Semi-Autonomous Grasp Planning Algorithm for Underwater Applications

Farooq Olanrewaju

Thesis to obtain the Master of Science Degree in

Electrical and Computer Engineering

Supervisors: Prof. Pedro Sanz
Prof. Rodrigo Martins de Matos Ventura

Examination Committee

Chairperson: Prof. José Eduardo Charters Ribeiro da Cunha Sanguino
Supervisor: Prof. Rodrigo Martins de Matos Ventura
Member of the Committee: Prof. Plinio Moreno López

November 2024

This work was created using $\LaTeX$ typesetting language
in the Overleaf environment (www.overleaf.com).

Declaration

I declare that this document is an original work of my own authorship and that it fulfills all the requirements of the Code of Conduct and Good Practices of the Universidade de Lisboa.

Acknowledgments

In the name of Allah, the Most Gracious, the Most Merciful. All praise is due to Allah, the Almighty, for His mercy, benevolence, guidance, and protection throughout my academic journey, during my time at Université de Toulon, Instituto Superior Técnico, and the Research Center in Robotics and Submarine Technologies (CIRTESU). I am grateful for His blessing of attaining the Erasmus Mundus Joint Master Degree (EMJMD) in Marine and Maritime Intelligent Robotics (MIR). Verily, it is He who has bestowed knowledge upon mankind through the pen, and I pray that He illuminate my path with beneficial knowledge and wisdom.

Afterwards, I express my sincere appreciation to my parents, siblings, in-laws, and my entire family for their unwavering care and support. Esteemed thanks to my supervisors, Prof Rodrigo Ventura and Prof. Pedro Sanz, for their invaluable constructive criticism, insightful suggestions, and continual guidance that played a pivotal role in the successful completion of this research work.

I would also like to extend my appreciation to my colleagues in the MIR program. The camaraderie, collaborative spirit, and shared pursuit of knowledge made this academic journey more enriching and enjoyable. Thank you for the exchange of ideas, support during challenging times, and the memorable experiences we shared. Furthermore, my sincere gratitude goes to the research staff and colleagues at CIRTESU: Prof. Raúl Marín Prades, Prof. Antonio Morales, Alejandro Solis Jiménez, Salvador López Barajas and others. Your criticism, technical support, research guide, teamwork, and collective effort significantly contributed to the success of the research project. I am truly grateful for the collaborative environment that fostered this innovation and excellence.

To my amazing and loving wife, Fareedah Adeteru, words cannot sufficiently convey my gratitude for your love, care, support, and encouragement. Allah has made you a source of strength and solace, and I am profoundly grateful to Allah for making you a beacon of support in my endeavours. JazaakiLlaah khayraa, sweetheart!

Finally, heartfelt appreciation goes to all those who have contributed to my journey and made this research work possible. A sincere thank you to everyone whose impact has been instrumental in shaping my academic and personal growth.

Abstract

Various activities in the underwater environment traditionally involve human divers in shallow waters or manned submersibles and Remotely Operated Vehicles (ROVs) equipped with manipulators in deeper waters. These approaches come with inherent drawbacks, including risk to human life at dangerous depths, operator fatigue in the case of ROVs, limited operational time, and high operational costs associated with manned submersibles or ROVs. The development Intervention Autonomous Underwater Vehicles (I-AUVs) became of importance and, in addition, the need for effective grasp planning algorithms became crucial for underwater intervention tasks. Existing methods for grasp planning often struggle with autonomously proposing reachable grasp poses from visual data due to constraints such as poor visibility and computational limitations. This thesis presents a semi-autonomous grasp planning algorithm that utilizes the geometric properties of target objects to propose suitable grasp poses, assuming object uniformity and rigidity. The algorithm was developed using real-time data from stereo cameras, yielding satisfactory grasp propositions. The results demonstrate significant improvements in crucial subsea operations, with potential applications in cooperative grasp planning.

Keywords

Underwater Robotics, Geometric Grasp Planning, Point Cloud Segmentation.

Resumo

O desenvolvimento de algoritmos eficazes de planeamento de grasp é crucial para tarefas de intervenção subaquática. Métodos existentes frequentemente enfrentam dificuldades em propor autonomamente posições de grasp alcançáveis a partir de dados visuais devido a limitações como baixa visibilidade e limitacoes computacionais. Esta tese apresenta um algoritmo semi-autônomo de planeamento de grasp que utiliza as propriedades geométricas dos objetos-alvo para propor posições de grasp adequadas, assumindo uniformidade e rigidez dos objetos. O algoritmo foi desenvolvido usando dados em tempo real de câmeras estéreo, resultando em propostas de grasp satisfatórias. Os resultados demonstram melhorias significativas em operações subaquáticas cruciais, com aplicações potenciais em planeamento de grasp cooperativa.

Palavras Chave

Robótica Subaquática; Planeamento de Grasp Geométrica; Segmentação de Nuvem de Pontos.

Contents

List of Figures

Acronyms

AI	Artificial Intelligence
AUV	Autonomous Underwater Vehicle
CAD	Computer Aided Design
CIRTESU	Research Center in Robotics and Submarine Technologies
CNN	Convolutional Neural Network
COLA2	Component Oriented Layer-based Architecture for Autonomy
COOPERAMOS	COOPErative Resident robots for Autonomous ManipulatiOn Subsea
DOF	Degree of Freedom
DL	Deep Learning
DVL	Doppler Velocity Log
EMJMD	Erasmus Mundus Joint Master Degree
GWS	Grasp Wrench Space
GUI	Graphical User Interface
HOG	Histogram of Gradients
I-AUV	Intervention Autonomous Underwater Vehicle
MARIS	Marine Robotics for Intervention
MIR	Marine and Maritime Intelligent Robotics
ML	Machine Learning
RCNN	Region-based Convolutional Neural Network
RFCN	Region-based Fully Convolutional Network
ROS	Robot Operating System
ROV	Remotely Operated Vehicle

SA Segment Anything

SAUVIM Semi-Autonomous Underwater Vehicle for Intervention Missions

SIFT Scale Invariant Feature Transform

SSD Single Shot Multibox Detector

SURF Speed Up Robust Transform

SVM Support Vector Machine

UJI Jaume I University

USBL Ultra-Short Baseline

YOLO You Only Look Once

1

Introduction

Contents

1.1 Background

Underwater technology has experienced significant advancements driven by the demand for safer and more efficient subsea operations. The exploration and utilization of underwater environments have always been challenging due to harsh conditions, limited visibility, and the complexities associated with deep-sea operations [11]. Initially, human divers were employed for underwater tasks, but the inherent risks and limited operational depth and duration necessitated the development of alternative solutions.

The introduction of Remotely Operated Vehicles (ROVs) in the 1960s marked a significant milestone in underwater technology [12, 13]. ROVs, controlled by operators from the surface via tethers, provided enhanced safety and the ability to operate at greater depths compared to traditional diver operations [11, 14]. These vehicles played a crucial role in various underwater applications, including inspection, maintenance, and exploration [15]. However, ROVs come with high operational costs and complexities, requiring support ships and continuous human oversight [16].

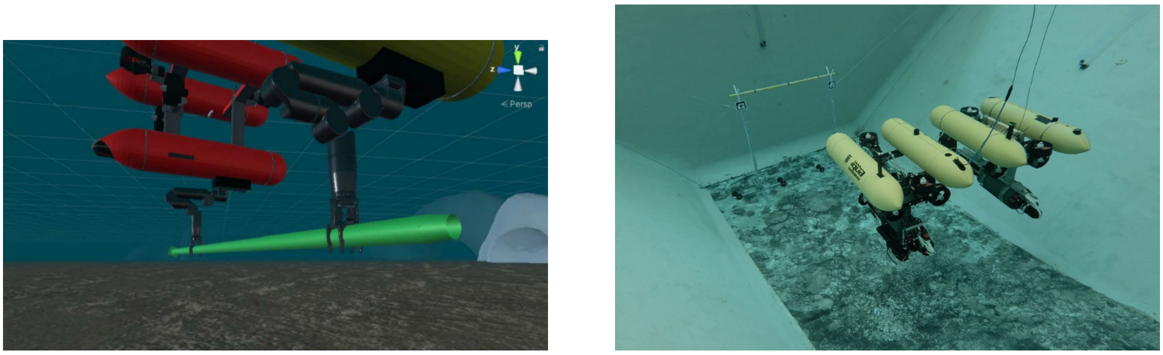
In response to these limitations, the 1980s and 1990s saw the development of Autonomous Underwater Vehicles (AUVs) [17], which offer greater autonomy and can perform remote missions without continuous human control [13]. AUVs have been utilized in a variety of applications, such as mapping [18], environmental monitoring [14, 19], and resource exploration. Despite their advantages, AUVs face challenges in performing intervention tasks due to their limited manipulation capabilities [20, 21].

To overcome these challenges, researchers have proposed the development of Intervention Autonomous Underwater Vehicles (I-AUVs). I-AUVs are equipped with advanced sensors, manipulator arms, and sophisticated control systems, enabling them to interact with and manipulate objects in their environment effectively [21, 22]. They are designed to perform tasks such as retrieving seabed samples, conducting repairs on underwater infrastructure, and other complex manipulation tasks without direct human intervention [14]. These vehicles combine the autonomy of AUVs with the manipulation capabilities of ROVs, making them versatile tools for underwater operations. Hence, researchers recently have focused on different techniques to improve the capabilities of I-AUVs through manipulator cooperation with grasp planning [23–27].

Grasp planning algorithms are critical for the effective operation of I-AUVs. These algorithms usually account for some of the known challenges of the underwater environment, such as limited visibility, variable lighting conditions, and the presence of currents. By integrating advanced sensors, mathematical models, and robust control systems, these algorithms enhance the precision and reliability of underwater manipulations [28]. Grasp planning algorithms will enable I-AUVs to perform tasks such as object retrieval, assembly, and maintenance with high accuracy, even in dynamic and uncertain underwater conditions.

The development of these technologies has led to significant advancements in underwater robotics, with various projects and research initiatives contributing to the field. Notable projects include Semi-

Autonomous Underwater Vehicle for Intervention Missions (SAUVIM), which demonstrated the feasibility of using I-AUVs for complex underwater operations [17]; Marine Robotics for Intervention (MARIS), which focused on developing advanced autonomous systems for underwater exploration and intervention [26]; and TWINBOT, which explored cooperative manipulation with multiple underwater robots [28]. These projects have paved the way for the development of sophisticated grasp planning algorithms and control strategies that enable I-AUVs to perform a wide range of underwater tasks autonomously.



(a) COOPERAMOS Simulation Engine **(b)** TWINBOT Experimental Set-Up

Figure 1.1: Simulation and Experimental Set-Up from COOPERAMOS and TWINBOT projects

The COOPERAMOS project is one of the projects aimed at improving the capabilities of I-AUVs by implementing a residual dual-arm I-AUV. This involves the coordination of two robotic arms to perform specified tasks in the underwater environment. The conceived task involves robot cooperation in three stages: mobile manipulation, transportation and assembly. Studies on coordinated transportation have been done in the TWINBOT project and the algorithm was experimented on a long pipe [24]. Previous studies within the COOPERAMOS project have demonstrated coordinated transportation and assembly of a long pipe, showcasing the potential for advancements in multi-robot system manipulation. This thesis seeks to further enhance the grasp planning mission of the cooperation algorithm for the manipulation of the multi-robot system, focusing on the transportation and assembly of some specific objects.

The evolution from ROVs to AUVs and the subsequent development of I-AUVs underscore the dynamic landscape of underwater technology. Ongoing research initiatives, such as the COOPERAMOS project, signify a commitment to addressing the limitations of existing technologies and pushing the boundaries of autonomous underwater interventions. As these technologies advance, the potential for safer, more efficient, and versatile underwater operations continues to grow, offering promising prospects for various industries reliant on subsea exploration and infrastructure maintenance.

1.2 Motivation of the Study

Various activities in the underwater environment, spanning marine search and rescue, underwater archaeology, dam inspections, oil well maintenance, and oceanography, traditionally involve human divers in shallow waters or manned submersibles and ROVs equipped with manipulators in deeper waters. However, these approaches come with inherent drawbacks, including risk to human life at dangerous depths, operator fatigue in the case of ROVs, limited operational time, and high operational costs associated with manned submersibles or ROVs. The emergence of I-AUVs presents a transformative solution that offers the potential to carry out these activities with reduced or eliminated drawbacks, making them a promising alternative in underwater operations.

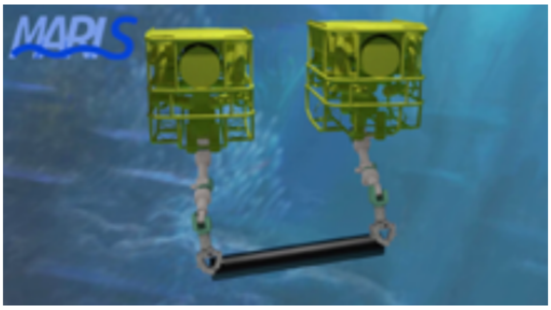
(a) SAUVIM Vehicle

(b) MARIS Single-Agent Operation

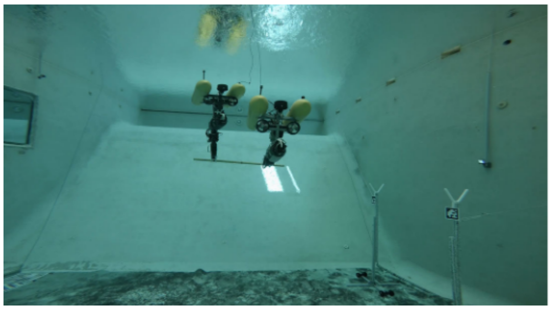
Figure 1.2: Intervention in Underwater Environment with Single-Agent Free-Floating I-AUVs

Over the last two decades, there has been significant research focused on the development of single-vehicle I-AUVs, particularly those designed for search and recovery tasks. Notable achievements in this domain include the SAUVIM project [29], where a free-floating autonomous vehicle accomplished the recovery of a pre-specified object by autonomously locating the object and hooking to the vehicle [24, 30]. Similar efforts have been undertaken in projects like MARIS [31], contributing to the exploration of autonomous search and recovery capabilities.

While single-vehicle I-AUVs have demonstrated success in some specific tasks, there is a growing recognition of the need for more sophisticated manipulation and transport capabilities, particularly for larger objects. To address this requirement, researchers have explored the use of dual-arm manipulators in I-AUVs or the cooperative control of multiple I-AUVs to accomplish these more complex tasks [24, 32]. The TWINBOT project stands out as one of the pioneering efforts in this direction, showcasing a cooperative transportation task through a leader-follower organization for cooperation and control, coupled with a visual-servoing technique for grasping [24]. This cooperative approach introduces a new dimension to underwater intervention capabilities, paving the way for enhanced dexterity and versatility in handling substantial objects in the underwater environment.



(a) Simulation of Cooperative Transportation in MARIS Project



(b) Experimental Cooperative Transportation with Girona-500 in TWINBOT Project

Figure 1.3: Intervention in Underwater Environment with Multi-Agent I-AUVs

1.3 Problem Description and Objectives

This thesis is intricately linked to the ongoing COOPERAMOS project at Jaume I University (UJI) which is funded by the Spanish Ministry of Science and Innovation. Its primary goal is to design advanced grasp planning algorithms for an underwater scenario using two I-AUVs with coordination. This project will build on the previous works in the TWINBOT project which uses two Girona500 AUVs equipped with 7 Degree of Freedom (DOF) robotic arm for transportation of a long pipe cooperatively. In the TWINBOT project, the grasping points are manually computed and the robot is controlled through a planned path to the defined point. Hence, the main objective of this project is to develop and automated grasp position and orientation proposal algorithm.

The objectives outlined to achieve the goal are:

1. Implementation of Visual Segmentation Algorithm in Intervention Area: this algorithm would aid in better identification of the objects in the scene by using data from stereo cameras.

2. Development of Efficient Grasp Planning Algorithm based on Visual Information.

3. Testing of Algorithms with cameras mounted on the I-AUVs: upon completion and rigorous successful testing in the simulation environment, the algorithms could be implemented on Girona500 AUVs at the Experimental Test Bed in Research Center in Robotics and Submarine Technologies (CIRTESU) to validate the simulation results.

1.4 Main Contribution

The project investigates the development of objective-specific geometric grasp proposals for a manipulator for transportation of slender rigid rods in an underwater environment.

We:

- design an experiment to test the applicability of plane and color segmentation methods to obtain a target from point cloud data,
- developed a geometric-based grasp proposition algorithm for a point cloud input or a CAD model input of the object,
- validated the developed algorithm on real-time data collected from stereo cameras in the lab which produced satisfactory grasp propositions.

1.5 Thesis Outline

This thesis is structured into five chapters. Chapter 1 provides an introduction to the background and motivation of the project, outlines the project objectives, presents the current state of the art in the field and lists the main contributions of this work. Chapter 2 delves into description of the existing system architecture and a review of existing techniques related to visual segmentation and grasp planning, and discusses some methods that were applied to underwater problems. Chapter 3 outlines the methodology that was employed in the project with some details on the mathematics underlying the segmentation methods and the grasp planning techniques developed. Chapter 4 presents the results of the designed experiments for visual segmentation and the geometric grasp planning. Chapter 5 presents the conclusions and system limitations of this work and the proposed future works.

2

State of the Art

Contents

In this chapter, we will review the literature on the main areas that are necessary for the development of a comprehensive grasping algorithm: image detection, segmentation and grasp planning. During the detection and segmentation stage, an image captured from a camera feed is processed to identify and segment the object to be grasped within the workspace. In the grasp planning stage, the segmented image is analyzed to determine stable and reachable grasp points, which are then output as position and orientation vectors. The motion planning phase involves steering the underwater vehicle towards the desired object based on the vectors obtained from the grasp planner. Finally, the manipulation stage involves actuating the mounted arm to grasp the object. The primary focus of this review is on applications within underwater environments.

2.1 Image Detection and Segmentation

In any process with vision capabilities, image detection and segmentation are usually the backbone or a critical component of the architecture. These processes are important for some underwater tasks, such as studying marine animals and plants, underwater archaeology, inspection of underwater infrastructures, etc. In this subsection, we will discuss different methods used in image detection and segmentation, challenges specific to underwater environments, and how these two techniques can be integrated.

2.1.1 Image Detection

Image detection involves the identification and localization of objects of certain classes within an image. There are traditional approaches (i.e., non-AI) and Deep Learning (DL)-based approaches to image detection [33].

2.1.1.A Traditional Methods for Image Detection

The traditional methods of image detection involve using some computational methods to extract some features from the image. The extracted features are then used to make detections from the image. These techniques have been used in Viola Jones Detectors, Histogram of Gradients (HOG), Scale Invariant Feature Transform (SIFT), Speed Up Robust Transform (SURF) methods and many others [33, 34].

These traditional methods have been increasingly replaced by DL based approaches in many domains. However, feature extraction-based approaches are still in use in the marine domain where the processed images usually lack quality and low-level analysis is important for improved accuracy. Gupta and Sharma [35] applied HOG and SIFT to image detection in underwater environment and also used these techniques to improve degraded images. Manonmani et al [36] used HOG and Canny Edge Detector for more accurate mine detection for naval defense applications. Wang et al [37] also used HOG to detect sea cucumbers for fishing applications. Demir and Yaman [38] used HOG to detect garbage in underwater environments.

Hence, traditional methods are still widely used in the marine context and could be very pertinent when high accuracy and low-level analysis are pertinent [35].

2.1.1.B Artificial Intelligence (AI) Methods for Image Detection

The AI methods have been dominant in the computer vision literature since the introduction of AlexNet in 2012 [39]. DL-based methods involve the learning of robust and high-level image representation with little or no handcrafted feature extraction [40, 41] unlike traditional methods.

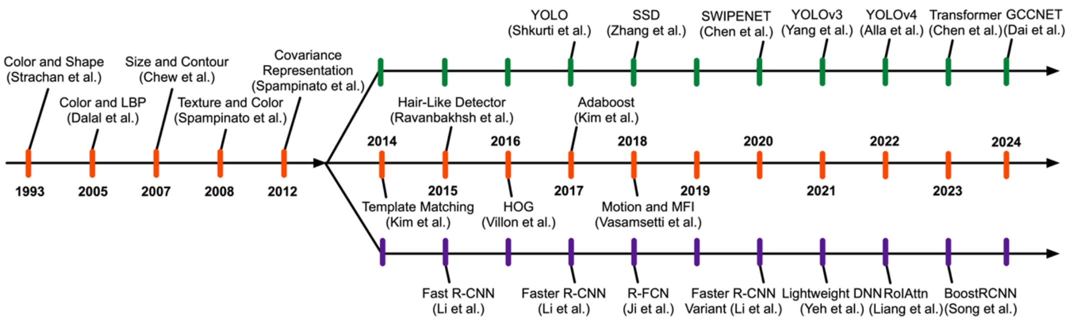


Figure 2.1: Timeline of Image Detection Techniques (including the traditional, one-stage and two-stage detection methods) [1].

Some of the popular AI-based techniques used in image detection include:

- **Traditional Machine Learning (ML) methods**: these methods usually use handcrafted methods for image extraction then the ML technique is used only for classification. Some of the techniques used are: Support Vector Machine (SVM) and decision trees.

- **Two-Stage DL-based Detectors**: in the DL methods, feature extraction is not handcrafted. Rather, this technique uses a separate deep network for object proposal and another for feature extraction, bounding-box regression for object localization in the image, and softmax classification. Some DL models that use this architecture include Region-based Fully Convolutional Network (RFCN) and Region-based Convolutional Neural Network (RCNN) as seen in Figure 2.2.

- **One-Stage DL-based Detectors**: these methods are developed to increase the speed of real-time image detection. The bounding box and classes are predicted directly from the images in one evaluation [3] without initially determining the predicting regions. Common models are the You Only Look Once (YOLO) model [3], RetinaNet [42] and Single Shot Multibox Detector (SSD) [43].

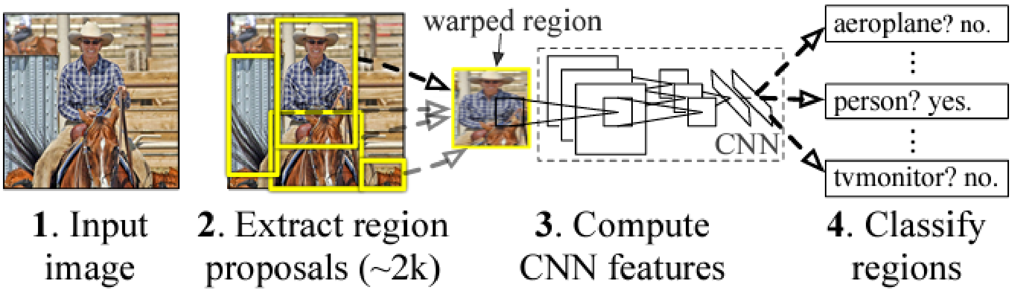


Figure 2.2: Schematics of the RCNN Detection Model, a Two-Stage Detection Approach [2].

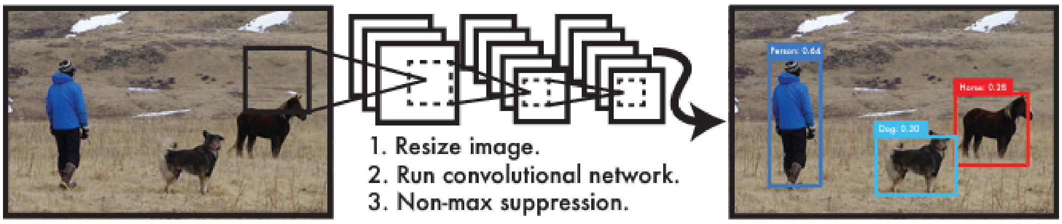


Figure 2.3: Schematics of the YOLO Detection Model, a One-Stage Detection Approach [3].

Both the two-stage and the one-stage detectors have been applied to underwater image detection problems. Some of these works have been highlighted by Sarkar et al [44]. Zhang et al [45] used CNN-based architecture to detect sea animals and their model has a mean average precision of 63.9% on the dataset it was tested on. Mahmood et al [46] used a VGG-based network to classify objects in the underwater environment. Chen et al [47] proposed a single-shot detection type architecture that is capable of detecting small objects. His model has a mean average precision of 46.3%. Although these deep learning methods are quite advanced and robust, they are yet to be able to provide good precision in detecting underwater objects.

2.1.1.C Challenges of Image Detection in Underwater Environment

Image Detection in underwater environment is characterized by many problems. Some of the challenges encountered are summarized by Chen et al [1] and some of these challenges are:

- **Low Quality**: most of the underwater images have poor constrast, distortions, poor lighting and problems with color.
- **Small Object Size**: many aquatic animals and objects to be detected in the marine environment are small-sized and might be clustered in a small environment. This makes detection more challenging.
- **Dynamic or Poor Illumination**: only the first few hundred meters of the underwater environment

is illuminated and this illumination becomes poorer with depth. Acoustic cameras are used more commonly in the marine domain due to their applicability in low visibility [48].

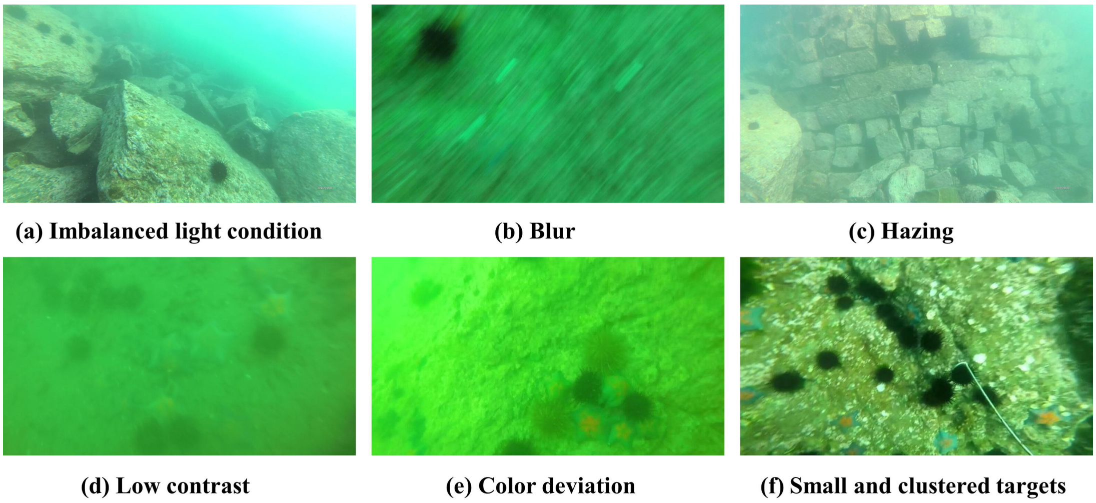


Figure 2.4: Some challenges encountered with underwater images [4].

These challenges are the reasons why many underwater image detection applications are mainly based on generic object detection models with addition of some image enhancements [4].

2.1.2 Image Segmentation

Image segmentation is an important problem in many computer vision applications. It involves partitioning of objects in an image into categories. Segmentation is important in robotic perception and manipulation where it might be required that the target object be segmented from the environment [49]. This enables the robot to distinguish between the target object and the background elements. The image segmentation techniques can be classified into the classical methods and the AI-based methods.

2.1.2.A Classical Image Segmentation Techniques

The classical techniques are broadly classified as techniques that do not involve the use of AI algorithms. The classical method mainly relies on handcrafted metrics to segment images. Such as edges, contrast, average intensity, etc. Some of the classical methods include the following:

- **region-based segmentation**: this method involves the segmentation of objects in an image by grouping them based on similarity in properties such as color or intensity. The segmentation begins at some initially seeded points and it progresses by including neighboring pixels with similar properties until a specified threshold is reached. Zhang et al [50] used this region-based method to segment fishes from complex underwater backgrounds, as seen in Figure 2.5. Li et al [51] and

Chen et al [52] have also applied this method to segmentation of sonar images in underwater applications.

- **edge-based segmentation**: this method detects the boundary of objects by identifying significant changes in the pixel intensity. These significant changes in pixel intensity usually occur at the edge of the segmented objects. The boundary of the identified object is simply formed by connecting the detected edges. Some operators, such as Canny and Sobel operators, are relied upon for this form of segmentation. Some recent research on underwater image segmentation use this approach. Priyadharsini and Sharmila [53] used the edge-based method to detect objects on the seabed. Setiawan et al [54] used this method for detection of edges in underwater image with very low contrast. Afreen et al [55] also recently used this method to track the effect of climate change on coral reefs.

- **layer-based segmentation**: this method involves the segmentation of image by analyzing the depth of the pixels. This is usually useful for scenarios where objects are layered at different depths in an image. This technique is particularly important for object localization and manipulation in robotics systems where the layer segmentation gives a better understanding of the scene [56].

(a) Original Image **(b)** Segmented Image

Figure 2.5: Region-based Segmentation of Objects in Underwater Environment

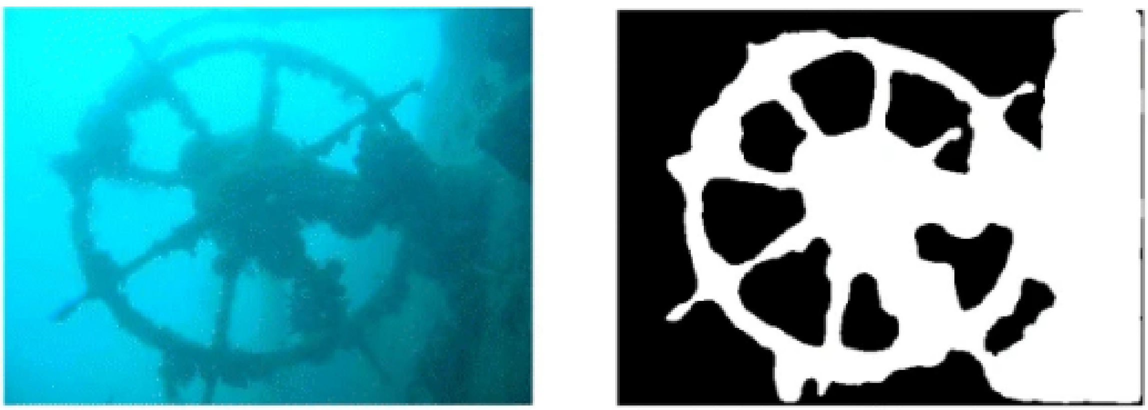
2.1.2.B AI-based Image Segmentation Techniques

Unlike the classical techniques, the AI-based segmentation techniques make use of the ML and DL algorithms which enables more autonomous and highly adaptive image segmentation. The AI-based segmentation methods can be classified into ML-based methods, Convolutional Neural Network (CNN)-based methods and transformer-based methods.

- **ML-based Segementation**: these methods are in use before the widespread of DL-based methods. Classical ML algorithms, such as SVMs and K-Means clustering, take specific image characteristics, such as color, texture and intensity gradient, as input and thereafter group the pixels based on their similarities.These methods are considered computationally efficient and effective for scenarios where data is limited.

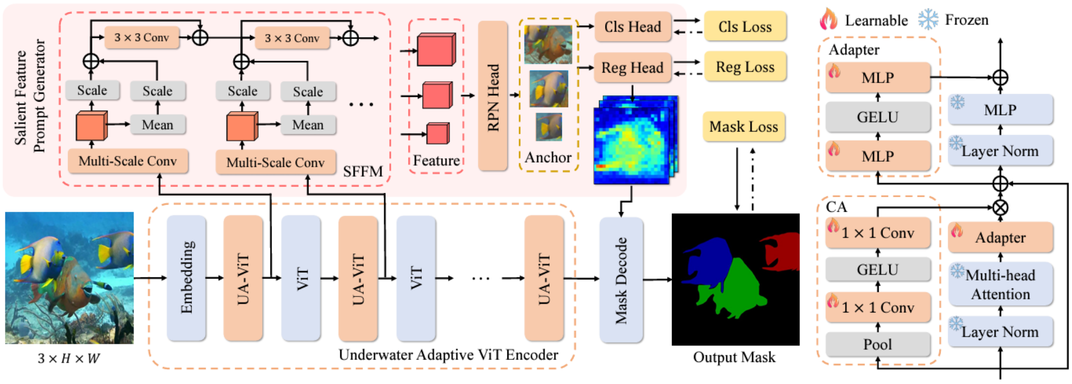
- **CNN-based Segmentation**: the CNN-based method revolutionarized the image segmention methods. Prior to the CNN-based methods, handcrafting of features are still required for image segmentation. However, these methods automated the extraction of features from large datasets and the segmentation of objects in the images. Some popular CNN-based segmentation models include Mask R-CNN and U-Net. Drews-Jr et al [57] used a U-Net based architecture to segmention divers, aquatic animals and static objects form underwater images, see Figure 2.6, with an accuracy of 91.9%.

- **Transformer-based Segmentation**: although the transformer architecture was developed for NLP applications, it has also been extended to image segmentation to capture global contexts in the images. One of the most recent and highly used transformer-based segmentation algorithm is the Segment Anything (SA) model [58]. Lian et al [59] adapted the SA model to underwater images by introducing an underwater adaptive vision transformer encoder. The schematics of this new method is shown in Figure 2.7 and its application to an underwater scenario is shown in Figure 2.8.



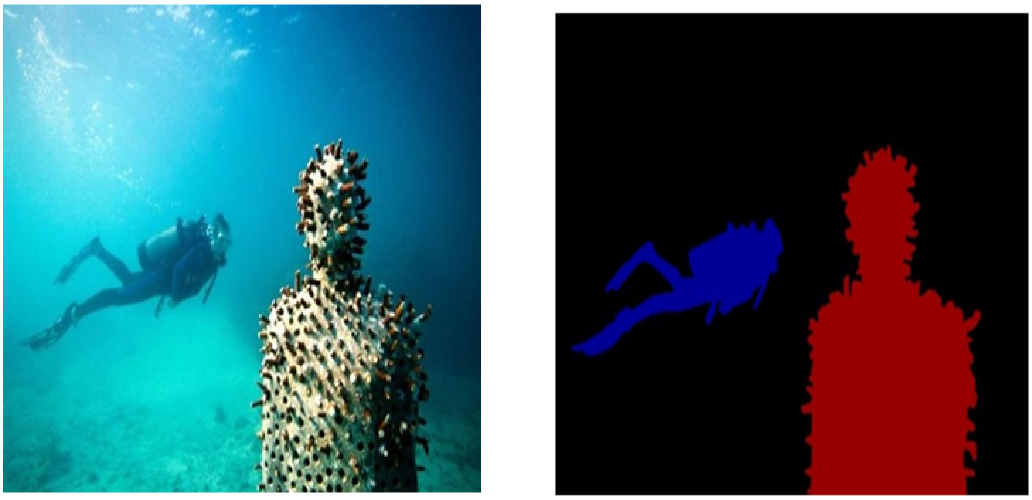
(a) Original Image **(b)** Segmented Image

Figure 2.6: CNN-based Segmentation of Objects in Underwater Environment



(a) USIS-SAM framework (b) UA-ViT Block

Figure 2.7: The Framework of the Underwater Salient Instance Segmentation - Segment Anything Model.



(a) Original Image **(b)** Segmented Image

Figure 2.8: Transformer-based Segmentation of Objects in Underwater Environment

2.2 Grasp Planning

Grasp planning is a fundamental concept in robotic manipulation. It constitutes the determination of the pose of the manipulator gripper to achieve a specific manipulation task. This is critical in many domains where robotics has been applied such as in industrial robots, surgical robots, etc. For autonomous grasp planning, perception and semantic segmentation of the target object is important.

2.2.1 Methods of Grasp Planning

Numerous approaches have been proposed for grasp planning in literature. These methods can be broadly classified as: manual computation methods, geometric methods and DL-based methods. Each method has its advantages and disadvantages depending on the application scenario, availability of computational power, and the nature of the target object.

2.2.1.A Pre-Computed Grasp Point Methods

This method involves an offline pre-computation of the optimal grasp pose of known objects. This method is generally applicable to scenarios where the environment is structured and static and the object to be manipulated is known prior to the manipulation task [60]. In this method, a database of object model is created and the grasp hypothesis are defined based on simulation or empirical estimations. For example, an industrial robot deployed for pick-and-place applications or an assembly task could have its grasp points precomputed depending on the task and the target object. The major shortcoming of this method is that it is neither applicable to dynamic environments nor applicable to objects that are not previously known [61]. One application of this method in manipulation is the research of Vahrenkamp et al [5] where they developed an algorithm to propose grasp position of familiar objects. The schematic diagram of their method is given in Figure 2.9 and an experimental example is given in Figure 2.10.

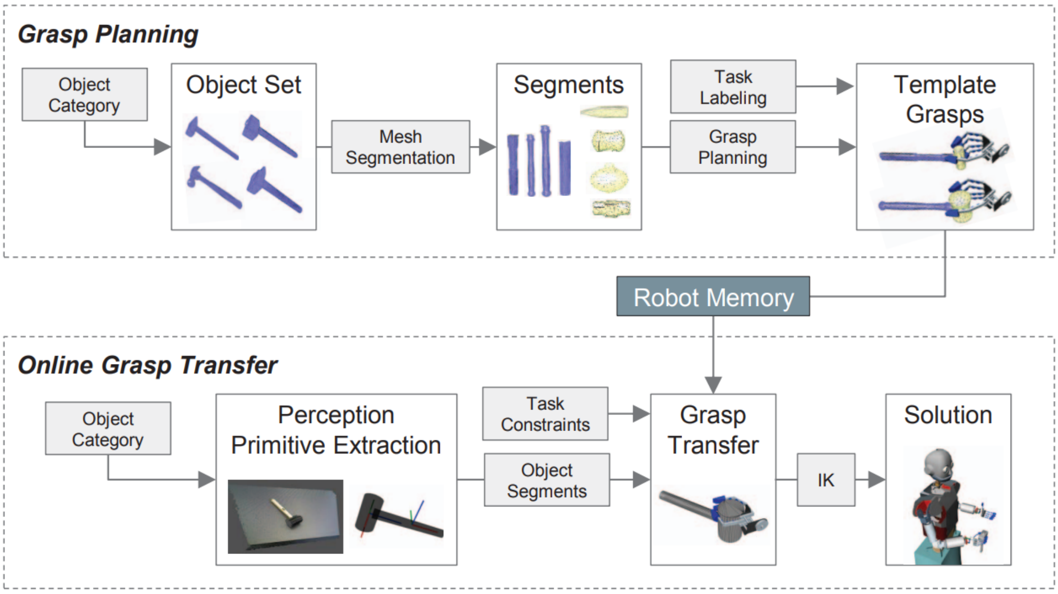


Figure 2.9: Part-based grasp planning for multiple known objects [5].

2.2.1.B Geometric Methods

The geometric grasp planning methods involve the identification of possible grasp pose of an object in real time based on its shape and spatial features. These methods are very suitable for scenarios where the geometry of the object is available either from a Computer Aided Design (CAD) model or a 3-D segmented image (or point cloud object). This method is very efficient and robust for handling unknown

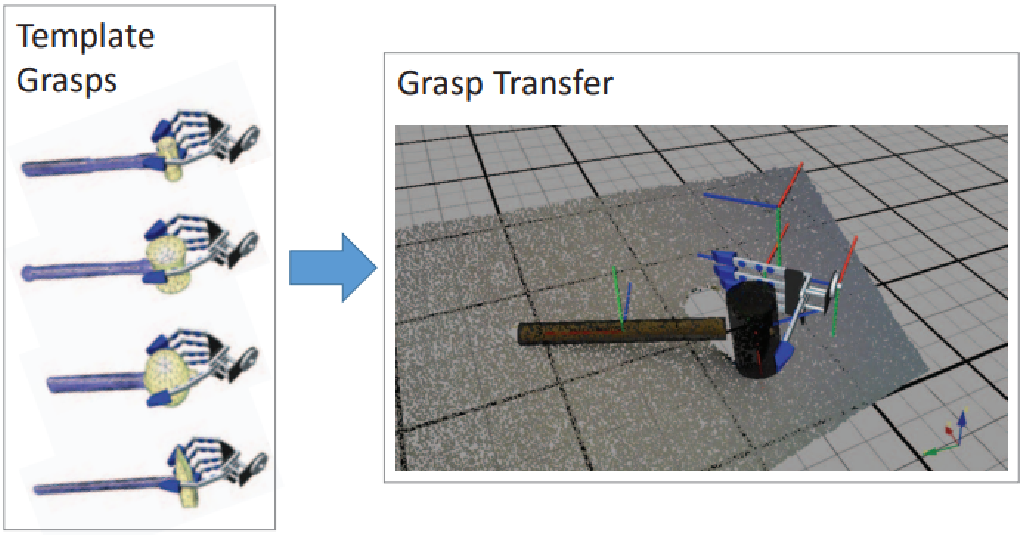


Figure 2.10: Template grasp planning for a wrench object and application to a familiar object in workspace [5].

object shapes. However, it has drawbacks with flexible and irregular objects. Miller et al [62] applied this technique to pick-and-place operation of service robots without explicitly defining the objects ahead. More recently, Akbari at al [6] used this technique for defining the grasp pose of objects by fitting the objects on an ellipsoid and grasping from one of the three focus points of the ellipsoid. The overview of their architecture is given in Figure 2.11 and the demonstration is given in Figure 2.12.

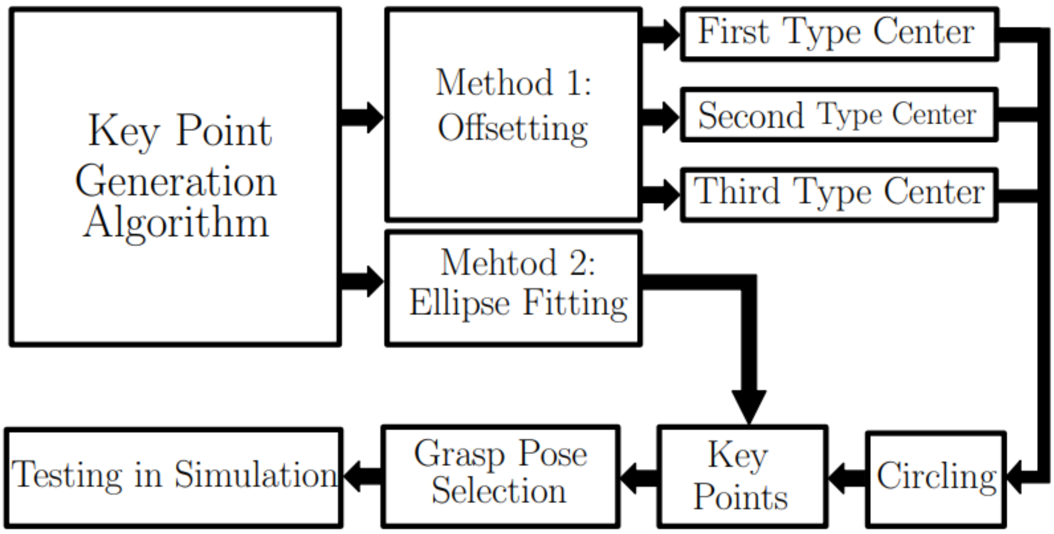


Figure 2.11: Block Diagram of the Geometric-Based Grasp Planner [6].

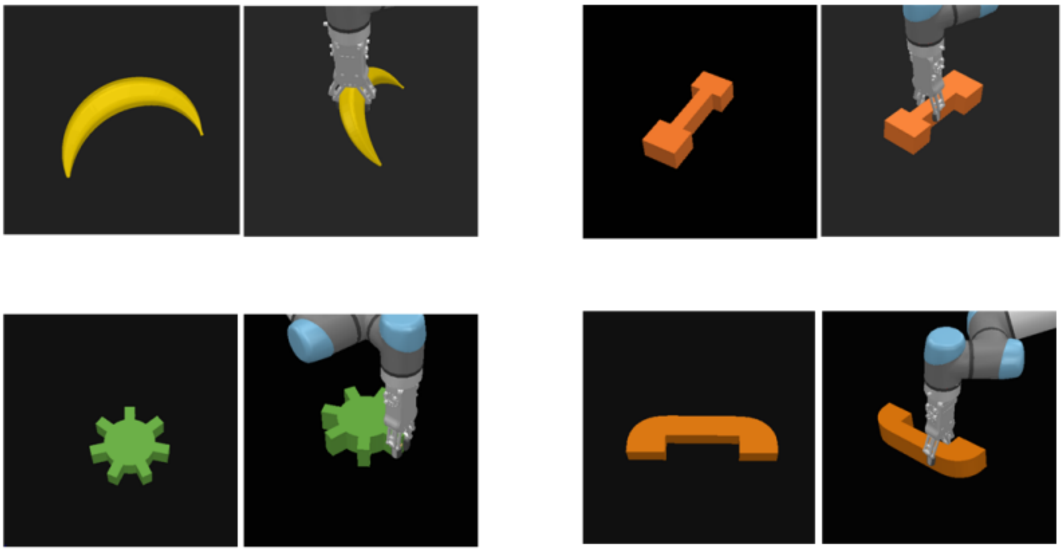


Figure 2.12: Successful grasps with geometric grasp planner from CoppeliaSim simulation [6].

2.2.1.C DL-based Methods

DL-based methods have revolutionized grasp planning in unstructured environments, enabling robots to autonomously grasp a wider range of objects in both structured and dynamic settings. Deep learning models are trained on large datasets of object images or point clouds, allowing them to generalize grasping strategies to unseen objects. These methods are computationally intensive but offer adaptability and high success rates in variable environments due to their data-driven nature.

One popular framework in this category is the DexNet, introduced by Mahler et al. [63], which uses a deep neural network trained on synthetic point clouds and grasp metrics to predict robust grasps across various object types. DexNet has proven particularly valuable in industrial applications where the variety of objects is extensive and handling unknown objects is necessary. Similarly, Levine et al. [64] developed a model for real-time robotic grasping by collecting a large-scale dataset which allows the model to adaptively plan grasps on objects of diverse shapes and textures.

Another significant contribution in DL-based methods is the 6-DOF GraspNet, which extends grasp prediction to three-dimensional space and calculates stable grasps even in cluttered environments [7]. This capability is illustrated in Figure 2.13 which shows an overview of the architecture of the 6-DOF GraspNet, and Figure 2.14 which demonstrates the successful application of this model in real-world manipulation tasks.

While these methods are highly effective in complex settings, their main drawbacks include the need for extensive training data, significant computational resources, and the inability to adapt without retraining when presented with novel object features outside of the trained dataset.

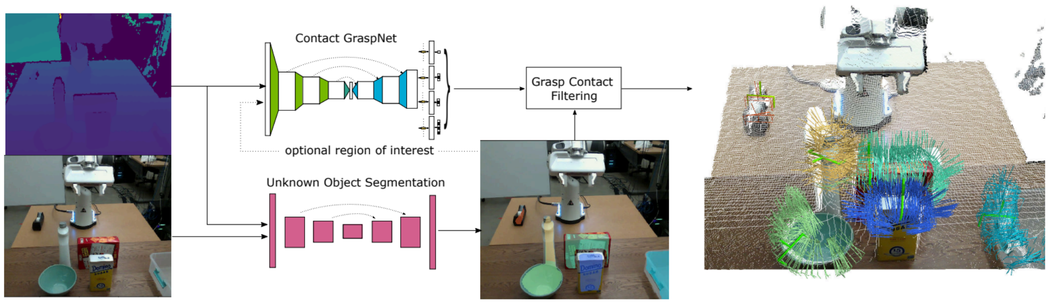


Figure 2.13: Overview of 6-DOF GraspNet Architecture for Robust Grasp Prediction [7].

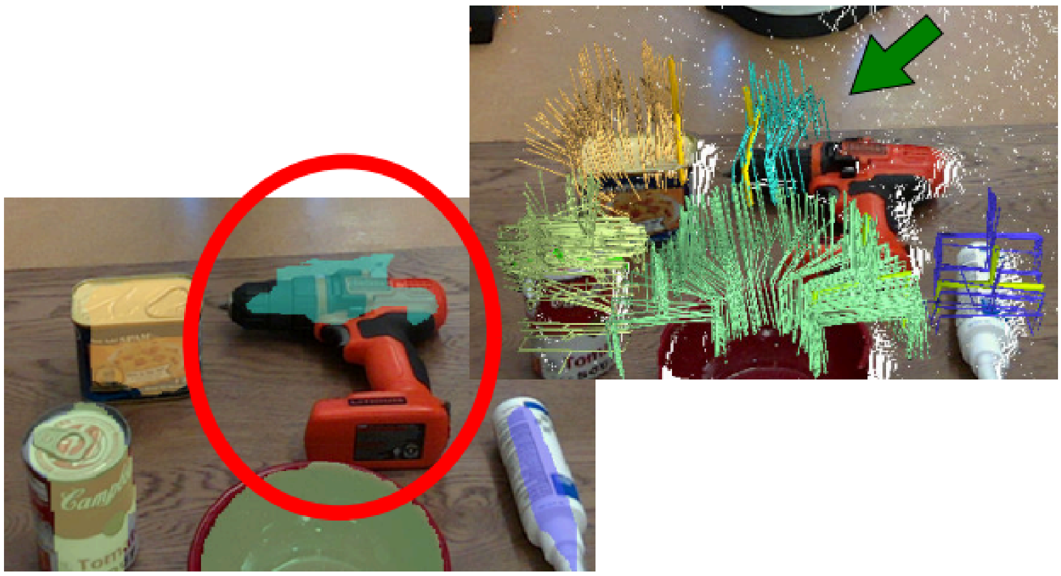


Figure 2.14: Example of successful grasps achieved by 6-DOF GraspNet on cluttered objects [7].

2.2.2 Merits of Geometric Grasp Planning

Geometric grasp planning offers multiple advantages in robotic manipulation:

- **Efficiency:** Geometric methods are computationally efficient since they rely on the direct analysis of the spatial properties of an object, making them suitable for real-time applications [65].
- **Simplicity and Generalizability:** Due to their reliance on geometric features, such as edges and shapes, these methods can be applied to a broad spectrum of objects without extensive data-driven training [65, 66].
- **Robustness in Known Object Classes:** Geometric approaches provide robust results in scenarios with limited object variation and where objects adhere closely to known shapes or models.
- **Adaptability to Limited Visibility:** Underwater environments are often characterized by low vis-

ibility, where it is easier to detect edges and shapes of objects than to detect finer details of the object. Geometric methods relying on shape-based features can be more effective than complex data-intensive models in such conditions, allowing for reliable performance despite environmental limitations.

These advantages make geometric grasp planning particularly useful in many robotics applications, where objects typically follow standard dimensions and shapes.

2.2.3 Further Review of Geometric Method: Image and Point Cloud Inputs

In underwater environments, grasp planning relies on effective segmentation and object recognition to provide reliable object interaction, especially given visibility challenges and dynamic conditions. Two primary input types for segmentation models used in underwater applications are RGB images [67] and point cloud data [68, 69], each offering unique benefits and limitations. While RGB-based methods rely on color and texture information for segmentation, point cloud-based methods leverage 3-D spatial data to model object shapes and contours, which is crucial in environments where visibility is impaired. Each approach can been applied to underwater manipulation in applications such as debris collection, pipe transportation, and repair operations [68, 70, 71].

2.2.3.A RGB-Based Geometric Grasp Planning

RGB-based geometric grasp planning leverages color, intensity, and texture information to generate grasp hypotheses based on the visible features of an object. This approach is particularly effective for underwater environments where objects possess distinct color characteristics that contrast with the surrounding environment.

- **Contour and Edge-Based Grasp Planning:** Geometric grasp planning based on RGB inputs often relies on edge and contour detection, where edges are used to approximate the shape and orientation of objects in the underwater scene. This approach is usually effective in identifying and grasping objects in shallow waters where color information is not too deteriorated.
- **Depth Augmented RGB Grasp Planning:** Depth-augmented RGB methods, where depth cues are overlaid onto RGB images, have shown improved performance for geometric grasp planning in underwater environments with limited visibility. Paul et al [72] and Yang et al [8] combined RGB image data with depth data for geometric grasping tasks, enabling precise manipulation of the target object. This RGB-based method improved grasp success by combining surface texture with inferred depth, accommodating environments with complex or poor lighting. Figure 2.15 gives the architecture of the Yang et al method and Figure 2.16 gives an example application.

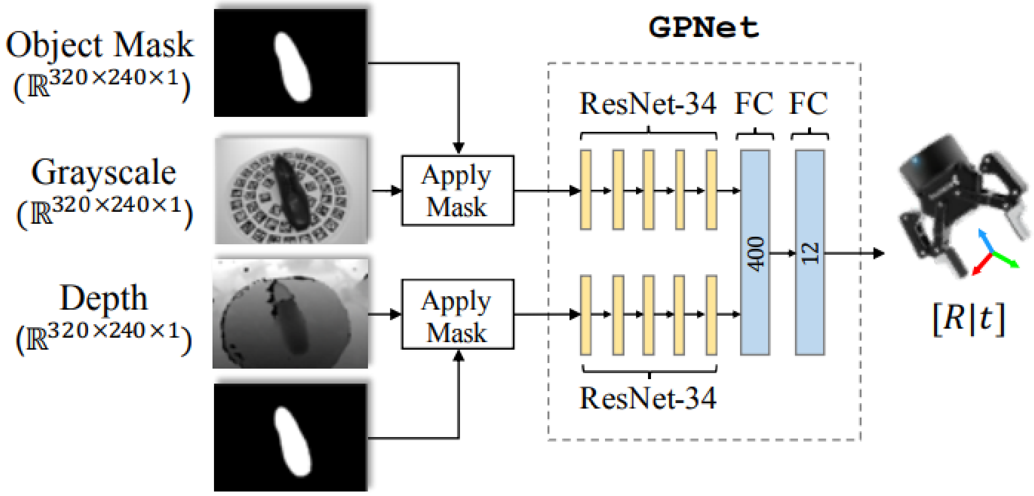


Figure 2.15: Overview of architecture of the depth-augmented RGB-based grasp planner proposed by Yang et al [8].

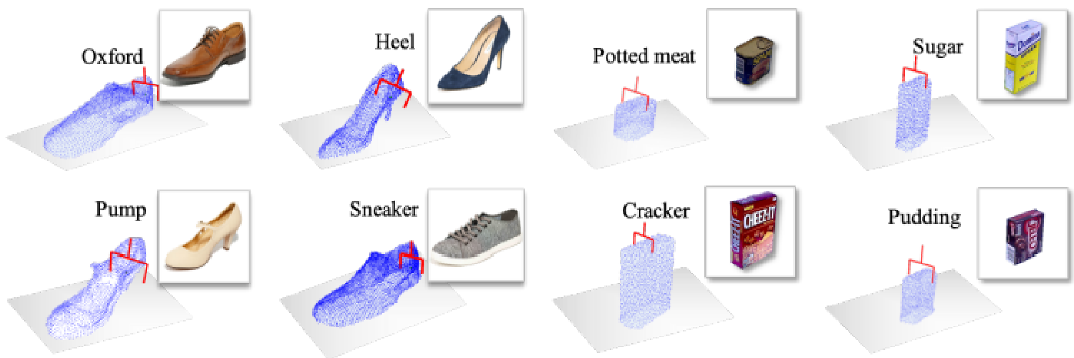


Figure 2.16: Example of successful grasps achieved by the depth-augmented RGB-based grasp planner [8].

Despite their advantages, RGB-based geometric methods are limited in highly turbid waters where visual cues degrade. Such limitations make them less suitable for dynamic underwater tasks or when objects lack sufficient contrast from the background, as encountered in deep-sea exploration or sediment-rich areas.

2.2.3.B Point Cloud-Based Geometric Grasp Planning

Point cloud-based geometric grasp planning uses 3-D spatial data derived from sonar or stereo camera systems to create detailed models of objects, allowing for accurate grasp point estimation [68] even in underwater environments with low visibility. This approach is particularly advantageous for underwater environments, where point cloud data can represent the shape and depth of objects in more details, which is crucial for grasping tasks [68, 73].

- **3-D Shape Fitting and Contour Analysis:** Point cloud-based geometric methods often involve fitting simple geometric shapes such as spheres or ellipsoids to approximate the contours of target objects. Monica et al [68] developed a point cloud-based geometric grasp planner for underwater robots, where ellipsoidal fittings allowed for the grasping of irregularly shaped marine objects, such as rocks and debris, in murky conditions. This method provided reliable grasp poses, significantly improving the stability of grasped objects in dynamic underwater currents. In this method, the best pose could be proposed to be at the centroid of the fitted 3-D shape.
- **Geometric Matching for Object Localization and Grasping:** Another point cloud-based geometric approach is to match predefined shapes to point clouds. Yu et al [69] proposed a grasp planning system that matches point cloud data from sonar with stored CAD models of objects, such as underwater tools or pipes. This method was effective for grasping tasks that required precise alignment, such as pipeline maintenance, as it enabled robots to determine optimal grasp points based on object geometry and orientation in 3-D space.
- **End-to-End DL Grasp Propositions:** An end-to-end grasp planning system using point clouds involves sequential steps that begin with data acquisition and preprocessing, using sonar or stereo cameras to capture and filter point cloud data for noise reduction and clarity. Segmentation and shape fitting follow, isolating and approximating the target object with simple geometric shapes (e.g., ellipsoids or cylinders) to facilitate accurate grasp pose estimation. After identifying the optimal grasp location, the system simulates grasp execution in a controlled environment to verify stability, followed by real-time grasping, often supported by feedback from force sensors to adapt to underwater currents and object resistance. This workflow has been proven effective for grasping tasks such as underwater debris retrieval and equipment handling, providing stability and accuracy in low-visibility conditions [70, 73]. Wang et al [9] has applied this method in simple table-top grasping demonstration. Their architecture (including a typical example) is described in Figure 2.17.

Point cloud-based geometric grasp planning has good applications in underwater scenarios requiring high precision, such as retrieval of irregular objects or manipulation of equipment in deep-sea maintenance tasks.

2.2.4 Metrics of Geometric Grasp Planning

Evaluating the effectiveness of geometric grasp planning requires robust metrics that quantify grasp stability, accuracy, and feasibility under various conditions. These metrics provide an essential framework for comparing algorithms and assessing grasp success in both controlled and dynamic environments. Some of the metrics generally considered are:

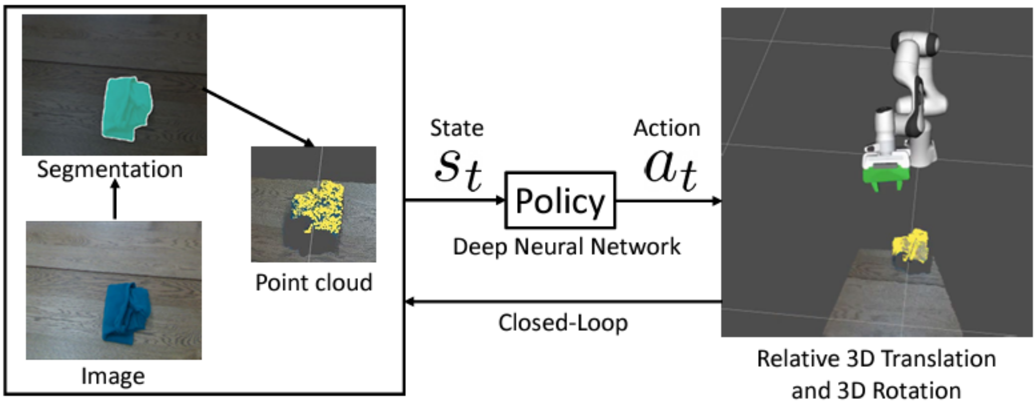


Figure 2.17: Overview of architecture of an end-to-end point cloud based grasp planner [9].

- **Grasp Stability:** Grasp stability is often evaluated by quantifying the resistance of an object to slippage or rotation under external forces. A common metric for stability is the Grasp Wrench Space (GWS), which represents the set of external forces and torques that the object can withstand while held by the robotic gripper. An optimal grasp maximizes the GWS, making it more resistant to perturbations. This approach has been widely adopted in grasp planning research, as seen in the work of Ferrari and Canny [74], where the GWS concept was applied to optimize grasping forces for diverse object geometries. In underwater settings, ensuring stability against external forces, such as currents, is critical, which requires integrating GWS analysis with real-time feedback from force sensors [75].

- **Contact Quality:** Contact quality evaluates how well the points of contact on an object contribute to a secure grasp. Metrics such as contact region stability and finger contact area help assess this quality. A broader and well-distributed contact area increases grasp reliability, particularly for irregularly shaped objects. For example, grasping a cubic-shaped object by its faces is more stable than grasping it by its edges, and grasping it by its edges is more stable than grasping it by its vertices. Song et al [65] highlighted that optimizing the contact points on a 3-D object improves grasp reliability, an approach that has been especially useful in underwater grasps where irregular marine objects often require maximized contact stability. This metric is crucial for applications requiring the secure manipulation of objects with complex geometries, such as coral samples or rock formations.

- **Success Rate of Grasp Execution:** The success rate quantifies the proportion of successful grasps in real-world or simulated trials, typically based on a large dataset or a series of test scenarios. For underwater applications, successful grasps can be defined as those where the object is securely held and transported without dislodgment due to environmental factors like turbulence. As

Yu et al. [69] demonstrated, a high success rate in grasp planning can be achieved by integrating grasp pose estimation with feedback mechanisms for real-time adjustments.

- **Computational Efficiency:** For real-time grasp planning applications, computational efficiency is essential, especially in underwater robotics where on-board processing power may be limited. Metrics such as grasp planning time and algorithmic complexity provide insights into the feasibility of implementing grasp algorithms on real-world systems. Monica et al. [68] discussed how geometric algorithms can reduce computational load, making them viable for low-power, on-board processing systems in underwater robotics.

Incorporating these metrics enables a comprehensive evaluation of geometric grasp planning approaches, helping to refine methods for specialized tasks like underwater manipulation. Each metric addresses a critical aspect of grasp reliability, from structural stability to efficiency, facilitating the development of robust algorithms that enhance robotic functionality in challenging environments.

3

Methodology

Contents

This chapter describes the methodology used in the development of the grasp planning algorithm for underwater applications. These methods incorporate some techniques for image pre-processing and segmentation, and point-cloud-based geometric grasp planning to enhance the efficiency and reliability of underwater object manipulation. The following sections provide a detailed account of each technique, elucidating their rationale, implementation, and some precedents in similar research works.

3.1 Overview of Existing Architecture

The general architecture of this project is structured into two core components: the intervention scenario and the algorithm implementation, as shown in Figure 3.1.

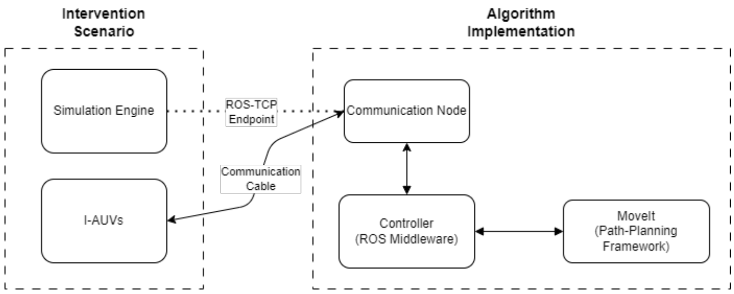


Figure 3.1: Schematic Diagram of the Project Overview

The intervention scenario provides an operational environment for I-AUVs to execute some specified intervention tasks. Initially, the scenario will exist as a simulation environment (described in Section 3.1.2). After rigorous testings, the scenario will transition to the experimental bed at CIRTESU for real-world validation. There exists a communication bridge between the scenario and the algorithm components for sensing and actuation.

The algorithm implementation component is responsible for processing the sensor data from the I-AUVs, generating the actuation commands and general coordination of the mission. It encompasses the mission planning, control and coordination of the I-AUVs. It also includes the Graphical User Interface (GUI). Details on the frameworks to be used in the implementation are provided in Section 3.1.3.

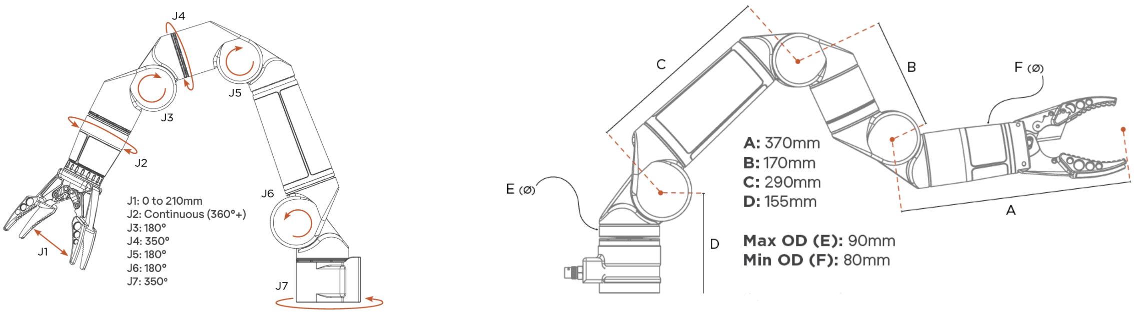
3.1.1 I-AUV for Experimental Testing

For the execution of the intervention tasks, two Girona500 AUVs which are each equipped with a 7DOF Reach Bravo manipulator could be used.

The Girona500 AUV is a compact-sized I-AUV, with size $1.5m \times 1m \times 1m$ and weight of about $140kg$, which can operate between 3 to 8 thrusters, giving the vehicle redundant DOFs [32, 76, 77]. It has three hulls that are arranged to provide the I-AUV with passive stability, as shown in Figure 3.2. The two upper hulls house the electronics and also contain floatation foams while the bottom hull contains heavier elements like batteries and manipulators (or other payloads) [32]. The Girona500 AUV is reconfigurable for different tasks. Hence, for the purpose of this project, the basic configuration which comprises navigation sensors (such as Doppler Velocity Log (DVL), Ultra-Short Baseline (USBL) and pressure gauge) will be used. In addition, the Reach Bravo manipulator will be coupled for intervention tasks.

Figure 3.2: Girona500 AUV without attached payload.

The Reach Bravo manipulator is a dextrous 7-DOF manipulator, weighing about $4.5kg$, specifically designed for aquatic inspection and intervention applications. It has six revolute joins (as shown in Figure 3.3(a)) and also offers a diverse array of mission-specific end-effectors which have accuracies less than $1cm$ and grasping forces up to $1000N$. A camera will be mounted on the end-effector which will enable the visual servoing technique to be used for the gripping process. This manipulator will be mounted on each of the Girona500 AUV to perform intervention tasks.



(a) Joints Description

(b) Default Configuration (with customization options)

Figure 3.3: Reach Bravo Manipulator [10]

In the experimental setting, two I-AUVs will be tasked with autonomous and cooperative identification and transportation of a long pipe to a designated location. Since the primary focus of this project is on developing and refining the planning and control algorithms, the communication interface is not a main concern. Hence, to facilitate real-time monitoring and management of the mission, a simple tethered cable will be employed to connect the I-AUVs to a command station. This approach ensures reliable data transfer while allowing the team to closely observe the mission and provide necessary interventions if required.

3.1.2 Simulation Environment

This project utilizes Unity Engine as its simulator engine. The Girona500 AUV and Reach Bravo have been modeled as rigid bodies and will be integrated into the simulation environment. Unity's recent advancements, particularly with the Unity Robotic Hub, have significantly enhanced manipulator and robotic simulations within the engine. Furthermore, Unity's capabilities extend to creating an HTTP server, enabling data transmission via TCP-IP and UDP protocols. To control the I-AUVs within the simulation, the ROS-TCP-Endpoint package serves as the bridge between the simulation environment and the ROS ecosystem.

3.1.3 Existing Algorithm Frameworks

There exist some frameworks that have implemented or under implementation for some aspects of this project. The MoveIt framework has package that could be adapted for path planning of the 7-DOF manipulator [78]. Some research team from other universities are actively working on an algorithm that could be employed for the Visual Segmentation of the desired object from the workspace and it can be integrated as a Robot Operating System (ROS) node [79]. Other aspects of the project would either be developed on existing packages from ROS or will be implemented as improvements on the existing source code for the TWINBOT project or Girona500 AUV architecture, such as Component Oriented Layer-based Architecture for Autonomy (COLA2) [32] which has existing motion planning, navigation and velocity control packages for some I-AUVs.

3.2 System Overview

This project forms a crucial segment of a larger initiative aimed at the planning and control of cooperative I-AUVs for underwater object transportation. As depicted in Figure 3.4, the overall system architecture integrates multiple techniques to ensure efficient and reliable operation in the complex underwater environments.

The workflow commences with the implementation of sophisticated image processing techniques tailored to detect desired objects within the workspace. These techniques are essential for accurately identifying target objects amidst the often cluttered and visually challenging underwater environment. By enhancing the reliability of object detection, the system ensures that subsequent processes are based on precise and accurate data.

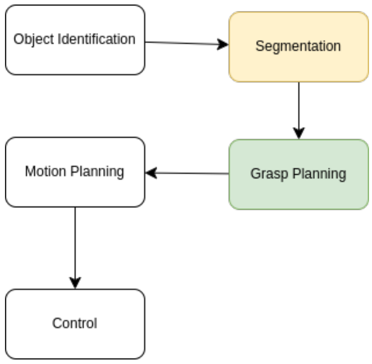


Figure 3.4: General System Overview.

Upon successful detection, the identified objects undergo a segmentation process designed to isolate them from the surrounding environment. This segmentation is crucial for extracting the object's precise geometry and spatial characteristics, which are vital for accurate manipulation. The segmented data is then converted into a standard CAD format or a point cloud format, facilitating seamless integration with various processing and planning algorithms.

The formatted data serves as the input for the grasp planning algorithm, which calculates the grasp points and orientations. This step is fundamental to ensure that the I-AUVs can securely and efficiently grasp the object, taking into consideration factors such as object geometry and material properties. The grasp planning algorithm leverages geometric models to determine the most stable and effective grasp configurations.

Following the determination of grasp points, a motion planning algorithm is employed to design a feasible trajectory for the I-AUVs. This algorithm will compute the optimal path from the current position to the target grasp points while avoiding obstacles and ensuring smooth and efficient movement. The trajectory planning takes into account the dynamic constraints of the I-AUVs and the underwater environment, ensuring that the planned path is both feasible and efficient.

Finally, a control algorithm is implemented to execute the planned grasp and manipulation tasks. This algorithm translates the planned trajectory into actionable control signals for the I-AUVs, ensuring precise and coordinated movements. The control algorithm continuously monitors the I-AUVs's state and the environment, making real-time adjustments to maintain stability and accuracy during the grasp and transportation processes.

In summary, this integrated system combines image processing, segmentation, grasp planning, motion planning, and control algorithms to enable the I-AUVs to effectively grasp and transport objects in underwater environments. Each component of the system plays a critical role in ensuring the overall robustness and efficiency of the operation, paving the way for sophisticated underwater manipulation tasks.

3.3 Pre-processing Techniques

In the context of this project, pre-processing techniques are essential to prepare the input data for further processing steps. These techniques are applied to the input CAD files and point cloud data to ensure they are in a suitable format and quality for the grasp planning algorithm.

For point clouds, two primary pre-processing techniques are employed: downsampling and voxelization. Downsampling reduces the number of points in the point cloud, which reduces computational load without significantly affecting the accuracy of the grasp planning. This is particularly useful in underwater environments where point cloud data can be dense and contain a lot of redundant information. The voxelization process, on the other hand, converts the point cloud into a set of volumetric pixels (called voxels), which provides a more manageable and structured representation of the 3-D data. This technique aids in reducing noise and reducing the computation overload of the subsequent steps.

For CAD models, the pre-processing involves converting the CAD files into point clouds. This step is crucial because it standardizes the input format, allowing the system to handle both CAD models and raw point cloud data uniformly. The conversion process typically involves sampling the surface of the CAD model to generate a dense point cloud representation, ensuring that the geometric details necessary for accurate grasp planning are preserved. After converting to point cloud data, the downsampling or voxelization approach maybe applied.

3.3.1 Downsampling and Voxelization

Downsampling a point cloud involves selecting a subset of points from the original point cloud to reduce its size while maintaining its overall structure. A common method for downsampling is to use a voxel grid filter. This technique divides the point cloud into a 3-D grid of voxels and replaces all the points within each voxel with a single point, usually the centroid of the points within the voxel.

Mathematically, if the point cloud P consists of N points $\{p_1, p_2, \ldots, p_N\}$ with coordinates (x_i, y_i, z_i), the voxel grid filter can be described as follows:

1. Define the voxel grid size $l \times w \times h$ where l, w and h represent the length, width and height of the voxel respectively.

2. For each point p_i, compute the voxel indexes in the x, y and z coordinates (v_x, v_y, v_z) as:

$$v_x = \left\lfloor \frac{x_i}{l} \right\rfloor, \quad v_y = \left\lfloor \frac{y_i}{w} \right\rfloor, \quad v_z = \left\lfloor \frac{z_i}{h} \right\rfloor$$

3. Aggregate points within the same voxel and compute the centroid $\bar{p}_v$ for each voxel V:

$$\bar{p}_v = \frac{1}{|V|} \sum_{p_i \in V} p_i$$

where V is the set of points within the voxel.

3.3.2 CAD to Point Cloud Conversion

Converting a CAD model to a point cloud typically involves sampling points on the surface of the model. If the CAD model is represented as a mesh with vertices $\{v_1, v_2, \ldots, v_M\}$ and faces $\{f_1, f_2, \ldots, f_K\}$, the conversion can be described as follows:

1. For each face f_k with vertices (v_{k1}, v_{k2}, v_{k3}), generate sample points $p_{k1}, p_{k2}, \ldots, p_{kn}$.
2. The sampling can be uniform or based on the area of the faces to ensure an even distribution of points.

3.4 Segmentation Techniques

Segmentation is a critical step in isolating the target object from the background and other elements in the environment. Two segmentation techniques are utilized in this project: plane segmentation and color segmentation.

3.4.1 Plane Segmentation

Plane segmentation is used to extract planar surfaces from the point cloud, assuming that the object is placed against a relatively simple background. This technique involves fitting a plane to the point cloud data and identifying points that lie on this plane. By segmenting out the planar background, the object of interest can be effectively isolated. This method is particularly useful in structured environments where the background can be approximated by a plane. Mathematically, plane segmentation can be achieved using the Random Sample Consensus (RANSAC) algorithm, which iteratively fits a plane model to subsets of the point cloud data and refines the fit by maximizing the number of inliers.

The equation of a plane in 3-D space is given by:

$$ax + by + cz + d = 0$$

where (a, b, c) is the normal vector of the plane and d is the distance from the origin.

RANSAC algorithm steps:

1. Specify the maximum number of iterations.

2. Define the set of inliers as an empty set.

3. Randomly select a subset of points from the point cloud.

4. Fit a plane to these points by solving the linear system:

$$\begin{bmatrix} x_1 & y_1 & z_1 & 1 \\ x_2 & y_2 & z_2 & 1 \\ x_3 & y_3 & z_3 & 1 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix} = 0$$

5. Count the number of inliers within a threshold distance ϵ from the plane:

$$|ax_i + by_i + cz_i + d| < \epsilon$$

6. Add these inliers into the set of inliers.

7. Repeat the last four steps till the defined amount of iteration is reached.

8. Select the set with the largest amount of inliers and compute its plane model.

In this project, the background is simply the walls of the pool and flowing water. With the plane segmentation approach, the wall and water considered as background can simply be segmented from the item to be grasped.

3.4.2 Color Segmentation

Color segmentation leverages the distinct color of the object to differentiate it from the rest of the environment. In this project, the pipe to be grasped is painted with a specific color, which simplifies the segmentation process. By applying color thresholding techniques, the system can identify and isolate points in the point cloud that match the specified color range. This approach is effective in scenarios where the object has a unique and easily distinguishable color.

Color segmentation relies on thresholding techniques in a chosen color space (e.g., RGB, HSV). For an object with a specific color, define the color range $[C_{min}, C_{max}]$. For each point p_i with color C_i:

$$C_{min} \leq C_i \leq C_{max}$$

Points falling within this range are segmented out. In the RGB color space, the segmentation process can be described as by applying the color threshold as follows:

$$R_{min} \leq R_i \leq R_{max}, \quad G_{min} \leq G_i \leq G_{max}, \quad B_{min} \leq B_i \leq B_{max}$$

In the context of this project, the pipe is painted yellow and the C_{min} and C_{max} are defined as $C_{min} = [192, 192, 0]$ and $C_{max} = [255, 255, 64]$ where the lower limit is 25% deduction from the pure yellow color and the upper limit is the pure yellow color with 25% variation tolerance of B channel from zero. Hence, the pipe can be segmented from the white-bluish background.

3.5 Point Cloud Based Geometric Grasp Planning

The grasp planning process involves several stages, each aimed at determining the optimal grasp points and orientations for the object. As illustrated in Figure 3.5, the workflow starts with the identification of the bounding box that encloses the target object. This bounding box provides a preliminary understanding of the object's dimensions and orientation.

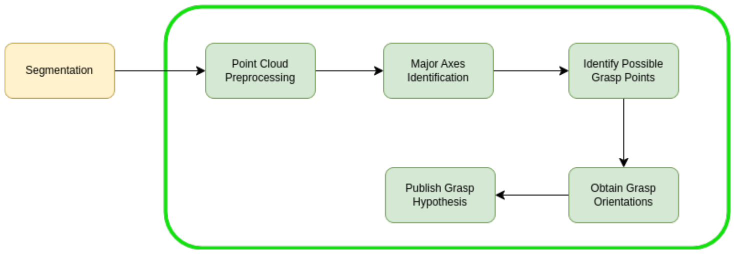


Figure 3.5: Overview of the Grasp Planning Approach.

Next, Principal Component Analysis (PCA) is applied to the point cloud data to identify the primary axes of the object. PCA reduces the dimensionality of the data while preserving the most significant features, allowing the system to determine the main geometric axes of the pipe. The longest axis of the pipe is identified as the primary axis, which guides the subsequent steps in the grasp planning process.

In determine the three principal axes of the pipe, we utilized the following steps:

1. Compute the centroid of the point cloud.

2. Computing the covariance matrix Σ:

$$\Sigma = \frac{1}{N} \sum_{i=1}^{N} (p_i - \bar{p})(p_i - \bar{p})^T$$

3. Eigen decomposition of Σ:

$$\Sigma v = \lambda v$$

   The eigenvectors v corresponding to the three largest eigenvalues λ represent the principal axes of the point cloud.

For a cylindrical object like a pipe, the grasp points are chosen perpendicular to the longest axis. Let v_1, v_2, v_3 be the principal axes from PCA, with v_1 being the longest axis. The grasp points are defined as positions p_g along v_1. Hence, the grasp propositions along the pipe is defined as:

$$p_g = \bar{p} + \alpha\vec{v_1} + \beta\vec{v_2} + \gamma\vec{v_3} \tag{3.1}$$

Where α, β and γ are simply scale values to define the location of the grasp. We uniformly spread the grasp along $\vec{v_1}$. Then, we use the 30 nearest neighbors to $\bar{p} + \alpha\vec{v_1}$ to determine the values of β and γ that ensures the grasp point is defined on the surface of the object. By ensuring that the grasps are perpendicular to principal axis, the system maximizes the stability and efficiency of the grasp.

The final grasp point is selected using a task-specific and end-effector dependent objective function. After the steps described above to give numerous amount of possible grasp positions, the object function would provide the best k grasps needed for the specified operation. For an operation requiring only one end-effector for grasping, the centrod would be returned as the best grasp. While another operation that uses multiple end-effector would require a different objective function to ensure that the grasp point are sufficiently away from each other to ensure appropriate grasp with minimal risk of collision.

4

Results and Discussions

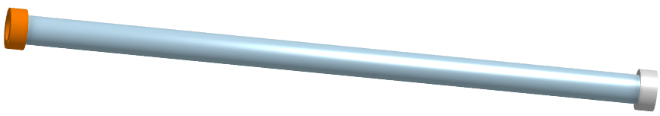
Contents

4.1 Grasp Pre-Processing

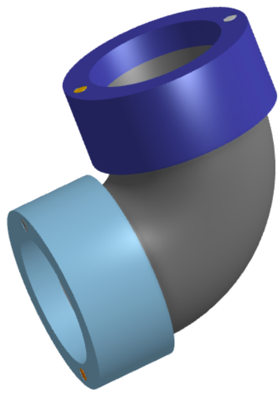
In the grasp pre-processing section, we discuss the results obtained from the conversion of the CAD model input to point clouds that are used for the grasp processing. This is one form of input that can be accepted into the grasp-planning algorithm. Afterwards, we discuss the scenario where the grasp-planning algorithm is implemented directly on point cloud inputs, which requires separating the desired object from its background. Here, we discuss the two simplified segmentation approaches we attempted: plane segmentation and color segmentation, along with their merits and shortcomings. In the next session, we will provide the results obtained from the grasp propositions for solid objects and point clouds.

Converting CAD files to Point Clouds

In this section, we present the results of converting CAD files to point clouds, a necessary step since our grasp planning algorithm accepts only point clouds as input. The conversion process involves sampling the CAD geometry to create point clouds that adequately represent the CAD model.



(a) Slender Pipe



(b) Pipe Connectors

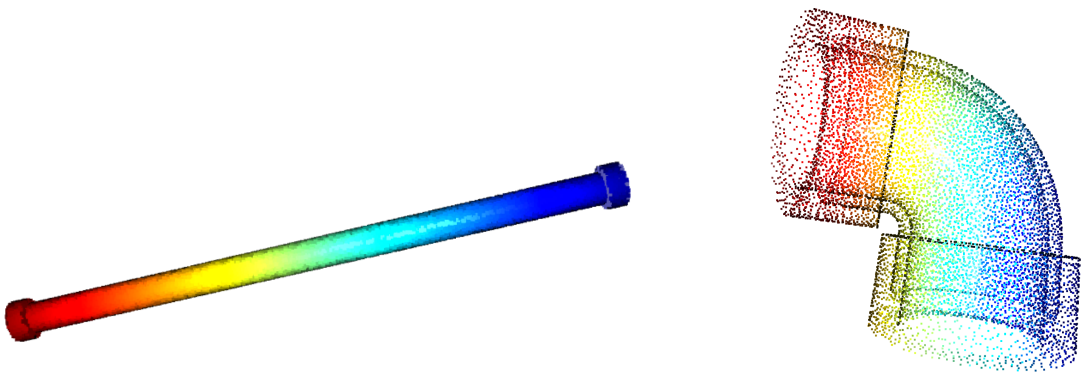
Figure 4.1: OnShape CAD files for the Pipe and Pipe Connector.

The original CAD models, as shown in Figure 4.1 were developed with OnShape software. The two objects initially considered for the research were a slender pipe (in Figure 4.1(a)) and a pipe connector (in Figure 4.1(b)). The conversion of the CAD files to point clouds were implemented using Open3D package and incorporated into the ROS package of the grasp planner.

A poisson distribution was used to sample the CAD model and the point cloud result obtained is presented in Figure 4.2. It is evident that most of the major salient features of the CAD model are lost. The reason is still unclear. However, upon introducing noise to the input, a better result as shown in Figure 4.3 was obtained. There is an hypothesis that perhaps this is becuase there are now many

(a) Slender Pipe **(b)** Pipe Connectors

Figure 4.2: Point Cloud of Pipe and Pipe Connector from CAD model (each point cloud has 10000 points).



(a) Slender Pipe **(b)** Pipe Connectors

Figure 4.3: Point Cloud of Pipe and Pipe Connector from CAD model with random noise.

different means and standard deviations in the noisy data but the cause is still unclear.

4.2 Segmentation

4.2.1 Plane Segmentation

Our grasp planning algorithm accepts both CAD models and direct point cloud images as inputs. However, point clouds obtained directly from sensors are not pre-segmented, necessitating the implementation of segmentation techniques to isolate the desired object from the background.

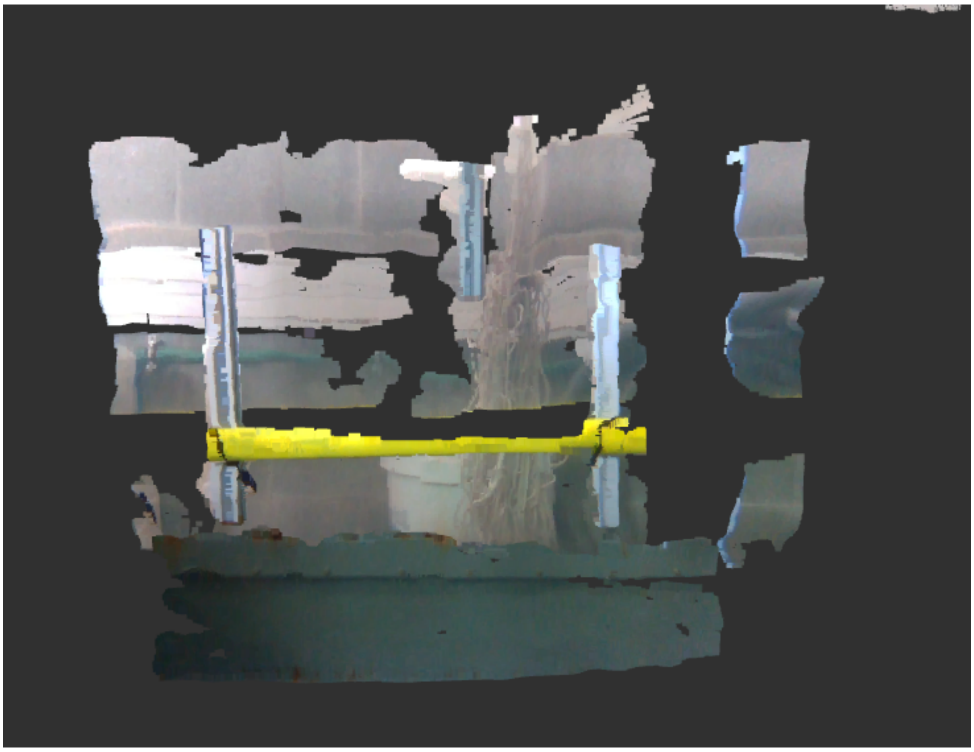


Figure 4.4: Image of the workspace (as point clouds) obtained from the Jetson Nano camera.

In our experimental setting, we placed a long slender pipe on a suspended bar and used a Jetson Nano (stereo) camera lowered on a submerssible into a pool to obtain a live video feed of the workspace. An image from the live video feed is provided in Figure 4.4. The cluttered background with water particles, blue wall and some rays of light from the top of the pool can be seen in the Figure. These made the image complicated to segment.

At first, we developed a plane segmentation algorithm to approximate the pipe as a place and thereafter we approximate the backgound as a plane both of these techniques were adequate for the separation of the pipe and the background in the air (i.e. in absence of water molecules). We assumed these should be an approximate for the underwater environment as the air molecules are negligible and

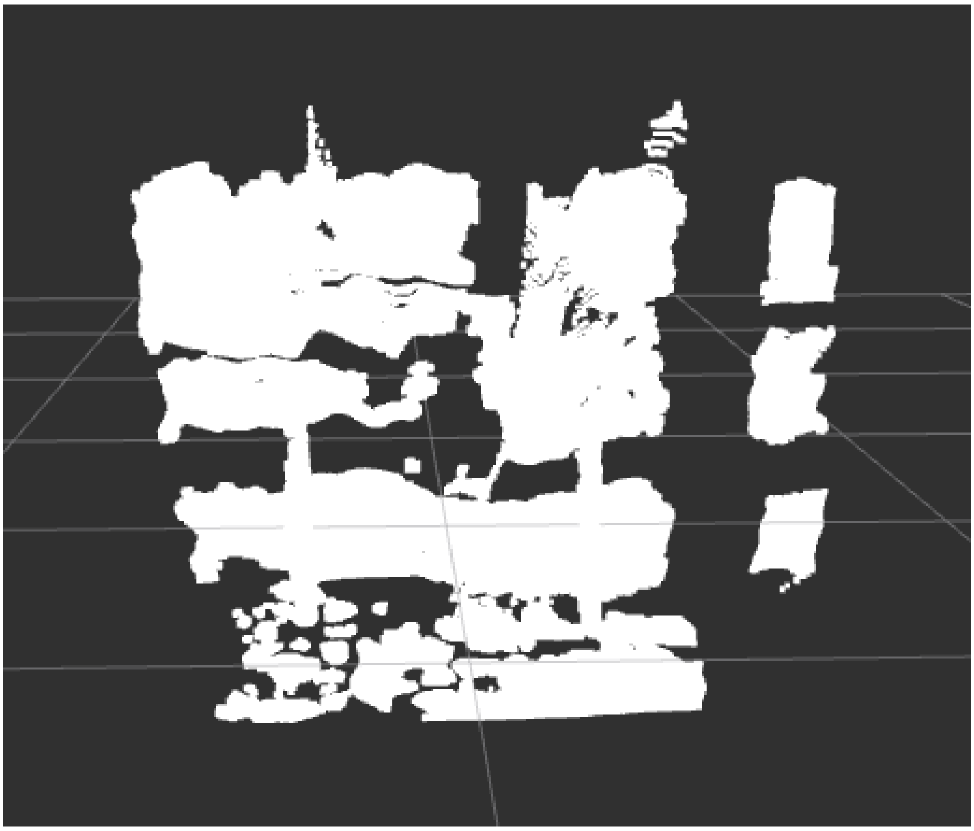


Figure 4.5: The result of segmenting a plane from the image. No plane was segmented because there are many candidate planes that could be segmented and the algorithm is designed to segment one.

almost not clearly visible in the images. However, when the algorithm was tested in the underwater environment, the result obtained is a shown in Figure 4.5. None of the two approaches worked in the underwater environment due to the presence of dynamic water molecules in the background and multiple plane surfaces detectable: the wall of the scene, the surface of the water and perhaps the forward facing region of the pipe.

Hence, another simplified approach is required for the segmentation.

4.2.2 Color Segmentation

Sequel to the failure of the plane segmentation technique, we resorted to painting the pipe with a specific and easily detectable color: yellow. Thereafter, we used the RGB channel of the image from the camera to segment the desired object from its background. To achieve this, we specific the upper and lower limits of the accepted colors in the workspace iteratively until a satisfactory result was obtained, as shown in Figure 4.6. While this method is not quite advanced, it was sufficient for the segmentation task. And the use of AI in this context is impossible because of the lack of data and it is not desired for deployment in autonomous underwater operations due to computational reasons.

In general, this technique should be highly effective in scenarios with significant color contrast, making it a valuable tool for underwater applications where objects can be isolated based on color properties. However, its effectiveness diminishes in low-light conditions or environments with low color contrast.

Figure 4.6: Result of color segmentation applied to the underwater scene, showing clear differentiation of the yellow pipe.

4.3 Grasp Proposals

In this section on grasp proposal, we provide the results obtained for the experimental demonstration of the grasp planning algorithm on both processed CAD files and real-time image data. It will be observed that only a long pipe is experimented on. This is because the broad groal of the project is to demonstrate the cooperative transportation of a slender pipe. An attempt was made to demonstrate the algorithm on the pipe connector as well. However, the connector was too small and complicated to be demonstrated on in the underwater environment, considering the available resources.

4.3.1 Grasp Proposals for Pipe CAD Model

In the case of the CAD models, obtaining the principal axes and determining the grasp propositions was straightforward. We assume that the object is uniform, solid, rigid and static. Then, we implemented the algorithm on the point cloud equivalence of the CAD model. The grasp propositions provided by our algorithm for this is provided in Figure 4.7.

The algorithm provided several grasp propositions perpendicular to the surface of the target object, highlighting the best grasp location around the centroid of the shape. This approach will ensure stable and effective manipulation of the pipe.

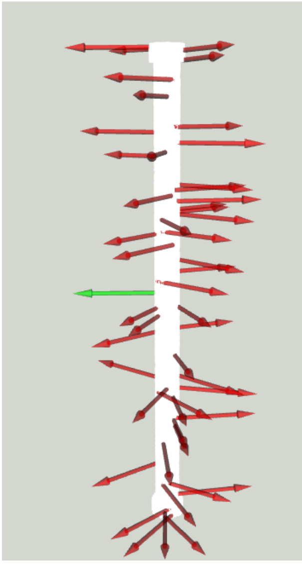


Figure 4.7: Grasp Proposition on CAD inputs.

4.3.2 Grasp Proposals for Direct Point Clouds from Underwater Images

After successfully applying the grasp planning algorithm on the CAD models, we implemented the algorithm as well on underwater images obtained from stereo cameras. The point cloud obtained from these cameras are more complicated to deal with as they are irregular and have several missing points, as earlier depicted in Figure 4.6.

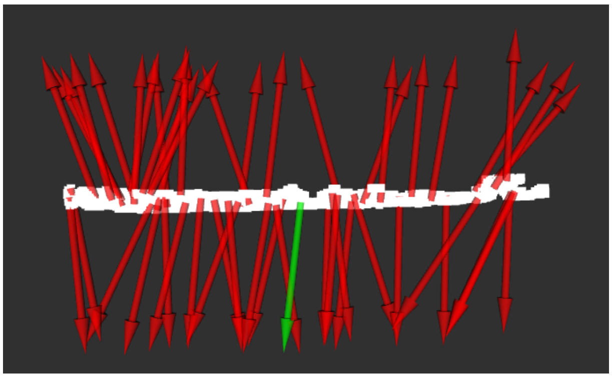


Figure 4.8: Grasp propositions on the segmented point clouds.

However, the developed grasp planning algorithm was able to propose adequate grasp positions and orientations that are feasible for the grasping and manipulation of the slender pipe, as shown in Figure 4.8. Figure 4.9 shows the propositions in the full scene of the underwater workspace from the robot arm frame.

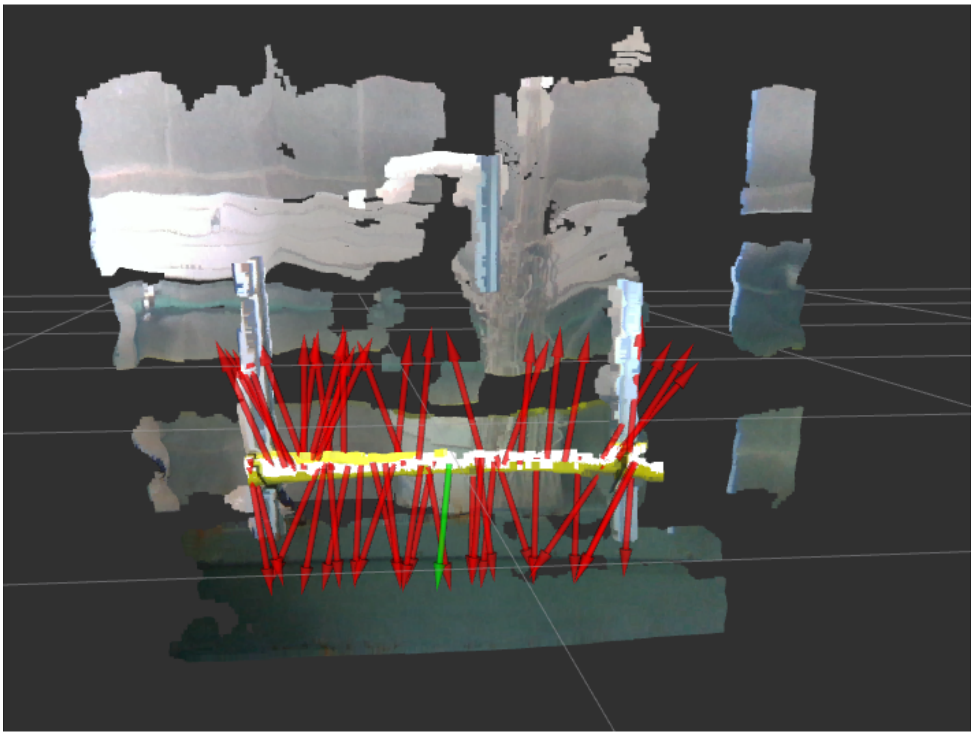


Figure 4.9: Grasp propositions in the full underwater scene.

4.4 Discussion

The results from converting CAD files to point clouds, plane and color segmentation techniques, and grasp propositions are reported and discussed in this chapter and they are the key aspects of our approach to grasp planning for underwater applications.

The conversion process from CAD models to point clouds preserved critical geometric details, essential for accurate grasp planning. Although, obtaining a desired point cloud data for grasping was problematic, the introduction of noise to the point clouds helps to obtain the desired point cloud, and ensuring that the system is ready for practical deployment. This step proved crucial as it enabled the algorithm to handle variability in the data, which is typical in underwater environments.

While plane segmentation was initially considered a viable method, it became clear that underwater conditions posed significant challenges. The presence of moving water molecules and multiple plane surfaces led to the failure of this technique. This outcome underscores the complexities of underwater imaging and the need for adequate segmentation methods capable of dealing with such environments.

Color segmentation, on the other hand, provided a more reliable solution. By painting the pipe a distinct color and using RGB-based segmentation, we achieved clearer differentiation between the object and the background. This method, although simplified, proved effective and computationally feasible for real-time applications. Its success highlights the importance of leveraging color contrast in underwater environments where traditional segmentation methods may fail.

The grasp proposals generated for both CAD-derived and direct underwater point clouds demonstrated the robustness and adaptability of our geometric algorithm. Handling noise and variability in the data is crucial for ensuring reliable grasping in dynamic underwater environments. The algorithm's ability to propose stable and effective grasp points, even in the presence of data irregularities, confirms its potential for practical applications.

In general, the methods and algorithms developed in this study provide a solid foundation for effective and efficient grasp planning in underwater applications. They address key challenges and demonstrate significant potential for real-world deployment, emphasizing the importance of applicability and adaptability in complex and variable environments.

5

Conclusion and Future Works

Contents

5.1 Conclusions

This research project has made significant contributions to the development of a semi-autonomous grasp planning algorithm tailored for underwater applications, thereby enhancing the intervention capabilities of I-AUVs. The algorithm employs downsampling and voxelization techniques to preprocess point cloud data, effectively reducing the computational overhead associated with grasp planning.

Initially, we explored a plane segmentation technique aimed at removing the underwater background to isolate the intended object. However, this method proved impossible due to the dynamics of water molecules and the presence of multiple planar surfaces in the captured point cloud data. As a result, we implemented a simple color segmentation algorithm that separates the object from the background using some thresholding techniques. This approach proved effective, provided that no objects in the background possessed similar colors to the target.

Subsequently, we developed a grasp planning algorithm that utilizes Principal Component Analysis (PCA) to identify the three principal axes of the object. Grasp positions were estimated by computing the normals on the object surface and normal to the major principal axis. The proposed grasps were shown to be adequate for effective manipulation and also yielding promising results that can be incorporated in future cooperative grasp planning missions.

5.2 System Limitations and Future Work

The limitations of the current system are as follows:

- The system is not suitable for flexible objects.
- The system may not perform optimally for objects with uneven mass distribution. However, it can be effectively utilized if the mass distribution is relatively even or if the center of gravity is close to the centroid (i.e., center of volume).
- The algorithm relies on other works for adequate image processing and segmentation, and for its eventual deployment for manipulation.

Future research directions would focus on the following areas:

- Developing more sophisticated grasp planning algorithms that can account for dynamic object properties and environmental factors, such as variations in mass distribution and the effects of water currents on manipulation.
- Investigating optimal poses for task-specific grasp planning such as wrenching, transporting, etc.

- Integrating the algorithm into a cooperative transportation scheme to demonstrate its applicability in intervention activities.

By addressing these future research directions, the capabilities of I-AUVs can be significantly enhanced, paving the way for more autonomous and efficient underwater operations across various industries.

Bibliography

[1] L. Chen, Y. Huang, J. Dong, Q. Xu, S. Kwong, H. Lu, H. Lu, and C. Li, "Underwater Object Detection in the Era of Artificial Intelligence: Current, Challenge, and Future," Oct. 2024, arXiv:2410.05577 [cs]. [Online]. Available: http://arxiv.org/abs/2410.05577

[2] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," 2013, version Number: 5. [Online]. Available: https://arxiv.org/abs/1311.2524

[3] J. Redmon, "You only look once: Unified, real-time object detection." in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016. [Online]. Available: https://www.cv-foundation.org/openaccess/content_cvpr_2016/papers/Redmon_You_Only_Look_CVPR_2016_paper.pdf

[4] S. Xu, M. Zhang, W. Song, H. Mei, Q. He, and A. Liotta, "A systematic review and analysis of deep learning-based underwater object detection," *Neurocomputing*, vol. 527, pp. 204–232, Mar. 2023. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0925231223000656

[5] N. Vahrenkamp, L. Westkamp, N. Yamanobe, E. E. Aksoy, and T. Asfour, "Part-based grasp planning for familiar objects," in *2016 IEEE-RAS 16th International Conference on Humanoid Robots (Humanoids)*. Cancun, Mexico: IEEE, Nov. 2016, pp. 919–925. [Online]. Available: http://ieeexplore.ieee.org/document/7803382/

[6] A. Akbari, A. Akbari, and M. T. Masouleh, "A 2D Geometry Based Grasping Pose Generation Algorithm for a Two-Finger Robot Hand," in *2023 31st International Conference on Electrical Engineering (ICEE)*. Tehran, Iran, Islamic Republic of: IEEE, May 2023, pp. 85–91. [Online]. Available: https://ieeexplore.ieee.org/document/10334804/

[7] M. Sundermeyer, A. Mousavian, R. Triebel, and D. Fox, "Contact-GraspNet: Efficient 6-DoF Grasp Generation in Cluttered Scenes," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. Xi'an, China: IEEE, May 2021, pp. 13 438–13 444. [Online]. Available: https://ieeexplore.ieee.org/document/9561877/

[8] D. Yang, T. Tosun, B. Eisner, V. Isler, and D. Lee, “Robotic Grasping through Combined Image-Based Grasp Proposal and 3D Reconstruction,” in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. Xi’an, China: IEEE, May 2021, pp. 6350–6356. [Online]. Available: https://ieeexplore.ieee.org/document/9562046/

[9] L. Wang, Y. Xiang, W. Yang, A. Mousavian, and D. Fox, “Goal-Auxiliary Actor-Critic for 6D Robotic Grasping with Point Clouds,” 2020, publisher: arXiv Version Number: 4. [Online]. Available: https://arxiv.org/abs/2010.00824

[10] “BRAVO 7: A Tough 7-Function Manipulator for Inspection Class Vehicles.” [Online]. Available: https://reachrobotics.com/media/docs/40983/Bravo-7-Datasheet.pdf

[11] A. A. Dalhatu, A. M. Sa’ad, R. Cabral De Azevedo, and G. De Tomi, “Remotely Operated Vehicle Taxonomy and Emerging Methods of Inspection, Maintenance, and Repair Operations: An Overview and Outlook,” *Journal of Offshore Mechanics and Arctic Engineering*, vol. 145, no. 2, p. 020801, Apr. 2023. [Online]. Available: https://asmedigitalcollection.asme.org/offshoremechanics/article/145/2/020801/1145947/Remotely-Operated-Vehicle-Taxonomy-and-Emerging

[12] Y. R. Petillot, G. Antonelli, G. Casalino, and F. Ferreira, “Underwater Robots: From Remotely Operated Vehicles to Intervention-Autonomous Underwater Vehicles,” *IEEE Robotics & Automation Magazine*, vol. 26, no. 2, pp. 94–101, Jun. 2019. [Online]. Available: https://ieeexplore.ieee.org/document/8706541/

[13] A. Sahoo, S. K. Dwivedy, and P. Robi, “Advancements in the field of autonomous underwater vehicle,” *Ocean Engineering*, vol. 181, pp. 145–160, Jun. 2019. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0029801819301623

[14] M. Fun Sang Cepeda, M. D. S. Freitas Machado, F. H. Sousa Barbosa, D. Santana Souza Moreira, M. J. Legaz Almansa, M. I. Lourenço De Souza, and J.-D. Caprace, “Exploring Autonomous and Remotely Operated Vehicles in Offshore Structure Inspections,” *Journal of Marine Science and Engineering*, vol. 11, no. 11, p. 2172, Nov. 2023. [Online]. Available: https://www.mdpi.com/2077-1312/11/11/2172

[15] G. Liu, “Remotely Operated Vehicle (ROV) in Subsea Engineering,” in *Encyclopedia of Ocean Engineering*, W. Cui, S. Fu, and Z. Hu, Eds. Singapore: Springer Nature Singapore, 2022, pp. 1466–1478. [Online]. Available: https://link.springer.com/10.1007/978-981-10-6946-8_225

[16] J. A. Ramírez-Macías, R. E. Vásquez, A. J. Sørensen, and S. Sævik, “A Methodology for DP Capability Studies on Remotely Operated Vehicles,” in *Volume 7A: Ocean Engineering*. Trondheim, Norway: American Society of Mechanical Engineers, Jun. 2017, p. V07AT06A041.

[Online]. Available: https://asmedigitalcollection.asme.org/OMAE/proceedings/OMAE2017/57731/Trondheim,%20Norway/281740

[17] J. Yuh, “Design and Control of Autonomous Underwater Robots: A Survey,” *Autonomous Robots*, vol. 8, no. 1, pp. 7–24, 2000. [Online]. Available: http://link.springer.com/10.1023/A:1008984701078

[18] M. Grasmueck, G. P. Eberli, D. A. Viggiano, T. Correa, G. Rathwell, and J. Luo, “Autonomous underwater vehicle (AUV) mapping reveals coral mound distribution, morphology, and oceanography in deep water of the Straits of Florida,” *Geophysical Research Letters*, vol. 33, no. 23, p. 2006GL027734, Dec. 2006. [Online]. Available: https://agupubs.onlinelibrary.wiley.com/doi/10.1029/2006GL027734

[19] V. A. I. Huvenne, P. A. Tyler, D. G. Masson, E. H. Fisher, C. Hauton, V. Hühnerbach, T. P. Le Bas, and G. A. Wolff, “A Picture on the Wall: Innovative Mapping Reveals Cold-Water Coral Refuge in Submarine Canyon,” *PLoS ONE*, vol. 6, no. 12, p. e28755, Dec. 2011. [Online]. Available: https://dx.plos.org/10.1371/journal.pone.0028755

[20] Z. Li, Y. Wang, W. Yang, and Y. Ji, “Development Status and Key Navigation Technology Analysis of Autonomous Underwater Vehicles,” in *2020 3rd International Conference on Unmanned Systems (ICUS)*. Harbin, China: IEEE, Nov. 2020, pp. 1130–1133. [Online]. Available: https://ieeexplore.ieee.org/document/9274926/

[21] P. Xu, J. Zheng, X. Wang, S. Wang, J. Liu, X. Liu, G. Xie, J. Tao, and M. Xu, “Design and Implementation of Lightweight AUV With Multisensor Aided for Underwater Intervention Tasks,” *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 69, no. 12, pp. 5009–5013, Dec. 2022. [Online]. Available: https://ieeexplore.ieee.org/document/9837304/

[22] B. Liu, “Recent Advancements in Autonomous Robots and Their Technical Analysis,” *Mathematical Problems in Engineering*, vol. 2021, pp. 1–12, Feb. 2021. [Online]. Available: https://www.hindawi.com/journals/mpe/2021/6634773/

[23] D. Kim, H.-S. Choi, J.-Y. Kim, J.-H. Park, and N.-H. Tran, “Trajectory generation and sliding-mode controller design of an underwater vehicle-manipulator system with redundancy,” *International Journal of Precision Engineering and Manufacturing*, vol. 16, no. 7, pp. 1561–1570, Jun. 2015. [Online]. Available: http://link.springer.com/10.1007/s12541-015-0206-y

[24] R. Pi, P. Cieslak, P. Ridao, and P. J. Sanz, “TWINBOT: Autonomous Underwater Cooperative Transportation,” *IEEE Access*, vol. 9, pp. 37 668–37 684, 2021. [Online]. Available: https://ieeexplore.ieee.org/document/9367135/

[25] Y. Wang, S. Wang, Q. Wei, M. Tan, C. Zhou, and J. Yu, "Development of an Underwater Manipulator and Its Free-Floating Autonomous Operation," *IEEE/ASME Transactions on Mechatronics*, vol. 21, no. 2, pp. 815–824, Apr. 2016. [Online]. Available: http://ieeexplore.ieee.org/document/7303960/

[26] E. Simetti, F. Wanderlingh, S. Torelli, M. Bibuli, A. Odetti, G. Bruzzone, D. L. Rizzini, J. Aleotti, G. Palli, L. Moriello, and U. Scarcia, "Autonomous Underwater Intervention: Experimental Results of the MARIS Project," *IEEE Journal of Oceanic Engineering*, vol. 43, no. 3, pp. 620–639, Jul. 2018. [Online]. Available: https://ieeexplore.ieee.org/document/8027199/

[27] E. Simetti and G. Casalino, "Manipulation and Transportation With Cooperative Underwater Vehicle Manipulator Systems," *IEEE Journal of Oceanic Engineering*, vol. 42, no. 4, pp. 782–799, Oct. 2017. [Online]. Available: http://ieeexplore.ieee.org/document/7790865/

[28] S. Heshmati-Alamdari, C. P. Bechlioulis, G. C. Karras, and K. J. Kyriakopoulos, "Cooperative Impedance Control for Multiple Underwater Vehicle Manipulator Systems Under Lean Communication," *IEEE Journal of Oceanic Engineering*, vol. 46, no. 2, pp. 447–465, Apr. 2021. [Online]. Available: https://ieeexplore.ieee.org/document/9118878/

[29] G. Marani, S. K. Choi, and J. Yuh, "Underwater autonomous manipulation for intervention missions AUVs," *Ocean Engineering*, vol. 36, no. 1, pp. 15–23, Jan. 2009. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S002980180800173X

[30] C. A. Simpkins, "Introduction to Autonomous Manipulation: Case Study with an Underwater Robot, SAUVIM [On the Shelf]," *IEEE Robotics & Automation Magazine*, vol. 21, no. 4, pp. 109–110, Dec. 2014. [Online]. Available: http://ieeexplore.ieee.org/document/6991031/

[31] G. Casalino, M. Caccia, S. Caselli, C. Melchiorri, G. Antonelli, A. Caiti, G. Indiveri, G. Cannata, E. Simetti, S. Torelli, A. Sperindè, F. Wanderlingh, G. Muscolo, M. Bibuli, G. Bruzzone, E. Zereik, A. Odetti, E. Spirandelli, A. Ranieri, J. Aleotti, D. Lodi Rizzini, F. Oleari, F. Kallasi, G. Palli, U. Scarcia, L. Moriello, and E. Cataldi, "Underwater Intervention Robotics: An Outline of the Italian National Project MARIS," *Marine Technology Society Journal*, vol. 50, no. 4, pp. 98–107, Jul. 2016. [Online]. Available: https://www.ingentaconnect.com/content/10.4031/MTSJ.50.4.7

[32] P. Ridao, M. Carreras, D. Ribas, P. J. Sanz, and G. Oliver, "Intervention AUVs: The Next Challenge," *IFAC Proceedings Volumes*, vol. 47, no. 3, pp. 12 146–12 159, 2014. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S1474667016435494

[33] Z. Zou, K. Chen, Z. Shi, Y. Guo, and J. Ye, "Object Detection in 20 Years: A Survey," *Proceedings of the IEEE*, vol. 111, no. 3, pp. 257–276, Mar. 2023. [Online]. Available: https://ieeexplore.ieee.org/document/10028728/

[34] E. Karami, S. Prasad, and M. Shehata, “Image Matching Using SIFT, SURF, BRIEF and ORB: Performance Comparison for Distorted Images,” 2017, version Number: 1. [Online]. Available: https://arxiv.org/abs/1710.02726

[35] A. Gupta and R. Sharma, “Underwater Image Enhancement Using Hog and Sift Method,” *International Journal of Electronics Engineering*, vol. 11, no. 2, pp. 314–321, 2019.

[36] M. S, A. L, A. S. L, A. S. L, and S. Rangaswamy, “Underwater Mine Detection Using Histogram of oriented gradients and Canny Edge Detector,” in *2021 International Carnahan Conference on Security Technology (ICCST)*. Hatfield, United Kingdom: IEEE, Oct. 2021, pp. 1–6. [Online]. Available: https://ieeexplore.ieee.org/document/9717404/

[37] M. Wang, X. Liu, Z. Chen, and E. Lorenzini, “Sea Cucumber Identification Fishing System Based on Hog Super Pixel Feature Extraction,” *Journal of Physics: Conference Series*, vol. 2560, no. 1, p. 012012, Aug. 2023. [Online]. Available: https://iopscience.iop.org/article/10.1088/1742-6596/2560/1/012012

[38] K. Demi˙R and O. Yaman, “A HOG Feature Extractor and KNN-Based Method for Underwater Image Classification,” *Firat University Journal of Experimental and Computational Engineering*, vol. 3, no. 1, pp. 1–10, Feb. 2024. [Online]. Available: http://dergipark.org.tr/en/doi/10.62520/fujece.1443818

[39] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” *Advances in neural information processing systems*, vol. 25, 2012.

[40] X. Ma, L. Chen, Z. Deng, P. Xu, Q. Yan, K.-S. Choi, and S. Wang, “Deep Image Feature Learning With Fuzzy Rules,” *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 8, no. 1, pp. 724–737, Feb. 2024. [Online]. Available: https://ieeexplore.ieee.org/document/10091676/

[41] S. Kan, Y. Cen, Z. He, Z. Zhang, L. Zhang, and Y. Wang, “Supervised Deep Feature Embedding With Handcrafted Feature,” *IEEE Transactions on Image Processing*, vol. 28, no. 12, pp. 5809–5823, Dec. 2019. [Online]. Available: https://ieeexplore.ieee.org/document/8651322/

[42] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, “Focal Loss for Dense Object Detection,” Feb. 2018, arXiv:1708.02002 [cs]. [Online]. Available: http://arxiv.org/abs/1708.02002

[43] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, “SSD: Single Shot MultiBox Detector,” 2016, vol. 9905, pp. 21–37, arXiv:1512.02325 [cs]. [Online]. Available: http://arxiv.org/abs/1512.02325

[44] P. Sarkar, D. Sourav, and S. Gurung, “A survey on underwater object detection.” in *Intelligence Enabled Research: DoSIER 2021*. Singapore: Springer Singapore, 2022, pp. 91–104.

[45] L. Zhang, X. Yang, Z. Liu, L. Qi, H. Zhou, and C. Chiu, "Single Shot Feature Aggregation Network for Underwater Object Detection," in *2018 24th International Conference on Pattern Recognition (ICPR)*. Beijing: IEEE, Aug. 2018, pp. 1906–1911. [Online]. Available: https://ieeexplore.ieee.org/document/8545677/

[46] A. Mahmood, M. Bennamoun, S. An, F. A. Sohel, F. Boussaid, R. Hovey, G. A. Kendrick, and R. B. Fisher, "Deep Image Representations for Coral Image Classification," *IEEE Journal of Oceanic Engineering*, vol. 44, no. 1, pp. 121–131, Jan. 2019. [Online]. Available: https://ieeexplore.ieee.org/document/8255621/

[47] L. Chen, Z. Liu, L. Tong, Z. Jiang, S. Wang, J. Dong, and H. Zhou, "Underwater object detection using Invert Multi-Class Adaboost with deep learning," May 2020, arXiv:2005.11552 [cs]. [Online]. Available: http://arxiv.org/abs/2005.11552

[48] Y. Wang, Y. Ji, D. Liu, Y. Tamura, H. Tsuchiya, A. Yamashita, and H. Asama, "ACMarker: Acoustic Camera-Based Fiducial Marker System in Underwater Environment," *IEEE Robotics and Automation Letters*, vol. 5, no. 4, pp. 5018–5025, Oct. 2020. [Online]. Available: https://ieeexplore.ieee.org/document/9127781/

[49] S. Minaee, Y. Y. Boykov, F. Porikli, A. J. Plaza, N. Kehtarnavaz, and D. Terzopoulos, "Image Segmentation Using Deep Learning: A Survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–1, 2021. [Online]. Available: https://ieeexplore.ieee.org/document/9356353/

[50] Peizhe Zhang and Canping Li, "Region-based color image segmentation of fishes with complex background in water," in *2011 IEEE International Conference on Computer Science and Automation Engineering*. Shanghai, China: IEEE, Jun. 2011, pp. 596–600. [Online]. Available: http://ieeexplore.ieee.org/document/5953291/

[51] J. Li, P. Jiang, and H. Zhu, "A Local Region-Based Level Set Method With Markov Random Field for Side-Scan Sonar Image Multi-Level Segmentation," *IEEE Sensors Journal*, vol. 21, no. 1, pp. 510–519, Jan. 2021. [Online]. Available: https://ieeexplore.ieee.org/document/9154438/

[52] Z. Chen, Y. Wang, W. Tian, J. Liu, Y. Zhou, and J. Shen, "Underwater sonar image segmentation combining pixel-level and region-level information," *Computers and Electrical Engineering*, vol. 100, p. 107853, May 2022. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0045790622001458

[53] R. Priyadharsini and T. S. Sharmila, "Object Detection In Underwater Acoustic Images Using Edge Based Segmentation Method," *Procedia Computer Science*, vol. 165, pp. 759–765, 2019. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S1877050920300235

[54] A. Setiawan, W. A. Triyanto, A. Setiawan, B. Warsito, and A. Wibowo, "Underwater Shrimp Digital Image Segmentation Using Edge Detection Method on Fog Network," in *2022 2nd International Conference on Information Technology and Education (ICIT&E)*. Malang, Indonesia: IEEE, Jan. 2022, pp. 81–86. [Online]. Available: https://ieeexplore.ieee.org/document/9759876/

[55] E. Afreen Awalludin, W. N. J. H. Wan Yussof, Z. Bachok, M. A. Firdaus Aminudin, M. S. Che Din, and M. Suzuri Hitam, "Monitoring climate change effects on coral reefs using edge-based image segmentation," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 14, no. 1, p. 398, Feb. 2024. [Online]. Available: https://ijece.iaescore.com/index.php/IJECE/article/view/32523

[56] R. Szeliski, *Computer vision: algorithms and applications*. Springer Nature, 2022.

[57] P. Drews-Jr, I. D. Souza, I. P. Maurell, E. V. Protas, and S. S. C. Botelho, "Underwater image segmentation in the wild using deep learning," *Journal of the Brazilian Computer Society*, vol. 27, no. 1, p. 12, Dec. 2021. [Online]. Available: https://journal-bcs.springeropen.com/articles/10.1186/s13173-021-00117-7

[58] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo, P. Dollár, and R. Girshick, "Segment Anything," 2023, version Number: 1. [Online]. Available: https://arxiv.org/abs/2304.02643

[59] S. Lian, Z. Zhang, H. Li, W. Li, L. T. Yang, S. Kwong, and R. Cong, "Diving into Underwater: Segment Anything Model Guided Underwater Salient Instance Segmentation and A Large-scale Dataset," Jun. 2024, arXiv:2406.06039 [cs]. [Online]. Available: http://arxiv.org/abs/2406.06039

[60] D. Katz and O. Brock, "Manipulating articulated objects with interactive perception," in *2008 IEEE International Conference on Robotics and Automation*. Pasadena, CA, USA: IEEE, May 2008, pp. 272–277. [Online]. Available: http://ieeexplore.ieee.org/document/4543220/

[61] C. Goldfeder, M. Ciocarlie, Hao Dang, and P. Allen, "The Columbia grasp database," in *2009 IEEE International Conference on Robotics and Automation*. Kobe: IEEE, May 2009, pp. 1710–1716. [Online]. Available: http://ieeexplore.ieee.org/document/5152709/

[62] A. Miller, S. Knoop, H. Christensen, and P. Allen, "Automatic grasp planning using shape primitives," in *2003 IEEE International Conference on Robotics and Automation (Cat. No.03CH37422)*. Taipei, Taiwan: IEEE, 2003, pp. 1824–1829. [Online]. Available: http://ieeexplore.ieee.org/document/1241860/

[63] J. Mahler, J. Liang, S. Niyaz, M. Laskey, R. Doan, X. Liu, J. A. Ojea, and K. Goldberg, "Dex-Net 2.0: Deep Learning to Plan Robust Grasps with Synthetic Point Clouds and Analytic Grasp Metrics," 2017, version Number: 3. [Online]. Available: https://arxiv.org/abs/1703.09312

[64] S. Levine, P. Pastor, A. Krizhevsky, J. Ibarz, and D. Quillen, "Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection," *The International Journal of Robotics Research*, vol. 37, no. 4-5, pp. 421–436, Apr. 2018. [Online]. Available: https://journals.sagepub.com/doi/10.1177/0278364917710318

[65] P. Song, Z. Fu, and L. Liu, "Grasp planning via hand-object geometric fitting," *The Visual Computer*, vol. 34, no. 2, pp. 257–270, Feb. 2018. [Online]. Available: http://link.springer.com/10.1007/s00371-016-1333-x

[66] M. Roa and R. Suarez, "Geometrical approach for grasp synthesis on discretized 3d objects," in *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*. San Diego, CA: IEEE, Oct. 2007, pp. 3283–3288. [Online]. Available: http://ieeexplore.ieee.org/document/4399440/

[67] U. Asif, M. Bennamoun, and F. A. Sohel, "RGB-D Object Recognition and Grasp Detection Using Hierarchical Cascaded Forests," *IEEE Transactions on Robotics*, vol. 33, no. 3, pp. 547–564, Jun. 2017. [Online]. Available: http://ieeexplore.ieee.org/document/7820198/

[68] R. Monica and J. Aleotti, "Point Cloud Projective Analysis for Part-Based Grasp Planning," *IEEE Robotics and Automation Letters*, vol. 5, no. 3, pp. 4695–4702, Jul. 2020. [Online]. Available: https://ieeexplore.ieee.org/document/9121693/

[69] S. Yu, D.-H. Zhai, H. Wu, H. Yang, and Y. Xia, "Object recognition and robot grasping technology based on RGB-D data," in *2020 39th Chinese Control Conference (CCC)*. Shenyang, China: IEEE, Jul. 2020, pp. 3869–3874. [Online]. Available: https://ieeexplore.ieee.org/document/9189078/

[70] J. Bowkett, S. Karumanchi, and R. Detry, "Grasping and Transport of Unstructured Collections of Massive Objects," *Field Robotics*, vol. 2, no. 1, pp. 385–405, Mar. 2022. [Online]. Available: https://fieldrobotics.net/Field_Robotics/Volume_2_files/Vol2_14.pdf

[71] D. Lodi Rizzini, F. Kallasi, J. Aleotti, F. Oleari, and S. Caselli, "Integration of a stereo vision system into an autonomous underwater vehicle for pipe manipulation tasks," *Computers & Electrical Engineering*, vol. 58, pp. 560–571, Feb. 2017. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0045790616302233

[72] S. K. Paul, M. T. Chowdhury, M. Nicolescu, M. Nicolescu, and D. Feil-Seifer, "Object Detection and Pose Estimation from RGB and Depth Data for Real-Time, Adaptive Robotic Grasping," in *Advances in Computer Vision and Computational Biology*, H. R. Arabnia, L. Deligiannidis,

H. Shouno, F. G. Tinetti, and Q.-N. Tran, Eds. Cham: Springer International Publishing, 2021, pp. 121–142, series Title: Transactions on Computational Science and Computational Intelligence. [Online]. Available: https://link.springer.com/10.1007/978-3-030-71051-4_10

[73] C. Wang, Q. Zhang, S. Li, X. Wang, D. Lane, Y. Petillot, and S. Wang, "Learning-Based Underwater Autonomous Grasping via 3D Point Cloud," in *OCEANS 2021: San Diego – Porto*. San Diego, CA, USA: IEEE, Sep. 2021, pp. 1–5. [Online]. Available: https://ieeexplore.ieee.org/document/9706035/

[74] C. Ferrari and J. Canny, "Planning optimal grasps," in *Proceedings 1992 IEEE International Conference on Robotics and Automation*. Nice, France: IEEE Comput. Soc. Press, 1992, pp. 2290–2295. [Online]. Available: http://ieeexplore.ieee.org/document/219918/

[75] R. Ozawa and K. Tahara, "Grasp and dexterous manipulation of multi-fingered robotic hands: a review from a control view point," *Advanced Robotics*, vol. 31, no. 19-20, pp. 1030–1050, Oct. 2017. [Online]. Available: https://www.tandfonline.com/doi/full/10.1080/01691864.2017.1365011

[76] Y. Dai, S. Yu, Y. Yan, and X. Yu, "An EKF-Based Fast Tube MPC Scheme for Moving Target Tracking of a Redundant Underwater Vehicle-Manipulator System," *IEEE/ASME Transactions on Mechatronics*, vol. 24, no. 6, pp. 2803–2814, Dec. 2019. [Online]. Available: https://ieeexplore.ieee.org/document/8846052/

[77] A. Villegas, V. Mishkevich, Y. Gulak, and F. Diez, "Analysis of key elements to evaluate the performance of a multirotor unmanned aerial–aquatic vehicle," *Aerospace Science and Technology*, vol. 70, pp. 412–418, Nov. 2017. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S1270963817302766

[78] C. Sachin, S. Ioan, and C. Steve, "MoveIt!" *IEEE Robotics & Automation Magazine*, vol. 19, no. 1, pp. 18–19, 2012.

[79] R. Buchanan, A. Röfer, J. Moura, A. Valada, and S. Vijayakumar, "Online Estimation of Articulated Objects with Factor Graphs using Vision and Proprioceptive Sensing," Sep. 2023, arXiv:2309.16343 [cs]. [Online]. Available: http://arxiv.org/abs/2309.16343