

# **Visual Methods Towards Autonomous Underwater Manipulation**

by

Gideon H. Billings

A dissertation submitted in partial fulfillment  
of the requirements for the degree of  
Doctor of Philosophy  
(Robotics)  
in the University of Michigan  
2022

Doctoral Committee:

Associate Professor Matthew Johnson-Roberson, Chair  
Associate Professor Maani G. Jadidi  
Professor Odest C. Jenkins  
Associate Professor Oscar Pizarro, University of Sydney

Gideon H. Billings

gidobot@umich.edu

ORCID iD: 0000-0003-4850-8789

© Gideon H. Billings 2022

To my parents,  
*Daniel and Susan Billings,*  
who blew on the spark that ignited my young imagination  
and encouraged me to set sail on the quest for my dreams.

*"They that go down to the sea in ships, that do business in great waters;  
These see the works of the LORD, and his wonders in the deep."*

*—Psalms 107:23-24*

## ACKNOWLEDGMENTS

Looking back from where I have come, it is evident that my path was paved by those many people who have encouraged me, believed in me, and given me opportunities to pursue my dreams. For all of these people, I am deeply thankful.

First, I would like to thank my Ph.D. adviser, Professor Matthew Johnson-Roberson, who has given support and mentorship along my entire graduate school journey. Your encouragement and advice were indispensable, and I am grateful for the freedom you gave me in my research pursuits. I would like to give a special thanks to Professor Oscar Pizarro, who has been both a mentor and a friend. Our candid conversations and field works together have helped shape my journey and greatly encouraged me along the way. I would also like to thank the other members of my doctoral committee – Professor Maani Jadidi and Professor Odest Jenkins – for your guidance and insightful feedback.

I would like to thank my collaborators at the Woods Hole Oceanographic Institution. First, a special thanks to Dr. Richard Camilli who's personal mentorship and guidance have been essential to my research accomplishments. Your friendship and wisdom have been a guiding light throughout my journey. I would also like to thank all of the NUI ROV team for their support throughout my field work. Your ability to overcome adversity and willingness to put in long hours in the field has enabled much of my work and has been a great inspiration.

I would like to thank my labmates in the DROP-lab – Eduino, Tianyi, Katie, Nick, Liz, and Laura – with whom I have many fond memories. Some of the greatest joys of my graduate school experience were sharing time together, whether it were long days and nights of field work or just having a game night, and I am deeply blessed to have such colleagues and friends.

I would like to thank all my family who have been a constant support and encouragement to me. In many ways you have been the wind beneath my wings. A deeply special thanks to my wife, Kia, who has stood by me through the lows and the highs. Thank you for your constant love and kindness and helping me find joy in life.

I would like to thank all those faculty, staff, and fellow students at the University of Michigan who have shaped my life in so many ways and are too numerous to name. You have challenged me to push my limits, think outside of the box, and achieve greater things. Thank you for being such a welcoming and supportive community who have made both my undergraduate and graduate years at the University of Michigan so rich and enjoyable.



Finally, I would like to thank all those in the Christians on Campus community who have been a strong spiritual support during my years on campus. Your prayers and fellowship have been a constant encouragement and a beacon of light throughout my journey. I also give thanks to God, who has blessed me with health in body and mind and whose ever present love has been my sustaining source.

This work was supported in part by a fellowship from the Robotics Institute at the University of Michigan, by NASA under a PSTAR grant NNX16AL08G, and by the National Science Foundation under grants IIS-1830660 and IIS-1830500.

# TABLE OF CONTENTS

DEDICATION . . . . .	ii
ACKNOWLEDGMENTS . . . . .	iii
LIST OF FIGURES . . . . .	viii
LIST OF TABLES . . . . .	xiii
ABSTRACT . . . . .	xv
CHAPTER	
<b>1 Introduction . . . . .</b>	<b>1</b>
1.1 Motivation . . . . .	1
1.1.1 Commercial and Military Applications . . . . .	2
1.1.2 Scientific Exploration and Sampling of The Deep Ocean . . . . .	3
1.1.3 Extraterrestrial Exploration in The Search for Life . . . . .	4
1.1.4 Challenges of Underwater Vision . . . . .	4
1.2 Problem Statement . . . . .	5
1.3 Contributions . . . . .	6
<b>2 6D Object Pose Estimation in RGB Perspective Images . . . . .</b>	<b>8</b>
2.1 Motivation . . . . .	8
2.2 Related Work . . . . .	9
2.3 Method . . . . .	11
2.3.1 Overview of the Network Pipeline . . . . .	11
2.3.2 Dataset . . . . .	16
2.3.3 Network Training . . . . .	16
2.4 Results . . . . .	17
2.4.1 Silhouette Prediction . . . . .	20
2.4.2 6D Pose Regression . . . . .	20
2.5 Conclusion . . . . .	22
<b>3 6D Object Pose Estimation in Fisheye and Omnidirectional Images . . . . .</b>	<b>23</b>
3.1 Motivation . . . . .	23
3.2 Related Work . . . . .	24
3.3 Method . . . . .	26

3.3.1	Spherical Mapping and Gnomonic Projection . . . . .	26
3.3.2	SilhoNet Adaptation to Fisheye . . . . .	29
3.3.3	Network Training . . . . .	30
3.3.4	Dataset . . . . .	31
3.4	Results . . . . .	31
3.5	Conclusion . . . . .	33
<b>4</b>	<b>Hybrid Visual SLAM for Underwater Vehicle Manipulator Systems . . . . .</b>	<b>35</b>
4.1	Motivation . . . . .	35
4.2	Related Work . . . . .	36
4.2.1	Feature Based Visual SLAM . . . . .	36
4.2.2	Underwater SLAM . . . . .	37
4.2.3	Kinematics in SLAM . . . . .	37
4.3	Method . . . . .	38
4.3.1	Hybrid Camera System . . . . .	39
4.3.2	Feature Representation . . . . .	39
4.3.3	System Initialization . . . . .	39
4.3.4	Stereo Odometry . . . . .	40
4.3.5	Tracking . . . . .	40
4.3.6	Inserting New Keyframes . . . . .	42
4.3.7	Loop Closing . . . . .	42
4.3.8	Datasets . . . . .	42
4.4	Results . . . . .	43
4.4.1	Comparative Feature Analysis . . . . .	43
4.4.2	Stereo SLAM . . . . .	45
4.4.3	Hybrid SLAM . . . . .	47
4.5	Conclusion . . . . .	50
<b>5</b>	<b>Design of Underwater Optical Systems . . . . .</b>	<b>52</b>
5.1	Motivation . . . . .	52
5.2	Underwater Image Formation . . . . .	53
5.2.1	Artificial Light systems . . . . .	54
5.2.2	Underwater Light Propagation . . . . .	54
5.2.3	Lensing effects . . . . .	56
5.2.4	Camera response . . . . .	57
5.2.5	Gain and Signal to Noise Ratio . . . . .	57
5.2.6	Operational Considerations . . . . .	58
5.3	Software . . . . .	60
5.4	Validation Experiments . . . . .	61
5.5	Conclusion . . . . .	64
<b>6</b>	<b>Automating Underwater Vehicle Manipulator Systems . . . . .</b>	<b>65</b>
6.1	Motivation . . . . .	65
6.2	Background . . . . .	68
6.3	System Overview . . . . .	70

6.3.1	Mission and Vehicle Platform Architecture . . . . .	70
6.3.2	Perception . . . . .	72
6.3.3	Control . . . . .	76
6.3.4	System Precision . . . . .	80
6.4	Experiments and Field Results . . . . .	80
6.4.1	Automated Pick-and-Place Demonstration on Testbed . . . . .	80
6.4.2	Real-Time Scene Reconstruction and Data Collection at the Costa Rican Pacific Shelf Margin . . . . .	81
6.4.3	Automated Sample Collection and Return within Active Submarine Vol- canoes . . . . .	83
6.4.4	Performance Analysis . . . . .	92
6.5	Discussion and Future Work . . . . .	93
6.6	Conclusions . . . . .	102
<b>7</b>	<b>Conclusions and Future Directions . . . . .</b>	<b>104</b>
7.1	Conclusions . . . . .	104
7.2	Future Directions . . . . .	105
	<b>BIBLIOGRAPHY . . . . .</b>	<b>107</b>

## LIST OF FIGURES

### FIGURE

1.1	Still frame from the HD science camera of the SuBastian remotely operated vehicle (ROV), operated by Schmidt Ocean Institute, while collecting a push core sample at a depth exceeding 1,000m. . . . .	3
1.2	Examples of degraded images in underwater environments, due to water column effects, showing suspended particulates, poor lighting, low scene contrast, and haze due to backscatter. . . . .	5
2.1	Overview of the SilhoNet pipeline for silhouette prediction and 6D object pose estimation. The 3D translation is predicted in parallel with the silhouettes. The predicted unoccluded silhouette is fed into a second stage network to predict the 3D rotation vector. . . . .	10
2.2	Example prediction of occluded and unoccluded silhouettes from a test image . . . . .	19
2.3	6D pose accuracy curve across all objects in the YCB-video dataset. Accuracy is percentage of errors less than the error threshold. The PoseCNN orientation predictions are reduced by the same geometric symmetries as SilhoNet. . . . .	21
3.1	Overview of the three different SilhoNet adaptations for processing full fisheye images. The Baesline method processes the raw fisheye image directly through the unmodified network. The Projective variant processes the raw fisheye image through the feature extraction stage and then projects the features within the region of interest (ROI) through a spherical mapping to the tangent plane centered on the ROI, before processing the features through the ROI-pooling stage. The Perspective adaptation maps the fisheye image to a sphere and then generates a virtual perspective image for each object detection using a gnomonic projection, centered on the ROI. Each virtual image is then processed through the network. . . . .	26
3.2	These objects have the same orientation relative to the rendered fisheye image frame but different translations, resulting in drastically different apparent orientations. Also, objects appear more distorted as they move from the image center. . . . .	27
3.3	Qualitative results with the perspective method on some sample test images for the whoihandle object. Predicted silhouettes and pose errors are shown for a range of errors from low to high. . . . .	33
4.1	System block diagram . . . . .	39
4.2	The LizardIsland spiral survey dataset was collected with a diver operated stereo rig. The ground truth reconstruction was generated with COLMAP. . . . .	43

4.3	Four hybrid image sequences were collected in deep seafloor environments of the Costa Rican shelf margin. Shown here is a sample left stereo image from each sequence. Mounds1 ((a)) is an area of rocks and bacterial matting. Mounds2 ((b)) is a mud flat with rubble. Seeps1 ((c)) is a dense bed of clams with bacterial matting. Seeps2 ((d)) is a mud flat with a small patch of bacterial matting. . . . .	44
4.4	Final stereo SLAM maps on the LizardIsland dataset, showing the densely connected keyframe graphs. . . . .	46
4.5	Stereo SLAM results for the LizardIsland dataset when loop closing is disabled ((a),(b)) and enabled ((c),(d)). . . . .	48
4.6	Snapshots of hybrid SLAM running on the UWHandles sequences. Top row is the left stereo camera frame, middle row is the manipulator mounted fisheye frame, and bottom row is the map with the keypoints and keyframes. . . . .	49
5.1	Schematic of underwater light propagation from light source to camera sensor, where the light signal is affected by scattering and absorption through the water column and the reflection characteristics of the seafloor. . . . .	53
5.2	Image formation pipeline describing the different steps through which light is subjected to form the underwater digital image. . . . .	55
5.3	Radiance spectrum for different light types . . . . .	58
5.4	Depth of field as a function of focus distance and aperture . . . . .	60
5.5	Spectrum of light as it propagates through the water, attenuates, reflects and travels through the lens onto the sensor. . . . .	61
5.6	Experimental setup for verifying image formation model. . . . .	62
5.7	Comparison of measured and estimated light spectrum at both the target board as well as the camera position . . . . .	62
5.8	Measured and model predicted camera response curves for two different sensors under the same experimental conditions. . . . .	63
5.9	Camera response for two different lenses and without a lens. . . . .	63
6.1	Conceptual graphic of the our control system for an underwater intervention vehicle. The autonomy system runs on a topside desktop computer with visual sensor data and manipulator coms streamed over a high bandwidth tether from the vehicle. Solid red flow lines represent standard teleoperated control from a surface ship. Blue flow lines represent our automated system. Red dashed lines represent interfacing between the pilot and the autonomous system, where, in this work, the pilot acts as the high level task planner and interfaces with the automated system through a graphical scene representation and task level controller. Eventually, the pilot would be replaced with an automated mission planner that could issue high level tasks. . . . .	67
6.2	Photograph taken by the <i>NUI</i> vehicle within the Kolumbo volcano crater that shows an overhanging vertical wall of columnar lava. Colonization of the lava surfaces by relatively uncommon lollipop sponges ( <i>Stylocordyla pellita</i> ) are visible as white dots within the image. . . . .	71
6.3	The vision system for autonomy is composed of (a) a wrist-mounted fisheye camera and (b) a vehicle-mounted stereo pair (shown here mounted on the <i>SuBastian</i> ROV). The vision system can be easily integrated onto existing vehicles. . . . .	73

6.4	A comparison of (top) the full view of the wrist-mounted fisheye camera in an underwater scene at close and far range compared to (bottom) a 60perspective rectification, which illustrates the significant increase in the field-of-view provided by a fisheye lens compared to a conventional perspective lens. This increased field-of-view provides significantly better contextual awareness to the vision and manipulation systems, especially when working at close range to the target, which is typical for manipulation tasks. . . . .	74
6.5	A single type of t-handle was used to manipulate the different tools. The vision system localizes the t-handles using (a) AprilTags affixed to 3D-printed mounts located beneath the t-handle. These tags are detected in (b) images of the ROV tool tray from the wrist-mounted fisheye camera. . . . .	75
6.6	A diagram of the overall system, where rectangular blocks represent processes and diamond-shaped blocks represent hardware. Blocks in blue relate to perception. Blocks in red relate to (left) high- and (right) low-level control. Blocks in green are part of the MoveIt! framework around which our system is built. Our system uses the stereo camera to estimate the vehicle configuration (e.g., the pose of the doors on the <i>NUI</i> HROV), generate point clouds of the scene that can be fused to produce a 3D reconstruction of the scene, and assist with tool localization. The fisheye camera is used to localize tools, obtain dynamic viewpoints of the workspace, and extend the scene reconstruction. For low-level control, a driver implements a position-based trajectory controller, which integrates between MoveIt! and the manipulator valve controller. For high-level control, we implemented an automation interface to MoveIt! that supports high-level commands. In this work, we implement this interface using a graphical front-end as well as a preliminary demonstration using natural language. . . . .	77
6.7	(right) An image of our testbed consisting of a Kraft TeleRobotics manipulator, a fisheye camera mounted to the end-effector, and a overhead stereo camera. Together with the manipulator base frame, there are four references frames (left) which must be calibrated in order to fuse sensor data into a common reference frame and to plan the motion of the arm. Calibration is performed in the order shown on the left, where each transform enables calibrating the next in a bootstrapping manner. The fiducial in the image is included to indicate that AprilTags were placed statically in the workspace to obtain the Gripper-to-Fisheye and Base-to-Stereo calibrations. . . . .	78
6.8	A simple interface to the automated system allows the user to configure and step through the automated pick-and-place pipeline. The motion plan for each step is visualized in the planning scene and is only executed upon confirmation by the user, which provides a high-level of safety for the system to be deployed on ocean-going systems. . . . .	79

6.9	We demonstrated fully autonomous pick-and-place with a t-handle on a testbed with the same camera and manipulator hardware used on the <i>NUI</i> HROV. First, (a) the t-handle was detected from the fisheye camera using the AprilTags, and the handle pose was projected into the planning scene. Next, (b) the manipulator was commanded to grasp the t-handle via the autonomy interface. Subsequently, (c) a sample location was set in the planning scene with an interactive marker based on the projected stereo point cloud, the manipulator planned a motion to reach the sample location, and executed the plan after the user verified it. The manipulator was then (d) commanded to return the t-handle to the location where it was first grasped. The rock in the environment was placed in a delicate balance on its end, yet the manipulator was controlled with enough precision to bring the tool into direct contact without knocking it over. . . . .	82
6.10	The vision system was integrated on the <i>SuBastian</i> ROV operated by Schmidt Ocean Institute, where we demonstrated real-time visualization of the planning scene with a Schilling Titan-4 manipulator and the projected stereo point clouds. This also demonstrates the flexibility of the system to be integrated with different vehicles and manipulators. . . . .	83
6.11	Bathymetric map of the survey area from the 2018 cruise on the Pacific continental margin showing data collection locations at seven different science goal sites, spanning over 62 km (linear distance between Locations 1 and 4) and ranging in depth from 600 m to 1100 m. Depth contours are spaced at 250 m intervals and the map is oriented with North up. . . . .	84
6.12	The fisheye imagery collected during the Costa Rica cruise was processed into a stand-alone dataset [15]. The images are annotated with the bounding box and six-DoF pose of the tool handles placed in the workspace. The top row (a) shows sample raw fisheye images from different sequences of the dataset, and the bottom row (b) shows sample annotations from a single sequence in the dataset. The images are center rectified here only for purposes of visualization. . . . .	85
6.13	Map of automated sample collection locations, with regional bathymetry adapted from [140, 139]. The sea level contour is indicated in black. The dashed line indicates the Christiana-Santorini-Kolumbo tectonic line [140]. Locations marked A, B, and D indicate automated sample collection and return sites, and location C indicates the site where a natural language proof-of-concept demonstration was conducted. Sampling depths ranged from 240 m to 501 m . . . . .	86
6.14	The <i>NUI</i> vehicle is outfitted with clam shell doors that can be closed to reduce drag when cruising and opened to perform manipulation tasks. The manipulator is mounted to the starboard door and the stereo cameras are mounted to the port door. . . . .	87
6.15	Fiducial-based visual SLAM from the left stereo camera was used to estimate the door angles in real-time using (left) tags mounted to the front of the vehicle frame and at the base of the manipulator on the starboard door. SLAM provided estimates of (middle) the relative transformations between the camera and the tag frames that were used (right) to estimate the door angles and update the vehicle model in the planning scene. The left stereo camera was also used in conjunction with the wrist mounted fisheye for (left) fiducial-based localization of tools. . . . .	87



6.16	A 2D schematic of the <i>NUI</i> HROV that relates the visual SLAM from the left stereo camera to the door positions. The green dashed lines represent transformations estimated from SLAM. The red dashed lines denote known transformations computed from the vehicle kinematic model and the measured position of the tags. The grey dashed lines represent the calculated transforms with respect to each door reference frame, which have a trigonometric relation to the door angles, $\theta_s$ and $\theta_p$ . . . . .	88
6.17	An example of a successful planner controlled slurp collection of a bacterial mat, with the yellow slurp hose attached to the manipulator. The manipulator was (a) commanded to the desired slurp location through the automated planning interface and then (b) directed to return to its home position following the slurp collection. . . .	89
6.18	A visualization of (top) the DCG factor graph for the expression “get the pushcore from the tooltray” aligned with (bottom) the associated parse tree. Shaded nodes denote observed random variables, while those rendered in white are latent. . . . .	90
6.19	Demonstration of a proof-of-concept framework that enabled operators to interact with our autonomous manipulation architecture using natural language. Given input in the form of free-form text, either entered by the operator or output by a cloud-based speech recognizer, we (left) infer the meaning of the command using a probabilistic language model. (a) In the case of the command to “go to the sample location”, our system (top-right) determines the goal configuration and solves for a collision-free path in configuration space. (b) Given the command to “execute now”, the manipulator then (bottom-right) executes the planned path to the goal. . . . .	91
6.20	Plot of the TagSLAM estimated trajectory (visual) of the fisheye camera versus the trajectory estimated from the manipulator joint feedback (kinematic). The trajectory is plotted separately for each coordinate axis with respect to the manipulator base frame.	92
6.20	Plot of commanded versus followed joint trajectories for the <i>testbed</i> manipulator. . . .	95
6.20	Plot of commanded versus followed joint trajectories for the <i>NUI</i> HROV manipulator during the Greece field trials. . . . .	97
6.21	The quality of stereo reconstruction is highly dependent on underwater conditions. Here, we compare stereo point clouds generated using the same camera system and stereo matching method, but with images captured within very different seafloor environments. The left images show the view from the left stereo camera, and the right images show the generated point clouds using a SGM-based stereo method. (u) The top row was captured in the clear waters off Costa Rica, with even scene lighting and highly textured seafloor. (v) The bottom row was captured in the Kolumbo caldera, with high backscatter and low texture microbial mats on the seafloor. . . . .	98

## LIST OF TABLES

### TABLE

2.1	Mean IoU accuracy for predicted silhouettes . . . . .	17
2.2	Mean 3D orientation error in degrees. The Sym tag indicates orientation predictions are reduced by geometric symmetries. . . . .	18
2.3	Mean 3D translation error in centimeters . . . . .	18
2.4	Area under accuracy-threshold curve for 6D pose evaluation using ADD-S metric . . .	19
2.5	Silhouette and orientation accuracy vs # of model images . . . . .	22
3.1	Percentage of translation predictions under the threshold error, where a higher percentage under a lower threshold means better accuracy. . . . .	31
3.2	Percentage of orientation predictions under the threshold error, where a higher percentage under a lower threshold means better accuracy. . . . .	31
3.3	Area under accuracy-threshold curve for 6D pose evaluation using ADD-S metric, where a higher area means better accuracy. Proj. is short for Projective and Persp. is short for Perspective . . . . .	31
4.1	Area under accuracy-threshold curve evaluation of feature matching performance on the UWHandles underwater hybrid image sequences. Accuracy is evaluated as angular error in the predicted rotation (AUC Rot) and translation direction vector (AUC Trans) between each hybrid left stereo and fisheye image pair. Also reported is the mean number of inlier feature matches across each sequence. . . . .	46
4.2	Stereo SLAM performance on the LizardIsland dataset, with the number of extracted features is set to 4000 and 2000. Performance is evaluated as RMSE of the absolute trajectory error. Results are reported with and without loop closing enabled. Also reported is the number of keyframes (KFs) and map points (MPs) in the final map and the average frame processing time in the tracking thread. . . . .	47
4.3	Evaluation of hybrid SLAM on the UWHandles dataset. Error is evaluated on the estimated pose difference between the left stereo and fisheye cameras for each synchronized hybrid frame, where $\Delta t$ is translation error and $\Delta q$ is rotation error. The "hybrid matches" column gives the number of fisheye frames registered in the map over the total number of frames in the sequence. The error is only evaluated over the registered frames. The "KFs" column is the number of keyframes in the final map for the hybrid SLAM mode versus stereo only mode, and the "MPs" column is the same format for the number of final keypoints in the map. . . . .	47
4.4	Hybrid SLAM timing evaluation, measured as the mean frame processing time in the tracking thread. . . . .	50

6.1 Comparison of *SuBastian* and *NUI* configurations. . . . . 72

6.2 Comparison of the bandwidth requirements for direct teleoperation (top two rows) of an ROV manipulator system compared to operating our high-level autonomy system (bottom two rows), running onboard the vehicle with communication through natural language commands and only the necessary scene state feedback to inform the high-level commands. . . . . 100

## ABSTRACT

Extra-terrestrial ocean worlds like Europa offer tantalizing targets in the search for extant life beyond the confines of Earth’s atmosphere. However, reaching and exploring the underwater environments of these alien worlds is a task with immense challenges. Unlike terrestrial based missions, the exploration of ocean worlds necessitates robots which are capable of fully automated operation. These robots must rely on local sensors to interpret the scene, plan their motions, and complete their mission tasks. Manipulation tasks, such as sample collection, are particularly challenging in underwater environments, where the manipulation platform is mobile, and the environment is unstructured.

This dissertation addresses some of the challenges in visual scene understanding to support autonomous manipulation with underwater vehicle manipulator systems (UVMSs). Specifically, this work addresses the problems of tool detection and pose estimation, 3D scene reconstruction, underwater camera system design, underwater dataset collection, and UVMS manipulator automation. The developed visual methods are demonstrated with a lightweight vision system, composed of a vehicle mounted stereo pair and a manipulator wrist mounted fisheye camera, that can be easily integrated on existing UVMSs. While the stereo camera primarily supports 3D reconstruction of the manipulator working area, the wrist mounted camera enables dynamic viewpoint acquisition for detecting objects, such as tools, and extending the scene reconstruction beyond the fixed stereo view. A further objective of this dissertation was to apply deep learning with the developed visual methods. While deep learning has greatly advanced the state-of-the-art in terrestrial based visual methods across diverse applications, the challenges of accessing the underwater environment and collecting underwater datasets for training these methods has hindered progress in advancing visual methods for underwater applications.

Following is an overview of the contributions made by this dissertation. The first contribution is a novel deep learning method for object detection and pose estimation from monocular images. The second contribution is a general framework for adapting monocular image-based pose estimation networks to work on full fisheye or omni-directional images with minimal modification to the network architecture. The third contribution is a visual SLAM method designed for UVMSs that fuses features from both the wrist mounted fisheye camera and the vehicle mounted stereo pair into the same map, where the map scale is constrained by the stereo features, and the wrist camera

can actively extend the map beyond the limited stereo view. The fourth contribution is an open-source tool to aid the design of underwater camera and lighting systems. The fifth contribution is an autonomy framework for UVMS manipulator control and the vision system that was used throughout this dissertation work, along with experimental results from field trials in natural deep ocean environments, including an active submarine volcano in the Mediterranean basin. The sixth contribution is a large scale annotated underwater visual dataset for object pose estimation and 3D scene reconstruction. The dataset was collected with our vision system in natural deep ocean environments and supported the development of the visual methods contributed by this dissertation.

# CHAPTER 1

## Introduction

### 1.1 Motivation

Since ancient times, humans have sought to penetrate the depths of the oceans and probe what lies beneath the surface. As in modern times, much of the early ocean diving was driven by commercial exploitation, salvage, and military operations [60]. Efforts to map and understand the deeper ocean, beyond the limits of human divers, began in the mid 19th century, driven largely by an initiative to survey the Gulf Stream [1]. Dredging from the H.M.S Lightning during the Gulf Survey recovered sea life from a depth exceeding 4km, discrediting previous speculation that the sea was lifeless below 549m and sparking scientific interest in deep ocean exploration [197]. In the late 19th century, the H.M.S. Challenger set the groundwork for modern oceanography by circumnavigating the globe while conducting scientific research. Until the early 20th century, deep ocean surveys were carried out from surface vessels using primitive sampling methods. The invention of acoustic sounders was a giant leap forward for oceanographic mapping technology, and the pioneering work of William Beebe and Otis Barton with the first manned bathysphere in 1934 heralded an era of manned submersible exploration of the deep ocean. The 1960s saw the first use of unmanned submersibles for oceanography with the development the Deep Tow System by Scripps Institution of Oceanography. This period also saw the construction of Alvin, the first of its ship class of Deep Submersible Vehicles, designed to replace bathyscaphes and other less maneuverable oceanographic vehicles [2]. The Alvin, operated by Woods Hole Oceanographic Institution (WHOI), was outfitted with a dexterous manipulator for performing pilot guided interventions. Building on this technology, the unmanned work-class ROV was developed for offshore oil and gas, becoming a crucial part of the industry for performing operations at depths exceeding the limits of human divers [3]. In the following decades, ROVs outfitted with pilot operated manipulator systems also became the primary tool in oceanography for deep ocean exploration and sample collection.

Despite rapid advancements in underwater vehicle and manipulator system technology from the

late 20th century to the present time, ROV field operations remain exclusively pilot controlled. The state of the art practice for ROV manipulation is direct teleoperation, typically through a miniature master arm acting as a joystick controller [169]. In the last decade, more attention has been given to automating underwater manipulation. However, demonstrations have mostly been limited to pools and shallow water environments, with manipulation tasks focused on interactions with man-made structures. The lag of underwater manipulation autonomy behind terrestrial based manipulation systems can be at least partly attributed to the expense of building underwater systems, the risk of operating in unstructured underwater environments, and the challenges of using visual sensors in underwater environments for localization, scene reconstruction and semantic understanding [102]. This dissertation seeks to address some of the challenges of using optical sensors in the underwater domain and to develop a framework for automating existing work-class manipulator systems while minimizing the risk of operation.

### **1.1.1 Commercial and Military Applications**

Work-class ROVs have become an essential tool of modern sub-sea industry [31, 169]. In the energy, oil and gas sectors, they are used heavily for sub-sea pipe and structure construction, inspection, maintenance and repair. They have also found use in the civil field for bridge and pier inspections and servicing, aquaculture for net inspection and dead fish removal, and other sectors of industry. Most commercial ROV operations involve a manipulation task, typically in a structured environment. Some examples of common tasks include turning a valve, plugging a connector, cleaning a surface or salvaging an object. Though ROV technology has become integral to commercial exploitation of the ocean, substantial infrastructure is required for their operation, including a team of trained pilots maintaining constant oversight and control of the vehicle during operation, an array of monitoring equipment and interfacing hardware to provide adequate visual, sensory, and system state feedback for pilot control, and a surface vessel from which the ROV is tethered and operated. Operational costs for a single ROV system, including personnel time, fuel consumption, and ship operations, can run up to a five or six figure sum per day.

Automating common ROV manipulation tasks for the sub-sea industry would reduce the need for trained pilots, minimize the required topside control center infrastructure, and enable integration of manipulators on AUVs and resident vehicles that do not require a surface vessel for operation. Besides the associated direct operational cost savings, an autonomous system can also reduce operational risk from human error, which may result in lost time or mission failure in the worst case.

Work-class ROVs also have a long history with the U.S. military. The first work-class ROV, the CURV-I, was developed by the Navy for recovering torpedoes and other ordinances from the

seabed [146] and was made famous in 1966 for recovering a hydrogen bomb, lost from a B-52 bomber collision over the Mediterranean Sea, at a depth of 880m. Recently, with the DARPA Angler program, the Navy has been pushing for unmanned underwater vehicles capable of performing fully autonomous search and manipulation objectives in deep ocean environments [142].

### 1.1.2 Scientific Exploration and Sampling of The Deep Ocean

ROVs have become the workhorse for deep-sea oceanography. Modern ROVs used in oceanography are outfitted with high definition science cameras, an array of scientific instruments, and tools for collecting and returning samples. Figure 1.1 shows a still frame from the science camera on the SuBastian ROV, operated by Schmidt Ocean Institute (SOI), captured while exploring and collecting biological samples on the Costa Rica shelf break. While ROVs have unlocked the deep ocean for scientific research, their cost of operation is prohibitive for many projects, and only limited numbers of deep ocean rated ROVs are dedicated to oceanographic research, making it highly competitive to procure research time with one of these vehicles.

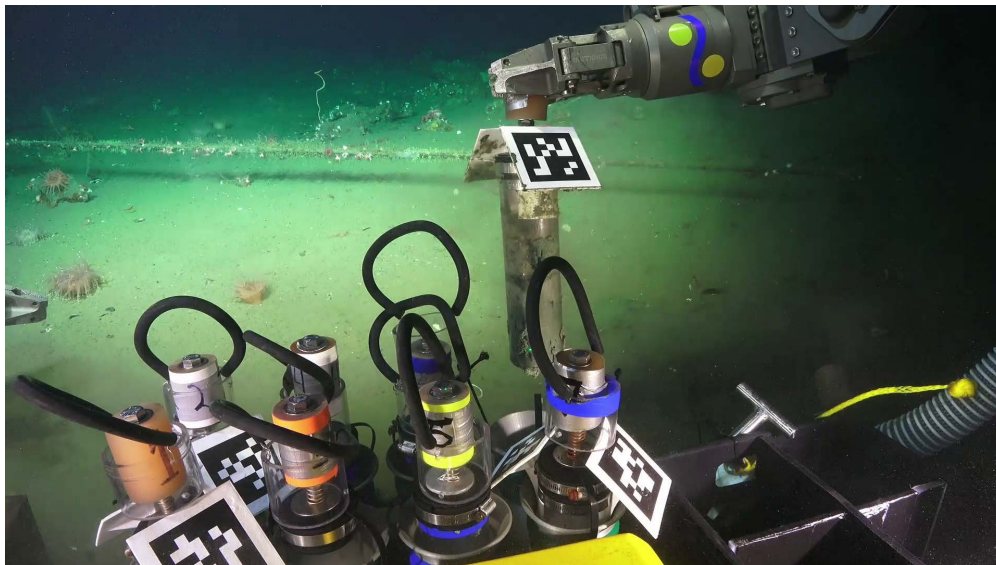


Figure 1.1: Still frame from the HD science camera of the SuBastian ROV, operated by Schmidt Ocean Institute, while collecting a push core sample at a depth exceeding 1,000m.

Automating underwater manipulation tasks common to oceanography would enable manipulators to be integrated onto AUVs, which are less costly to build, maintain and operate compared to ROVs. By removing the cost overhead of ship time, pilots, and operating infrastructure, AUV based manipulation would greatly enhance the field of oceanography and open the way for many projects where funding or ship time might otherwise be a barrier.



### **1.1.3 Extraterrestrial Exploration in The Search for Life**

In 1977, Robert Ballard confirmed the existence of hydrothermal vents [197] in the deep ocean with thriving ecosystems of chemolithoautotrophic organisms, which survive in the absence of light by obtaining energy through chemical processes [7]. This discovery not only vitalized the field of deep sea oceanography but also captured the attention of astrobiologists; similar geothermal processes which drive hydrothermal venting could operate in the extreme environments of extraterrestrial oceans, such as that believed to exist beneath the ice bound surface of Europa. These environments are a prime target to search for extraterrestrial life, which may have developed beyond the reach of the sun's harsh radiation.

Planning for a Europa lander mission is well under way [66], and the vehicle concept includes a sample manipulation system, which will operate under similar hardware constraints to deep-sea manipulator systems, due to the extreme environments of space. Because of the high communication delay between the Earth and Europa and the risk of intermittent communication failure, it is critical that the manipulation system be capable of conducting a fully autonomous sampling cycle. In the eventuality that a probe is deployed into the sub-surface ocean, communication with Earth will be completely cut off, and the sample search, identification and collection processes must operate fully autonomously.

### **1.1.4 Challenges of Underwater Vision**

There are unique challenges when using visual sensors underwater, compared to terrestrial based applications. Figure 1.2 shows some example images that illustrate how visual sensor data can be degraded in underwater environments. The fundamental challenge is that photons propagating through the water column are scattered and absorbed in a wavelength dependent manner [171]. For imaging sensors, these physical processes induce haze and distortions in the collected images and reduce the overall photometric and color contrast. These effects are also highly variable with the water column properties and the quality of the scene lighting, making them extremely challenging to model. Visual methods that operate on RGB image data largely depend on either extracting and matching features or making direct photometric comparisons of images. However, the wavelength dependent attenuation of light signals in the water column breaks the photometric consistency assumption, which underlies many feature representations or methods that make direct pixel comparisons between images. For example, an image patch that is imaged underwater with the same camera and lighting system at 1m distance will have a significantly different color and contrast when imaged at 2m. These underwater imaging effects result in many visual methods that perform very well in terrestrial based environments to have very brittle performance in the underwater domain. There is a large visual domain shift between in-air and underwater environments, so visual

methods must be designed specifically for the underwater domain to achieve robust performance.

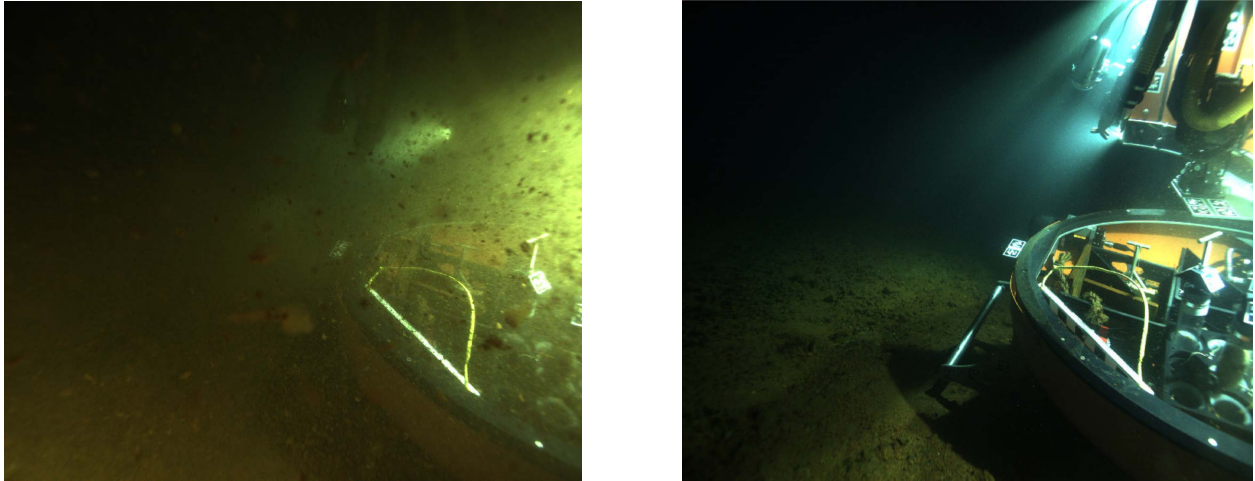


Figure 1.2: Examples of degraded images in underwater environments, due to water column effects, showing suspended particulates, poor lighting, low scene contrast, and haze due to backscatter.

## 1.2 Problem Statement

This dissertation addresses the following problems in automating underwater manipulation

1. ROVs generally carry an array of tools for performing various sampling and intervention tasks. An autonomous system must be able to localize these tools accurately in order to grasp and manipulate them.
2. An autonomous manipulator system must be able to reconstruct the local workspace in real-time to safely perform manipulation tasks in the environment. While much attention has been given to visual based underwater localization and large-scale post-processed reconstructions from surveys, there is a lack of works that address real-time SLAM and scene reconstruction to support autonomous underwater manipulation in unstructured and diverse seafloor environments.
3. Hydraulic manipulators are standard for work-class ROVs because of their reliability in underwater environments, superior power to weight ratio, and generally greater depth rating, compared to electric alternatives. However, hydraulic manipulators provide a minimal command interface with limited joint feedback and are less precise than electric manipulators. To be widely adoptable, a framework for autonomous underwater manipulation should be

largely agnostic to the particular hardware configuration and be readily integrated with existing work-class hydraulic manipulator systems.

4. Deep learning has greatly advanced the field of computer vision but relies on visual datasets to learn the computation models. However, annotated datasets from the underwater domain are scarce, due to the expense and challenges of gathering and annotating underwater data. In order to advance the state-of-the-art in visual methods for the underwater domain, annotated datasets collected in natural and diverse sub-sea environments are needed to support the development of learning based methods.

## 1.3 Contributions

The following list enumerates the specific contributions of this dissertation with the corresponding chapters.

- SilhoNet: a novel deep learning method to estimate object pose and occlusion in cluttered scenes from monocular images. A key novelty of SilhoNet is the use of an intermediate silhouette representation to bridge the sim-to-real domain shift and facilitate learning a model from synthetic data. (Chapter 2)
- SilhoNet-Fisheye: a mathematical framework for adapting ROI-based networks for predicting 6D object pose from monocular images to work on full fisheye and omni-directional images. This method builds on prior work in object detection from omni-directional images to extend the application of the gnomonic projection from an intermediate spherical mapping to compensate for image distortions and viewpoint ambiguities when predicting object pose. (Chapter 3)
- A SLAM method designed for underwater vehicle manipulator systems that fuses an independent monocular wrist mounted fisheye camera with a vehicle mounted perspective stereo pair, enabling active extension of the scene map with the manipulator camera beyond the limited view of the stereo pair. (Chapter 4)
- An open source tool to aid underwater optical system design. The tool incorporates an experimentally verified underwater image formation model to enable parametric exploration of the system design space through an intuitive graphical user interface. (Chapter 5)
- A camera system and automation framework for underwater vehicle manipulator systems. Demonstrations were made on a manipulator testbed and in natural deep seabed habitats of the Costa Rica Shelf Break, the Santa Monica Basin, and Kolumbo, an active submarine

volcano in the Mediterranean basin. This work culminated in planner controlled biological sample collection with an ROV and hydraulic manipulator system. Full planner controlled pick-and-place of a tool handle was also demonstrated on the manipulator testbed. (Chapter 6)

- **UWHandles**: an underwater fisheye dataset and annotation tool for 6D object pose estimation. This dataset addresses a lack of widely available annotated fisheye datasets and the general lack of publicly available underwater image datasets for deep learning applications. (Chapter 6)

Work presented in this dissertation, as well as related research, has been published in the following manuscripts:

**G. Billings** and M. Johnson-Roberson, "SilhoNet: An RGB Method for 6D Object Pose Estimation," in IEEE Robotics and Automation Letters, vol. 4, no. 4, pp. 3727-3734, Oct. 2019.

**G. Billings** and M. Johnson-Roberson, "SilhoNet-Fisheye: Adaptation of A ROI Based Object Pose Estimation Network to Monocular Fisheye Images," in IEEE Robotics and Automation Letters, vol. 5, no. 3, pp. 4241-4248, July 2020.

**G. Billings**, E. Iscar and M. Johnson-Roberson, "Parametric Design of Underwater Optical Systems." in IEEE Global OCEANS: Singapore-U.S. Gulf Coast, 2020.

**G. Billings**, R. Camilli, M. Walter, O. Pizarro and M. Johnson-Roberson, "Towards Automated Sample Collection and Return in Extreme Underwater Environments." Under review. Submitted to Field Robotics

**G. Billings**, R Camilli and M. Johnson-Roberson, "Hybrid Visual SLAM for Underwater Vehicle Manipulator Systems." Under review. Submitted to IEEE Robotics and Automation Letters

Some of my work has been a collaborative effort. Dr. Eduardo Iscar had equal contribution in the experiments and development of the underwater optical system design tool presented in Ch. 5. Prof. Mathew Walter developed the natural language network and code that interfaced with the automated system presented in ch. 6. Dr. Oscar Pizarro aided in the design and mounting configuration of the camera system used throughout my dissertation work. Dr. Richard Camilli led the field expeditions where the automated system was trialed and the datasets were collected that supported the development of the visual methods in this dissertation.

## CHAPTER 2

# 6D Object Pose Estimation in RGB Perspective Images

### 2.1 Motivation

Robots are revolutionizing the way technology enhances our lives. From helping people with disabilities perform various tasks around their house to autonomously collecting data in humanly inaccessible environments, robots are being applied across a spectrum of exciting and impactful domains. Many of these applications require the robot to grasp and manipulate an object in some way (e.g., opening a door by a handle, or picking up an object from the seafloor), but this poses a challenging problem. Specifically, the robot must interpret sensory information of the scene to localize the object. Beyond robot manipulation, there are also applications, such as augmented reality, which require accurate localization of an object in an image.

Previous methods for object pose estimation largely depend on RGB-D data about the 3D working environment [19, 145, 128, 11]. However, there are cases where such depth information is not readily available. Some examples include systems that operate outdoors where common depth sensors like the Kinect do not work well because of projection range limitations, embedded systems where space and cost may limit the size and number of sensors, and underwater vehicles where the variable absorption and scattering properties of the water column attenuates light signals and degrades the performance of active depth sensors and stereo matching can be sparse and noisy due to water column effects. In these scenarios, methods that operate on monocular camera data are needed. When the sensor modality is limited to monocular images, estimating the pose of an object in a natural setting is a challenging problem due to variability in scene illumination, the variety of object shapes and textures, and occlusions caused by scene clutter.

Recently, there has been progress in state-of-the-art methods for monocular image pose estimation on difficult datasets, where the scenes are cluttered and objects are often heavily occluded [30, 151, 188, 177, 97, 113, 117]. The presented work improves on the performance of these recent methods to deliver a novel deep learning based method for 6D object pose estimation

on monocular images. Unlike prior methods, we explicitly incorporate prior knowledge of the 3D object appearance into the network architecture, and we make use of an intermediate silhouette based object viewpoint representation to improve on orientation prediction accuracy. Further, this method provides occlusion information about the object, which can be used to determine which parts of an object model are visible in the scene. Knowing how the target object is occluded in the monocular image can be important for certain applications, such as augmented reality, where it is desirable to project over only the visible portion of an object.

In this chapter, we present the following contributions: 1. SilhoNet, a novel RGB-based deep learning method to estimate pose and occlusion in cluttered scenes; 2. The use of an intermediate silhouette representation to facilitate learning a model on synthetic data to predict 6D object pose on real data, effectively bridging the sim-to-real domain shift [33]; 3. A method to determine which parts of an object model are visually unoccluded, using the projection of inferred silhouettes, in novel scenes; 4. An evaluation on the visually challenging YCB-Video dataset [188] where the proposed approach outperforms two state-of-the-art RGB method.

The rest of this chapter is organized in the following sections: section 2.2 discusses related work; section 2.3 presents our method with an overview of the CNN design for 6D pose estimation and occlusion mask prediction; section 2.4 presents the experimental results; and section 2.5 concludes the chapter.

## 2.2 Related Work

Extensive research has focused on 6D object pose estimation using RGB/D data. Several works rely on feature- and shape-based template matching to locate the object in the image and coarsely estimate the pose [74, 153, 30]. This is often followed by a refinement step using the Iterative Closest Point (ICP) algorithm with the 3D object model and a depth map of the scene [74]. While these methods are computationally efficient, their performance often degrades in cluttered environments. Other methods have exploited point cloud data to match 3D features and fit the object models into the scene [50, 75, 75]. While point cloud based methods achieve state-of-the-art performance, they can be very computationally expensive. Recent works have demonstrated the power of machine learning for object detection and pose estimation using RGB/D data. [162] used a CNN pretrained on ImageNet to extract features from an RGB image and a colorized depth map. They learned a series of Support Vector Machines (SVM) on top of these extracted features to predict the object category and a single axis rotation about a planar surface normal. In [21], they trained a decision forest to regress every pixel from an RGB/D image to an object class and a coordinate position on the object model. Other work has used a CNN to map the pose of an object in an observed RGB/D image to a rendered pose of the model through an energy function [104]. The minimiza-

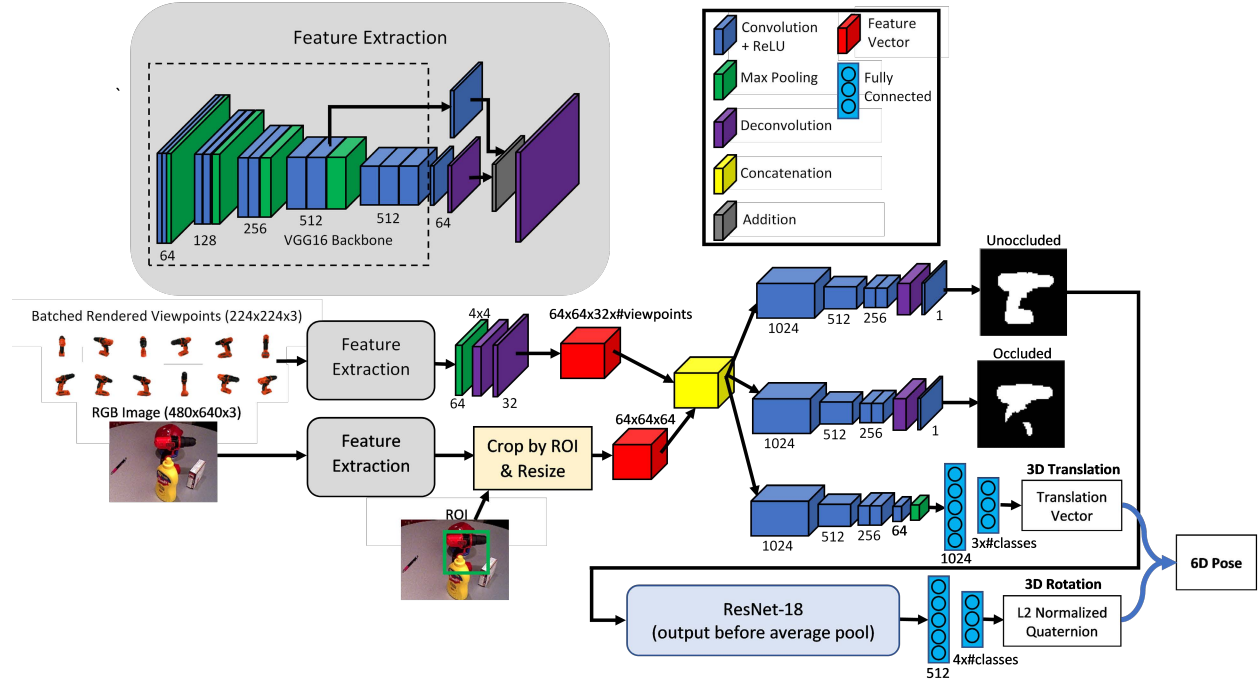


Figure 2.1: Overview of the SilhoNet pipeline for silhouette prediction and 6D object pose estimation. The 3D translation is predicted in parallel with the silhouettes. The predicted unoccluded silhouette is fed into a second stage network to predict the 3D rotation vector.

tion of the energy function gives the object pose. [127] trained a Conditional Random Field (CRF) to output a number of pose-hypotheses from a dense pixel wise object coordinate prediction map computed by a random forest. A variant of ICP was used to derive the final pose estimate. While these learning-based methods are powerful, efficient, and give state-of-the-art results, they rely on RGB/D data to estimate the object pose.

There are several recent works extending deep learning methods to the problem of 6D object pose estimation using RGB data only. [151, 177] used a CNN to predict 2D projections of the 3D object bounding box corners in the image, followed by a PnP algorithm to find the correspondences between the 2D and 3D coordinates and compute the object pose. [188] proposed a multistage, multibranch network with a Hugh Voting scheme to directly regress the 6D object pose as a 3D translation and a unit quaternion orientation. [97] predicted 2D bounding box detections with a pool of candidate 6D poses for each box. After a pose refinement step, they choose the best candidate pose for each box. [113] used an end-to-end CNN framework to predict discretely binned rotation and translation values with corrective delta offsets. They proposed a novel method for infusing the class prior into the learning process to improve the network performance



for multi-class prediction. [117] proposed a deep-learning-based iterative matching algorithm for RGB based pose refinement, which achieves performance close to methods that use depth information with ICP and can be applied as post refinement to any RGB based method. These RGB-based pose estimation methods demonstrate competitive performance against state-of-the-art approaches that rely on depth data. Our work extends these recent advancements in monocular pose estimation by combining the power of deep learning with prior knowledge of the object model to estimate pose from silhouette predictions. Also, our method provides information about how the object is visually occluded in the form of occlusion masks, which can be projected onto the object model, given the predicted 3D orientation.

## 2.3 Method

We introduce a novel method that operates on monocular color images to estimate the 6D object pose. The 3D orientation is predicted from an intermediate unoccluded silhouette representation. The method also predicts an occlusion mask which can be used to determine which parts of the object model are visible in the image. The method operates in two stages, first predicting an intermediate silhouette representation and occlusion mask of an object along with a vector describing the 3D translation and then regressing the 3D orientation quaternion from the predicted silhouette. The following sections describe our method in detail.

### 2.3.1 Overview of the Network Pipeline

Figure 2.1 presents an overview of the network pipeline. The input to the network is an RGB image with ROI proposals for detected objects and the associated class labels. The first stage uses a VGG16 [167] backbone with deconvolution layers at the end to produce a feature map from the RGB input image. This feature extraction network is the same as used in PoseCNN [188]. Extracted features from the input image are concatenated with features from a set of rendered object viewpoints and then passed through three network branches, two of which have identical structure to predict a full unoccluded silhouette and an occlusion mask. The third branch predicts a 3D vector encoding the object center in pixel coordinates and the range of the object center from the camera. The second stage of the network passes the predicted silhouette through a ResNet-18 [70] architecture with two fully connected layers at the end to output an L2-normalized quaternion, representing the 3D orientation.



### 2.3.1.1 Predicted ROIs

We trained an off-the-shelf Faster-RCNN implementation from Tensorpack [187] on the YCB-video dataset [188] to predict ROI proposals. This network was trained across two Titan V GPUs for 3,180,000 iterations on the training image set with the default parameters and without any synthetic data augmentation. The ROI proposals are provided as input to the network after the feature extraction stage, where they are used to crop the corresponding region out of the input image feature map. The cropped feature map is then resized to a width and height of 64x64 by either scaling down the feature map or using bi-linear interpolation to scale it up.

### 2.3.1.2 Rendered Model Viewpoints

We were able to boost the silhouette prediction performance by generating a set of synthetic pre-rendered viewpoints associated with the detected object class as an additional input to the first stage of the network. For each class, We rendered a set of 12 viewpoints from the object model, each with dimension 224x224. These viewpoints were generated using Phong shading at azimuth intervals from 0° to 300° with elevation angles of -30° and 30°. As the intermediate goal is silhouette prediction, these synthetic renders are able to capture the shape and silhouette of real objects, in different orientations, despite the typical domain shift in the visual appearance of simulated objects [33].

All the viewpoints for the detected object class are passed through the feature extraction stage and then resized to 64x64 with channel dimension 32 by passing them through a max-pooling layer with width 4 and stride 4, followed by two deconvolution layers that each increase the feature map size by 4. In this implementation, we extracted the feature maps of the rendered viewpoints on-the-fly for each object detection. However, to reduce computation time, these extracted feature maps can be precomputed and stored offline. These rendered viewpoint feature maps were provided to the network by stacking them on the channel dimension and then concatenating with the cropped and resized input image feature map (Fig.2.1).

### 2.3.1.3 Silhouette Prediction

The first stage of the network predicts an intermediate silhouette representation of the object as a 64x64 dimensional binary mask. This silhouette represents the full unoccluded visual hull of the object as though it were rendered with the same 3D orientation but centered in the frame. The size of the silhouette in the frame is invariant to the scale of the object in the image and is determined by a fixed distance of the object from the camera at which the silhouette appears to be rendered. This distance is chosen for each object so that the silhouette just fits within the frame for any 3D orientation. Given the smallest field of view of the camera  $A$ , determined by the minimum of the

width and height of the image sensor, and the 3D dimensions of the object as the width, height and depth  $(w, h, d)$ , we calculate the render distance  $r$  as

$$r = 1.05 \frac{\sqrt{w^2 + h^2 + d^2}}{2 \tan(A/2)}. \quad (2.1)$$

This stage of the network also has a parallel branch that outputs a similar silhouette, with only the unoccluded parts of the object visible. We refer to this occluded output as the ‘occlusion mask’.

The first part of the network is a VGG16 feature extractor [167], which generates feature maps at 1/2, 1/4, 1/8, and 1/16 scale. The 1/8 and 1/16 scale feature maps both have an output channel dimension of 512. The channel dimension for both is reduced to 64 using two convolution layers, after which the 1/16 scale map is upsampled by a factor of 2 using deconvolution and then summed with the 1/8 scale map. The summed map is upsampled by a factor of 8 using a second deconvolution to get a final feature map of the same dimension as the input image with a feature channel width of 64 (Fig.2.1).

After the input image is passed through the feature extractor, the input ROI proposal for the detected object is used to crop out the corresponding area of the resulting feature map and resize it to 64x64. This feature map is concatenated with the rendered viewpoint feature maps, resulting in a single feature vector matrix with size 64x64x448.

The feature vector matrix is fed into two identical network branches, one of which outputs the silhouette prediction and the other outputs the occlusion mask. Each branch is composed of 4 convolution layers, each with a filter width, channel dimension, and stride of (2, 1024, 1), (2, 512, 2), (3, 256, 1), and (3, 256, 1) respectively, followed by a deconvolution layer with filter width, channel dimension, and stride of (2, 256, 2). The output of the deconvolution layer is fed into a dimension reducing convolution filter with a single channel output shape of 64x64. A sigmoid activation function is applied at the output to produce a probability map.

### 2.3.1.4 3D Translation Regression

The 3D translation is predicted as a three dimensional vector, encoding the object center location in pixel coordinates and range from the camera center in meters. Other region proposal based pose estimation methods [188, 48] regress the Z coordinate directly from the ROI. However, this suffers from ambiguities. If an object at a given range is shifted along the arc formed by the circle with the camera center as the focus, the Z coordinate will change while the object appearance in the shifted ROI will be unchanged. This ambiguity is especially prevalent in wide field of view cameras. By predicting the object range rather than directly regressing the Z coordinate, our method does not suffer from ambiguities and can recover the Z coordinate with good accuracy. Given the camera focal length  $f$ , the pixel coordinates of the object center  $(px, py)$  with respect to the image center,

and the range  $r$  of the object center from the camera center, similar triangles can be used to show that the 3D object translation,  $(X, Y, Z)$ , can be recovered as

$$Z = \frac{rf}{\sqrt{px^2 + py^2 + f^2}}, \quad (2.2)$$

$$X = Z * px/f, \quad Y = Z * py/f. \quad (2.3)$$

The pixel coordinates of the object center are predicted with respect to the ROI box as an offset from the lower box edge bounds normalized by the box dimensions and passed through a sigmoid function. Given a ROI with width  $w$ , height  $h$ , lower x and y coordinate bounds  $(bx, by)$ , the coordinates of the image principal point  $(cx, cy)$  and the predicted normalized output from the network  $(nx, ny)$ , the object center pixel coordinates  $(px, py)$  are recovered as

$$rx = -\log(1/nx - 1), \quad ry = -\log(1/ny - 1), \quad (2.4)$$

$$px = bx + rx * w - cx, \quad py = by + ry * h - cy. \quad (2.5)$$

Note that only the pixel coordinates of the object center are offset by the principal point in these equations. While other methods limit the prediction of the object center to lie within the ROI [188] or treat the ROI center as the coordinates of the object center [48], if the object is not completely in the image frame, the center may not lie within the ROI, and because ROI predictions are imperfect, the object center rarely lies at the ROI center. Our formulation for predicting the object center does not constrain the point to lie within the ROI and is robust to imperfect ROI proposals.

The translation prediction branch is identical to the silhouette prediction branches, except the deconvolution layer is replaced with a 5th convolution layer with filter width, channel dimension, and stride of  $(2, 64, 2)$  followed by max pooling. The output is fed into a fully connected layer of dimension 1024 followed by a fully connected layer of dimension  $3 \times (\# \text{ classes})$ , where each class has a separate output vector. The predicted vector for the class of the detected object is extracted from the output, and the first two entries are normalized with a sigmoid activation (Fig.2.1).

### 2.3.1.5 3D Orientation Regression

We use a quaternion representation for the 3D orientation, which can represent arbitrary 3D rotations in continuous space as a unit vector of length 4. The quaternion representation is especially attractive, as it does not suffer from gimbal lock like the Euler angle representation. Predicting orientation from a ROI gives rise to visual ambiguities, as the true object orientation varies depending on the location within the image from which the ROI is extracted. To address these ambiguities, the network predicts the apparent orientation as though the ROI were extracted from the center of

the image. Given the predicted object translation, the true orientation is recovered by applying a pitch,  $\delta\theta$ , and roll,  $\delta\phi$ , adjustment to the predicted orientation. These adjustments are calculated as

$$\delta\theta = \arctan(X/Z), \quad \delta\phi = -\arctan(Y/Z), \quad (2.6)$$

The second stage of the network takes in the predicted silhouette probability maps, thresholded at some value into binary masks, and outputs a quaternion prediction for the object orientation. This stage of the network is composed of a ResNet-18 [70] backbone, with the layers from the average pooling and below replaced with two fully connected layers. The last fully connected layer has output dimension  $4 \times (\# \text{ classes})$ , where each class has a separate output vector. The predicted vector for the class of the detected object is extracted from the output and normalized using an L2-norm (Fig.2.1).

Because the silhouette representation of objects is featureless, this method treats symmetries in object shape as equivalent symmetries in the 3D orientation space. In many robotic manipulation scenarios, this is a valid assumption. For example, a tool such as a screwdriver that may not be symmetric in RGB feature space is symmetric in shape and equivalently symmetric in grasp space.

By regressing the 3D orientation from an intermediate silhouette representation, we were able to train this stage of the network using only synthetically rendered silhouette data. In the results, we show that the network generalized well to predicting pose on real data, showing that this intermediate representation as an effective way to bridge the domain shift between real and synthetic data.

### 2.3.1.6 Occlusion Prediction

Given the predicted apparent 3D orientation of the object, the predicted occlusion mask can be projected onto the object model to determine which portions of the model are visible in the scene. Mathematically, this can be accomplished by taking every vertex  $v$  of the object model and projecting it onto the occlusion mask. We construct a transform matrix  $T$  with a z translation component equal to the render distance  $r$  for the corresponding object class and the x and y translation components set to 0. The rotation sub-matrix is formed from the predicted apparent orientation. Using the following equation, each vertex of the object model can be projected onto the occlusion mask, which is scaled up to fit the minimum dimension of the input image,

$$\gamma = KTv \quad (2.7)$$

where  $K$  is the camera intrinsic matrix,  $v$  is the 3D homogeneous coordinates of the vertex in the object frame, and  $\gamma$  is the homogeneous pixel coordinates of the projected vertex on the scaled occlusion mask. Not accounting for object self occlusions, those vertices which lie on the visible

portion of the occlusion mask are predicted to be visible in the image.

### 2.3.2 Dataset

We evaluated our method on the YCB-video dataset [188], which consists of 92 video sequences composed of 133,827 frames, containing a total of 21 objects, appearing in different arrangements with varying levels of occlusion. Twelve of the video sequences were withheld from the training set for validation and testing. In the silhouette space, the objects in this dataset are characterized by five different types of symmetry: non-symmetric, symmetric about a plane, symmetric about two perpendicular planes, symmetric about an axis, symmetric about an axis and a plane. We applied a rotation correction to the coordinate frame of all objects that exhibit any form of symmetry so that each axis or plane of symmetry aligns with a coordinate axis. Ground truth quaternions were generated from the labeled object poses such that only one unique quaternion is associated with every viewpoint that produces the same visual hull. Having a consistent quaternion label for all matching silhouette viewpoints enabled the pose prediction network to be trained effectively for all types of object symmetries using a very simple distance loss function.

Supplementing the real image data in the YCB-video dataset are 80,000 synthetically rendered images, with all of the 21 objects appearing in various combinations and random poses over a transparent background. We supplement the training data by randomly sampling images from the COCO-2017 dataset [24] and applying them as background to these synthetic images at training time.

### 2.3.3 Network Training

All networks were trained with the Adam optimizer on either a Titan V or Titan X GPU. The VGG16 backbone was initialized with ImageNet pre-trained weights, and the silhouette prediction network without the translation branch was trained using cross entropy loss with a batch size of 6 for 325,000 iterations. We trained the network with ground truth ROIs and tested against both ground truth ROIs and predicted ROIs from a Faster-RCNN network [187] trained on the YCB-video dataset. The translation prediction branch was then added, and all network weights not part of this branch were frozen. The translation branch was trained for 230,000 iterations using an l2 loss. All network weights were then unfrozen and the entire network was fine-tuned for 208,000 iterations.

The orientation regression network was trained using the following log distance function between the predicted and ground truth quaternions

$$QLoss(\tilde{q}, q) = \log(\epsilon + 1 - |\tilde{q} \cdot q|), \quad (2.8)$$

where  $q$  is the ground truth quaternion,  $\tilde{q}$  is the predicted quaternion, and  $\epsilon$  is a small value for stability, in our case  $e^{-4}$ . The orientation regression network was trained for 380,000 iterations with a batch size of 16, using only perfect ground truth silhouettes for training. Testing was done on the predicted silhouettes from the first stage network.

To reduce overfitting during training of the networks, dropout was applied at a rate of 0.5 before the last deconvolution layer of the feature extraction network, on the fourth convolutional layer of each silhouette prediction branch, and after the max pooling layer of the translation branch. During training of the orientation regression network, dropout was applied at a rate of 0.8 before the first fully connected layer. As a further strategy to reduce overfitting and extend the training data, the hue, saturation, and exposure of the training images were randomly scaled by a factor of up to 1.5

Table 2.1: Mean IoU accuracy for predicted silhouettes

Object	Unoccluded GT ROI	Occluded GT ROI	Unoccluded Pred ROI	Occluded Pred ROI
master_chef_can	96.75	91.08	96.84	88.42
cracker_box	92.94	82.20	90.50	68.91
sugar_box	94.28	91.79	92.32	88.27
tomato_soup_can	96.41	93.25	96.73	94.09
mustard_bottle	95.02	94.49	94.68	94.25
tuna_fish_can	95.96	93.81	96.06	93.95
pudding_box	90.08	79.57	88.73	71.58
gelatin_box	95.72	94.65	95.31	94.78
potted_meat_can	92.53	87.11	93.77	87.18
banana	88.48	87.23	81.76	78.05
pitcher_base	94.63	93.80	94.58	93.71
bleach_cleanser	92.48	89.64	91.74	87.95
bowl	79.74	67.01	82.03	76.63
mug	93.92	86.84	90.97	84.24
power_drill	86.61	85.08	78.57	73.64
wood_block	89.30	74.92	90.72	78.84
scissors	52.20	65.12	61.70	65.97
large_marker	84.37	84.15	83.96	82.65
large_clamp	84.03	79.50	85.73	80.93
extra_large_clamp	86.16	82.34	76.13	70.14
foam_brick	91.00	86.17	89.99	82.78
ALL	89.17	85.23	88.23	82.71

## 2.4 Results

The following sections present the performance of SilhoNet, tested on the YCB-video dataset [25]. Section A presents the accuracy of the silhouette prediction stage, and section B compares the 6D pose estimation performance of SilhoNet against the performance of PoseCNN [188]. We also compare performance against the method in [113] for RGB input.

Table 2.2: Mean 3D orientation error in degrees. The Sym tag indicates orientation predictions are reduced by geometric symmetries.

Object	RGB				RGB-D	
	PoseCNN [188]	PoseCNN Sym [188]	SilhoNet-GT ROI	SilhoNet-Pred ROI	PoseCNN +ICP [188]	PoseCNN +ICP Sym [188]
master_chef_can	50.71	7.57	<b>1.11</b>	1.21	51.88	1.06
cracker_box	19.69	19.69	<b>9.53</b>	19.86	9.51	9.23
sugar_box	<b>9.29</b>	<b>9.29</b>	11.50	12.28	1.06	1.06
tomato_soup_can	18.23	8.40	<b>1.82</b>	1.91	31.74	1.98
mustard_bottle	9.94	9.59	<b>5.07</b>	5.78	2.72	2.22
tuna_fish_can	32.80	12.74	1.50	<b>1.46</b>	37.70	6.28
pudding_box	<b>10.20</b>	<b>10.20</b>	18.39	20.95	2.27	2.26
gelatin_box	<b>5.25</b>	<b>5.25</b>	8.48	12.52	1.03	1.03
potted_meat_can	28.67	19.74	10.93	<b>7.27</b>	23.06	13.93
banana	15.48	15.48	<b>5.70</b>	16.29	12.17	12.17
pitcher_base	11.98	11.98	<b>6.61</b>	6.64	2.55	2.55
bleach_cleanser	<b>20.85</b>	<b>20.85</b>	48.42	51.28	11.02	11.02
bowl	75.53	75.53	53.95	<b>49.95</b>	55.71	55.71
mug	19.44	19.44	<b>7.02</b>	18.14	23.11	23.11
power_drill	<b>9.91</b>	<b>9.91</b>	10.66	30.54	1.64	1.64
wood_block	23.63	23.63	<b>23.23</b>	25.52	15.12	15.12
scissors	<b>43.98</b>	<b>43.98</b>	154.82	155.53	30.77	30.76
large_marker	92.44	13.59	10.72	<b>10.44</b>	84.34	3.38
large_clamp	38.12	38.12	6.03	<b>3.54</b>	33.99	33.99
extra_large_clamp	34.18	34.18	<b>7.30</b>	29.18	37.89	37.89
foam_brick	22.67	22.67	17.36	<b>13.84</b>	18.82	18.82
ALL	27.79	17.82	<b>13.48</b>	16.04	24.54	10.94

Table 2.3: Mean 3D translation error in centimeters

Object	RGB			RGB-D
	PoseCNN [188]	SilhoNet-GT ROI	SilhoNet-Pred ROI	PoseCNN +ICP [188]
master_chef_can	3.29	3.14	<b>3.02</b>	0.52
cracker_box	4.02	<b>2.38</b>	5.24	1.28
sugar_box	3.06	<b>1.67</b>	2.10	0.26
tomato_soup_can	3.02	<b>2.24</b>	2.40	0.33
mustard_bottle	1.72	<b>1.41</b>	1.65	0.14
tuna_fish_can	2.41	<b>1.49</b>	1.57	0.37
pudding_box	3.69	<b>1.91</b>	7.15	0.31
gelatin_box	2.49	<b>0.79</b>	1.09	0.19
potted_meat_can	3.65	<b>2.74</b>	4.30	1.06
banana	<b>2.43</b>	2.59	4.12	0.63
pitcher_base	4.43	<b>1.29</b>	1.31	0.14
bleach_cleanser	4.86	3.99	<b>3.60</b>	0.49
bowl	5.23	4.08	<b>3.30</b>	3.73
mug	4.00	<b>1.43</b>	2.61	0.97
power_drill	4.59	<b>3.19</b>	6.77	0.17
wood_block	6.34	<b>3.23</b>	5.59	2.68
scissors	6.40	<b>2.59</b>	9.91	1.49
large_marker	3.89	<b>2.31</b>	3.24	0.89
large_clamp	9.79	<b>3.51</b>	6.27	5.25
extra_large_clamp	8.36	<b>2.12</b>	4.86	4.19
foam_brick	2.48	<b>2.31</b>	3.98	0.48
ALL	4.16	<b>2.45</b>	3.49	1.06

Table 2.4: Area under accuracy-threshold curve for 6D pose evaluation using ADD-S metric

Object	PoseCNN [188]	SilhoNet-GT ROI	SilhoNet-Pred ROI	MCN [113]	MV5-MCN [113]
master_chef_can	82.6	83.6	84.0	87.8	<b>90.6</b>
cracker_box	77.2	<b>88.4</b>	73.5	64.3	72.0
sugar_box	84.0	<b>88.8</b>	86.6	82.4	87.4
tomato_soup_can	81.7	89.4	88.7	87.9	<b>91.8</b>
mustard_bottle	91.1	91.0	89.8	92.5	<b>94.3</b>
tuna_fish_can	84.0	<b>89.9</b>	89.5	84.7	89.6
pudding_box	79.4	<b>89.1</b>	60.1	51.0	51.7
gelatin_box	85.7	<b>94.6</b>	92.7	86.4	88.5
potted_meat_can	78.5	84.8	78.8	83.1	<b>90.3</b>
banana	85.9	<b>88.7</b>	80.7	79.1	85.0
pitcher_base	76.9	<b>91.8</b>	91.7	84.8	86.1
bleach_cleanser	71.5	72.0	73.6	76.0	<b>81.0</b>
bowl	63.5	72.5	79.6	76.1	<b>80.2</b>
mug	78.1	92.1	86.8	91.4	<b>93.1</b>
power_drill	72.7	<b>82.9</b>	56.5	76.0	81.1
wood_block	61.5	<b>79.2</b>	66.2	54.0	58.4
scissors	56.6	78.3	49.1	71.6	<b>82.7</b>
large_marker	68.3	<b>83.1</b>	75.0	60.1	66.3
large_clamp	55.3	<b>84.5</b>	69.2	66.8	77.5
extra_large_clamp	42.8	<b>88.4</b>	72.3	61.1	68.0
foam_brick	86.7	<b>88.4</b>	77.9	60.9	67.7
ALL	75.3	<b>85.8</b>	79.6	75.1	80.2

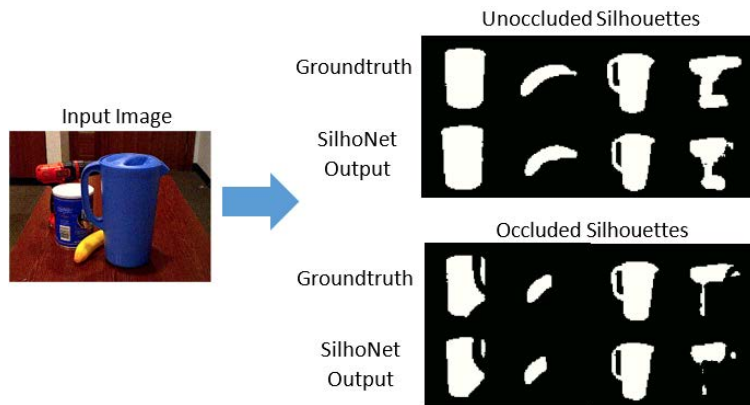


Figure 2.2: Example prediction of occluded and unoccluded silhouettes from a test image



### 2.4.1 Silhouette Prediction

We tested the performance of the silhouette prediction stage of SilhoNet with both ground truth ROI inputs from the YCB dataset and predicted ROI inputs from the FasterRCNN network. Figure 2.2 shows an example of the silhouette predictions for one of the images in the test set. Table 2.1 presents the accuracy for the occluded and unoccluded silhouette predictions, measured as the mean intersection over union (IoU) of the predicted silhouettes with the ground truth silhouettes. Overall, the performance degrades by a few percent when the predicted ROIs (Pred ROI) are provided as input rather than the ground truth (GT ROI), but in general, the predictions are robust to the ROI input.

### 2.4.2 6D Pose Regression

We compare the accuracy of the 6D pose predictions from SilhoNet against the published results of PoseCNN. We include the performance of PoseCNN with depth based Iterative Closest Point (ICP) refinement as an RGB-D method reference point. To provide greater insight into the model performance, we first analyze the orientation and translation prediction results separately. Because our method predicts orientation in a space reduced by geometric symmetries, we compare against the performance of PoseCNN both before and after reducing the PoseCNN predictions to the same symmetry invariant space. Figure 2.3 shows the accuracy curves for PoseCNN before and after ICP refinement and SilhoNet with YCB ground truth ROI input (GT ROI) and FasterRCNN predicted ROI input (Pred ROI). SilhoNet shows a visually higher area under the accuracy curve than PoseCNN before ICP refinement. The improvement of SilhoNet in area under the accuracy curve is especially obvious for the rotation angle prediction accuracy, demonstrating the effectiveness of the intermediate silhouette representation for orientation prediction. Table 2.2 presents the mean orientation errors for each class across both the PoseCNN and SilhoNet methods. The classes with the worst prediction accuracy for SilhoNet relative to PoseCNN are "bleach\_cleanser" and "scissors". SilhoNet treats both of these objects as non-symmetric in silhouette space, but the shape of both objects is nearly planar symmetric, especially if they are partially occluded, so pose predictions from silhouettes may be easily confused. SilhoNet shows the strongest performance on cylindrical objects like "master\_chef\_can" and "tomato\_soup\_can", which exhibit the highest reduction in orientation space through symmetries. Across every type of geometric symmetry exhibited in the dataset, there are objects where SilhoNet performs significantly better than PoseCNN, demonstrating the general effectiveness of silhouettes as an intermediate representation for object 3D orientation estimation. The orientation prediction accuracy of SilhoNet is reduced when predicted ROIs are provided as input, but overall there is still significant improvement over PoseCNN, showing that SilhoNet is robust to the quality of region proposals.

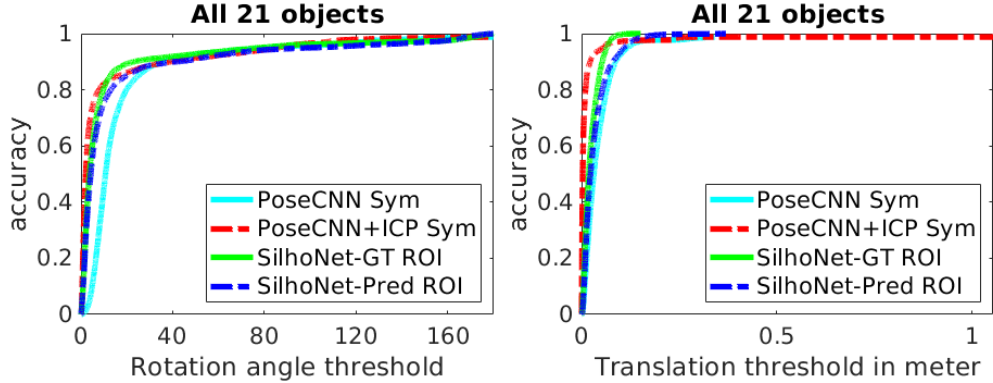


Figure 2.3: 6D pose accuracy curve across all objects in the YCB-video dataset. Accuracy is percentage of errors less than the error threshold. The PoseCNN orientation predictions are reduced by the same geometric symmetries as SilhoNet.

Table 2.3 presents the mean translation errors for each object class. SilhoNet outperforms PoseCNN across most classes before ICP refinement. The translation prediction accuracy of SilhoNet is also reduced when predicted ROIs are provided as input, but there is still significant improvement over PoseCNN.

In Table 2.4, we compare the full 6D pose prediction performance of SilhoNet against PoseCNN (without depth refinement) [188] and another recently proposed RGB based method [113]. We use the area under the accuracy-threshold curve (ADD-S) metric proposed in [188]. The ADD-S metric is particularly suited to SilhoNet, as it is invariant to geometric symmetries. We note that the method MV5-MCN [113] is a multiview variant of MCN [113] and requires that each input image is labelled with a camera pose. Typically, labelling camera pose would require some extra sensory input besides a monocular RGB camera in order to disambiguate the scale of motion in a SLAM system. The results in the table show that SilhoNet outperforms PoseCNN and MCN by a large margin with both ground truth and predicted ROIs as input. SilhoNet performs better than MV5-MCN with ground truth ROIs as input and performs on par with predicted ROIs as input. Overall, SilhoNet shows a significant performance improvement over related methods when the input is limited to RGB images only.

As an ablation study, we performed an experiment to determine the contribution of the rendered viewpoint image priors to the network performance. Table 2.5 shows the results of this experiment. Note that the network was trained without the translation prediction branch, and ground truth ROIs were given as input. When no rendered viewpoints are provided as a prior input, the network performance drops with nearly twice the error in orientation predictions for both shared and class specific output. However, providing more than one rendered viewpoint image as a prior input does not significantly affect the network performance. This result motivates future investigation into how the network incorporates the rendered viewpoint inputs into the learned network structure.

Table 2.5: Silhouette and orientation accuracy vs # of model images

# Model Images	Unoccluded (IoU)	Occluded (IoU)	Mean Angle Error (Degrees)
0 (class output)	78.85	77.15	29.90
0 (shared output)	77.87	74.95	31.32
1	89.20	86.31	14.27
4	89.38	86.05	13.60
6	89.54	86.36	15.19
12	88.68	85.25	13.48

## 2.5 Conclusion

In this chapter, we presented a method for object 6D pose estimation from monocular camera images, where detected object ROI proposals are provided as input. We showed that this method outperforms the state-of-the-art PoseCNN network and another recent RGB based method across the majority of object classes in the YCB-video dataset. The most significant contribution of this method is an intermediate silhouette representation for object viewpoints, which is shown to be a robust and effective abstraction from which to predict 3D orientation and also greatly reduces the sim-to-real domain shift when learning a model on synthetic data. This silhouette abstraction is demonstrated to improve accuracy of orientation predictions over previous methods. Also, by using an intermediate silhouette representation for detected objects, this method enables determining which parts of an object model are unoccluded in the scene. We proposed a novel strategy for predicting 3D translation from ROI proposals, which does not suffer from ambiguities in apparent viewpoint, leading to improved translation accuracy over previous methods.

## CHAPTER 3

# 6D Object Pose Estimation in Fisheye and Omnidirectional Images

### 3.1 Motivation

The advantages of fisheye imaging systems in robotics applications has long been recognized. With technological improvements in imaging sensor resolution and dynamic range, fisheye cameras can capture significantly greater information about the surrounding environment without appreciably increasing the imaging sensor footprint, compared to their perspective model counterparts. However, little work has been done on applying CNN based methods to the problem of 6D object pose estimation on full fisheye images. Dealing with fisheye images is challenging, due to the large distortions and viewpoint ambiguities arising from the wide field of view. We address the problem of 6D object pose estimation in full fisheye images by proposing a method whereby the image is first projected to the surface of a sphere, where we mathematically define a consistent apparent viewpoint which the network is trained to predict. The true orientation relative to the fisheye frame can then be recovered using the predicted translation. The gnomonic projection is used in our method to undistort the ROI from the sphere surface, and we investigate applying this projection both before and after the feature extraction stage.

In summary, the main contribution of this chapter is a framework for adapting ROI-based networks for predicting 6D object pose from monocular images to work on full fisheye images, through an intermediate mapping onto a sphere and ROI processing through the gnomonic projection. This adaptation is demonstrated with the SilhoNet method presented in [14].

The rest of this chapter is organized in the following sections: Section 3.2 discusses related work; Section 3.3 presents our method for adapting SilhoNet to the fisheye domain; Section 3.4 presents the experimental results; and Section 3.5 concludes the chapter.

## 3.2 Related Work

In general, state-of-the-art works that apply CNN methods to full fisheye images process the raw images directly through the network without special consideration of the fisheye distortions [46, 64, 156]. These networks are mostly applied to the problems of segmentation or ROI detection in the fisheye images. Due to the sparsity of available benchmarking datasets for fisheye images, these works report their results on synthetic datasets, generated by projecting perspective images to distorted fisheye images. [195] used a CNN in the prediction of ground vehicle positions relative to an aerial fisheye imaging platform. They directly train the CNN on the raw fisheye images to generate ROI proposals. They assume the detected object is on the ground plane and fuse measurements from height and orientation sensors on the camera platform to recover only the object’s 3D translation in the world. [158] extended the Cascaded Pose Regression algorithm to estimate the 3D pose of mice in fisheye images from detected 3D keypoints. However, their method incorporates priors about the structured lab environment, and the fisheye camera is fixed in the scene, allowing them to easily segment the mice from the background image. In contrast to these works, our method incorporates knowledge of the fisheye distortion model through a spherical mapping, which improves network performance and is also necessary to create visually consistent pose annotations which can be regressed directly from ROI proposals across the full fisheye field of view. Further, we report the performance of our method on a real fisheye dataset captured in a natural unstructured environment.

Closely related to fisheye image processing is the extensive body of work on omni-directional imaging, as both fisheye and omni-directional image distortions can be represented on a sphere. Beyond naively applying CNNs directly to a flattened equirectangular projection of an omni-directional image, which has been shown to suffer from the nonlinear distortions of the spherical mapping to the plane and attain sub-optimal performance [163], the methods of dealing with omni-directional distortions can be roughly categorized under three approaches: generating multiple perspective projections from the sphere, such as cube map, and processing each projection separately through the CNN [130]; adapting the kernel sampling locations based on a spherical distortion model or a learned mapping [194, 40, 174, 175]; re-sampling the spherical image based on a uniform sampling geometry such as the icosahedron, and processing the spherical representation with specialized convolution operations [89, 53, 108, 192]; or transforming the spherical feature signals and convolution operations into the spectral domain, typically by representation of the spherical image as a graph [147, 98, 38]. Methods that operate on multiple perspective projections suffer from discontinuities at the projection borders, due to variance in feature appearance on different tangent plane mappings. Methods that operate on graphical representations of the sphere in the spectral domain are memory limited in scaling to full resolution images and have

some level of rotation invariance in the convolution response function, which is undesirable when regressing 6D object pose. Methods that re-sample the convolution kernel sampling location based on a learned or distortion based mapping are most relevant to our work. The methods of [40] and [194] sample regular kernel locations on a tangent plane and then project the sampling locations to the spherical surface, encoding the spherical distortions directly into the convolution operation. [193] adapts a region proposal network with the distortion aware convolutions of [40, 194] in a two-stage architecture to predict region proposals from omni-directional images. However, these distortion aware convolutions are designed to operate on full 360° images. Because fisheye images represent only a partial view of the sphere, they can also be analyzed under different planar projections than omni-directional images. Further, application of omni-directional CNNs to 6D object pose estimation is so far lacking in the literature. Our method takes inspiration from these prior works [40, 194] that incorporate a mapping to a spherical surface and the Gnomonic projection to a tangent plane to deal with feature distortions in omni-directional images. The main technical contribution of our work is the mathematical formulation of applying a spherical mapping and the Gnomonic projection to the problem of 6D object pose estimation in wide field-of-view imagery. Though we developed the method assuming the equidistant fisheye projection model, the formulation is valid for any camera projection that can be mapped to a spherical surface, including omni-directional images.

The body of work applying CNN methods to underwater imagery is mostly limited to the problems of species detection and classification [55, 36, 152, 118, 101, 189, 124, 159, 129], or underwater image correction [115, 114] on perspective images. [105] used a simple color distortion model based on image depth to generate a synthetic dataset of omni-directional images that were color cast as though captured underwater. They trained a distortion aware CNN to predict image depth from an omni-directional image, and reported results on their synthetic dataset. While they did not test with real omni-directional data, the perspective image equivalent of their method performed very poorly on real underwater images. Most related to our work in the underwater domain is the work of [87], who proposed a CNN based method for underwater object detection and pose estimation, using a synthetic dataset generated from CAD models to train the network. However, the objects used in their dataset were very simple, and their tests were limited to tank environment with high contrast between the object models and the scene background. Further, they only regressed the 3D orientation of the detected objects. Also related to our work is [137], where a PoseNet CNN was trained to regress the 6D pose of a mock-up sub-sea connector relative to a small ROV. The dataset was collected in a tank environment, with high contrast between the connector target and the low featured background. In contrast to these works, our method addresses the problem of full 6D object pose estimation from monocular underwater images captured in wild unstructured environments. Further, our method is applied to full view fisheye images, which capture a signif-

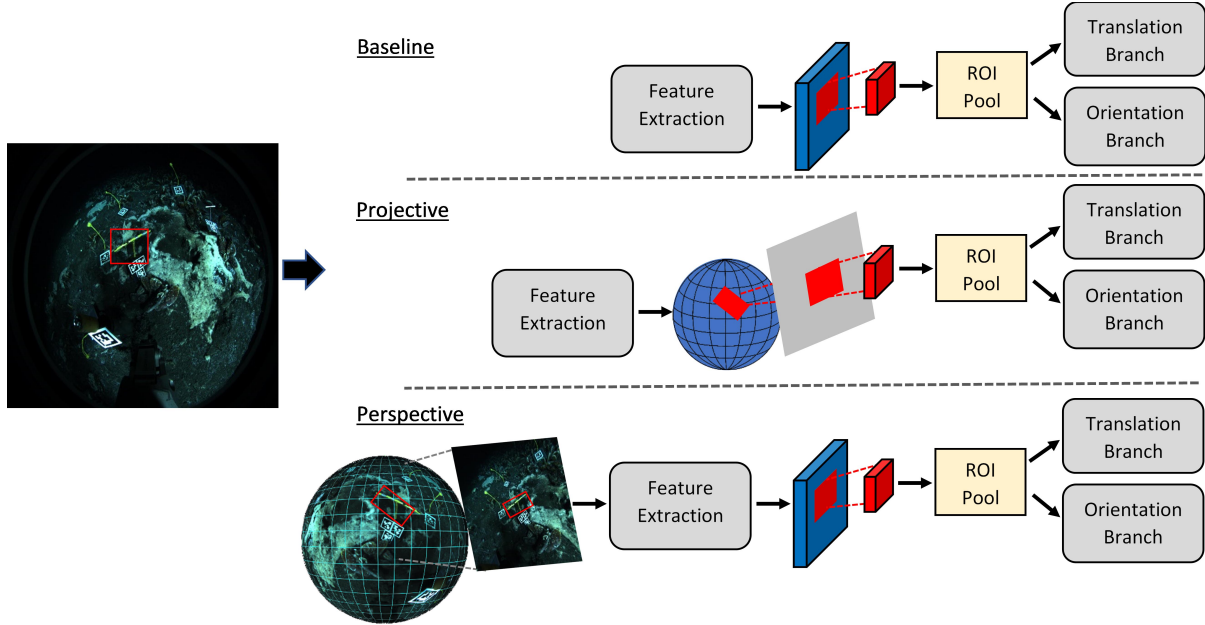


Figure 3.1: Overview of the three different SilhoNet adaptations for processing full fisheye images. The Baseline method processes the raw fisheye image directly through the unmodified network. The Projective variant processes the raw fisheye image through the feature extraction stage and then projects the features within the ROI through a spherical mapping to the tangent plane centered on the ROI, before processing the features through the ROI-pooling stage. The Perspective adaptation maps the fisheye image to a sphere and then generates a virtual perspective image for each object detection using a gnomonic projection, centered on the ROI. Each virtual image is then processed through the network.

icantly greater field of view over perspective images. Also, our dataset is composed with visually challenging handle objects used to manipulate ROV tools in real life applications.

### 3.3 Method

Special care must be taken in regressing 6D pose from full fisheye images, as there can be large distortions and ambiguity in the object viewpoint (Fig. 3.2). In the following sections, we outline how we use an intermediate spherical representation and the gnomonic projection to attain visually consistent pose annotations, followed by an overview of three different adaptations of the SilhoNet method [14] for 6D pose prediction from full fisheye images (Fig.3.1).

#### 3.3.1 Spherical Mapping and Gnomonic Projection

While a class of different projection models exist for fisheye cameras [4], the model followed by the camera system used in this work, and the most common model in practice, is the equidistant

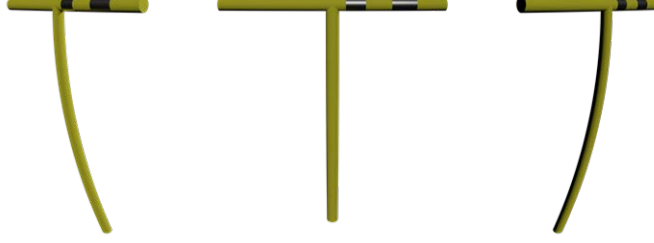


Figure 3.2: These objects have the same orientation relative to the rendered fisheye image frame but different translations, resulting in drastically different apparent orientations. Also, objects appear more distorted as they move from the image center.

projection

$$R = f\theta \quad (3.1)$$

where  $\theta$  is the angle in radians from a point in the world to the optical axis,  $f$  is the lens focal length, and  $R$  is the radial position of the point projected on the imaging plane. A major challenge of fisheye images when regressing the object orientation is the large space of visual ambiguity. We define the global reference frame as coincident with the fisheye camera frame. As the angle between the object center in the world to the camera optical axis increases, there is increasing discrepancy between the object orientation relative to the global frame and the apparent orientation relative to a cropped ROI (Fig. 3.2). We deal with this visual ambiguity by first mapping the fisheye image onto the unit sphere. The mapping between the pixel coordinates  $(x, y)$  on the fisheye image with focal length  $f$  and the polar coordinates  $(\theta, \phi)$  on the unit sphere is given as

$$r = \sqrt{x^2 + y^2}; \rho = r/f; z = \frac{r}{\tan \rho} \quad (3.2)$$

$$\theta = \sin^{-1} \left( \frac{y}{\sqrt{x^2 + y^2 + z^2}} \right); \phi = \tan^{-1} \frac{x}{z}. \quad (3.3)$$

The inverse mapping can also be calculated by first converting the spherical coordinates to cartesian and then projecting onto the image plane with the fisheye model

$$x_s = \cos \theta \sin \phi; y_s = \sin \theta; z_s = \cos \theta \cos \phi \quad (3.4)$$

$$\rho = \cos^{-1} \left( \frac{z_s}{\sqrt{x_s^2 + y_s^2 + z_s^2}} \right); r = f\rho \quad (3.5)$$

$$x = \frac{x_s r}{\sqrt{x_s^2 + y_s^2}}; y = \frac{y_s r}{\sqrt{x_s^2 + y_s^2}}. \quad (3.6)$$



By mapping the fisheye image to a unit sphere centered on the global origin, we can define the apparent viewpoint of the object as the appearance of the object when projected onto a tangent plane centered on the vector extending from the sphere center to the center of the object. The projection from the sphere onto the tangent plane is known as a gnomonic projection and has a long history in mapping as well as recent application in omni-directional CNN methods [40, 194]. Given a spherical mapping of an image and the tangent plane centered on the sphere at polar coordinates  $(\theta_0, \phi_0)$ , the gnomonic projection of the spherical point  $(\theta, \phi)$  onto the tangent plane is given as

$$x = \frac{\cos \theta \sin (\phi - \phi_0)}{\sin \theta_0 \sin \theta + \cos \theta_0 \cos \theta \cos (\phi - \phi_0)} \quad (3.7)$$

$$y = \frac{\cos \theta_0 \sin \theta - \sin \theta_0 \cos \theta \cos (\phi - \phi_0)}{\sin \theta_0 \sin \theta + \cos \theta_0 \cos \theta \cos (\phi - \phi_0)}, \quad (3.8)$$

and an optimized inverse mapping from the tangent plane onto the sphere is given as

$$\theta = \sin^{-1} \left( \frac{\sin \theta_0 + y \cos \theta_0}{\sqrt{1 + x^2 + y^2}} \right) \quad (3.9)$$

$$\phi = \phi_0 + \tan^{-1} \left( \frac{x}{\cos \theta_0 - y \sin \theta_0} \right), \quad (3.10)$$

where  $x$  and  $y$  are the coordinates of the pixel on the tangent plane normalized by the virtual perspective camera focal length  $f_p$  [41, 185]. The gnomonic projection is core to our method of regressing the object 6D pose from ROI proposals on the distorted fisheye image. The orientation of the object  $R_p$  relative to a virtual perspective camera frame centered on the apparent viewpoint can be calculated as a rotation correction to the object orientation  $R$  that is referenced to the global frame. The rotation correction matrix  $R_{adj}$  can be constructed as follows. First, the polar coordinates  $(\theta_0, \phi_0)$  of the intersection of the virtual camera optical axis with the sphere is calculated based on the 3D translation  $(x, y, z)$  of the object relative to the global frame

$$\theta_0 = \sin^{-1} \left( \frac{y}{\sqrt{x^2 + y^2 + z^2}} \right) \quad (3.11)$$

$$\phi_0 = \tan^{-1} \left( \frac{x}{z} \right). \quad (3.12)$$

The rotation adjustment matrix is then constructed column-wise using the coordinates of the rotated virtual camera frame axes in the global reference frame

$$X = [\cos \phi_0, 0, -\sin \phi_0] \quad (3.13)$$

$$Y = [-\sin \theta_0 \sin \phi_0, \cos \theta_0, -\sin \theta_0 \cos \phi_0] \quad (3.14)$$

$$Z = [\cos \theta_0 \sin \phi_0, \sin \theta_0, \cos \theta_0 \cos \phi_0] \quad (3.15)$$

$$R_{adj} = [X; Y; Z] \quad (3.16)$$

The orientation of the object relative to the virtual camera frame is then given as

$$R_p = R_{adj}R \quad (3.17)$$

The orientation branch of the network is trained to regress the apparent orientation  $R_p$ . The predicted true orientation  $R$  can be recovered using the predicted object translation by constructing the inverse  $R_{adj}$  matrix.

### 3.3.2 SilhoNet Adaptation to Fisheye

We compare three different variants of SilhoNet adapted for processing full fisheye images (Fig.3.1). For all variants, the size of the predicted silhouettes was increased to 128x128, because the handle objects in the UWHandles dataset have very thin features. The translation prediction output was also modified to predict the normalized pixel offset of the object center relative to the ROI directly without passing through a sigmoid function, and the predicted offsets were thresholded to lie within the ROI bounds. Because the dataset does not include segmentation annotations, the occluded silhouette branch was removed from the network. The orientation predictions of the handle objects were also reduced by their shape symmetries, as described in the SilhoNet paper [14]. Under these symmetry reductions, the network predicts orientations unique to shape symmetries only, which is appropriate for many object manipulation tasks, such as grasping tool handles, which are generally agnostic in feature space to how they are grasped. Also, because the symmetry reduction is applied directly to the training labels, no special care is needed to deal with symmetric objects in the training, and a simple distance loss function for orientation regression is used, as in the original method. The annotated ROIs were used as input to the network for both training and testing.

The first variant we consider as a baseline, which is essentially the vanilla SilhoNet architecture with the orientation branch output modified to regress the apparent orientation  $R_p$ , as described in the previous section. All variants of the network retain this prediction strategy. The second variant, which we refer to as "projective", processes the raw fisheye image through the feature extraction stage and then projects the features within the ROI through a spherical mapping to the tangent plane centered on the ROI, using the gnomonic projection. The projected features are then passed to the ROI-pooling stage. The motivating idea behind this projective strategy is that local features do not appear heavily distorted in fisheye images, but the spacial relationship of features across the ROI can be significantly distorted. The local feature map is thus generated directly on the raw

fisheye image and then the spacial relationship of these local features is corrected through the projection onto the tangent plane. This projection operation is implemented as a Tensorflow layer for efficient and simple integration into the original network. The third variant, which we refer to as "perspective", projects a region of the fisheye image to a virtual perspective image centered on the ROI using the gnomonic projection. We chose the virtual image dimension to be 400x400 with a pixel relative focal length of 350. These virtual perspective image parameters are tunable per the target application and should take into consideration the desired field of view of the perspective image, the mapping of the fisheye resolution onto the virtual perspective plane, and the computational efficiency in relation to image size for processing the virtual image through the network. This virtual perspective image is processed through the feature extraction stage and then the ROI is cropped from the center of the feature map and passed to the ROI-pooling stage. Essentially, this method generates a virtual perspective image for each detected object and processes each of these virtual images separately through the network. This methods corrects for the fisheye distortions through the entire network pipeline. However, the computation scales with the number of detected objects, as a separate virtual image is processed for each one.

As a further comparison point, we take each of the three variants described above and replace the silhouette prediction branch with a branch that directly regresses the quaternion orientation, rather than first predicting a silhouette and passing it to a second stage network for orientation prediction. This orientation branch has the same structure as the translation branch, but with the output size equal to  $4 \times (\# \text{ classes})$ . The predicted quaternion for the class of the detected object is extracted from the output and normalized using an L2-norm. These methods which bypass the silhouette prediction to directly regress the orientation are referred to in the following sections by appending "\_direct" to the name of the associated variant: "baseline\_direct", "projective\_direct", and "perspective\_direct".

### 3.3.3 Network Training

The networks were trained with the same loss functions and dropout rates as in [14] on Titan V GPUs. All networks were trained for 400,000 iterations on the training set except for the "perspective\_direct" method which was only trained for 356,000 iterations because of time constraints. Due to GPU memory limitations, the raw fisheye images of dimension 2,448x2,048 were downsampled by a factor of 3 for the baseline and projective variants and by a factor of 2 for the perspective variant. The baseline and projective variants were trained with a batch size of 2 and the perspective variant with a batch size of 3. As with the original SilhoNet method, the second stage network which regresses orientation from silhouettes was trained using only perfect rendered silhouettes.

### 3.3.4 Dataset

We analyzed the method performance on the UWHandles dataset, which is discussed in chapter 6 of this thesis. The dataset is composed of underwater fisheye images of graspable handle objects and was collected in natural seafloor environments of the deep ocean.

## 3.4 Results

The following section presents the performance of the different SilhoNet adaptations on the UWHandles dataset. Table 3.1 and Table 3.2 show the percentage of translation and orientation predictions under different error thresholds, respectively. Table 3.3 shows the overall 6D pose prediction accuracy using the ADD-S metric from [188].

Table 3.1: Percentage of translation predictions under the threshold error, where a higher percentage under a lower threshold means better accuracy.

Method	< 5cm	< 10cm	< 20cm	< 30cm
Baseline	69.88	90.81	98.48	99.92
Projective	71.91	87.35	96.22	98.39
Perspective	74.63	90.61	96.73	98.14
Baseline-Direct	47.55	72.04	94.56	99.21
Projective-Direct	46.71	73.56	93.54	98.36
Perspective-Direct	57.69	81.29	96.11	99.07

Table 3.2: Percentage of orientation predictions under the threshold error, where a higher percentage under a lower threshold means better accuracy.

Method	< 5°	< 10°	< 20°	< 30°
Baseline	12.75	34.77	62.78	75.31
Projective	11.26	35.33	63.03	74.89
Perspective	16.05	39.08	66.08	77.28
Baseline-Direct	22.58	45.58	69.00	81.31
Projective-Direct	28.91	50.45	69.48	83.19
Perspective-Direct	29.81	55.01	74.21	85.02

Table 3.3: Area under accuracy-threshold curve for 6D pose evaluation using ADD-S metric, where a higher area means better accuracy. Proj. is short for Projective and Persp. is short for Perspective

Handle Type	Baseline	Proj.	Persp.	Baseline Direct	Proj. Direct	Persp. Direct
umichhandle	72.71	69.53	78.81	61.70	61.79	64.46
soihandle	71.48	79.98	75.11	47.51	53.33	60.77
whoihandle	61.82	57.34	61.95	48.54	47.39	59.90
ALL	68.65	68.92	71.94	52.57	54.15	61.69

For translation prediction errors under 5cm, which is a common measure of interest for pose estimation methods, the perspective variant shows significant performance improvement over the baseline method, while the projective method shows some improvement. All of the direct variants that remove the intermediate silhouette prediction branch show a drastic drop in translation prediction accuracy, indicating that even though the silhouettes are not directly used in the translation prediction, they enhance the networks ability to learn accurate feature scaling. The perspective-direct variant still shows significant improvement over the baseline-direct method, indicating that compensating for distortions in the fisheye image rather than directly predicting from the raw image is important for accurate pose predictions.

For orientation prediction errors under  $5^\circ$ , the perspective variant also shows significant performance improvement over the baseline method, while the projective variant does not perform as well as the baseline. In contrast to the translation predictions, all of the direct methods improve on the orientation prediction accuracy by approximately a factor of two across all variants, while the perspective-direct method still outperforms the baseline-direct method by a large margin. We observe that these initial results for orientation prediction fall short of the general target accuracy of less than 5deg error for manipulation applications. The UWHandles dataset is especially challenging for several reasons: the amount of training data is relatively small compared to terrestrial datasets, due to the expense of gathering underwater imagery; images are degraded by underwater back-scatter and lighting effects; the variance in camera viewpoints across an image sequence is high, due to the relatively low image collection frame-rate and large manipulator motions. Though these attributes make the dataset very challenging, they also motivate the development of methods that can work in real-world underwater environments with sparse training data. Future work could explore incorporating explicit methods of dealing with underwater effects, such as color correction and haze removal. We also note that the performance of the original SilhoNet [14] method was greatly improved through additional training on rendered data. Synthetically generated data can fill gaps in camera viewpoint representation missing in the real training data, allowing the network to better learn the full manifold of viewpoint representation.

The ADD-S results also show a strong improvement in performance for the perspective variant against the baseline, both with and without the silhouette predictions, while the projective and baseline methods perform similarly. Because the ADD-S metric is generally most sensitive to translation errors, the results show stronger performance for the methods that retain the intermediate silhouette prediction over the direct methods. However, taking into account the separate orientation and translation results, better overall performance on this dataset might be achieved by a method which directly predicts the orientation but retains a silhouette prediction branch during training to boost the translation accuracy. Overall, the results indicate that accounting for fisheye distortions before feature extraction, as the perspective method does, gives the best performance.

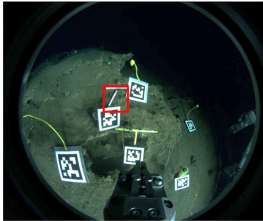
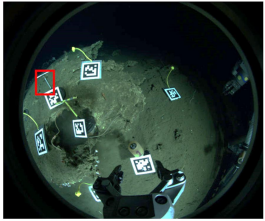
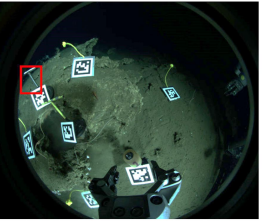
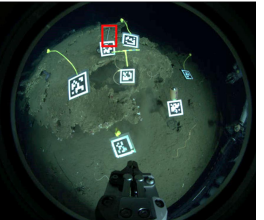








Input Image				
Pred Silhouette				
GT Silhouette				
Angle Error	4.55°	7.14°	18.11°	25.33°
Trans Error	0.11cm	1.11cm	4.63cm	2.55cm

Figure 3.3: Qualitative results with the perspective method on some sample test images for the whoihandle object. Predicted silhouettes and pose errors are shown for a range of errors from low to high.

Figure 3.3 shows some qualitative results with the perspective method for the whoihandle object on some test samples, exhibiting a range of prediction errors. It is evident that the network successfully learns the silhouette representation of the handle object. However, some silhouette predictions are distorted or regress to offset viewpoints. We conjecture that these issues reflect the sparse coverage of the training data over the full viewpoint manifold of the objects and could be addressed through additional training on synthetic data.

### 3.5 Conclusion

In this chapter, we presented a framework for adapting a ROI-based 6D object pose estimation method to work on full fisheye images. We demonstrated the adaptation of the SilhoNet [14] method on a new dataset of annotated fisheye images, called UWHandles, collected in natural underwater seafloor environments. The objects in the dataset are visually challenging handles, used in ROV operations to manipulate tools. The testing results on this dataset show that directly accounting for the fisheye distortions in the network before feature extraction is important for improving pose prediction accuracy, where the best performance was obtained with a method that generates a virtual perspective image centered on each ROI detection and processes these virtual undistorted images separately through the network. The results also show that the intermediate silhouette pre-

dictions of the SilhoNet method are important for the network to learn feature scaling to accurately predict translation. However, for this dataset, directly regressing the orientation rather than predicting from an intermediate silhouette achieves the greatest orientation accuracy. Supplementing the training with synthetic data could be an effective method for boosting the network performance.

## CHAPTER 4

# Hybrid Visual SLAM for Underwater Vehicle Manipulator Systems

### 4.1 Motivation

Exploration vehicles for remote environments, such as rovers, planetary landers, or underwater Remotely Operated Vehicles (ROVs) are often equipped with manipulator systems for collecting samples, placing sensors, or otherwise interacting with the environment. These systems largely rely on direct tele-operation or manually scripted commands to execute manipulation tasks, due to the risks associated with acting in unstructured and often complex remote environments. Despite these risks, there are some remote environments, such as Europa, the ice moon of Jupiter, so extreme that any kind of tele-operation or pre-scripted manipulator control is highly impractical. Considering environments closer to home, the deep ocean is a hot-bed of scientific research and exploration, but the expense of operating existing depth rated ROVs with their supporting ships and pilot teams is extravagant, while gaining operational time with one of these vehicles is also highly competitive. These considerations motivate the automation of manipulator systems for exploration vehicles, to enable complex scene interactions in communication denied environments, reduce the expenses associated with human operational teams or supporting tele-operation infrastructure, and increase the availability of these systems for scientific research. Critical to achieving safe and robust autonomy of such vehicle-manipulator systems is scene perception and reconstruction. In this chapter, we address the problem of feature based 3D scene mapping for underwater vehicle-manipulator systems (UVMSs). A key novelty of the mapping system is the fusion of feature points from both a vehicle mounted stereo camera and a dynamically positioned manipulator mounted fisheye camera into the same mapping framework. In situations where a UVMSs movement is limited or risky, this method addresses the problems of having limited viewpoints from the vehicle mounted cameras and incomplete scene reconstruction due to shadowing from scene structure by enabling the wrist mounted camera to dynamically extend the map beyond the vehicle fixed camera views and fill in shadowed areas of the scene.



This chapter makes the following contributions: 1. To our knowledge, the first SLAM system, designed for manipulator systems, that fuses a manipulator mounted fisheye camera into the same map with a vehicle mounted stereo camera. 2. An adaptation of the ORB-SLAM2 framework to GPU accelerated SIFT features, with improved odometer based tracking and real-time performance. 3. An evaluation of the SLAM method on both shallow reef and natural deep seafloor environments, where the method achieves good performance and standard ORB-SLAM2 fails. The evaluation datasets are also contributed with this work.

The rest of this chapter is organized as follows. Section 4.2 provides the background of related works. Section 4.3 describes our method. Section 4.4 presents an analysis of our method performance on underwater datasets. Section 4.5 concludes the chapter.

## 4.2 Related Work

3D scene mapping is a very mature problem in computer vision and robotics, and a rich body of literature has been generated from decades of study on the topic. Here we present a review of the works which we consider most relevant to the developed method and from which we took inspiration in the approach.

### 4.2.1 Feature Based Visual SLAM

Since its inception, ORB-SLAM [131] and its later adaptation to stereo, ORB-SLAM2 [132], remains one of the most widely adopted and complete feature based SLAM systems. ORB-SLAM demonstrated that a bundle adjustment approach can attain more accurate camera localization than direct methods or ICP, with the advantage of being less computationally expensive. Given the proven robustness of ORB-SLAM across a variety of applications and camera systems, the efficient computational performance based on a parallel thread architecture, and the demonstrated accuracy of keyframe based bundle adjustment for pose estimation, we chose to develop the method based on the ORB-SLAM2 framework.

CoSLAM [196] proposed an innovative solution for fusing multiple synchronized but independently moving monocular cameras into a single framework that can also differentiate between dynamic and static feature points. We took inspiration from this approach in the method design, with the key differences being the use of stereo features to constrain the map scale, the fusion of independent hybrid camera frames into the same map (i.e. the manipulator mounted fisheye camera and a vehicle mounted perspective stereo camera), and the specific adaptations of the method to underwater environments.

## 4.2.2 Underwater SLAM

Significant progress has been made in underwater vision applied to large scale survey reconstructions [91, 92], terrain aided navigation [134, 56, 72], and ship hull inspection [76, 144]. However, dense scene reconstruction methods generally process the image data offline, and methods designed for navigation generally provide very sparse feature maps if any. In contrast, our method emphasises real-time scene mapping, suitable for natural seafloor environments, that is robust to underwater visual effects and provides an optimized feature map and camera pose graph that can underlie dense reconstruction methods.

[134] proposed a stereo based SLAM method specifically designed for operating in underwater feature-poor environments. The map is constructed as a pose graph connecting to feature clusters. For inter-frame pose estimation of non keyframes, they used the VISO2 stereo odometer [63], which they found to perform better than the tracking stage in ORB-SLAM. For detecting loop closures, they generated a HALOC [34] signature for each feature cluster, which can be efficiently matched across very large image sets and does not require a prior training step like a bag of words representation. This work informed our choice of using a modified version of VISO2 for the initial inter-frame pose estimations. While their method was tailored specifically to the problem of localization through the optimization of keypoint cluster locations, our method, based on ORB-SLAM2, optimizes the location of the individual map points, which is desirable for scene reconstruction. [72] studied off the shelf monocular ORB-SLAM applied in different shallow oceanic underwater environments. Their results showed that ORB-SLAM performed very well in structured or feature rich environments, with adequate lighting and low flickering. ORB-SLAM performed poorly in areas with highly dynamic lighting, large numbers of moving objects, or low textured regions such as sand beds.

## 4.2.3 Kinematics in SLAM

Some prior work has been done on eye-in-hand based SLAM, where a camera is mounted near the endeffector of a manipulator. ARM-SLAM [100] used a Kinect depth sensor mounted on a manipulator with a fixed base to capture point clouds of the scene and fused them into a reconstruction using a method based on Kinect Fusion. SKCLAM [116] used feature based pose tracking with an RGB-D camera on the endeffector to calibrate the full kinematic parameters of an industrial manipulator with a fixed base. Point clouds from the RGB-D camera were integrated to construct a 3D map. [35] used ORB-SLAM3 and a stereo camera on a mobile manipulator to map an orchard. Novel to these prior works, our method fuses features from both an independent manipulator mounted fisheye camera and a vehicle mounted stereo in a common feature graph. We use a monocular camera on the wrist rather than relying on a depth sensor, which would be very bulky to

fit in a pressure rated housing for mounting on the manipulator. [42] proposed a method for calibrating a dynamic camera cluster, where one camera is articulated with respect to the other cameras in the rig. They demonstrated multi-camera SLAM with one camera mounted on a pan-tilt unit. However, their method assumed accurate calibration of the pan-tilt unit’s extrinsics. In contrast, our method is demonstrated with the manipulator mounted camera having 5-DoF actuation and a very large baseline to the vehicle mounted cameras relative to the stereo baseline, and our method does not assume accurate extrinsic measurements of the articulated camera. This method is also, to our knowledge, the first to demonstrate eye-in-hand SLAM on mobile underwater manipulator platforms in natural deep ocean environments.

### 4.3 Method

The hybrid SLAM system builds on top of the ORB-SLAM2 framework [132]. In this section, we highlight the changes made to adapt the ORB-SLAM2 system to SIFT features, the underwater environment, and the hybrid camera system. For details on the system architecture that remain unchanged from ORB-SLAM2, we defer the reader to [132].

Figure 4.1 shows a high level block diagram of the hybrid SLAM system, where our method retains the same four threaded architecture as the original ORB-SLAM2 system. The most significant modifications were made in the tracking thread, which follows the top horizontal flow of the diagram, with separate functional flow branches for stereo and monocular fisheye frames. Both stereo and fisheye frames share a common keyframe representation which is processed through the local mapping, loop closing, and full bundle adjustment threads. The core of the system is the feature based stereo mapping framework, which can be operated stand-alone or in a hybrid mode, where frames from an independently moving fisheye camera are fused into the same map. In our collected datasets, the fisheye camera is synchronized with the stereo camera. However, this synchronization is not a requirement of our current method. However, future work may extend the method with a kinematic factor between the stereo and fisheye camera, in which case synchronizing the cameras with the joint state feedback would be important.

The constructed map is represented as a covisibility graph of optimized keyframe and keypoint poses, with factors between keyframes formed through common keypoint observations. Like ORB-SLAM2, the covisibility graph is used to retrieve a local neighborhood of keypoints for the tracking and local mapping stages and forms the graph structure for the bundle adjustment optimizations. A minimum spanning tree is also maintained, which connects every keyframe to the neighbor with the maximum number of shared keypoint observations. The spanning tree is used to propagate keyframe pose optimizations from full bundle adjustment to new keyframes that were not included during the optimization. A DBoW2 module [62], adapted to SIFT features, is used for place

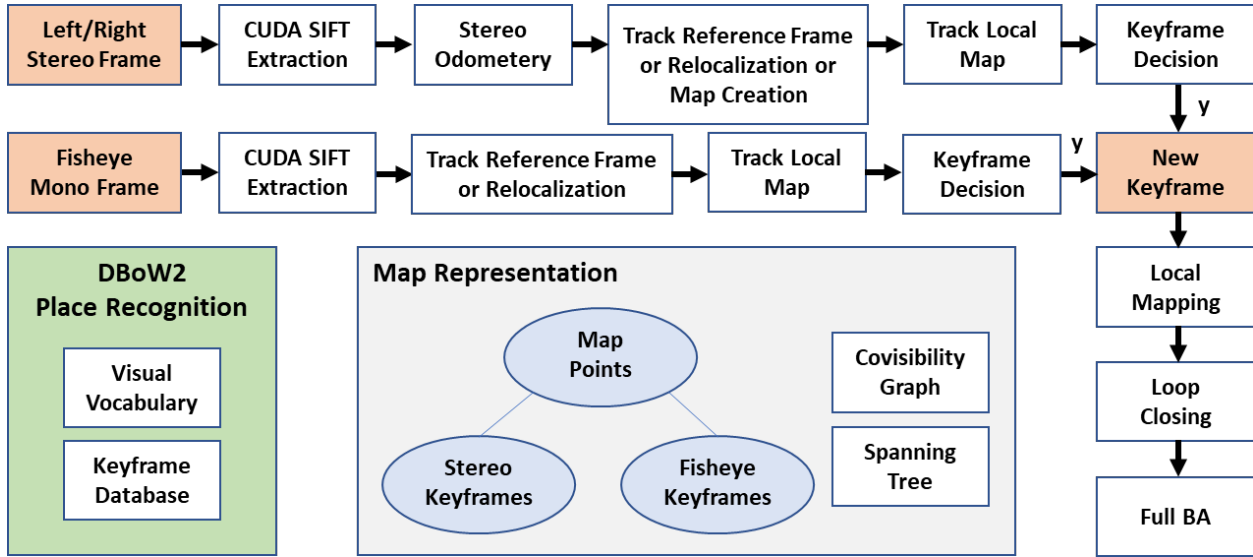


Figure 4.1: System block diagram

recognition during relocalization and loop closing.

### 4.3.1 Hybrid Camera System

The hybrid camera system is specifically tailored to mobile manipulator systems, with a stereo camera mounted on the vehicle and an independent fisheye camera mounted on the manipulator wrist. In our evaluations, the stereo pair uses a pinhole camera model and the fisheye camera uses the Kannala-Brandt [95] model. We adapted the camera model code from ORB-SLAM3 [29] to support the hybrid camera system.

### 4.3.2 Feature Representation

While ORB-SLAM2 uses ORB [154] features, ORB performs poorly in many underwater environments compared to other feature types. We conducted an analysis of the matching performance of different feature types in the underwater domain, presented in the results section, which motivated our choice of the SIFT [119] feature for the system. We adopted CudaSIFT [18], which is one of the fastest GPU accelerated SIFT implementations, for real-time feature extraction.

### 4.3.3 System Initialization

On system startup, the first keyframe is created from the first stereo frame that retains at least 8% of the maximum number of features that can be extracted. This keyframe is set as the origin of the map and the initial map is constructed from all of the stereo keypoints of the frame. After the map

is initialized, new keyframes can be added from both the stereo and monocular fisheye frames, with the map scale constrained by the initial and new stereo map points.

#### 4.3.4 Stereo Odometry

Similar to [134], we found that the tracking stage of ORB-SLAM2 failed on the underwater datasets, even when adapted to SIFT features. A considerable limitation of the ORB-SLAM2 tracking stage is a constant velocity model, which has poor accuracy at the low frame rates typical for underwater imaging systems. [134] used the VISO2 stereo odometer for initial frame pose estimation. We took inspiration from this and also adopted VISO2 for our system. However, we found that off-the-shelf VISO2 failed to track the underwater stereo dataset, due to poor performance of the simple blob and corner response features, described in the Sobel operator space. We modified VISO2 to use CudaSIFT features, which are extracted once for each image and then propagated through the rest of the SLAM pipeline for efficient computation. While the original VISO2 implementation used a search window to circularly match features across the current and previous stereo pair, we use GPU accelerated brute force matching, followed by circular filtering for improved computational performance. In this scheme, brute force matching is applied between the *previous left* and *previous right* frames, *previous right* and *current right* frames, *current right* and *current left* frames, and *current left* and *previous left* frames. A feature is accepted only if the same feature is matched across all image pairs in a circular fashion. Like in the original VISO2 implementation, feature matches between a *left* and *right* stereo image pair are further filtered by an epipolar constraint of 1 pixel error tolerance. However, we found the outlier removal step of the original VISO2 by 2D Delaunay triangulation to be too restrictive in high rugosity coral reef imagery, resulting in the filtering of many correct feature correspondences.. Through extensive experimentation, we found the circular matching and epipolar constrained filtering steps were sufficient for removing the majority of outliers before processing the matches through the ego-motion estimation stage.

#### 4.3.5 Tracking

Given the current **stereo frame** with an initial pose estimate from odometry relative to the previous frame, the map points observed in the previous frame are tracked in the current frame by projecting them into the current left stereo image and searching for feature correspondences within a small window. If enough map point correspondences are found, the keyframe pose is optimized based on the reprojection error of the map points. If not enough correspondences are found, the current stereo frame is tracked relative to the map points observed in its reference keyframe, using the BoW vocabulary levels to guide the matching, and the pose is optimized if enough correspon-

dences are found. If not enough inlier matches with the reference keyframe map points are found but the odometry estimate has enough feature match inliers with the previous frame, then the pose of the current frame is set to the odometry estimate with no initial map matches. Given the initial pose estimate from this tracking step, the system proceeds to track the current stereo frame to a local map of keypoints observed by a neighborhood of keyframes, as in ORB-SLAM2, with the difference that neighborhood keyframes can be both monocular fisheye or stereo keyframes. If tracking fails, the system enters relocalization mode until tracking is recovered for a stereo frame.

In hybrid mode, the current monocular **fisheye frame** is only tracked if the current stereo frame was successfully tracked. If a fisheye keyframe has already been added to the map, the current fisheye frame is first tracked relative to the map points of its reference keyframe, using BoW guided matching, and the pose is optimized if enough correspondences are found. For the iterative optimization procedure, the pose is initialized to the previous fisheye frame pose. If tracking the reference keyframe fails or no fisheye keyframe has yet been added to the map, the current fisheye frame is tracked relative to the map points of the current reference stereo keyframe. If tracking succeeds, the system proceeds to track the current fisheye frame to the local neighborhood of keyframes in the same way as the stereo frame, and, if no fisheye keyframe has yet been added to the map, a new fisheye keyframe is created and added to the map. If tracking fails for the fisheye frame but not the current stereo frame, the system enters relocalization mode for only the fisheye camera, while continuing mapping of the stereo frames. In this relocalization mode, the current fisheye frame is first attempted to be matched against all keyframes in the map using the BoW place recognition to identify match candidates. If place recognition fails, tracking of the current fisheye frame is then attempted against the current stereo reference keyframe. If this tracking succeeds, the fisheye frame is processed through the local mapping step.

During the local mapping stage for both stereo and fisheye frames, the reference keyframe for each is updated to the keyframe that shares the most feature matches, agnostic to the type of keyframe (i.e. stereo or fisheye). When a new keyframe is inserted, it is made the reference keyframe for the next frame of the same type.

During relocalization or when the fisheye frame is tracked against the reference stereo frame, a perspective-n-point (PnP) solver is constructed to estimate an initial pose. Like ORB-SLAM3, we adopt the Maximum Likelihood Perspective-n-Point algorithm (MLPnP) [181], which uses projective rays in the optimization that are agnostic to the camera model, in order to accurately optimize the feature correspondences between the hybrid fisheye and perspective stereo frames.

### 4.3.6 Inserting New Keyframes

New stereo and fisheye keyframes are decided following the same scheme as ORB-SLAM2 for stereo and monocular keyframes respectively, with some thresholds tuned for lower framerates and higher keypoint counts.

When a new keyframe is inserted, new map points are triangulated and added into the map. For each of these keypoints the maximum and minimum distances that the point can be detected in a frame are calculated based on the scale of the keypoint in the reference keyframe. With the hybrid camera system, the scale of the keypoint can be different at the same distance, depending on which type of frame observes the keypoint. We resolve this ambiguity by normalizing the keypoint scale factor by the focal length of the observing frame. This normalization enables consistent keypoint scale prediction and comparison between hybrid frames.

### 4.3.7 Loop Closing

For place recognition, we adapted DBoW2 to SIFT features and trained a million word vocabulary with ten branching factors and six levels, like the ORB vocabulary used in ORB-SLAM2. The vocabulary was trained on an extensive set of underwater imagery data, including the UWHandles and LizardIsland datasets presented in this paper, plus three large imagery datasets from the Australian Center for Field Robotics: Tasmania CSP [12], Scott Reef 25 [173], and Tasmania O’Hara 7 [172]. 2000 CudaSIFT features were extracted per image, with the image upsampled by a factor of 2 for the first scale pyramid level, the initial blur set to 1.6, and the difference of Gaussian threshold set to 1.0.

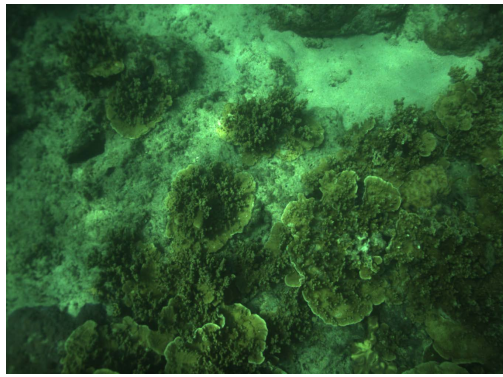
### 4.3.8 Datasets

#### 4.3.8.1 Stereo Survey Dataset

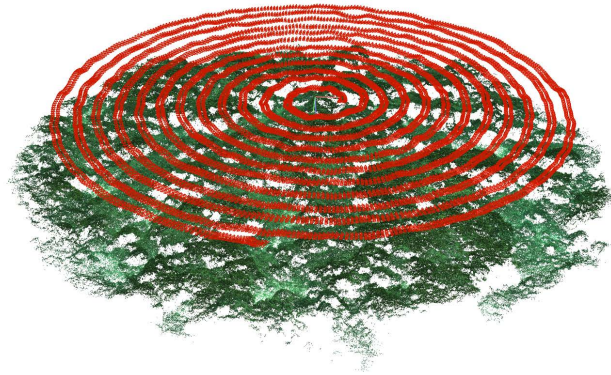
A stereo SLAM evaluation dataset was collected with a diver operated camera rig on a shallow coral reef of Lizard Island in Australia (fig. 4.2). The dataset was collected using a spiral survey technique [150] that fully covered a circular area of approximately 14m in diameter, with natural sunlight providing the only illumination. The rectified stereo image size is 1355x1002 pixels and the images were collected at 5Hz. We refer to this dataset as **LizardIsland**.

To obtain a ground truth comparison for evaluating the stereo SLAM method, we processed the dataset through COLMAP [161] to generate a sparse 3D reconstruction with optimized camera poses. COLMAP does not fix the scale during optimization, so the reconstruction was scaled in post-process to match the mean left and right stereo pair baseline to the calibrated value.





(a) Left stereo image



(b) COLMAP reconstruction

Figure 4.2: The LizardIsland spiral survey dataset was collected with a diver operated stereo rig. The ground truth reconstruction was generated with COLMAP.

#### 4.3.8.2 Hybrid Vehicle-Manipulator Dataset

During a cruise in 2019, a hybrid dataset of synchronized vehicle mounted stereo and wrist mounted fisheye imagery was collected in natural deep ocean environments of the Costa Rican continental shelf with the SuBastian ROV, operated by Schmidt Ocean Institute. The fisheye imagery portion of this data was published as the **UWHandles** dataset [15]. For this work, we have extended this dataset by further processing four environmentally unique stereo and fisheye image sequences for evaluation of the hybrid SLAM method. We refer to these sequences as Mounds1, Mounds2, Seeps1, and Seeps2 (fig. 4.3). For these sequences, TagSLAM [149] was used to obtain ground truth pose estimates for the stereo and fisheye cameras, based on the detection of AprilTags [184] distributed in the scenes.

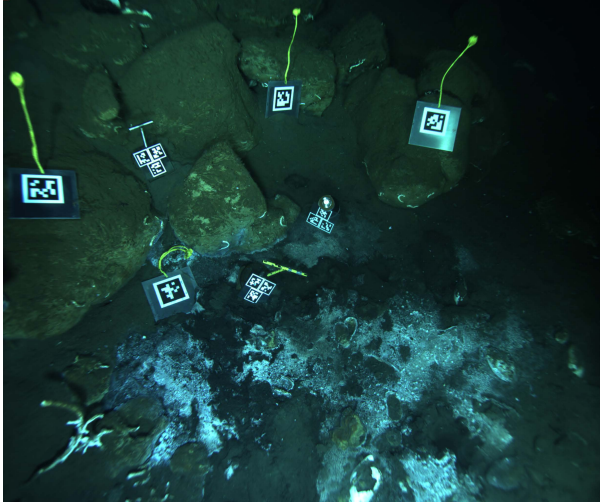
## 4.4 Results

All evaluations were run on a desktop computer with an AMD Ryzen Threadripper 2990WX CPU and an NVIDIA Titan V GPU.

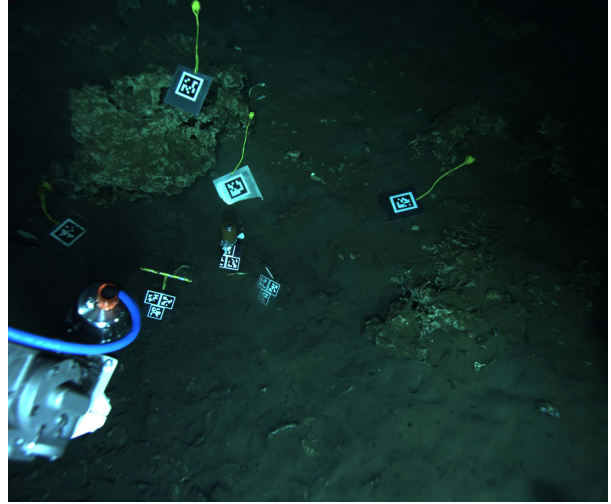
### 4.4.1 Comparative Feature Analysis

We conducted an evaluation to determine which feature representation is best adapted to the visual degradation of underwater environments and can be robustly matched between hybrid perspective and fisheye frames with variable relative poses. We sampled every fifth hybrid frame from each of the UWHandles dataset sequences and, to reduce any bias from artificial features, we used the

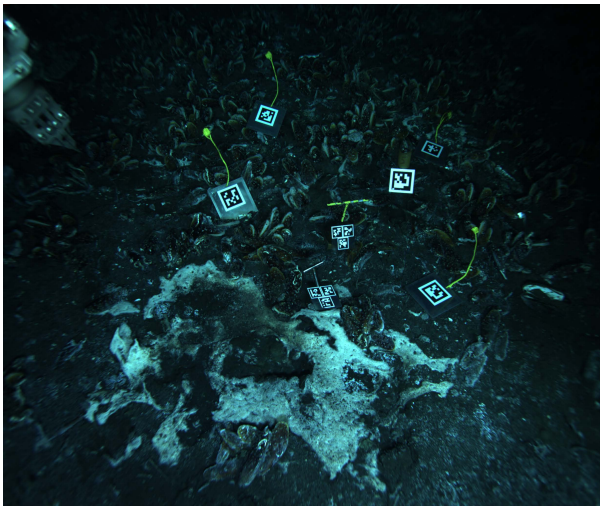




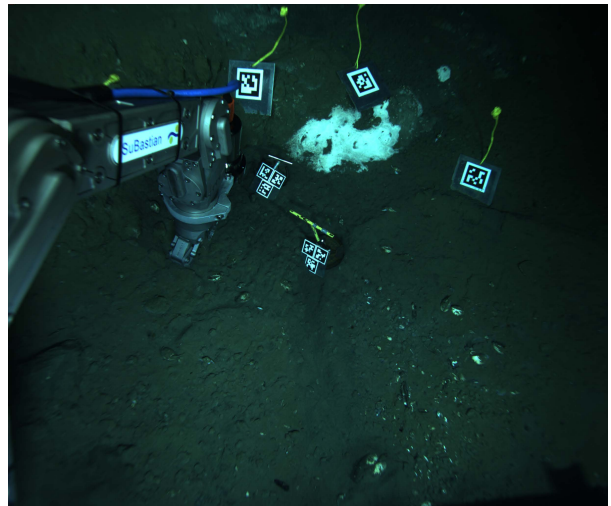
(a) Mounds1



(b) Mounds2



(c) Seeps1



(d) Seeps2

Figure 4.3: Four hybrid image sequences were collected in deep seafloor environments of the Costa Rican shelf margin. Shown here is a sample left stereo image from each sequence. Mounds1 ((a)) is an area of rocks and bacterial matting. Mounds2 ((b)) is a mud flat with rubble. Seeps1 ((c)) is a dense bed of clams with bacterial matting. Seeps2 ((d)) is a mud flat with a small patch of bacterial matting.

tracked AprilTag poses from TagSLAM to project circular masks over the tags in the image frames. For each feature type, 2000 features were extracted from each image, and the features were brute force matched across each hybrid fisheye and left stereo image pair. Lowe’s ratio test was applied to remove ambiguous matches, with a ratio threshold of 0.8 for all feature types except ContextDesc, which achieved significantly improved performance with a ratio of 0.9. OpenCV’s RANSAC based essential matrix fitting was used to filter the matches and recover a relative pose estimate between each fisheye and stereo frame. Table 4.1 shows the results of this evaluation. Given that an essential matrix based pose estimate does not provide scale, both the orientation and translation errors of the pose estimates were evaluated as angular errors. For translation, this error is the angular difference between the translation direction vector from the left stereo frame to the fisheye frame. The performance was evaluated using the area under the accuracy-threshold curve (AUC) with a max angular error of  $180^\circ$ . While most of the tested feature types were popular conventional features, we also tested two deep learned feature variants: ContextDesc and SuperPoint. We note that the learned features were used with their provided model weights and were not fine-tuned on underwater data. Of the conventional feature types, ROOT\_SIFT and SIFT perform the best, achieving significantly better performance than ORB. Of the deep learned features, SuperPoint had highly variable performance across the different sequences, and the mean number of inlier matches was lower than other conventional features. Interestingly, ContextDesc performed the best overall out of all the feature types, consistently matching more than double the features of ROOT\_SIFT and achieving very high AUC scores. It is noteworthy that ContextDesc uses SIFT interest points but learns the descriptor, so all of the best performing features are based on the SIFT detector. These results merit further investigation into the application of learned features for underwater vision. For our initial implementation in this work, we chose to use a highly optimized GPU accelerated implementation of SIFT, but we note that the learned descriptors of ContextDesc are 128-d, like SIFT, and are directly compatible with the entire method pipeline.

#### 4.4.2 Stereo SLAM

The core of our system is a stereo SLAM pipeline, which must be robust to underwater environments. We used the LizardIsland survey dataset to evaluate the stereo SLAM performance. We tested ORB-SLAM2 on this dataset, both with and without loop closing enabled, but it lost track after only a few frames and was unable to relocalize. We also tested the vanilla VISO2 stereo odometer, but, even with extensive tuning, VISO2 failed to track the dataset with sensible accuracy. We evaluated our stereo SLAM method with both 2000 and 4000 CudaSIFT features extracted each frame, with an interest point Difference of Gaussian threshold of 1.2. Figure 4.5 shows the results for 4000 features, both with and without loop closing enabled, and table 4.2 gives

Table 4.1: Area under accuracy-threshold curve evaluation of feature matching performance on the UWHandles underwater hybrid image sequences. Accuracy is evaluated as angular error in the predicted rotation (AUC Rot) and translation direction vector (AUC Trans) between each hybrid left stereo and fisheye image pair. Also reported is the mean number of inlier feature matches across each sequence.

Sequence		SIFT [119]	ROOT SIFT[8]	ORB [155]	SURF [13]	AKAZE [6]	CONTEXTDESC [121]	SUPERPOINT [47]
Mounds1	AUC Trans	0.949	0.936	0.856	0.877	0.922	0.970	<b>0.98</b>
	AUC Rot	0.937	0.948	0.750	0.812	0.858	0.951	<b>0.964</b>
	Mean Matches	91	101	25	34	46	<b>210</b>	74
Mounds2	AUC Trans	0.946	0.940	0.864	0.886	0.906	<b>0.976</b>	0.947
	AUC Rot	0.810	0.853	0.488	0.676	0.629	<b>0.959</b>	0.873
	Mean Matches	31	33	14	21	21	<b>80</b>	41
Seeps1	AUC Trans	0.964	0.980	0.885	0.869	0.904	<b>0.986</b>	0.938
	AUC Rot	0.944	0.965	0.745	0.770	0.811	<b>0.980</b>	0.885
	Mean Matches	64	73	23	28	43	<b>150</b>	53
Seeps2	AUC Trans	0.960	0.964	0.935	0.930	0.954	<b>0.974</b>	0.917
	AUC Rot	0.942	0.953	0.894	0.891	0.926	<b>0.965</b>	0.763
	Mean Matches	89	100	60	50	92	<b>146</b>	39

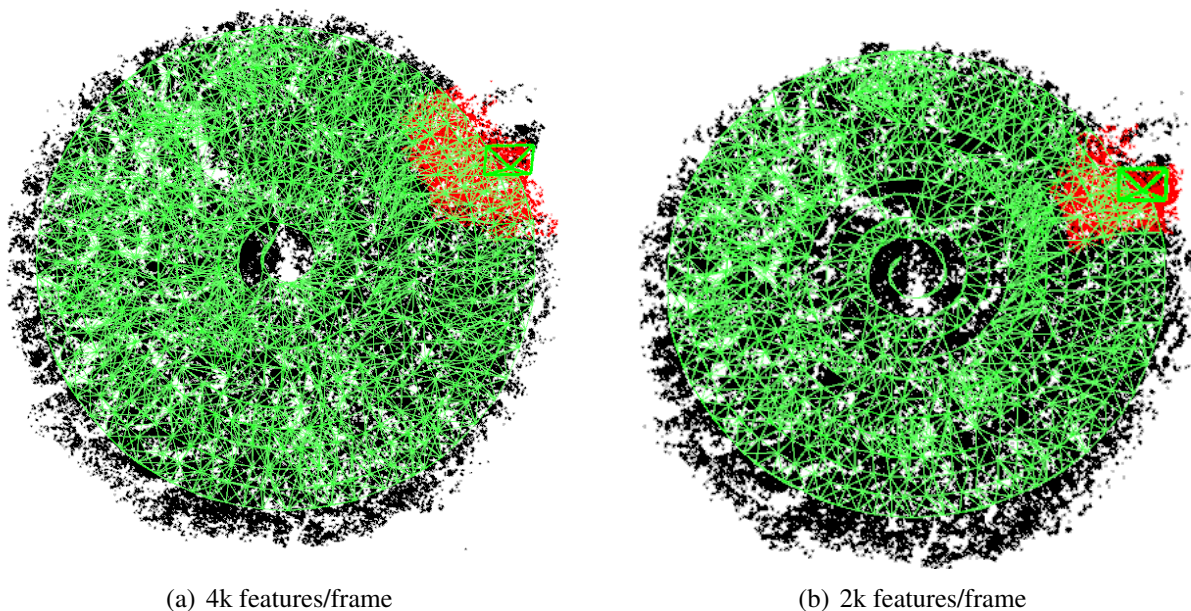


Figure 4.4: Final stereo SLAM maps on the LizardIsland dataset, showing the densely connected keyframe graphs.



Table 4.2: Stereo SLAM performance on the LizardIsland dataset, with the number of extracted features is set to 4000 and 2000. Performance is evaluated as RMSE of the absolute trajectory error. Results are reported with and without loop closing enabled. Also reported is the number of keyframes (KFs) and map points (MPs) in the final map and the average frame processing time in the tracking thread.

System Mode	RMSE (cm)	KFs	MPs	Avg Time (ms)
Tracking Only (4000)	49.1	-	-	94.2
Loop Closing (4000)	1.4	562	190,474	117.9
Tracking Only (2000)	58.2	-	-	55.7
Loop Closing (2000)	1.8	622	98,812	64.8

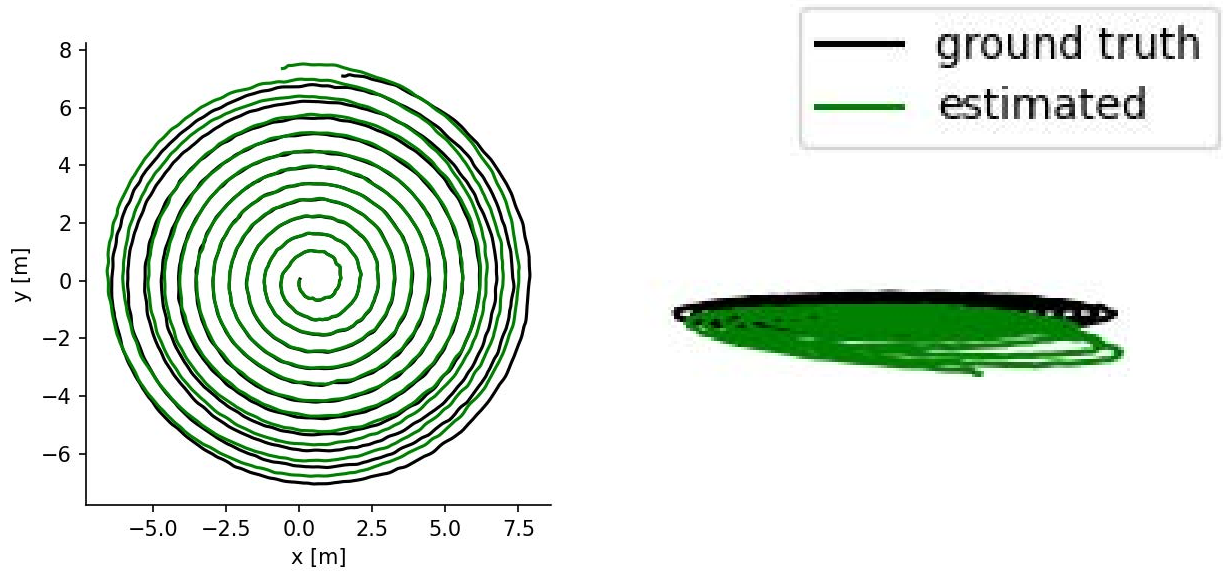
Table 4.3: Evaluation of hybrid SLAM on the UWHandles dataset. Error is evaluated on the estimated pose difference between the left stereo and fisheye cameras for each synchronized hybrid frame, where  $\Delta t$  is translation error and  $\Delta q$  is rotation error. The "hybrid matches" column gives the number of fisheye frames registered in the map over the total number of frames in the sequence. The error is only evaluated over the registered frames. The "KFs" column is the number of keyframes in the final map for the hybrid SLAM mode versus stereo only mode, and the "MPs" column is the same format for the number of final keypoints in the map.

Sequence	$\Delta t$ mean (cm)	$\Delta t$ median (cm)	$\Delta q$ mean (deg)	$\Delta q$ median (deg)	hybrid matches	KFs hybrid/stereo	MPs hybrid/stereo
Mounds1	2.04	2.06	0.58	0.50	652 / 783	21 / 11	4271 / 2671
Mounds2	1.38	1.22	0.98	0.82	713 / 756	24 / 11	4086 / 1773
Seeps1	2.82	2.02	1.17	0.46	1059 / 1089	24 / 11	4636 / 2847
Seeps2	2.12	1.84	1.38	0.97	778 / 802	23 / 16	4320 / 2365

the performance of the system in all tests. The test trajectories were aligned with the COLMAP ground truth using the Horn method [77] without scaling. The figure shows that the visual odometer based tracking method without loop closing tracks very well in the horizontal plane with most of the drift error being accumulated in the z-depth estimate. For both extracted feature counts, the table shows that a high accuracy, with less than 2cm root mean squared absolute trajectory error (RMSE), is attained by the full SLAM system with loop closing. For 4000 features per frame, the number of map points in the final map is approximately double the map point count for 2000 features per frame, showing that the system scales well with the number of extracted features. The system can achieve  $>10\text{Hz}$  for 2000 features per frame, which is a high framerate for underwater systems. Figure 4.4 shows the densely connected keyframe graphs for the final SLAM maps, demonstrating consistent loop closing between neighboring spiral trajectories.

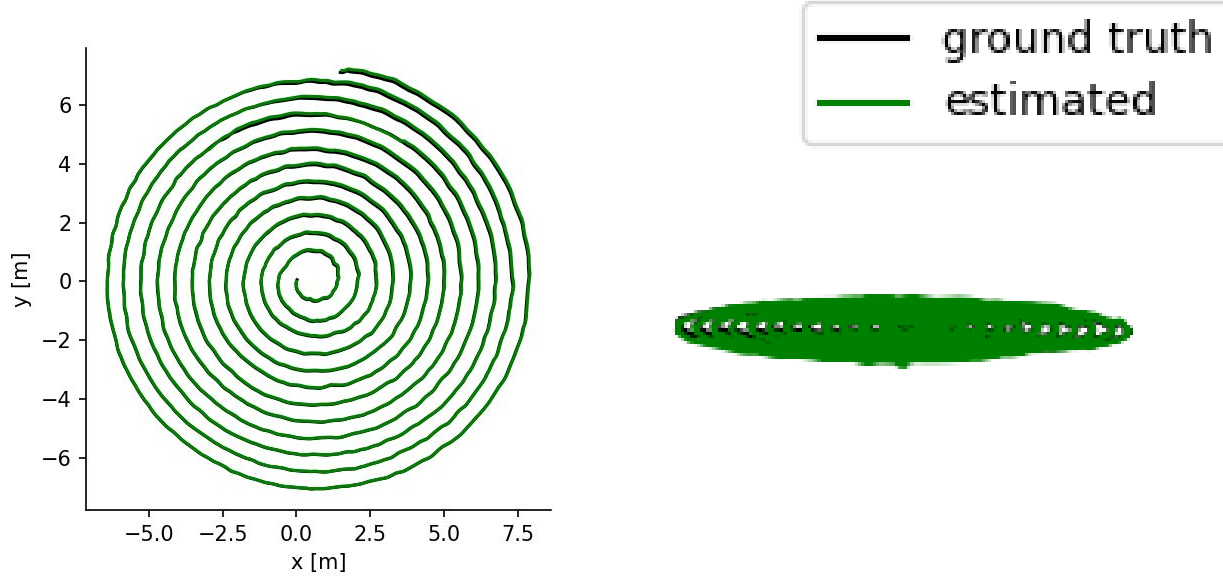
### 4.4.3 Hybrid SLAM

We evaluated the performance of the hybrid SLAM system on the four sequences of the UWHandles dataset. The results are reported in table 4.3. For all sequences, every stereo frame was successfully registered in the SLAM map. Given that the stereo camera is mostly stationary across these image sequences, and to reduce the effect of noise in the imperfect ground truth, we evalu-



(a) Tracking top view

(b) Tracking side view



(c) SLAM top view

(d) SLAM side view

Figure 4.5: Stereo SLAM results for the LizardIsland dataset when loop closing is disabled ((a),(b)) and enabled ((c),(d)).

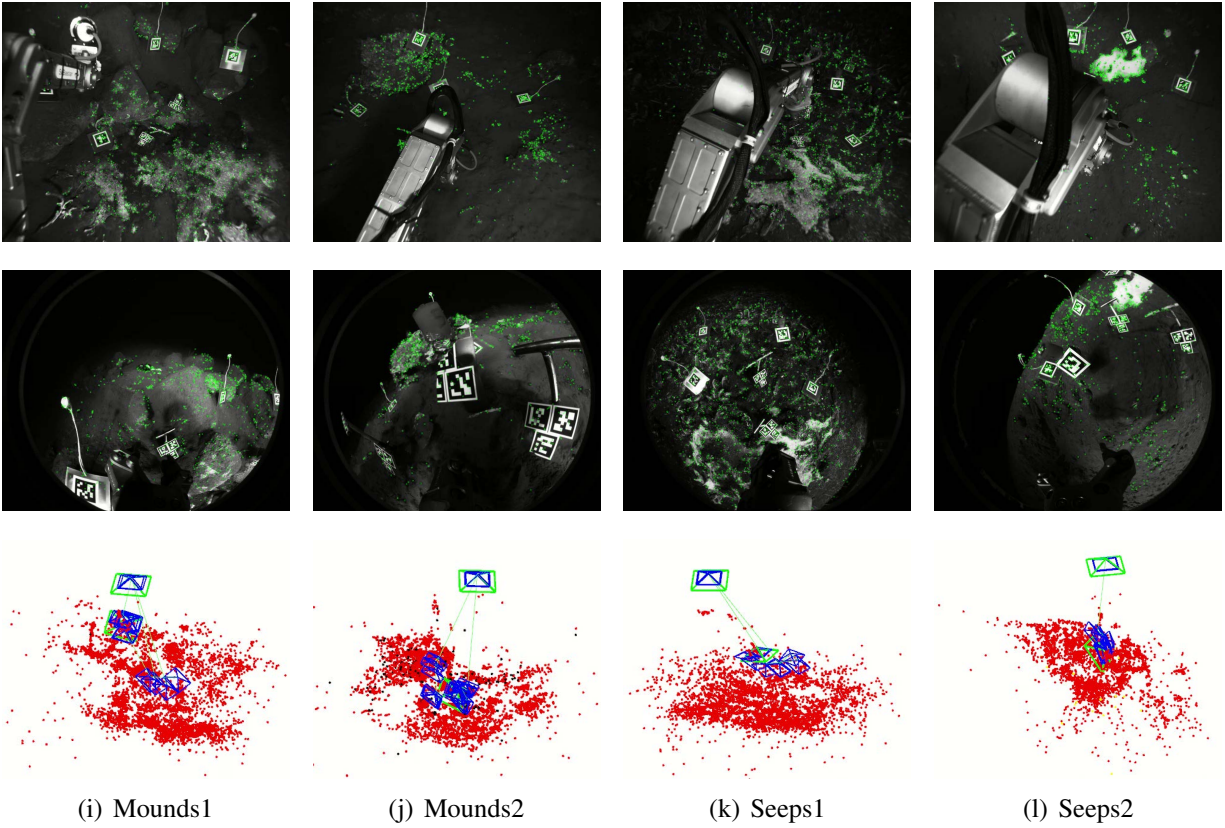


Figure 4.6: Snapshots of hybrid SLAM running on the UWHandles sequences. Top row is the left stereo camera frame, middle row is the manipulator mounted fisheye frame, and bottom row is the map with the keypoints and keyframes.

Table 4.4: Hybrid SLAM timing evaluation, measured as the mean frame processing time in the tracking thread.

# Features / Frame	2k	4k	6k
Mean Time	179ms	249ms	314ms

ated the hybrid SLAM error using the relative pose estimates between the left stereo and fisheye cameras for each synchronized hybrid frame. Only hybrid frames where the fisheye frame was successfully registered into the map were included in the error evaluation. According to the table, the system generated approximately twice as many keyframes and map points when running in hybrid mode versus stereo only mode, demonstrating the ability to extend the map beyond the limited stereo camera viewpoint. Also, the majority of fisheye frames were successfully registered into the map for all sequences. Despite the sequences varying significantly in environment type, the hybrid SLAM mode is able to generate a similar amount of keyframes and map points for each sequence, and the estimated pose errors are very similar across each sequence, demonstrating the system can operate in challenging and diverse, natural seafloor environments. Figure 4.3 shows a frame capture from running hybrid SLAM on each of the sequences. Table 4.4 gives the timing evaluation for processing a hybrid stereo and fisheye frame pair through the tracking thread for different feature count settings. For 4000 features extracted per image, the system can easily attain 3hz, which is the rate that the UWHandles data was collected.

## 4.5 Conclusion

In this chapter, we have presented a novel hybrid SLAM method, targeting deployment on underwater vehicle manipulator systems, that can operate in real-time. The method can fuse features from both a vehicle mounted stereo camera and a manipulator mounted fisheye camera into the same map, enabling dynamic viewpoint acquisition and map extension with the manipulator mounted camera. We have demonstrated the robustness of the method on both a shallow reef stereo image survey dataset and on four hybrid image sequences captured in natural, deep seafloor environments.

There are several promising directions for future development of this SLAM system. First, a kinematic factor could be formulated on the optimization graph from the manipulator joint states between the manipulator camera and the vehicle mounted stereo to improve registration of the manipulator camera into the map and the overall robustness of the mapping method. This factor would also enable real-time feedback for the kinematic calibration of the manipulator, which is a challenging problem for the imprecise hydraulic manipulators common for underwater systems. Second, the use of learned feature descriptors such as ContextDesc could be explored to improve

system performance. Finally, the system could be extended with a dense reconstruction stage that is optimized on the sparse feature maps and camera poses to build a complete real-time scene reconstruction method for UVMSs.



## CHAPTER 5

# Design of Underwater Optical Systems

### 5.1 Motivation

Optical cameras are increasingly being applied in the underwater domain for a range of applications including inspection tasks [26], ecosystem monitoring [186] and vehicle navigation [54]. Cameras represent low cost, low power sensors that provide rich information about the underwater scene and frequently complement other sensors deployed on autonomous underwater vehicles (AUVs) or ROVs. However, the design of an underwater camera system presents a very large space of possible design choices and system configurations, with many inter-dependencies. Additionally, field tuning of the camera settings is frequently cumbersome and time consuming due to reduced equipment accessibility when deploying underwater.

In this chapter we review a simplified underwater image formation model that allows the estimation of the average camera sensor response given different lens, light, water and seafloor characteristics. The sensor response is the average intensity of pixels in a camera image and is a metric that can be used to determine correct image exposure. A user-friendly interface for the model is developed that will allow researchers and scientists to narrow down the equipment requirements and operational settings for an underwater imaging system by parametrically exploring the design space.

In order to estimate the camera response, a model of underwater image formation is required. One of the main drivers for the study of the underwater image formation process and the development of models has been the need to correct underwater image degradation such as haze, low contrast and color cast due to water impurities and wavelength dependent attenuation. Early efforts by Duntley [52] laid the foundation for modelling underwater light propagation. Computer models developed by McGlamery [125, 126] were extended by Jaffe in 1990 [82], leveraging advances in computational processing capabilities to create the UNCLES computer simulation system, which is capable of analyzing the performance of underwater camera systems. The UNCLES simulator helped guide the design of the video equipment for the ARGO underwater imaging platform [83],

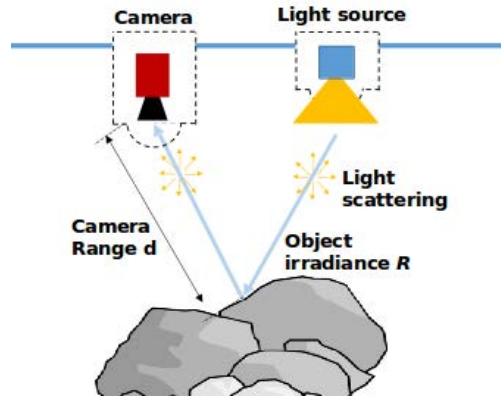


Figure 5.1: Schematic of underwater light propagation from light source to camera sensor, where the light signal is affected by scattering and absorption through the water column and the reflection characteristics of the seafloor.

but the tool was not released for public use. The theory for underwater light propagation has previously been developed, but there lacks a consolidation of this knowledge into a framework broadly usable by the science and engineering communities for the design of underwater camera systems. The tool introduced in this chapter incorporates the model developed through these prior works with an interface focused on user friendliness and minimal complexity. Some assumptions are made to simplify the model, based on common characteristics of underwater imaging systems, and the validity of this model is demonstrated through experimentation. The contributions presented in this work are 1. A review of the underwater image formation model with a procedure to characterize underwater camera systems. 2. An open source tool<sup>3</sup> to aid the design process for an underwater camera system through exploration of the parameter space. 3. Validation experiments supporting the presented model as a good characterization of an underwater camera system.

The rest of the chapter is structured as follows: Section 5.2 introduces the underwater image formation model used in the software toolbox to compute sensor responses underwater; Section 5.3 presents the developed software toolbox, with an overview of the intended design use and user interface; Section 5.4 presents the experiments validating the proposed image formation model; and Section 5.5 concludes the chapter.

## 5.2 Underwater Image Formation

In this section we introduce the underwater image formation model. As light travels from a source through the water column, it is attenuated through absorption and scattering. The light that reaches the seafloor or other obstacle is reflected by a fractional amount, dependant on the albedo of the

<sup>3</sup><https://github.com/gidobot/UWOpticalSystemsDesignTools>

surface. The reflected light is further attenuated in the water column as it travels back towards the camera. Light is refracted at the water interface of the camera housing viewport before reaching the camera lens. Photons passing through the lens generate electrical signals on the camera sensor that are amplified and digitized to form the final image. This process is illustrated in figure 5.1, and figure 5.2 provides an overview of how the model equations describe the image formation pipeline.

### **5.2.1 Artificial Light systems**

Natural light is attenuated exponentially in the oceans and frequently does not penetrate deeper than 100m. The model assumes all light in the scene is generated from artificial light sources mounted on the vehicle. This situation represents the worst case scenario, as constraints on camera systems are relaxed if natural light is present. The presented model describes a light source by three main parameters:

1. Luminous flux emitted by the light source, measured in lumens: This can be obtained for most underwater lights, strobes or LED modules in custom designs.
2. Normalized light spectrum: The spectrum of the light source describes how the luminous flux is spread over the different wavelengths. When the spectrum is not available, it can be approximated based on known spectra for common light sources. Figure 5.3 shows spectrum characteristics of common light types such as LED, fluorescent or natural sunlight.
3. Beam pattern: The beam pattern describes how the light spreads as it travels away from the source. We assume a simple conical beam pattern defined by its aperture half-angle  $\beta$ , which is typical for most underwater strobes.

### **5.2.2 Underwater Light Propagation**

Light traveling underwater from the strobe to the camera sensor is modified through absorption, scattering, reflection, and refraction at optical interfaces. We describe how each of these effects is modeled in the system.

#### **5.2.2.1 Attenuation**

The Jaffe-McGlamery model describes the propagation of light underwater as the sum of direct, backscattered and forward-scattered light. Attenuation of the light signal is modeled as an exponential decay, with function parameters depending on the water type and clarity. Coefficients describing the attenuation effects for different classes of water, known as Jerlov water bodies, have

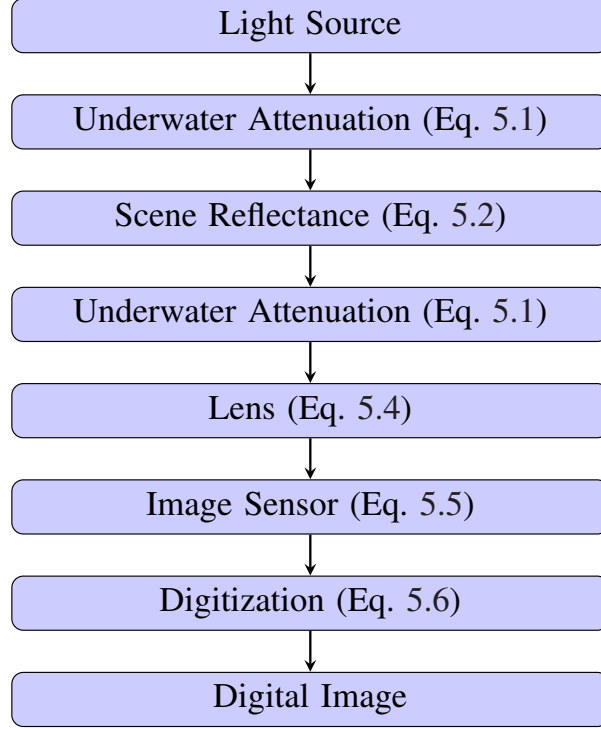


Figure 5.2: Image formation pipeline describing the different steps through which light is subjected to form the underwater digital image.

been cataloged [171]. The exponential decay modeling attenuation of the light signal in water is given as

$$L = R e^{-b(\lambda)d} \quad (5.1)$$

where  $R$  is the initial irradiance,  $b(\lambda)$  the wavelength dependent attenuation coefficient and  $d$  the distance of propagation. Absorption and scattering coefficients are mostly dependent on chlorophyll and dissolved organic matter in the water column [171]. Experiments performed by Jerlov [88] established a set of attenuation profiles for different types of water bodies, both coastal and oceanic, with varying clarity levels. These profiles are provided with the model as default selections. The user also has the option to load custom profiles.

### 5.2.2.2 Object reflectivity

The reflectance of light by a surface is modeled by the Bidirectional Reflectance Distribution Function (BRDF) [136] that relates the outgoing radiance  $L$  of the surface with the incoming irradiance  $E$ . Assuming diffuse reflection in the model, where  $\theta_i$  is the light incident angle and  $M(\lambda)$  is the material and wavelength dependent reflection coefficient, the BRDF is simplified to:

$$L = E \frac{M(\lambda)}{\pi} \cos(\theta_i) \quad (5.2)$$

### 5.2.2.3 Light refraction

Underwater cameras are housed inside enclosures that protect the electronic systems from water damage and pressure. In order for light to reach the sensor, these enclosures employ an optical port made of translucent material such as glass or acrylic, most frequently in either a spherical or flat geometry. As light travels through the port, it is refracted at each optical interface as a function of the change in index of refraction and the direction of the incident ray relative to the surface normal. In effect, the optical port of the housing must be considered as part of the camera lens system.

In the case of a domed viewport, the dome is treated as a thick lens formed by two concentric hemispherical surfaces. Analysis of the thick lens equations show that objects at infinity are mapped to a virtual image in the front of the dome that is curved concentrically with the dome [86, 5]. A camera housed with a dome viewport must be focused at the distance of the virtual image when immersed in water rather than the distance to the imaging target in air. The distance of the virtual image from the front of the dome is derived in [86, 5], and we incorporate these equations into the camera system design tool.

When the camera lens principal point is aligned with the dome center of curvature, the field of view of the camera remains unchanged [86, 5]. A common method to verify the camera is correctly aligned with the center of the dome is to look at an image of a checkerboard taken with the camera in the housing while only half immersed in water. There should be no magnification difference between the part of the image below the water and the part above the water if the camera is centered.

For the case of flat viewports, the effects of refraction result in a change in the effective lens focal length [106], given as

$$f_{uw} = 1.33f_{air} \quad (5.3)$$

where  $f_{uw}$  is the effective focal length in water and  $f_{air}$  the focal length in air. This increase in the effective focal length of the system reduces the camera field of view and must be accounted for when computing the lens aperture number.

### 5.2.3 Lensing effects

The fundamental radiometric relation expresses the amount of light incident on the lens that reaches a pixel at the sensor surface [176]:

$$E_I = L \frac{\pi}{4} \frac{1}{N^2} \cos^4(\alpha) \quad (5.4)$$

where  $L$  is the scene radiance,  $N$  is the lens aperture number and  $\alpha$  is the angle between the principal ray and the ray through the pixel. The  $\cos^4(\alpha)$  term models natural vignetting, a process by which illumination decays towards the sensor edges. Additionally, some light is lost as it travels through the lens. This transmission loss depends on the quality and construction of the lens and usually ranges between 5% and 20% [143].

## 5.2.4 Camera response

Light that reaches the camera sensor is converted into an electrical signal. In the model, we assume the use of machine vision cameras with linear sensor response functions, though we note some consumer cameras have non-linear camera response functions, designed to mimic the chemical response of analog film. Grossberg et al. [65] studied the space of camera response functions. Debevec et al. [44] presented experimental methods to determine the camera response function from a set of images. Jiang et al. [90] further modelled spectral sensitivity functions of color camera sensors and proposed experimental methods to obtain them from color board images. The model assumes the sensor response is linearly dependent on the light intensity, with varying sensitivity to different wavelengths. The dependency of the sensor response on wavelength is described by the quantum efficiency curve. The total number of absorbed photons can be computed by dividing the spectrum energy, weighted with the quantum efficiency curve, by the energy of a photon:

$$\mu_e = \frac{At_{exp}}{hc} \int_{\lambda_a}^{\lambda_b} \Phi(\lambda) \cdot \lambda \cdot \eta(\lambda) d\lambda \quad (5.5)$$

where  $A$  is the pixel area [ $m^2$ ],  $\Phi$  is the irradiance spectrum [ $W/(m^2nm)$ ],  $t_{exp}$  [s] is the exposure time,  $h$  is Planck's constant,  $c$  is the speed of light in air [m/s],  $\lambda$  is the wavelength [m] and  $\eta(\lambda)$  is the sensor quantum efficiency as a function of wavelength. Following the EMVA1288 standard [84], the digital sensor response signal  $\mu_y$  can be computed as:

$$\mu_y = \mu_{y,dark} + K\mu_e \quad (5.6)$$

where  $\mu_{y,dark}$  is the sensor mean dark signal, and  $K$  is the system gain.

The physical parameters for each sensor are published by camera manufacturers (eg. [57]) or can be obtained experimentally.

## 5.2.5 Gain and Signal to Noise Ratio

Similar to changing the ISO for film cameras, digital machine vision cameras can have a gain applied to the sensor response signal. This decreases the amount of scene light necessary to expose

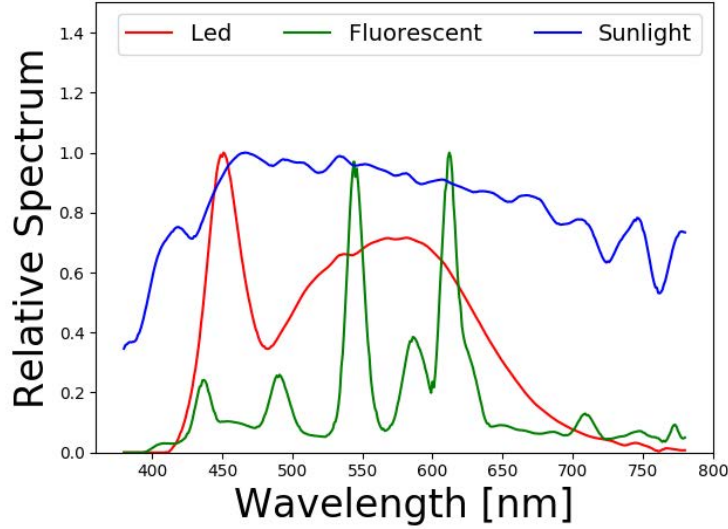


Figure 5.3: Radiance spectrum for different light types

the image. However, the image noise is also amplified when a gain is applied, resulting in a reduction of the image Signal to Noise Ratio (SNR). SNR is an important consideration, especially for image tasks requiring feature matching [107], and should be a parameter decided by the camera system designer. There are three sources of image noise: dark current noise, described by the normally distributed variance  $\sigma_d^2$ ; quantization noise from the analog digital conversion, described by the normally distributed variance  $\sigma_q^2$  and the overall system gain  $K$ ; and shot noise inherent to light, described by the number of incident photons on the sensor  $\mu_p$  and the sensor quantum efficiency  $\eta$ . The image SNR is calculated as [84]

$$SNR = \frac{\eta\mu_p}{\sqrt{\sigma_d^2 + \sigma_q^2/K + \eta\mu_p}}. \quad (5.7)$$

The camera system design tool allows setting a gain value and will display the calculated image SNR for the target average exposure value.

## 5.2.6 Operational Considerations

Besides the physical characteristics of the water and selected equipment (camera, lens and lights), the operational requirements also highly influence the design space. The most significant of these requirements include:

1. Minimum overlap between images: Overlap between consecutive images is required in order to perform photomosaics, 3D reconstructions or visual navigation. The amount of required

overlap, together with the vehicle speed and working distance will determine the image acquisition frequency  $f$ :

$$f = \frac{v}{FOV_{x/y}(1 - OVR)} \quad (5.8)$$

where  $v$  is vehicle speed [m/s],  $FOV_{x/y}$  is the spacial field of view of the image in the direction of motion [m], and  $OVR$  is the fraction of consecutive image overlap.

2. Focal depth of field (DoF): When running AUV imaging surveys over rocky bottoms or coral reefs, it is frequent for the terrain height to vary significantly. It is desirable that the entire image remains in focus, so the required focal DoF must be selected accordingly. Whether a pixel is in focus or not is determined by the circle of confusion, which describes the area of the sensor across which a point source of light is spread. Light rays originating within the focal range will project a circle of confusion on the sensor under an acceptable area threshold. The DoF is controlled by an inverse relationship with the camera aperture. However, there is a trade off, as decreasing the size of the camera aperture decreases the amount of light that reaches the lens and therefore increases the required amount of light in the scene. The DoF can be computed as:

$$DoF = \frac{2Ncf^2s^2}{f^4 - N^2c^2s^2} \quad (5.9)$$

where  $N$  is the lens aperture number,  $c$  is the diameter of the circle of confusion,  $f$  is the focal length, and  $s$  is the distance at which the camera is focused.

3. Motion blur: Motion blur is a great concern for underwater imaging platforms operating in low light. The amount of blur is dependent on the speed of the vehicle  $v$  [m/s], the camera field of view in the direction of motion  $FOV_{x/y}$ , the sensor resolution in the direction of motion  $RES_{x/y}$ , and the exposure time. The maximum exposure time  $t_{exp}$  [s] to keep motion blur less than a set number of pixels  $PIX_{blur}$  is given as:

$$t_{exp} = \frac{PIX_{Blur} \cdot FOV_{x/y}}{v \cdot RES_{x/y}} \quad (5.10)$$

4. Spacial field of view (FOV): The camera spacial FOV or area covered by the image is influenced by lens selection and distance to the target  $D$  [m]. It can be computed as:

$$FOV_{x/y} = D * \frac{SS_{x/y}}{f} \quad (5.11)$$

where  $f$  is the lens focal length [mm], and  $SS_{x,y}$  is the physical dimension of the sensor in  $x$



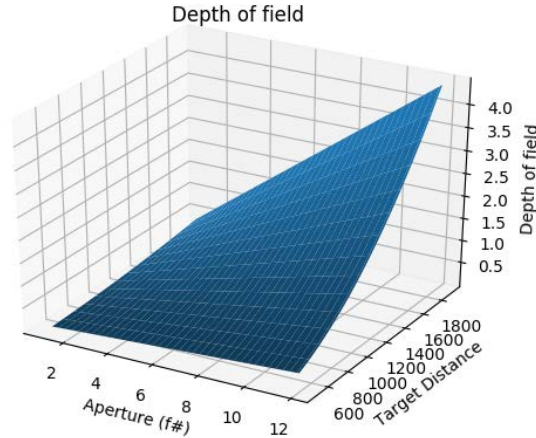


Figure 5.4: Depth of field as a function of focus distance and aperture

or  $y$  [mm].

### 5.3 Software

Taking the previously defined relations between sensors, lenses, light sources and water light propagation into account, users and designers of underwater camera systems may wish to answer questions like what sensor is best for a given operational profile? What are the lighting requirements for a specific camera? Or what aperture and shutter speed should be used for a given deployment scenario? In order to quickly answer questions like these we have developed an open source software design tool that performs parametric analysis of an underwater camera system.

The tool allows the user to either input the light type and lumen intensity or load a custom light spectrum if available. Three Jerlov oceanic water types and five coastal water profiles are provided to analyze different attenuation rates, with an option to also load custom attenuation profiles. Lenses are defined by their focal length and their transmission loss, which may be specified either as a constant or by loading a custom wavelength dependent attenuation profile. Profiles are included with the program for five different camera sensors, and new sensors can easily be added if EMVA specifications are available from the manufacturer. The operational requirements of the camera system are specified in terms of the maximum acceptable motion blur, the minimum acceptable DoF, the expected vehicle altitude and speed above the seafloor, and the desired percentage overlap of consecutive images. Other selectable parameters include the camera orientation with respect to the direction of vehicle motion, and the geometry of the camera housing viewport. With

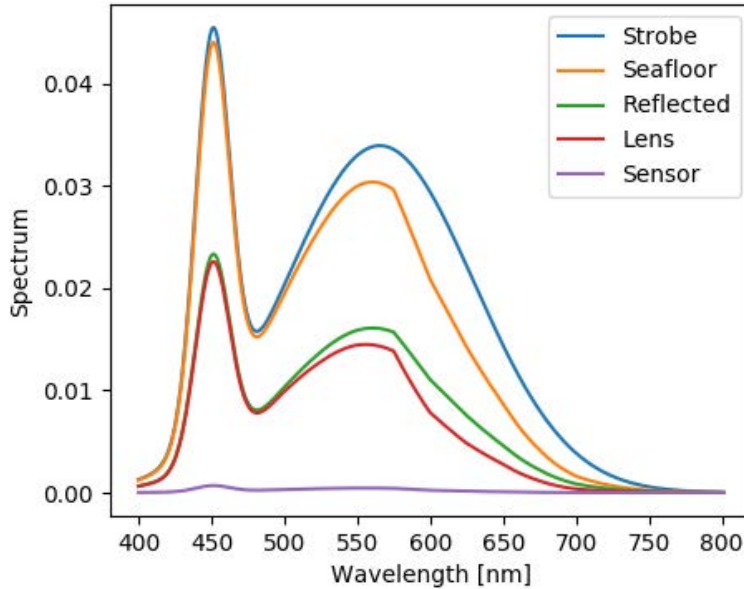


Figure 5.5: Spectrum of light as it propagates through the water, attenuates, reflects and travels through the lens onto the sensor.

a given set of these parameters, the software computes the average camera response, minimum operational framerate, minimum exposure time, and minimum aperture number. In addition to the average camera response, the software can also generate visualizations of the parameter space for the given configuration. Figure 5.4 shows an example plot over a set of parameters, where the dependence of the DoF on aperture and the distance to the imaged target is visualized. Figure 5.5 shows an example plot of how the light spectrum is decayed as it propagates from the light source to the camera, helping contextualize the main sources of light reduction for a specified water environment. Similar plots may be generated by the software for the camera frame rate, exposure time or water attenuation profiles.

## 5.4 Validation Experiments

The camera response simulation pipeline is validated experimentally in a lab environment. We tested with two monochrome cameras, a Blackfly BFS-U3-51S5M from FLIR with a Sony IMX250 sensor and a Prosilica GT-1380 from Allied Vision with a Sony ICX285 sensor. The cameras were mounted on the outside of an 46 cm x 46 cm x 46 cm freshwater tank, with the camera axial direction perpendicular to the clear acrylic tank wall. A diffuse white target board was placed on the opposite side of the tank. Figure 5.6 illustrates the experimental setup. Measurements were taken in dark ambient light conditions, with scene light being provided by a Fix-Neo25000DX 25 klm diving light positioned above the camera and against the outside tank wall.

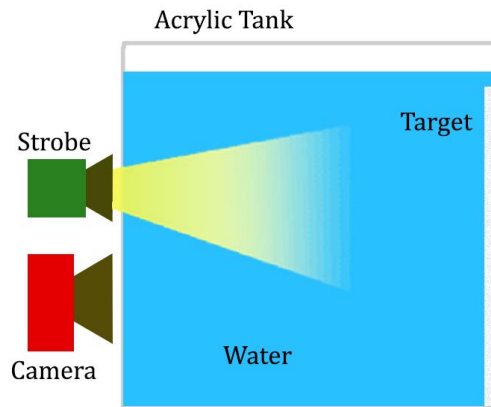


Figure 5.6: Experimental setup for verifying image formation model.

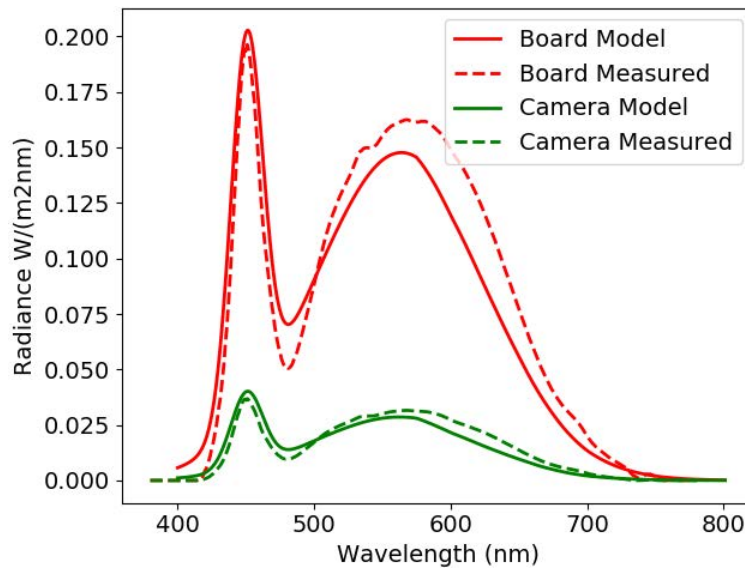


Figure 5.7: Comparison of measured and estimated light spectrum at both the target board as well as the camera position

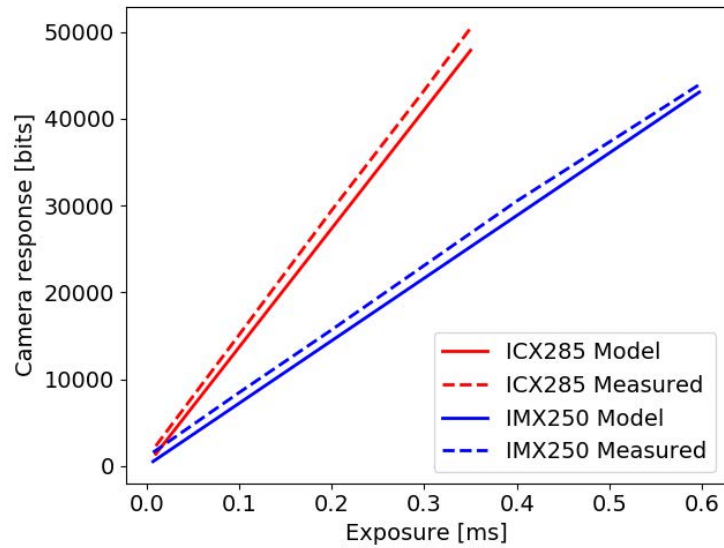


Figure 5.8: Measured and model predicted camera response curves for two different sensors under the same experimental conditions.

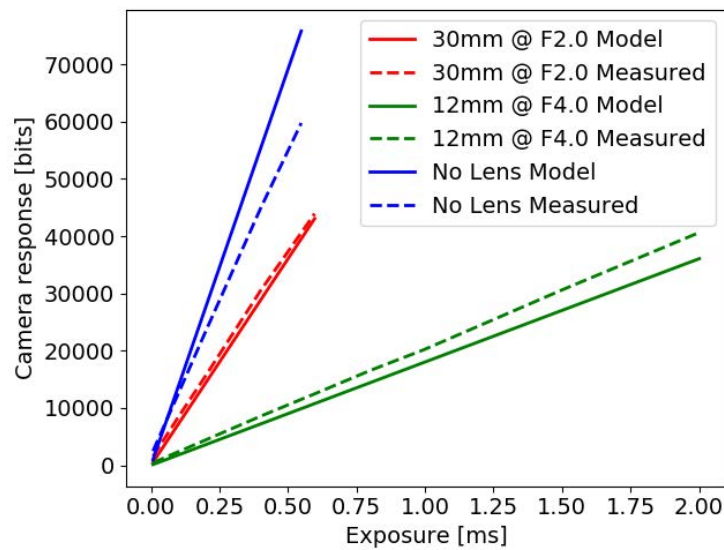


Figure 5.9: Camera response for two different lenses and without a lens.

The light spectrum incident on the camera sensor was measured using a Sekonic SpectroMaster C-7000 lightmeter. The spectrometer was placed in a waterproof enclosure to perform spectrum measurements inside the tank

Figure 5.7 shows the measured light spectra versus those predicted by the model for a generic LED light source. The spectra are plotted for the light that was incident on the target surface, in red, and the light reflected back to the camera lens, in green. The model source light spectrum was calculated with the nominal luminous intensity provided by the manufacturer and a half beam angle of 40deg, accounting for the change in beam angle from the manufacturer stated value due to refraction. The predicted model spectra, both at the target surface and at the camera lens, are very similar to the measured spectra in shape and size. Figure 5.8 shows the response of the two different cameras to the light spectrum shown in Figure 5.7. Both cameras had the same 30 mm lens mounted with the aperture set at F2.0. The predicted responses from the model closely follow the measured values. We also compared the response of one camera with different lens and aperture configurations, including no lens, a 30 mm lens with aperture F2.0, and a 12 mm lens with aperture F4.0. Figure 5.9 shows the measured versus the model predicted average camera responses for this experiment. For all camera experiments, the predicted responses from the model closely follow the measured responses, demonstrating the model is a good approximation of the real system and will give reliable predictions over the design space.

## 5.5 Conclusion

In this chapter we have shown how underwater optical systems can be coarsely simulated by a set of simple equations, and we have developed a user-friendly interface to guide the component and parameter selections of such systems. The presented tool will enable researchers and engineers tasked with the development of underwater camera systems to better understand the available design space, analyze trade-offs in light, sensor and lens selection, and guide early design choices.

## CHAPTER 6

# Automating Underwater Vehicle Manipulator Systems

### 6.1 Motivation

A growing body of evidence suggests that the Earth is not unique in containing liquid water [112, 61, 122, 99], an essential ingredient for carbon-based life. Recent indications of water geysers emanating from moons of Saturn and Jupiter, including Enceladus [138] and Europa [9], suggest that they may contain subsurface oceans with active hydrothermal venting [120, 79]. Here on Earth, ocean floor hydrothermal systems and cold seep sites have long been known to host diverse chemosynthetic ecosystems that rely on the redox potentials of deep Earth fluids emitted from these sites to derive biochemical energy [85, 22], and may serve as analogs for oases of life elsewhere in our solar system and beyond. However, exploration for life within the distant oceans of Europa and Enceladus remains a daunting technological challenge. Robotic submersible vehicles equipped with manipulators provide a practical means for sample analysis and collection, enabling flexibility and dexterity without requiring precise and energetically costly positioning of the vehicle. Planetary landers such as the Mars Rovers have historically relied on human teleoperated manipulation using manually generated scripts [110, 58, 109] to collect samples. However, teleoperation of robotic subsea vehicles within these putative ocean worlds is impractical because of high communication latencies (e.g., on the order of an hour for Europa). Thus, robotic missions must be capable of fully automated manipulation.

Marine robotic platforms such as remotely operated vehicles (ROVs) equipped with manipulators provide a useful testbed to develop automated manipulation and sampling technologies as analogs for space missions. Although Earth's gravitational constant is higher than Europa and Enceladus, these moons' estimated ice thicknesses of up to 30 km [80, 16] are expected to present operational challenges, such as extreme pressure, near-freezing temperatures, and corrosion that are similar to Earth's deep ocean environments. While autonomous underwater vehicles (AUVs) have been used for under-ice surveys for nearly 50 years [59], deep

ocean missions that require sample collection and return using manipulator arms are generally conducted using ROVs under direct human piloted control with cable-tethered communication. Only limited attempts at autonomous manipulation have been made in natural ocean environments [123, 164, 43, 160, 148, 170]. The comparative lag in subsea manipulator autonomy behind terrestrial systems can be at least partly attributed to commercial systems being historically designed for direct teleoperation, with limited command modes and feedback, low control loop frequency, and poor repeatability [169]. Despite these challenges, we demonstrate an automation framework that is compatible with existing commercial manipulator systems and that automates many high level tasks, while reducing risk through visual based scene understanding and pilot supervision.

In this paper, we consider the challenge of automated subsea manipulation and sample collection using existing ROV platforms as a technology analog for an under-ice exploration missions to Europa or Enceladus. We discuss the challenges that deep seafloor environments pose to automated robotic intervention and propose an architecture that overcomes many of these challenges. The system that we describe can be integrated on existing ROVs with minimal hardware requirements, namely, a vehicle-mounted stereo camera and a manipulator-mounted fisheye camera. We investigate the practical use of our perception methods to estimate the vehicle configuration, dynamically localize tools, and ground the transform between the natural scene reconstruction and the structured vehicle workspace. The manipulator control and vision processes that serve as the basis of this automation framework can be readily adapted to a variety of hardware configurations, making them suitable for a wide range of robotic platforms, including space flight systems. We demonstrate the flexibility of this framework through separate field trials performed with two different classes of ROVs equipped with substantively different manipulators. Figure 6.1 shows a conceptual diagram of how our system integrates with an ROV. In the current system implementation, a topside machine performs all processing using camera and manipulator data streamed from the vehicle over a high-bandwidth tether.

We conducted testing and field trials in progressively challenging environments, initially in laboratory settings and tank testing, followed by 11 dive missions at the Central American Pacific shelf margin of Costa Rica to operational depths of approximately 1800 m. This area of the Costa Rican accretionary prism is a well studied region with localized ocean floor fluid expulsion sites that host diverse assemblages of extremophile organisms [71, 103, 111, 157, 165]. Following the completion of the Costa Rica expedition, our team conducted a series of five dive missions to depths of 500 m within the potentially hazardous craters of the Kolumbo and Santorini calderas. These sites of active volcanism contain localized areas of high-temperature hydrothermal venting causing environmental hypercapnia [27, 32], which host non-calcifying chemosynthetic organisms that may resemble those that arose early on Earth, prior to the advent of its oxidizing atmosphere.



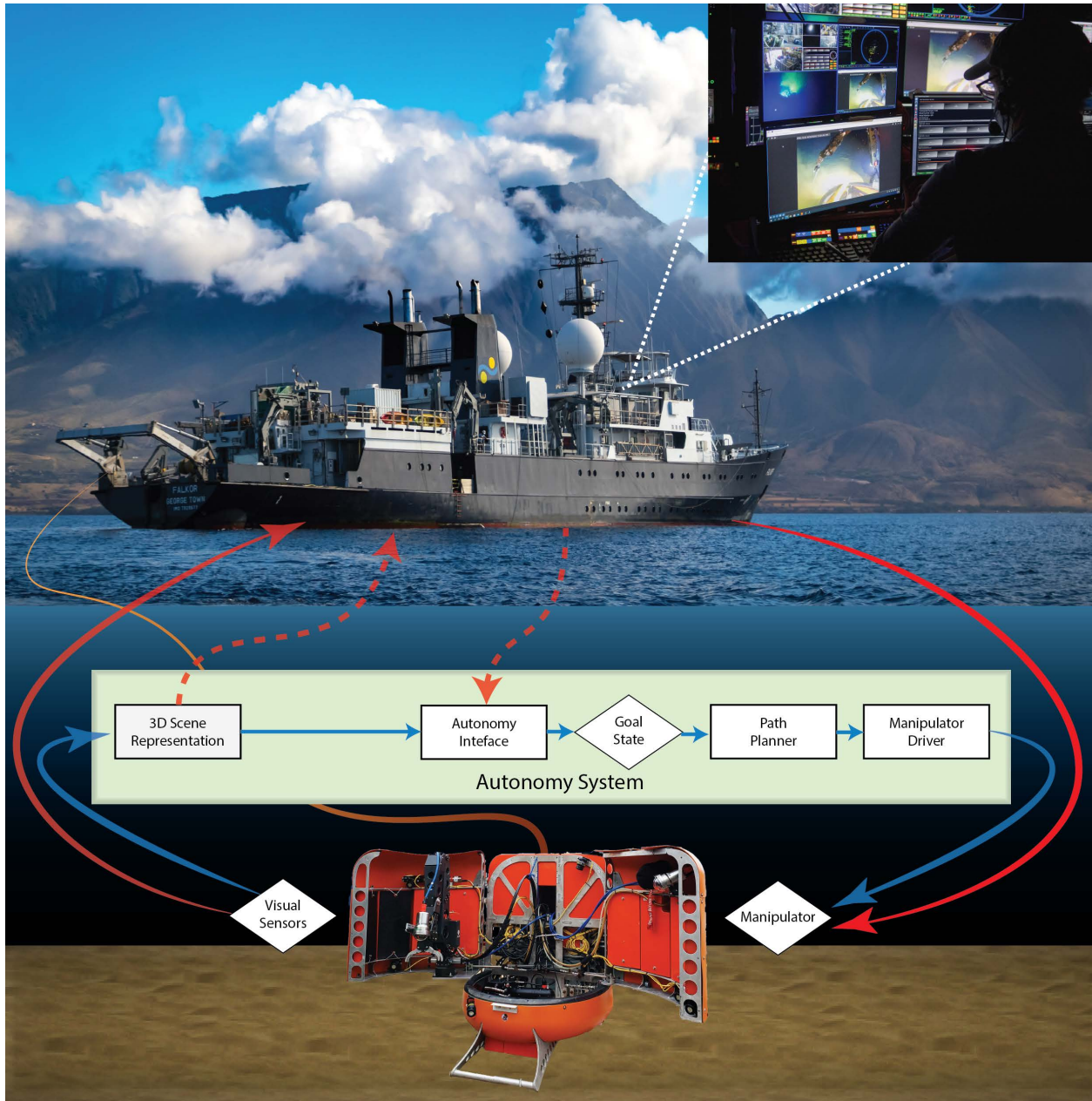


Figure 6.1: Conceptual graphic of the our control system for an underwater intervention vehicle. The autonomy system runs on a topside desktop computer with visual sensor data and manipulator coms streamed over a high bandwidth tether from the vehicle. Solid red flow lines represent standard teleoperated control from a surface ship. Blue flow lines represent our automated system. Red dashed lines represent interfacing between the pilot and the autonomous system, where, in this work, the pilot acts as the high level task planner and interfaces with the automated system through a graphical scene representation and task level controller. Eventually, the pilot would be replaced with an automated mission planner that could issue high level tasks.



The paper is organized as follows. We begin in Section 6.2 with an overview of previous work on automated underwater manipulation. Section 6.3 briefly outlines our strategy for mission planning and operations, and describes in detail the architecture of our automated manipulation system. Section 6.4 examines the results of experimental missions during field demonstrations using the *SuBastian* ROV [28] and *NUI* Hybrid Remotely Operated Vehicle (HROV) [20]. Section 6.5 draws on these field results and experiences to discuss advances as well as the limitations and potential failure modes of our perception and control methods and examines how this research may help to advance both automated ROV operations here on Earth and future space flight missions to explore for life within ocean worlds elsewhere. Section 6.6 identifies promising directions for future research.

## 6.2 Background

There is a rich body of literature on underwater vehicle manipulator system (UVMS) control. This section provides a brief review of the work most related to our approach which have demonstrated their methods in experimental trials. For a thorough discussion of the prior work on UVMS systems, we refer the reader to [169].

[81] and [23] describe some of the pioneering work on automating UVMSs, where demonstrations included 3D graphical renderings of an ROV's configuration and workspace, real-time visualization of manipulator motion plans, and Cartesian space end-effector control. More recent works under the large-scale research projects RAUVI [43], TRIDENT [160], TRITON [148], PANDORA [37], and MARIS [166] focus on tightly coupled control of the 140 kg displacement *Girona 500* AUV outfitted with a customized electric manipulator to perform free-floating intervention tasks. The PANDORA project explores the ability to learn the vehicle and manipulator trajectories by demonstration. The other projects combine vehicle and manipulator motion generation under a task priority framework, where the manipulator control law is a function of the vehicle velocity. Building on these works, the MERBOTS project [190] offers a significant advancement towards automated UVMS control by integrating the ROS-based MoveIt! motion planning framework with the intervention AUV to generate combined vehicle and manipulator motion trajectories in Cartesian space for free-floating intervention tasks. While this body of work provides key advancements towards automated free-floating intervention, limitations make it difficult for many actively operated UVMSs to adopt these methods. Among them, integrating such a tightly coupled control system with existing UVMS platforms would require significant modification to the software architecture, which is particularly problematic for commercial systems. Additionally, the dynamic coupling effect between the vehicle and manipulator during free-floating intervention can strongly affect the trajectory tracking performance, necessitating very slow actuation of the

vehicle and manipulator. Lastly, this control approach is designed for high-precision electric manipulators that support velocity-based control, whereas most manipulators on operational UVMS platforms are hydraulic and support only position set point commands with limited precision and repeatability.

Hydraulic manipulators have orders of magnitude higher power-to-weight ratios compared to their electric counterparts and are generally more reliable, making them the manipulator of choice for commercial ROV systems. Though recent commercial electric manipulators have entered the market, their significantly higher power requirements make them practical only for ROVs that have power supplied over a tether. For vehicles like the *NUI* HROV, which carries all power onboard, low-power hydraulic manipulators remain the most practical choice. However, the limited precision and feedback of hydraulic manipulators present challenges for automation, and little work exists that addresses these challenges. [73] demonstrated precision control of a hydraulic manipulator to plug a deep-sea connector. [164] perform pre-programmed motion following and operator control of a hydraulic work class manipulator. Using Cartesian space end-effector control, they demonstrate operator-guided push-core sampling in the deep ocean. [191] perform visual servoing and target grasping with a custom 7-DoF hydraulic manipulator. [170] demonstrate impressive visual servoing of a working class hydraulic manipulator using position-based control, with feedback provided by fiducials detected from a wrist-mounted camera. Their results include grasping and turning T-bar valves and tracking targets in motion with the end-effector.

Our system builds on these prior approaches to UVMS control, where we demonstrate the effective integration of the MoveIt! motion planning framework [39] with a work class ROV manipulator system for automated planning and control in obstructed scenes. We take a decoupled approach to manipulator control that assumes the vehicle holds station (i.e., rests on the bottom) during the manipulation task. This assumption is motivated by the goal of having the system widely transferable among existing ROV systems. This decoupled approach enables our manipulator control system to be integrated externally from the existing UVMS control systems, providing high-level autonomy with flexibility to be integrated onto a wide array of ROV classes and manipulator arms, including both electric and hydraulic systems.

Important to automating UVMSs are the problems of visual scene understanding and target localization, whether the target be a tool to grasp, a valve to turn, or a sample location in an unstructured environment. Subsea perception is a particularly challenging problem for a number of reasons: turbidity degrades image quality; evenly lighting the scene is very difficult; variable wavelength-dependent absorption and scattering properties of the water column attenuate light and reduce color and photometric contrast; and gathering underwater datasets for developing computer vision methods is expensive. Despite these challenges, computer vision remains the primary means of performing target localization for automated UVMS platforms. Most prior works on UVMS au-

tomation rely on fiducials or known geometric shapes that retain high contrast underwater. [123] use large spherical markers attached to a target and a circle shape edge detector algorithm to localize the marker from a video feed. [160] localize a black box object on a harbor seabed by first constructing a visual mosaic from a pre-intervention survey dive with a downward-facing stereo camera, and then matching an image template of the black box to the mosaic. [37] localize a known panel during intervention operations by registering interest points against a template image. They then estimate the orientation of valves on the panel based on edge detection. [148] and [190] use fiducial markers to localize a panel with *a priori* known relative positions of the turn valves and connector plugs. [190] also use fiducials on the end-effector of the manipulator to update the manipulator calibration in real-time. [166] use color and geometric shape segmentation of RGB images to detect the pose of a cylindrical pipe of known size. Under the DexROV project, [17] process stereo point clouds into a 3D occupancy map, while also using fiducial markers to detect and localize a panel with known structure that was projected into the planning scene.

Building on the long history of fiducials as a robust visual cue for underwater computer vision methods, our work extends the use of fiducials to detect the pose of graspable tools carried on-board the ROV, estimate dynamic vehicle configurations in real-time, and ground the relative reference frames in the planning environment. We demonstrate the use of fiducials in a way that is practical for field deployments with an underlying vision system that can effectively localize tools and target objects within the workspace, as well as reconstruct the workspace for obstructed motion planning. Fiducials also enable the collection of annotated image datasets in natural deep seabed environments that support the development of advanced perception methods for scene reconstruction, and target detection and localization.

## 6.3 System Overview

The following sections provide an overview of the different components of our system, including the mission architecture for field operations and the methods for perception and control.

### 6.3.1 Mission and Vehicle Platform Architecture

Field demonstration and validation include two research cruises, conducted east of the Cocos and Caribbean tectonic subduction zone along Central America’s Pacific continental margin (9.0 N 84.5 W), and within the Kolumbo and Santorini Calderas of the Hellenic volcanic arc in the southern Aegean Sea (36.52 N 25.48 E and 36.45 N 25.39 E, respectively). The sites, which are known to host oases of chemosynthetic communities associated with hydrothermal and seafloor hydrocarbon seeps, were chosen as NASA TRL-6 demonstration locations for analog astrobiology exploration



Figure 6.2: Photograph taken by the *NUI* vehicle within the Kolumbo volcano crater that shows an overhanging vertical wall of columnar lava. Colonization of the lava surfaces by relatively uncommon lollipop sponges (*Stylocordyla pellita*) are visible as white dots within the image.

missions. These campaigns utilized a sequentially nested survey method with a coordinated team of heterogeneous robotic platforms that relied on automated planning tools to rapidly synthesize vehicle missions in response to newly acquired information [183, 10]. To better approximate an analog space flight mission scenario, surface ships operated as orbiters, conducting multibeam sonar bathymetric mapping of the Pacific [182] and Aegean [141] campaign sites, with coverage areas of  $2.000 \text{ km}^2$  at 30 m resolution and  $48 \text{ km}^2$  at 10 m resolution, respectively. These maps informed the mission planning for autonomous underwater gliders (AUG), which acted as long-range in-situ reconnaissance drones, conceptually similar to NASA's *Ingenuity* and *Dragonfly* vehicles, conducting reconnaissance missions of between 1 km and 500 km in length at standoff distances to within 15 m of ocean floor obstacles in order to identify potential areas of scientific interest [183, 51]. Automated AUG mission planning considered resource (e.g., time and power) and risk constraints [179, 178], and adaptively replanned missions based on inferred sites of scientific interest that correlated with the presence of active ocean floor hydrocarbon cold seeps and hydrothermal vents. Using information gained by the surface ship sonar and AUG missions, the automated planning process then generated viable mission sequences that the ROV used to investigate areas of highest estimated information gain [183]. During these missions, the ROV acted as a lander, outfitted with a manipulator for automated sample collection and return. The hazardous deep ocean environments explored as part of these ROV missions are considered probable analogs for environments (Fig. 6.2) that may exist on other ocean worlds.

Table 6.1: Comparison of *SuBastian* and *NUI* configurations.

	Depth rating meters	Displacement kilograms	Lateral excursion (tethered) meters	Power draw (typical) watts	Endurance hours	Manipulator type	Manipulator reach meters	Payload capacity kilograms
<i>SuBastian</i>	4500	3200	< 500	40000	unlimited	2 x 7-DoF	1.9	200
<i>NUI</i>	2000	2000	20000	2500	6 to 8	7-DoF	1.3	100

The two ROVs used for these demonstration campaigns, *SuBastian* and *Nereid Under Ice (NUI)* are substantially different in design and purpose (Table 6.1). *SuBastian* is an exemplar of modern deep ocean work class ROVs, with its power, communications, and navigation net provided via an armored cable by its attendant surface ship, the *R/V Falkor*. *SuBastian* is equipped with twin 7-DoF Schilling Titan-4 hydraulic manipulator arms (Schilling Robotics, Davis, California) and is a fully teleoperated vehicle that can operate at horizontal excursions of up to 500 m laterally from the *R/V Falkor*. In contrast, *NUI* is a HROV that relies on its own battery power and uses an un-armored fiber optic link (roughly the diameter of a human hair) for optional communication with an attendant surface ship, and can operate as both an ROV and an AUV. When in tethered ROV mode, *NUI*'s power and telemetry architecture enables lateral excursions of up to 20 km from the attendant surface ship. To aid hydrodynamic efficiency, *NUI* has articulating bow doors that can be closed and act as a fairing during transits and AUV missions. In contrast to *SuBastian*'s twin Titan-4 architecture, which is configured to maximize ROV work area and dexterity, *NUI*'s starboard door is equipped with a single, custom 7-DoF hydraulic manipulator (Kraft Telerobotics, Overland Park, Kansas) that is optimized for energy efficiency. This emphasis on efficiency comes at the expense of reductions in available payload and work space, the usable range of motion, control precision, lighting field, and available viewing perspectives.

### 6.3.2 Perception

A vehicle capable of automated intervention must have an effective means to self-localize within the environment and visually reconstruct the workspace to complete the mission tasks. We adopt a vision system consisting of computer vision cameras, that takes into consideration three primary criteria. First, the system must be capable of generating a 3D reconstruction of the manipulator workspace, enabling the motion planner to avoid obstacles and generate safe, collision-free paths. Second, the system must be able to localize a set of known objects, such as tools, and guide the manipulator to grasp them. Third, the system must easily integrate with existing robotic platforms. Our vision system is composed of a vehicle chassis-mounted stereo camera pair with a fixed-baseline and a manipulator wrist-mounted fisheye camera (Fig. 6.3). The stereo pair observes the manipulator workspace, including part of the tool tray and the scene working area. The system



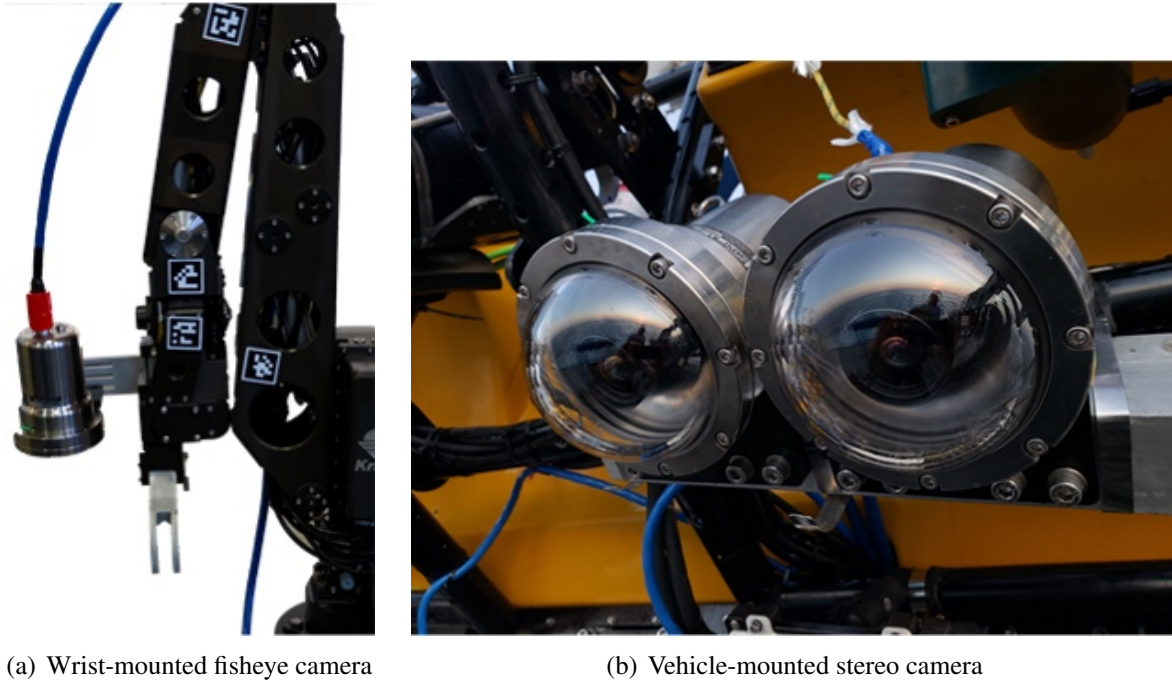
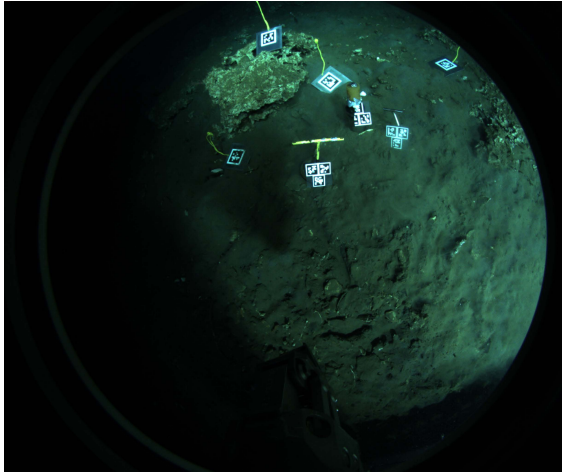


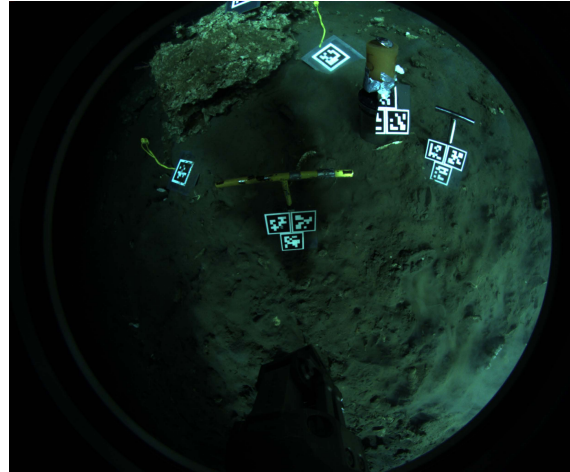
Figure 6.3: The vision system for autonomy is composed of (a) a wrist-mounted fisheye camera and (b) a vehicle-mounted stereo pair (shown here mounted on the *SuBastian* ROV). The vision system can be easily integrated onto existing vehicles.

uses the stereo to generate 3D point clouds of the workspace for scene reconstruction, assist with localizing tools in the tool tray, and visually track dynamic vehicle reference frames that are otherwise not observable (e.g., the position of the *NUI* HROV doors). The wrist-mounted fisheye camera provides a wide-angle view of the scene, and is used to detect objects and acquire dynamic viewpoints of the scene, which may be occluded or outside the field-of-view of the stereo pair. The wide field-of-view of the fisheye compared to a perspective camera enables clear views of objects and scene context at both close and far range (Fig. 6.4), which is advantageous for manipulation.

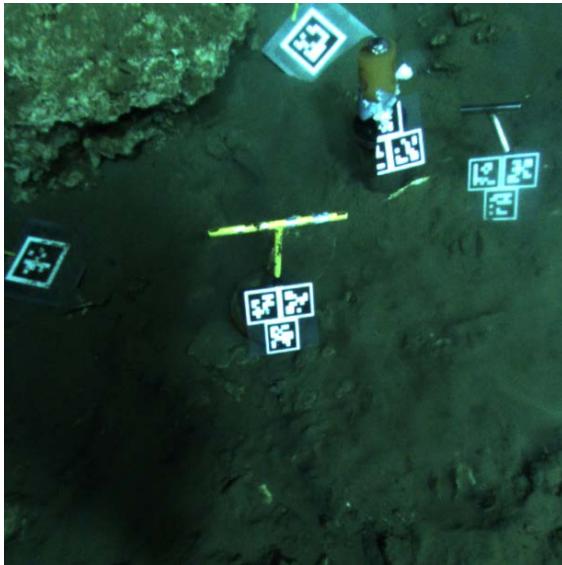
All three cameras are Blackfly model BFLY-PGE-50S5C-C (FLIR, Wilsonville, OR). The stereo cameras use the VS Technology SV-0614H 6 mm  $f/1.4$  lens (VS Technology Corporation, Tokyo, JP), and the fisheye lens is the Fujinon FE185C086HA-1 2.7 mm  $f/1.8$  (Fujinon, Tokyo, JP). The camera housings are custom fabricated with titanium shells and dome viewports (Sexton Corporation, Salem, OR), with a depth rating of 6000 m. A hardware trigger synchronizes the cameras. We calibrate the cameras using images of a checkerboard that the ROV manipulator moves throughout each camera’s field-of-view while the vehicle is submerged. We calibrate the stereo cameras using the ROS stereo camera calibration package. We calibrate the fisheye camera using the Kalibr toolbox [94]. Because the usable field-of-view for the fisheye camera is less than  $180^\circ$  due to occlusions from the housing, we use the pinhole projection model with equidistant distortion. We



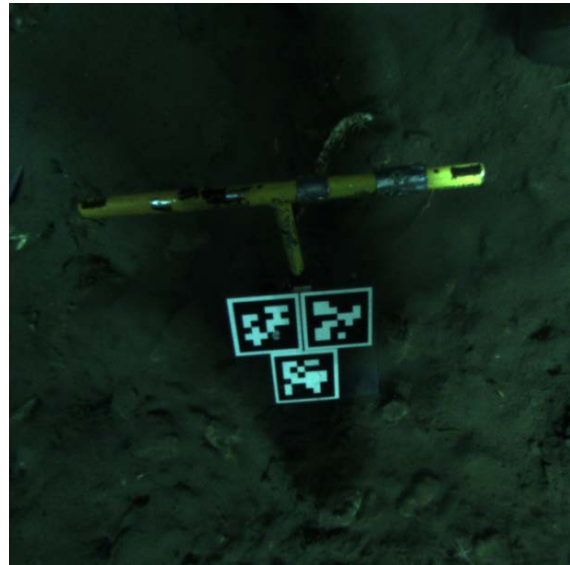
(a) Far fisheye view



(b) Close fisheye view



(c) Far perspective view



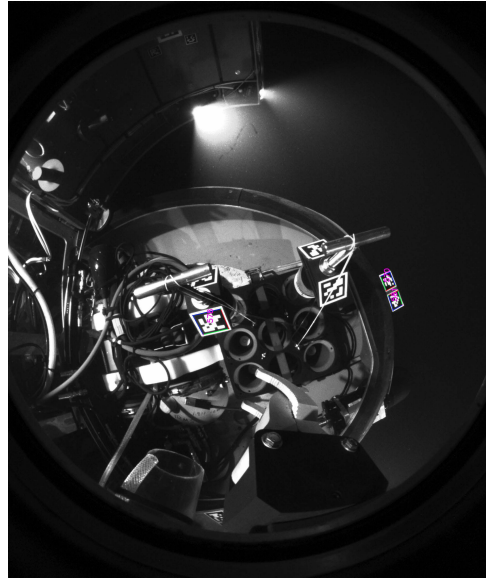
(d) Close perspective view

Figure 6.4: A comparison of (top) the full view of the wrist-mounted fisheye camera in an underwater scene at close and far range compared to (bottom) a  $60^\circ$  perspective rectification, which illustrates the significant increase in the field-of-view provided by a fisheye lens compared to a conventional perspective lens. This increased field-of-view provides significantly better contextual awareness to the vision and manipulation systems, especially when working at close range to the target, which is typical for manipulation tasks.

verify that both the stereo and fisheye calibrations achieve sub-pixel reprojection errors for the checkerboard corners.



(a) AprilTag mount for tools



(b) Fisheye view of tools in tool tray

Figure 6.5: A single type of t-handle was used to manipulate the different tools. The vision system localizes the t-handles using (a) AprilTags affixed to 3D-printed mounts located beneath the t-handle. These tags are detected in (b) images of the ROV tool tray from the wrist-mounted fisheye camera.

### 6.3.2.1 Tool Handle Pose Estimation

Tools carried by the ROV must be localized by the vision system before they can be grasped. It is general practice in ROV operations to use a single type of handle on every tool to provide consistency for ROV pilots. Given a known type of tool and its model, the vision system need only localize the handle for a tool to be grasped and manipulated.

Using data collected with our vision system during the field trials, we developed a novel deep learning-based method, SilhoNet [14] and SilhoNet-Fisheye [15], that estimates the pose of tool handles detected from the wrist-mounted fisheye camera, without the need for fiducials. SilhoNet uses an intermediate silhouette representation to regress the detected object poses. This silhouette representation improves pose regression performance and facilitates training the network on synthetic data, which is especially beneficial when real training data is limited, as is the case for underwater environments. This method achieves promising results on the recorded datasets, but was not ready for integration with the system during the field trials.

During our field demonstrations, we relied on AprilTag markers [?] to localize the tool handles. Our choice of the AprilTag marker was motivated by the results of [49], which show that AprilTags yield the best performance in underwater environments, with the lowest sensitivity to turbidity and variable lighting conditions in comparison to other popular fiducial markers. In this study,



the minimum marker size detectable in an image was approximately 20 pixels, which, for the  $50 \times 50$  mm markers used in our system, equates to an expected maximum detection range of approximately 1.0 m for the fisheye camera and 2.4 m for the stereo cameras. These distances are within the typical working ranges of the manipulators used in our demonstrations. We designed 3D printed mounts that screw onto the t-handle bases and hold AprilTag vinyl stickers (Fig. 6.5, right).

We use the ROS TagSLAM package [149] to detect the fiducials from the wrist-mounted fisheye camera. TagSLAM is built on the GTSAM [45] factor graph library and uses the ISAM2 [93] incremental optimizer for efficient run-time performance. TagSLAM operates in a transform tree completely separate from the world planning environment. Within the TagSLAM environment, the fisheye camera is set as the origin, while the tools with the tag mounts are set as dynamic objects. We optimize the pose of each detected tool with respect to the fisheye camera frame using TagSLAM. The optimized tool pose with respect to the fisheye frame is projected into the world frame through the manipulator kinematics. If the fisheye camera loses sight of a tool, the tool pose within the world scene remains static until the tool is tracked again with TagSLAM.

### 6.3.3 Control

While many existing methods tightly couple vehicle and manipulator motion planning and control, our approach decouples the manipulator and imaging system from other systems on the ROV. This makes it easier to integrate the system with different ROVs and also minimizes risk to the vehicle, as the automation system runs independently of the vehicle’s software stack. This approach also mimics standard ROV operation procedures, in which one pilot controls the vehicle while another pilot controls the manipulator. Our system seeks to replace the direct pilot control of the manipulator with a high-level automation interface that naturally integrates with standard ROV operational procedures. A current limitation of this control approach is a fixed-base assumption while the manipulator is activated. During manipulator operations, the ROV is assumed to be set down on the seabed and essentially acts as a fixed-base manipulator platform during a sampling tasks. When a manipulator command is executed, our system assumes that the scene state remains static until the activation is completed. This assumption of fixing the vehicle position before activating the manipulator follows the standard practice for operating work-class ROVs.

Figure 6.6 shows a diagram of our system architecture. We use the MoveIt! Motion Planning Framework [39] to integrate the outputs of the perception system into the planning scene and to generate collision-free motion plans. MoveIt! directly supports a diverse set of state-of-the-art motion planners and inverse kinematic (IK) solvers. For this work, we used the RRT\* planner [96] with the KDL IK solver. We visualize the planning scene using RVIZ, with out-of-the-box inte-

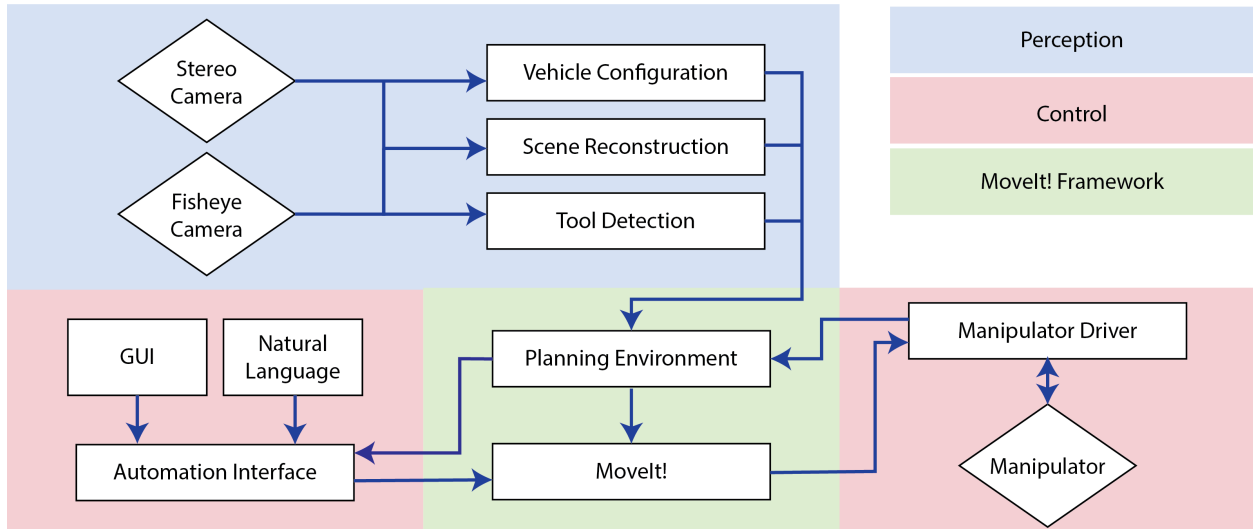


Figure 6.6: A diagram of the overall system, where rectangular blocks represent processes and diamond-shaped blocks represent hardware. Blocks in blue relate to perception. Blocks in red relate to (left) high- and (right) low-level control. Blocks in green are part of the MoveIt! framework around which our system is built. Our system uses the stereo camera to estimate the vehicle configuration (e.g., the pose of the doors on the *NUI HROV*), generate point clouds of the scene that can be fused to produce a 3D reconstruction of the scene, and assist with tool localization. The fisheye camera is used to localize tools, obtain dynamic viewpoints of the workspace, and extend the scene reconstruction. For low-level control, a driver implements a position-based trajectory controller, which integrates between MoveIt! and the manipulator valve controller. For high-level control, we implemented an automation interface to MoveIt! that supports high-level commands. In this work, we implement this interface using a graphical front-end as well as a preliminary demonstration using natural language.

gration with MoveIt!. We generated a kinematic description of the manipulator and vehicle from CAD models, and configured a motion planning environment with MoveIt!. A low-level driver for the manipulator exposes a position trajectory control interface to MoveIt! and interprets motion plans as command packets that it sends to the manipulator. Most work class hydraulic manipulators support only position setpoint commands. For this work, the system encodes the target joint positions and sends them directly to the manipulator valve controller.

### 6.3.3.1 Calibration Procedure

Figure 6.7 illustrates the coordinate frame transforms that must be calibrated for motion planning and kinematic-based control of the manipulator. The end-effector pose follows from the kinematic chain of transforms from the manipulator base frame through each consecutive link, where each transform is parameterized by the joint angle. Hydraulic manipulators generally provide limited joint feedback from position sensors like potentiometers or resolvers, which must be calibrated to

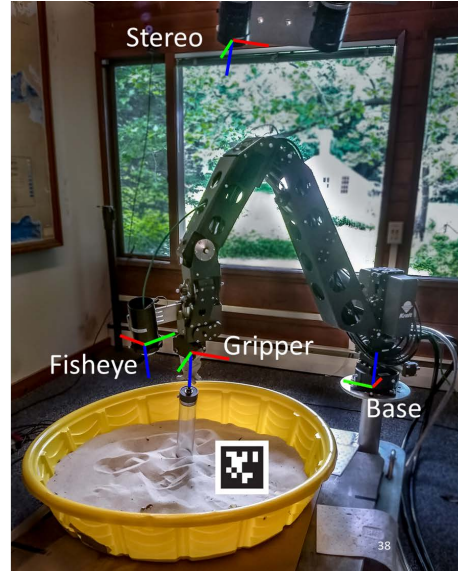
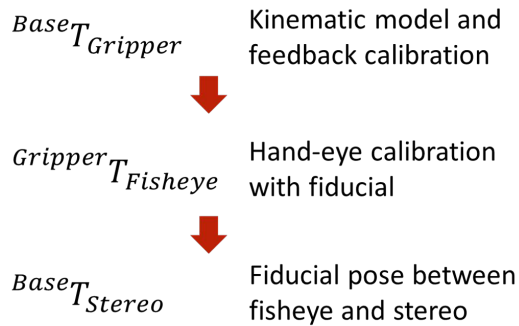


Figure 6.7: (right) An image of our testbed consisting of a Kraft TeleRobotics manipulator, a fisheye camera mounted to the end-effector, and a overhead stereo camera. Together with the manipulator base frame, there are four reference frames (left) which must be calibrated in order to fuse sensor data into a common reference frame and to plan the motion of the arm. Calibration is performed in the order shown on the left, where each transform enables calibrating the next in a bootstrapping manner. The fiducial in the image is included to indicate that AprilTags were placed statically in the workspace to obtain the Gripper-to-Fisheye and Base-to-Stereo calibrations.

the kinematic model. For this work, we assume a linear interpolation between the feedback values at the joint limits. We calibrate the hand-eye transform between the fisheye camera and the wrist link by detecting the fisheye-relative pose of an AprilTag positioned at a fixed location relative to the manipulator base. We perform these detections for a set of different kinematic configurations and then optimize the hand-eye transform using the ROS `easy_handeye` package [180]. When the manipulator base is rigidly fixed relative to the stereo pair, as would be the case for most ROV configurations, the transformation between the stereo pair and the manipulator base frame is calibrated by detecting the pose of a vehicle-affixed AprilTag in the scene in both the left stereo and the wrist-mounted fisheye cameras. We transform the pose of the fiducial from the fisheye camera frame to the manipulator base frame through the kinematic chain, giving the stereo-to-base frame transform as the difference in the tag pose between the two frames. For the *NUI* vehicle, where the stereo is not fixed relative to the manipulator, we used a different approach to estimate the stereo-to-base transform in real-time (see Section 6.4.3).

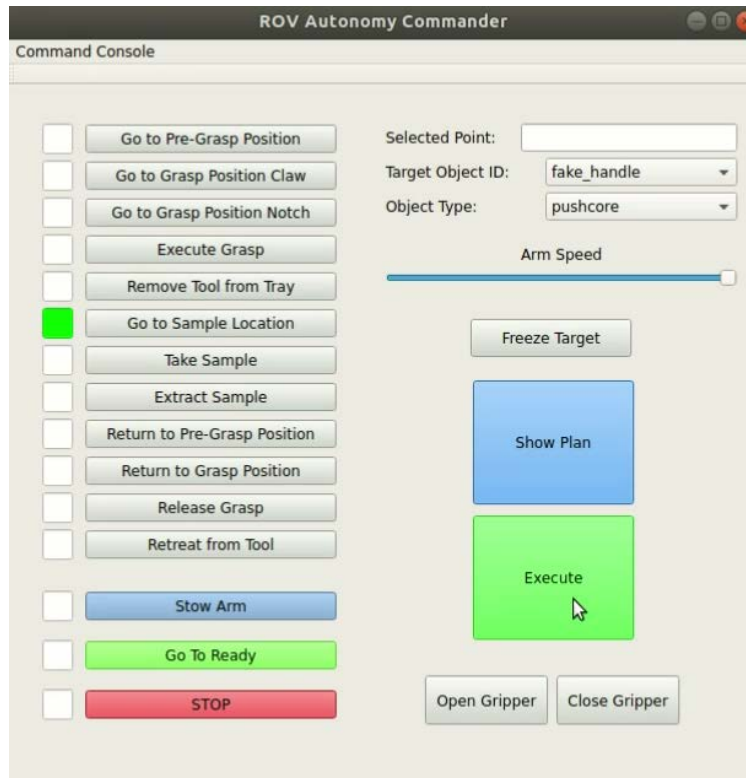


Figure 6.8: A simple interface to the automated system allows the user to configure and step through the automated pick-and-place pipeline. The motion plan for each step is visualized in the planning scene and is only executed upon confirmation by the user, which provides a high-level of safety for the system to be deployed on ocean-going systems.

### 6.3.3.2 Pick-and-Place Interface

Our autonomy framework targets manipulation operations that involve pick-and-place tasks, such as taking a push-core sample. We implement a simple front-end interface (Fig. 6.8) that allows a user to step through a pick-and-place state machine that automates each step of the process, while maintaining a high level of safety through human oversight. The interface visualizes the manipulator motion plan at each step and only proceeds to execute the plan after the operator provides confirmation. The interface allows the user to select a target among a set of tools detected in the scene and then activate a sequence of automated steps to grasp and manipulate the target using pre-defined grasp points. An interactive marker enables the user to indicate the desired sample location in the 3D planning scene. Besides the pick-and-place state machine controller, the interface enables one-click planning of the manipulator to a set of pre-defined poses, immediate stopping of any manipulator motion, and opening and closing of the gripper. The MoveIt! planning environment also allows the operator to command the manipulator to an arbitrary configuration within the workspace through an interactive 3D visualization.

### 6.3.4 System Precision

The maximum precision of our system is limited by both kinematic and visual factors. The KRAFT manipulator uses 11 bit encoders, for an approximate per-joint angular resolution of  $0.176^\circ$ . When the arm is fully extended to 1.3 m, the angular resolution for the shoulder joints equates to a metric arc length resolution of 4 mm at the end-effector. However, this estimate does not account for non-linear effects in the hydraulic actuators, bias in the joint actuation, inaccurate feedback from the joint sensors, or flexing of the arm's mounting base/vehicle door, any of which may significantly reduce the kinematic accuracy of the system. The visual factors that limit precision include the accuracy of localizing the AprilTags from the fisheye camera and the resolution of the stereo reconstruction. Visual precision is dependent on the metric resolution of a pixel projected into the world. For a tag that is 1 m from the fisheye camera, the pixel metric resolution is 1.3 mm, which is the expected best precision for localizing the tags. High distortion of tags near the edges of the fish-eye image is expected and has been observed to reduce the localization accuracy. When processing the stereo images to produce depth maps, the maximum working distance can be tuned based on the maximum disparity over which a feature match is searched across a rectified image pair. In our system, the maximum practical distance we target for stereo reconstruction is 3 m, which is well beyond the manipulator reach and beyond which lighting and haze effects severely degraded the image quality. For a viewing range of 3 m, the metric pixel resolution in the stereo view is 1.7 mm. Due to feature smoothing by the SGM correlation window, the actual reconstructed spatial resolution is coarser. In practice, we have observed that the kinematic accuracy is the limiting factor on the precision of our system, due to the many sources of kinematic error in hydraulic manipulator systems.

## 6.4 Experiments and Field Results

### 6.4.1 Automated Pick-and-Place Demonstration on Testbed

To prove the viability of our system before deploying it in the field, we demonstrated the full pick-and-place pipeline on a hardware testbed (Fig. 6.9) that mimics the configuration of the vision system and manipulator as they would be mounted on an ROV. The testbed includes a Kraft TeleRobotics manipulator identical to the one that we use for the field deployments with the *NUI* vehicle. The planning environment simulates the manipulator being mounted on the *NUI* HROV. The stereo point cloud is projected into the planning scene to inform placement of the sample marker. As described previously, we estimate the t-handle pose from the wrist-mounted fisheye camera by detecting the AprilTags mounted below the handle. We executed each step of the automated pick-and-place interface successfully, with no manual control input. The system grasped

the t-handle based on the detected pose, and the planner found and executed a manipulation path to the sample location marker, which was placed at a non-trivial angle, touching a rock in the scene. The rock was placed on its end in a delicately balanced position, and the manipulator was able to bring the tool into contact with the rock with enough precision that the rock remained standing. The tool was then returned to the position from where it was grasped. This full experiment was repeated multiple times, though not without some grasp failures, due to noise in the visually estimated pose of the t-handle. However, the interface made it easy to recover from any failed step of the state-machine without ever requiring the operator take manual control of the manipulator.

### **6.4.2 Real-Time Scene Reconstruction and Data Collection at the Costa Rican Pacific Shelf Margin**

Demonstrations at the Pacific continental margin were conducted during a two-week research cruise aboard the R/V *Falkor* using the *SuBastian* ROV. The automated manipulation component of this expedition focused on a demonstration of the vision system and data collection to aid the development of visual methods. The integration time of our system took approximately two days during cruise mobilization, highlighting the relative ease and flexibility with which the system can be implemented on a variety vehicles and manipulators. Camera image data was streamed over a GigE interface at 3 Hz to a topside workstation, which handled all processing and visualization. Joint encoder feedback from the manipulator was obtained by passively monitoring the serial communication between the manipulator and the ship's control computer. We visualized the real-time configuration of the manipulator in the 3D planning environment with the stereo point clouds projected into the scene. The point clouds were generated from the stereo imagery using the standard semi-global matching (SGM) method built into the ROS image processing pipeline, and the parameters were hand-tuned to achieve the best results. Figure 6.10 shows a frame from the real-time visualization captured on the seafloor during one of the dives. A good camera calibration combined with high water clarity, rich seafloor texture, and evenly distributed scene lighting resulted in high quality point clouds. These early results demonstrated the effectiveness of the vision system to capture the 3D structure of the workspace and the ability to fuse the information into a real-time scene representation that is useful for both manipulation planning and 3D visualization of the ROV configuration and planning environment.

During this expedition we collected an extensive dataset [15] of synchronized stereo and wrist mounted fisheye images along with the manipulator joint feedback from a diverse set of seafloor environments (Fig. 6.11). AprilTags mounted on plates were dispersed into the scenes to provide ground truth for the camera poses, and three different types of graspable handle objects were also randomly placed into the scenes. This dataset supported the development of our visual methods



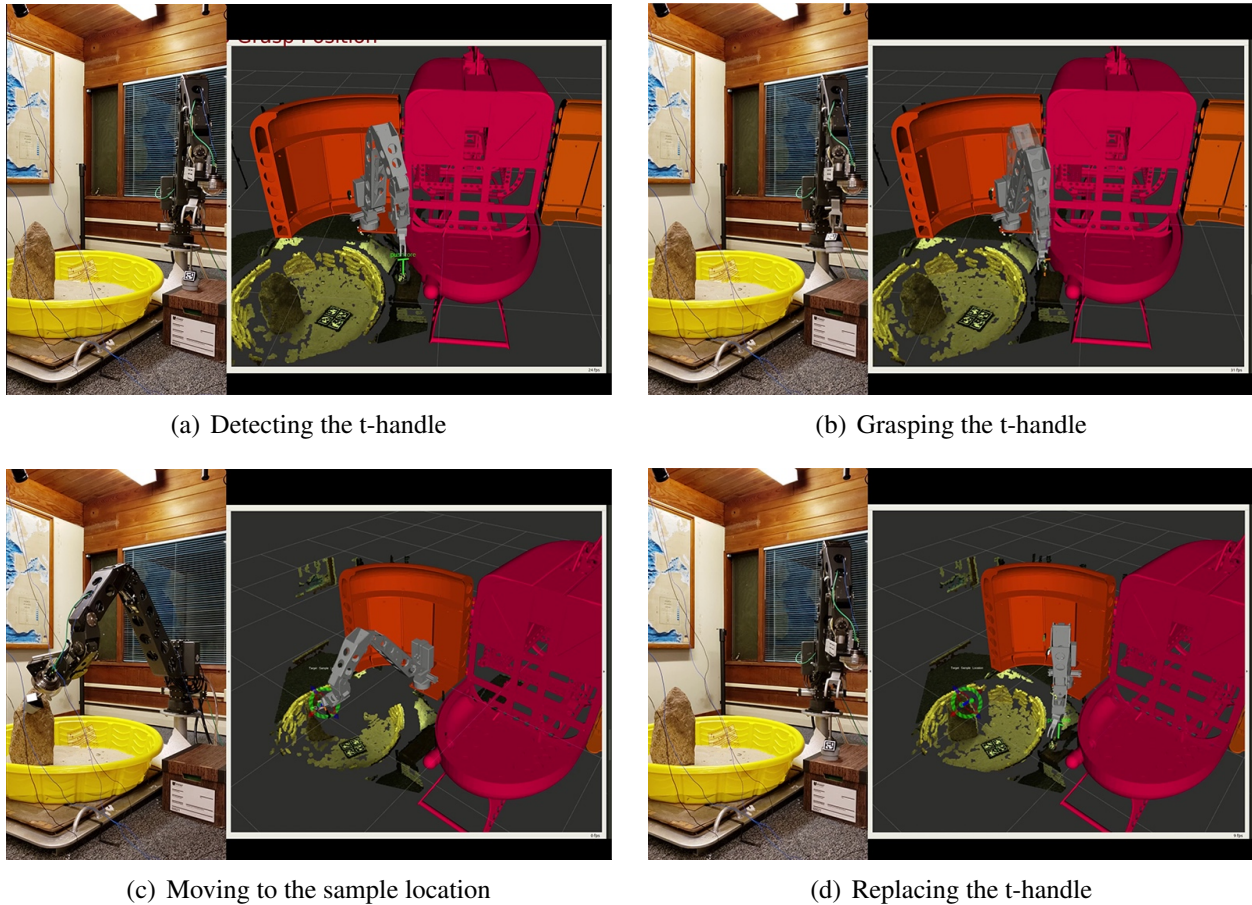


Figure 6.9: We demonstrated fully autonomous pick-and-place with a t-handle on a testbed with the same camera and manipulator hardware used on the *NUI HROV*. First, (a) the t-handle was detected from the fisheye camera using the AprilTags, and the handle pose was projected into the planning scene. Next, (b) the manipulator was commanded to grasp the t-handle via the autonomy interface. Subsequently, (c) a sample location was set in the planning scene with an interactive marker based on the projected stereo point cloud, the manipulator planned a motion to reach the sample location, and executed the plan after the user verified it. The manipulator was then (d) commanded to return the t-handle to the location where it was first grasped. The rock in the environment was placed in a delicate balance on its end, yet the manipulator was controlled with enough precision to bring the tool into direct contact without knocking it over.

and is also intended to serve the underwater research community for the development of scene reconstruction, object detection, and pose estimation methods that work robustly in real seafloor environments.

The fisheye images were processed into a standalone dataset with annotated 2D bounding boxes and 6D poses for the handle objects visible in each frame. This dataset was released with the SilhoNet-Fisheye publication [15]. Figure 6.12 shows sample images from this dataset. The com-

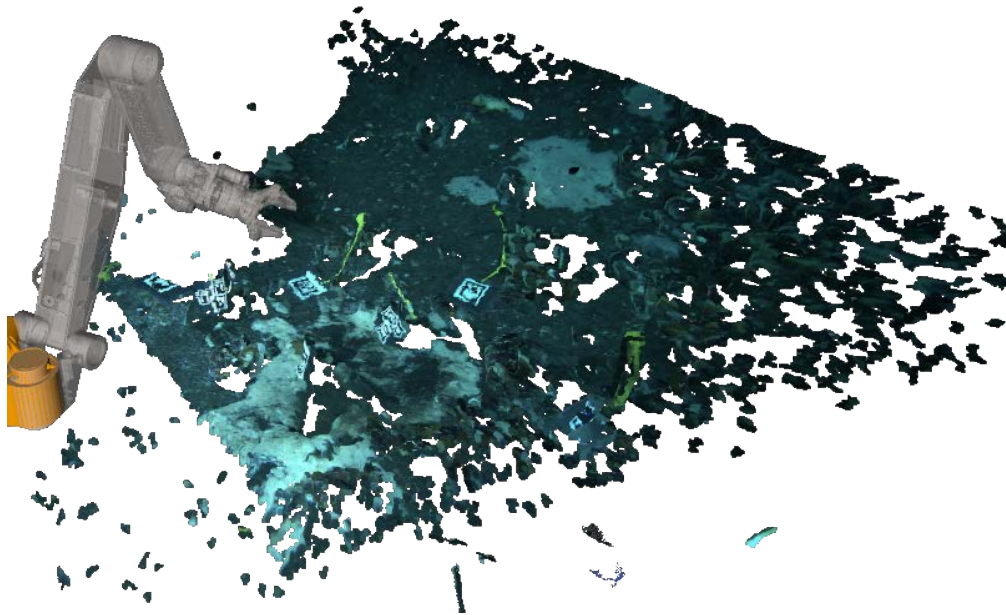


Figure 6.10: The vision system was integrated on the *SuBastian* ROV operated by Schmidt Ocean Institute, where we demonstrated real-time visualization of the planning scene with a Schilling Titan-4 manipulator and the projected stereo point clouds. This also demonstrates the flexibility of the system to be integrated with different vehicles and manipulators.

bined dataset of stereo and fisheye imagery with synchronized manipulator joint feedback supported our development of visual methods for scene reconstruction.

### 6.4.3 Automated Sample Collection and Return within Active Submarine Volcanoes

For exploration of the Kolumbo and Santorini calderas, *NUI*'s manipulator was mounted to the starboard door and the stereo cameras were mounted to the port door (Fig. 6.14). Having the manipulator and stereo camera on opposite articulating doors allowed for flexibility in configuring the position of the arm according to the specific manipulation task and enabled on-the-fly adjustment of the manipulator and stereo positions separately. Unfortunately, the doors are actuated using hydraulic rams that lack position feedback. For safe motion planning, it was necessary to estimate the door positions in real-time. The estimated door positions were used to update the kinematic configuration of the vehicle in the planning scene. However, we observed that the doors could flex, introducing some error in the kinematic estimates that negatively impacted the accuracy of the stereo point cloud projection into the planning scene. To minimize accumulated error in the transform between the stereo camera frame and the manipulator base frame, the stereo frame was



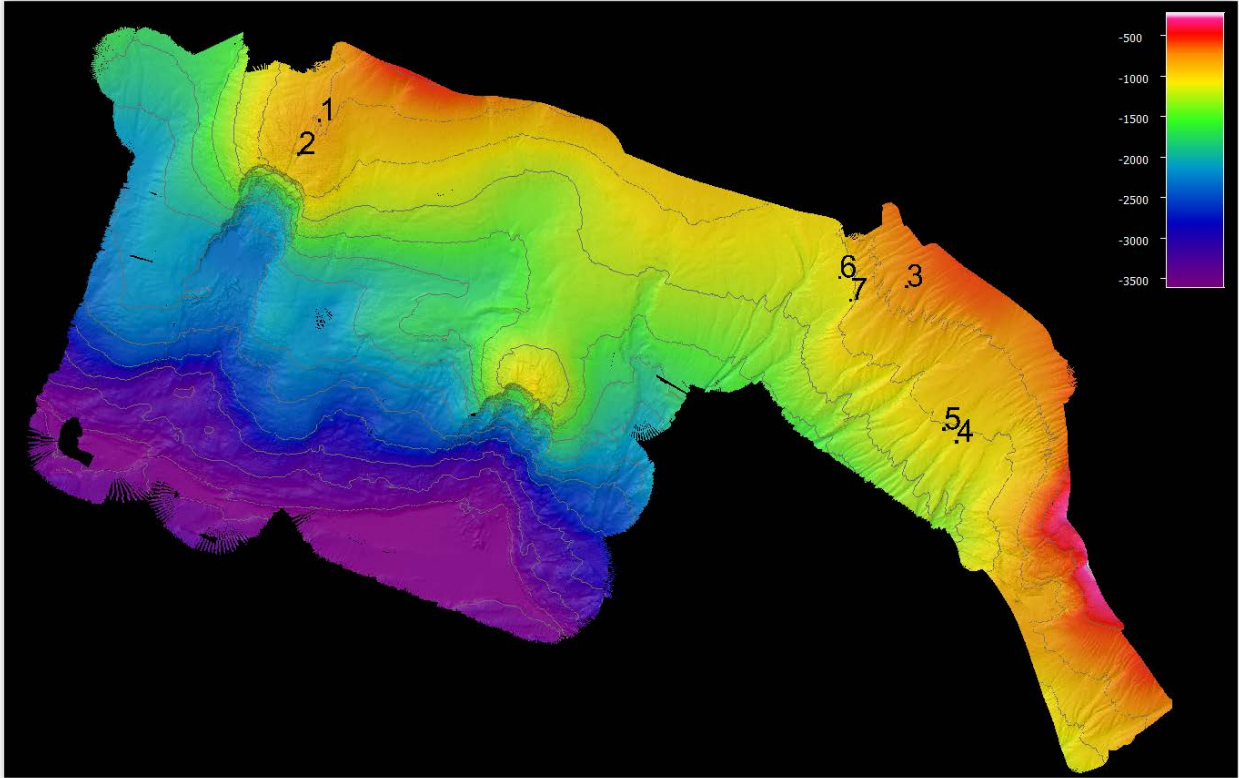
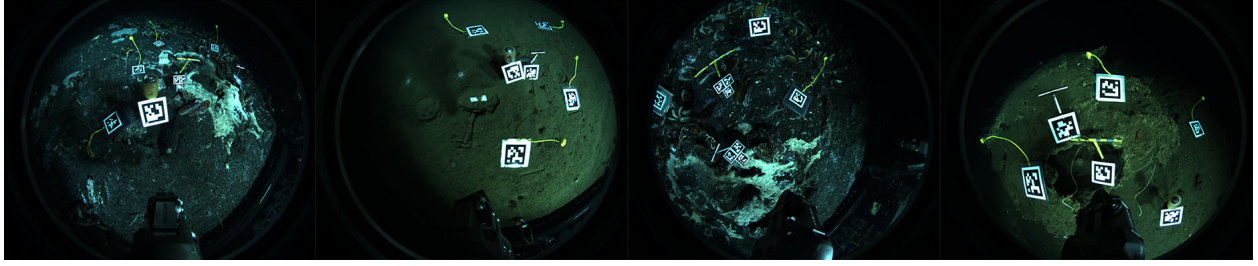


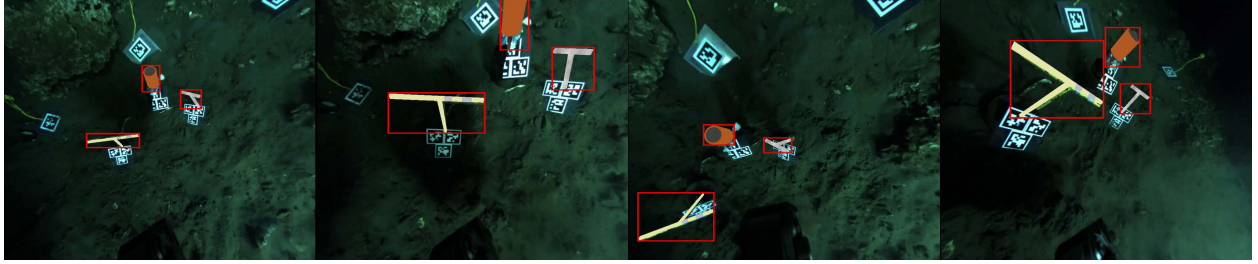
Figure 6.11: Bathymetric map of the survey area from the 2018 cruise on the Pacific continental margin showing data collection locations at seven different science goal sites, spanning over 62 km (linear distance between Locations 1 and 4) and ranging in depth from 600 m to 1100 m. Depth contours are spaced at 250 m intervals and the map is oriented with North up.

referenced directly to the base frame in the ROS transform tree. The base frame was accurately localised directly from the stereo camera through detection of tags fixed to the manipulator base.

To estimate the door positions in real-time, we affixed AprilTags to the starboard door and to the bow of the vehicle’s payload bay (Fig. 6.15 (left)). The tags on the vehicle bow were mounted at a measured location relative to the vehicle reference frame, with the reference tag’s  $Z$ -axis aligned with the  $Z$ -axis of the vehicle reference frame. The door joint axes of rotation were also aligned to the  $Z$ -axis of the vehicle frame, enabling a simple trigonometric calculation of the door angles based on the relative tag locations in the  $X$ - $Y$  plane. We used the left stereo camera to track the relative pose of the AprilTags and used these estimates as observations in AprilTag-based visual SLAM [149] (Fig. 6.15). Figure 6.16 shows a schematic of the vehicle and visual SLAM system with the relevant transforms in the  $X$ - $Y$  plane used to calculate the door angles. The visual SLAM provided the relative translations between the vehicle tag frame and the starboard tag frame,  $T_{vs}$ , and between the vehicle tag frame and the stereo camera frame,  $T_{vp}$ . Given that the translations between the vehicle tag and the door joint frames,  $T_{os}$  and  $T_{op}$ , were measured and known, the



(a) Sample raw fisheye images from different sequences of the dataset



(b) Sample annotations from a single sequence of the dataset

Figure 6.12: The fisheye imagery collected during the Costa Rica cruise was processed into a stand-alone dataset [15]. The images are annotated with the bounding box and six-DoF pose of the tool handles placed in the workspace. The top row (a) shows sample raw fisheye images from different sequences of the dataset, and the bottom row (b) shows sample annotations from a single sequence in the dataset. The images are center rectified here only for purposes of visualization.

angle of the starboard and port doors,  $\theta_s$  and  $\theta_p$  respectively, were recovered as

$$\theta_s = \arctan \frac{T_{s,y}}{T_{s,x}} - \theta_{s_0} \quad (6.1a)$$

$$\theta_p = \arctan \frac{T_{p,y}}{T_{p,x}} - \theta_{p_0}, \quad (6.1b)$$

where

$$T_s = T_{vs} - T_{os} \quad (6.2a)$$

$$T_p = T_{vp} - T_{op}, \quad (6.2b)$$

where the  $x$  and  $y$  subscripts indicate the corresponding component of the translation vector, and  $\theta_{s_0}$  and  $\theta_{p_0}$  are the measured angle offsets.

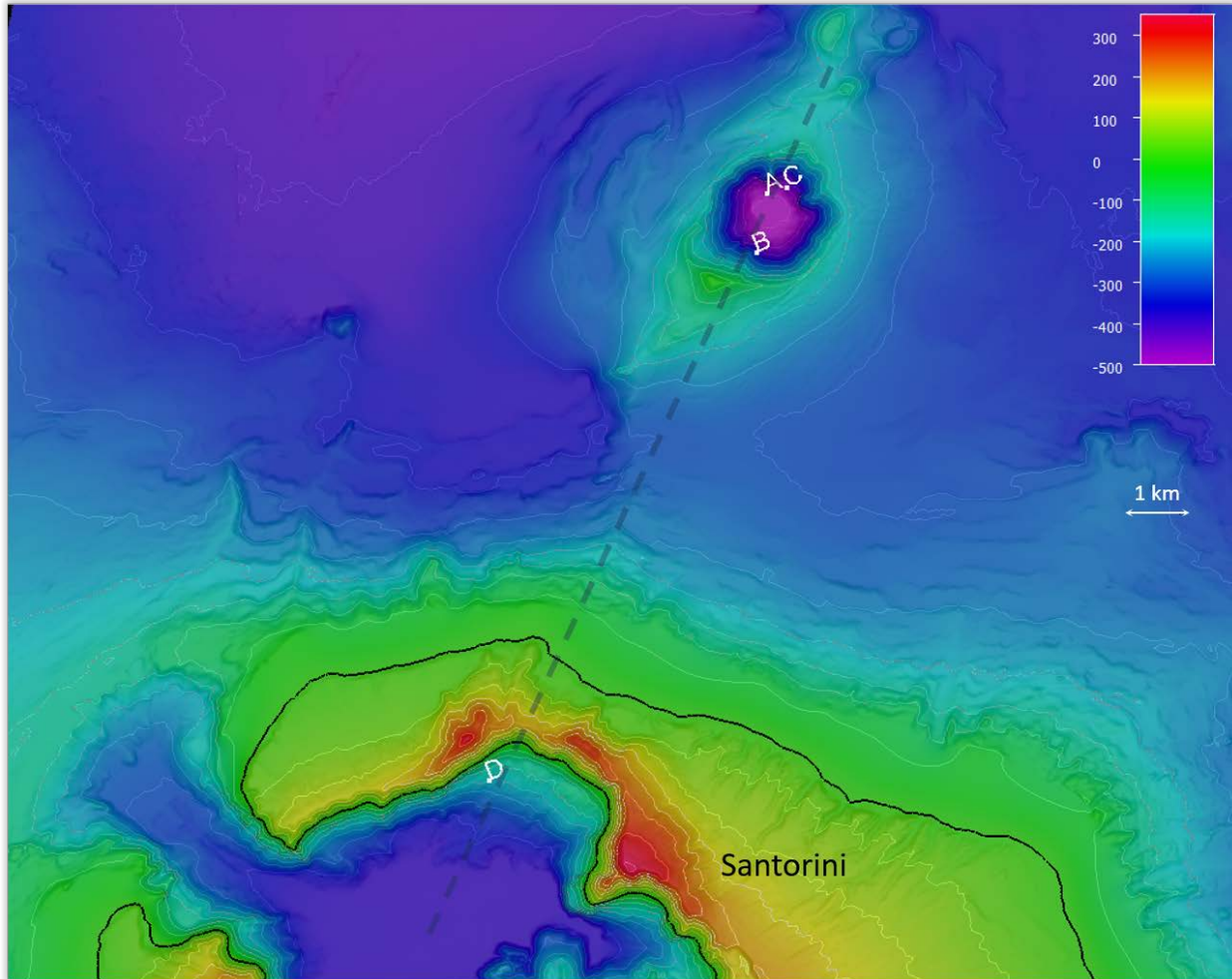


Figure 6.13: Map of automated sample collection locations, with regional bathymetry adapted from [140, 139]. The sea level contour is indicated in black. The dashed line indicates the Christiania-Santorini-Kolumbo tectonic line [140]. Locations marked A, B, and D indicate automated sample collection and return sites, and location C indicates the site where a natural language proof-of-concept demonstration was conducted. Sampling depths ranged from 240 m to 501 m

#### 6.4.3.1 Planner Controlled Biological Sample Collection

The Kolumbo-Santorini expedition resulted in several scientific achievements, including verification of the persistence of *Kalliste Limnes* [27], 3D reconstruction of extremophile habitats within the calderas' craters, and sampling of benthic fluids, seafloor sediments, and biological materials. One of the most useful subsea tools for sample collection and return is a "slurp gun" vacuum sampler. For these operations, the slurp nozzle is in close proximity to the sample of interest and a vacuum pump sucks the sample through the hose into a collection chamber. To demonstrate automated slurp collection, we attached the slurp hose to the side of the manipulator, so that the end-effector could be commanded to the desired location to collect the slurp sample. We completed





Figure 6.14: The *NUI* vehicle is outfitted with clam shell doors that can be closed to reduce drag when cruising and opened to perform manipulation tasks. The manipulator is mounted to the starboard door and the stereo cameras are mounted to the port door.

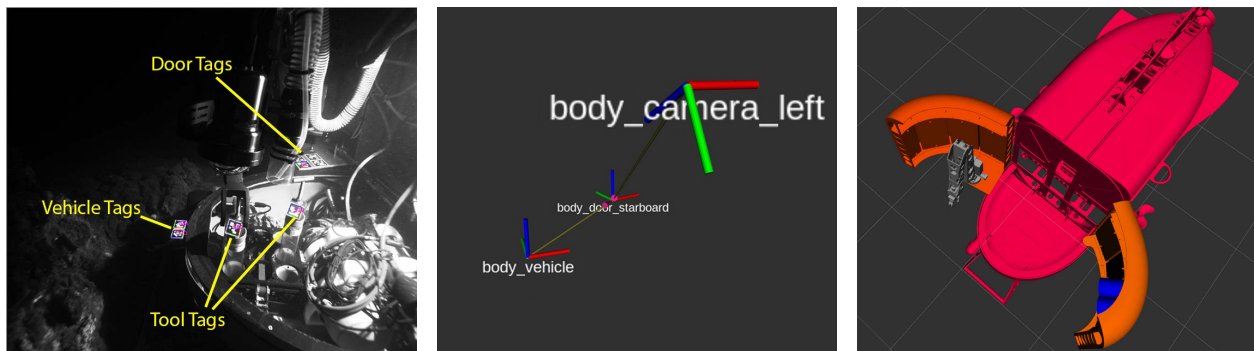


Figure 6.15: Fiducial-based visual SLAM from the left stereo camera was used to estimate the door angles in real-time using (left) tags mounted to the front of the vehicle frame and at the base of the manipulator on the starboard door. SLAM provided estimates of (middle) the relative transformations between the camera and the tag frames that were used (right) to estimate the door angles and update the vehicle model in the planning scene. The left stereo camera was also used in conjunction with the wrist mounted fisheye for (left) fiducial-based localization of tools.

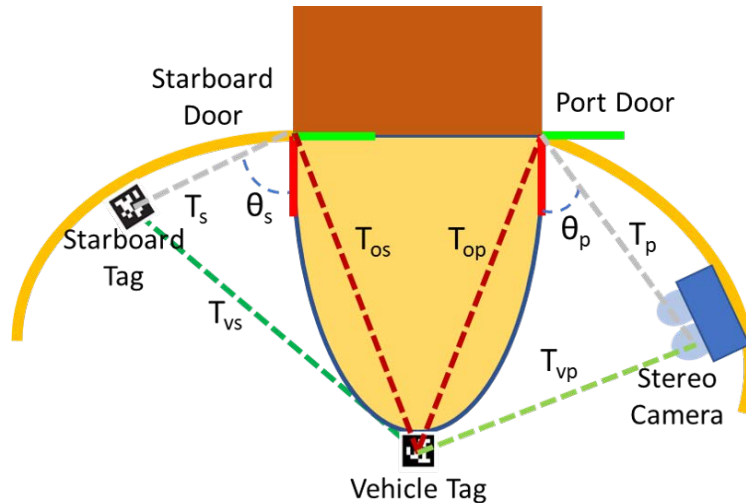


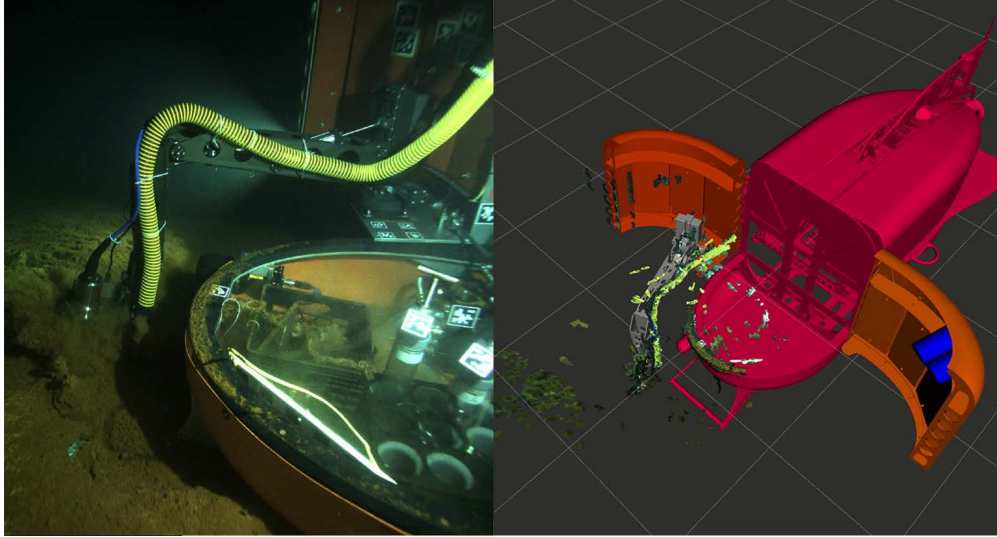
Figure 6.16: A 2D schematic of the *NUI HROV* that relates the visual SLAM from the left stereo camera to the door positions. The green dashed lines represent transformations estimated from SLAM. The red dashed lines denote known transformations computed from the vehicle kinematic model and the measured position of the tags. The grey dashed lines represent the calculated transforms with respect to each door reference frame, which have a trigonometric relation to the door angles,  $\theta_s$  and  $\theta_p$ .

multiple successful sample collections, including that shown in Figure 6.17, where a slurp sample of a sediment microbial mat was collected using the planner interface to command the manipulator to the desired sample location, after which the manipulator was returned to the home position.

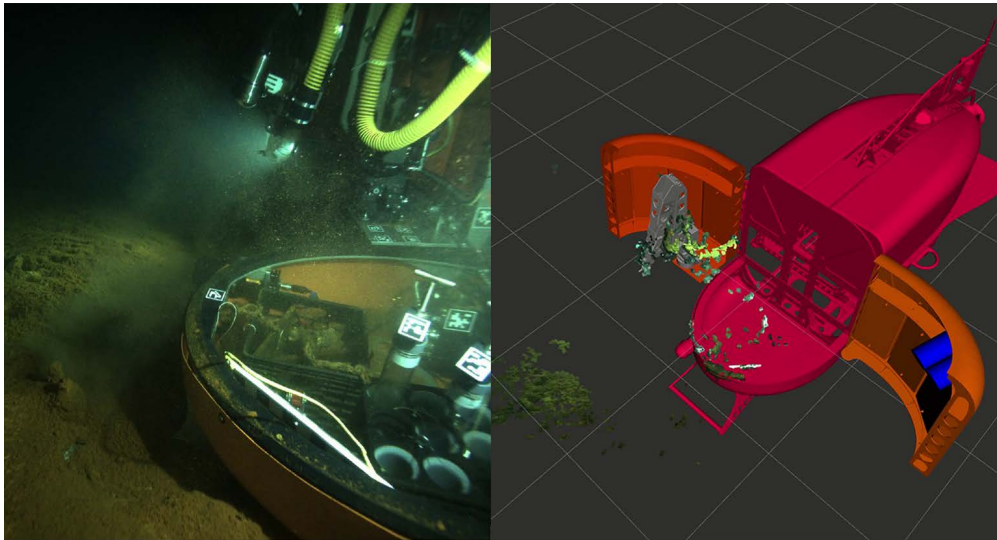
### 6.4.3.2 Natural Language Control

Subsea ROV missions require close collaboration between the ROV pilots and scientists. The primary means by which pilots and scientists communicate is through spoken language—scientists use natural language to convey specific mission objectives to ROV pilots (e.g., requesting that a sample be taken from a particular location), while the pilots engage in dialogue to coordinate their efforts. Natural language provides a flexible, efficient, and intuitive means for people to interact with our automated manipulation framework. The inclusion of a natural language interface would support our goal to realize a framework that can be integrated seamlessly with standard ROV operating practices and may eventually mitigate the need for a second pilot.

Using the *NUI HROV*, we performed a proof-of-concept demonstration of an architecture that allows user control of an ROV manipulator using natural language provided as text or speech using a cloud-based speech recognizer. We frame natural language understanding as a symbol grounding problem [69], whereby the objective is to map words in the utterance to their corresponding referents in a symbolic representation of the robot’s state and action spaces. Consistent with contem-



(a) Taking slurp sample



(b) Returning to home position

Figure 6.17: An example of a successful planner controlled slurp collection of a bacterial mat, with the yellow slurp hose attached to the manipulator. The manipulator was (a) commanded to the desired slurp location through the automated planning interface and then (b) directed to return to its home position following the slurp collection.

porary approaches to language understanding, we formulate grounding as probabilistic inference over a learned distribution that models this mapping. In particular, given the syntactic parse of a natural language command  $\Lambda$ , we employ maximum a posteriori inference over the power set of referent symbols  $\mathcal{P}(\Gamma)$

$$\Gamma^* = \arg \max_{\mathcal{P}} p(\Gamma | \Lambda, S). \quad (6.3)$$



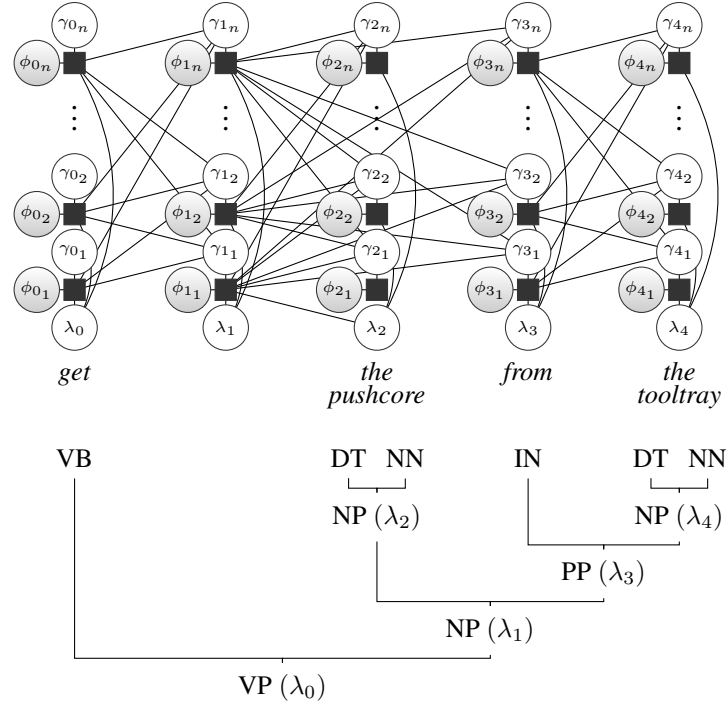
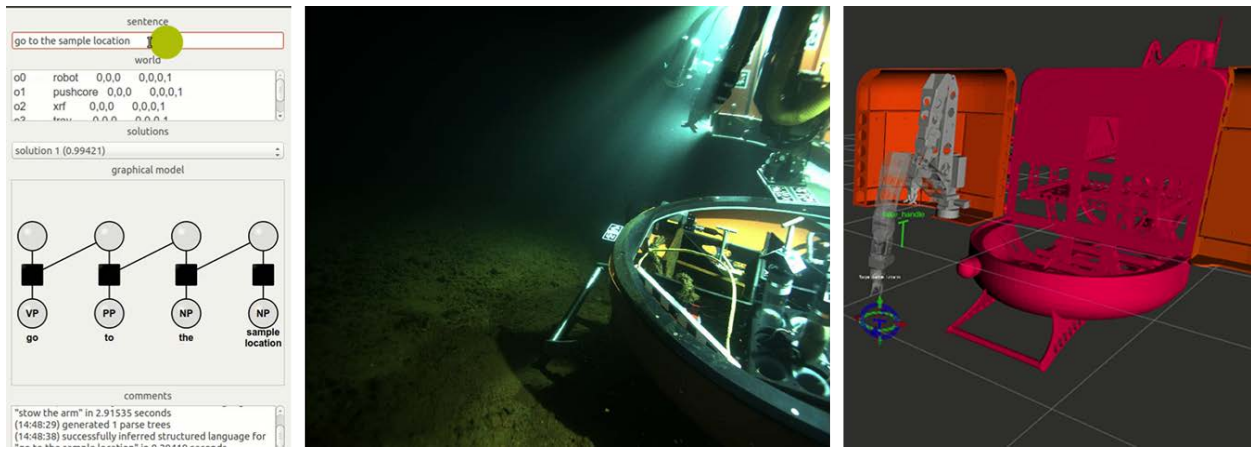


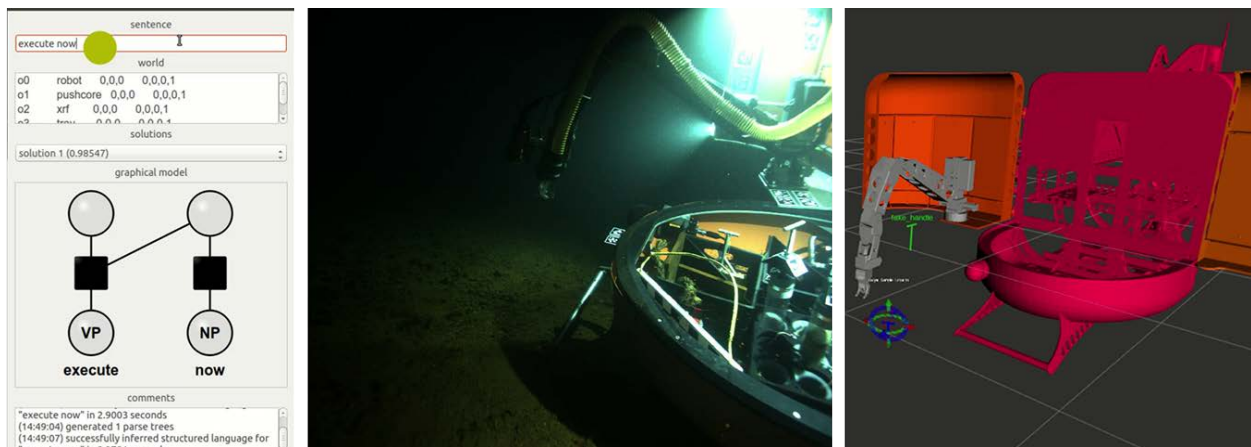
Figure 6.18: A visualization of (top) the DCG factor graph for the expression “get the pushcore from the tooltray” aligned with (bottom) the associated parse tree. Shaded nodes denote observed random variables, while those rendered in white are latent.

where  $S$  is a variable that denotes the robot’s model of the environment (e.g., the type and location of different tools). We model this distribution using the Distributed Correspondence Graph (DCG) [78], a factor graph (Fig. 6.18) that approximates the conditional probabilities of a Boolean correspondence variable  $\phi_{i_j}$  that indicates the association between a specific symbol  $\gamma_{i_j} \in \Gamma$ , which may correspond to an object, action, or location, and each word  $\lambda_i \in \Lambda$ . Critically, the composition of the DCG factor graph follows the hierarchical structure of language. The model is trained on corpora of annotated examples (i.e., words from natural language utterances paired with their corresponding groundings), whereby we independently learn the conditional probabilities for the different language elements, such as nouns (e.g., “the pushcore”, “tool”, and “tool tray”), verbs (e.g., “retrieve”, “release”, and “stow”), and prepositions (e.g., “inside” and “towards”). Together with the fact that the factor graph exploits the compositional nature of language, the DCG model is able to generalize beyond the specific utterances present in the training data.

For our initial implementation, the space of symbols  $\Gamma$  included the tools that the arm was able to grasp and the different steps that comprised the state machine underlying the pick-and-place pipeline. Figure 6.19 presents an example from a deployment at the Kolumbo caldera in which natural language was used to initiate path planning to the sample location and then to command the manipulator to execute the planned path. Several tests were conducted in which the manipula-



(a) Language commands the system to plan a path to the sample location



(b) Language command to execute planned path

Figure 6.19: Demonstration of a proof-of-concept framework that enabled operators to interact with our autonomous manipulation architecture using natural language. Given input in the form of free-form text, either entered by the operator or output by a cloud-based speech recognizer, we (left) infer the meaning of the command using a probabilistic language model. (a) In the case of the command to “go to the sample location”, our system (top-right) determines the goal configuration and solves for a collision-free path in configuration space. (b) Given the command to “execute now”, the manipulator then (bottom-right) executes the planned path to the goal.

tor was commanded through natural language input to move to a location specified by the sample marker in the planning interface and then return to the home position. These tests demonstrated the flexibility of our system to incorporate different operational modalities through high level abstraction.



#### 6.4.4 Performance Analysis

We evaluated the overall accuracy of the calibrated kinematic and visual system on the testbed. For this evaluation, we placed an AprilTag grid in the scene and activated every joint of the manipulator while keeping the tag grid in view of the fisheye camera. We used TagSLAM [149] to generate a visual SLAM estimated trajectory of the fisheye camera, and we used the manipulator joint feedback to also generate a kinematic based trajectory. These trajectories are plotted against each other in figure 6.20. The overall mean error between the kinematic and visual based trajectories is 1.16cm, the maximum trajectory error is 3.27cm, and the standard deviation is 0.65cm. These results are a conservative estimate of the system calibration accuracy as there are several sources of error: the visual SLAM accuracy degrades when the tags are near the edge of the fisheye image; the agreement between the kinematic and visual based pose of the fisheye camera depends on the accuracy of the hand-eye calibration; the joint feedback and fisheye images are not synchronized; and the SLAM and kinematic reference frames were mapped to each other through a single fisheye frame estimate of the tag grid pose, projected from the fisheye frame through the kinematic chain to the manipulator base frame. However, we have demonstrated in our experimental trials that the system accuracy is good enough to perform high level automation tasks.

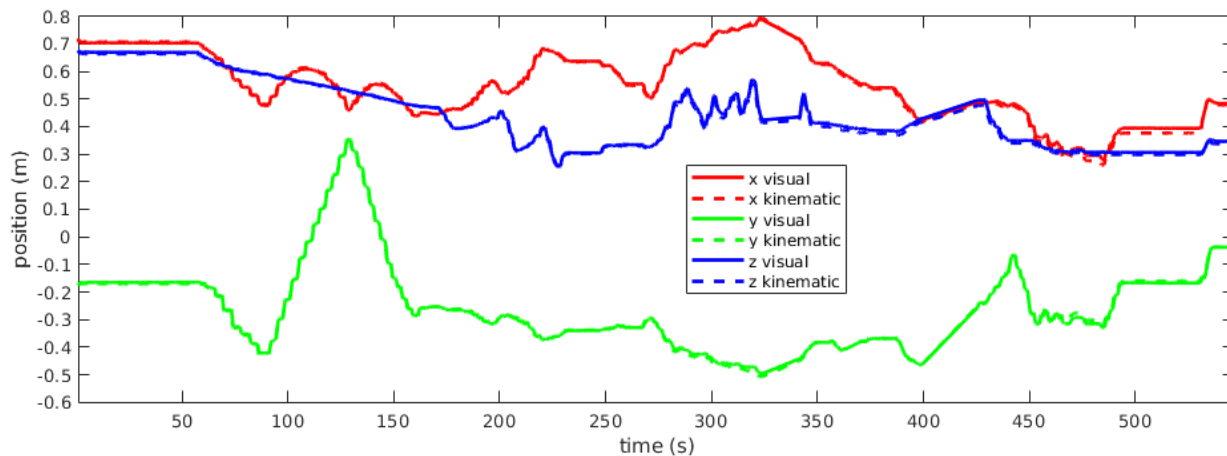


Figure 6.20: Plot of the TagSLAM estimated trajectory (visual) of the fisheye camera versus the trajectory estimated from the manipulator joint feedback (kinematic). The trajectory is plotted separately for each coordinate axis with respect to the manipulator base frame.

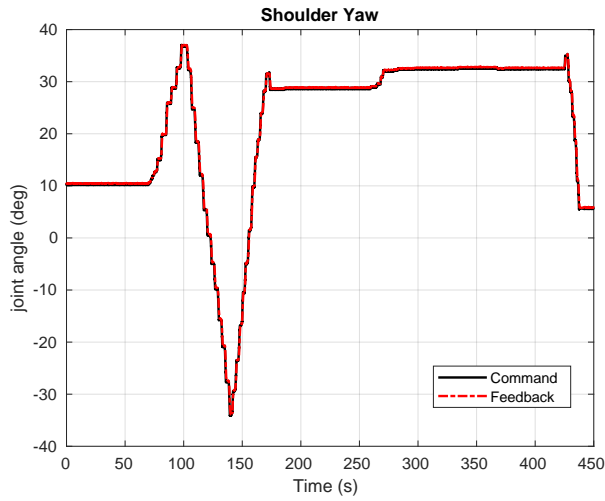
The first five joints of the KRAFT manipulator are controlled through joint position set-point commands. We analyzed the manipulator control response for bias or hysteresis, as these are known issues with hydraulic actuation. Figure 6.20 shows plots of the commanded versus feedback positions during actuation of each joint of the testbed manipulator. The figure also shows a histogram of errors, binned at  $0.5^\circ$ , between the commanded and followed joint trajectories. All

of the joints except the wrist pitch exhibit small bias and no major hysteresis is evident. The wrist pitch exhibits a bias of approximately  $1.5^\circ$ , which is significant, but did not prevent completion of high level automation tasks. Figure 6.20 shows the same plots for the *NUI HROV* manipulator made from data recorded during the field trials in Greece. The elbow and wrist joints exhibit little bias or hysteresis. However, both of the shoulder joints exhibited high bias, particularly the shoulder yaw joint, which had a bias of approximately  $8^\circ$ . This high bias prevented the completion of a pick-and-place manipulation task during the field trial. Our current control system relies on the manipulator valve controller to move to the desired set-point and does not account for bias in the control response. It will be critical in the future to incorporate an adaptive controller into the system that can account for bias and hysteresis in the hydraulic actuators.

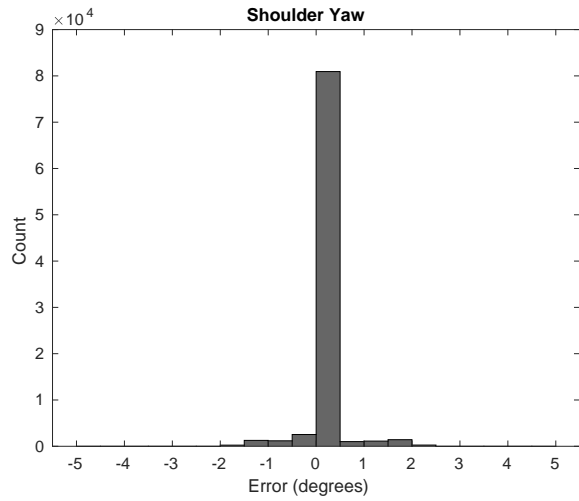
## 6.5 Discussion and Future Work

During the course of our field trials, we identified operational challenges and failure modes for both the manipulator and vision systems. Addressing these issues is necessary to improve the robustness of the system and is the objective of ongoing research.

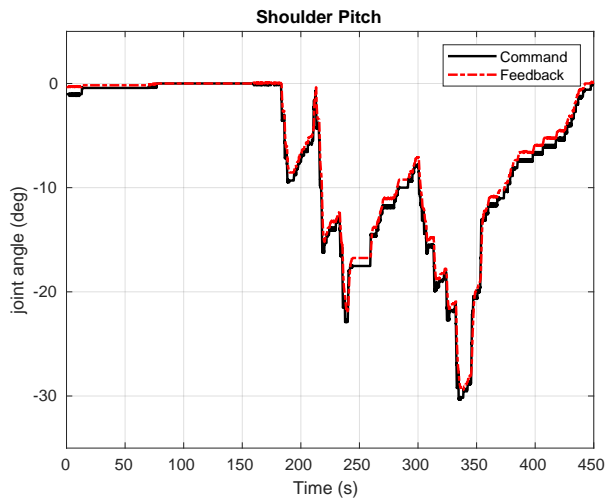
Underwater hydraulic manipulators have inherent characteristics which make them especially challenging to automate and can lead to mission failure if they are not accommodated by the planning and control systems. We identify three particular challenges. The first challenge is the sensors that provide joint position feedback (e.g. potentiometers or encoders) can be noisy and prone to drift, resulting in an inaccurate estimate of the manipulator configuration, which can lead to self-collisions or collision with the vehicle or obstacles in the environment. This issue could be mitigated by continuously calibrating the arm using the vision system to detect and compensate for proprioceptive sensor drift. Such a fully automated kinematic calibration procedure is also a practical necessity for a system to be deployed on a space flight mission and would improve calibration accuracy over the manual procedure used in this report. Our ongoing work seeks to apply a feature-based mapping/structure-from-motion framework that jointly performs scene reconstruction and kinematic calibration of the manipulator using features from the fisheye camera. The second challenge is that hydraulic actuators can be imprecise. Typical hydraulic actuator characteristics include a bias between the commanded and reached joint positions, which we observed in the *KRAFT* manipulator, and hysteresis, where the offset between commanded and reached positions is variable with the direction of joint actuation and the position of the joint. These actuator effects could be mitigated through an adaptive control strategy that adjusts the joint commands to account for detected anomalies or offsets between the commanded and reached configurations. [170] reported hysteresis as high as  $1.5^\circ$  in a Schilling Titan 2 manipulator and subsequently learned joint command offsets in a calibration procedure to compensate for it. The third challenge is that com-



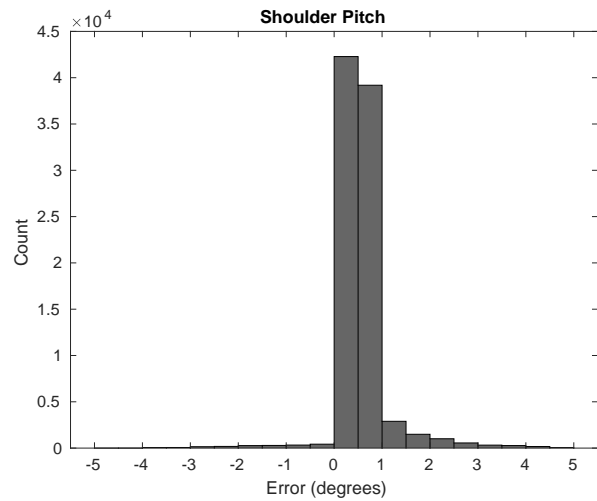
(a) Shoulder Yaw Trajectory



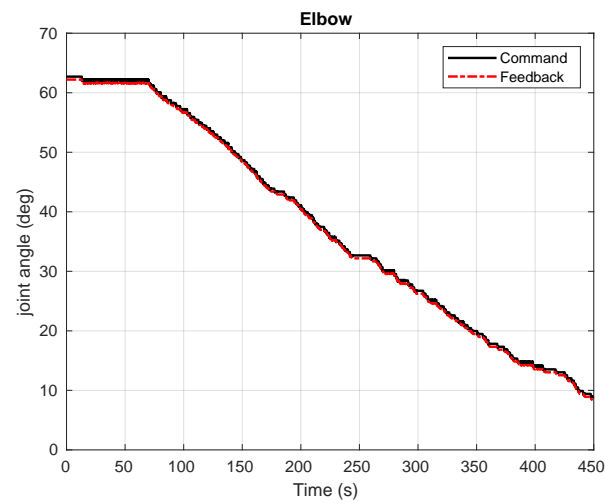
(b) Shoulder Yaw Error Histogram



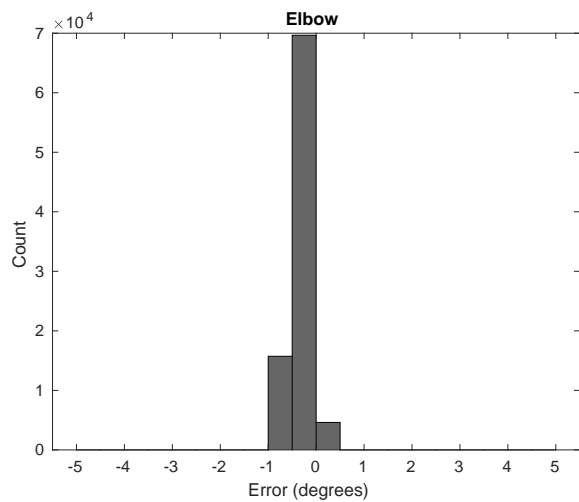
(c) Shoulder Pitch Trajectory



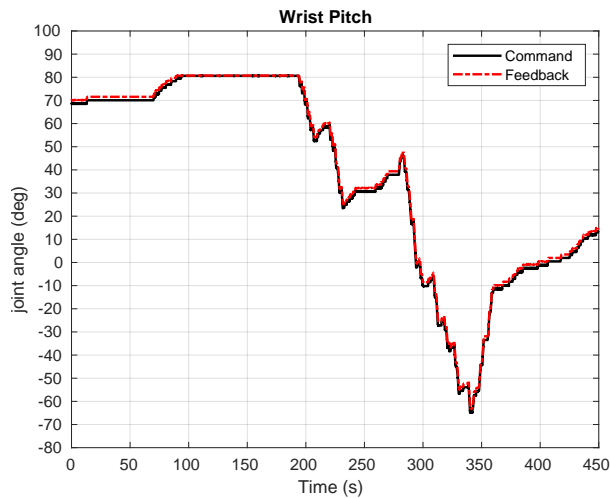
(d) Shoulder Pitch Error Histogram



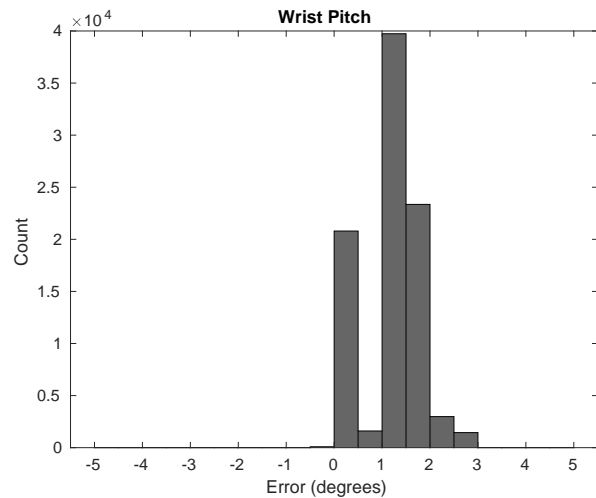
(e) Elbow Trajectory



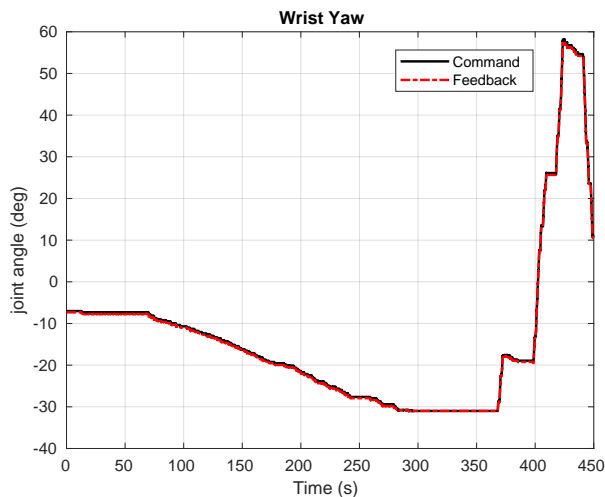
(f) Elbow Error Histogram



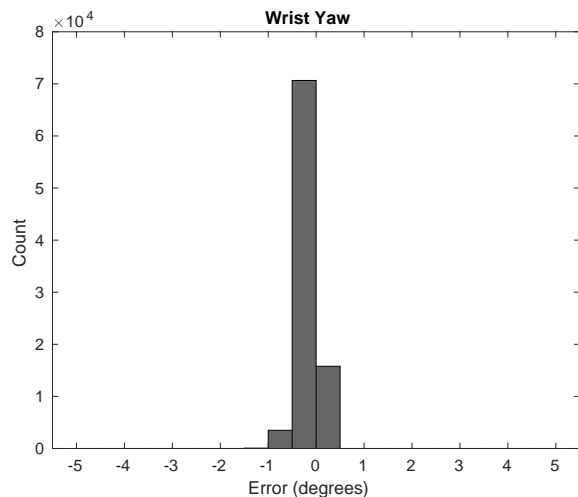
(g) Wrist Pitch Trajectory



(h) Wrist Pitch Error Histogram



(i) Wrist Yaw Trajectory

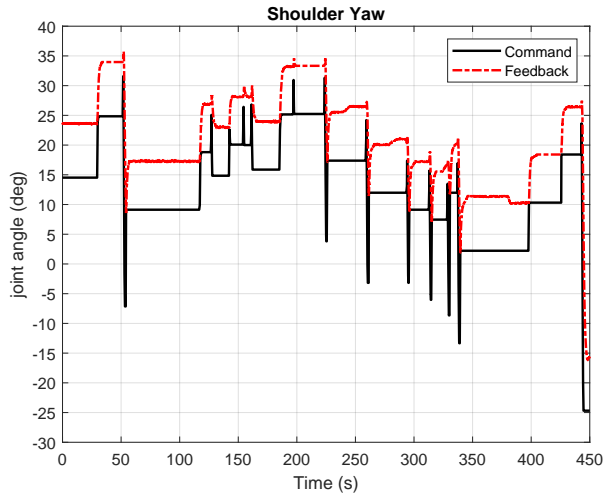


(j) Wrist Yaw Error Histogram

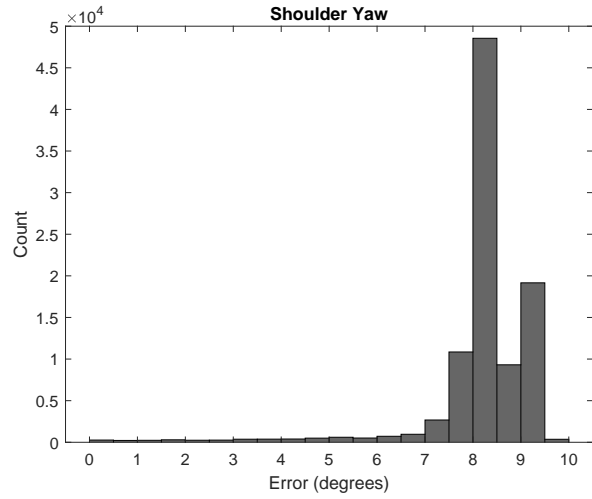
Figure 6.20: Plot of commanded versus followed joint trajectories for the *testbed* manipulator.

plete joint failure is common for underwater manipulators, reducing the degrees-of-freedom by at least one. Mitigation of this failure would require planning level adaptation to determine what manipulation tasks are still feasible. In this under-actuated operational state, the vehicle mobility might be considered within the kinematic planning to compensate for the loss of manipulator dexterity, drawing from the prior work on free-floating intervention.

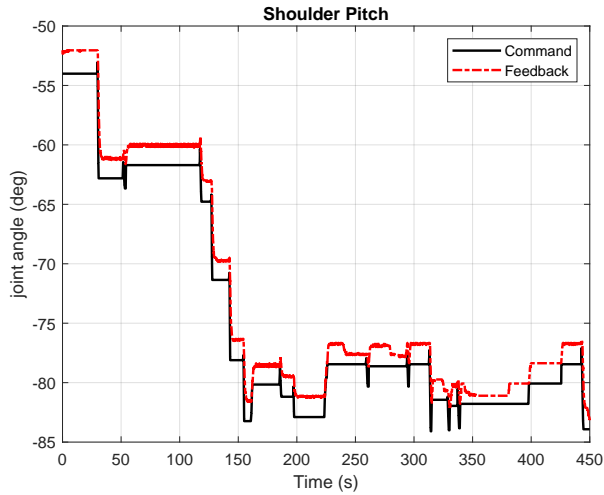
Existing visual reconstruction methods are typically sensitive to lighting, image contrast, and the presence of texture, all of which are highly variable in underwater environments. Figure 6.21 compares point clouds generated using a standard SGM method from the same stereo camera under two different visual conditions. Under near-ideal conditions that include clear water, uniform



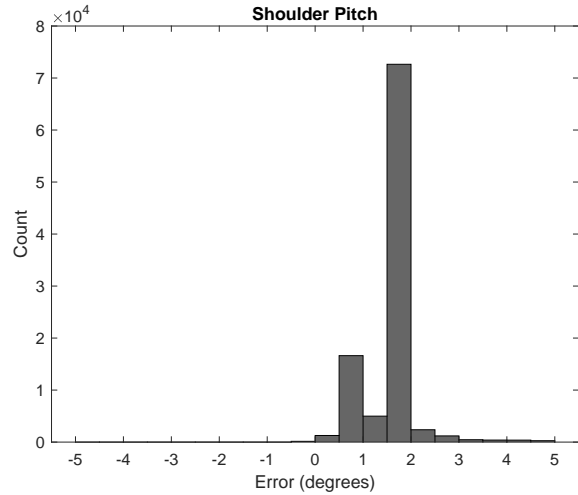
(k) Shoulder Yaw Trajectory



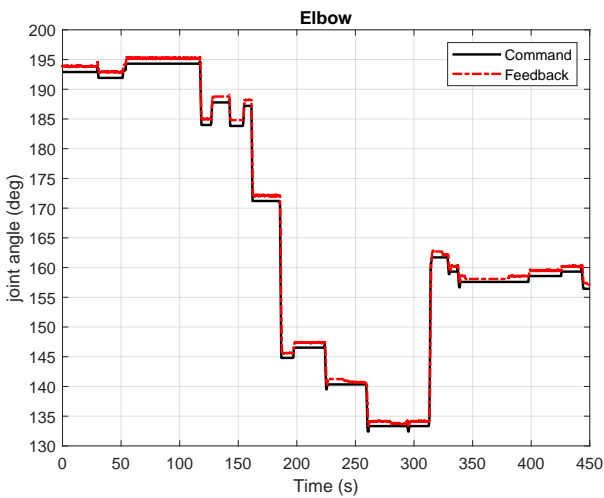
(l) Shoulder Yaw Error Histogram



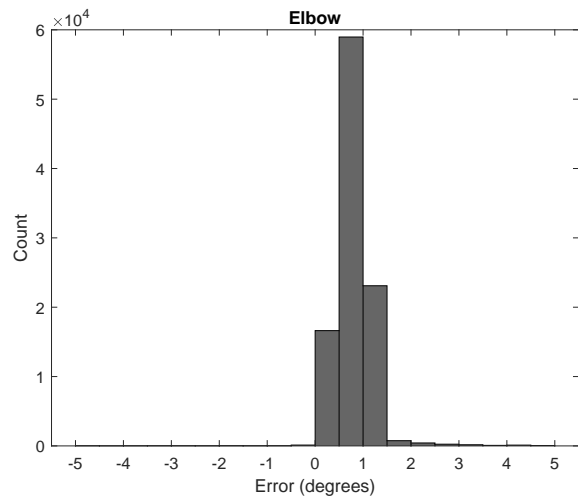
(m) Shoulder Pitch Trajectory



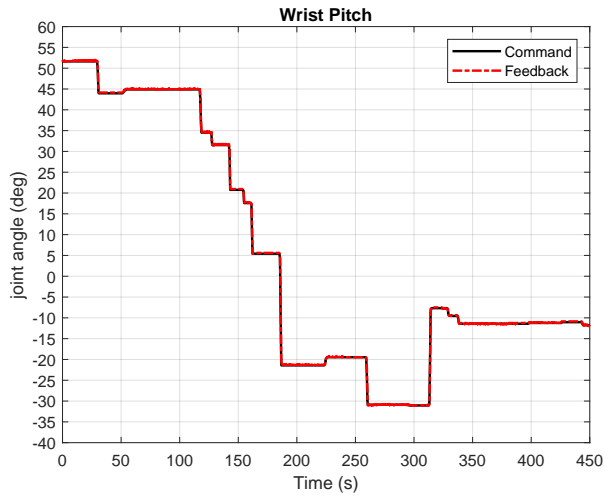
(n) Shoulder Pitch Error Histogram



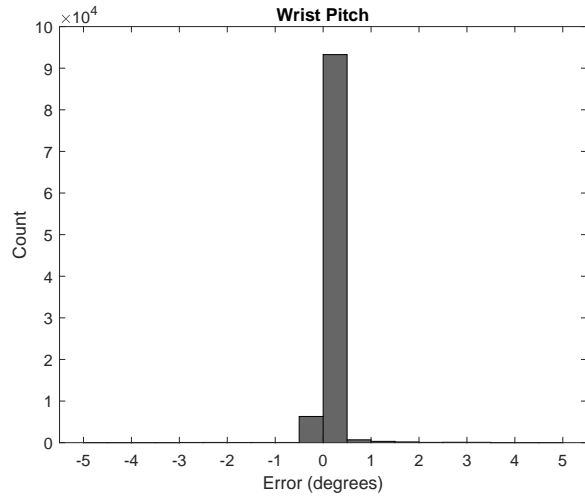
(o) Elbow Trajectory



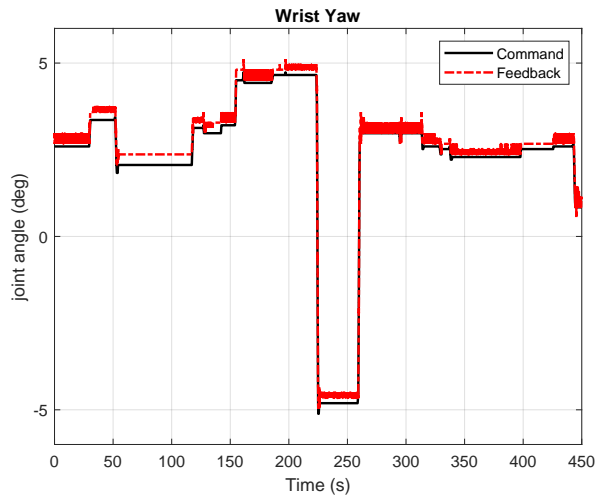
(p) Elbow Error Histogram



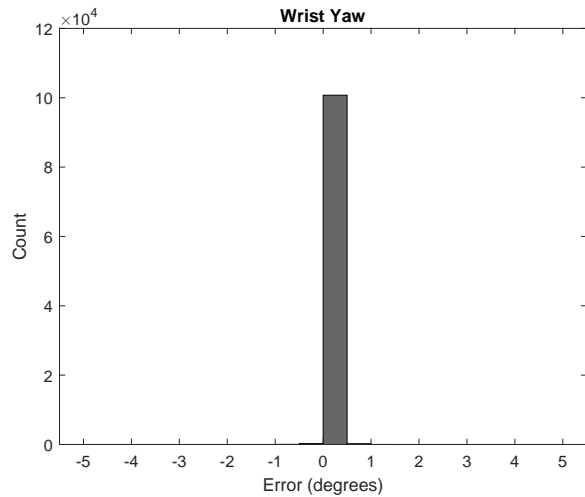
(q) Wrist Pitch Trajectory



(r) Wrist Pitch Error Histogram



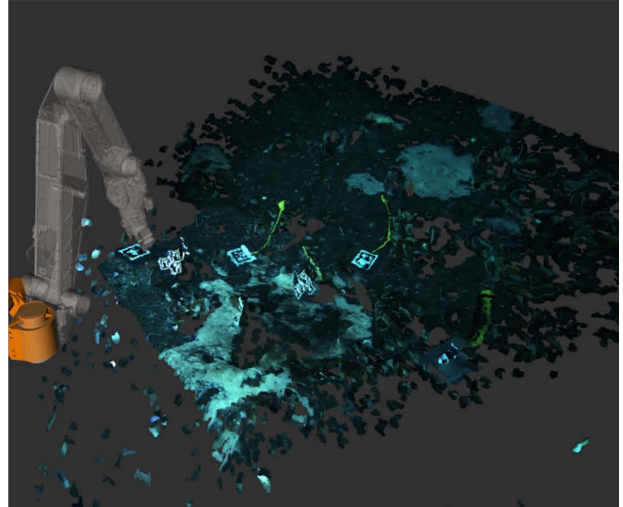
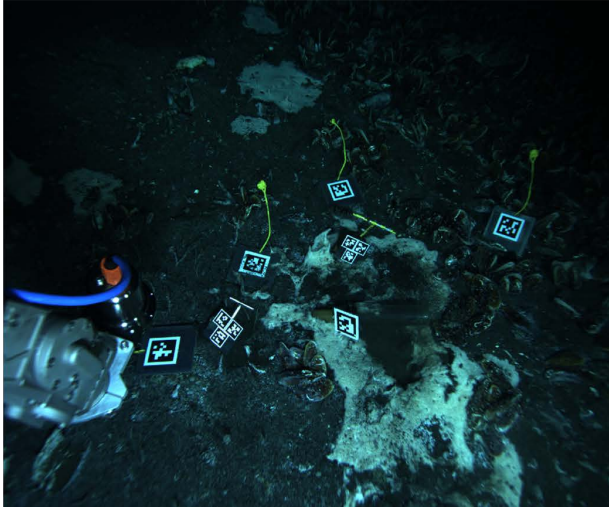
(s) Wrist Yaw Trajectory



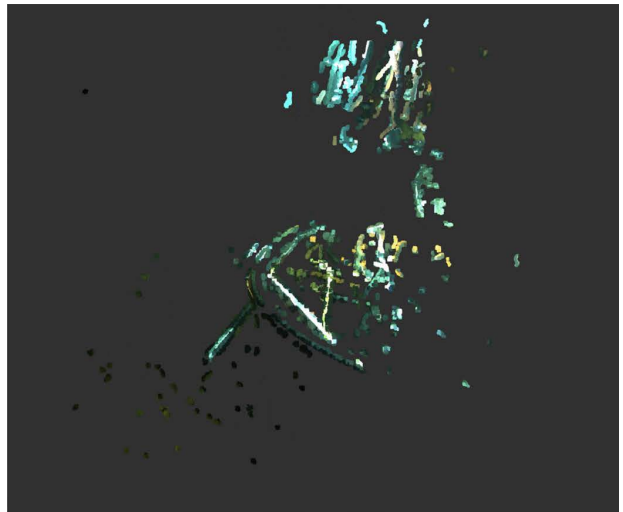
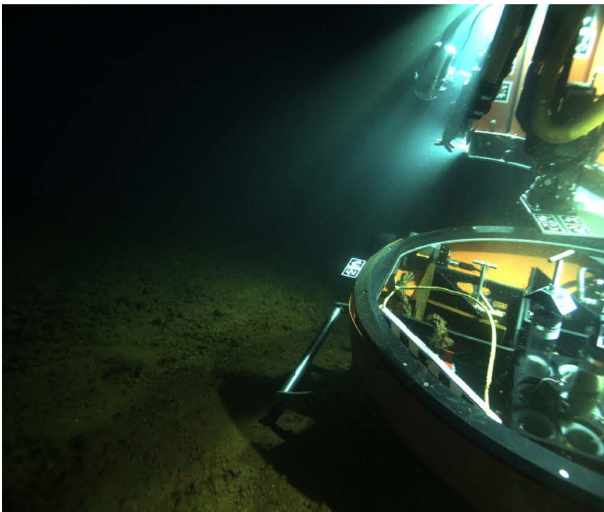
(t) Wrist Yaw Error Histogram

Figure 6.20: Plot of commanded versus followed joint trajectories for the *NUI* HROV manipulator during the Greece field trials.

illumination, and a richly textured seafloor as was the case during our Costa Rica expedition, the point cloud is highly detailed and exhibits a low amount of noise, resulting in a reconstruction that captures fine details of the scene. During the Kolumbo caldera operations, however, fine-grain unconsolidated sediments and amorphous microbial mats blanketed the seafloor, providing little texture for stereo matching. The illumination was uneven and particulates suspended in the water column caused turbidity and light scattering effects that degraded the quality of the images. Under these conditions, stereo matching is only able to recover the well defined edges of the vehicle, while very little of the seafloor is reconstructed. While these examples represent different extremes



(u) Good image quality and dense point cloud



(v) Poor image quality and sparse point cloud

Figure 6.21: The quality of stereo reconstruction is highly dependent on underwater conditions. Here, we compare stereo point clouds generated using the same camera system and stereo matching method, but with images captured within very different seafloor environments. The left images show the view from the left stereo camera, and the right images show the generated point clouds using a SGM-based stereo method. (u) The top row was captured in the clear waters off Costa Rica, with even scene lighting and highly textured seafloor. (v) The bottom row was captured in the Kolumbo caldera, with high backscatter and low texture microbial mats on the seafloor.

in underwater visual conditions, it is critical to develop scene reconstruction algorithms that can operate reliably across this range of conditions to achieve robust autonomy. We are currently investigating information-theoretic ways to exploit our ability to control the pose of the wrist-mounted fisheye camera and an adjacent light source to acquire targeted views and actively illuminate the



scene in order to improve and extend reconstruction under both good and degraded visual conditions. Our system implementation currently assumes that the scene is semi-static, i.e., that the ROV position and scene state remain constant during the execution of a manipulator command. For example, if a command is given to grasp a detected tool, the pose of the tool in the scene is assumed to remain fixed during the execution of the grasp. If the tool were to move due to some disturbance before the grasp was completed, the grasp action would likely fail. Future work may integrate an obstacle-aware visual servoing controller to complete grasps or perform precise tool placement, which would reject disturbances to either the scene or the manipulator during task execution. Because our system relies on visual sensing, any disturbance to the scene that results in degraded water clarity, such as stirring up bottom sediment, can necessitate waiting for the water column to clear before the manipulation task can continue. While the KRAFT manipulator used in our field trials is particularly low power when idle, minimizing the energy cost of waiting for visual conditions to improve, future research may improve robustness of the system to degraded visual conditions by fusing acoustic imaging sonar data into the scene mapping framework. Compared to visual sensors, acoustic signals are not dependent on lighting conditions and are not degraded by haze in the water column or a sparsely textured seafloor.

The technology presented in this report can be directly integrated onto terrestrial-based underwater manipulation platforms in order to decrease operational risk, reduce system complexity, and increase overall efficiency. The current standard for ROV manipulation requires one or more pilots to operate the UVMS based on image feeds from an array of cameras on the vehicle that are displayed on a set of monitors in a ship-side control van. Existing systems do not provide pilots with an estimate of the 3D scene structure, putting the system at risk of collision between the arm and the vehicle or workspace objects. This, together with the cognitive load imposed by having to interpret multiple sensor streams makes it extremely challenging for pilots to establish and maintain situational awareness. The technology presented in this report can be integrated at three different levels with existing ROV systems. At the first and most basic level, the system can act as a decision support tool that provides a detailed real-time 3D visualization of the scene, including the vehicle and manipulator configuration and a reconstruction of the workspace, enabling a pilot to position the manipulator with greater accuracy, speed, and safety. At the second level, the system can be integrated into the manipulator control system for execution monitoring to limit the motion of the manipulator based on scene structure, preventing the pilot from moving the manipulator into collision or a risky configuration [168]. At the third and highest level, manipulation tasks may be fully automated so that a pilot simply selects a desired function or indicates an intent through some mode of communication such as natural language, whereupon the system plans and executes the task while providing visual feedback to the pilot. In this case, it is critical that the pilot be able to override the automated process and take over control of the arm at will.

Table 6.2: Comparison of the bandwidth requirements for direct teleoperation (top two rows) of an ROV manipulator system compared to operating our high-level autonomy system (bottom two rows), running onboard the vehicle with communication through natural language commands and only the necessary scene state feedback to inform the high-level commands.

Mode	Data Type	Bandwidth
Teleoperation Cameras	Compressed SD or HD @ 10–30 Hz	100 KB/s–3 MB/s
Teleoperation Manipulator Coms	2 way $\times 15 - 200 \text{ Hz} \times 18 \text{ B}$	540 B/s–7.2 KB/s
Natural Language	1 B/letter $\times \sim 7 \text{ letters/word} \times \sim 2.5 \text{ words/s}$	17.5 B/s
Scene State Feedback	State and Compressed Images @ 0.1–1 Hz	3–30 KB/s

For teleoperation of ROV manipulators, it is standard practice to stream multiple high-definition (HD) camera feeds at 30 Hz to the operating pilots. In the most bandwidth constrained circumstances, Compressed standard-definition (SD) cameras can be streamed at 10 Hz to the pilots. At lower image resolutions or framerates, it becomes difficult for pilots to teleoperate the manipulator safely. Our system enables high-level command of the manipulator and mitigates the need for continuous image streams back to the controlling pilot. Single image frames need only be sent when a scene change is detected or on request. Future work on the vision system will develop methods for semantic-level scene understanding, which will further reduce the need for direct image streams back to the pilot. For a semantic aware system, natural language is well suited for human-machine interaction and can drastically reduce the data communication load between the vehicle platform and a remote operator by on-boarding data heavy computation (e.g., image processing) onto the vehicle’s local compute system and interfacing with the remote operator through small bandwidth language packets. For our system to operate with pilot oversight, high level commands and sensory feedback need only be streamed at rates which match the dynamics of the scene. In the scenario where the vehicle is set down on the seafloor to collect samples, the relevant scene dynamics can be on the order of seconds, minutes or longer, enabling significant reduction of the communication bandwidth which is vital for remote operations over bandwidth limited connections, such as satellite links. Table 6.2 shows estimated bandwidth range requirements for the manipulator coms and image streams necessary to support direct teleoperation of an ROV manipulator system compared to the bandwidth requirements for natural language communication with the vehicle and only the necessary scene state feedback to inform the high level commands. In the case of direct teleoperation, the manipulator coms can range from 15 Hz to 200 Hz two-way communication with a typical packet size of 18 B. We estimate the image bandwidth for a single SD or HD camera with compressed data streamed at 10 Hz–30 Hz, though generally multiple camera views are streamed simultaneously back to the pilot for safe manipulator control. In the case of our high-level automation system, the natural language data rates are based on approximate estimates for the average

letter count per word and the speech rate. This data rate represents the expected maximum bandwidth load when transmitted in real-time, as language based communication is intermittent and can be compressed. The scene state feedback includes the vehicle state such as the manipulator joint states and semantic information, such as the type and pose of detected tools. However, the visual scene state feedback takes up the bulk of the bandwidth and is assumed to be encoded as a compressed camera frame or view of the 3D scene reconstruction. As demonstrated in the table, communication requirements to support our high-level system reduce the necessary bandwidth load by at least an order of magnitude compared to the requirements of the most limited direct teleoperation modality.

Despite the technological challenges in reaching extraterrestrial worlds, the NASA Science Mission Directorate (SMD) sets its first priority “to discover the secrets of the universe, to search for life, and to protect and improve life on Earth” [133] and “is undertaking a flagship mission to Jupiter’s moon Europa, as its subsurface ocean has great potential to harbor extraterrestrial life.” A Europa mission concept for a surface lander has reached relative maturity, having passed its delta Mission Concept Review [68]. The sampling system is recognized as being critical to the success of the mission and relies on a robotic arm for “excavation, collection, and presentation (or transfer) of samples to scientific instruments for observation and analysis” [67]. Due to the anticipated communication limitations, it is likely that the lander will be required to self-select sampling sites, in which “the sampling system would be capable of conducting a sampling cycle in a fully autonomous fashion with no input from ground operators, from target selection to sample delivery. This autonomous capability is to guard against a prolonged telecommunications fault during the short mission lifetime, and will be in place to provide added assurance that the mission threshold science would be met” [67]. Challenges to the sampling system will be exacerbated by “poorly-characterized terrain at small scales”, and “the terrain immediately in front of the landing spot must suffice for sampling locations; there is no mobility system that can be used to search for a better site” [67]. The methods we demonstrated in this report for automated manipulator control and sample collection are directly applicable to operations focused on a Ladder of Life detection mission scenario [135]. With the exception of the wrist mounted camera, the manipulator and imaging system used in our demonstrations are very similar to the hardware for the Europa lander concept, consisting of a multi-DoF manipulator and vehicle-mounted stereo pair. A primary limiting factor on the integration of our autonomy methods with the lander would be the available computational power. However, for a stationary lander, the visual processing, which is the primary computational bottleneck, could operate at low-frame rates suitable for extraterrestrial exploration, assuming the environment dynamics are sufficiently slow. The methods we describe are also suitable to run on embedded systems and may be optimized accordingly.

## 6.6 Conclusions

An exobiology search mission to distant ocean worlds will require a highly automated exploratory vehicle, capable of operating in extreme conditions for an extended period of time. Such a platform will likely be outfitted with a manipulator to maximize the types of samples that could be collected. In this report we describe a vision system and control framework for automating an ROV manipulator. This architecture is readily integrated onto a wide array of vehicle platforms, and we have demonstrated the viability of the system in the field on two ROVs with different manipulators, including the *NUI* HROV which is dynamically reconfigurable. In November of 2019, we demonstrated planner-controlled sample collection and return within active submarine volcanoes that host diverse assemblages of extremophile organisms. These operation locations served as analogs to environments that may exist within other ocean worlds in our solar system and beyond.

A current limitation of our approach is a semi-static vehicle and scene assumption, where the ROV is held stationary and the scene does not change during execution of a manipulator motion, though the vehicle and scene state may change between motions. The vehicle is typically kept stationary by setting it down on the seafloor before manipulation is initiated. This assumption limits the type of sampling tasks that may be performed with the described system. For example collecting samples from a vertical wall, the underside of an ice shelf, or other moving objects would require free-floating control. Free-floating manipulation is an open problem in robotics, and a promising research direction that directly builds on our demonstrated system is obstacle aware disturbance rejection control of the manipulator. This method is similar to obstacle aware visual servoing, using feature based SLAM with the vision system to compensate for vehicle motions and stabilize the end-effector. A disturbance rejection approach would enhance the flexibility of the system to be easily integrated on different vehicles and manipulators without requiring the generation of complex vehicle and manipulator dynamic models.

While the demonstrated system represents a significant step towards autonomous sample collection and return from seafloor environments, more advancements are required before the system can be deployed reliably in a fully automated fashion. In particular, visual methods must be developed that are robust to the optical challenges of the underwater environment in order to enable safe and targeted sample collection and precision tool handling. These methods must be robust to dynamic scenes, insensitive to the intensity inconsistency of underwater lighting and perform well in sparsely featured and low-textured environments. Fusion of sparse feature based methods for SLAM with learning-based methods for dense scene reconstruction and high-level semantic scene understanding, such as segmentation, object detection and tool pose estimation may provide an appropriate path forward to overcome this challenge.

In summary, automated exploration of unstructured seafloor environments is within reach of

current underwater robotic technology. More development is needed, particularly in methods for scene reconstruction and understanding, to make this technology sufficiently reliable for fully automated deployment, but results from our oceanographic expeditions described in this report demonstrate that a wide range of existing ROVs and manipulator systems can be adapted, with moderate effort, for high level automation capabilities.

### **Acknowledgments**

This work was funded under a NASA PSTAR grant, number NNX16AL08G, and by the National Science Foundation under grants IIS-1830660 and IIS-1830500. The authors would like to thank the Costa Rican Ministry of Environment and Energy and National System of Conservation Areas for permitting research operations at the Costa Rican shelf margin and the Schmidt Ocean Institute (including the captain and crew of the *R/V Falkor*, and ROV *SuBastian*) for their generous support and making the FK181210 expedition safe and highly successful. Additionally, the authors would like to thank the Greek Ministry of Foreign Affairs for permitting the 2019 Kolumbo Expedition to the Kolumbo and Santorini calderas, as well as Prof. Evi Nomikou and Dr. Aggelos Mallios for their expert guidance and tireless contributions to the expedition. We would also like to thank Maritech and crew of the *CLV OceanLink*, and the HROV *NUI* crew for their skillful and friendly assistance with integration, testing, and field operations. Finally we would like to thank Prof. Blair Thornton for graciously sharing his underwater camera housing design, which was used in our work described here.

## CHAPTER 7

# Conclusions and Future Directions

### 7.1 Conclusions

This dissertation addresses some of the challenges of applying visual methods for robotic systems in underwater environments. In particular, the developed visual methods improve scene understanding to support high level automation of underwater vehicle manipulator systems (UVMSs). First, this dissertation contributes a novel deep learning based method for object pose estimation from monocular cameras and an extension of this method to fisheye and omni-directional cameras, which can aid UVMSs in tool detection and grasping. The method uses an intermediate silhouette abstraction to improve learning performance and mitigate the feature domain shift when training on synthetic datasets. Second, this dissertation contributes a feature based SLAM method that fuses features from an independent fisheye camera with a perspective stereo pair. This method is particularly suited to UVMSs or other mobile manipulator systems, where the stereo is mounted on the vehicle frame and provides scale accurate feature points of the workspace, while the fisheye camera is mounted near the manipulator wrist to enable active viewpoint acquisition and extension of the map beyond the stereo viewpoint. Third, this dissertation contributes an open source software tool that aids the design of underwater camera systems. The tool combines physics based models with practitioner knowledge acquired from working in the field to guide design choices through parametric selection. Fourth and finally, this dissertation contributes an automation framework for UVMSs, with validation from field trials conducted in natural deep ocean environments. Imagery datasets collected during these field trials supported the development of the visual methods in this dissertation and have been made publicly available for the underwater research community.

Ultimately, this work has advanced the state-of-the-art for underwater perception, and brings us closer to the realization of automated UVMSs, which can operate across our terrestrial oceans and explore the oceans of other worlds.

## 7.2 Future Directions

There are several immediate goals for future work that should be addressed before UVMSs can be deployed safely and reliably in natural and unstructured ocean environments. In particular, future work should focus on the development of dense real-time scene reconstruction methods that can inform safe, goal oriented, manipulation planning. Also, future work should develop fault-tolerant control methods that can adapt to the systematic failures of subsea manipulator hardware. Some particular future research directions are outlined below.

### **Synthetic Datasets for Learning in Underwater Domains**

Collecting underwater image datasets in natural environments is a challenging and expensive task, especially for datasets which require localized annotations referenced to some ground truth, such as object poses. Image appearance and quality in the underwater domain is also highly variable and dependent on both lighting and camera hardware design choices, as well as uncontrollable environmental parameters of the water column. These challenges have hindered progress in developing deep learning methods for the underwater domain. A promising direction for future research is the development of synthetic image rendering processes which incorporate the physics of underwater image formation, through either modeling or learning. These rendering engines could be combined with recent advancements in domain transfer methods, which facilitate the learning of features that bridge the appearance gap between simulated and real data.

### **Robust and Dense Real-Time Scene Reconstruction**

In field robotics, sensor fusion is an effective way to improve the robustness and accuracy of visual methods. For underwater vehicle SLAM systems, it is common to fuse measurements from localization sensors like an IMU or DVL. In the case of UVMSs, the vehicle remains relatively stationary while the manipulator is activated to complete the mission task. Building on our hybrid SLAM system, which enables active mapping with a wrist mounted camera, the fusion of a kinematic factor between the manipulator camera and the vehicle mounted cameras would improve the robustness of the system and enable real-time kinematic calibration of the manipulator.

Another direction to improve underwater scene reconstruction is the application of learned features and descriptors to the underwater domain, to improve feature matching and place recognition in diverse underwater environments. Experimental results in chapter 4 of this dissertation showed that deep learned descriptors can greatly improve feature matching performance. Deep learning could also be applied to the problem of dense mapping, by using depth estimation networks for monocular images to generate depth maps from the manipulator camera that are constrained by the sparse but optimized feature map.



### **Free-Floating Manipulation**

So far, free-floating manipulation demonstrations with UVMSs have been limited to pools or calm shallow water environments, with generally very slow execution of the manipulation task. For work class ROV, such as those used in the field trials for this dissertation, the weight of the vehicle is much greater than the manipulator, so dynamic coupling between the vehicle and manipulator is almost negligible. Also, it is general practice during ROV operations to set down the vehicle on the seafloor or rigidly secure it to a structure before performing the manipulation task, so the vehicle essentially becomes a stationary platform. However, when a manipulator is integrated onto a lighter weight AUV, the dynamic coupling between the vehicle and manipulator can be significant. Further, manipulation tasks may be performed above the seafloor, such as when collecting samples from a delicate reef, working under ice, or inspecting offshore infrastructure. In these scenarios, control methods which support free-floating manipulation are needed. A promising research direction that builds on the autonomy framework developed in this dissertation is obstacle aware disturbance rejection control of the manipulator, using feature based SLAM to compensate for vehicle motions and stabilize the end-effector. This approach to manipulator stabilization would not require complex dynamic models for the vehicle and manipulator, maintaining the flexible integration of the system with diverse UVMSs.

### **Fault-Tolerant Control**

Underwater manipulator systems are subjected to extended exposure to corrosive salt water, high pressures, and low temperatures, leading to inevitable hardware failures. These failures can express in loss of precise joint feedback and control or, in the worst case, complete loss of a joint's actuation. For an autonomously deployed UVMS, such a failure might cascade into aborting the mission, unless the vehicle and manipulator control system can adapt to such failures. Future research should focus on fault tolerant control of UMVSs, that can adapt to known types of systematic failures. In such a control architecture, the mobility of the vehicle might be used to compensate for a manipulator joint failure. Also, the 3D scene understanding provided by the vision system could inform what mission tasks are still viable, given the degraded system state.

## BIBLIOGRAPHY

- [1] Historical timeline. <https://oceanexplorer.noaa.gov/history/timeline/welcome.html?page=1>.
- [2] 2013. [https://www.bluebird-electric.net/submarines/alvin\\_dsv\\_submersible\\_woods\\_hole\\_oceographic\\_institution\\_us\\_navy.htm](https://www.bluebird-electric.net/submarines/alvin_dsv_submersible_woods_hole_oceographic_institution_us_navy.htm).
- [3] A brief history of rovs sea technology magazine, Sep 2019. <https://sea-technology.com/a-brief-history-of-rovs>.
- [4] Fisheye projection, 2019.
- [5] Optics of dome ports, 2019. <https://www.scubageek.com/articles/wwwdome.html>.
- [6] Pablo F Alcantarilla and T Solutions. Fast explicit diffusion for accelerated features in nonlinear scale spaces. *IEEE Trans. Patt. Anal. Mach. Intell*, 34(7):1281–1298, 2011.
- [7] Ricardo Amils. *Chemolithoautotroph*, pages 289–289. Springer Berlin Heidelberg, Berlin, Heidelberg, 2011. [https://doi.org/10.1007/978-3-642-11274-4\\_272](https://doi.org/10.1007/978-3-642-11274-4_272).
- [8] Relja Arandjelović and Andrew Zisserman. Three things everyone should know to improve object retrieval. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2911–2918. IEEE, 2012.
- [9] Hannes Arnold, Lucas Liuzzo, and Sven Simon. Magnetic signatures of a plume at Europa during the Galileo E26 flyby. *Geophysical Research Letters*, 46(3):1149–1157, 2019.
- [10] Benjamin Ayton, Brian Williams, and Richard Camilli. Measurement maximizing adaptive sampling with risk bounding functions. In *aaai*, pages 7511–7519, 2019.
- [11] V. Azizi, A. Kimmel, K. Bekris, and M. Kapadia. Geometric reachability analysis for grasp planning in cluttered scenes for varying end-effectors. In *2017 13th IEEE Conference on Automation Science and Engineering (CASE)*, pages 764–769.
- [12] NS Barrett, L Meyer, N Hill, and PH Walsh. Methods for the processing and scoring of auv digital imagery from south eastern tasmania. 2011.
- [13] Herbert Bay, Andreas Ess, Tinne Tuytelaars, and Luc Van Gool. Speeded-up robust features (surf). *Computer vision and image understanding*, 110(3):346–359, 2008.

- [14] Gideon Billings and Matthew Johnson-Roberson. Silhonet: An rgb method for 6d object pose estimation. *IEEE Robotics and Automation Letters*, 4(4):3727–3734, 2019.
- [15] Gideon Billings and Matthew Johnson-Roberson. Silhonet-fisheye: Adaptation of a roi based object pose estimation network to monocular fisheye images. *IEEE Robotics and Automation Letters*, 5(3):4241–4248, 2020.
- [16] Sandra E Billings and Simon A Kattenhorn. The great thickness debate: Ice shell thickness models for Europa and comparisons with estimates based on flexure at ridges. *Icarus*, 177(2):397–412, 2005.
- [17] A. Birk, T. Doernbach, Christian Mueller, T. Łuczyński, Arturo Gomez Chavez, D. Koehn-topp, Andras Kupcsik, S. Calinon, A. Tanwani, G. Antonelli, Paolo Di Lillo, E. Simetti, G. Casalino, Giovanni Indiveri, L. Ostuni, A. Turetta, A. Caffaz, P. Weiss, T. Gobert, B. Chemisky, J. Gancet, T. Siedel, S. Govindaraj, X. Martínez, and P. Letier. Dexterous underwater manipulation from onshore locations: Streamlining efficiencies for remotely operated underwater vehicles. *IEEE Robotics & Automation Magazine*, 25:24–33, 2018.
- [18] Mårten Björkman, Niklas Bergström, and Danica Kragic. Detecting, segmenting and tracking unknown objects using multi-label mrf inference. *Computer Vision and Image Understanding*, 118:111–127, 2014.
- [19] J. Bohg, A. Morales, T. Asfour, and D. Kragic. Data-driven grasp synthesis—a survey. 30(2):289–309.
- [20] Andrew D Bowen, Dana R Yoerger, Christopher C German, James C Kinsey, Michael V Jakuba, Daniel Gomez-Ibanez, Christopher L Taylor, Casey Machado, Jonathan C Howland, Carl L Kaiser, et al. Design of Nereid-UI: A remotely operated underwater vehicle for oceanographic access under ice. In *2014 Oceans-St. John's*, 2014.
- [21] Eric Brachmann, Alexander Krull, Frank Michel, Stefan Gumhold, Jamie Shotton, and Carsten Rother. Learning 6d object pose estimation using 3d object coordinates. In David Fleet, Tomas Pajdla, Bernt Schiele, and Tinne Tuytelaars, editors, *Computer Vision – ECCV 2014*, pages 536–551, Cham, 2014. Springer International Publishing.
- [22] James M Brooks, MC Kennicutt, CR Fisher, SA Macko, K Cole, JJ Childress, RR Bidi-gare, and RD Vetter. Deep-sea hydrocarbon seep communities: evidence for energy and nutritional carbon sources. *Science*, 238(4830):1138–1142, 1987.
- [23] David Broome, Trevor Larkum, and M Hall. Subsea weld inspection using an advanced robotic manipulator. In 'Challenges of Our Changing Global Environment'. *Conference Proceedings. OCEANS'95 MTS/IEEE*, volume 2, pages 1216–1224. IEEE, 1995.
- [24] Holger Caesar, Jasper Uijlings, and Vittorio Ferrari. Coco-stuff: Thing and stuff classes in context. In *Computer vision and pattern recognition (CVPR), 2018 IEEE conference on*. IEEE, 2018.

- [25] Berk Calli, Arjun Singh, James Bruce, Aaron Walsman, Kurt Konolige, Siddhartha Srinivasa, Pieter Abbeel, and Aaron M Dollar. Yale-cmu-berkeley dataset for robotic manipulation research. *The International Journal of Robotics Research*, 36(3):261–268, 2017.
- [26] Oscar Calvo, Alejandro Rozenfeld, Aandre Souza, Fernando Valenciaga, Pablo F Puleston, and G Acosta. Experimental results on smooth path tracking with application to pipe surveying on inexpensive auv. In *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on*, pages 3647–3653. IEEE, 2008.
- [27] Richard Camilli, Paraskevi Nomikou, Javier Escartín, Pere Ridao, Angelos Mallios, Stephanos P Kiliadis, Ariadne Argyraki, Muriel Andreani, Valerie Ballu, Ricard Campos, et al. The kallisti limnes, carbon dioxide-accumulating subsea pools. *Scientific reports*, 5:12152, 2015.
- [28] Errol Campbell, Nic Bingham, and Jason Williams. 4500 m remotely operated vehicle (ROV SuBastian), Oct 2019.
- [29] Carlos Campos, Richard Elvira, Juan J Gómez Rodríguez, José MM Montiel, and Juan D Tardós. Orb-slam3: An accurate open-source library for visual, visual–inertial, and multi-map slam. *IEEE Transactions on Robotics*, 2021.
- [30] Zhe Cao, Y. Sheikh, and N. K. Banerjee. Real-time scalable 6dof pose estimation for textureless objects. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2441–2448.
- [31] Romano Capocci, Gerard Dooly, Edin Omerdić, Joseph Coleman, Thomas Newe, and Daniel Toal. Inspection-class remotely operated vehicles—a review. *Journal of Marine Science and Engineering*, 5(1):13, 2017.
- [32] Steven Carey, Paraskevi Nomikou, Katy Croff Bell, Marvin Lilley, John Lupton, Chris Roman, Eleni Stathopoulou, Konstantina Bejelou, and Robert Ballard. Co2 degassing from hydrothermal vents at kolumbo submarine volcano, greece, and the accumulation of acidic crater water. *Geology*, 41(9):1035–1038, 2013.
- [33] Alexandra Carlson, Katherine A. Skinner, Ram Vasudevan, and M. Johnson-Roberson. Modeling camera effects to improve deep vision for real and synthetic data. In *European Conference on Computer Vision: Workshop on Visual Learning and Embodied Agents in Simulation Environments*, 2018.
- [34] Pep Lluís Negre Carrasco, Francisco Bonin-Font, and Gabriel Oliver-Codina. Global image signature for visual loop-closure detection. *Autonomous Robots*, 40(8):1403–1417, 2016.
- [35] Mingyou Chen, Yunchao Tang, Xiangjun Zou, Zhaofeng Huang, Hao Zhou, and Siyu Chen. 3d global mapping of large-scale unstructured orchard integrating eye-in-hand stereo vision and slam. *Computers and Electronics in Agriculture*, 187:106237, 2021.
- [36] Zhe Chen, Hongmin Gao, Zhen Zhang, Helen Zhou, Xun Wang, and Yan Tian. Underwater salient object detection by combining 2d and 3d visual features. *Neurocomputing*, 391:249–259, 2019.

- [37] P. Cieslak, P. Ridao, and M. Giergiel. Autonomous underwater panel operation by girona500 uvms: A practical approach to autonomous underwater manipulation. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 529–536, 2015.
- [38] Taco S. Cohen, Mario Geiger, Jonas Köhler, and Max Welling. Spherical cnns. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018.
- [39] David Coleman, Ioan Sucan, Sachin Chitta, and Nikolaus Correll. Reducing the barrier to entry of complex robotic software: A MoveIt! case study. *Journal of Software Engineering for Robotics*, 5(1):3–16, 2014.
- [40] Benjamin Coors, Alexandru Paul Condurache, and Andreas Geiger. Spherenet: Learning spherical representations for detection and classification in omnidirectional images. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 518–533, 2018.
- [41] Mike Cowlshaw, 2014.
- [42] Arun Das and Steven L. Waslander. Calibration of a dynamic camera cluster for multi-camera visual slam. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4637–4642, 2016.
- [43] G. De Novi, C. Melchiorri, J.C. Garcíanda, P.J. Sanz, P. Ridao, and G. Oliver. New approach for a reconfigurable autonomous underwater vehicle for intervention. *Aerospace and Electronic Systems Magazine, IEEE*, 25(11):32–36, nov. 2010.
- [44] Paul E Debevec and Jitendra Malik. Recovering High Dynamic Range Radiance Maps from Photographs. page 10.
- [45] Frank Dellaert. Factor graphs and gtsam: A hands-on introduction. Technical report, Georgia Institute of Technology, 2012.
- [46] L. Deng, M. Yang, Y. Qian, C. Wang, and B. Wang. Cnn based semantic segmentation for urban traffic scenes using fisheye camera. In *2017 IEEE Intelligent Vehicles Symposium (IV)*, pages 231–236, 2017.
- [47] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superpoint: Self-supervised interest point detection and description. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 224–236, 2018.
- [48] Thanh-Toan Do, Ming Cai, Trung Pham, and Ian Reid. Deep-6dpose: recovering 6d object pose from a single rgb image. *arXiv preprint arXiv:1802.10367*, 2018.
- [49] Diego Brito dos Santos Cesar, Christopher Gaudig, Martin Fritsche, Marco A. dos Reis, and Frank Kirchner. An evaluation of artificial fiducial markers in underwater environments. In *OCEANS 2015 - Genova*, pages 1–6, 2015.
- [50] B. Drost, M. Ulrich, N. Navab, and S. Ilic. Model globally, match locally: Efficient and robust 3d object recognition. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 998–1005, 2010.

- [51] Z Duguid and R Camilli. Improving resource management for unattended observation of the marginal ice zone using autonomous underwater gliders. *Frontiers in Robotics and AI*, 7:184, 2020.
- [52] Seibert Q. Duntley. Light in the Sea\*. *JOSA*, 53(2):214–233, February 1963.
- [53] Marc Eder and Jan-Michael Frahm. Convolutions on spherical images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 1–5, 2019.
- [54] Ryan M Eustice, Oscar Pizarro, and Hanumant Singh. Visually augmented navigation for autonomous underwater vehicles. *IEEE Journal of Oceanic Engineering*, 33(2):103–122, 2008.
- [55] Hui Feng, Xinghui Yin, Lizhong Xu, Guofang Lv, Qi Li, and Lulu Wang. Underwater salient object detection jointly using improved spectral residual and fuzzy c-means. *Journal of Intelligent & Fuzzy Systems*, 37(1):329–339, 2019.
- [56] Maxime Ferrera, Julien Moras, Pauline Trouvé-Peloux, and Vincent Creuze. Real-time Monocular Visual Odometry for Turbid and Dynamic Underwater Environments. *arXiv:1806.05842 [cs]*, February 2020. arXiv: 1806.05842.
- [57] FLir. Flir blackfly usb3 imaging performance specification, 2017. <https://www.ptgrey.com/support/downloads/10297>.
- [58] Terrence Fong, Charles Thorpe, and Charles Baur. *Collaborative control: A robot-centric model for vehicle teleoperation*, volume 1. Carnegie Mellon University, The Robotics Institute Pittsburgh, 2001.
- [59] RE Francois and WE Nodland. *Unmanned Arctic research submersible (UARS) system development and test report*. Applied Physics Laboratory, University of Washington, 1972.
- [60] Frank J. Frost. Scyllias: Diving in antiquity. *Greece & Rome*, 15(2):180–185, 1968. <http://www.jstor.org/stable/642431>.
- [61] Jodi Gaeman, Saswata Hier-Majumder, and James H Roberts. Sustainability of a subsurface ocean within Triton’s interior. *Icarus*, 220(2):339–347, 2012.
- [62] Dorian Gálvez-López and J. D. Tardós. Bags of binary words for fast place recognition in image sequences. *IEEE Transactions on Robotics*, 28(5):1188–1197, October 2012.
- [63] Andreas Geiger, Julius Ziegler, and Christoph Stiller. Stereoscan: Dense 3d reconstruction in real-time. In *Intelligent Vehicles Symposium (IV)*, 2011.
- [64] P. Goodarzi, M. Stellmacher, M. Paetzold, A. Hussein, and E. Matthes. Optimization of a cnn-based object detector for fisheye cameras. In *2019 IEEE International Conference on Vehicular Electronics and Safety (ICVES)*, pages 1–7, 2019.

- [65] M.D. Grossberg and S.K. Nayar. Modeling the space of camera response functions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(10):1272–1282, October 2004.
- [66] KP Hand, A Murray, J Garvin, W Brinckerhoff, B Christner, K Edgett, B Ehlmann, C German, A Hayes, T Hoehler, et al. Europa lander study 2016 report: Europa lander mission. *NASA Jet Propuls. Lab., La Cañada Flintridge, CA, USA, Tech. Rep. JPL D-97667*, 2017.
- [67] KP Hand, AE Murray, JB Garvin, et al. Europa lander study 2016 report, 2017.
- [68] KP Hand, CB Phillips, E Maize, G Reeves, J Pitesky, K Craft, M Cameron, G Tan-Wang, A San Martin, R Crum, et al. Europa lander mission concept (update 2021). In *Lunar and Planetary Science Conference*, number 2548, page 2120, 2021.
- [69] S. Harnad. The symbol grounding problem. *Physica D*, 42:335–346, 1990.
- [70] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.
- [71] Christian Hensen, Klaus Wallmann, Mark Schmidt, César R Ranero, and Erwin Suess. Fluid expulsion related to mud extrusion off costa rica—a window to the subducting slab. *Geology*, 32(3):201–204, 2004.
- [72] Franco Hidalgo, Chris Kahlefeldt, and Thomas Bräunl. Monocular orb-slam application in underwater scenarios. In *2018 OCEANS-MTS/IEEE Kobe Techno-Oceans (OTO)*, pages 1–4. IEEE, 2018.
- [73] M. Hildebrandt, J. Kerdels, J. Albiez, and F. Kirchner. A multi-layered controller approach for high precision end-effector control of hydraulic underwater manipulator systems. In *OCEANS 2009*, pages 1–5, 2009.
- [74] Stefan Hinterstoisser, Vincent Lepetit, Slobodan Ilic, Stefan Holzer, Gary Bradski, Kurt Konolige, and Nassir Navab. Model Based Training, Detection and Pose Estimation of Texture-Less 3d Objects in Heavily Cluttered Scenes. In Kyoung Mu Lee, Yasuyuki Matsushita, James M. Rehg, and Zhanyi Hu, editors, *Computer Vision – ACCV 2012*, pages 548–562, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg.
- [75] Stefan Hinterstoisser, Vincent Lepetit, Naresh Rajkumar, and Kurt Konolige. Going further with point pair features. In Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling, editors, *Computer Vision – ECCV 2016*, pages 834–848, Cham, 2016. Springer International Publishing.
- [76] Seonghun Hong, Dongha Chung, Jinwhan Kim, Youngji Kim, Ayoung Kim, and Hyeon Kyu Yoon. In-water visual ship hull inspection using a hover-capable underwater vehicle with stereo vision. *Journal of Field Robotics*, 36(3):531–546, 2019.
- [77] Berthold KP Horn. Closed-form solution of absolute orientation using unit quaternions. *Josa a*, 4(4):629–642, 1987.



- [78] Thomas M. Howard, S. Tellex, and N. Roy. A natural language planner interface for mobile manipulators. In *icra*, pages 6652–6659, 2014.
- [79] Hsiang-Wen Hsu, Frank Postberg, Yasuhito Sekine, Takazo Shibuya, Sascha Kempf, Mihály Horányi, Antal Juhász, Nicolas Altobelli, Katsuhiko Suzuki, Yuka Masaki, et al. Ongoing hydrothermal activities within Enceladus. *Nature*, 519(7542):207–210, 2015.
- [80] L. Iess, D. J. Stevenson, M. Parisi, D. Hemingway, R. A. Jacobson, J. I. Lunine, F. Nimmo, J. W. Armstrong, S. W. Asmar, M. Ducci, and P. Tortora. The gravity field and interior structure of Enceladus. *Science*, 344(6179):78–80, 2014.
- [81] K Ishimi, Y Ohtsuki, T Manabe, and K Nakashima. Manipulation system for subsea operation. In *Fifth International Conference on Advanced Robotics’ Robots in Unstructured Environments*, pages 1348–1353. IEEE, 1991.
- [82] Jules S Jaffe. Computer modeling and the design of optimal underwater imaging systems. *IEEE Journal of Oceanic Engineering*, 15(2):101–111, 1990.
- [83] Jules S Jaffe. To sea and to see: That is the answer. *Methods in Oceanography*, 15:3–20, 2016.
- [84] Bernd Jähne. Emva 1288 standard for machine vision. *Optik & Photonik*, 5(1):53–54, 2010.
- [85] Holger W Jannasch and Carl O Wirsén. Chemosynthetic primary production at East Pacific sea floor spreading centers. *Bioscience*, 29(10):592–598, 1979.
- [86] Francis A. Jenkins and Harvey E. White. *Fundamentals of Optics Fourth Edition*. McGraw-Hill, Inc, 1976.
- [87] MyungHwan Jeon, Yeongjun Lee, Young-Sik Shin, Hyesu Jang, and Ayoung Kim. Underwater object detection and pose estimation using deep learning. *IFAC-PapersOnLine*, 52(21):78–81, 2019.
- [88] Nils Gunnar Jerlov and Friedrich Franz Koczy. *Photographic measurements of daylight in deep water*. Elanders boktr., 1951.
- [89] Chiyu Max Jiang, Jingwei Huang, Karthik Kashinath, Prabhat, Philip Marcus, and Matthias Nießner. Spherical cnns on unstructured grids. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019.
- [90] J. Jiang, D. Liu, J. Gu, and S. Süsstrunk. What is the space of spectral sensitivity functions for digital color cameras? In *2013 IEEE Workshop on Applications of Computer Vision (WACV)*, pages 168–179, January 2013.
- [91] M. Johnson-Roberson, Mitch Bryson, Ariell Friedman, Oscar Pizarro, Giancarlo Troni, Paul Ozog, and Jon C. Henderson. High-resolution underwater robotic vision-based mapping and 3d reconstruction for archaeology. *Journal of Field Robotics*, 2016.

- [92] Matthew Johnson-Roberson, Oscar Pizarro, Stefan B. Williams, and Ian Mahon. Generation and visualization of large-scale three-dimensional reconstructions from underwater robotic surveys. *Journal of Field Robotics*, 27(1):21–51, 2010. [\\_eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/rob.20324](https://onlinelibrary.wiley.com/doi/pdf/10.1002/rob.20324).
- [93] Michael Kaess, Hordur Johannsson, Richard Roberts, Viorela Ila, John J Leonard, and Frank Dellaert. isam2: Incremental smoothing and mapping using the bayes tree. *The International Journal of Robotics Research*, 31(2):216–235, 2012.
- [94] J. Kannala and S.S. Brandt. A generic camera model and calibration method for conventional, wide-angle, and fish-eye lenses. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(8):1335–1340, 2006.
- [95] Juho Kannala and Sami S Brandt. A generic camera model and calibration method for conventional, wide-angle, and fish-eye lenses. *IEEE transactions on pattern analysis and machine intelligence*, 28(8):1335–1340, 2006.
- [96] Sertac Karaman and Emilio Frazzoli. Sampling-based algorithms for optimal motion planning. *ijrr*, 30(7):846–894, jun 2011.
- [97] Wadim Kehl, Fabian Manhardt, Federico Tombari, Slobodan Ilic, and Nassir Navab. Ssd-6d: Making rgb-based 3d detection and 6d pose estimation great again. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1521–1529, 2017.
- [98] Renata Khasanova and Pascal Frossard. Graph-based classification of omnidirectional images. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 869–878, 2017.
- [99] KK Khurana, MG Kivelson, DJ Stevenson, G Schubert, CT Russell, RJ Walker, and C Polanskey. Induced magnetic fields as evidence for subsurface oceans in Europa and Callisto. *Nature*, 395(6704):777–780, 1998.
- [100] Matthew Klingensmith, Siddhartha S Sirinivasa, and Michael Kaess. Articulated robot motion for simultaneous localization and mapping (arm-slam). *IEEE robotics and automation letters*, 1(2):1156–1163, 2016.
- [101] Dmitry A Konovalov, Alzayat Saleh, Michael Bradley, Mangalam Sankupellay, Simone Marini, and Marcus Sheaves. Underwater fish detection with weak multi-domain supervision. In *2019 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2019.
- [102] Kevin Köser and Udo Frese. Challenges in underwater visual navigation and slam. In *AI Technology for Underwater Robots*, pages 125–135. Springer, 2020.
- [103] Stefan Krause, Philip Steeb, Christian Hensen, Volker Liebetrau, Andrew W Dale, Marianne Nuzzo, and Tina Treude. Microbial activity and carbonate isotope signatures as a tool for identification of spatial differences in methane advection: a case study at the pacific costan margin. *Biogeosciences*, 11(2):507–523, 2014.

- [104] Alexander Krull, Eric Brachmann, Frank Michel, Michael Ying Yang, Stefan Gumhold, and Carsten Rother. *Learning Analysis-by-Synthesis for 6D Pose Estimation in RGB-D Images*, pages 954–962. 2015.
- [105] Haofei Kuang, Qingwen Xu, and Sören Schwertfeger. Depth estimation on underwater omni-directional images using a deep neural network. In *Workshop on Underwater Robotics Perception*, 2019.
- [106] Jean-Marc Lavest, Gérard Rives, and Jean-Thierry Lapresté. Underwater camera calibration. In *European Conference on Computer Vision*, pages 654–668. Springer, 2000.
- [107] Philippe Leclercq and John Morris. Robustness to noise of stereo matching. In *12th International Conference on Image Analysis and Processing, 2003. Proceedings.*, pages 606–611. IEEE, 2003.
- [108] Yeonkun Lee, Jaeseok Jeong, Jongseob Yun, Wonjune Cho, and Kuk-Jin Yoon. Spherephd: Applying cnns on a spherical polyhedron representation of 360deg images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9181–9189, 2019.
- [109] Patrick C Leger, Robert G Deen, and Robert G Bonitz. Remote image analysis for Mars Exploration Rover mobility and manipulation operations. In *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, pages 917–922, 2005.
- [110] Peter Lehner, Sebastian Brunner, Andreas Dömel, Heinrich Gmeiner, Sebastian Riedel, Bernhard Vodermayr, and Armin Wedler. Mobile manipulation for planetary exploration. In *Proceedings of the IEEE Aerospace Conference*, 2018.
- [111] Lisa A Levin, Guillermo F Mendoza, Benjamin M Grupe, Jennifer P Gonzalez, Britany Jellison, Greg Rouse, Andrew R Thurber, and Anders Waren. Biodiversity on the rocks: macrofauna inhabiting authigenic carbonate at costa rica methane seeps. *PLoS One*, 10(7):e0131080, 2015.
- [112] John S Lewis. Satellites of the outer planets: Their physical and chemical nature. *Icarus*, 15(2):174–185, 1971.
- [113] Chi Li, Jin Bai, and Gregory D Hager. A unified framework for multi-view multi-class object pose estimation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 254–269, 2018.
- [114] Chongyi Li, Jichang Guo, and Chunle Guo. Emerging from water: Underwater image color correction based on weakly supervised color transfer. *IEEE Signal processing letters*, 25(3):323–327, 2018.
- [115] Jie Li, Katherine A Skinner, Ryan M Eustice, and Matthew Johnson-Roberson. Watergan: Unsupervised generative network to enable real-time color correction of monocular underwater images. *IEEE Robotics and Automation letters*, 3(1):387–394, 2017.

- [116] Jinghui Li, Akitoshi Ito, and Yusuke Maeda. A slam-integrated kinematic calibration method for industrial manipulators with rgb-d cameras. In *2019 19th International Conference on Control, Automation and Systems (ICCAS)*, pages 686–689. IEEE, 2019.
- [117] Yi Li, Gu Wang, Xiangyang Ji, Yu Xiang, and Dieter Fox. Deepim: Deep iterative matching for 6d pose estimation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 683–698, 2018.
- [118] Vanesa Lopez-Vazquez, Jose Manuel Lopez-Guede, Simone Marini, Emanuela Fanelli, Espen Johnsen, and Jacopo Aguzzi. Video image enhancement and machine learning pipeline for underwater animal detection and classification at cabled observatories. *Sensors*, 20(3):726, 2020.
- [119] David G Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004.
- [120] Robert P Lowell and Myesha DuBose. Hydrothermal systems on Europa. *Geophysical Research Letters*, 32(5), 2005.
- [121] Zixin Luo, Tianwei Shen, Lei Zhou, Jiahui Zhang, Yao Yao, Shiwei Li, Tian Fang, and Long Quan. Contextdesc: Local descriptor augmentation with cross-modality context. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2527–2536, 2019.
- [122] Michael C Malin and Kenneth S Edgett. Evidence for recent groundwater seepage and surface runoff on Mars. *Science*, 288(5475):2330–2335, 2000.
- [123] Giacomo Marani, Song K. Choi, and Junku Yuh. Underwater autonomous manipulation for intervention missions auvs. *Ocean Engineering*, 36(1):15 – 23, 2009. Autonomous Underwater Vehicles.
- [124] Simone Marini, Emanuela Fanelli, Valerio Sbragaglia, Ernesto Azzurro, Joaquin Del Rio Fernandez, and Jacopo Aguzzi. Tracking fish abundance by underwater image recognition. *Scientific reports*, 8(1):1–12, 2018.
- [125] BL McGlamery. Computer analysis and simulation of underwater camera system performance. *SIO ref*, 75:2, 1975.
- [126] BL McGlamery. A computer model for underwater camera systems. In *Ocean Optics VI*, volume 208, pages 221–232. International Society for Optics and Photonics, 1980.
- [127] Frank Michel, Alexander Kirillov, Eric Brachmann, Alexander Krull, Stefan Gumhold, Bogdan Savchynskyy, and Carsten Rother. Global hypothesis generation for 6d object pose estimation. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 115–124, 2017.
- [128] W. Miyazaki and J. Miura. Object placement estimation with occlusions and planning of robotic handling strategies. In *2017 IEEE International Conference on Advanced Intelligent Mechatronics (AIM)*, pages 602–607.

- [129] Md Moniruzzaman, Syed Mohammed Shamsul Islam, Mohammed Bennamoun, and Paul Lavery. Deep learning on underwater marine object detection: a survey. In *International Conference on Advanced Concepts for Intelligent Vision Systems*, pages 150–160. Springer, 2017.
- [130] Rafael Monroy, Sebastian Lutz, Tejo Chalasani, and Aljosa Smolic. Salnet360: Saliency maps for omni-directional images with cnn. *Signal Processing: Image Communication*, 69:26–34, 2018.
- [131] Raul Mur-Artal, Jose Maria Martinez Montiel, and Juan D Tardos. Orb-slam: a versatile and accurate monocular slam system. *IEEE transactions on robotics*, 31(5):1147–1163, 2015.
- [132] Raul Mur-Artal and Juan D Tardós. Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras. *IEEE Transactions on Robotics*, 33(5):1255–1262, 2017.
- [133] NASA Science Mission Directorate. Science 2020-2024, a vision for scientific excellence, 2020.
- [134] Pep Lluís Negre, Francisco Bonin-Font, and Gabriel Oliver. Cluster-based loop closing detection for underwater slam in feature-poor regions. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2589–2595, May 2016.
- [135] Marc Neveu, Lindsay E Hays, Mary A Voytek, Michael H New, and Mitchell D Schulte. The ladder of life detection. *Astrobiology*, 18(11):1375–1402, 2018.
- [136] Fred E. Nicodemus. Directional reflectance and emissivity of an opaque surface. *Appl. Opt.*, 4(7):767–775, 6 1965.
- [137] Mikkel Cornelius Nielsen, Mari Hovem Leonhardsen, and Ingrid Schjøberg. Evaluation of posenet for 6-dof underwater pose estimation. In *OCEANS 2019 MTS/IEEE SEATTLE*, pages 1–6. IEEE, 2019.
- [138] Francis Nimmo, JR Spencer, RT Pappalardo, and ME Mullen. Shear heating as the origin of the plumes and heat flux on Enceladus. *Nature*, 447(7142):289–291, 2007.
- [139] P Nomikou, S Carey, KL Croff Bell, D Papanikolaou, K Bejelou, M Alexandri, K Cantner, and J F Martin. Morphological analysis and related volcanic features of the Kolumbo submarine volcanic chain (NE of Santorini Island, Aegean Volcanic Arc). *Zeitschrift für Geomorphologie*, 57(3):029–047, 2013.
- [140] P Nomikou, S Carey, D Papanikolaou, K Croff Bell, D Sakellariou, M Alexandri, and K Bejelou. Submarine volcanoes of the Kolumbo volcanic zone NE of Santorini Caldera, Greece. *Global and Planetary Change*, 90:135–151, 2012.
- [141] P Nomikou, MD Hannington, Sven Petersen, S Wind, V Heinath, S Lange, Marcel Rothenbeck, L Triebe, and Emanuel Wenzlaff. Advanced mapping of kolumbo submarine volcano (santorini) using auv abyss. 2019.
- [142] Andrew Nuss. Angler. <https://www.darpa.mil/program/angler>.

- [143] Marianne Oelund. Photons missing in action: Part 1: Lens t-stop. *Digital Photography Review*. <https://www.dpreview.com/forums/post/33785655>.
- [144] Paul Ozog, Matthew Johnson-Roberson, and Ryan M Eustice. Mapping underwater ship hulls using a model-assisted bundle adjustment framework. *Robotics and Autonomous Systems*, 87:329–347, 2017.
- [145] Chavdar Papazov, Sami Haddadin, Sven Parusel, Kai Krieger, and Darius Burschka. Rigid 3d geometry matching for grasping of known objects in cluttered scenes. 31(4):538–553.
- [146] Claudio Paschoa. Pioneer work class rovs (curv-i) – part 1, Jul 2014. [https://www.marinetechologynews.com/blogs/pioneer-work-class-rovs-\(curv-i-iii\)-e28093-part-1-700495](https://www.marinetechologynews.com/blogs/pioneer-work-class-rovs-(curv-i-iii)-e28093-part-1-700495).
- [147] Nathanaël Perraudin, Michaël Defferrard, Tomasz Kacprzak, and Raphael Sgier. Deep-sphere: Efficient spherical convolutional neural network with healpix sampling for cosmological applications. *Astronomy and Computing*, 27:130–146, 2019.
- [148] A. Peñalver, J. Pérez, J.J. Fernández, J. Sales, P.J. Sanz, J.C. García, D. Fornas, and R. Marín. *Annual Reviews in Control*, 40:201 – 211, 2015.
- [149] Bernd Pfrommer and Kostas Daniilidis. Tagslam: Robust slam with fiducial markers. *arXiv preprint arXiv:1910.00679*, 2019.
- [150] Oscar Pizarro, Ariell Friedman, Mitch Bryson, Stefan B Williams, and Joshua Madin. A simple, fast, and repeatable survey method for underwater visual 3d benthic mapping and monitoring. *Ecology and Evolution*, 7(6):1770–1782, 2017.
- [151] Mahdi Rad and Vincent Lepetit. Bb8: a scalable, accurate, robust to partial occlusion method for predicting the 3d poses of challenging objects without using depth. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3828–3836, 2017.
- [152] Dhruv Rathi, Sushant Jain, and S Indu. Underwater fish species classification using convolutional neural network and deep learning. In *2017 Ninth International Conference on Advances in Pattern Recognition (ICAPR)*, pages 1–6. IEEE, 2017.
- [153] R. Rios-Cabrera and T. Tuytelaars. Discriminatively trained templates for 3d object detection: A real time scalable approach. In *2013 IEEE International Conference on Computer Vision*, pages 2048–2055, 2013.
- [154] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski. Orb: An efficient alternative to sift or surf. In *2011 International conference on computer vision*, pages 2564–2571. Ieee, 2011.
- [155] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski. ORB: An efficient alternative to SIFT or SURF. In *2011 International Conference on Computer Vision*, pages 2564–2571, November 2011. ISSN: 2380-7504.

- [156] Alvaro Sáez, Luis M Bergasa, Eduardo Romeral, Elena López, Rafael Barea, and Rafael Sanz. Cnn-based fisheye image real-time semantic segmentation. In *2018 IEEE Intelligent Vehicles Symposium (IV)*, pages 1039–1044. IEEE, 2018.
- [157] Heiko Sahling, Douglas G Masson, César R Ranero, Veit Hühnerbach, Wilhelm Weinrebe, Ingo Klaucke, Dietmar Bürk, Warner Brückmann, and Erwin Suess. Fluid seepage at the continental margin offshore costa rica and southern nicaragua. *Geochemistry, Geophysics, Geosystems*, 9(5), 2008.
- [158] G. Salem, J. Krynitsky, M. Hayes, T. Pohida, and X. Burgos-Artizzu. Cascaded regression for 3d pose estimation for mouse in fisheye lens distorted monocular images. In *2016 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, pages 1032–1036, 2016.
- [159] Ahmad Salman, Shoaib Ahmad Siddiqui, Faisal Shafait, Ajmal Mian, Mark R Shortis, Khawar Khurshid, Adrian Ulges, and Ulrich Schwanecke. Automatic fish detection in underwater videos by a deep neural network-based hybrid motion learning system. *ICES Journal of Marine Science*, 02 2019.
- [160] Pedro J Sanz, Pere Ridao, Gabriel Oliver, Giuseppe Casalino, Yvan Petillot, Carlos Silvestre, Claudio Melchiorri, and Alessio Turetta. Trident an european project targeted to increase the autonomy levels for underwater intervention missions. In *2013 OCEANS-San Diego*, pages 1–10. IEEE, 2013.
- [161] Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.
- [162] M. Schwarz, H. Schulz, and S. Behnke. Rgb-d object recognition and pose estimation based on pre-trained convolutional neural network features. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1329–1335, 2015.
- [163] Yuhao Shan and Shigang Li. Discrete spherical image representation for cnn-based inclination estimation. *IEEE Access*, 2019.
- [164] Hyungwon Shim, Bong-Huan Jun, Pan-Mook Lee, Hyuk Baek, and Jihong Lee. Workspace control system of underwater tele-operated manipulators on an rovs. *Ocean Engineering*, 37(11):1036 – 1047, 2010.
- [165] Eli Silver, Miriam Kastner, Andrew Fisher, Julie Morris, Kirk McIntosh, and Demian Saffer. Fluid flow paths in the middle america trench and costa rica margin. *Geology*, 28(8):679–682, 2000.
- [166] Enrico Simetti, Francesco Wanderlingh, Sandro Torelli, Marco Bibuli, Angelo Odetti, Gabriele Bruzzone, Dario Lodi Rizzini, Jacopo Aleotti, Gianluca Palli, Lorenzo Moriello, et al. Autonomous underwater intervention: Experimental results of the maris project. *IEEE Journal of Oceanic Engineering*, 43(3):620–639, 2017.
- [167] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations*, 2015.



- [168] Satja Sivčev, Matija Rossi, Joseph Coleman, Edin Omerdić, Gerard Dooly, and Daniel Toal. Collision detection for underwater roV manipulator systems. *Sensors*, 18(4):1117, 2018.
- [169] Satja Sivčev, Joseph Coleman, Edin Omerdić, Gerard Dooly, and Daniel Toal. Underwater manipulators: A review. *Ocean Engineering*, 163:431 – 450, 2018.
- [170] Satja Sivčev, Matija Rossi, Joseph Coleman, Gerard Dooly, Edin Omerdić, and Daniel Toal. Fully automatic visual servoing control for work-class marine intervention rovs. *Control Engineering Practice*, 74:153 – 167, 2018.
- [171] Michael G. Solonenko and Curtis D. Mobley. Inherent optical properties of jerlov water types. *Appl. Opt.*, 54(17):5392–5401, 5 2015.
- [172] D Steinberg, Ariell Friedman, Oscar Pizarro, and Stefan B Williams. A bayesian non-parametric approach to clustering data from underwater robotic surveys. In *International Symposium on Robotics Research*, volume 28, pages 1–16. Citeseer, 2011.
- [173] Daniel M Steinberg, Stefan B Williams, Oscar Pizarro, and Michael V Jakuba. Towards autonomous habitat classification using gaussian mixture models. In *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4424–4431. IEEE, 2010.
- [174] Yu-Chuan Su and Kristen Grauman. Learning spherical convolution for fast features from 360 imagery. In *Advances in Neural Information Processing Systems*, pages 529–539, 2017.
- [175] Yu-Chuan Su and Kristen Grauman. Kernel transformer networks for compact spherical convolution. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9442–9451, 2019.
- [176] Richard Szeliski. *Computer vision: algorithms and applications*. Springer Science & Business Media, 2010.
- [177] Bugra Tekin, Sudipta N Sinha, and Pascal Fua. Real-time seamless single shot 6d object pose prediction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 292–301, 2018.
- [178] Eric Timmons, Benjamin Ayton, Andrew Wang, Nicholas Pascucci, Yuening Zhang, Nikhil Bhargava, Marlyse Reeves, Zachary Duguid, Daniel Strawser, Cheng Fang, et al. Risk-bounded, goal-directed mission planning and execution for autonomous ocean exploration. In *Proceedings of the Astrobiology Science Conference*, 2019.
- [179] Eric Timmons, Tiago Vaquero, Brian Williams, and Richard Camilli. Preliminary deployment of a risk-aware goal-directed executive on autonomous underwater glider. In *PlanRob Workshop, ICAPS (London, UK:)*, 2016.
- [180] Roger Y Tsai, Reimar K Lenz, et al. A new technique for fully autonomous and efficient 3 d robotics hand/eye calibration. *IEEE Transactions on robotics and automation*, 5(3):345–358, 1989.

- [181] Steffen Urban, Jens Leitloff, and Stefan Hinz. Mlpnp-a real-time maximum likelihood solution to the perspective-n-point problem. *arXiv preprint arXiv:1607.08112*, 2016.
- [182] P Vrolijk, R Camilli, L Summa, and P Nomikou. Cruise FK181210 on RV Falkor, 2019.
- [183] P Vrolijk, L Summa, B Ayton, P Nomikou, A Huepers, F Kinnaman, S Sylva, D Valentine, and R Camilli. Using a ladder of seeps with computer decision processes to explore for and evaluate cold seeps on the Costa Rica Active Margin. *Frontiers in Earth Science*, 9:143, 2021.
- [184] John Wang and Edwin Olson. AprilTag 2: Efficient and robust fiducial detection. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, October 2016.
- [185] Eric W Weisstein. Gnomonic projection, 2020.
- [186] Stefan B Williams, Oscar R Pizarro, Michael V Jakuba, Craig R Johnson, Neville S Barrett, Russell C Babcock, Gary A Kendrick, Peter D Steinberg, Andrew J Heyward, Peter J Doherty, et al. Monitoring of benthic reference sites: using an autonomous underwater vehicle. *IEEE Robotics & Automation Magazine*, 19(1):73–84, 2012.
- [187] Yuxin Wu et al. Tensorpack. <https://github.com/tensorpack/>, 2016.
- [188] Yu Xiang, Tanner Schmidt, Venkatraman Narayanan, and Dieter Fox. Posecnn: A convolutional neural network for 6d object pose estimation in cluttered scenes. In *Proceedings of Robotics: Science and Systems*, Pittsburgh, Pennsylvania, June 2018.
- [189] Wenwei Xu and Shari Matzner. Underwater fish detection using deep learning for water power applications. In *2018 International Conference on Computational Science and Computational Intelligence (CSCI)*, pages 313–318. IEEE, 2018.
- [190] D. Youakim, P. Ridao, N. Palomeras, F. Spadafora, D. Ribas, and M. Muzzupappa. Moveit!: Autonomous underwater free-floating manipulation. *IEEE Robotics & Automation Magazine*, 24(3):41–51, 2017.
- [191] Z. Zhang, C. Wang, Q. Zhang, Y. Li, X. Feng, and Y. Wang. Research on autonomous grasping control of underwater manipulator based on visual servo. In *2019 Chinese Automation Congress (CAC)*, pages 2904–2910, 2019.
- [192] Fenqiang Zhao, Shunren Xia, Zhengwang Wu, Dingna Duan, Li Wang, Weili Lin, John H Gilmore, Dinggang Shen, and Gang Li. Spherical u-net on cortical surfaces: methods and applications. In *International Conference on Information Processing in Medical Imaging*, pages 855–866. Springer, 2019.
- [193] Pengyu Zhao, Ansheng You, Yuanxing Zhang, Jiaying Liu, Kaigui Bian, and Yunhai Tong. Reprojection r-cnn: A fast and accurate object detector for 360° images. *arXiv preprint arXiv:1907.11830*, 2019.

- [194] Qiang Zhao, Chen Zhu, Feng Dai, Yike Ma, Guoqing Jin, and Yongdong Zhang. Distortion-aware cnns for spherical images. In *IJCAI*, pages 1198–1204, 2018.
- [195] Jun Zhu, Jiangcheng Zhu, Xudong Wan, Chao Wu, and Chao Xu. Object detection and localization in 3d environment by fusing raw fisheye image and attitude data. *Journal of Visual Communication and Image Representation*, 59:128–139, 2019.
- [196] Danping Zou and Ping Tan. Coslam: Collaborative visual slam in dynamic environments. *IEEE transactions on pattern analysis and machine intelligence*, 35(2):354–366, 2012.
- [197] Samantha Zuhlke. Ocean exploration: Timeline, Nov 2012. <https://www.nationalgeographic.org/media/ocean-exploration-timeline/>.