ICP [10][11] for color-based refinement. Since each technique uses different features of the point cloud to reach the lowest RMSE, combine each type by using the process respectively and select the lower RMSE. Furthermore, the parameter of the ICP process can affect the performance potentially causing the result to get stuck in local minimum. To avoid this problem, the multi-scale ICP takes place to calculate multi scale by using different voxel downsampling sizes to extract the features from different scale perspectives with each ICP technique type.

3.3 Interaction section

The interaction system begins with obtaining the object's dimensions, position, and orientation derived from prior processing stages. These parameters are essential for planning and controlling the manipulator arm’s trajectory to achieve specific tasks. For example, the system can handle tasks like inspecting symbols inside five tubes or pressing emergency buttons positioned at varying angles as in **FIGURE 3-7**.

Path Planning and Kinematics The TRAC-IK kinematic solver is employed to calculate the inverse kinematics of the manipulator arm, providing the precise joint positions needed to execute movements. To ensure safety and efficiency, the TRAC-IK solver is integrated with the MoveIt Stomp planner, which generates collision-free trajectories. For instance, when inspecting a tube, the arm's trajectory is planned to achieve an optimal end-effector angle for viewing symbols while avoiding obstructions. Similarly, when pressing emergency buttons, predefined "via points" help guide the gripper to align correctly with buttons placed at various angles.

Impact Detection The manipulator arm incorporates an IMU sensor integrated with an RGB-D camera (L515) mounted on the gripper. The IMU monitors XYZ-axis accelerations to detect sudden changes, signaling impacts during tasks [12]. For instance, when the gripper contacts an emergency button, the impact is confirmed by a noticeable acceleration change, ensuring task completion is verified in real-time.

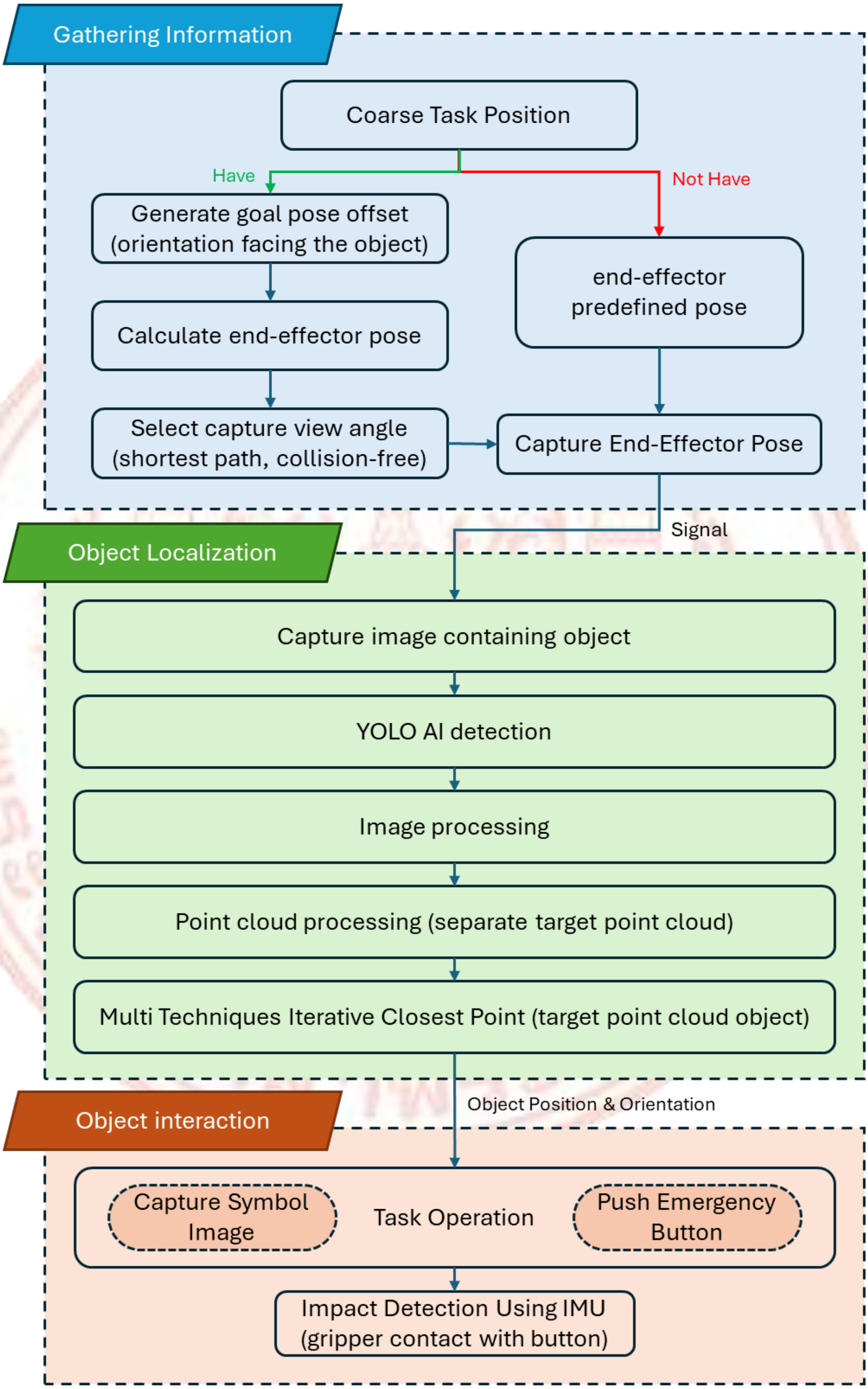

FIGURE 3-7 Diagram of Object Interaction

3.3.1 Gathering Information

Process Pipeline for Object Interaction, the pipeline begins with capturing the object's image. Two strategies are employed, Predefined Position Scanning: The arm moves to a preset position and scans for the object, Coarse Localization: Utilizing the robot's odometry and the surroundings, the arm determines a rough object’s location and orientation. In scenarios like the RoboCup Rescue competition, task zones (e.g., control panels) are predefined, enabling the robot to approach these zones autonomously.

Once in the task zone, image processing refines the object’s reference position relative to the robot. The optimal end-effector pose is calculated to capture the object within the image frame [13]. For example, the pose is adjusted to ensure the shortest, collision-free path for capturing an image of a control panel button.

3.3.2 Object Localization

By captured images undergo advanced processing using the YOLO AI model for object detection, segmentation, and localization. Once identified, the system generates a refined point cloud of the scene to isolate the target object from its surroundings. A combination of Iterative Closest Point (ICP) techniques point-to-point ICP, point-to-plane ICP, and color registration ensures precise alignment of the object’s point cloud to the manipulator’s base link. This multi-technique approach minimizes Root Mean Square Error (RMSE) during alignment and ensures accuracy.

3.3.3 Object Interaction

When interacting with objects, the system was evaluated using two main types of tasks inspection and emergency button pressing.

3.3.3.1 Inspection Tasks: The robotic arm aligns its position to accurately capture visual details, such as symbols inside tubes. Multi-angle captures ensure comprehensive inspection.

3.3.3.2 Emergency Button Tasks: The arm aligns and presses buttons while the arm verifies successful interaction using real-time impact detection. Sensors monitor contact force to confirm task completion.

The diagram in **FIGURE 3-8** illustrates an integrated robotic system leveraging TRAC-IK for kinematic solving, STOMP for planning, and real-time velocity controllers within MoveIt to achieve high-precision operations in a variety of scenarios. Starting from the top of the process, the system identifies an end-effector goal by capturing a 3D view of the environment, which involves detailed scene reconstruction with point clouds. This enables accurate identification of object positions and potential paths. The TRAC-IK solver efficiently computes inverse kinematics under constraints, such as avoiding joint limits or obstacles, while STOMP (Stochastic Trajectory Optimization for Motion Planning) refines the trajectory for smooth execution. This is crucial in applications like symbol inspection or button pressing, where even slight deviations could lead to errors. By relying on a velocity controller, the system ensures the robotic arm executes the planned motions smoothly and accurately. Incorporating scene visualization and real-time impact monitoring enhances the adaptability of the system. The hardware interface communicates the refined trajectories to low-level controllers, which interact with the actuators to perform the tasks. The hardware, including robotic arms and manipulators, is capable of operating in constrained and complex terrains, as highlighted in the RoboCupRescue competition trials.

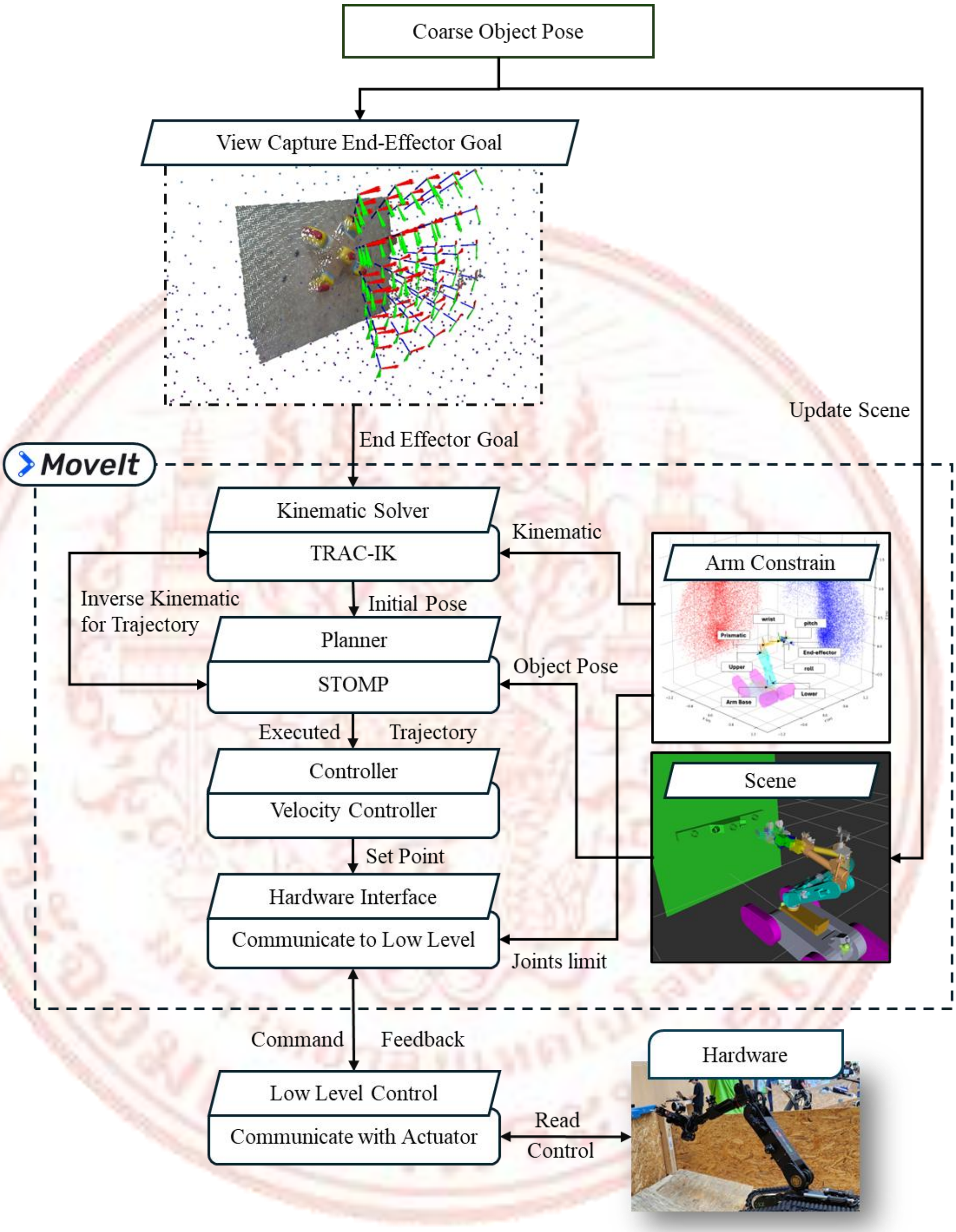

FIGURE 3-8 Hardware Communication System Process

CHAPTER 4
EXPERIMENTAL RESULTS

In this chapter, we describe the experimental methodology and present the resulting data, together with an evaluation of the proposed method's performance and accuracy. The chapter is divided into the following sections.

4.1 Experimental Setup and Competition Framework
4.2 Tasks and Scoring
4.3 Performance Evaluation and Observations

FIGURE 4-1 Bird-Eye-View of The Robocup Rescue 2024 Area in Netherlands

4.1 Experimental Setup and Competition Framework

The RoboCup Rescue 2024 League, which took place in Eindhoven, Netherlands **FIGURE 4-1**, was a prominent platform for evaluating state-of-the-art robotic systems under simulated disaster conditions. This annual competition drew teams from across the globe, representing a convergence of academic and industry leaders in robotics. The league had been established over two decades ago, inspired by the need for robotic systems capable of supporting emergency responders during hazardous missions. The event was held in an advanced arena specifically designed to emulate real-world urban disaster scenarios, allowing researchers and practitioners alike to rigorously test their robots against a wide range of challenges.

Over 25 teams from diverse countries participated in the competition, as shown in TABLE 4-1, showcasing robotic systems that ranged from compact models to advanced manipulators weighing over 70 kg.

TABLE 4-1 Qualified Teams for RoboCup Rescue 2024 League

No.	Country	Team Name	Organization
1	Austria	Team Dynamics	University of Applied Sciences Upper
2	Bangladesh	BRACU ALTER	BRAC University
3	China	Creative Town	Hefei Youth Science and Technology
4	China	NuBot	National University of Defense Technology
5	China	RERA	Beijing Information Science & Technology
6	COTE D'IVOIRE	Alp Robotics	Alpha Space Robotics
7	France	RMS	ISTY: Institut des Sciences et Techniques
8	Germany	ALeRT	MASCOR - FH Aachen
9	Germany	AutonOHM	Technische Hochschule Nürnberg
10	Germany	CJT-Robotics	Christoph-Jacob-Treu Gymnasium
11	Germany	Hector Darmstadt	Technical University of Darmstadt
12	Japan	NITRo	Nagoya Institute of Technology
13	Japan	Quix	Tohoku University
14	Japan	SHINOBI	Kyoto Univ. and OIT
15	Mexico	Ghost Robots	Tecnológico de Monterrey
16	Mexico	Robotec	Instituto Tecnológico y de Estudios
17	Mexico	UP Robotics	Universidad Panamericana
18	South Korea	ROBIT	Kwangwoon UNV(Seoul)
19	Switzerland	Solidus	Technical School of Applied Sciences
20	Thailand	BART LAB Rescue Robotics	Mahidol University
21	Thailand	iRAP_ROBOT	King Mongkut's University of Technology North Bangkok
22	Turkey	ITU RAKE	Istanbul Technical University
23	USA	ATR Kent	Kent State University
24	USA	BSM Robotics	Benilde-St. Margaret's
25	USA	RatSBU	Robotics at Stony Brook University

In addition to the field trials at the competition, extensive laboratory experiments were carried out, enabling teams to refine their systems in a controlled environment. Robotic platforms were methodically tested for precision, efficiency, and resilience, crucial factors in task missions. Once validated under lab conditions, competition conditions, these technologies were subsequently deployed in real-world experiments, where they operated in real building collapse sites and were put to use in authentic rescue operations. This multi-layered approach compassed rigorous lab testing, structured field trials, and on the ground real-world evaluations ensured that the RoboCup Rescue League not only advanced research in robotics but also made tangible contributions to emergency response capabilities worldwide. In previous rescue robot competitions, the iRAP Robot relied solely on manual manipulator control using inverse kinematics. This year, however, marked a significant leap forward, as the team focused on developing a more advanced system to contend for best-in-class dexterity.

4.1.1 Competition Arena Setup

The venue includes ten concurrent test lanes configured to challenge robots in multiple operational aspects such as mobility, dexterity, and autonomous mapping. The competition prioritizes inclusivity, allowing teams to schedule their own trials within the available time slots. This approach ensures each team can rigorously test their systems under various conditions. The arena incorporates meticulously designed terrains and obstacles that simulate real-world disaster environments. These include:

4.1.1.1 Maneuvering Lanes (MAN): These include continuous ramps, crossing ramps, and K-rails, tailored to test a robot's bi-directional navigation capabilities and situational awareness. These lanes provide incremental challenges, ranging from flat terrains for basic navigation to sloped ramps requiring advanced maneuvering skills.

4.1.1.2 Mobility Obstacles (MOB): Featuring elements such as sand, gravel, stairs, and pallet hurdles, these obstacles simulate complex terrains. The difficulty is adjustable, progressing from simpler tasks to scenarios that test the robot's control systems under extreme conditions, such as debris-laden stairs.

4.1.1.3 Dexterity Zones (DEX): These tasks are designed to evaluate the precision, reach, and control of robotic manipulators. These zones incorporate tasks such as button pressing, valve turning, and key insertion, challenging the robot's ability to perform intricate operations with accuracy. This thesis will assess the performance of robotic systems in these dexterity zones, focusing on their ability to execute precise manipulations under various constraints and conditions. There are 9 dexterity tasks consist of linear-inspection, omni-inspection, linear-touch, omni-touch, linear-insert, omni-insert, close valves, emergency push button, and key-insert

4.1.1.4 Exploration and Mapping Areas (EXP): Labyrinthine mazes, elevated paths, and mapping fiducials are designed to evaluate the robot's ability to autonomously navigate and generate accurate 2D/3D maps. Autonomous behaviors in these areas are rewarded with higher scores due to their complexity and operational significance.

Each test lane is designed to provide statistically significant performance data through repetitive trials. Each trial spans 30 minutes, including 20 minutes of operation and 5 minutes each for setup and exit. Robots are scored on their ability to complete tasks, with multipliers based on autonomy levels, Teleoperated tasks earn a baseline score, Autonomous tasks are rewarded with a 4x multiplier, reflecting their operational complexity. Tasks completed under degraded communication conditions earn additional multipliers, mimicking real-world challenges like signal interference in collapsed structure

Each task is aligned with ASTM International Standards for Urban Search and Rescue Robots, ensuring the relevance of competition outcomes to real-world disaster scenarios. By encouraging teams to proctor one another, the league fosters a spirit of shared learning of testing methodologies. Teams can reset their robots during trials with a minor penalty, enabling iterative improvements during the competition. The RoboCup Rescue 2024 league is not just a competition but a global collaborative effort to advance robotics for life-saving applications. Its carefully structured framework, including diverse test maps, scoring multipliers, and inclusivity, ensures a comprehensive assessment of robotic systems while pushing the boundaries of innovation

4.1.2 Laboratory Experiment

In the laboratory experiment, the system was evaluated by performing a linear-inspection task, where the robotic manipulator autonomously initiated from a home position to detect and execute the dexterity task from various starting angles. The experiment collected several key performance metrics, including detection success rate, sub-mission completion status, processing times for 3D preprocessing and 3D ICP procedures, the operation time required for each sub-mission, and the RMSE of the ICP algorithm.

4.1.3 Field Experiment

The RoboCup Rescue Robot League served as an operational test. Since the competition integrated multiple dexterity tasks designed to simulate authentic rescue conditions but in this thesis are focused on linear-inspection, omni-inspection, and emergency push button., it provided a robust setting to evaluate the performance of the autonomous manipulator system in high-stakes scenarios. By leveraging the league’s comprehensive challenges and scoring protocols, the system’s reliability and adaptability were thoroughly assessed under conditions closely approximating real-world urban disaster responses.

4.1.4 Real-World Experiment

On March 28, 2025, an 8.2 magnitude earthquake struck central Myanmar, causing intense tremors that reached Bangkok, Thailand. One of the most severe consequences was the collapse of a 30-storey building under construction on Kamphaeng Phet Road, close to the Chatuchak Weekend Market. In the aftermath of this incident, members of the iRAP robotics club volunteered to support local rescue teams, bringing advanced 3D mapping technology to assist with site assessment. By generating comprehensive maps of the collapsed structure and its surroundings, iRAP helped rescuers and engineers evaluate structural stability and minimize the risk of additional collapses. However, although the club’s team had been researching an autonomous manipulator arm for such rescue scenarios, it was not feasible to deploy in this real-world environment. The tight confined and unpredictable conditions of the disaster site underscored the current limitations of robotic arms, which still require significant development before they can reliably operate alongside human responders in hazardous conditions. This real-world experience highlighted the gap between laboratory and field test, emphasizing the need for continued research and refinement of autonomous technologies to ensure they can effectively support future rescue efforts.

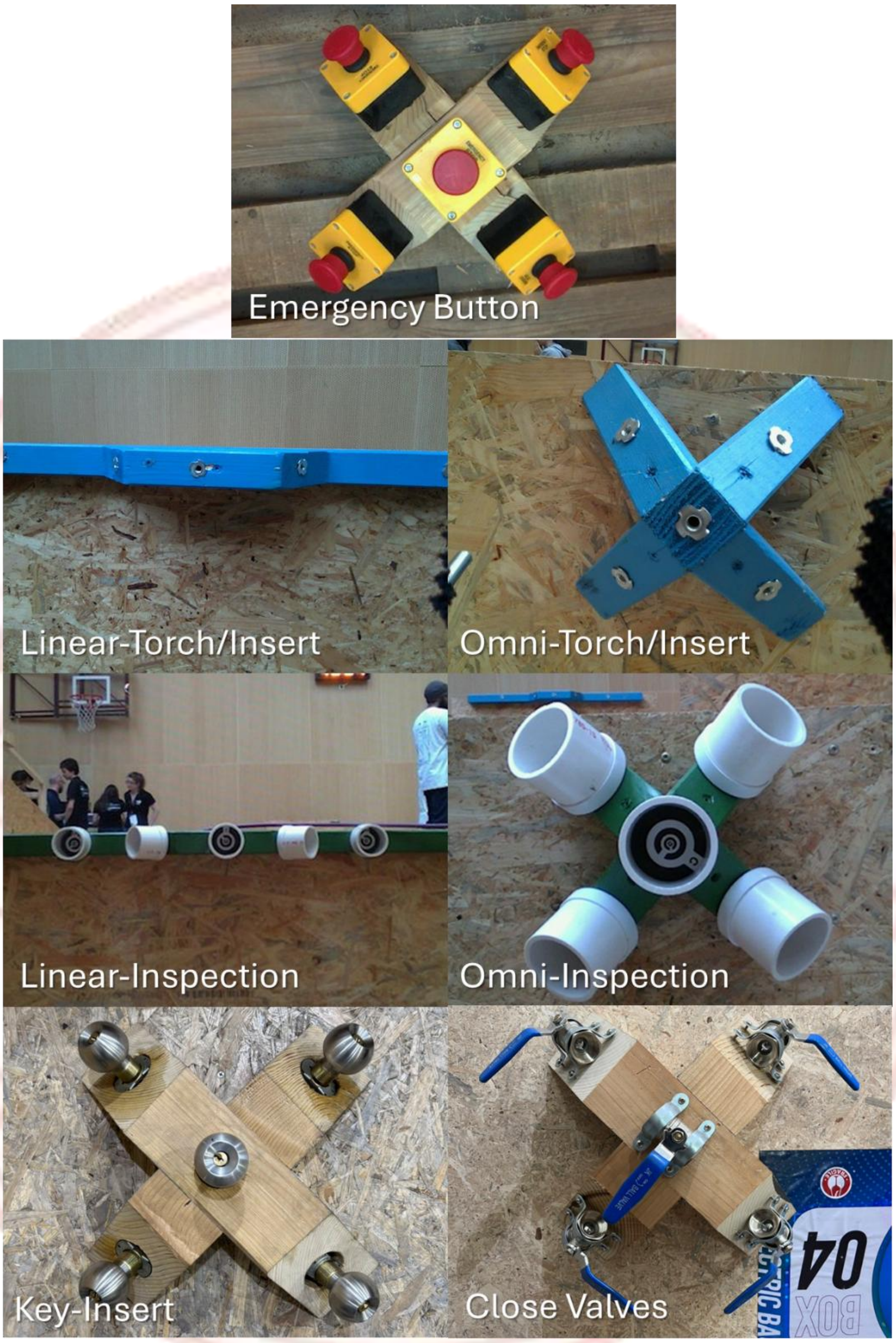

FIGURE 4-2 The Illustration of Dexterity Tasks That Operate Automatically

4.2 Tasks and Scoring

Guided by the objective of developing a system capable of efficiently detecting and interacting with objects in rescue environments, the research followed a three-tiered experimental approach. First, controlled laboratory experiments were conducted to refine the robot's core functionalities, focusing on object identification, localization, and manipulation under predictable conditions. Here, tasks such as linear inspection were systematically tested to measure detection accuracy and maneuverability of manipulator arm.

Building on insights from the lab, the second phase involved participation in a field experiment at the RoboCup Rescue competition. This environment exposed the robot to complex terrains and diverse dexterity tasks such as linear inspection, omni-inspection, and most notably emergency push-button pressing (E-stop) mirroring high-pressure conditions often encountered in real-world disasters as shown in **FIGURE 4-2**. The rigorous demands of the competition allowed for an in-depth assessment of the robot's ability to adapt to unforeseen challenges while maintaining precision in object handling.

Where Linear-Inspection: Focuses on straight-line manipulation path.
Omni-Inspection: Involves more complex, multi-angle interactions.
Push-Button tasks: Measure precision under tight tolerances

Lastly, although a real building collapse scenario occurred due to a regional earthquake, the conditions made it impractical to deploy the autonomous manipulator arm. Instead, only the 3D mapping component was utilized to support rescuers and engineers in evaluating structural risks. This experience underscored the inherent difficulties of operating in actual disaster sites and highlighted the significant development still required before an autonomous arm can be safely and effectively deployed in such volatile conditions. Consequently, no experimental data from this real-world scenario is included in this thesis.

4.2.1 Laboratory experiments

The linear-inspection task was tested with the robot positioned in front of the target, as well as at 45-degree angles to its left and right. Each of these setups was executed 50 times, recording essential metrics such as detection success rate, sub-mission completion status, processing times for both 3D preprocessing and 3D ICP, operation time for each sub-mission, and the RMSE of the ICP algorithm. Additionally, the percentage of successful operations will be calculated from the aggregate laboratory trial data to represent the overall efficiency of the system. This metric provides a clear indicator of how consistently the system can detect and manipulate objects under controlled conditions. In **FIGURE 4-3**, The laboratory experiment is depicted via GUI, providing a clear snapshot of the system in operation under controlled testing conditions.

TABLE 4-2 Process Time for Linear-Inspection Task

Linear-Inspection		Average of Process Time (sec)						
Type	n	Preprocess	ICP	sub1	sub2	sub3	sub4	sub5
Left	50	0.43	1.95	10.71	6.04	9.96	9.80	6.44
In front	50	0.43	1.88	10.61	6.12	9.80	9.69	6.53
Right	50	0.40	2.02	10.03	6.12	10.17	9.74	6.73
Total	150	0.42	1.95	10.45	6.09	9.98	9.74	6.57

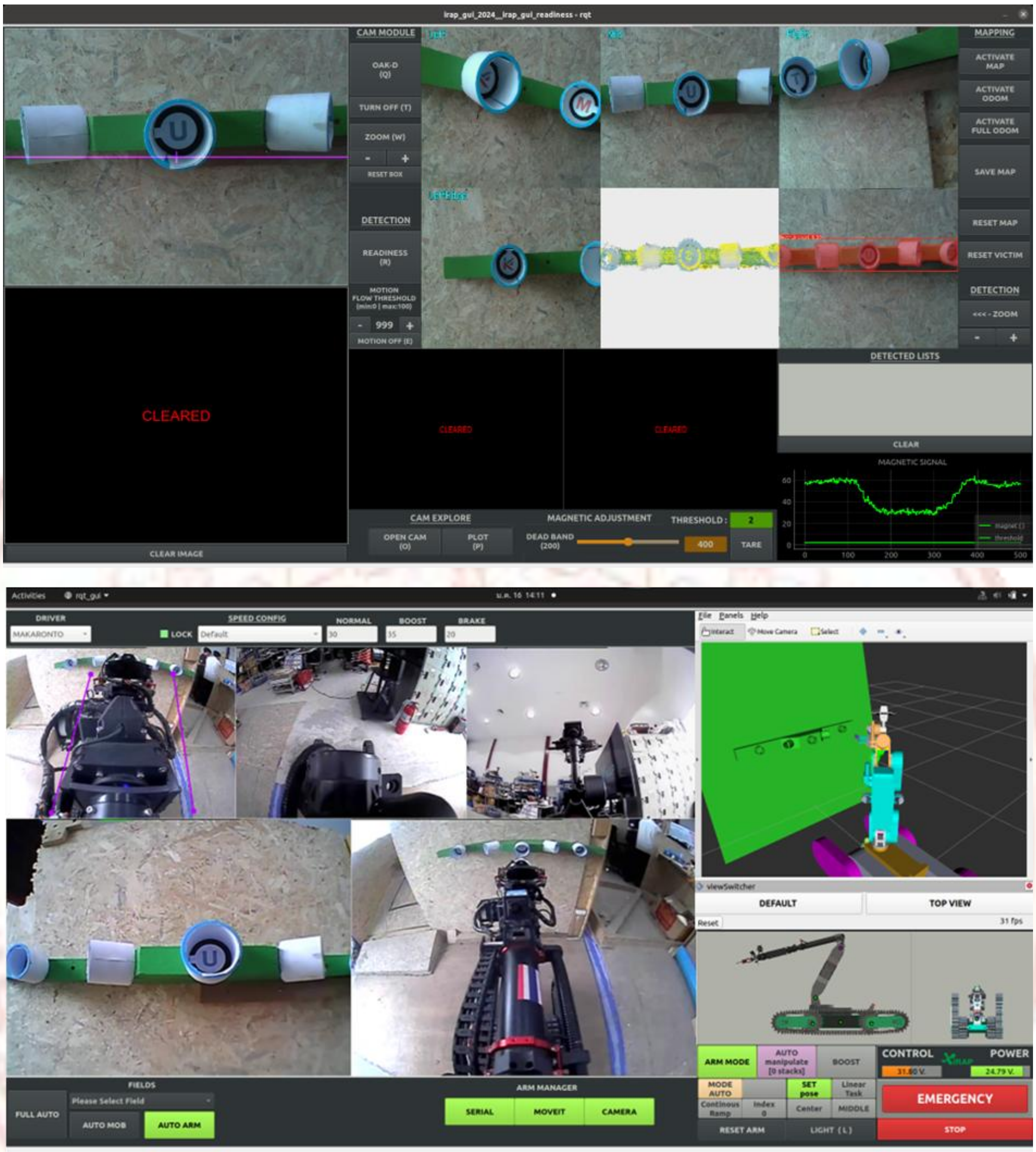

FIGURE 4-3 Graphic User Interface of The System

TABLE 4-3 RMSE of ICP Multi-Techniques

Linear-Inspection		Average of ICP RMSE			
Type	n	P2P	C2C	P2PL	C2C
Left	50	7.90E-03	7.71E-03	8.01E-03	7.70E-03
In front	50	7.85E-03	7.91E-03	7.88E-03	7.90E-03
Right	50	7.90E-03	7.71E-03	8.01E-03	7.70E-03
Total	150	7.88E-03	7.78E-03	7.97E-03	7.77E-03

The laboratory experiment has done overall 150 iterations of operating to determine the performance of the autonomous manipulator with dexterity task at the iRAP Robot club, King Mongkut's University of Technology North Bangkok as shown in TABLE 4-2, TABLE 4-3, and TABLE 4-4. From the TABLE 4-2, Preprocessing involved preparing the point cloud by constructing it from the color and depth images, reducing its resolution through downsampling, and extracting relevant features. The ICP phase utilized an iterative closest point algorithm to align the newly acquired point cloud with a reference model. Within each task, sub(n) refers to the five sub-missions that collectively define the full objective. As shown in TABLE 4-3 , three loss calculation methods Point-to-Point (P2P), Point-to-Plane (P2PL), and Color-to-Color (C2C, also called color registration) were used to evaluate ICP accuracy. The table clearly indicates both the time the system required to complete each sub-mission and the resulting ICP errors across these different loss computation approaches.

TABLE 4-4 The Statistic of The Auto Dexterity Laboratory Experiment Result

Angle	Detect			Sub-mission		
	n	Score	Percentage	n	Score	Percentage
Left	50	47	94.00%	235	224	95.32%
In front	50	49	98.00%	245	239	97.55%
Right	50	48	96.00%	240	231	96.25%
Total	150	144	96.00%	720	694	96.39%

4.2.1 Field experiment

The RoboCup Rescue League implemented a structured scoring system that assigned each task a point value reflecting its complexity and operational significance. For instance, the linear-inspection task granted 1 point per subtask, enabling participants to earn up to 5 points. By contrast, the more complex omni-inspection task was awarded 2 points per subtask, allowing for a maximum of 10 points. Lastly, the emergency push-button pressing (E-stop) presented the highest challenge, offering 10 points per subtask and thus totaling a possible 50 points. During the competition itself, only the contestants were permitted in the arena, making it difficult to capture photographs once the event began. As a result, most of the available images were taken prior to the start of the official rounds as show in FIGURE 4-4.

4.2.1.1 Score Multiplier, the scoring system incorporates multipliers based on the level of autonomy employed by the robot. Tasks completed via teleoperation are multiplied by a factor of 1. If radio communication degradation is introduced, the multiplier increases to 2, reflecting the added complexity. When the robot autonomously performs dexterity tasks after being teleoperatively guided to the task zone, the score is multiplied by 4. For fully autonomous operations, where the robot independently approaches the task and completes it without human intervention, the score receives the highest multiplier of 8.

4.2.1.2 This experiment criterion is placed on evaluating the success of the robot’s processes, which are divided into two critical components: the pose estimation process (Detect) and the object interaction process (sub-mission). These elements are integral to the robot’s ability to navigate and execute dexterity tasks effectively, ensuring reliable performance in complex environments.

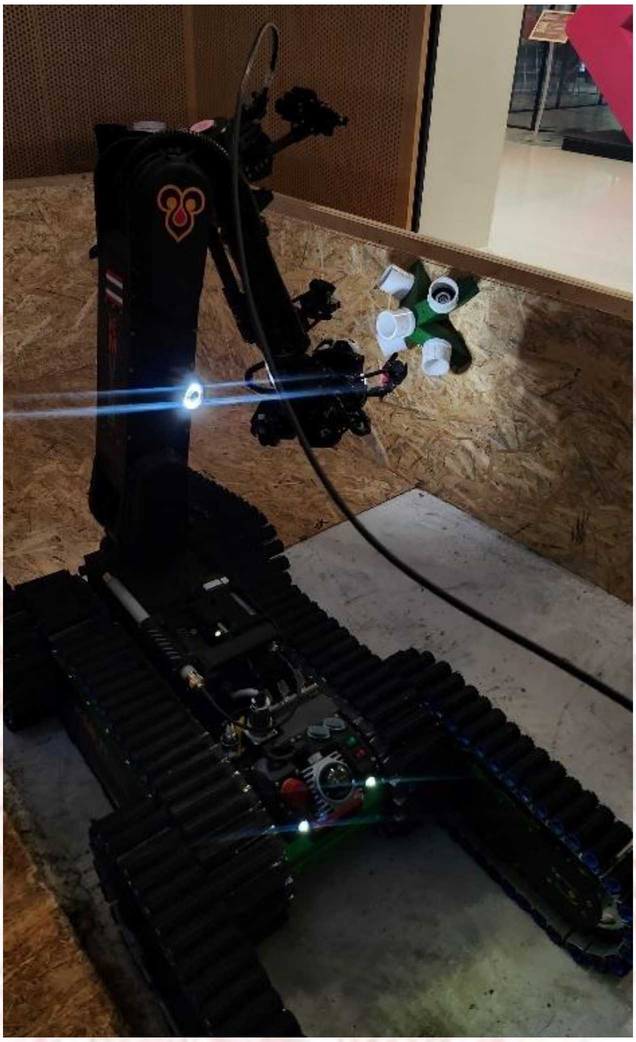

FIGURE 4-4 The Task Operation at The Arena Before Competition Begins

TABLE 4-5 Stats of Detect and Sub-mission Across Maps and Sessions

<table>
<tr><th rowspan="2">No.</th><th rowspan="2">Map</th><th colspan="2">Prelims</th><th colspan="2">Semi-Final</th><th colspan="2">Final</th></tr>
<tr><th>Detect</th><th>Sub-mission</th><th>Detect</th><th>Sub-mission</th><th>Detect</th><th>Sub-mission</th></tr>
<tr><td>1</td><td>K-Rails</td><td>2/2</td><td>9/10</td><td rowspan="3">2/2</td><td rowspan="3">8/10</td><td rowspan="3">1/2</td><td rowspan="3">4/10</td></tr>
<tr><td>2</td><td>Pallets Hurdles</td><td>1/1</td><td>3/5</td></tr>
<tr><td>3</td><td>Continuous Ramps</td><td>1/2</td><td>4/10</td></tr>
<tr><td>4</td><td>Crossing Ramps</td><td>1/2</td><td>4/10</td><td rowspan="3">2/2</td><td rowspan="3">8/10</td><td rowspan="3">2/2</td><td rowspan="3">4/10</td></tr>
<tr><td>5</td><td>Incline/Center</td><td>1/1</td><td>3/5</td></tr>
<tr><td>6</td><td>Sand & Gravel</td><td>2/2</td><td>4/10</td></tr>
<tr><td>7</td><td>Avoid Holes</td><td>1/1</td><td>4/5</td><td rowspan="4">2/2</td><td rowspan="4">8/10</td><td rowspan="4">2/2</td><td rowspan="4">7/10</td></tr>
<tr><td>8</td><td>Doors</td><td>1/1</td><td>3/5</td></tr>
<tr><td>9</td><td>Stairs</td><td>0</td><td>0</td></tr>
<tr><td>10</td><td>Labyrinth</td><td>0</td><td>0</td></tr>
</table>

From TABLE 4-5 the header consists of, No.: Represents the serial number of the rows for each map category, Map: Lists the different types of maps or terrains where the detect and sub-mission are recorded (e.g., K-Rails, Pallets Hurdles), Prelims (detect/sub-mission): Indicates the count of automated detect and sub-mission in the preliminary session for each map, Semi-Final (detect/sub-mission): Shows the count of detect and sub-mission during the semi-final session for each map, Final (detect/sub-mission): Displays the count of detect and sub-mission recorded during the final session for each map. Each column is subdivided into "detect" (count of successful poses estimation) and "sub-mission" (count of successful operate the sub-mission), providing detailed data for each session and map type. During the preliminary and semi-final rounds, the competition tasks that were able to be attempted included linear-inspection, omni-inspection, linear-touch, omni-touch, linear-insert, and omni-insert. In the final round, the range of tasks expanded to incorporate linear-inspection, omni-inspection, linear-touch, omni-touch, linear-insert, omni-insert, valve-closing, emergency push-button, and key-insertion. However, given the scope of this system, the focus remained on linear-inspection, omni-inspection, and the emergency push-button. As a result, linear-inspection and omni-inspection tasks were executed during the preliminary and semi-final rounds, while all three targeted tasks linear-inspection, omni-inspection, and emergency push-button were performed in the final.

TABLE 4-6 The Statistic of The Auto Dexterity Field Experiment Result

No.	Type	Detect			Sub-mission		
		Round	Score	Percentage	Round	Score	Percentage
1	Linear Inspect	12	11	92%	55	41	75%
2	Omni Inspect	11	10	91%	50	36	72%
3	Push E-stop	3	2	67%	10	6	60%

4.3 Performance Evaluation and Observations

The evaluation of the system's overall performance encompassed a series of staged experiments ranging from controlled laboratory setups to the high-pressure environment of the RoboCup Rescue League, followed by a deeper analysis of the collected data. Across these trials, the focus remained on assessing how effectively the robot could detect, localize, and manipulate various objects while adapting to different levels of environmental complexity. Broadly, the results revealed that while the robot demonstrated strong capabilities in recognizing and inspecting targets, the success rate and efficiency varied according to both the task's complexity and the operational context. Stable detection and precise object handling were achieved more consistently under controlled conditions, whereas unpredictability in the field setting posed additional challenges. Nonetheless, the knowledge gained from each phase contributed valuable insights into the strengths and limitations of the system's autonomous manipulator, laying a foundation for targeted improvements in future deployments.

4.3.1 Laboratory Experiment

In the laboratory setting, a multi-step process was used to both detect and interact with objects, providing insights into overall system efficiency and reliability. As indicated in TABLE 4-2, the detection phase which integrates both 3D preprocessing and the initial ICP calculation took an average of 2.37 seconds per trial. This part of the workflow involves creating the point cloud from the color and depth images, downsampling it to manage computational load, and then using ICP to match the scene's cloud to a pre-defined reference model. Once an object's position and orientation had been established, the sub-mission phase ensued, with timing that depended primarily on the trajectory generated by the MoveIt STOMP planner. This planner was responsible for producing a collision-free path for the robotic arm, thus contributing to variations in execution times across different runs. In scenarios where the arm needed to navigate around obstacles or approach the target from a challenging angle, sub-mission durations naturally increased. TABLE 4-3 further illustrates the comparative performance of ICP when augmented by color registration. Generally, applying color registration after point-to-point or point-to-plane ICP reduced the RMSE between the scene and reference model, indicating a more accurate alignment. However, in certain instances, particularly those involving inconsistent lighting or a significant discrepancy in the number of points between the reference model and the real scene, the color-based approach did not yield improved results. Such variations can arise when the controlled lighting conditions under which the reference model was created differ substantially from those in the lab environment at the time of testing, affecting how colors are perceived and, consequently, how effectively they can be used in the registration process.

Despite these occasional setbacks, the system demonstrated a robust performance overall. TABLE 4-4 shows that from the moment the robot-initiated object detection to the point at which the sub-mission concluded, the average completion time was approximately 45.2 seconds. Throughout 150 test iterations, the system maintained a detection accuracy of 96% and a sub-mission completion rate of 96.39%, underscoring both the repeatability and precision of the proposed method. These results confirm that while certain environmental factors like lighting and point cloud quantity can introduce variability, the system's foundational architecture largely succeeds in reliably detecting and interacting with objects within controlled laboratory conditions.

4.3.2 Field Experiment

Three major tasks linear-inspection, omni-Inspection, and emergency push button were assessed across multiple arenas, as summarized in TABLE 4-5 and then combined in TABLE 4-6. The data indicates that both linear and omni-inspection tasks generally achieved higher success rates than the more complex emergency push button task. Specifically, the linear-inspection maintained a 92% detection rate over 12 attempts and a 75% sub-mission completion rate across 55 sub-missions. Omni-inspection showed similarly strong performance, with a 91% detection success across 11 attempts and a 72% sub-mission completion rate out of 50 sub-missions. The emergency push button task, attempted only three times, exhibited the lowest rates, completing detection successfully 67% of the time and registering a 60% sub-mission success over 10 attempts. This discrepancy largely stems from the different degrees of complexity and precision each task requires. Inspection tasks typically involve identifying and localizing objects a process that is less constrained by the robot's sensor limitations and mechanical tolerances.

By contrast, the emergency push button task demands precise interaction and close tolerance, conditions that can be challenging for robotic arms and sensors to fulfill reliably. This challenge is further amplified by the high sensitivity of the robot's feedback mechanisms, making it difficult to maintain consistent performance when physically interacting with the environment. Additionally, it is important to note that these success percentages were influenced by the limited number of competition rounds. Because teams only had a handful of attempts to demonstrate each capability, any single error had a pronounced effect on the overall percentages. With more trial iterations, it would expect the data to stabilize, presenting a clearer assessment of true system performance. Nonetheless, the existing results underscore the system's strengths in inspection tasks while highlighting areas such as high-precision detection that require further refinement.

4.3.3 Summary of System Performance

The results from both the laboratory and field experiments illuminate the system's key strengths while also underscoring specific limitations that emerge in less predictable environments. Under controlled lab conditions, the manipulator consistently demonstrated high detection accuracy and robust pose estimation. With a stable light source, clearly defined object shapes, and a regulated workspace, advanced methods like ICP had minimal difficulty aligning reference models to actual objects. In this setting, sub-mission completion rates and task precision remained consistently high. However, the transition to a competitive field setting introduces significant uncertainty. In the RoboCup Rescue League, external factors such as compressed timeframes, restrictive safety protocols, and the complexities of competition logistics can quickly erode the system's performance margins. Operators often need to limit or forgo attempts if there is any substantial risk to the robot, given that a single hardware failure may affect not only the team's performance but also the result of the overall event. Consequently, decisions made under pressure tend to be on the side of caution, reducing opportunities to test the system's full capabilities.

Moreover, the complexity of certain objects and tasks amplifies these challenges. When objects lack distinct edges or uniform shapes, ICP can converge on local minima or fail to align properly, thereby compromising subsequent manipulations. In a controlled lab environment, such errors can be diagnosed and corrected in iterative cycles, but the high-stakes nature of a competition permits little time to troubleshoot. Likewise, tasks demanding narrower tolerances like emergency push button presses magnify even small inaccuracies in pose estimation or motion planning. Taken together, these observations highlight the necessity for further refinement in robotic design and algorithms. While controlled tests confirm the feasibility of high-performance detection and manipulation, real-world scenarios including demanding competition settings require solutions that can reliably cope with partial data, fluctuating lighting, and safety-driven constraints. Overcoming these obstacles will be essential for advancing autonomous manipulators capable of responding effectively to the unpredictable demands of real disaster environments.

CHAPTER 5
CONCLUSION

In this section, a comprehensive review of the research findings, an analysis of encountered issues, and a set of recommendations for future improvement have been compiled in the following order.

1. Summary of the Research
2. Problems Encountered
3. Suggestions

5.1 Summary of the Research

This thesis focuses on the development and implementation of an advanced object interaction system, integral to the autonomous manipulator arm designed by the iRAP Robot team. The project, titled "Implementing Advanced Object Interaction in Rescue Robotics Using an Autonomous Manipulator Arm," highlights the design and performance of a system that earned the team the Best-in-Class Dexterity award at the RoboCup Rescue Robot Competition 2024. The research emphasizes the integration of perception, planning, and control to enable precise and reliable object manipulation in disaster scenarios. The object interaction system leverages custom-built mechanical, electrical, and software components to achieve robust functionality. Central to the system are the following modules:

5.1.1 Object Detection

The process of object detection in this system utilizes RGB-D cameras, which combine regular RGB images with depth information, providing a detailed 3D view of the environment. This rich data allows for identifying objects with greater accuracy, even in cluttered or complex surroundings. The system leverages YOLOv8, a state-of-the-art object detection model, which excels at real-time performance and high detection accuracy. By training YOLOv8 on a diverse dataset of object images, the model can recognize and localize various objects efficiently. The combination of RGB-D cameras and YOLOv8 ensures reliable detection and spatial positioning of objects, which is critical for downstream processes like pose estimation and manipulation.

5.1.2 Pose Estimation

Pose estimation involves determining the exact position and orientation of objects detected within the robot's workspace. This process employs advanced point cloud processing techniques, where the depth data from RGB-D cameras is converted into 3D point clouds. These point clouds are then aligned with pre-scanned 3D models of the objects using Iterative Closest Point (ICP) methods. Multiple ICP variations, such as point-to-point and point-to-plane, are used to refine alignment accuracy by minimizing discrepancies between the detected object and its model. The use of ICP not only ensures precise alignment but also addresses challenges posed by occlusions or noise in the data, providing the robot with reliable pose information to execute tasks effectively.

5.1.3 Trajectory Planning

To ensure smooth and collision-free movements of the robotic manipulator, the system utilizes trajectory planning tools. The TRAC-IK solver, known for its fast and reliable inverse kinematics calculations, is employed to determine the required joint configurations for the manipulator to reach its target. This is integrated with MoveIt STOMP planner, a powerful motion planning framework that generates optimized paths for the robot's movements. MoveIt considers the manipulator's kinematics, obstacles in the environment, and predefined safety constraints, ensuring that the robot's actions are efficient, precise, and safe. The resulting trajectories allow the manipulator to interact with objects seamlessly, avoiding collisions with obstacles or other parts of the robot.

5.1.4 Impact Detection,

To enhance safety and precision during object manipulation, the system integrates Inertial Measurement Unit (IMU) sensors into the robotic setup. These sensors are capable of detecting changes in acceleration and orientation, which helps identify any unintended contact between the manipulator and the objects or environment. If an impact is detected, the system can immediately respond by adjusting the manipulator's actions or halting operations to prevent damage. This feature not only ensures the safety of the robot and the objects it interacts with but also improves the reliability of tasks performed in dynamic or unpredictable environments. The integration of IMU sensors represents a critical safeguard in maintaining the systems operational.

The system successfully demonstrates advanced dexterity by performing tasks such as linear inspection, omni-inspection, and emergency button pressing with high levels of accuracy and autonomy. Key innovations include multi-method ICP alignment for pose estimation and trajectory planning designed to adapt to variable task geometries and tolerances. These enhancements enable the system to excel in tasks requiring precise object handling, even in environments with constrained spaces and degraded sensor data. While the unified system enhances overall performance, it also introduces challenges in diagnosing and resolving compounded errors across interconnected modules. The research underscores the necessity of balancing sophisticated algorithms with practical considerations, such as hardware reliability and real-world adaptability. Furthermore, the system's robustness is tested against unanticipated object geometries, revealing the importance of human intervention for objects beyond predefined parameters.

The iRAP Robot team's achievement at RoboCup 2024 reflects the success of this integrated approach, demonstrating that advanced object interaction systems can significantly enhance robotic dexterity. This research serves as a foundational step toward bridging the gap between competition-grade robotics and real-world deployment in urban search and rescue scenarios.

5.2 Problems Encountered

The development and implementation of the advanced object interaction system revealed several challenges, each significantly influencing the system's performance and reliability. These problems were not isolated but deeply interconnected, amplifying their impact on the system.

5.2.1 Integration Challenges

Bringing together the core modules of perception, planning, and control into a unified framework introduced unforeseen complexities. While each module operated effectively in isolation, their integration revealed a cascade effect, where errors in one component propagated through the system. For instance, if the perception module provided an incorrect pose estimation, the planning module would generate misaligned trajectories, and the control module would execute inaccurate movements. Troubleshooting such issues became a significant bottleneck, as pinpointing the origin of a failure was often obscured by the interconnected nature of the system. This not only consumed valuable time during development but also led to inefficiencies in real-world applications, such as repeated failed attempts to grasp objects or aborted tasks in critical scenarios.

5.2.2 Environment Adaptation

Another major limitation was the system's reliance on static trajectory planning, which lacked the ability to adapt to dynamic environments. This rigidity meant that the system was ill-equipped to handle unexpected changes in its surroundings, such as shifting debris or the sudden appearance of obstacles in the manipulator's path. Without the capability to recalibrate its trajectory in real-time, the robot frequently misinterpreted environmental factors, leading to collisions or task failures. For example, during the competition's dexterity challenges, the robot might attempt to press an emergency push button but fail due to a small, unnoticed obstruction. Such limitations not only reduced task success rates but also highlighted the system's vulnerability in high-stakes scenarios where adaptability is crucial.

5.2.3 Unfamiliar Objects

A significant challenge arose when the system encountered objects outside its predefined database. The robot's object interaction capabilities were heavily reliant on a prior catalog of known objects, complete with detailed geometric and feature data. When presented with unfamiliar items, the system lacked the ability to generalize or adapt autonomously, instead requiring manual operator intervention to define the object's parameters. This interruption undermined the system's autonomy, introducing delays and breaking the seamless execution of tasks. In real-world rescue missions, where robots often face unpredictable and diverse objects, this limitation becomes a critical hindrance, reducing the system's effectiveness and scalability.

Together, these challenges hindered the system's efficiency and reliability. Integration issues compounded errors, making operations less predictable and harder to optimize. The inability to adapt to dynamic environments exposed the system to failures in unpredictable scenarios, while its dependency on predefined object data restricted its application to controlled settings. These factors collectively diminished the system's autonomy, extended task completion times, and reduced overall success rates, both in the competition and in simulated rescue environments.

Addressing these problems is essential for improving the system's robustness and applicability. Incorporating dynamic environmental sensing, real-time adaptive planning, and advanced object recognition capabilities will be pivotal steps in enhancing the system's reliability, scalability, and autonomy in future iterations.

5.3 Suggestions

Addressing the challenges faced during the development of the advanced object interaction system requires targeted improvements across multiple areas. These refinements are not just incremental enhancements but pivotal steps toward creating a more robust, adaptive, and efficient system. The following suggestions outline transformative changes that would significantly enhance the system's performance, reliability, and applicability in dynamic environments.

5.3.1 Refinement of ICP and Contextual Pose Estimation

One of the core improvements lies in refining the Iterative Closest Point (ICP) algorithm, which plays a critical role in aligning the robot's perception with real-world object geometries. Currently, ICP primarily relies on minimizing Root Mean Square Error (RMSE) to align point clouds, which, while effective, can lead to false alignments in scenarios with complex or ambiguous object features. By incorporating contextual information such as the object's known position relative to its environment or prior task-specific knowledge the system could significantly reduce alignment errors. For example, when handling irregularly shaped objects, integrating spatial awareness or task-specific constraints could improve pose estimation precision. This refinement would enhance the robot's accuracy in identifying and manipulating objects, particularly in cluttered or high-stakes environments.

5.3.2 Closed-Loop Control for Real-Time Adaptation

Introducing a closed-loop control mechanism with continuous environmental monitoring would enable the system to dynamically adapt to changes in its operating environment. Real-time feedback from sensors, such as depth cameras and IMUs, could be used to adjust the robot's trajectory mid-operation. This would address the current limitation of static trajectory planning, allowing the robot to respond effectively to dynamic scenarios, such as moving obstacles or shifting debris. For instance, in a disaster scenario, if a piece of debris unexpectedly falls into the manipulator's path, the system could recalibrate its movement in real time to avoid collisions and maintain task accuracy. This capability would not only improve operational efficiency but also significantly increase the robot's reliability in unpredictable environments.

5.3.3 Trajectory Optimization for Unique Kinematics

The robot's central prismatic joint introduces a unique kinematic configuration that requires specialized trajectory planning. Current methods often result in suboptimal joint movements, leading to inefficient or unnecessarily complex motion paths. By optimizing the trajectory planner to account for the robot's specific kinematics, the system could ensure smooth, energy-efficient, and precise motions. For example, incorporating kinematic constraints that balance the extension of the prismatic joint with rotational movements of other joints would minimize unnecessary adjustments, enhancing both speed and accuracy. This improvement would be particularly beneficial in scenarios requiring precise positioning, such as inserting objects into tight spaces or navigating narrow corridors.

5.3.4 Integration of Vibrational Sensing

Embedding flexible vibration-sensing components within the manipulator's gripper would add a tactile dimension to the robot's interaction capabilities [16]. These sensors would provide real-time feedback on contact with objects, enabling the system to detect and respond to subtle changes in force and alignment. For instance, during a delicate task like pressing an emergency push button, the gripper could use vibrational feedback to ensure the button is fully pressed without applying excessive force that could damage the mechanism. Similarly, vibration sensing could enhance pose estimation by detecting micro-movements during object alignment, reducing the likelihood of errors. This improvement would make the system more adept at handling tasks requiring precision and care, significantly increasing its effectiveness in both competition and real-world rescue scenarios.

REFERENCES

1. Tadokoro, S., etal. (2000). "The RoboCup-Rescue project: a robotic approach to the disaster mitigation problem," Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation, San Francisco, CA, USA, : 4089-4094

2. Narasetthakul, A., Sukcharoen, J., Thanawutthianan, P., Inthiam, J., Pudchuen, N., and Blattler, A. (2023). “RoboCup Rescue 2023 Team Description Paper iRAP ROBOT” ROBOCUP RESCUE 2023 TDP COLLECTION

3. Kalakrishnan, M., Chitta, S., Theodorou, E., Pastor, P., and Schaal, S. (2011). "STOMP: Stochastic trajectory optimization for motion planning," 2011 IEEE International Conference on Robotics and Automation, Shanghai, China, : 4569-4574

4. Oonariya, T., Booleua, K., Blattler, A., and Inthiam, J. (2024). "System Development of 3D Machine Vision For Automated Bin Picking," 2024 1st International Conference on Robotics, Engineering, Science, and Technology (RESTCON), Pattaya, Thailand, : 170-174.

5. Chen, Y. K., Sun, G. J., Lin, H. Y., and Chen, S. L. (2018). “Random Bin Picking with Multi-View Image Acquisition and CAD-Based Pose Estimation” 2018 IEEE International Conference on Systems, Man, and Cybernetics

6. Kuo, H. Y., Su, H. R., Lai, S. H., and Wu, C. C. (2014). “3D Object Detection and Pose Estimation from Depth Image for Robotic Bin Picking” 2014 IEEE International Conference on Automation Science and Engineering (CASE) Taipei, Taiwan

7. Kanso, A., Xu, X., Rekik, K., and Muller, R. (2022). “A concept to solve Bin Picking problem in the robotics based on CAD Model and Measurement data” 2022 10th International Conference on Systems and Control, Marseille, France

8. Pudchuen, N., and Jitviriya, W. (2019). "3-D Mapping and Localization using RGB-D Camera," 2019 First International Symposium on Instrumentation, Control, Artificial Intelligence, and Robotics (ICA-SYMP)

9. Mihalyi, R., Pathak, K., Vaskevicius, N., Doernbach, T., and Birk, A. (2015). “Robust 3D object modeling with a low-cost RGBD-sensor and AR-markers for applications with untrained end-users. Robotics and Autonomous Systems”

10. Low, K. (2004). “Linear Least-Squares Optimization for Point-to-Plane ICP Surface Registration”

11. Rusinkiewicz, S., and Levoy, M. (2001). "Efficient variants of the ICP algorithm," Proceedings Third International Conference on 3-D Digital Imaging and Modeling, Quebec City, QC, Canada

12. Krieg, R., and Ebner, M. (2022). "Time Series Classification of IMU Data for Point of Impact Localization," 2022 Sixth IEEE International Conference on Robotic Computing (IRC), Italy, : 93-94.

13. Connolly, C. (1985). "The determination of next best views," Proceedings. 1985 IEEE International Conference on Robotics and Automation, St. Louis, MO, USA, : 432-435.

14. Dolezel, P., Pidanic, J., Zalabsky, T., and Dvorak, M. (2019). “Bin Picking Success Rate Depending on Sensor Sensitivity” 2019 IEEE

15. Quyen, N. V., Hai Linh, B. T., Tuan, D. Q., Thanh, N. T., and Quang, N. H. (2023). "Dynamic Stability Control and Calculating Inverse Dynamics of a Single-link Flexible Manipulator," International Journal of Mechanical Engineering and Robotics Research, Vol. 12, No. 4 : 200-209.

16. Mohammed, M. J., Abbas, B. A., and Al-Azawiey, S. A. (2019). "Flexible Underwater Manipulator Modelling Using Intelligent Method" International Journal of Mechanical Engineering and Robotics Research, Vol. 8, No. 6, : 860-866.

VITA

Name Mr. Thoranin Oonariya

Thesis Title Implementing Advanced Object Interaction in Rescue Robotics Using an Autonomous Manipulator Arm

Major Field Automation Engineering

Biography Year 2562-2565, completed bachelor's degree studies, Faculty of Engineering, Department of Production and Robotics Engineering, Robotics and Automation Engineering major, King Mongkut's University of Technology North Bangkok.

Year 2566-present, currently pursuing a master's degree in Automation Engineering, Department of Instrumentation and Electronics Engineering, , King Mongkut's University of Technology North Bangkok

The work was presented at an international academic conference, The 4th International Conference on Robotics, Automation, and Artificial Intelligence (RAAI 2024), the 4th iteration, during December 19–21, under the title "Development and Integration of an Autonomous In-House Manipulator Arm for Enhanced Object Interaction in Rescue Robotics."