# Localization of an underwater robot chain

Juliette Drupt

## To cite this version:

Juliette Drupt. Localization of an underwater robot chain. Robotics [cs.RO]. Université de Toulon (UTLN), FRA., 2023. English. NNT : . tel-04488866

HAL Id: tel-04488866
https://theses.hal.science/tel-04488866
Submitted on 4 Mar 2024

University of Toulon
École Doctorale 548 — Mer et Sciences
COSMER Laboratory — EA 7398

Doctoral Thesis

# Localization of an underwater robot chain

*Author:*
Juliette Drupt

*Supervisor:*
Pr. Vincent Hugel
*Advisors:*
Dr. Claire Dune
Dr. Andrew Comport

*Jury:*

| | | |
|---|---|---|
| Pr. David Filliat | Rapporteur | Professor, ENSTA Paris |
| Pr. Luc Jaulin | Rapporteur | Professor, ENSTA Bretagne |
| Pr. Vincent Creuze | Président du jury | Professor, University of Montpellier |
| Pr. Juan D. Tardós | Examinateur | Professor, University of Zaragoza |
| Dr. Maxime Ferrera | Examinateur | Docteur, ingénieur IFREMER |
| Pr. Vincent Hugel | Directeur | Professor, University of Toulon |
| Dr. Claire Dune | Co-encadrante | Maître de conférences, University of Toulon |
| Dr. Andrew Comport | Co-encadrant | Chargé de recherche HDR, CNRS-I3S |

*A thesis submitted in fulfillment of the requirements
for the degree of Doctor of Philosophy — Docteur en automatique, signal,
productique, robotique*

Defended on November 29, 2023

# Declaration of Authorship

I, Juliette DRUPT, declare that this thesis titled, “Localization of an underwater robot chain” and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed: Juliette Drupt

Date: 29/11/2023

UNIVERSITY OF TOULON

# *Abstract*

École Doctorale 548 — Mer et Sciences
COSMER Laboratory — EA 7398

Doctor of Philosophy — Docteur en automatique, signal, productique, robotique

**Localization of an underwater robot chain**

by Juliette DRUPT

The current thesis focuses on the localization of a an underwater robot chain. Two different but complementary approaches are studied, including the proprioceptive localization of the chain based on and estimation of the three-dimensional state of its cable parts, using inertial measurements, and the exteroceptive, multi-agent localization of the chain with respect to its environment, using visual simultaneous localization and mapping techniques.

# *Acknowledgements*

J'ai longtemps pensé que l'écriture des remerciements dans la version finale du manuscrit était une étape simple et gratifiante, comme une respiration en fin de thése. Quelle naïveté. En réalité je me trouve aujourd'hui face à ma page blanche, cherchant comment articuler en l'espace d'une page ou deux la reconnaissance immense que j'ai envers toutes celles et ceux qui ont fait de mes trois années de thèse une expérience forte, et inoubliable. Vais-je vraiment réussir à n'oublier personne ?

Par qui pourrais-je commencer sinon par mon trio d'encadrants, Claire, Andrew et Vincent, qui ont assez cru en moi pour m'inviter sous le soleil du sud pour "faire plouf" et tenter de faire SLAMer un système maison de robots sous-marins encordés. Je vous dois énormément pour votre investissement, votre soutien et vos conseils avisés. Je suis très fière d'avoir été votre doctorante. D'un point de vue plus personnel, un merci tout particulier à Claire pour m'avoir entraînée joyeusement dans tes plans (d'expériences mais pas que) plus ou moins improbables, acrobatiques et aquatiques (j'aurai croisé deux poulpes). Merci à Andrew pour son implication malgré la distance qui sépare Toulon de Sophia-Antipolis, et pour m'avoir encadrée jusqu'au Japon pour ma première conférence internationale — une expérience incroyable. Merci d'avoir été le premier à m'envoyer des encouragements le jour de la soutenance malgré ta situation compliquée actuelle, j'espère que tu te rétabliras au plus vite. Merci enfin, au "chef", Vincent, toujours présent pour relire des équations, opérer un BlueROV à cœur ouvert, et rappeler à Claire de rédiger sa H. . . pardon, je n'ai rien dit !

Un merci immense à mes petits frères de thèse, Bilal et Clémentin. La thèse n'aurait pas eu la même saveur sans vous. Du gossip, de la subversion, beaucoup d'humour, des animaux, du sens de la vie, une sirène en chemise à fleurs, et un vrai soutien dans les moments plus compliqués.

Merci à Basile, mon fidèle disciple sous-marin, brave petit BlueROV qui aura vu la mer et bourlingué jusqu'en Espagne pour servir la science dans la joie.

Merci aux copains doctorants venus enrichir nos rangs : Cristiam, son vélo, ses avions et sa clavicule, mais aussi Léo le chef d'orchestre, Aurélien, Charly, les "filles du MAPIEM"... Evidemment un grand merci à Martin pour son soutien en premier lieu musical, et finalement pas que. Un clin d'œuil également aux "petits" et aux "expats" : François et Giacomo de l'Imath, Alexis "loin là-bas à l'IUT", et Louis, le benjamin. Martin, Louis, prenez bien soin de vos encadrants ! Un clin d'œil aux "vieux" aussi : merci aux anciens de m'avoir accueillie lorsque je n'étais encore qu'une petite nouvelle, merci à Ornella, camarade de mes premiéres manips ! Je m'aperçois que j'oublie les escl. . . pardon, les stagiaires : merci à vous également, Nil, Mathis 1, Mathis 2, Ryane, Mathis 3, Lorenzo.

Merci à Christophe de m'avoir donné l'opportunité de collaborer sur la question de la localisation des ROVs par câble, qui a été une expérience très enrichissante. Merci à Sabine pour le soutien matériel sur les BlueROVs et autres accessoires de manips, merci à Nicolas pour son excellente question lors de ma soutenance, à laquelle je n'ai pas encore eu l'occasion de pleinement répondre mais que je n'oublie pas, merci aux chercheurs de l'Imath pour l'ambiance lors des pauses dej', et à tous ceux que je pourrais avoir oubliés, et j'en suis contrite.

Merci à mes amis, et en particulier à Nico d'avoir fait le déplacement pour le jour J, Mathilde bien sûr, Lag depuis ses propres galères de thèse londoniennes, Isabelle

(j'aurai vu Shaka Ponk en tant que thésarde et docteur : j'ai vérifé, ça ne change absolument rien), Valérie et Sebastian mes profs de guitare niçois, et beaucoup de gens que je vois moins mais qui m'ont permis de faire des chouettes pauses mentales. Merci à ceux qui sont entrés puis sortis de ma vie mais m'ont aidée à faire les bons choix, Alexandre, Cyrille. Je profite de ce paragraphe pour amener une pensée pour Victor, camarade de promo pas si proche de moi mais si chouette, qui faisait le choix de la robotique sous-marine en même tant que moi en 2020, et est disparu en montagne l'an dernier.

Poursuivons par ma famille. Merci à ma mère — la meilleure — pour son soutien inconditionnel. Merci à mon père pour s'être intéressé à mon travail, à mon frère pour s'être levé à 4h pour suivre la soutenance outre-atlantique. Merci à feu mes grands-parents, pour avoir légitimé mon intérêt pour les sciences, à Mamy Pompon pour m'avoir encouragée à persévérer jusqu'en prépa, j'espère qu'elle aurait été fière de la suite.

Enfin, merci bien sûr à mon jury, pour s'être intéressés à mon travail et avoir amené des discussions intéressantes au cours de le lecture du manuscrit et de la soutenance.

# Contents

# List of Figures

# List of Tables

# Acronyms

**AHRS** Attitude and Heading Refence System. 18

**APE** Absolute Position Error. 53, 54, 115–119

**AUV** Autonomous Underwater Vehicle. vii, xi, 1, 4, 8, 10, 11, 21, 22, 26, 54, 108, 163, 165, 166

**BoW** Bags of Worlds. 39, 40, 53

**DVL** Doppler Velocity Log. 16, 18, 19, 28

**EKF** Extended Kalman Filter. 29

**FOG** Fiber Optic Giroscope. 19

**GBA** Global Bundle Adjustement. xv, 39, 40, 51, 54, 114

**HROV** Hybrid Remotely Operated Vehicle. xi, 8, 11, 12, 165, 166

**IMU** Inertial Measurement Unit. vii, viii, xii–xv, 5, 16, 18, 19, 25, 28, 49, 59, 60, 72–78, 80–93, 127, 128, 152, 154, 166, 167, 170

**INS** Inertial Navigation Systems. 19

**KF** KeyFrame. ix, 30, 34–40, 51, 53, 54, 107, 110, 111, 114, 116, 121–123, 172

**LARS** Launch And Recovery System. xi, 8, 12–14

**LBA** Local Bundle Adjustement. xii, 37

**LBL** Long BaseLine. xi, 16, 17

**MP** Map Point. 37

**RMS** Root Mean Square. 53, 54, 64, 65, 115–119

**ROV** Remotely Operated Vehicle. vii, viii, x–xiii, xv, 1, 2, 4–9, 11–15, 21–23, 25, 26, 54, 57, 59, 70–74, 76, 84–89, 91–93, 95, 104, 108, 123–125, 127–129, 141, 142, 152, 163–167, 169–173

**SBL** Short BaseLine. xi, 16, 17

**SfM** Structure from Motion. xiii, 49, 89, 90, 113

**SLAM** **S**imultaneous **L**ocalization **A**nd **M**apping. vii, viii, x, xii–xv, 5–7, 27–33, 35–41, 43, 45, 47–53, 57, 96–98, 104, 105, 107–112, 115, 117, 119, 123–125, 128, 129, 157–161, 168, 169, 173, 174

**SONAR** **SO**und **NA**vigation **R**anging system. xi, 16–18, 28, 48, 49, 98, 166

**TMS** **T**ether **M**anagement **S**ystem. xi, 13, 14, 91

**USBL** **U**ltra**S**hort **B**ase**L**ine. xi, 16, 17

**USV** **U**nmanned**S**urface **V**ehicle. 21

**UUV** **U**nderwater **U**nderwater **V**ehicle. vii, xiv, 4, 7–9, 11, 13, 15–17, 19, 165, 166

**VSLAM** **V**isual **S**imultaneous **L**ocalization **A**nd **M**apping. viii–x, xv, 5, 28–34, 36, 40, 41, 47, 48, 50–55, 58, 83, 84, 95–99, 101, 103–105, 107–111, 113–115, 117, 119, 121, 123–125, 127–130, 157–161, 168, 170, 171, 173, 174

# Chapter 1

# Introduction

Underwater robots are essential to conduct a wide variety of tasks which cannot be performed by human beings for physical or safety reasons. These unmanned vessels, however, face important challenges. Because of the rapid absorption of electromagnetic waves within a few meters, real-time communication between an underwater vehicle and a surface operator is limited. These high communication constraints led to two opposite strategies in underwater robots conception, which correspond to two vehicle families:

- Underwater Autonomous Vehicles (AUVs), designed to operate fully autonomously;
- Remotely Operated Vehicles (ROVs), connected to a surface station by a physical link, in order to be teleoperated by a human operator.

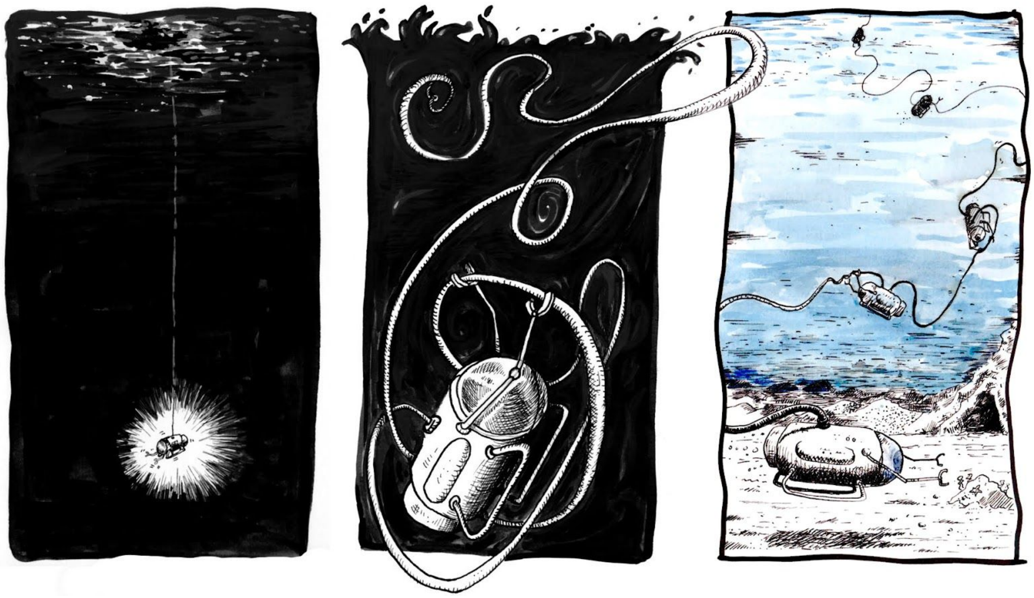


FIGURE 1.1: Artists'view of an underwater robot chain, Pierre Straumann

ROV cable management is one of the research axis of the COSMER Lab, University of Toulon, France, and lead to the introduction of the underwater robot chain concept, which consists in managing the cable of a ROV using intermediary robotic devices placed along the cable such that they control its shape, as in the artist's view provided in Figure 1.1. This concept was initially introduced as part as the PhD work of Matheus Laranjeira (Laranjeira et al., 2017; Laranjeira et al., 2018; Laranjeira et al., 2019; Laranjeira, 2019; Laranjeira et al., 2020). In continuation of these works, the current thesis focuses on the localization of a chain of ROV using embedded sensors measurements, and has been prepared in the COSMER Lab in collaboration with the CNR-I3S, Sophia-Antipolis, France.

The current chapter offers a general introduction to this thesis. The context of underwater robotic applications and operational challenges is described in Section 1.1. The scientific focus and the contribution are detailed in Section 1.2. The publications issued in this context are listed in Section 1.3. Section 1.4 gives the outline of the thesis.

## 1.1 Underwater robotic applications and operational challenges

### 1.1.1 Applications

Oceans cover 70% of Earth, but more than 95% of their volume remains unexplored. Underwater operations, thus, represent a challenge for both scientific exploration, and civil and military applications.

The scientific study of oceans, or oceanography, encompasses several fields: geology, biology, ocean physics and chemistry, meteorology. . . These fields are also interconnected. For instance, the study of the seafloor not only allows studying the seismic behavior of the ocean's floor and thus predict earthquakes and tsunamis, but also locate and characterize marine natural resources and to build accurate maps of the oceans, which are a key tool for all other marine related fields, by associating observations to a precise location, whithin a geographic context. The joint observation of soil erosion, marine ecosystems, physical phenomenon and water composition, for instance, is also crucial in measuring the impact of human activities on the oceans. These scientific applications need to acquire measurements in places which are not accessible by human beings. Oceans also house many archaeological sites which have been preserved by water over years, whose exploration require both measurements and manipulations.

Oceans have always played an important part in the industry for its exploitable resources. Today, offshore industries exploit seabed resources, like oil and gas and metals, but also use water currents as a renewable energy resource, through marine wind farms. These applications imply infrastructure installation, inspection, maintenance, and eventual dismantling at depth sometimes exceeding 1,000 meters. Oceans are also crucial in terms of trade and communication. Commercial routes and intercontinental communications through optical fiber cables are particularly important in today's world. For these aspects, oceans are also a source of economic, industrial and information power, and controlling the oceans involves political and military issues.

Underwater operations also include rescue missions and wreck search in case of plane crash or marine vessel accident. Water decontamination, more specially from plastics and oils, is another actual challenge to preserve oceans from accidents, like oil slicks, and man's long-term impact.

Finally, various applications which require ocean's observation but also marine interventions. The underwater environment is, though, hostile to human beings.

Because of the effect of pressure on the human body, divers are limited in terms of depth, the recommended technical diving limit being of 100 meters. To this end, manned submarine vessels have been developed. In 1948, Auguste Piccard performed the first dive of a manned, controllable submarine engine. Such engins were further investigated for scientific and military purposes. While military submarines have a maximum immersion depth of a few hundred meters, scientific vessels were designed to reach several thousands of meters, arising important technical challenges and safety issues, and overall cost. Alongside with the development of robotics and automation, scientific manned underwater vehicles have been gradually replaced by robots. Only few of them remain in activity, one noteworthy example being the *Nautile* of the French Marine Institute *Ifremer*, which has been developed in the 1980s' and is able to perform 8-hour dives, up to 6,000 meters, with 3 people onboard.

**Short Stories #1– HERGÉ AND THE BATHYSCAPHE**

In 1943, Hergé made a new character ringing Tintin's doorbell, with a one-seat shark-shaped submarine to sell. Resemblance between Professor Calculus (or Tournesol) and its model Auguste Piccard is striking. The main physical difference relies on the character's height, far lower than Piccard's 1.96 m height.

FIGURE 1.2: Extract from *Red Rackham's Treasure* (*Le Trésor de Rackham la Rouge*), p.12, Hergé, 1945.

Closely following the scientific progress of his time, Hergé made him the author of key works on the conquest of space, which would lead him to the moon in 1950, 19 years before the Americans.

### 1.1.2 Underwater environment and specific challenges

Underwater robotics is particularly challenged by the water medium and ocean physics. Underwater systems need to be protected from hostile condition, but also to cope with water physics, waves and currents when moving. In addition, the physical properties of the water impact the functions of some sensors and transceivers.

Increasing pressure, but also the water and the salt it contains are aggressive for the engines, and in particular for the sensors and connection cables, which have thus to be protected. Connection wires are commonly inserted into deformable tubes full of oil, in order to protect them from water contact but allow equi-pressure with respect to the water. A special attention has also to be given to sensor housing, in order to protect the sensors but not to deteriorate measurements. The aquatic environment is also challenging in terms of operational reliability, since vessel assistance or recovery in case of system failure is particularly difficult.

In addition, underwater robots control is necessarily 3-dimensional and is challenged by water dynamics, currents, and waves when operating close to the surface. Environment perception can also be limited. During deep dives, robots may evolve without features within their sensors' ranges when diving down the water column, before reaching their operating depth. Some sensors are also directly impacted by water's physical properties. For instance, the absorption of electromagnetic waves by the water, but also diffusion phenomenon modify the visual perception from cameras.

Electromagnetic waves' absorption by the water medium has had a particularly strong impact on underwater systems design by preventing real-time wireless communications. Radio-waves are indeed completely absorbed by the water within a few centimeters. Consequently, underwater unmanned vehicles (or UUVs) are divided into two families: Underwater Autonomous Vehicles (AUVs), and Remotely Operated Vehicles (ROVs).

AUVs are designed to be fully autonomous during their missions with no need to communicate in real time with the surface. They are mainly used to cover wide areas in missions such as seabed mapping or wreck search. However, such robots can only achieve limited tasks and cannot face properly to unexpected situations nor realize very precise tasks. On the other hand, remotely operated vehicles (ROVs) are connected to the surface through a physical link and can thus be teleoperated in order to accomplish tasks that require precision and adaptability. The tether can also provide energy to the ROV for greater autonomy and reduce the ROV's payload thanks to the absence of onboard batteries. Still, the tether limits the mobility of the ROV by involving an energy-intensive drag force, mechanical actions and risks of entanglements. Tether management strategies are therefore crucial to limit these effects and allow exploration in cluttered environments such as caves, flooded mines or shipwrecks. Such environments require the cable to be shaped to fit into a constrained free space. This can only be achieved by controlling the global shape of the cable itself, which gives rise to a new type of active tether management strategy: the chain of ROVs concept, also called underwater robot chain. This tether management strategy consists in the addition of intermediary robots between the lead ROV and the surface vessel, in order to actively control the three-dimension global shape of the tether. Implementing, and more specially controlling such a system implies a knowledge of its location with respect to its environments, what is the scientific focus of the current thesis.

## 1.2 Scientific focus and contributions

This thesis focuses on the self-localization of a chain of ROVs using embedded sensors measurements. While underwater robot localization with respect to its environment can be estimated from the robot's embedded sensors, the presence of an umbilical can be taken as an advantage to localize the robot at its end point, as it can be deduced from the knowledge of the cable three-dimensional state. As a result, ROV chain localization has been examined with these two different but complementary approaches, namely:

- the proprioceptive localization of a robot chain based on the estimation of the three-dimensional state of the cable with respect to the robots, using inertial measurements,
- the exteroceptive, multi-agent localization of the robot chain with respect to its environment, using visual simultaneous localization and mapping techniques.

We list our contributions as follows:

**Contribution 1.** Chapter 3 presents the contributions in cable-based state estimation for underwater tethered vehicles, which are:

- **experimental evaluation of the validity of the catenary model for moving submarine cables with negative buoyancy**. This work focuses on a simple model of flexible non-rigid hanging cables, namely the catenary model, and presents an experimental evaluation of the validity of this model to approximate the shape of dynamically moving underwater tethers with negative buoyancy using motion tracking.
- **an inertial-measurement-based catenary shape estimation method for a negatively buoyant cable, connecting a pair of underwater robots**. This work uses the calculation of local cable tangents using the data acquired from IMUs attached to the cable in one or two points, near its ends. The cable is modeled by a catenary, whose parameters are deduced analytically from the tangent measurements. The proposed method is evaluated experimentally, in the air and in pool, and is compared to a visual-based shape estimation from previous works, demonstrating better robustness and accuracy.
- **a ROV localization method based on umbilical angle measurement**. This work is the result of a collaboration with Christophe Viel from the CNRS Lab-STICC, ENSTA Bretagne. The umbilical is constrained to a piecewise linear shape using sliding buoys and ballasts placed on it. Cable segments orientation is deduced from IMU measurements, from which the global cable shape is deduced, allowing to locate the ROV at its end.

**Contribution 2.** Chapter 4, presents our contributions in underwater visual SLAM for single and multi-agent scenarios, which can be listed as follow:

- **monocular VSLAM benchmarking on underwater datasets**. A qualitative benchmark of six state-of-the-art monocular VSLAM algorithms (namely ORB-SLAM, ORB-SLAM3, Dual-SLAM, DSO, LDSO and DSM) was performed on eight underwater datasets featuring different underwater environments, camera motions, and visual conditions, including two new datasets.
- **multi-agent VSLAM**. A new multi-agent and multi-map monocular SLAM framework based on ORB-SLAM3, namely MAM$^3$SLAM was introduced and compared to the state-of-the-art multi-agent VSLAM on aerial and underwater scenarios. New multi-agent underwater datasets were acquired and released.

## 1.3 Publications

**Journal papers:**

- ROV localization based on umbilical angle measurement.
  C. Viel, J. Drupt, C. Dune, V. Hugel,
  Ocean Engineering, Volume 269, 2023, 113570, ISSN 0029-8018.

**Conference papers:**

- Inertial-measurement-based catenary shape estimation of underwater cables for tethered robots.
  J. Drupt, C. Dune, A. I. Comport, S. Seillier and V. Hugel,
  2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Kyoto, Japan, 2022, pp. 6867-6872.
  Video 1: `https://www.youtube.com/watch?v=TKLLVTSUN8s`
  Video 2: `https://www.youtube.com/watch?v=x-SXut75vHk`

- Qualitative evaluation of state-of-the-art DSO and ORB-SLAM-based monocular visual SLAM algorithms for underwater applications.
  J. Drupt, C. Dune, A. I. Comport and V. Hugel,
  OCEANS 2023, Limerick, Ireland, 2023, pp. 1-7.

- Estimation de forme de câble pesant pour la localisation de robots sous-marins encordés : comparaison d'une approche visuelle à une nouvelle approche inertielle.
  J. Drupt, C. Dune, A. I. Comport and V. Hugel,
  ORASIS 2023, Carqueiranne, France, 2023.

- An augmented catenary model for underwater tethered robots.
  M. Filliung, J. Drupt, C. Peraud, C. Dune, N. Boizot, A. I. Comport and V. Hugel,
  Submitted to 2024 IEEE International Conference on Robotics and Automation (ICRA).

**Workshops:**

- Validity of the catenary model for moving submarine cables with negative buoyancy.
  J. Drupt, C. Dune, A. I. Comport and V. Hugel,
  3rd workshop on RObotic MAnipulation of Deformable Objects: challenges in perception, planning and control for Soft Interaction (ROMADO-SI), Best Paper Award, Kyoto, Japan, 2022.
  Video: `https://www.youtube.com/watch?v=kmcs9xKf3KQ`

## 1.4 Thesis outline

Chapter 1 introduces the context of this thesis and its scientific focus. Chapter 2 presents the state-of-the art related works. Chapter 3 presents the contributions in cable-based state estimation for underwater tethered systems. Chapter 4 describes the contributions in visual SLAM for multiple underwater vehicles. Chapter 5 provides a general conclusion of this thesis and discusses perspectives.

# Chapter 2

# State of the art

## 2.1 Introduction

The scientific focus of this thesis is the localization of a chain of ROVs in cluttered environnement, with embedded sensors measurements. This subject arises three main issues. First, robot chain localization involves a knowledge of its configuration, which can be defined as the relative localization of the robots of the chain and of the cable portions between them. Second, a knowledge of the environment is necessary in order to locate the system with respect to the surrounding obstacles for future cable control purposes. Third, chain geo-referencing may be required.

The main specificities of the system under study is to be underwater, tethered, and multi-robot, with embedded sensors. In addition, cable and robot localization problems are strongly coupled:

- knowing the shape of the cable allows localizing the robot at its end point;
- conversely, cable location can be deduced from robot location given a cable model.

Consequently, the current chapter brings together several states-of-the-art, namely underwater robotics including ROV cable management and underwater perception, tethered robotic systems and online localization and mapping using embedded sensors for multiple agents.

This chapter is therefore organized as follows. Section 2.2 focuses on the underwater context of the ROV chain by presenting underwater unmanned vehicles, ROV tether management strategies and the usual embedded sensors used for underwater vehicles perception. Section 2.3 presents the state-of-the-art on state estimation for tethered robotic systems, including cable perception and shape estimation. Section 2.4 focuses on simultaneous localization and mapping (SLAM) for an underwater robot using a monocular camera, and provides a discussion on the choice of this sensor. Section 2.5 gives the state-of-the art on multi-agent monocular visual SLAM.

## 2.2 Unmanned underwater vehicles (UUVs) and cables

This section focuses on unmanned underwater vehicles (UUVs). These vehicles can be teleoperated or autonomous. Teleoperation requires real-time communication between the UUV and a control station. Because of the rapid absorption of the electromagnetic waves by the water medium, a wireless real-time communication scheme cannot be set for UUV teleoperation. Consequently, UUVs can be classified in three categories:

- Remotely Operated Vehicles (ROVs), which are connected to a surface station via a cable for real-time control and feedback
- Autonomous Underwater Vehicles (AUVs) which cannot communicate to the surface, or only in a very limited manner, and rely on embedded intelligence in operating autonomously
- Hybrid ROVs (HROVs), which can operate as ROVs or AUVs.

Section 2.2.1 presents these three kinds or UUVs, Section 2.2.2 focuses on the specific problem of ROV cable management, and Section 2.2.3 describes the usual sensors involved in underwater perception for UUVs.

### 2.2.1 UUVs classification

UUVs can be classified into ROVs, AUVs and HROVs. Historically, ROVs were the very first solution to the problem of UUV design. AUVs design has been investigated more recently, in parallel with scientific advances in embedded intelligence. HROVs are the most recent, and intend to combine the assets of both ROVs and AUVs. ROVs, AUVs and HROVs are presented in Sections 2.2.1.1, 2.2.1.2 and 2.2.1.3 respectively.

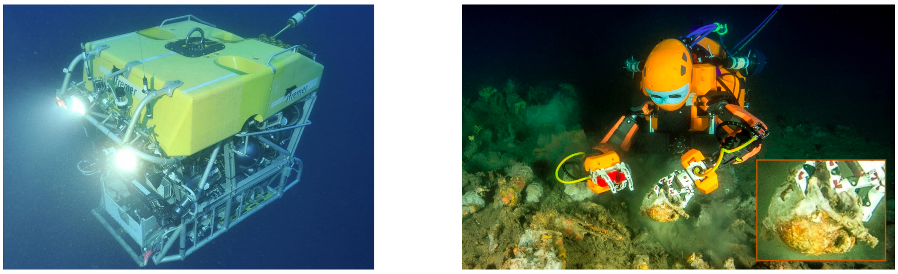
#### 2.2.1.1 Remotely Operated Vehicles (ROVs)

Remotely Operated Vehicles, or ROVs, are connected to the surface through a physical link and can thus be teleoperated in order to accomplish tasks that cannot be fully automated because they require precision or adaptability. This cable can also provide energy to the ROV for better autonomy with a lighter embedded payload. ROVs are commonly used in missions involving manipulations or high safety requirements, and are designed to be stable and maneuverable. Their applications encompass many fields, including infrastructure inspection and maintenance, scientific research for oceanographic or archaeological purposes, and search and rescue.

ROVs can be classified into three main categories according to their size and capabilities (Christ et al., 2014; Laranjeira, 2019). Work class ROVs weigh more than 1,000 kg and are designed to operate from 3,000 to 10,000 meter depth. They are usually designed to handle tasks requiring heavy manipulation, such as offshore infrastructure construction and maintenance for the oil and gas industry, or heavy civil engineering. Because of the important amount of force involved, work class ROVs are hydraulically propelled and actuated. Mid-sized ROVs are smaller vehicles, with weight from 100 to 1,000 kg. They are designed to conduct similar operations as work class ROVs but at intermediate depth, around 1,000 meters. They are usually all-electric vehicles, except from their manipulators which can by hydraulically powered. Both work class and mid-sized ROVs need to be deployed from large vessels, with specific launch and recovery systems (LARS). Finally, observation class ROVs range from 100 kg to hand-carryable robots. They operate at depth below 300 m and are fully electrically powered. Their missions include observation and inspection tasks for various purposes including offshore industry and archaeological or oceanographic research, but also dexterous tasks like biological sampling or archaeological excavation.

The design of the cable depends on the ROV class and the intended application. The cables of work class and mid-sized ROVs are designed to provide them a significant amount of energy over distances of several hundred meters, and are thus very long, heavy cables, with negative buoyancy. On the other hand, smaller observation class vehicles can embed their batteries, such that their cable only allows communications. Such cables are therefore lighter, with positive or neutral buoyancy, and avoid lifting sediments or getting stuck in reliefs by dragging along the seabed if the robot operates close to it. Nonetheless, ROV's mobility is still limited by their cable, by involving an energy-intensive drag force, mechanical actions and risks of entanglements. Cable management strategies are therefore crucial to limit these effects and allow exploration in cluttered environments such as caves, flooded mines or shipwrecks. Cable management is also critical when the ROV travels between the surface and its working depth, or if multiple ROVs happen to be deployed together.

Figure 2.1 represents some examples of ROVs. Victor 6000 (Nokin, 1998) is a 4.6-ton work class engine developed by the French Marine Institute Ifremer. It has been gradually modernized since its very first dive in 1999 and can operate up to 6,000 m depth, involving an 8,500 m cable. Its missions include seabed mapping, wildlife sampling and water analysis. Ocean One (Khatib et al., 2016) is a 240 kg humanoid archaeologist ROV from Stanford University. It has been designed to conduct fine manipulation in archaeological explorations up to 1,000 m and provide haptic feedback to the operator. The BlueROV2[1] is a hand-carryable 10 kg ROV commercialized by Blue Robotics. It is an affordable, expandable ROV with open-source electronics and software, intended for inspections and research up to 100 m. Last, the ROVINGBAT[2] is a 135 kg maintenance and inspection ROV developed by Exail. It can crawl or be propelled by its thrusters, and operate up to 100 m. While Victor 6000, Ocean One and the ROVINGBAT are supplied in energy *via* their cable, the BlueROV2 embeds battery, and its cable is only intended for communication.



(A) Victor 6000, Ifremer.
Courtesy of Ifremer

(B) Ocean One, Stanford University.
Courtesy of (Khatib et al., 2016)

(C) BlueROV2, Blue Robotics

(D) ROVINGBAT, Exail.
Courtesy of Exail

FIGURE 2.1: Examples of ROVs

---

[1] https://bluerobotics.com/store/rov/bluerov2/

[2] https://www.ecagroup.com/en/solutions/rovingbat-hybrid-rov

#### 2.2.1.2 Autonomous Underwater Vehicles (AUVs)

Autonomous Underwater Vehicles, or AUVs, are not physically linked to any surface station and are used for missions which can be fully automated. Embedded intelligence is nonetheless still unable to cope with some tasks with an acceptable safety level, and AUV applications are thus restricted. Excluded tasks include properly coping with unexpected situations, and tasks requiring high level environment perception ans some safety critical tasks, which might be performed by an artificial intelligence but with a non-acceptable safety level. Typical AUV tasks include seabed mapping, wreck and mine detection, offshore infrastructure inspection and data collection for oceanographic purposes. Some AUVs embed acoustic modems to maintain a communication with the surface, but it is limited in terms of bandwidth and is not real-time. Such an equipment only allows communicating very simple orders and light-weight information about mission progress (Creuze, 2014).

Because they are not physically connected to the surface, multiple AUVs can be deployed quite easily from a single ship, simultaneously. AUV fleets can be used to cover large areas in applications including seabed mapping, inspections or search and rescue missions (Bechlioulis et al., 2019). AUV structure strongly depends on the intended application. Torpedo-shape AUVs are the most common, and their hydrodynamic shape enables them to cover great distances, at depth up to several thousand meters. These AUVs are mainly propelled by three means: propellers or thrusters, jet-pumps (Brizard, 2014) and buoyancy driven systems (Alam et al., 2014). Torpedo-shape AUVs are although predominantly propelled by multiple propellers or thrusters. Rudders or fins can be used for directional control (Sahoo et al., 2019).

Buoyancy driven AUVs are called gliders, and are an important sub-category of AUVs. Gliders only control their buoyancy, pitch and roll. When diving, their vertical displacement generates a forward motion using wings, or hydrofoils. Gliders are used to collect chemical and physical measurements along the water column. They can cover wide areas by gliding up and down, alternating ascent and descent cycles. These missions can reach thousands of kilometers, and last several months (Merci et al., 2023).

Alternatively, some bioinspired AUVs have been developed to mimic aquatic animals, like fishes, snakes, or turtles. Biomimetic AUVs are also being investigated to study aquatic life without disturbing the natural habitats (Sahoo et al., 2019). These robots usually travel in water using undulatory propulsion. While electric motors are commonly used for inducing this motion, some systems rely on hydraulically driven elements (Katzschmann et al., 2018) or piezoelectric fibre composite (Ming et al., 2014).

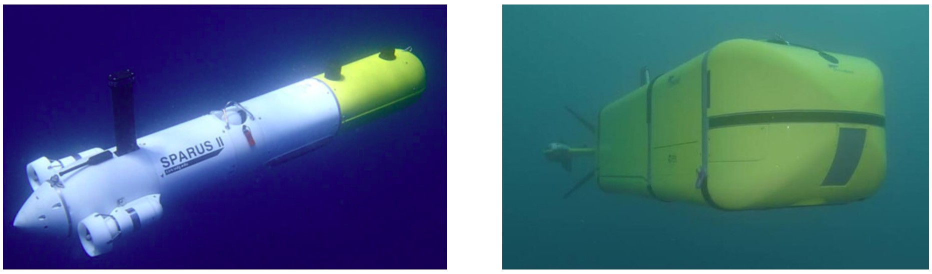
Various examples of AUVs are represented in Figure 2.2. Sparus II (Carreras et al., 2018) is a 52 kg and 1.6 m long torpedo-shaped, multipurpose AUV developed by the University of Girona. It can operate up to 200 m, with an 8 to 10-hour autonomy. UlyX[3] is a 2.7 t and 4.5 m long torpedo-shaped vessel developed by the French Marine Research Institute Ifremer. It has been designed to collect scientific data up to 6,000 meter depth, and has a 48-hour autonomy. SeaExplorer[4] is a 59 kg, 2-m long underwater glider designed and commercialized by Alseamar. It can cover

---

[3] https://www.ifremer.fr/fr/flotte-oceanographique-francaise/decouvrez-les-navires-de-la-flotte-oceanographique-francaise/ulyx

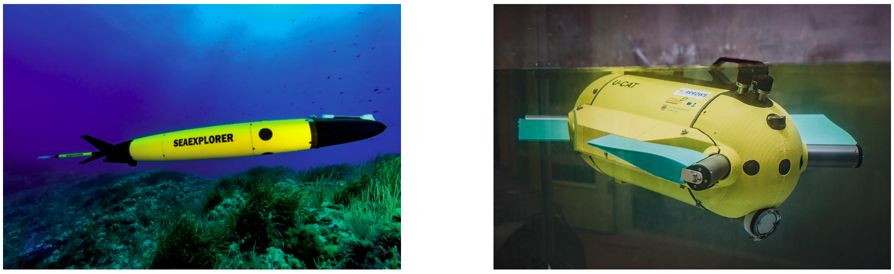
[4] https://www.alseamar-alcen.com/products/underwater-glider/seaexplorer

1,700 km within 110 days, up to 6,000 m depth. It can embed an on-demand sensor payload to carry out scientific measurements. Finally, U-CAT (Salumäe et al., 2014) is an experimental biomimetic turtle AUV from Tallin University of Technology. It has been designed for inspecting the interior of shipwrecks, hence its small size and 4-fin actuation for precise manoeuvering.



(A) Sparus II, University of Girona. Courtesy of (Carreras et al., 2018)

(B) UlyX, Ifremer. Courtesy of Ifremer



(C) SeaExplorer glider, Alseamar. Courtesy of Alseamar

(D) U-CAT, Tallin University of Technology (Taltech). Courtesy of Taltech

FIGURE 2.2: Examples of AUVs

#### 2.2.1.3 Hybrid Vehicles

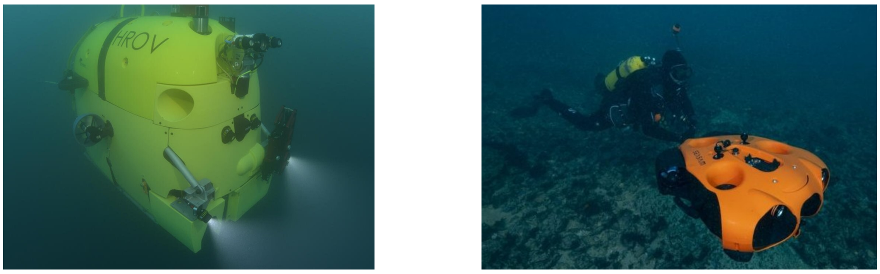
Lastly, some underwater unmanned vehicles are hybrid, and can work as a ROV or as an AUV depending on their operational context. The main motivation in developing such hybrid vehicles (HROV) is to be able to disconnect a ROV from its cable in case of entanglement, too important motion limitation or cable break. The cable is then only a communication medium, and the vehicle embeds its own batteries. An example of such robot is the Ariane HROV from Ifremer (Brignone et al., 2015; Raugel et al., 2019), which is a 1.8 ton vehicle intended for oceanographic research applications up to 2,500 m depth. HROVs have also been introduced for the leisure sector or the diving sector (Ghader et al., 2023): Seasam drone[5] has been initially developed by Delair Marine (ex-Notilo Plus) as a hand-carryable sea diving companion HROV, which can be teleoperated from the surface or used in autonomous mode for specific operations and reach 100 m depth. These vehicles are illustrated in Fig. 2.3.

### 2.2.2 ROV cable management strategies

As mentioned in Section 2.2.1.1, cable management strategies are essential in preventing ROV cable entanglements and limiting their mechanical actions on the vehicle. ROV cable management solutions are divided in two main kinds: passive or

---

[5] https://seasam.notiloplus.com/autonomous-rov-seasam/



(A) Ariane, Ifremer. Courtesy of Ifremer

(B) Seasam, Delair Marine (ex-Notilo Plus). Courtesy of Delair Marine (ex-Notilo Plus)

FIGURE 2.3: Examples of HROVs

active (Christ et al., 2014), and are described in Sections 2.2.2.1 and 2.2.2.2 respectively.

#### 2.2.2.1 Passive strategies

Heave compensators are the most classic passive solution. They consist in a lumped mass fixed on the cable and positioned at the working depth, absorbing the cable drag from the surface vessel, as shown in Figure 2.4. The cable is then composed in two parts: the umbilical is used to reach the working depth from the boat to the heave compensator, while the tether links the ROV to the heave compensator and allow it to operate within a distance of around a few hundred meters from the heave compensator, at working depth. Because the umbilical and the tether are not subject to the same physical constraints, their design is usually different. A solution to reduce cable entanglements when the ROV travels between the surface and the working depth consists in moving the ROV is a cage which is winched from the boat. This cage can be used as a passive heave compensator.

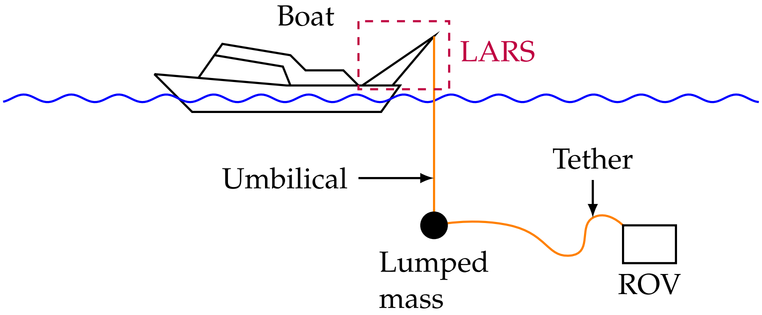


FIGURE 2.4: Passive heave compensator (boat and LARS are optional)

Fixed buoys can also be used to keep a part of the cable at the surface, to enable ROVs to work in more shallow waters (see Figure 2.5). Recent works investigate a new passive cable management strategy by placing freely moving ballasts and buoys on the cable to constrain its shape to taut segments and thus prevent entanglements (Viel, 2022b; Viel, 2022a). Depending on the combination of sliding buoys and ballasts employed, this last approach can prevent the cable from dragging the seafloor, ice caps, or ceilings of marine caves. An example of configuration is displayed in Figure 2.6.

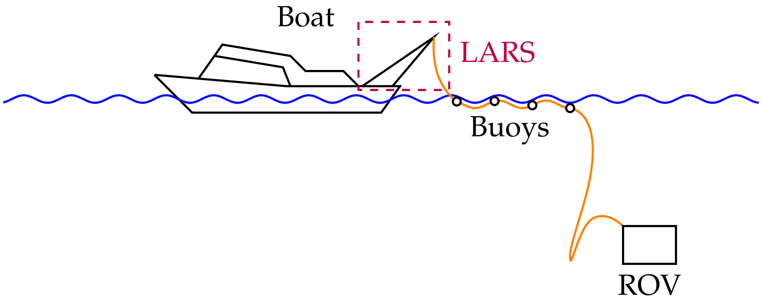


FIGURE 2.5: Fixed buoys (boat and LARS are optional)

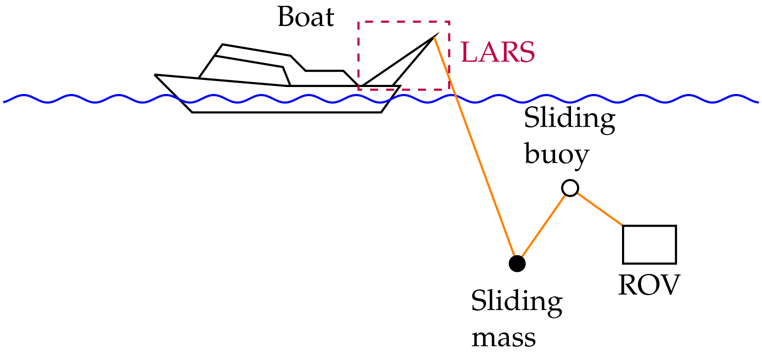


FIGURE 2.6: Sliding ballasts and buoys (boat and LARS are optional)

#### 2.2.2.2 Active strategies

Active strategies allow a finer control of the cable. The most common ones use a system of winches that regulates the length of free tether available to the ROV. These winch systems can be placed on the boat (Figure 2.7) or at the working depth, in association with a lumped mass (Figure 2.8). When this system is positioned at working depth, the cable is divided into an umbilical part and a tether part, as explained in Section 2.2.2.1. If the ROV is deployed from a cage (see Section 2.2.2.1), the winches can be positioned in this cage, at working depth. Winch systems are usually referred to as Tether Management Systems (TMS). Tether length regulation aims at reducing the risk of entanglements, since too long cable are more likely to get tangled, but it can can also compensate heave actively by rolling and unrolling the cable. Some recent works propose to regulate the tether length according to a measure of the tension of the cable (Tortorici et al., 2023).

However, these strategies are not well-suited for the exploration of cluttered environments such as caves, immersed mines or wrecks, which require the cable to be shaped to fit into a constrained free space. This can only be achieved by controlling the global shape of the cable itself by additional robotic devices placed along it, resulting in a robot chain, as represented in Figure 2.9. Control of the cable by a single additional remotely operated vehicle has already been employed during the exploration of the shipwreck *La Lune* by the archaeological robot Ocean One (Khatib et al., 2016) off the coast of Toulon, France, as pictured in Figure 2.10. Ocean One's last 3 cable meters were operated by the Leonard ROV from the LIRMM, University of Montpellier, France, in order to counteract currents and tensions during manipulations requiring high precision and stability. Some recent works investigate the

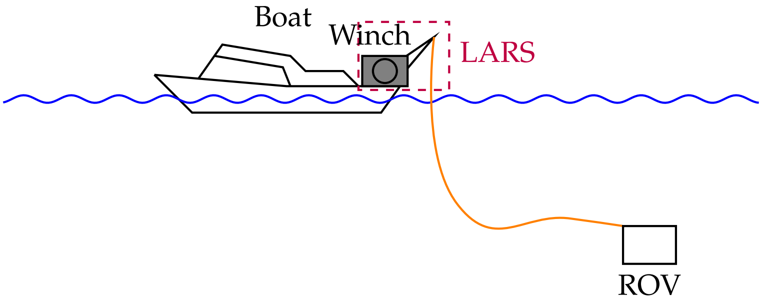


FIGURE 2.7: Surface winch (boat and LARS are optional)

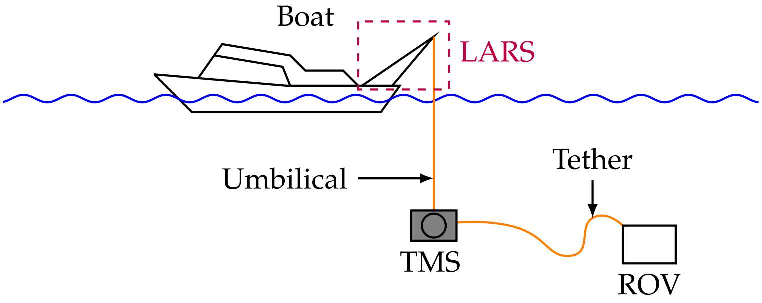


FIGURE 2.8: TMS at working depth (boat and LARS are optional)

automation of cable control in a robot chain configuration, with (Laranjeira et al., 2020) studying the visual servoing of an underwater vehicle to the shape of the cable in the camera image.

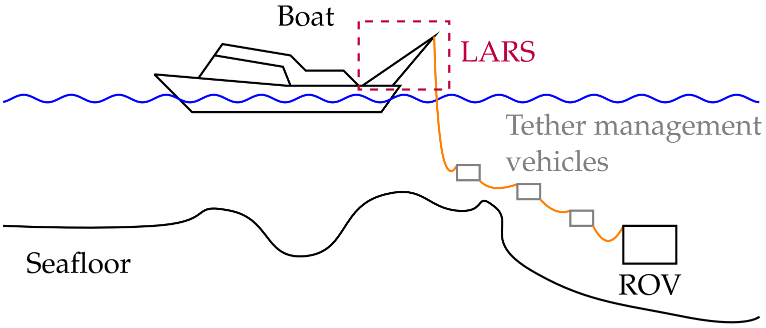


FIGURE 2.9: Robot chain (boat and LARS are optional)

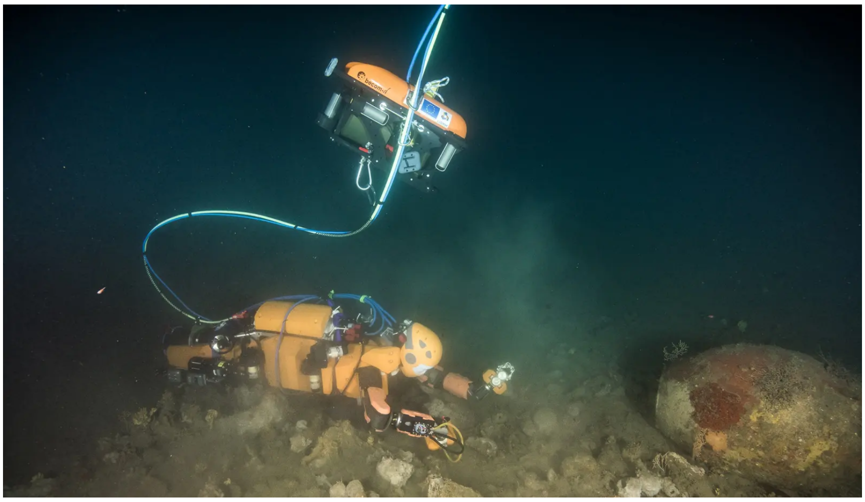


FIGURE 2.10: Ocean One cable being managed by the Leonard ROV on the shipwreck *La Lune*. Courtesy of the LIRMM, 2016.

### Short Stories #2– DRAMATIC ENTANGLEMENTS: *La Lune*'S STERN LANTERN

In 1664, the French warship *La Lune* was sent by the French King Louis XIV to fight the barbary pirates in the Mediterranean sea. On her return, a series of incidents led to more than 3 times the number of passengers on board than the ship could handle. Nonetheless, *La Lune* manages to reach the port of Toulon, but was unfortunately denied access for preventive sanity reasons and the ship was asked to respect a quarantine. On his way off the port, the overweight finally resulted in *La Lune* sinking to the bottom off the coast of Toulon, causing more than 700 deaths among the 1,000 men on board.

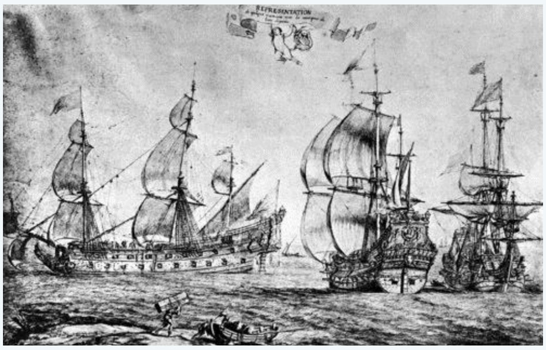


FIGURE 2.11: 1960 painting of *La Lune*. Courtesy of Musée National de la Marine, S. Dondain, France.

The wreck of *La Lune* was discovered in 1993, at 100 m depth. Some parts of the wreck were remarkably well preserved, including its stern lantern, what is extremely rare and precious for archaeological research. Because of the important depth of the wreck, archaeological investigations were performed by ROVs. But in the early 2000s, the incredibly well preserved stern lantern was dramatically damaged by the cable of a ROV exploring the wreck.

### 2.2.3 Perception for UUVs

GNSS positioning is not available to underwater robots because of the absorption of electromagnetic waves in the first centimeters of the water column. In such GNSS-denied environments, robotic systems strongly rely on their embedded sensors to estimate their location. The most common sensors in underwater robotics are acoustic sensors, inertial measurement units (IMUs), pressure sensors and cameras.

#### 2.2.3.1 Acoustic sensors

Acoustic sensors are the only short-to-wide-ranging sensors available underwater. Indeed, while electromagnetic waves are absorbed by the water within short distances, acoustic waves can propagate up to a few kilometers. Underwater acoustic sensors can be classified in 3 categories: positioning sensors, ranging and imaging sensors, or SONARs, and Doppler Velocity Logs (DVLs). These acoustic sensors are active. As a result, acoustic beams can interfere with each other and with their environment. Therefore, the deployment of multiple acoustic transceivers requires beams synchronization, which can increase measurement errors.

**2.2.3.1.1 Positionning sensors** Acoustic positioning sensors are transceivers designed to measure a distance and optionally an angle with respect to other transceivers of known position, allowing to triangulate the position of the mobile system (Cong et al., 2021).

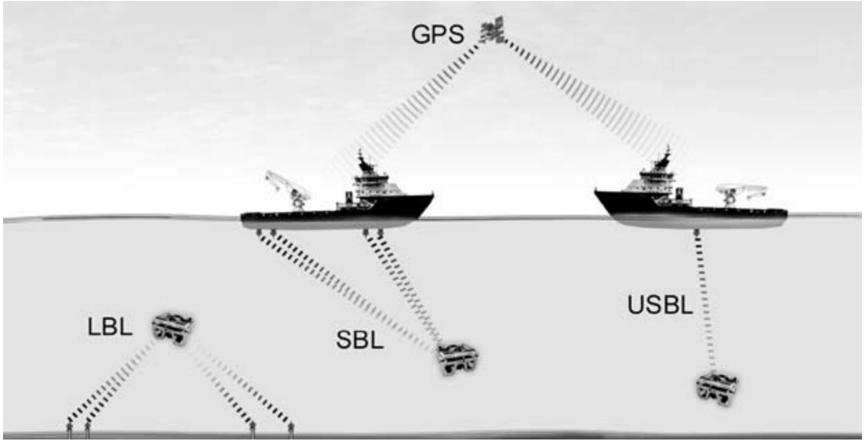


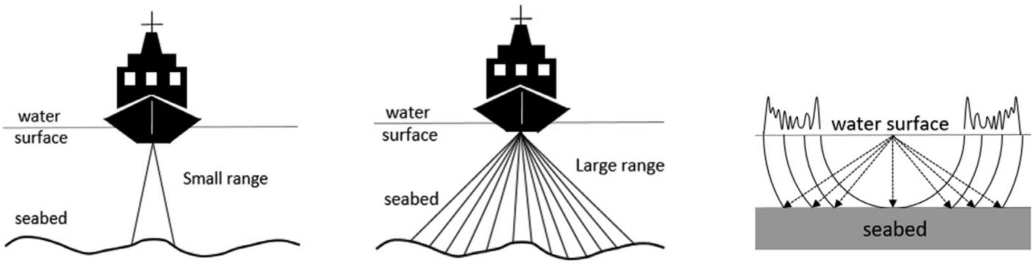
FIGURE 2.12: LBL, SBL and USBL positioning systems. Courtesy of (Mallios et al., 2009)

Ultrashort baseline (USBL) and short baseline (SBL) positioning systems rely on an array of acoustic transducers placed under a ship and a transponder installed in the mobile robot. The pose of the robot can be estimated with respect to the boat by travel time and phase difference. The global position of the vehicle can then be deduced from the GNSS localization of the ship. The distance between the transceivers placed on the boat is less than one meter for USBL, and usually from 20 to 50 meters for SBL. USBL systems have the advantage of smaller size and more simple structure, but are also less accurate, since positioning accuracy increases with the baseline distance (Zhu et al., 2020). Pose estimation accuracy is although strongly affected by ocean environmental parameters and by sizing uncertainties of the transceiver array.

Such systems are therefore highly dependent on an accurate calibration. Pose estimation error increases with the distance between the boat and the vehicle. Though USBL systems usually range down to 10,000 m, their ranging error is of 1 or 2% for most common systems, leading to poor vehicle localization accuracy in deep water (Liu et al., 2020).

Contrary to USBL and SBL, long baseline (LBL) systems use a set of transducers placed on the sea floor, with a baseline of 100 m to 20 km (Cong et al., 2021). These systems are more accurate and allow localization at important depth. However, they are very costly to install and need maintenance operations. LBL, SBL and USBL positioning systems are illustrated in Figure 2.12.

**2.2.3.1.2 SONAR** SONARs (sound navigation ranging systems) measure their distance to solid obstacles based on the time-of-flight of acoustic beams. SONARs mainly include single beam, multi-beam and side-scan SONARs (Cong et al., 2021), which are illustrated in Figure 2.13.



(A) Single-beam SONAR (B) Multi-beam SONAR (C) Side-scan SONAR

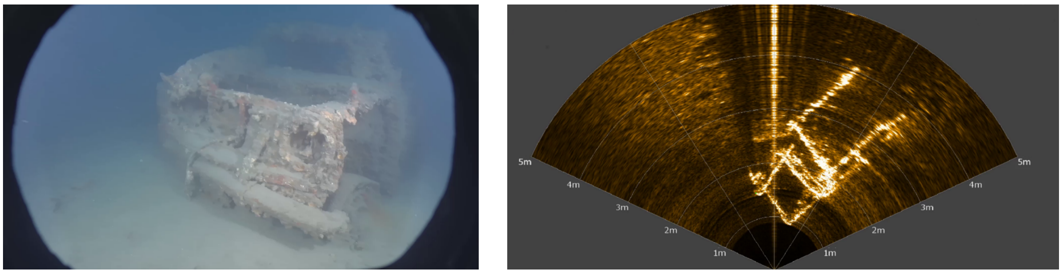
FIGURE 2.13: Typical SONARs. Courtesy of (Cong et al., 2021)

Single-beam SONARs consist of transceivers which emit a beam of short-pulse acoustic signal and receive the signal reflected by the environment. The distance between the system and the closest obstacles along the signal direction can be deduced from the time-of-flight. Signal response analysis can also provide information about the material of the object which reflected the initial signal. Multi-beam SONAR is a combination of multiple single beam SONARs, allowing larger coverage, faster speed and better accuracy.

Side-scan SONARs are designed to create depth images of large areas — usually, the sea floor. They emit acoustic pulses with a very small horizontal angle but a wide vertical beam angle. The smaller the horizontal angle, the more accurate the scanning. Some SONARs directly emit such a fan-shaped beam, while others use a small rotating beam with a small vertical angle.

The output of a SONAR is named a SONAR image and represents a sectional vue of the surroundings, along the axes of acoustic pulse emissions, where edges indicate reflections of acoustic waves at the interface between two media with different physical properties. SONAR images are generally noisy because of reverberation, self interferences and marine environment noise (Yuan et al., 2021). Reverberation consists in multiple reflections from non-target objects, and is particularly important in cluttered environments such as ports or flooded caves and mines.

An example of SONAR image is given in Figure 2.14. This figure is taken from the work of (Pecheux et al., 2023) which focus on the calibration of a SONAR-vision system, and illustrates well the differences in SONAR and optical images of the same object. Multi-beam SONAR and optical images are recorded from the same robot, and both sensors are oriented towards the wreck. However, while the optical images is a frontal view of the wreck, the SONAR outputs a sectional view of the submerged car where one can recognize the engine and passenger compartments.



(A) Optical image (B) Multi-beam SONAR image

FIGURE 2.14: Multi-beam SONAR and optical image recorded with both sensors facing a car wreck. Courtesy of (Pecheux et al., 2023)

**2.2.3.1.3 Doppler Velocity Log (DVL)** DVLs measure their local, linear velocity with respect to a reference surrounding surface — usually, the sea floor. It is composed of four transducers which emit acoustic beams to the seafloor. The velocity of the vehicle is deduced from the reflected beam using the Doppler effect and the frequency shift between the transmitted and received signals. DVLs are usually able to measure their linear velocity with an accuracy of about 0.2% of the actual velocity (Cohen et al., 2022). Their main limitation is that they require the vehicle to be close enough to their reference surface. The maximum allowed distance is about a few dozen meters, depending on the system.

While DVLs are most commonly used for measuring a velocity with respect to the sea floor, some applications involve different referent reference surfaces. A example can be found in robotic ship hull inspection, where a closed-loop control based on DVL velocity measurements with respect to the hull can be implemented, as in the ship hull inspection service of Delair Marine (ex-Notilo Plus)[6].

### 2.2.3.2 Inertial Measurement Unit (IMU)

IMUs measure linear accelerations and angular velocities in 3-dimensions. They often integrate a magnetometer and a filter in order to estimate a 3D orientation relative to gravity vector and magnetic North. Such systems are also known as attitude and heading reference systems (AHRS). AHRS filters are generally Kalman-based (Creuze, 2014). Although Kalman-based filters give the most accurate orientation estimation, they demand high sampling rates and need large state vector representations to describe the three-dimensional rotational kinematics, as well as problem linearization. As a result, other filters are sometimes preferred to provide a lighter computation load or allow lower sampling rates. Complementary filter and Madgwick filter or examples of such alternative options (Madgwick et al., 2011).

---

[6] https://seasam.notiloplus.com/rov-inspection-of-ship-hull-at-anchorage/

IMUs can be classified in two categories (Creuze, 2014). On the one hand, fiber optic gyroscope (FOG) IMUs have extremely small measurement bias. Their measurements can therefore be integrated into a velocity and a position. Such IMUs are called inertial navigation systems (INS) and are particularly expensive, about several thousands euros. However, their velocity and positioning error is not bounded, and they are therefore commonly associated with other sensors, usually a DVL, in order to bound these errors. On the other hand, micro electro-mechanical IMUs involve significantly higher bias and cost only a few euros. Although their measurements are not accurate enough to estimate a position, they can still output an accurate orientation estimation.

If no magnetometer is available, the horizontal component of the estimated orientation can drift over time. This drift is commonly approximated by an affine additive offset whose parameters need to be calibrated.

#### 2.2.3.3 Pressure sensors

Pressure sensors are commonly used to measure robot depth, which is an affine function of water pressure. Note that depth is measured with respect to the surface and not with respect to the mean sea level, and is affected by tides. By abuse of language, water pressure sensors are also denoted as depthmeters.

At small depth, such pressure sensors give a very accurate depth measure, with a measurement resolution of a few millimeters. However, the deeper the measurement the coarser the accuracy. First, pressure sensors designed for very important depth need to resist very high pressured, but see their precision decrease significantly because of this design (Creuze, 2014). An example can be given from the pressure sensors series commercialized by BlueRobotics[7], where the Bar02 sensor has a maximum depth of 10 m and a 0.16 mm accuracy, the Bar30 has a maximum depth of 300 m but a reduced 2 mm accuracy and the Bar100 has a maximum depth of 1000 m but a coarser 3 cm accuracy. Second, pressure measurements at high depth are limited by the resolution of the sensor, which is defined by the smallest degree of pressure change that can be detected. Consequently, for a given sensor, deeper measurements will necessarily decrease the accuracy of the reported depth. For instance, Athen's 8000 Series Ultra High-Precision Pressure Sensor[8] can measure water pressure up to 8,000 m depth with a 0.01% full scale resolution, involving 80 m accuracy range at 8,000 m, since depth is proportional to water pressure.

#### 2.2.3.4 Cameras

Although electromagnetic waves are absorbed by the water within a short distance, optical sensors are still commonly used for small range underwater perception. While some works investigate active, underwater time-of-flight optical sensors for depth perception (Massot-Campos et al., 2015; Digumarti et al., 2016), cameras are the most common underwater optical sensor, and most underwater vehicles embed one or multiple color or grayscale cameras. Cameras are indeed cheap and small-sized sensors providing a rich feedback, which can be interpreted directly by a human operator. Cameras being passive sensors is also an advantage compared

---

[7] https://bluerobotics.com/store/sensors-cameras/sensors/bar30-sensor-r1/

[8] https://www.althensensors.com/sensors/pressure-sensors/ultra-high-precision-pressure-sensors/8000-series-ultra-high-precision-pressure-sensor/

to usual acoustic sensors, since multiple cameras will not interfere with each other nor with their surroundings.

Underwater vision is however challenging, due to the effect of water on visual perception. First, camera protection from the water usually implies an external housing. Light rays are therefore deflected from the refraction at diopters between water, the transparent material of the housing, and the air inside it. Second, light propagation through the water is affected by backscattering, selective color absorption and turbidity. These phenomena are illustrated in Figure 2.15, which is inspired from (Wang et al., 2019). Finally, embedded light affect the aspect of the scene in deep sea missions.

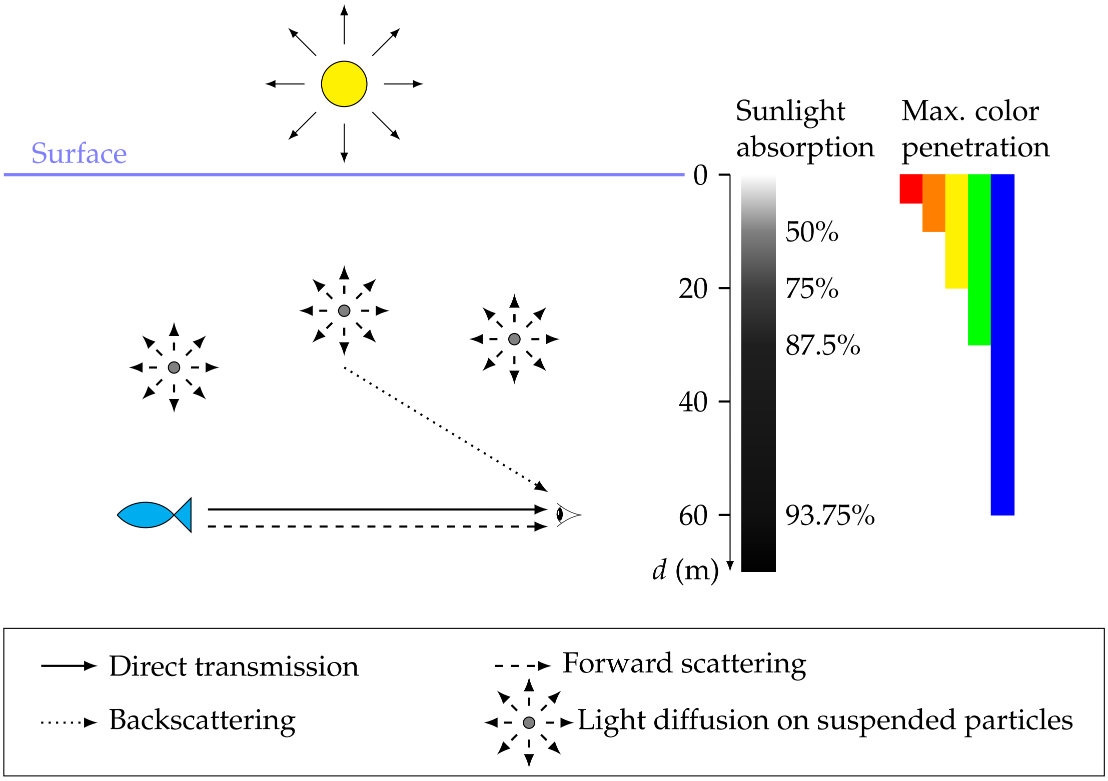


FIGURE 2.15: Underwater visual conditions and related physical phenomena

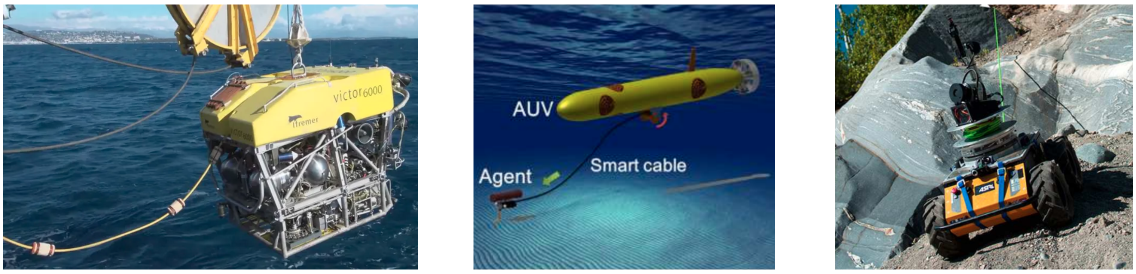
Correcting underwater images in order to remove the visual distortions caused by the water is called underwater image restoration. Underwater image restoration works can be classified between model-based ones, which rely on a physical model of underwater vision (Akkaynak et al., 2019; Boittiaux et al., 2023), and model-agnostic ones, which can consist in classical image enhancement techniques assumed to correct the effect of water but without any formal certainty (Torres-Méndez et al., 2005), and deep learning techniques (Islam et al., 2020).

## 2.3 Cable-based state estimation for tethered vehicles

Tethered robots can be found in many applications, including marine, terrestrial and aerial robotics. State estimation of such systems is necessary for their deployment and control. Tethered robots state include both cable and robot localization in the operational space. Cable shape estimation can be performed according to different models and sensors. In addition, cable three-dimensional shape encodes the position of the robot, since the robot is located at one of the cable's end points. It is therefore also investigated for robot localization purpose. Note that, in this section, the words 'cable' and 'tether' are used as synonyms.

### 2.3.1 Tethered robotic applications

Tethered robots are involved in marine, terrestrial, aerial and hybrid robotics. Figure 2.16 shows some examples of tethered robotic systems.



(A) ROV tethered to a surface vessel. Courtesy of Ifremer

(B) Agent tethered to an AUV. Courtesy of (Yu et al., 2013)

(C) Tethered Robotic Explorer (TRex). Courtesy of (McGarey et al., 2017)

(D) Flying drones transporting a cable. Courtesy of (D'Antonio et al., 2021)

(E) ROV tethered to an USV. Courtesy of IMODCO

(F) Aerial-underwater hybrid system. Courtesy of (Debruyn et al., 2020)

FIGURE 2.16: Examples of tethered robotic systems

In the field of marine robotics, tethers provide real-time communication and, sometimes, energy to underwater vehicles. The most common operational scenario consists in a ROV tethered to a control station in a surface vessel (Christ et al., 2014). Other underwater tethered connection schemes are also in used, or investigated. A ROV can be tethered to an unmanned surface vehicle (USV) instead of a boat (Tortorici et al., 2019). Such scheme can be interesting in automating ROV deployment with the USV acting as a relay station for wireless communication to a further control station. In addition, the USV can be designed to be small and maneuverable enough to facilitate ROV navigation by allowing access to shallow water and moving around to extend its operational area. These USV/ROV systems are currently being investigated for offshore inspection tasks[9]. Alternatively, a ROV can be tethered to a smaller surface GPS and WiFi relay station which provides georeferencing and teleoperation from a distant surface location. An example of such device is the

---

[9] https://www.hydro-international.com/content/news/fully-remote-rov-inspection-of-offshore-wind-farm-completed-by-fugro-s-blue-essence

Seasam navigator of Delair Marine (ex-Notilo Plus)[10]. A tether can also connect underwater vehicles together. The work of (Yu et al., 2013) considers the deployment of a small tethered vehicle from a large, difficult-to-manoeuver AUV, when manipulation or precise monitoring is necessary. The cable offers a connection to the AUV central computation unit and supplies power to the agent. Recent works investigate ROV tether management by controlling the cable shape using intermediary devices placed along it. Such configuration has been investigated using passive buoys and ballasts which can slide freely to their lowest potential energy position along the cable (Viel, 2022a; Viel, 2022b), or using intermediary, active, robotic agents connected pairwise up to a surface station, forming a robot chain (Laranjeira et al., 2020).

In aerial and terrestrial robotics, robots transporting or manipulating deformable linear objects are another kind of tethered systems (Alonso-Mora et al., 2015; D'Antonio et al., 2021). A stretched tether can allow a terrestrial robot to explore a steep, rugged, and dangerous terrain by clinging it to a fixed support (McGarey et al., 2016). Finally, a tether can connect parts of a robotic hybrid system, providing communication, power, and the insurance to keep the parts of the system together (Debruyn et al., 2020). This last work presents an aerial/underwater hybrid system, which consists in an aerial vehicle which can land at the water surface and deploy a very small ROV. The two parts of the vehicle are connected by a communication and power supply cable.

### 2.3.2 Cable modeling

Cable models can be classified in two main categories: dynamic models and quasi-static model. While dynamic models are the most complete ones, and are derived from the forces that apply to the cable, given its physical properties, simplified, quasi-static models are usually preferred for real-life, online cable shape estimation. Dynamic and quasi-static models are described in Sections 2.3.2.1 and 2.3.2.2 respectively.

#### 2.3.2.1 Dynamic models

A physical cable model can be derived from the dynamics of the cable and the forces that apply to it. Cables are subject to external forces, which are exerted on the cable by its environment, and internal forces caused by cable stiffness, damping and elasticity. External forces exerted on underwater cables include its buoyancy, its weight, drag forces, the mechanical action of currents, cable inertia, as well as the tension at cable attachment points. Internal forces include flexion, torsion and elasticity (Tortorici, 2021).

Lumped-mass-spring and finite-element models are the most common approaches in cable dynamics modeling. Lumped-mass-spring methods model a cable as mass points joined together by massless elastic elements (Buckham, 1997; Soylu et al., 2010; Hong et al., 2020), as illustrated in Figure 2.17. Alternatively, finite-element-based methods describe a cable as a continuous system and solve the resulting partial differential equations (Eidsvik et al., 2016; Meng et al., 2020).

Dynamic models are commonly used for cable simulation. For instance, cables are modeled using a lumped-mass-spring method in the Vortex® mechanical

---

[10] https://seasam.notiloplus.com/rov-gps-seasam-navigator/

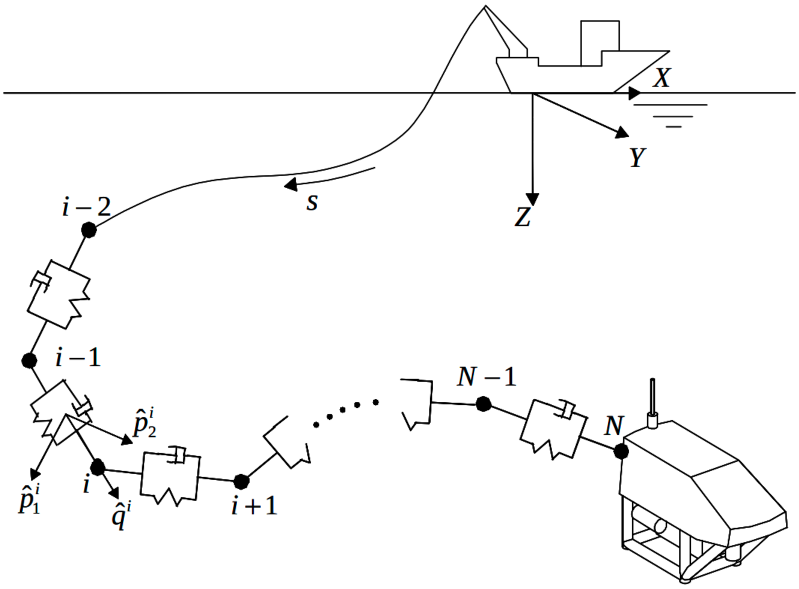


FIGURE 2.17: Lumped-mass-spring modeling of a ROV cable. Courtesy of (Buckham, 1997)

simulator from CM Labs[11], as well as in the mooring simulator MoorDyn[12] (Hall, 2020). In the open-source physics-based simulation platform SOFA[13], cables dynamics modeling is based on a different model known as discrete Cosserat model (Renda et al., 2018), which models linear structures with a series of rigid solids whose relative position is defined by a strain state[14].

Because dynamic models can be computationally expensive and may require detailed knowledge of the environment which is not always available in real-life applications, simplified models are often preferred for real-life, real-time applications.

#### 2.3.2.2 Quasi-static models

Many works rely on a quasi-static model, involving less parameters. The most widely used model for hanging cables is the catenary model (Laranjeira et al., 2020; D'Antonio et al., 2021), which is a parametric curve defined as the shape of an idealized homogeneous hanging cable with fixed length and fixed ends, only subject to its own weight. This definition can be extended to any such cable submitted to a vertical force proportional to mass, like a cable in a liquid, also submitted to Archimedes buoyancy. The quasi static shape of a non-elastic cable is sometimes approximated geometrically by a parabolic curve (Smolentsev et al., 2023). Figure 2.18 represents the catenary and parabolic curves between the same end points, with same curve length.

---

[11] https://www.cm-labs.com/en/vortex-studio/

[12] https://github.com/FloatingArrayDesign/MoorDyn

[13] https://www.sofa-framework.org/about/story/

[14] https://www.sofa-framework.org/applications/plugins/cosserat-beam-cable-needle/

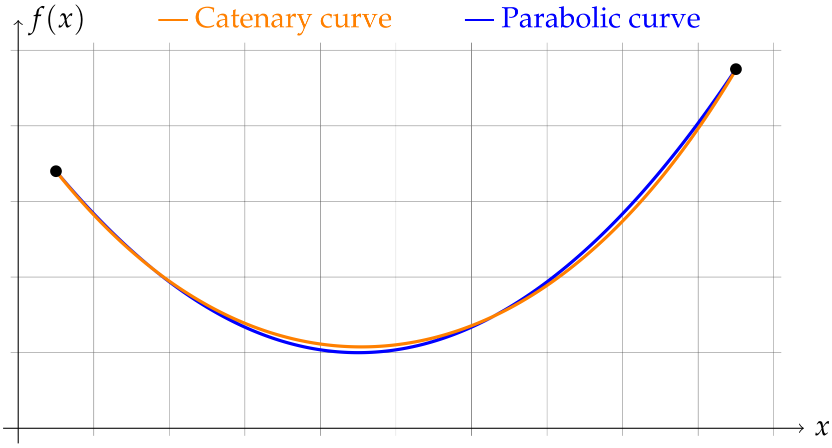


FIGURE 2.18: Catenary and parabolic curves

### Short Stories #3– GAUDÍ AND THE CATENARY

In architecture, a catenary arch is a type of arch which follows an inverted catenary curve. Symmetrically to the catenary curve, this arch is the only physical shape from which an arch of uniform density and thickness can withstand the weight of its material without collapsing. The architecture of Gaudí's Sagrada Familia (Barcelona, Spain) is completely based on catenary arches. The plans for the basilica were drawn up using suspended wires equipped with regularly spaced small weights in order to model an homogenous weighting cable. The resulting catenary curves were then inverted to create the basilica as we know it today.

FIGURE 2.19: Example of Gaudí's hanging chain model. Courtesy of (Dragicevic et al., 2012).

A quasi-static model can also be derived for elastic cables (Wu et al., 2021). While the quasi-static approximation is well suited for aerial systems at low speed, underwater scenarios involve hydrodynamic forces, and optionally waves and currents.

Quasi-static modeling is nonetheless a reasonable assumption for modeling heavy enough, negatively buoyant, underwater, hanging cables with moving ends, in the absence of waves and currents (Laranjeira et al., 2020).

If taut, the cable can be simply modeled as straight. This model applies to the stretched anchoring cables of some terrestrial robots (McGarey et al., 2017). A cable can also be constrained to adopt a piecewise linear shape in order to simplify its modeling. In (Viel, 2022b), the cable of a ROV is constrained into linear segments by the addition of sliding buoys and ballasts.

### 2.3.3 Cable perception for shape estimation

Cables can be designed as proprioceptive sensors in order to compute their own shape. Deformation of an optical fiber cable can be estimated based on interferometry techniques (Duncan et al., 2007; Yu et al., 2013). IMU-coated cables can also estimate their shape from IMU measurement integration (Frank et al., 2013). While such cable can estimate any shape, their use is limited. Optical fiber cables are very expensive, and IMU-coated cables cannot be handled properly by winches because of the protrusions the IMUs make on the cable. In addition, both are very specific and uncommon in real-life systems, and in both solutions the longer the cable, the greater the shape estimation error.

**Short Stories #4– OPTICAL FIBER CABLES AND OCEANOGRAPHY**

Around 1.2 million kilometers of fiber optic communication cables run along the seabed. The deformation of these cables can be measured using interferometric techniques that exploit the presence of small impurities within the optical fiber. It is therefore possible to turn this cable network into a large-scale distributed sensor in order to measure many physical phenomena on a large scale, such as the Earth's tectonic activity, swell and underwater noise. Such seafloor fiber optic sensing is the focus of recent research works (Sladen et al., 2019; Cheng et al., 2021).

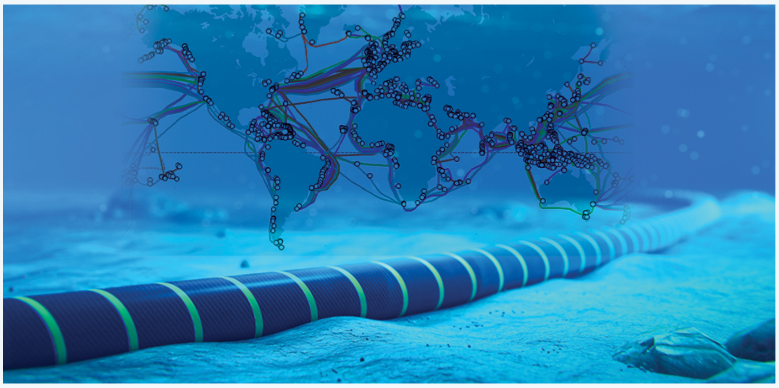


FIGURE 2.20: Underwater communication cable network and cable artist's view. Courtesy of the International society for optics and photonics

Alternative, less invasive, cable instrumentation solutions can be preferred. External sensors can be positioned on an existing cable to compute local orientation or curvature measurements. In (Merlet, 2018), an accelerometer is placed on a taut cable in order to estimate its 3D orientation. In (Tortorici et al., 2023), a curvature sensor is positioned along the cable of a ROV to characterize cable tension. Cable angle and tension can also be measured at its free length start or end point. In (McGarey et al., 2017), these measurements are used to detect entanglements and estimate the global 2D shape of the stretched, anchoring cable of a terrestrial robot. If the cable is managed with a winch, the length of the unfolded cable can also be measured and used for cable shape estimation purpose (Murtra et al., 2013).

Other solutions do not instrument the tether itself, and use external sensor feedback. Cameras and, in the air, RGB-D sensors are widely used for this purpose. The projection of the cable in the image is often segmented based on a color filter (Laranjeira et al., 2017; Laranjeira et al., 2020; Wu et al., 2020; Zhu et al., 2021). The cable must be a distinctive color from everything else around it, which can be difficult to ensure in an uncontrolled environment. In addition, this strategy must be limited in range for underwater applications, because of color absorption by the water (Laranjeira et al., 2020). Sim-to-real cable detection learning strategies are proposed to track 2D Bézier curves or splines (Sundaresan et al., 2020; Yan et al., 2020). RBG-D perception is commonly used for aerial cable manipulation. In (Jin et al., 2022), the two-dimensional state of a cable laying on a plane is estimated from color-based segmentation and RGB-D perception. Visual and depth perception can also be combined with a model for cable state estimation. In (Laranjeira et al., 2020), a colored underwater tether is segmented in RGB images using a color threshold, and a catenary curve is fitted on the resulting points in order to estimate the three dimensional shape of the tether. Similarly, in (Smolentsev et al., 2023), a parabolic model is fitted on a 3D point-cloud measured by a RGB-D camera for an aerial application.

### 2.3.4 Cable-based robot localization

Recently, a few works have been starting investigating cable shape estimation as a solution for tethered robots localization. In (Yu et al., 2013), a small underwater vehicle tethered to an AUV by a 1 meter optical fiber cable is located using the measure of cable deformation. The cable used was commercialized by Measurand[15] under the name ShapeTape™ and was reported by the constructor to exhibit approximately 5% position error with regards to its length, leading to 5 cm for a 1 m cable. This product is however discontinued. The small vehicle and cable used in this work are pictured in Figure 2.21, where the cable is in blue.

In a terrestrial application, (McGarey et al., 2017) take advantage of cable entanglements to estimate obstacles and robot locations in two-dimensions. The major originality of this work is to use entanglements as a source of information on system and scene geometry instead of preventing them. Cable tension and local orientation at its attachment point on the robot are measured using embedded tension and angle sensors. These measurements are used to detect new entanglements and update robot location and obstacle map, in combination with the knowledge of the cable length unwound from the robot. This two dimensional localization and mapping framework is called T-SLAM. Evaluations are conducted with a TRex robot (see Figure 2.16c) with a tether of length varying from 1 to 37 m along the experiments as robot's cable winds in and out. A robot position error between 0 and 2 m along the experiments is reported. An example of obstacles settings and robot trajectory along one of their test sequences named 'Telephone Cord' is shown in Figure 2.22.

---

[15] https://measurand.com/

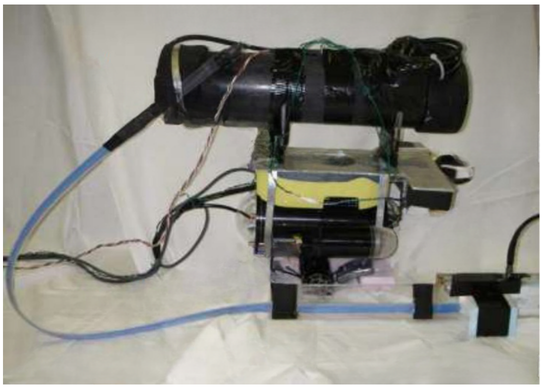


FIGURE 2.21: Small underwater vehicle and ShapeTape™ cable. Courtesy of (Yu et al., 2013)

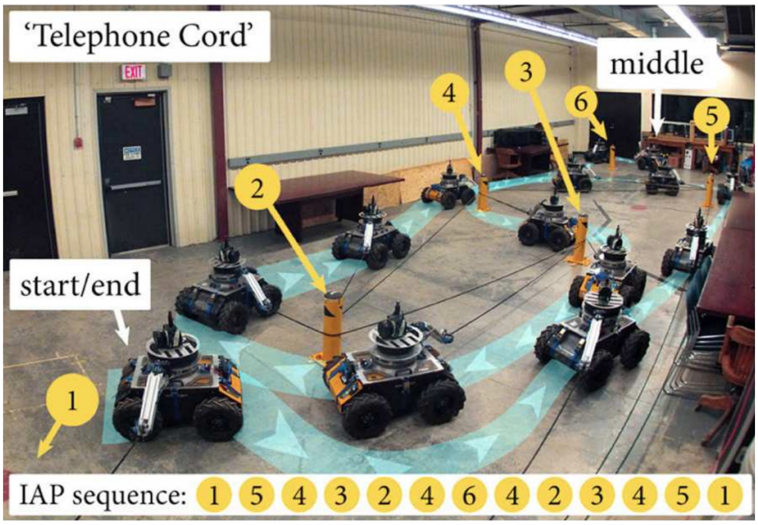


FIGURE 2.22: Example of obstacles settings and robot trajectory for T-SLAM validation. Courtesy of (McGarey et al., 2017)

## 2.4 Monocular visual SLAM for underwater

Although the three-dimensional state of an underwater robot chain may be estimated using proprioceptive, cable-based strategies as described in Section 2.3, controlling this system in cluttered environments requires localization with respect to the surrounding obstacles. In other words, the system must be located with respect to a three-dimensional map of its environment. In addition, this map may be unknown, and the system should therefore be able to map its environment online. The problem of online, simultaneous localization and mapping using embedded sensors is known as SLAM.

Underwater SLAM algorithms usually rely on multi-sensor fusion. An overview of the state-of-the-art underwater SLAM sensor sets is given in Table 2.1. Because cameras, IMUs and pressure sensors can be low cost, lightweight and low-power sensors, more especially when compared to acoustic sensors, they are widely used in underwater perception. Therefore, even the smallest and cheapest underwater robots embed at least a pressure sensor, an IMU and a monocular camera.

TABLE 2.1: Sensors used in underwater specific SLAM

| | **Camera** | **SONAR** | **IMU** | **Pressure** | **DVL** |
|---|---|---|---|---|---|
| (Bellavia et al., 2015) | stereo | ∅ | ∅ | ∅ | ∅ |
| (Zhang et al., 2018) | stereo | ∅ | ∅ | ∅ | ∅ |
| (Ferrera, 2019) | mono | ∅ | ✓ | ✓ | ∅ |
| (Mallios et al., 2016) | ∅ | ✓ | ∅ | ∅ | ∅ |
| (Pairet et al., 2022) | ∅ | ✓ | ∅ | ∅ | ∅ |
| (Rahman et al., 2019) | stereo | ✓ | ✓ | ✓ | ∅ |
| (Silveira et al., 2015) | mono | ✓ | ✓ | ∅ | ✓ |
| (Vargas et al., 2021) | stereo | ∅ | ✓ | ✓ | ✓ |
| (Xu et al., 2021) | stereo | ∅ | ✓ | ✓ | ✓ |

Although underwater visual conditions are challenging (see Section 2.2.3.4), cameras still provide rich perception of their environment, to the point that it is a key feedback sensor for underwater robot teleoperation. Visual SLAM (VSLAM) may thus be a solution for underwater localization and online mapping. Because the smallest and cheapest underwater robots embed a single camera, a more specific focus should be made on monocular VSLAM.

A first mention of visual-based SLAM for underwater applications can be found in (Eustice et al., 2005; Eustice et al., 2006), where the wreck of the Titanic was mapped using a visual-based SLAM information filter, combined with navigation measurements involving a tilt sensor, a magnetometer, a DVL, and pressure and altitude sensors. In line with this first work, many more recent, underwater visual-based SLAM algorithms actually rely on additional navigation sensors (Zhang et al., 2022) and are therefore not purely visual works, such that they will be considered out of the scope of the VSLAM definition used in the current section.

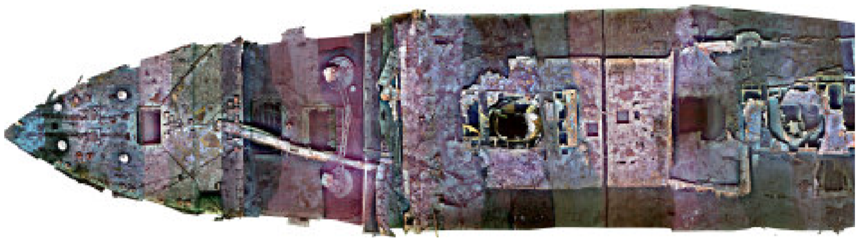


FIGURE 2.23: Visual-based mapping of the Titanic realized in 2004. Courtesy of (Eustice et al., 2005)

VSLAM is widely investigated for airborne applications but fewer works focus on underwater VSLAM. As explained in Section 2.2.3.4, underwater visual perception is affected by several physical phenomena such that the image of an underwater three-dimensional point depends on its viewing angle, distance, depth with respect to the surface, but also on the possible presence of embedded lights. Coping with this photometric distortion is therefore crucial in applying VSLAM techniques underwater. While some works investigate an image restoration step for each incoming frame (Salvi et al., 2008; Cho et al., 2018), this operation is always an approximation and can be computationally expensive depending on model physical accuracy. Therefore, an interest has been shown for off-the-shelf, underwater robust algorithms. Evaluations of state-of-the-art VSLAM algorithms on underwater scenarios demonstrate that while some stereo VSLAM works are robust to underwater visual conditions, monocular ones are still limited (Quattrini Li et al., 2017; Joshi et al., 2019). Two monocular works are nonetheless reported as able to give partial but promising results on underwater scenarios: ORB-SLAM (Mur-Artal et al., 2015) and DSO (Engel et al., 2018). It is worth noticing that DSO is not rigorously a VSLAM algorithm but a *visual odometry* (VO), where difference between both is that VOs only use information from the last few seconds in their computations.

The current section focuses on monocular VSLAM for underwater application. In line with observations from previous studies (Quattrini Li et al., 2017; Hidalgo et al., 2018; Joshi et al., 2019), the focus is made on monocular, ORB-SLAM and DSO related works. The current section is structured as follows. An overview of ORB-SLAM and DSO related algorithms is given in Section 2.4.1. Then, the problem of monocular VSLAM for underwater is developed functionality by functionality. Section 2.4.2 describes the pose tracking problem and how it is solved in DSO and ORB-SLAM-based works. Sections 2.4.3, 2.4.4 and 2.4.5 do the same for local mapping, loop closing and SLAM recovery functionalities. Section 2.4.6 synthesizes the computational workflows of the VSLAM algorithms under focus. Because evaluation datasets are crucial in VSLAM validation, a review of opensource underwater VSLAM evaluation datasets is given in Section 2.4.7. Finally, Section 2.4.8 discusses underwater monocular VSLAM challenges and perspectives.

### 2.4.1 Algorithms overview

This sections provides an overview of the algorithms under focus. Section 2.4.1.1 gives a brief introduction to KeyFrame-based SLAM, which includes ORB-SLAM and DSO-based works. Section 2.4.1.2 lists and explains the main functionalities involved in ORB-SLAM and DSO-based works. A first classification of these works is provided given the functionalities implemented. Finally, Section 2.4.1.3 focuses on VSLAM works classification according to visual information used.

#### 2.4.1.1 From filter-based to KeyFrame-based SLAM

Monocular VSLAM was first solved by seminal work Mono-SLAM (Davison et al., 2007) using an extended Kalman filter (EKF). Map feature locations and camera pose were estimated jointly at each incoming frame, in a single iteration. Filter-based VSLAM, however, involved unnecessary high computational cost when processing new frames with little new information, but also a significantly increasing frame processing duration with the number of map features, as well as the accumulation of linearization errors (Mur-Artal et al., 2015; Zou et al., 2019).

KeyFrame-based approaches were introduced in order to avoid these drawbacks by separating camera pose estimation and mapping in two different tasks, where pose estimation is performed for each incoming frame, and map features triangulation and refinement operations are performed only on a subsample of frames, denoted KeyFrames (KF). In KF-based works, mapping is not bound to framerate, and more costly mapping operations can therefore be conducted. A representative early KF-based work is PTAM (Klein et al., 2007), which splits tracking and mapping tasks on two parallel threads. In practice, KFs can be defined as a data structure created from a frame and composed of a viewing position and orientation, or pose, and of visual information including map features observations. KF representation can be slightly different from a work to another. At the time of writing, state-of-the-art VSLAM works are KF-based, including ORB-SLAM (Mur-Artal et al., 2015) and DSO (Engel et al., 2018).

#### 2.4.1.2 Main functionalities and functional classification

ORB-SLAM (Mur-Artal et al., 2015) and DSO (Engel et al., 2018) solve the monocular localization and mapping problem very differently but are still based on a quite similar architecture which decomposes the VSLAM problem into four functionalities:

- *pose tracking* is the computation of the camera pose from its current image view of the scene;
- *local mapping* consists in creating and completing a map sequentially with local, three-dimensional information which can be deduced from the camera views;
- the *loop closing* problem involves recognizing a previously visited place and update the map accordingly;
- *SLAM recovery* consists in handling tracking failures in order to keep computing a localization and building a map.

While solving the pose tracking and local mapping is part of the definition of VSLAM, not all algorithms implement loop closing and SLAM recovery functionalities.

ORB-SLAM (Mur-Artal et al., 2015) implements pose tracking, local mapping, loop closing and a SLAM recovery strategy. However, underwater evaluations demonstrate that one of the main limitations of ORB-SLAM in underwater environments relies in its recovery module being sometimes inefficient. Consequently, the current state-of-the-art focuses on ORB-SLAM-based works which aim at improving its SLAM recovery capability, namely ORB-SLAMM (Daoud et al., 2018), ORB-SLAM Atlas (Elvira et al., 2019) and Dual-SLAM (Huang et al., 2020).

Conversely, DSO (Engel et al., 2018) is a VO and therefore only implements localization and short term mapping, and does not include loop closing nor SLAM recovery. Underwater evaluations of DSO report that its main drawback relies in the absence of long-term data association, which can lead to map inconsistency and to a lack of robustness to bad data association. These problems can cause critical SLAM failure, from which the system cannot recover since it does not implement SLAM recovery. Whereas no work addresses the problem of extending DSO with a SLAM recovery strategy, recent works extend it with loop closing, namely LDSO (Gao et al., 2018) and DSM (Zubizarreta et al., 2020), where LDSO's loop closing strategy is inspired by ORB-SLAM.

The implementation of loop closing and SLAM recovery functionalities in the ORB-SLAM and DSO-based works under focus is recapped in Table 2.2. The ORB-SLAM and DSO families are represented schematically in Figure 2.24.

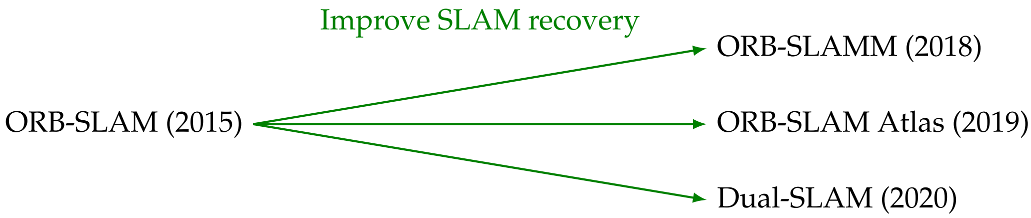
#### 2.4.1.3 Visual information-based classification

VSLAM works can be classified according to the visual information used at each step of the VSLAM pipeline.

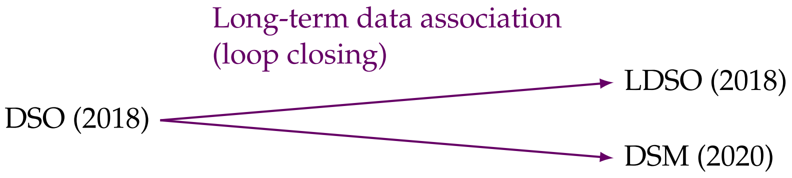
A first criteria consists in the selection of the pixels used. If dense, all pixels from the image are used. If sparse, only a subsample of pixels are used. These pixels can

TABLE 2.2: Functional classification of VSLAM works under focus

| | **Loop closing** | **SLAM recovery** |
|---|---|---|
| ORB-SLAM (Mur-Artal et al., 2015) | ✓ | ✓ |
| ORB-SLAMM (Daoud et al., 2018) | ✓ | ✓ |
| ORB-SLAM Atlas (Elvira et al., 2019) | ✓ | ✓ |
| Dual-SLAM (Huang et al., 2020) | ✓ | ✓ |
| DSO (Engel et al., 2018) | ∅ | ∅ |
| LDSO (Gao et al., 2018) | ✓ | ∅ |
| DSM (Zubizarreta et al., 2020) | ✓ | ∅ |



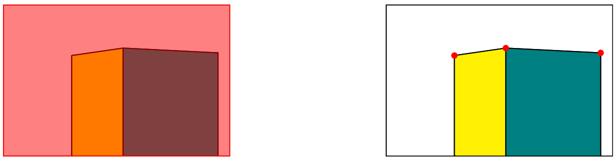
(A) ORB-SLAM family



(B) DSO family

FIGURE 2.24: ORB-SLAM and DSO-based works under focus

be selected according to different criteria in order to be representative of the scene and easier to match from a frame to another. An example of dense and sparse point selection is represented in Figure 2.25, where selected points are highlighted in red. While dense methods avoid losing information from the incoming images and lead to a more dense scene perception by triangulating a dense three-dimensional point cloud, they are far more expensive than sparse ones in terms of computation and memory. Both ORB-SLAM (Mur-Artal et al., 2015) and DSO (Engel et al., 2018) are sparse, but rely on different image points selection criteria: ORB-SLAM selects FAST corners at different scale levels (Rosten et al., 2006), and DSO selects pixel patches centered on high gradient points.



(A) Dense pixel selection (B) Sparse pixel selection (example)

FIGURE 2.25: Examples of dense and sparse pixel selection

A second classification can be done according to the information used to characterize the selected points. Direct methods rely on the photometric pixel value, while indirect methods use higher level descriptors. Some hybrid works combine direct and indirect approaches depending on the step in the SLAM pipeline, like OV$^2$SLAM (Ferrera et al., 2021), SVO (Forster et al., 2017) or LDSO (Gao et al., 2018). The main motivation of direct methods is to prevent loss of information caused by the descriptor-based representation. In addition, direct methods can cope with poorly textured environment and are robust to blur. Conversely, an argument in favor of indirect methods is that the use of high level descriptors instead of photometry is more robust to lighting variations and geometric and photometric distortion. Direct methods can take distortion effects into account by accurate geometric and photometric camera calibration (Engel et al., 2018), but the computation of these calibrations is constraining, and does not account for possible lighting changes. While DSO (Engel et al., 2018) is a fully direct approach, ORB-SLAM (Mur-Artal et al., 2015) is indirect and relies on ORB descriptors (Rublee et al., 2011).

Table 2.3 provides a classification of the VSLAM works under focus according to the visual information used in each of their main functionalities between direct (D), indirect (I), sparse (S) and dense (D) approaches.

TABLE 2.3: Visual information-based classification of VSLAM works under focus

| | **Pose tracking** | **Local mapping** | **Loop closing** | **SLAM recovery** |
|---|---|---|---|---|
| ORB-SLAM | I/S | I/S | I/S | I/S |
| ORB-SLAMM | I/S | I/S | I/S | I/S |
| ORB-SLAM Atlas | I/S | I/S | I/S | I/S |
| Dual-SLAM | I/S | I/S | I/S | I/S |
| DSO | D/S | D/S | ∅ | ∅ |
| LDSO | D/S | D/S | I/S | ∅ |
| DSM | D/S | D/S | D/S | ∅ |

## 2.4.2 Pose estimation and tracking

In monocular VSLAM, the pose estimation problem consists in estimating the current pose of a mobile camera in real time, given the image views or *frames* acquired by the camera simultaneously while moving around a static scene. Section 2.4.2.1 describes the tracking problem with more details. Sections 2.4.2.2 and 2.4.2.3 explain how the tracking process is implemented in ORB-SLAM-based and DSO-based works, respectively.

### 2.4.2.1 Problem description

Let us denote $\mathcal{F}_c$ the mobile camera coordinate frame. At time $t_n$, an image $\mathcal{I}(t_n)$. $\mathcal{I}(t_n)$ is an array of pixels which is a discrete representation of the projection of the scene on a plane, from the camera's point of view (see Appendix A.3). Figure 2.26 represents two image views from the same cubic, three-dimensional object, recorded at $t = t_n$ and $t = t_m$. Axes of frame $\mathcal{F}_c$ at each timestamp are represented in red, green and blue respectively.

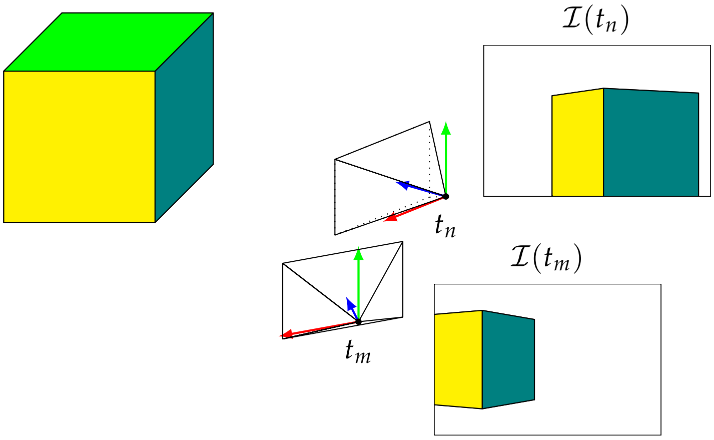


FIGURE 2.26: Multiple views of the same object

The pose estimation problem consists in estimating the pose ${}^{w}\mathbf{T}_{c_n}$ of the camera at $t = t_n$, were $\mathcal{F}_w$ indicates a world reference coordinate frame. Usually, $\mathcal{F}_w$ is selected such that ${}^{w}\mathbf{T}_{c_0} = \mathbf{I}_4$ where index 0 accounts for the first incoming frame. The tracking problem is solved by using the geometrical relations between a set of three dimensional points $\{\mathbf{P}_i\}_{i \in \{0 \dots N\}}$ and their projection in $\mathcal{I}(t_n)$ and one or multiple other image views of known pose. Figure 2.27 illustrates the correspondences between three-dimensional points from the scene and their projection in images $I(t_n)$ and $I(t_m)$ by red, dotted line. Points from images $\mathcal{I}(t_n)$ and $\mathcal{I}(t_m)$ which correspond to the same three-dimensional point are indicated in blue.

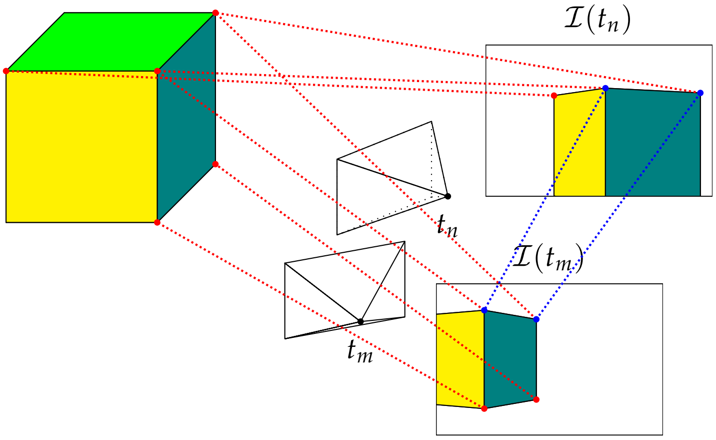


FIGURE 2.27: Image points and three-dimensional points correspondences

At the very first iteration of a VSLAM algorithm, environment map is still empty, and no environment knowledge is therefore available. The first tracking iteration is called *SLAM initialization*, and computes a pose ${}^{c_1}\mathbf{T}_{c_0}$ with respect to the initial camera frame by aligning observations of the same three dimensional points between frames $\mathcal{I}(t_1)$ and $\mathcal{I}(t_0)$. This initialization step also leads to map initialization, which will be discussed further in Section 2.4.3. Because the scale of the transformation cannot be computed from monocular camera input without a priori knowledge of the scene, a random scale factor is set at initialization. Further transformations and scene reconstruction will thus be computed with this unknown scale factor with respect to the real dimensions of the scene.

After map initialization, three-dimensional environment knowledge from the map is available and can therefore be integrated in the pose tracking process. The tracking process is also responsible for deciding wether data from the current frame should be inserted in the map. In this case, the current frame is turned into a KeyFrame (KF), which can be defined as a data structure which corresponds to a viewing position and orientation, and map features observed in the corresponding image. The two first KFs created come from initialization frames $\mathcal{I}(t_0)$ and $\mathcal{I}(t_1)$. Later KF insertion is conditioned by some criteria which will be discussed further in Section 2.4.3.

#### 2.4.2.2 Tracking in ORB-SLAM-based works

ORB-SLAM (Mur-Artal et al., 2015) is a sparse, indirect VSLAM framework relying on ORB features and descriptors (Rublee et al., 2011), which are extracted from each incoming frame.

At initialization, ORB-SLAM first matches ORB points from frames $\mathcal{I}(t_0)$ and $\mathcal{I}(t_1)$ using the ORB descriptors. These matches are used to compute transformation according to the coordinates of the matched image points using a geometric model. More specifically, a homography and a fundamental matrix (Hartley et al., 2004) are computed. If the corresponding three-dimensional points are planar, nearly planar or with low parallax, then the homography matrix characterizes well the transformation ${}^{c_1}\mathbf{T}_{c_0}$; otherwise ${}^{c_1}\mathbf{T}_{c_0}$ is better described by the fundamental matrix. The model that best fits the observations is then conserved and used for computing the initial pose and three-dimensional points triangulation. While this initialization algorithm is preserved in most ORB-SLAM-based works, ORB-SLAMM (Daoud et al., 2018) proposes to reduce initialization duration by using the fundamental matrix directly instead of selecting the best model between the homography and the fundamental matrix, based on the observation that the homography is rarely selected in practice.

After SLAM initialization, the tracking process uses both the previous frame and three-dimensional points from the map. Let us consider a frame acquired at time $t = t_n$. First, an initial guess of the camera motion is performed by taking ${}^{c_n}\mathbf{T}_{c_{n-1}} \approx {}^{c_{n-1}}\mathbf{T}_{c_{n-2}}$. This initial guess is used for a guided matching between the ORB points from frames $\mathcal{I}(t_{n-1})$ and $\mathcal{I}(t_n)$, according to their ORB descriptors. The pose is then optimized with the matched points. In a second time, some local map points are selected and projected in $\mathcal{I}(t_n)$ to perform a guided matching with ORB points from $\mathcal{I}(t_n)$. The pose is optimized again according to the resulting matches. The third and last step consists in deciding wether data from the current frame $\mathcal{I}(t_n)$ should be turned into a KF and used to increment the map. KF decision criteria include sufficient scene view change from the last KF inserted, time elapsed since the last KF insertion and lack of new incoming information and the computation resources available.

Except ORB-SLAMM's initialization, all ORB-SLAM-based algorithms described in Section 2.4, namely ORB-SLAMM (Daoud et al., 2018), ORB-SLAM Atlas (Elvira et al., 2019) and Dual-SLAM (Huang et al., 2020) follow the same tracking process as ORB-SLAM.

#### 2.4.2.3 Tracking in DSO-based works

DSO (Engel et al., 2018) is a sparse, fully direct VO. If available, a photometric calibration of the camera is used to apply a photometric correction of all pixel value in each new incoming frame. Pixel intensity is then assumed to depend only on the irradiance of the corresponding three-dimensional scene point and on the exposure time. If the exposure time is known, it is also corrected, otherwise its impact is ignored by the tracking process.

DSO's tracking uses a selection of image points chosen such that they are well-distributed in the image and with a high gradient magnitude with respect to their immediate surroundings. These points are characterized by the values of pixels from a patch which is centered on the selected image point. They are selected in a reference image, which is $\mathcal{I}(t_0)$ at initialization and the image from the most recent KF otherwise. The transformation between the frames is estimated using these active points, based on a conventional two-frame direct image alignment, a multi-scale image pyramid and a constant motion model. If the final residual is above a threshold, the algorithm tries applying up to 27 small rotations to the current relative pose estimate in different directions in a RANSAC-like procedure in order to find a pose estimate with an acceptable photometric residual. New KF decision criteria include sufficient field of view changes, similarly to ORB-SLAM-based works, but also occlusion and dis-occlusion detection and significant exposure time change, if available. Pose estimation according to the current tracking process is coarse, and is only refined if the frame is selected for KF creation, in a local map optimization which will be described in Section 2.4.3.

The same tracking process is implemented in the DSO-based work DSM (Zubizarreta et al., 2020). Implementation in LDSO (Gao et al., 2018) is very similar, but with a slightly different point selection in order to include a proportion of ORB points. All selected points are still processed as in DSO during the tracking process, and the inclusion of ORB points is reported not to impact on the tracking process (Gao et al., 2018).

### 2.4.3 Local mapping

The local mapping process consists in the insertion of new, local observations in the map. In the VSLAM approaches described in Section 2.4, this process consists in new KF creation according to the decision of the tracking algorithm, insertion of this new KF in the map and local map update including optimization and obsolete data marginalization. Section 2.4.3.1 develops the local mapping problem, including map representation and new data insertion steps. Sections 2.4.3.2 and 2.4.3.3 explain the local mapping process implementations in ORB-SLAM-based and DSO-based works respectively.

#### 2.4.3.1 Problem description

The VSLAM approaches studied in Section 2.4 are KF pose-graph based works, what is the predominant category of VSLAM at the time of writing. The map therefore consists in a graph of KF and of a set of map points which have been triangulated based on their observations in KFs. KF connections in the graph are determined based on a combination of temporal and covisible criteria. The temporal criterion connects KFs inserted consecutively in the map, and the covisibility criterion connects KFs sharing a sufficient quantity of map points observations. The resulting graph is usually referred to as *covisibility graph*. Figure 2.28 represents an example of map generated by running the monocular ORB-SLAM Atlas (Elvira et al., 2019) implementation from ORB-SLAM3 (Campos et al., 2021) on the *V1_01* sequence from the EuRoC dataset (Burri et al., 2016). It is a sample view from ORB-SLAM3's viewer. KF poses are represented in blue, except from the first KF which is red. Covisibility connections are drawn in green, and map points are indicated in black and red.

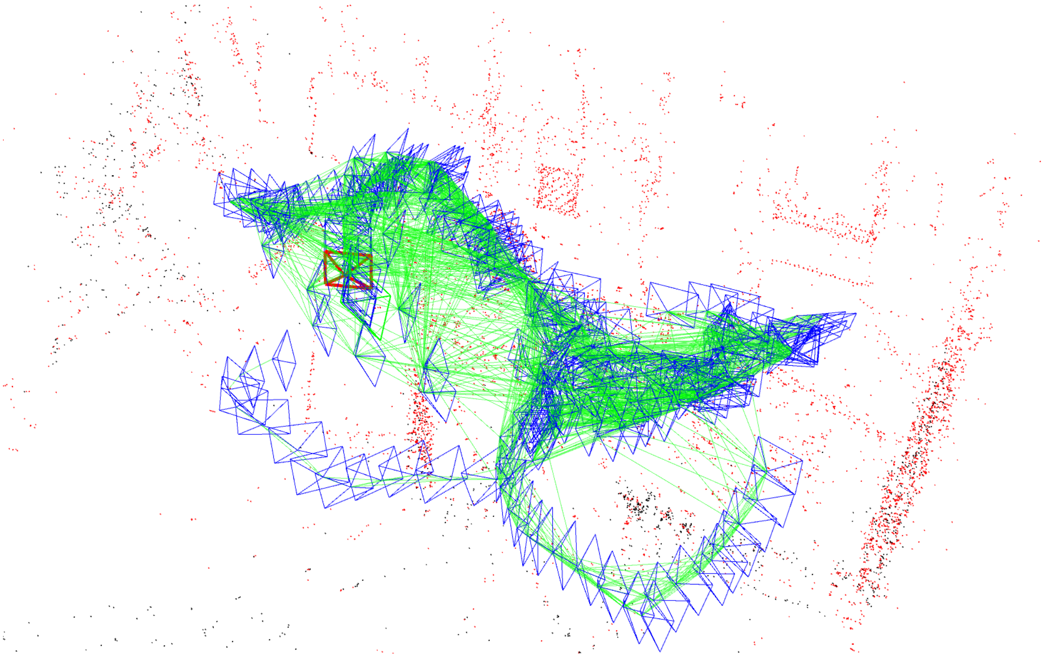
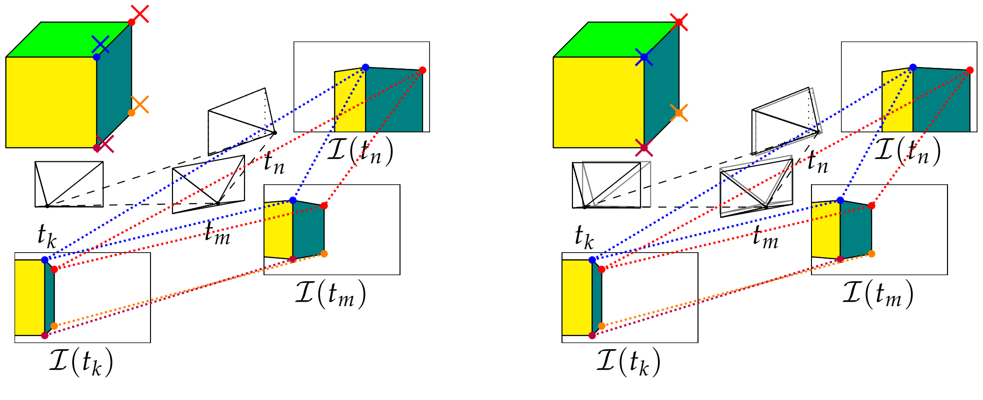


FIGURE 2.28: Example map visualization from ORB-SLAM3's viewer

Map initialization is performed from the two frames used at SLAM initialization (see Section 2.4.2), which are turned into two connected KFs and used to triangulate the first map observations. When a new KF decision is taken at the end of the tracking process, a new KF is created and inserted in the map. KF insertion consists in completing the map with the creation of a new node in the covisibility graph and its connection to previous KF nodes, but also new observations of previous map points with respect to the new KF and new map points triangulation. Map data is then updated in a local window in order to best fit new map data. Updates include the identification and marginalization of inaccurate, too redundant or no more relevant map points or KFs, but also map optimization according to new observations. Map optimization is usually performed by a local bundle adjustment (LBA) which optimizes simultaneously KF poses and map points coordinates on a local window of covisible KFs.

This optimization process is illustrated in Figure 2.29 using three connected KFs recorded at timestamps $t = t_k$, $t = t_n$ and $t = t_m$. Three-dimensional points and their projection in the frames are highlighted using the same color. The crosses denote the estimated location of the three-dimensional points. The LBA uses map point (MPs) observations among the KFs to correct three-dimensional point location and KF poses from the initial state of Figure 2.29a to the optimized state represented in Figure 2.29b. KF pose estimations as initialization are represented in gray in Figure 2.29b to visualize the adjustment.



(A) Initial state (B) Optimized state

FIGURE 2.29: LBA process illustration

#### 2.4.3.2 Local mapping in ORB-SLAM-based works

In ORB-SLAM (Mur-Artal et al., 2015), KF data structures include a pose, the ORB points detected in the original frame, the list of map points which have been detected in the image and the coordinates of the corresponding points in the image. The covisibility graph is then updated by connecting the new KF to the KF with most map point observations in common, according to the map points detected in the new KF's frame during the tracking step. In a second time, map points are updated by culling old map points according to observation consistency criteria and triangulating new map points from the new KF. KF poses and map points coordinates are then updated by a LBA on a local window composed of the newly inserted KF, all KFs connected to it in the covisibility graph and all map points visible from these KFs. Finally, too redundant KF are discarded from the map. In ORB-SLAM, the KF creation decision criteria are very permissive in order to prevent missing important information. KF without significant contribution to the map will be culled later on, as part of the local mapping process. KF culling also regulates the number of KF, since the more KFs in the map the higher memory and computational cost.

This process is similar in ORB-SLAMM (Daoud et al., 2018), ORB-SLAM Atlas (Elvira et al., 2019) and Dual-SLAM (Huang et al., 2020).

#### 2.4.3.3 Local mapping in DSO-based works

In DSO (Engel et al., 2018), KF data structures include a pose and the original frame. When a new KF is created, its pose is initialized with the coarse estimation computed at tracking. DSO maintains and updates a window of a fixed, small number of *active KF* selected among the most recent KFs. This KF window is updated by inserting new incoming KF and marginalizing obsolete ones based on timestamp, KF pose prior and redundancy such that the oldest, furthest and most redundant KFs are marginalized. Only map information from the last seconds is therefore used by the algorithm. One can notice that, similarly to ORB-SLAM, DSO's KF insertion strategy consists in creating many new KF and reduce their number afterwards by redundant KF marginalization.

At new KF insertion, a fixed number of *active points* is selected among active KF images. As for the tracking step, these points are described by the value of a pixel patch around the selected image point. Active points are chosen such that they are well-distributed in the images of the active KF and have sufficiently high image gradient magnitude with respect to their local neighborhood, but also that they can be matched in subsequent KFs. Candidate active points are therefore tracked in successive KFs along the epipolar line, by minimizing the photometric error, and the active points are selected from the best matches.

A photometric bundle adjustment is then conducted on the active KF window based on a photometric cost. Optimized parameters include KF poses, but also active point inverse depth, camera intrinsic parameters, and exposure parameters if the exposure time is unknown.

A similar process is implemented in LDSO (Gao et al., 2018) but with the inclusion of a proportion of ORB points, as for tracking, although their inclusion is reported not to deteriorate the photometric optimization result compared to DSO. ORB points are also included in the KF representation.

A fundamental difference between local mapping implementation in DSO and DSM (Zubizarreta et al., 2020) relies in the selection of the active KFs. DSM's local KF window includes both *temporal KFs*, selected like in DSO, and *covisible KFs*, which are older KFs selected such that they should provide complementary viewing information. Covisible KFs selection is based on estimated KF poses and prior, coarse pose estimation of the new KF, and thus accurate covisible KFs choice strongly relies on a good pose prior. Robustness of covisible KF selection and integration in the optimization can therefore be questioned.

### 2.4.4 Loop closing

Some SLAM algorithms are able to recognize a previously visited place and update the map accordingly. This process is known as loop closing and can be decomposed in two problems: loop detection, or place recognition, and map update. Loop closure implementation in ORB-SLAM-based and DSO-based works is described in Sections 2.4.4.1 and 2.4.4.2 respectively.

#### 2.4.4.1 Loop closing in ORB-SLAM-based works

Place recognition in ORB-SLAM (Mur-Artal et al., 2015) is based on DBoW2 bags of visual worlds (BoW) (Galvez-López et al., 2012). The visual vocabulary is given as input to the system, and a BoW representation of each new incoming KF $\mathcal{K}_i$ is computed at insertion, using its ORB points. These representations are stored in a database. Each new KF insertion triggers a query on the BoW database. KFs sharing enough visual worlds with the new, incoming KF are identified as loop closure candidates. KF candidates are validated using further map point observations matching and geometrical and covisibility criteria. If a valid candidate $\mathcal{K}_l$ is found, the similarity transformation ${}^{l}\mathbf{S}_i$ between the poses of $\mathcal{K}_l$ and $\mathcal{K}_i$ is computed geometrically from map point matches. The similarity transformation includes a rotation, a translation and a scale, which accounts for scale drift between insertions of $\mathcal{K}_l$ and $\mathcal{K}_i$.

If a consistent estimation of ${}^{l}\mathbf{S}_i$ is found, the map update is triggered. First, the poses of the neighbors of $\mathcal{K}_i$ in the covisibility graph are corrected using ${}^{l}\mathbf{S}_i$. Duplicated map points are fused, and new ones are intended to be triangulated according to the updated map geometry. A global bundle adjustment (GBA) is then launched in order to optimize map point coordinates and KF pose at a global scale. The cost of this operation is limited by not including all covisibility connection constraints but only those of a subgraph, called *essential graph*, which includes the covisibility graph spanning tree and covisibility edges accounting for a particularly high number of common map points between the connected KFs. Because these map updates can be concurrent with local mapping update, the local mapping process is stopped temporarily during these operations.

The exact same process is preserved in ORB-SLAMM (Daoud et al., 2018), ORB-SLAM Atlas (Elvira et al., 2019) and Dual-SLAM (Huang et al., 2020).

#### 2.4.4.2 Loop closing in DSO-based works

While DSO (Engel et al., 2018) does not implement any loop closure handling by definition, loop closing is solved very differently in LDSO (Gao et al., 2018) and DSM (Zubizarreta et al., 2020).

LDSO proposes an indirect loop closure handling strategy relying on an ORB points map representation and BoW place recognition, similarly to ORB-SLAM, using an improved version of DBoW2 (Galvez-López et al., 2012), DBoW3. The ORB points included in the set of active points are used to compute a BoW representation of each new KF, at insertion. The motivation in integrating these ORB points in the tracking and local mapping steps is to ensure the consistency between these processes and loop closing. When a new KF $\mathcal{K}_i$ is inserted, the BoW database is queried in order to find loop candidates, just like in ORB-SLAM. ORB points from $\mathcal{K}_i$ and candidates are matched and a similarity transformation with respect to the loop candidate $\mathcal{K}_l$ is deduced. If a consistant ${}^{l}\mathbf{S}_i$ transformation is found, map update is triggered in the form of a pose graph optimization of the map, where each pair of consecutively inserted KF is connected by an edge, and $\mathcal{K}_l$ and $\mathcal{K}_i$ are connected together. The poses of the KFs from the local window are set fixed in the optimization process so that the global optimization does not interfere with the local bundle adjustment.

In DSM, the integration of covisible KFs in the local optimization can be considered as a specific kind of loop closure process. Covisible KF selection can be assimilated to a place recognition process where loop candidates are selected based on a pose prior, within a long-term data association process. Long and short term map data are then fused as part of the local mapping process.

### 2.4.5 SLAM recovery

In KF-based VSLAM works, the tracking process feeds the mapping process — including local mapping and optionnally loop closing — by propagating a pose estimation from frame to frame. However, it may happen that the tracking process fails. In this case, the current pose is unknown and thus the SLAM pipeline is broken.

While SLAM recovery has not been investigated so far in a DSO-based framework, several strategies have been investigated in the ORB-SLAM family. In ORB-SLAM, it is performed by a relocalization process aiming at finding correspondences between the current frame an a KF from the map using a BoW query. If a KF match is found, the system perfoms a brute-force ORB features matching between the current frame and the reference KF and a relative pose is deduced from the matches. Relocalization is validated according to geometrical and consistency checks.

The relocalization-based strategy is however limited. Indeed, depending on the visual conditions and on the system getting out of the already mapped area, relocalization may never succeed, or at least lead to important time gaps without localization estimation, which can be critical for real-life applications. ORB-SLAMM (Daoud et al., 2018) first investigated multi-mapping as a SLAM recovery strategy. In case of SLAM failure, ORB-SLAMM initializes a new map without any relocalization attempt. This strategy avoids waiting for relocalization and allows the SLAM algorithm to continue running even if the maps are disjoint. A map merging thread continuously performs a DBoW2-based place recognition in all previous maps for each incoming KF. If a match is found between a pair of maps, it is used to compute a transformation between them. Dual-SLAM (Huang et al., 2020) and ORB-SLAM Atlas (Elvira et al., 2019) also rely on new map creation and multi-map fusion. In case of tracking loss, Dual-SLAM initializes a new map and tries to fuse it with the previous one by running a backward SLAM. ORB-SLAM Atlas initializes a new map only if relocalization fails. All old maps are stored as disconnected entities. The loop closure place recognition queries all maps, and two matched maps are then merged similarly to loop closure optimization. All maps can be used for relocalization. A major difference between ORB-SLAM Atlas and ORB-SLAMM is that ORB-SLAMM does not fuse map observations nor reuse map information from old maps. In addition, ORB-SLAMM's intermap place recognition is more permissive, with less consistency checks.

### 2.4.6 Workflows of VSLAM works under focus

Figures 2.30 to 2.36 represent the workflows of the ORB-SLAM and DSO-based works under focus. On can see that tracking and local mapping ar usually computed on different, parallel threads, as well as loop closing if implemented. GBA is always executed on an additional thread. The computational implementation of SLAM recovery can be very different from the VSLAM work and the recovery algorithm.

In DSO, LDSO and DSM (respectively Figures 2.34, 2.35 and 2.35), the photometric optimization optimizes simultaneously points inverse depth, camera intrinsic parameters, exposure parameters and camera poses.

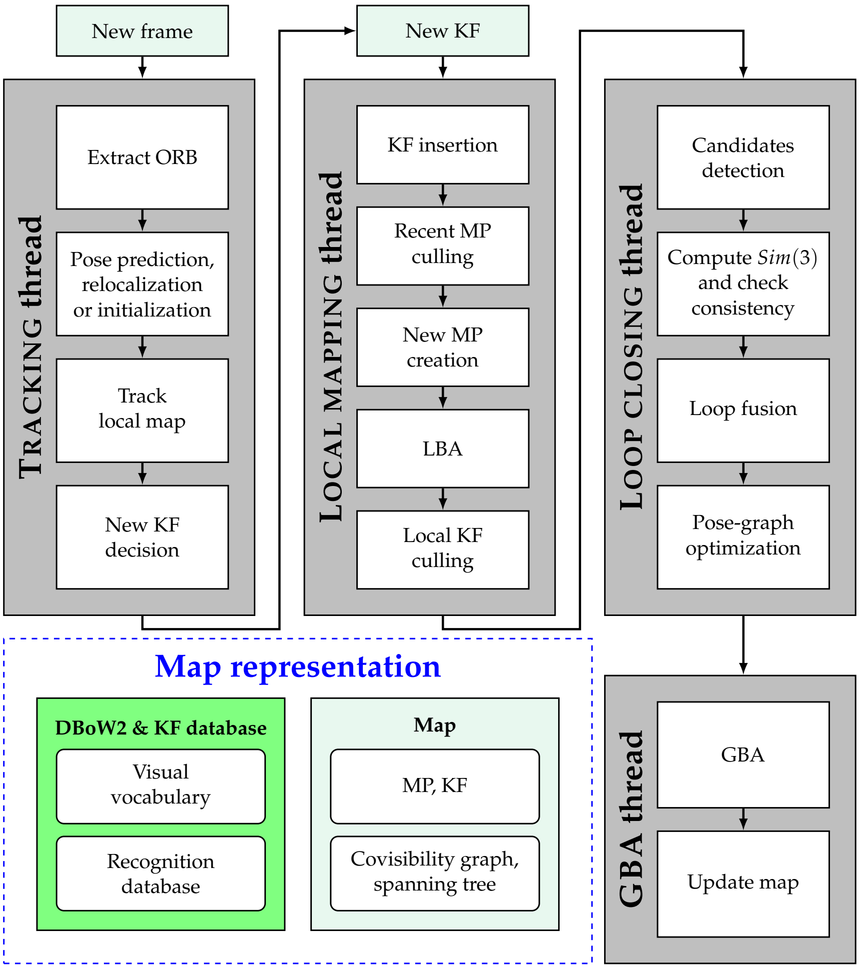


FIGURE 2.30: ORB-SLAM workflow

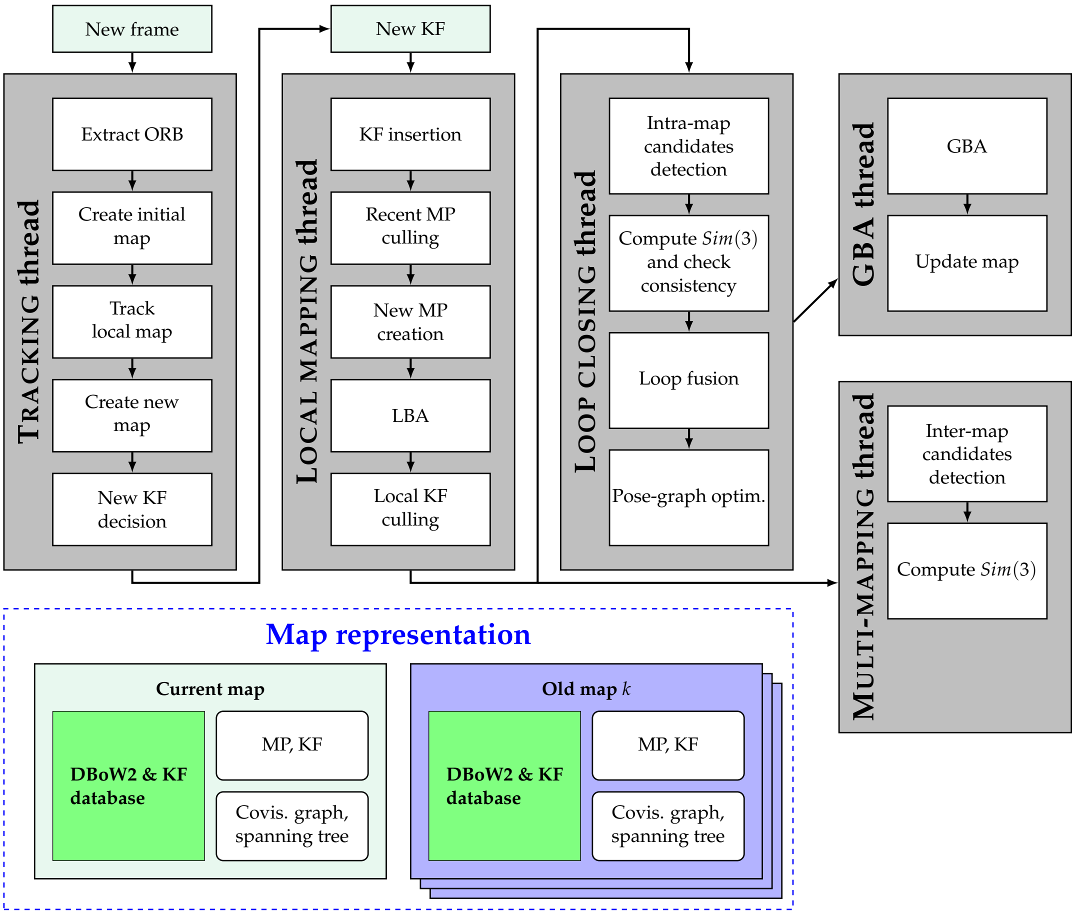


FIGURE 2.31: Single-agent ORB-SLAMM workflow

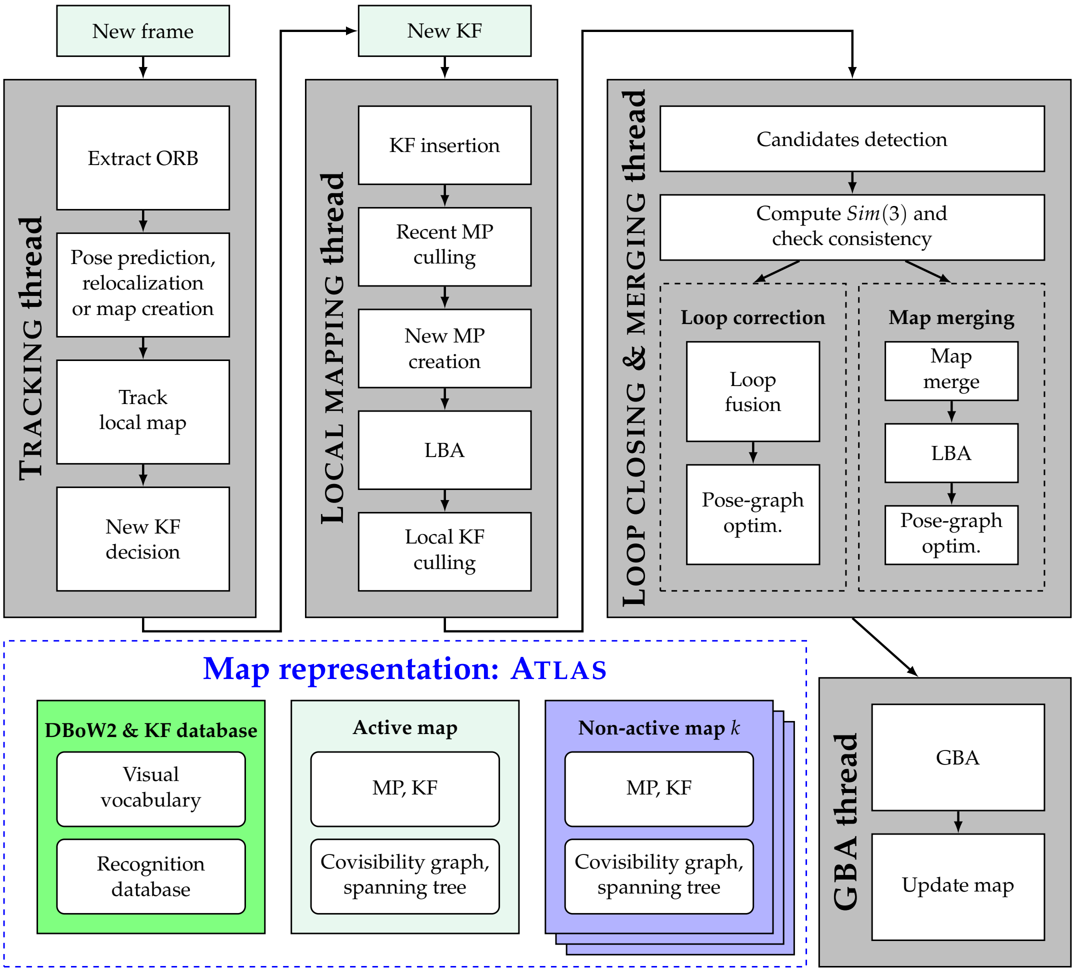


FIGURE 2.32: ORB-SLAM Atlas workflow

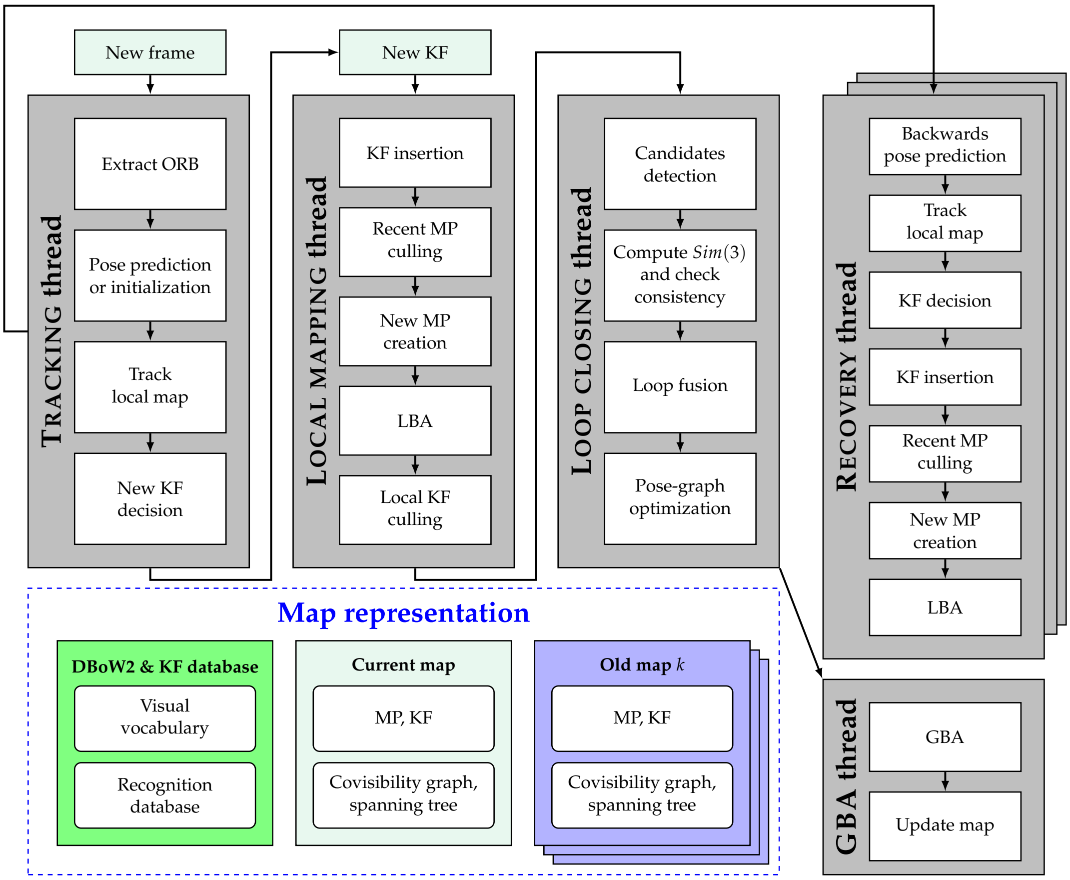


FIGURE 2.33: Dual-SLAM workflow

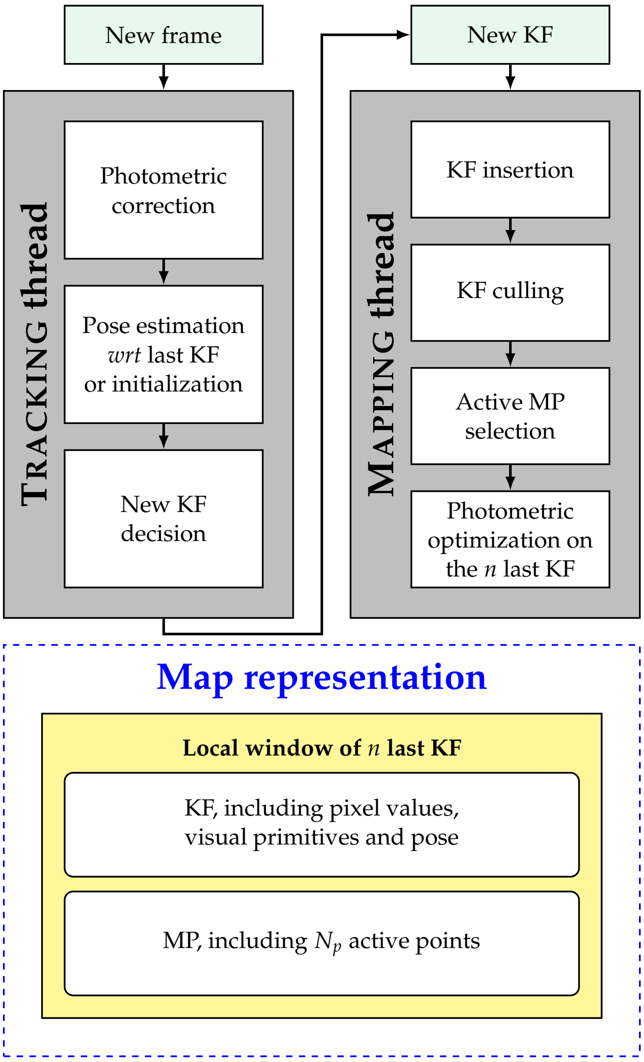


FIGURE 2.34: DSO workflow

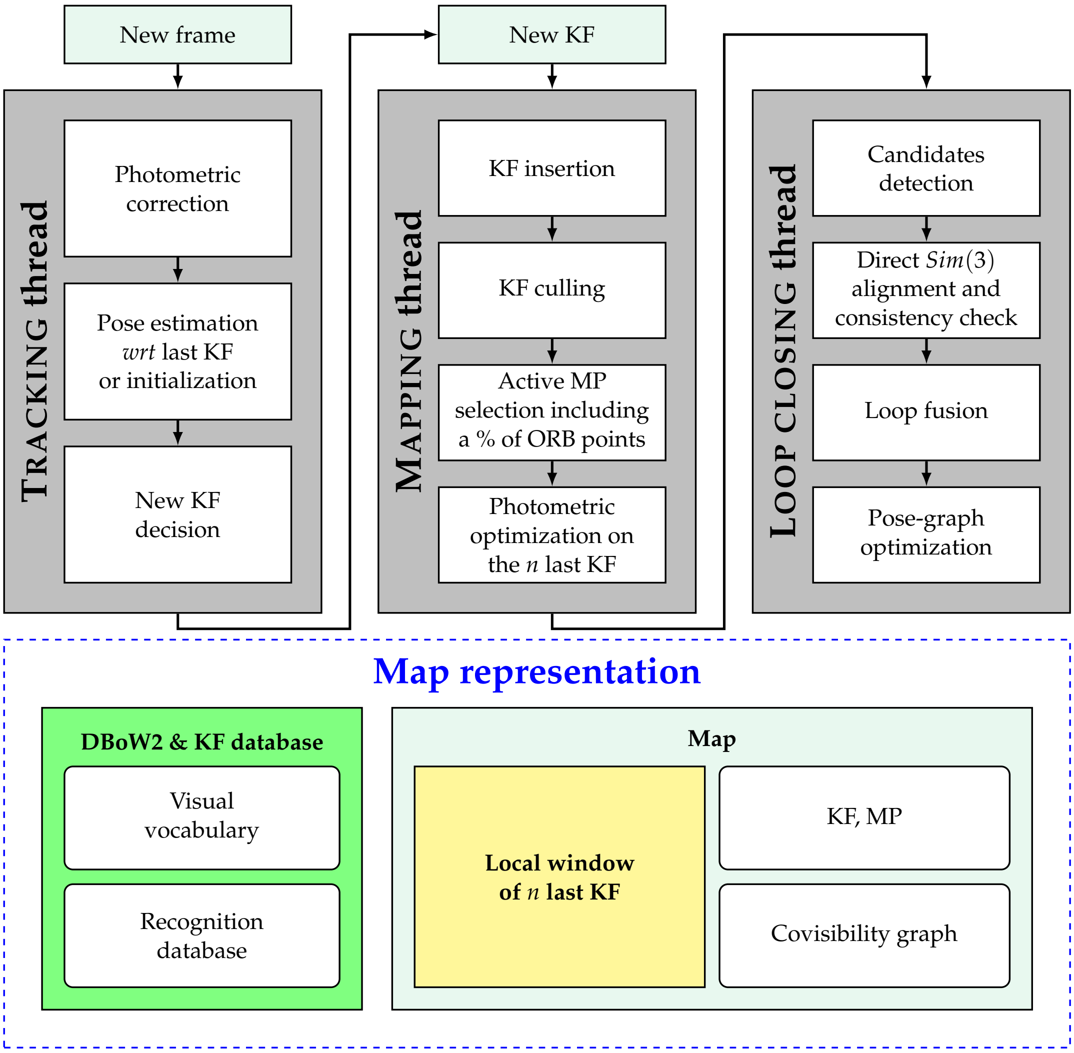


FIGURE 2.35: LDSO workflow

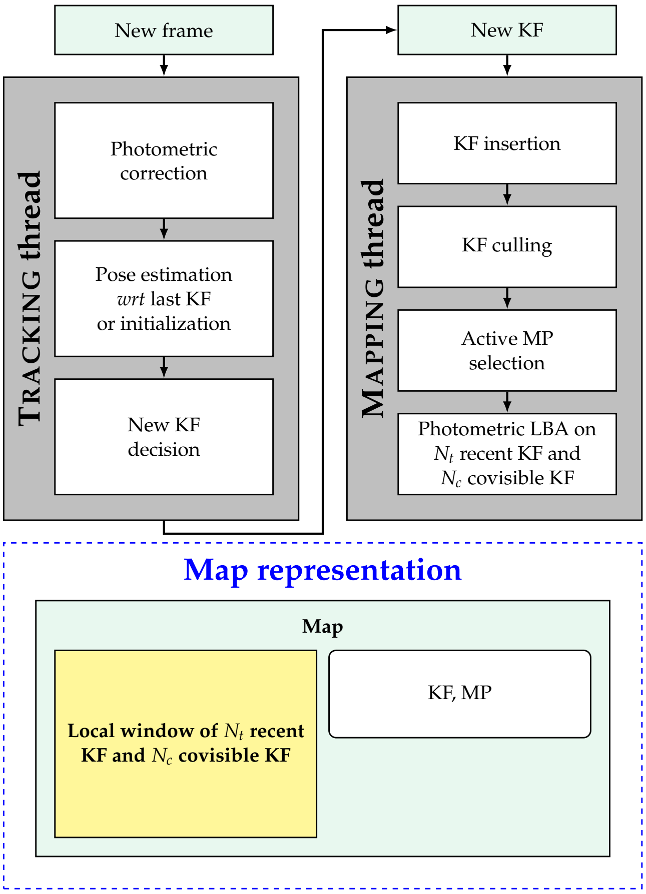


FIGURE 2.36: DSM workflow

### 2.4.7 Underwater datasets for VSLAM evaluation

While airborne VSLAM evaluation can rely on standard, public datasets recorded in different environments, featuring several sequences in the similar environments and visual conditions with various trajectories of gradual difficulty (Geiger et al., 2012; Burri et al., 2016; Schubert et al., 2018), there is no equivalent in the underwater field at the time of writing, because of the important cost and resources required for acquiring such data. Previous works on VSLAM benchmark under underwater conditions released their evaluation datasets, which are composed of heterogeneous sequences recorded in completely different environments from one to another, with various lighting conditions and camera settings (Quattrini Li et al., 2017; Joshi et al., 2019). Such heterogeneous datasets are particularly interesting for comparing VSLAM methods under very different conditions, but are not suitable for a detailed evaluation under specific conditions. (Ferrera et al., 2019) released AQUALOC, an underwater visual-inertial-pressure dataset. Similarly to standard aerial datasets, it is composed of several gradually more difficult sequences recorded in similar environments, on three different marine sites. All these sequences, however, show quite similar conditions by featuring man-made objects lying on a planar sandy area and involving only slow camera motion. As a result, the AQUALOC dataset only represents a small portion of the wide variety of underwater environments and visual conditions. The generation of ground truth trajectories relative to underwater datasets is more difficult than for aerial datasets. Whereas airborne datasets' ground truth commonly rely on laser scans or, sometimes, motion capture systems in smaller scale indoor environments, such systems are not available in the sea. In (Joshi et al., 2019), the output trajectory of a visual-inertial-SONAR-depth SLAM (Rahman et al., 2019) is used as a reference for visual and visual-inertial SLAM evaluation, but this can only apply to data acquired with a very specific sensor system. In AQUALOC (Ferrera et al., 2019), the offline Structure-from-Motion Colmap is used to compute a reference trajectory. These two strategies assume that the use of more sensors or time and computational resources will lead to a more reliable state estimation than real-time visual-only SLAM. Although the absence of a ground truth only allows a coarse comparison between VSLAM approaches, it has been shown that such qualitative evaluations can be sufficient to discriminate many VSLAM works in underwater fields (Quattrini Li et al., 2017; Joshi et al., 2019).

The current section describes the opensource underwater VSLAM evaluation datasets available at the time of writing, which were released as part of the work of (Quattrini Li et al., 2017) (*A/In*, *A/Out – Line*, *A/Out – Front*, *D/W* and *G/W* sequences)[1], (Joshi et al., 2019) (*Cemetery*, *Bus* and *Cave* sequences)[16] and (Ferrera et al., 2019) (AQUALOC dataset, composed of *Aqualoc Harbor* and *Aqualoc Arquaeo*)[17]. Section 2.4.7.1 describes data content of these datasets. Section 2.4.7.2 provides descriptions of the recording conditions and of the motion and trajectory operated.

#### 2.4.7.1 Sensors and data content

Table 2.4 presents the cameras involved in all the datasets. The *A/Out – Line* includes two cameras indexed #1 and #2 with different framerates, no field overlapping and no extrinsic calibration provided for a stereo use. Additional embedded sensors and sensor framerates are given in Table 2.5.

TABLE 2.4: Camera description

| | **Channels** | **Mono/Stereo** | **Distortion** | **Dimensions** | **Orientation** |
|---|---|---|---|---|---|
| *A/In* | RGB | Mono | Rad-tan | 776×640 | Forward |
| *A/Out – Line #1* | RGB | Mono | Rad-tan | 776×640 | Forward |
| *A/Out – Line #2* | RGB | Mono | Rad-tan | 1024×768 | Downward |
| *A/Out – Front* | RGB | Mono | Rad-tan | 1024×768 | Forward |
| *D/UW* | RGB | Mono | Rad-tan | 640×480 | Downward |
| *G/UW* | RGB | Stereo | Rad-tan | 1920×1080 | Forward |
| *Cemetery* | RGB | Stereo | Rad-tan | 1200×1600 | Forward |
| *Bus* | RGB | Stereo | Rad-tan | 1200×1600 | Forward |
| *Cave* | RGB | Stereo | Rad-tan | 1200×1600 | Forward |
| *AQUALOC Harbor* | Gray | Mono | Fisheye | 512×640 | Downward |
| *AQUALOC Arquaeo* | Gray | Mono | Rad-tan | 608×968 | Downward |

---

[1] https://afrl.cse.sc.edu/afrl/resources/datasets/

[17] https://www.lirmm.fr/aqualoc/

TABLE 2.5: Sensors and framerates

| | **Camera** | **IMU** | **SONAR** | **Pressure sensor** | **GPS** |
|---|---|---|---|---|---|
| *A/In* | 15 Hz | 50 Hz | ∅ | ∅ | ∅ |
| *A/Out – Line* | #1: 4 Hz<br>#2: 15 Hz | 50 Hz | ∅ | ∅ | ∅ |
| *A/Out – Front* | 15 Hz | 50 Hz | ∅ | ∅ | ∅ |
| *D/UW* | 2.5 Hz | 10 Hz | ∅ | ∅ | 2 Hz |
| *G/UW* | 30 Hz | ∅ | ∅ | ∅ | ∅ |
| *Cemetery* | 15 Hz | 100 Hz | ∅ | ∅ | ∅ |
| *Bus* | 12.5 Hz | 100 Hz | 100 Hz | 1 Hz | ∅ |
| *Cave* | 12.5 Hz | 100 Hz | 100 Hz | 1 Hz | ∅ |
| *AQUALOC Harbor* | 20 Hz | 200 Hz | ∅ | 5-10 Hz | ∅ |
| *AQUALOC Arquaeo* | 20 Hz | 200 Hz | ∅ | 60 Hz | ∅ |

The *Aqualoc Harbor* and *Aqualoc Arquaeo* datasets are composed of 7 and 10 sequences respectively, recorded in quite similar conditions. They include a reference trajectory computed from SfM. All other datasets are composed of a single sequence, and provide no ground truth or reference trajectory for quantitative benchmarking.

#### 2.4.7.2 Recording conditions and sequence description

Table 2.6 indicates the duration, approximate depth, presence of embedded lights and presence of loop closures in the trajectory. Table 2.7 gives a qualitative description of the visual conditions and motion.

TABLE 2.6: Datasets description

| | **Duration** | **Depth** | **Embedded light** | **Loop closure(s)** |
|---|---|---|---|---|
| *A/In* | 88 s | Unknown | ∅ | ∅ |
| *A/Out – Line* | 53 s | Unknown | ∅ | ∅ |
| *A/Out – Front* | 88 s | Unknown | ∅ | ∅ |
| *D/UW* | 602 s | Unknown | ∅ | ∅ |
| *G/UW* | 272 s | Unknown | ∅ | ✓ |
| *Cemetery* | 433 s | 20 m | ∅ | ✓ |
| *Bus* | 584 s | 20 m | ∅ | ✓ |
| *Cave* | 709 s | 20 m | ✓ | ✓ |
| *AQUALOC Harbor* | 2-15 min | 3 m | ✓ | ✓ |
| *AQUALOC Arquaeo* | 2-15 min | 270 m and 380 m | ✓ | ✓ |

TABLE 2.7: Datasets visual condition and motion description

| | **Description** |
|---|---|
| *A/In* | Forward traveling inside a shipwreck off the Barbados coast. Water is globally clear, but parts of the sequence feature suspended particles and fishes. |
| *A/Out – Line* | Forward traveling over a coral reef on the off the Barbados coast, including some mobile elements like seaweeds, fishes and suspended particles. A part of camera #2's field of view is occupied by its housing. |
| *A/Out – Front* | Alongside a shipwreck, outside. A part of the camera housing is visible. |
| *D/UW* | Drifter off the Barbados coast. Low image resolution and framerate. Fast, abrupt motion. |
| *G/UW* | Complete tour around a shipwreck, but with small loop closure overlap. Important scene aspect change between the two lateral sides of the wreck. Turbid water. Quite slow motion. |
| *Cemetery* | Fake submerged cemetery, composed of large rocks aligned on a sandy area. The camera follows the path delimited by the rock and never films sand only areas. Quite slow motion. |
| *Bus* | Complete tour around a sunken bus, in turbid water. Small part of the trajectory inside the bus. Important scene aspect change between the two lateral sides of the wreck, with one side very poorly illuminated. Quite slow motion. |
| *Cave* | Inside an underwater cave, showing natural mineral-only environment in clear water. Quite slow motion. |
| *AQUALOC Harbor* | Around large man-made objects lying on the sand. Some loop closures in the sequences are marked by an apriltag target. Quite slow motion. |
| *AQUALOC Arquaeo* | Around amphora hills with hight texture but sometimes facing low textured sandy areas. Turbidity, backscattering, and presence of suspended particles and fishes in some sequences. Quite slow motion. |

### 2.4.8 Discussion

In state-of-the-art, opensource, monocular VSLAM evaluations in underwater scenarios, ORB-SLAM is demonstrated to be quite robust to underwater visual conditions but subject to initialization difficulties and critical relocalization failure after tracking failure, and DSO is reported to cope with poorly textured environment and be robust to blur, as expected for a direct method, but also to suffer occasionnaly from important inconsistencies and early failure (Quattrini Li et al., 2017; Hidalgo et al., 2018; Joshi et al., 2019).

More frequent tracking failures in underwater scenarios can be explained by the visual distortion, which impacts both visual data detection and matching. In ORB-SLAM, the SLAM can still recover from a tracking failure by relocalizing in the map, but this strategy is useless if the camera moves out of the previously mapped area, hence the interest in new SLAM recovery techniques based on the creation of a new map which is intended to be fused to the previous one. Such algorithms are implemented in ORB-SLAMM, ORB-SLAM Atlas and Dual-SLAM with different map merging strategies. While these approaches may be particularly interesting for underwater scenarios because of their robustness to SLAM failures, they have not been compared yet on underwater scenarios. They are indeed posterior to the underwater evaluation works mentionned in the current section (Quattrini Li et al., 2017; Hidalgo et al., 2018; Joshi et al., 2019).

While no such SLAM recovery algorithms have been developped on a DSO-basis at the time of writing, the long-term data association functionnalities implemented in LDSO and DSM are expected to improve robustness to underwater conditions by reducing the risks of tracking failures and map inconsistencies. Indeed, longer-term data associations may improve map optimization by using older KFs with more parallax, leading to better KF pose prior and knowledge of the scene geometry. However, integration of long-term information is still limited in DSM and LDSO. In LDSO, covisibility connections between two KFs from which a loop closure has been completed are definitively lost after the GBA. This covisibility information are therefore never used for local KF window selection in the pose tracking and local mapping processes. Conversely, DSM takes advantage of longer-term map data to select higher parallax views in the optimization window. The main drawback of DSM is the selection process of these KFs, which entirely relies on a good pose prior. Therefore, inconsistent pose prior from a bad pose tracking may lead to wrong long-term data association and result in even more inconsistent mapping.

## 2.5 Multi-agent VSLAM for underwater

Some robotic applications involve the deployment of multiple agents, which can be a team of moving ground vehicles, flying drones, underwater vehicles, or any group of multiple mobile sensor sets. Applications include, for instance, environment exploration for multi-view mapping purpose (Michael et al., 2014; Özkahraman et al., 2022) or within a rescue mission (Murphy et al., 2012; Cho et al., 2018), and object transport and manipulation (D'Antonio et al., 2022). Such scenarios require localizing each robot with respect to each other, a possible collaborative environment mapping, hence an increased interest in multi-agent SLAM. These algorithms can involve individual measurements, which characterize the state of the agent performing the measurement, and optionally relative measurements, which describe the state of one agent relatively to another.

### 2.5.1 Architectures and communication schemes

Multi-agent SLAM approaches can be classified according to the distribution of their computations among the agents and a central server, as illustrated in Figure 2.37.

In fully centralized approaches, all localization computations are done on a central server, which can be one of the agents or an external server. Other agents only perform measurements and send their data to the server. Optionally, the server can send their estimated location to the agents. Centralized approaches strongly rely on the server having sufficient computational resources and bandwidth. As a result, they are not scalable to high number of agents and are not robust to communication failures.

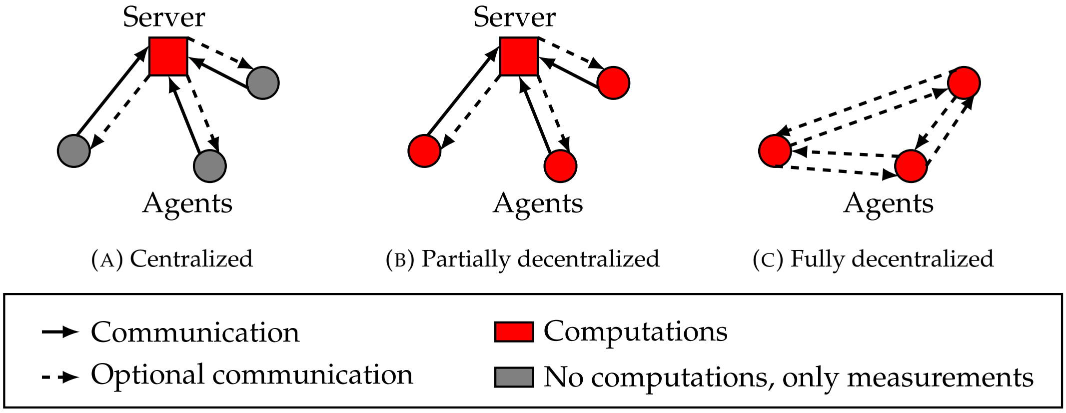

(A) Centralized (B) Partially decentralized (C) Fully decentralized

FIGURE 2.37: Multi-agent SLAM communication schemes

Conversely, decentralized approaches distribute the computations among the agents (fully decentralized) or between the agents and a central server (partially decentralized). The main motivations of partially decentralized approaches are to reduce the server's computational payload and provide a better robustness to communication failures, since agents can keep estimating a localization based on their own measurements if the server is unavailable. Scalability of such methods is although still dependent on the server's computation resources and require to exchange significant amounts of data.

Lastly, fully decentralized algorithms do not involve any server. Agents communicate their measurements and optionally a state prior to each other, and all computations are distributed among them. The motivation of such approaches is to reduce the communication bandwidth and computational requirements into a more scalable multi-agent localization scheme. Works on these approaches focus on reducing bandwidth requirements by communication sparsification, data marginalization and condensing inter-robot data exchanges in order to improve scalability and robustness to communication restrictions (Dubois et al., 2019). However, fully decentralized algorithms are less accurate than the ones involving a central server, since observations are never fused at a global scale.

### 2.5.2 Airborne works

Existing multi-agent VSLAM approaches adopt a centralized or partially decentralized approach (Zou et al., 2019), even though a few works investigate the theoretical aspects of a fully decentralized architecture (Cunningham et al., 2010; Cunningham et al., 2013; Leonardos et al., 2017). In addition, none of these works investigate the integration of relative measurement in a multi-agent VSLAM system.

Historically, multi-agent VSLAM has been investigated alongside the very first VSLAM approaches, mostly building upon state-of-the-art single-agent algorithms (Zou et al., 2019). The first multi-agent VSLAM built on PTAM. Taking advantage of the keyframe map structure, PTAMM (Castle et al., 2008) builds a common map with two PTAM tracking inputs. Agent-keyframe association is preserved though, in order to limit the relocalization only on keyframes created by the lost agent, and thus reduce its computational cost. C$^2$TAM (Riazuelo et al., 2014) extends PTAM's system to a set of $n$ collaborative agent with RGB-D input, and distributes the computations between the agents themselves and a central server. Each agent runs a tracking thread and sends keyframes to the server, which runs a single common mapping thread. The mapping thread initially creates and updates a map for each agent, and tries to fuse maps according to place recognition. Using RGB-D input allows to have the scale of the maps. After being updated by the server, the maps are sent to the agents. Similarly to PTAM, both PTAMM and C$^2$TAM are, though, restricted to small office environments. Contrary to these works, CoSLAM (Zou et al., 2013) implements a collaborative tracking which takes advantage of the overlapping views of groups of camera to track the scene and triangulate points. In addition, this approach allows to map dynamic objects in the environment. However, this implies camera synchronization. Place recognition is also handled very differently from PTAMM and C$^2$TAM. All agents are initialized with a common overlapping area, allowing to align their world reference frames. A loop closure detection is implemented based on pose prior instead of keyframe descriptor matching, neglecting the drift of individual agents localization and mapping. No relocalization is implemented. CoSLAM is then very dependent on good tracking performances, and is not adapted for agents with rare view overlaps. This work is although interesting because it tries to tackle the multi-agent VSLAM problem in an original manner and take advantage of the current view overlaps in all the steps of VSLAM's pipeline, contrary to most of the state of the art. Fusing the maps of multiple agents got transformed by the place recognition capabilities provided by bags of visual words (BoW) (Nister et al., 2006). A server backend for multiple keyframe maps fusion is introduced in (Deutsch et al., 2016). Map overlaps are detected by BoW and a pose graph optimization is performed. All maps, however, need to be at the same scale.

While interest in aforementioned multi-agent works is mostly historical, the most recent multi-agent VSLAM approaches build upon ORB-SLAM (Mur-Artal et al., 2015) and ORB-SLAM2 (Mur-Artal et al., 2017). CORB-SLAM (Li et al., 2018) introduces a first ORB-SLAM based multi-agent architecture. Each agent runs ORB-SLAM individually and builds its own maps. These maps are sent to a central server which tries to detect overlapping regions by a DBoW2 place recognition similar to ORB-SLAM's loop closure. Matched maps are then fused into a single common map. The global map is sent to the agents after each update. CCM-SLAM (Schmuck et al., 2019) introduces a monocular partially decentralized framework which applies to vehicles with limited onboard memory and computational resources, under communication bandwidth constraints. Each agent conducts ORB-SLAM's tracking and local mapping and only maintains a window of local KFs. New KFs and map points are sent to a central server, which stores the database of maps, handles loop closures for each agent and performs inter-map DBoW2 place recognition and map merging, with the same place recognition algorithm as ORB-SLAM. In addition, the server provides agents KFs for their local window. Several agents can then localize in the same map and share the same KFs. A key contribution of that work is the ability of the agents to keep performing an individual visual odometry if communication with the server is broken. A downside of this approach is that no SLAM failure handling strategy is implemented. The evaluation of CCM-SLAM provided in (Schmuck et al., 2019) focuses on real-time capabilities and limited bandwidth handling, resulting in a limited evaluation of the localization and mapping accuracy, which is only based on Root-Mean-Square (RMS) Absolute Position Error (APE) on agent's trajectories. A comparison with state-of-the-art VSLAM is also missing, as well as an evaluation of relative localization capabilities between agents and global map accuracy characterization. While ORB-SLAMM (Daoud et al., 2018) is presented initially as a single-agent monocular approach, it can be extended to a multi-agent fully centralized framework. In the multi-agent extension, a tracking thread, a local mapping thread and a loop closing thread are run for each agent, but the system includes one single multi-map database and one single map merging thread. This multi-agent implementation is, however, only evaluated superficially, according to the global RMS APE over agent trajectories, showing the same evaluation limits as CCM-SLAM. A synthesis of these ORB-SLAM-based works is provided in Table 2.8. The workflows of CORB-SLAM (Li et al., 2018), multi-agent ORB-SLAMM (Daoud et al., 2018) and CCM-SLAM (Schmuck et al., 2019) are represented in Figures 2.38, 2.39 and 2.40 respectively.

TABLE 2.8: ORB-SLAM-based multi-agent VSLAM synthesis. Unless otherwise stated, tracking (T), local mapping (LM), loop closing (LC), place recognition (PR) and GBA operations are the same as the one implemented in ORB-SLAM. Map merging is abbreviated into MM.

| | **Agents** | **Server** | **Map merging** | **Recovery** |
|---|---|---|---|---|
| CORB-SLAM<br>(Li et al., 2018) | 1 T thread<br>1 LM thread<br>1 LC thread<br>(1 LC GBA thread)<br>+ send maps<br>to the server | 1 inter-map PR thread<br>(1 MM GBA thread)<br>+ send map updates<br>to the agents | Similar to<br>ORB-SLAM's<br>LC + GBA | ∅ |
| ORB-SLAMM<br>(Daoud et al., 2018) | Send frames<br>to the server | 1 T thread per agent<br>1 LM thread per agent<br>1 LC thread per agent<br>(1 LC GBA thread<br>per agent)<br>1 inter-map PR thread | Close to<br>ORB-SLAM's LC<br>but with less<br>geometric<br>consistency<br>checks | Multi-map |
| CCM-SLAM<br>(Schmuck et al., 2019) | 1 T thread<br>1 LM thread<br>+ send new KF<br>to the server | 1 inter-map PR thread<br>(1 MM GBA thread)<br>+ send local KF<br>to the agents | Similar to<br>ORB-SLAM's<br>LC + GBA | ∅ |

### 2.5.3 Waterborne works

Multi-agent systems are also studied for underwater applications. Multiple AUVs can be deployed simultaneously to cover wider areas in various applications including infrastructure inspection, seabed mapping, search and rescue, oceanographic studies (Bechlioulis et al., 2019). Some works investigate the problem of multi-agent VSLAM for AUVs (Mangelson et al., 2018; Özkahraman et al., 2022). The main challenge in these works consists in managing information sharing between AUVs under the bandwidth limitations of underwater wireless communications. Multiple ROV applications are more rare and include archaeological exploration, recovery missions (Murphy et al., 2012) or tether management strategies like robot chains (Laranjeira et al., 2020). Although fully communicating multi-ROV configurations allow the theoretical implementation of the state-of-the-art airborne multi-agent VSLAM works, there are no works addressing this problem in the literature.

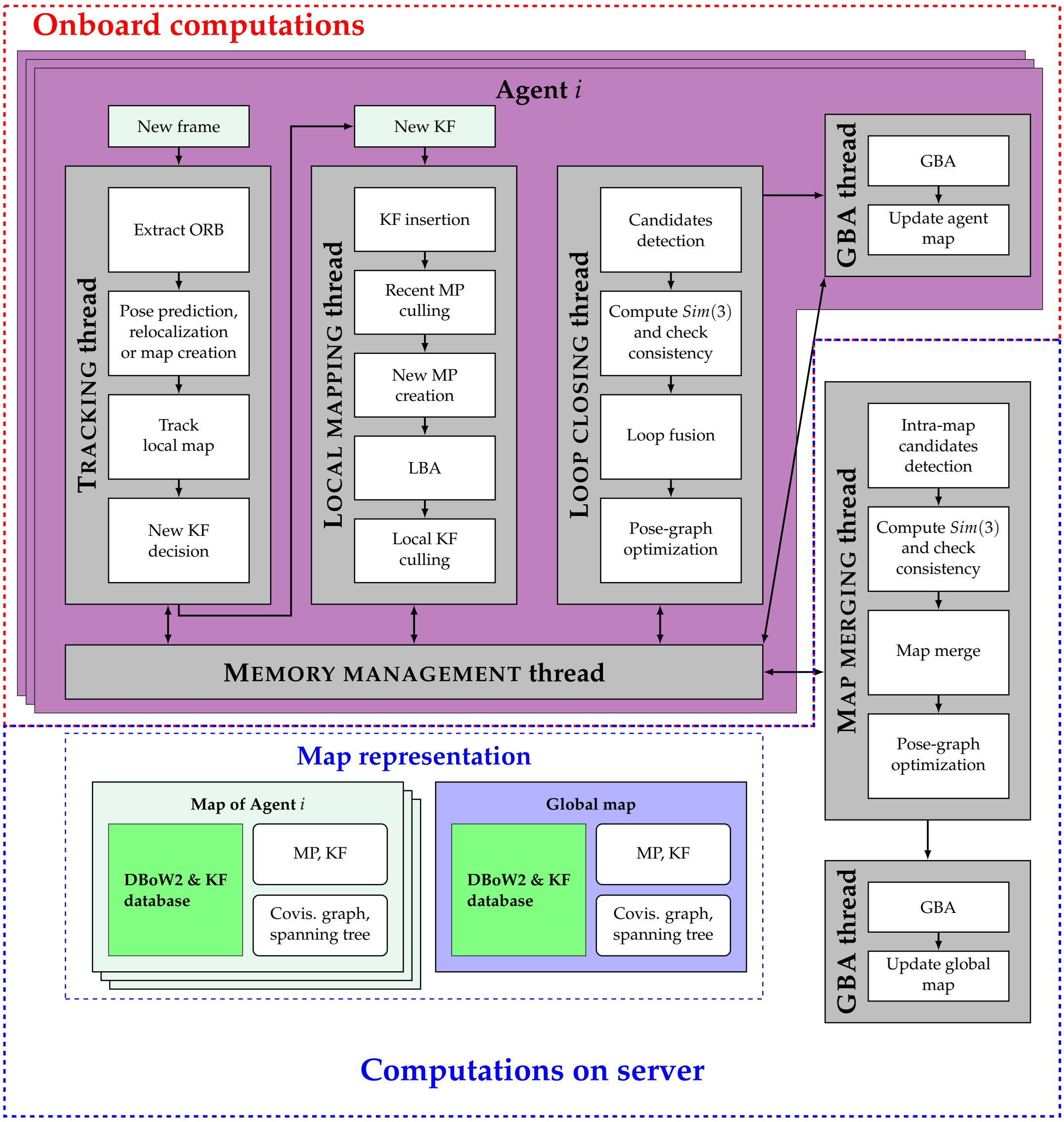


FIGURE 2.38: CORB-SLAM workflow

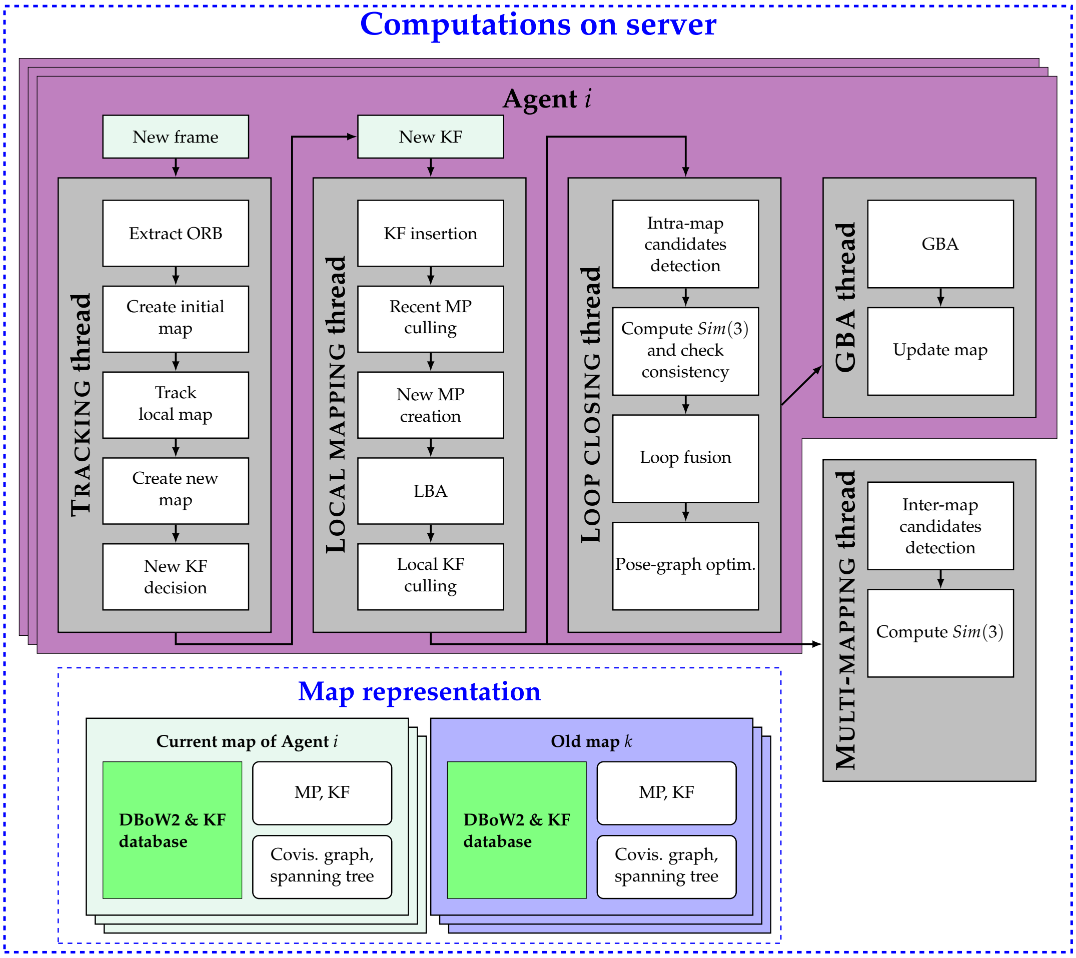


FIGURE 2.39: Multi-agent ORB-SLAMM workflow

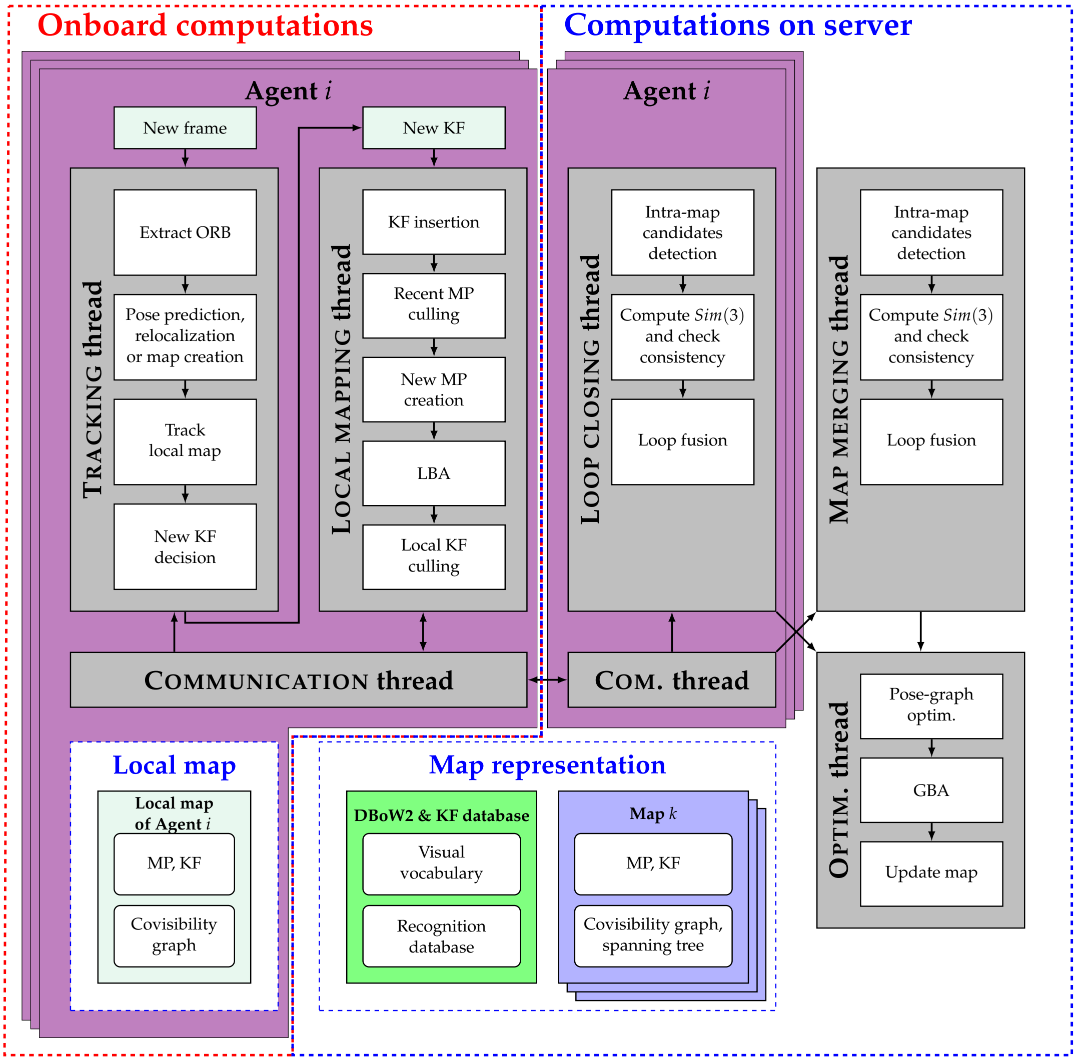


FIGURE 2.40: CCM-SLAM workflow

## 2.6 Discussion and key scientific challenges

The scientific focus of this thesis is the localization of an underwater ROV chain, which is a system of underwater, tethered robots, which can communicate together and to a surface station, in real time. This localization problem involves locating both the robots and the cable parts between them, with respect to their environment. The relative pose of the robots and cable parts 3-dimensional configurations are strongly linked. On the one hand, cable shape can be deduced from robot's relative pose according to a cable model. In this case, visual SLAM is considered as a promising localization approach, which is also interesting in providing environment knowledge. On the other hand, robot's relative pose can be deduced from cable 3D shape knowledge. This brings the following key scientific challenges:

- how can the shape of a cable linking a pair of underwater robots be estimated with embedded sensors? and which what level of accuracy?
- to what extent can such a cable shape estimation be used for estimating the relative pose of tethered underwater robots?

- to what extent are the state-of-the-art VSLAM algorithms suitable for underwater robot localization, in the single agent case?
- how can multiple underwater agents be localized within a multi-agent VSLAM framework? and with what accuracy?
- may cable-based and VSLAM-based localization approaches be fused to improve robot chain localization?

# Chapter 3

# Cable-based state estimation

## 3.1 Introduction

In this chapter, we focus on the state estimation of a ROV connected to another vehicle, based on an estimation of the shape of its umbilical. The tether is then turned into an advantage by being converted into a localization system which can be used when other systems are unusable or as a redundant system to another location system. In addition, if the shape of the cable is known between the ROV and the surface, it is then possible to deduce the GPS coordinates of the ROV from those of the boat and the position of the cable end point placed on the ROV, which is then known with respect to the boat.

In the work presented in this chapter, the shape of a submarine cable is estimated according to a simple model and external sensor measurements involving pressure sensors and IMUs. Two configurations have been studied, with different models. First, the catenary model is investigated as a promising quasi-static model for a submarine cable with negative buoyancy connecting a pair of ROV within a robot chain. In a second time, a straight line model for a cable constrained into a piecewise linear shape is investigated in line with the work of Christophe Viel, CNRS-LabSTICC, Brest, France (Viel, 2022a; Viel, 2022b), as part of a collaboration. The aim of this collaboration was to invert the cable models developed in (Viel, 2022a; Viel, 2022b) in which the shape of the cable of a ROV is expressed as a function of the ROV's position, in order to compute the position of the ROV from an estimation of the cable state, obtained from local inertial and depth measurements.

The contributions of this chapter are the following:

- a study of the validity of the catenary model for moving submarine cables with negative buoyancy
- a new inertial-measurement based catenary shape estimation of a underwater cable connecting a pair of underwater robots and its experimental validation and comparison with a previous visual-based shape estimation method
- a new inertial-measurement based straight-line shape estimation for an underwater cable constrained into a piecewise linear shape by a system of sliding buoys and ballasts, including three models corresponding to different configurations and an experimental validation on several scenarios including a robot connected to a fixed surface point and a pair of underwater robots connected together by a cable in the context of a robot chain.

Section 3.2 presents the cable models considered and the experimental validation of their validity for underwater cables with moving ends. Section 3.3 describes how the parameters of these models can be estimated using embedded sensors, namely IMUs and depthmeters. Section 3.4 focuses on the model-based state estimation of underwater tethered robotic systems according to the models and embedded measurements of their parameters. Conclusions and perspectives are developed and discussed in Section 3.5

## 3.2 Cable modeling

Two cables models are studied, namely the *catenary model* for a hanging cable, and a *straight-line* model for a cable constrained into a piecewise linear shape by the addition of sliding buoys and ballast along it. Sections 3.2.1 and 3.2.2 focus on these two models respectively.

### 3.2.1 Catenary model

The catenary model is defined as the shape of an idealized homogeneous hanging cable with fixed length and fixed ends, only subject to its own weight in the air. Underwater cables are also submitted to a buoyancy force, opposed to the weight. The catenary model then only applies to cables for which the resulting vertical force is non-zero, *i.e.* only positively and negatively buoyant cables. This model is particularly interesting because of its simplicity, but extending it to an underwater cable with moving ends assumes that the hydrodynamic forces exerted on the cable are negligible compared to its weight, and that the motion is such that the cable is always close to its equilibrium state.

Section 3.2.1.1 introduces this model and the corresponding equations. An experimental validation of the model is presented in Section 3.2.1.2, and Section 3.2.1.3 provides a conclusion and a discussion.

#### 3.2.1.1 Catenary equations for cable modeling

**3.2.1.1.1 Static hanging cable** Let us consider an homogeneous hanging cable with fixed length and fixed ends, only subject to its own weight. It can be shown that it conforms to a catenary shape defined in a vertical plane (Leibniz, 1691). The catenary curve is defined in the orthogonal frame $\mathcal{F}_{2D}$ with origin the curve's lowest point by the equation:

$$\forall x \in \mathbb{R}, y = \frac{1}{C}[\cosh(Cx) - 1] \tag{3.1}$$

where $C \in \mathbb{R}^*$ and $(x, y)$ denote the coordinates of a 2D point in $\mathcal{F}_{2D}$.

Figure 3.1 shows the parameterization of a catenary shape for a cable with fixed length $L$ whose attachment points are $\mathbf{P}_{a_1}$ and $\mathbf{P}_{a_2}$. Let us define a direct Cartesian frame $\mathcal{F}_0$ for the catenary, such that its center is $\mathbf{P}_0$, the lowest point of the catenary, $\mathbf{z}_0$ is vertical, and $\mathbf{y}_0$ is orthogonal to the vertical plane that contains $\mathbf{P}_{a_1}$ and $\mathbf{P}_{a_2}$. The coordinates $(X, Y, Z)$ of the cable points expressed in frame $\mathcal{F}_0$ are:

$$\forall X \in [-D, D + \Delta D], \begin{cases} Y = 0 \\ Z = \frac{1}{C}[\cosh(CX) - 1] \end{cases} \tag{3.2}$$

$C \in \mathbb{R}^*$ can be expressed geometrically as a function of parameters the cable's sag $H$, the difference of elevation of the attachment points $\Delta H$ and the cable length

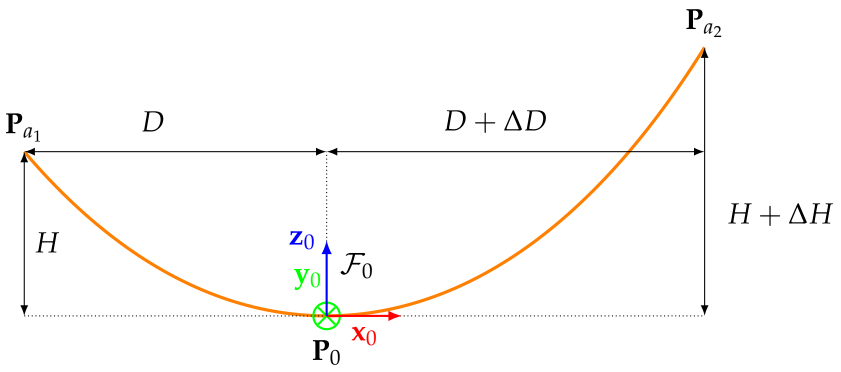


FIGURE 3.1: Catenary shape parameterization

$L$. $H$ and $\Delta H$ are shown in Figure 3.1.

$$C = \frac{2(2H + \Delta H + 2L\sqrt{H\frac{H+\Delta H}{L^2-\Delta H^2}})}{L^2 - (2H + \Delta H)^2} \tag{3.3}$$

$C$ can also be defined from the laws of physics as:

$$C = \frac{\mu g}{T_0} \tag{3.4}$$

where $\mu$ is the cable's linear mass, $g$ is the value of Earth gravity and $T_0$ is the horizontal tension, which is constant throughout the cable.

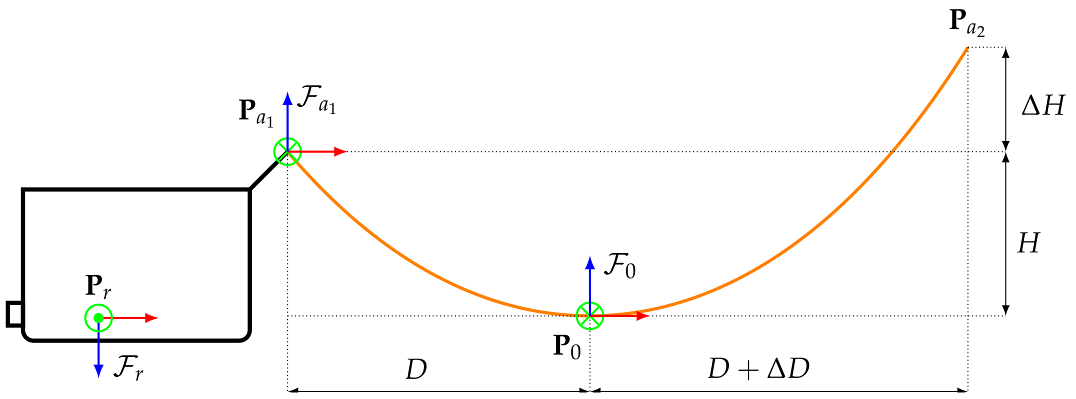
It can be shown similarly that this model extends to an homogeneous hanging cable with fixed length and fixed ends in a fluid, only subject to its own weight and to Archimedes' buoyancy with a non zero resulting force, with:

$$C = \frac{(\mu - v_l \rho) g}{T_0} \tag{3.5}$$

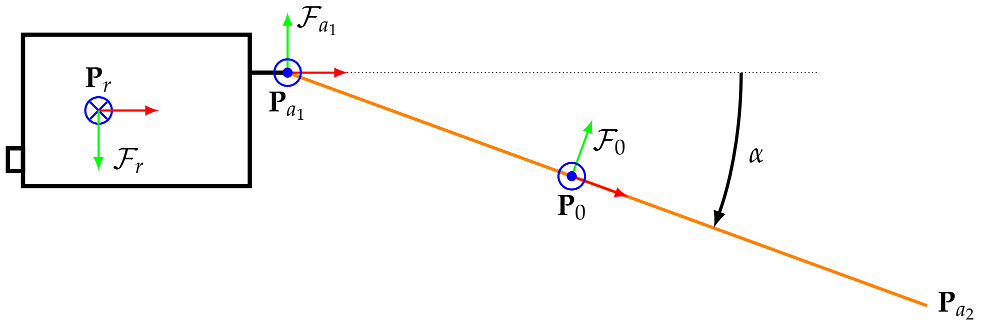
where $v_l$ and $\rho$ denote respectively the cable's linear volume and the fluid's density.

**3.2.1.1.2 Static hanging cable in a robot chain** We consider a subsystem of a robot chain, composed of a robot and the section of cable in front of it. Let us introduce two new coordinate frames, namely $\mathcal{F}_r$ and $\mathcal{F}_{a_1}$, where $\mathcal{F}_r$ is the robot frame in SNAME convention and $\mathcal{F}_{a_1}$ has origin $\mathbf{P}_{a_1}$ and axes aligned with $\mathcal{F}_r$. The rotation from $\mathcal{F}_{a_1}$ to $\mathcal{F}_0$ is considered to be a pure yaw rotation of magnitude $\alpha$ about the vertical axis. If the cable conforms into a catenary curve, its three-dimensional shape can be described by the set of parameters $\{H, \Delta H, \alpha\}$, given its fixed length $L$. This system and its parameters are represented in Figure 3.2.

**3.2.1.1.3 Inclined hanging cable in water** As soon as the cable is moved in the water, it may happen that the current or the movement of the ends leads the cable to a transitional position in an inclined plane. Let the inclined catenary curve model be defined as a 3-dimensional catenary included in a non-vertical plane $\mathcal{P}$, as represented in Figure 3.3. $\mathcal{F}_w$ is a world frame with origin $\mathbf{P}_w$ and $\mathbf{z}_w$ vertical and pointing upwards. Frame $\mathcal{F}_0$ is defined as previously, such that its center is $\mathbf{P}_0$,



(A) Side view



(B) Top view

FIGURE 3.2: Static hanging cable connected to a robot

the lowest point of the catenary, $\mathbf{z}_0$ is vertical upwards, and $\mathbf{y}_0$ is orthogonal to the vertical plane that contains $\mathbf{P}_{a_1}$ and $\mathbf{P}_{a_2}$. The transformation from $\mathcal{F}_0$ to $\mathcal{F}_w$ is then:

$$ {}^{w}\mathbf{T}_0 = \begin{bmatrix} \mathbf{R}_z(\psi) & {}^{w}\mathbf{P}_0 \\ \mathbf{0}_{1\times 3} & 1 \end{bmatrix} \tag{3.6} $$

where $\mathbf{R}_z(\psi)$ denotes a yaw rotation of angle $\psi$ and ${}^{w}\mathbf{P}_0$ is the coordinate vector of $\mathbf{P}_0$ in $\mathcal{F}_w$.

Let $\mathcal{F}_p$ be defined with origin $\mathbf{P}_0$, such that $\mathbf{x}_p = \mathbf{x}_0$ and $\mathbf{z}_p$ is the normalized projection of $\mathbf{z}_0$ in plane $\mathcal{P}$. $\mathbf{y}_p$ is then normal to $\mathcal{P}$. The transformation between frames $\mathcal{F}_p$ and $\mathcal{F}_0$ is then a pure rotation of angle $\gamma$ around $\mathbf{x}_0$:

$$ {}^{0}\mathbf{T}_p = \begin{bmatrix} \mathbf{R}_x(\gamma) & \mathbf{0}_{3\times 1} \\ \mathbf{0}_{1\times 3} & 1 \end{bmatrix} \tag{3.7} $$

The coordinates of the inclined catenary curve contained in plane $\mathcal{P}$ are given in frame $\mathcal{F}_p$ by $({}^{p}X, {}^{p}Y, {}^{p}Z)$ where:

$$ \begin{cases} {}^{p}Y = 0 \\ {}^{p}Z = \frac{1}{C}[\cosh(C\,{}^{p}X) - 1] \end{cases} \tag{3.8} $$

In frame $\mathcal{F}_w$, the coordinate vector of a point $\mathbf{P}$ belonging to the inclined catenary curve is finally ${}^{w}\mathbf{P} = {}^{w}\mathbf{T}_0\,{}^{0}\mathbf{T}_p\,{}^{p}\mathbf{P}$. Alternatively, an inclined catenary curve is fully defined by the set of parameters $(\gamma, C, {}^{w}X_0, {}^{w}Z_0)$ where ${}^{w}X_0$ and ${}^{w}Z_0$ denote the

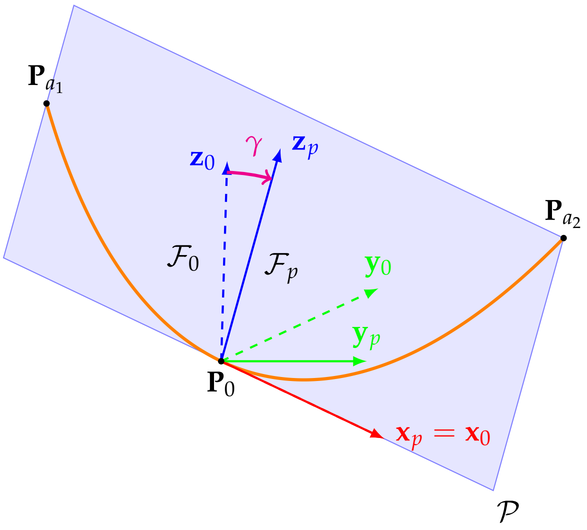


FIGURE 3.3: Inclined catenary model. Frames $\mathcal{F}_p$ and $\mathcal{F}_0$ are represented in solid and dashed lines respectively.

$x$ and $y$-coordinates of $\mathbf{P}_0$ in $\mathcal{F}_w$. ${}^wY_0$ can be computed from ${}^wX_0$ and ${}^wZ_0$ based on the constraint $\mathbf{P}_0 \in \mathcal{P}$.

Note that the inclined catenary model is not a physical model. It is although interesting as a potential approximation for describing the shape of a non-neutrally buoyant cable in presence of currents, or in a transitional state. It is therefore necessary to study the cable shape in an inclined plane in order to determine if its deviation from a catenary curve is significant. In the following, we make the distinction between the vertical catenary model, which lies in a vertical plane, and the inclined catenary model.

#### 3.2.1.2 Catenary model experimental validation

The 3D shape of real-life underwater, negatively-buoyant, moving cables is tracked and compared to the vertical and inclined catenary models described in Section 3.2.1.1. This section presents the methodology used, describes the experiments and presents the results of model evaluation.

**3.2.1.2.1 Numerical estimation method** Let a set of $N$ 3D points that belong to a cable be defined as $\{\mathbf{P}_i\}_{i\in\{1\ldots N\}}$. The plane $\mathcal{P}$ is defined as the plane that best contains the points $\{\mathbf{P}_i\}_{i\in\{1\ldots N\}}$. It is computed by minimizing point-plane distance. Let us introduce $\mathbf{n}$ the unit vector perpendicular to $\mathcal{P}$ and $d \in \mathbb{R}$ such that:

$$\forall\, \mathbf{P} \in \mathbb{R}^3, \mathbf{P} \in \mathcal{P} \iff \mathbf{P} \cdot \mathbf{n} + d = 0 \tag{3.9}$$

where $\cdot$ denotes the Euclidian inner product. The distance of a point $\mathbf{P}_i$ to plane $\mathcal{P}$ is:

$$d_i(\mathbf{n}, d) = \frac{(\mathbf{P}_i \cdot \mathbf{n} + d)}{\sqrt{\mathbf{n} \cdot \mathbf{n}}} \tag{3.10}$$

Plane parameters $\mathbf{n}$ and $d$ are estimated by minimizing the cost function:

$$f_{\mathcal{P}}(\mathbf{n}, d) = \sum_{i=1}^{N} d_i(\mathbf{n}, d)^2 \tag{3.11}$$

using the `fminsearch` Matlab function which implements the Nelder-Mead simplex algorithm as described in (Lagarias et al., 1998). An initial guess is computed using three arbitrarily chosen tracked points.

In a second time, the points $\{\mathbf{P}_i\}_{i \in \{1...N\}}$ are projected on $\mathcal{P}$. Let us denote $\{\mathbf{P}_{i,\text{proj}}\}_{i \in \{1...N\}}$ the projections of points $\{\mathbf{P}_i\}_{i \in \{1...N\}}$ in plane $\mathcal{P}$.

$$\forall\, i \in \{1...N\}, \mathbf{P}_{i,\text{proj}} = \mathbf{P}_i - (\mathbf{P}_i \cdot \mathbf{n} + d)\mathbf{n} \tag{3.12}$$

Unit vectors $\mathbf{x}_p$, $\mathbf{y}_p$ and $\mathbf{z}_p$ are then computed as follows:

$$\mathbf{y}_p = \mathbf{n} \tag{3.13}$$

$$\mathbf{z}_p = \frac{\mathbf{z}_w - (\mathbf{z}_w \cdot \mathbf{n} + d)\mathbf{n}}{\|\mathbf{z}_w - (\mathbf{z}_w \cdot \mathbf{n} + d)\mathbf{n}\|} \tag{3.14}$$

$$\mathbf{x}_p = \mathbf{y}_p \times \mathbf{z}_p \tag{3.15}$$

where × denotes the vector cross product and $\|\cdot\|$ is the Euclidian norm.

The rigid transformation matrix between frames $\mathcal{F}_p$ and $\mathcal{F}_w$ is then

$$^{p}\mathbf{T}_w = \begin{bmatrix} ^{p}\mathbf{R}_w & ^{p}\mathbf{P}_w \\ \mathbf{0}^T & 1 \end{bmatrix} \tag{3.16}$$

where

$$^{p}\mathbf{R}_w = \begin{bmatrix} ^{w}\mathbf{x}_p & ^{w}\mathbf{y}_p & ^{w}\mathbf{z}_p \end{bmatrix}^T \tag{3.17}$$

Let $^{p}\mathbf{P}_{i,\text{proj}} = \begin{bmatrix} ^{p}X_{i,\text{proj}} & ^{p}Y_{i,\text{proj}} & ^{p}Z_{i,\text{proj}} \end{bmatrix}^T$ and $^{w}\mathbf{P}_{i,\text{proj}}$ be the coordinates of a point $\mathbf{P}_{i,\text{proj}}$ in $\mathcal{F}_p$ and $\mathcal{F}_w$ respectively. By definition,

$$^{p}\mathbf{P}_{i,\text{proj}} = {^{p}\mathbf{T}_w}\, {^{w}\mathbf{P}_{i,\text{proj}}} \tag{3.18}$$

The inclined catenary parameters $^{p}\mathbf{P}_w$ and $C$ are estimated by minimizing the cost function:

$$f_C(^{p}\mathbf{P}_w, C) = \frac{\sum_{i=1}^{N} (^{p}Z_{i,\text{proj}}(^{p}\mathbf{P}_w) - \frac{1}{C}[\cosh(C\, ^{p}X_{i,\text{proj}}(^{p}\mathbf{P}_w)) - 1])^2}{N} \tag{3.19}$$

An initial guess is computed by calculating the vertical catenary curve between the pair of visible tracked points that are the closest to the cable's ends. Again, the optimization uses `fminsearch` Matlab function.

**3.2.1.2.2 Evaluation metrics** Three indicators have been selected to characterize the catenary shape of the cable:

- the root mean square (RMS) distance $e_{\mathcal{P}}$ between the tracked points and plane $\mathcal{P}$.

- the inclination of the plane $\mathcal{P}$ with respect to the vertical, *i.e.* the absolute value $\gamma$ of the angle between the normal to the plane $\mathcal{P}$ and the normal to the vertical plane with the same yaw orientation as plane $\mathcal{P}$ in $\mathcal{F}_w$.
- the RMS distance $e_\mathcal{C}$ between the projection of points $\{\mathbf{P}_i\}_{i\in\{1\ldots N\}}$ in plane $\mathcal{P}$ and the points with same $x$-coordinate in frame $\mathcal{F}_p$ that belong to the fitted inclined catenary curve.

$e_\mathcal{P}$ is defined by the following relation:

$$e_\mathcal{P} = \sqrt{\frac{\sum_{i=1}^{N} d_i^2}{N}} \tag{3.20}$$

where $d_i$ is the point-plane distance as defined in (3.10), and $e_\mathcal{C}$ can be expressed as follows:

$$e_\mathcal{C} = \sqrt{\frac{\sum_{i=1}^{N} \left({}^pZ_{i,\text{proj}} - \frac{1}{C}\left[\cosh\left(C\,{}^pX_{i,\text{proj}}\right) - 1\right]\right)^2}{N}} \tag{3.21}$$

**3.2.1.2.3 Experimental set-up** Figure 3.4 represents the experimental setup. A cable is deployed in a pool, with one end tied to a fixed point and the other end tied to a stick. The stick is then moved by an operator. The cable is equipped with regularly spaced reflective markers that are tracked by a motion capture system (see Figure 3.4). This experiment is reproduced with two cables of different linear mass $\mu$, diameter $\Phi$, and material: a metal chain and a thin rope, denoted respectively Cable 1 and Cable 2. Table 3.1 gives their respective characteristics. $\widehat{L}$ is the approximate cable length.

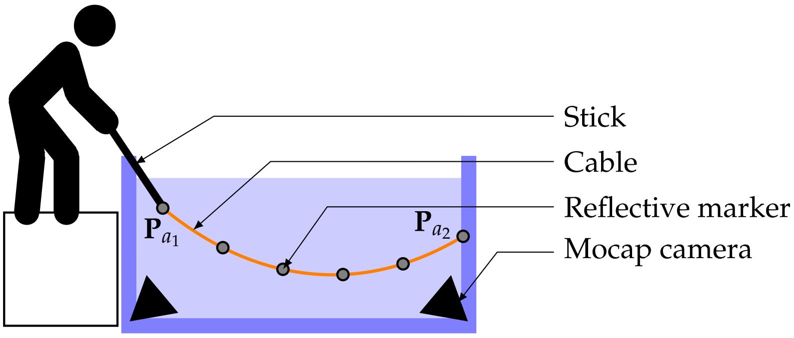


FIGURE 3.4: Experimental setup. Attachment point $P_{a_2}$ is fixed with respect to the pool, while $P_{a_1}$ is attached on a mobile stick.

The experiments are conducted in a pool, in open water, at shallow depth (<4m). A 166.98s tracking sequence was recorded for Cable 1, as well as a 252.68s long sequence for Cable 2.

**3.2.1.2.4 Results** Figures 3.5 and 3.6 illustrate the typical behavior of the cables during subsequences that include movements of the tether's end point in various directions with slow and rapid motion, and correspond to time periods when the cable

TABLE 3.1: Characteristics of the cables. Since Cable 1 is a chain, the given diameter $d$ corresponds to the width of a link.

| | **Cable 1** | **Cable 2** |
|---|---|---|
| $\mu$ when wet (kg/m) | $9.20 \times 10^{-}2$ | $2.26 \times 10^{-}2$ |
| $\widehat{L}$ (mm) | 3500 | 2000 |
| $\Phi$ (mm) | 10 | 5 |
| Material | Steel | Fabric |
| Max. number of tracked points | 12 | 9 |
| Spacing between markers (mm) | 245 | 200 |

was well tracked by the motion capture system. Displayed parameters are the computed values of $\gamma$, $e_{\mathcal{P}}$, $e_{\mathcal{C}}$ and catenary parameter $C$, the number $N$ of tracked points and which tracked points are visible, as well as the velocity ${}^{w}\mathbf{v} = \begin{bmatrix} {}^{w}v_x & {}^{w}v_y & {}^{w}v_z \end{bmatrix}^T$ and acceleration ${}^{w}\mathbf{a} = \begin{bmatrix} {}^{w}a_x & {}^{w}a_y & {}^{w}a_z \end{bmatrix}^T$ of the mobile end of the cable in $\mathcal{F}_w$ and the distance $d$ between the cable's ends for Cable 1 and Cable 2 respectively.

One can observe that the greater the distance $d$, the smaller the parameter $C$. Indeed, an increased distance between the end points implies more tension in the cable and thus a lower $C$ according to equations (3.4) and (3.5). The peaks in the curves of $\gamma$, $e_{\mathcal{P}}$ and $e_{\mathcal{C}}$ occur when only few points are visible or when the set of visible tracked points changes. The set of visible points changes when the tracking system stops detecting some points or detects new ones. The more visible tracked points, the more reliable the catenary likelihood indicators.

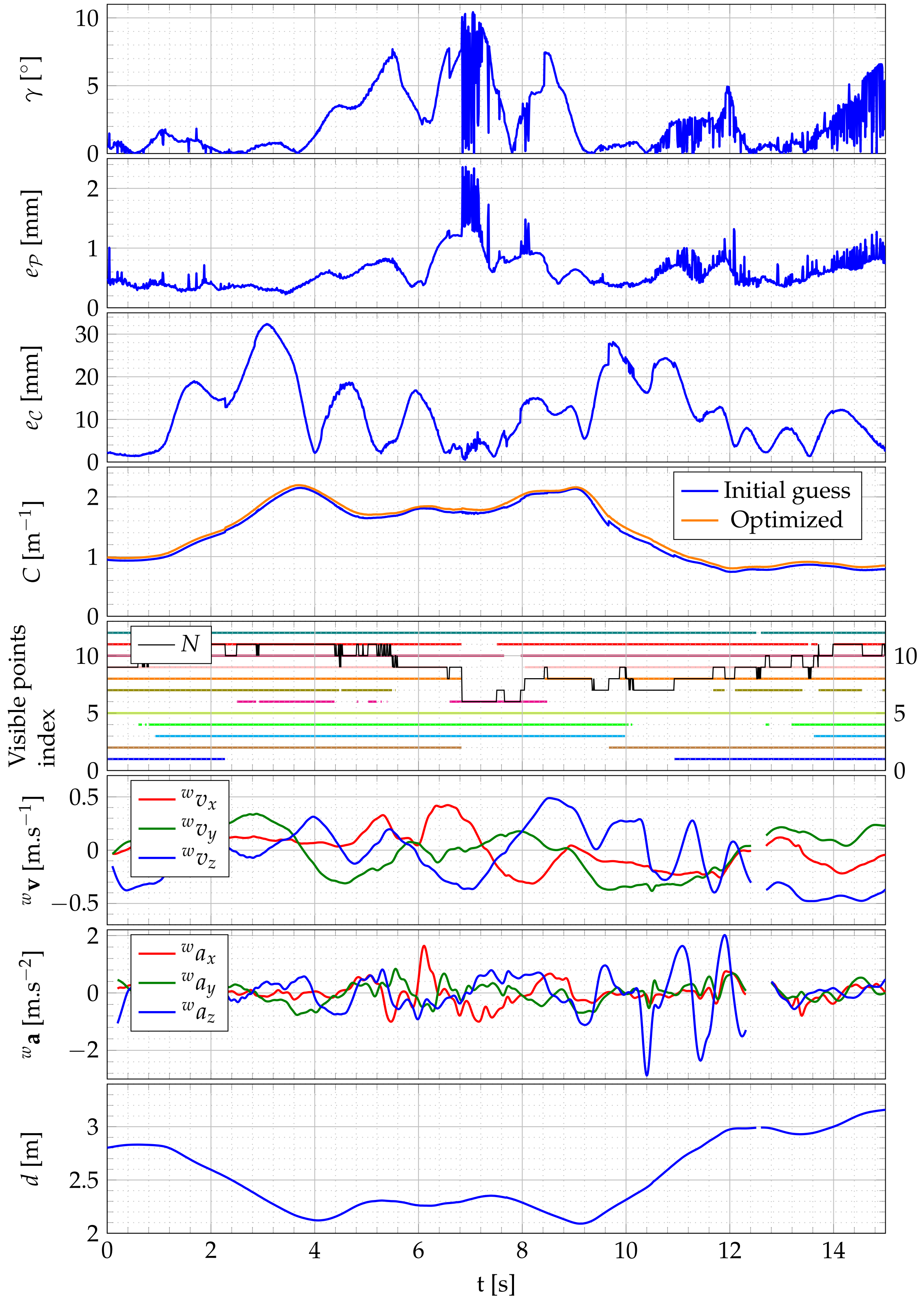


FIGURE 3.5: Variations of $\gamma$, $e_{\mathcal{P}}$, $e_{\mathcal{C}}$, $C$, $N$, tracked points visibility, $^{w}\mathbf{v}$, $^{w}\mathbf{a}$ and $d$ during a part of the whole Cable 1 tracking sequence: $t \in [0, 15]$ s. The tracked point on the mobile end has index 12.

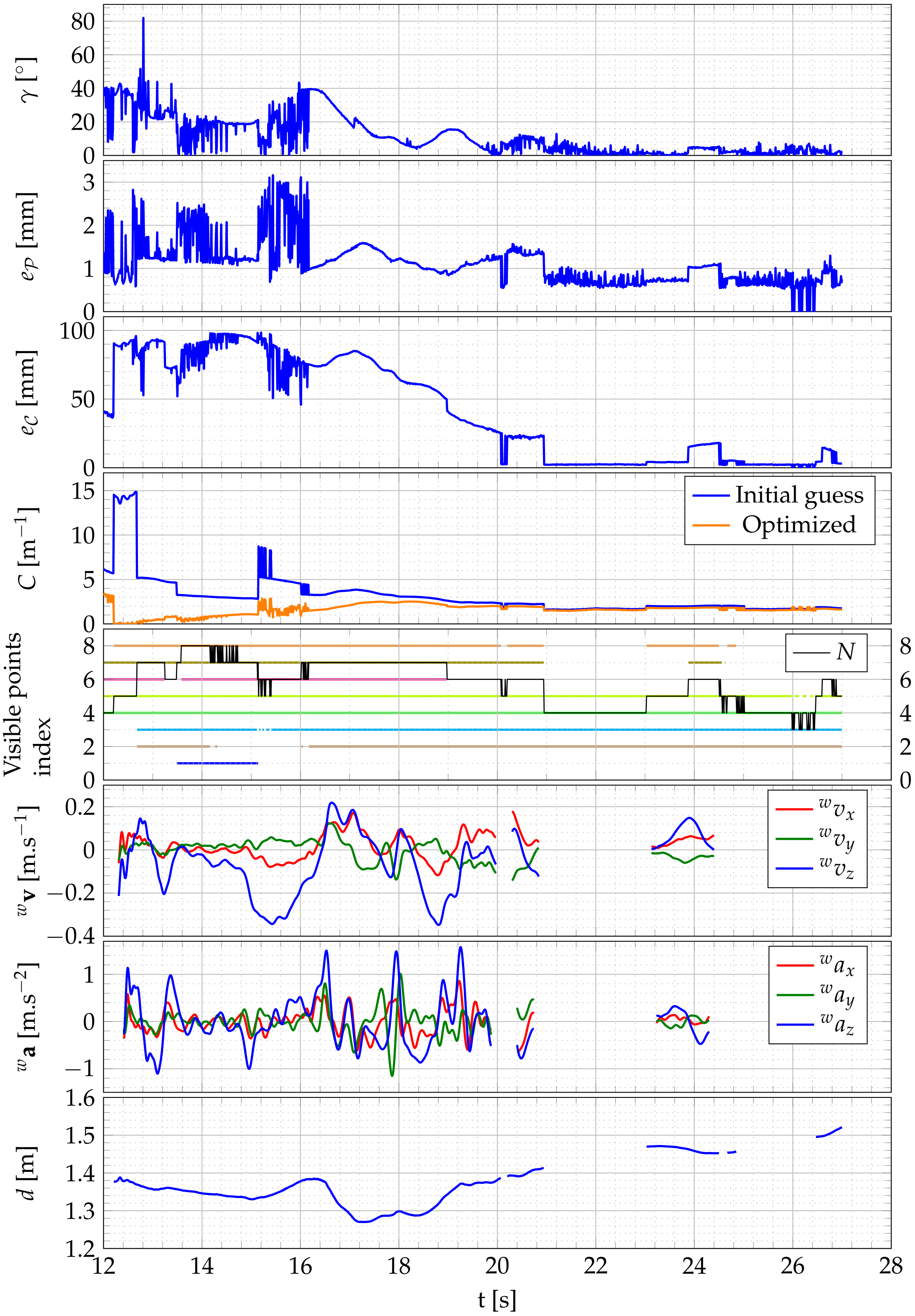


FIGURE 3.6: Variations of $\gamma$, $e_{\mathcal{P}}$, $e_{\mathcal{C}}$, $C$, $N$, tracked points visibility, ${}^{w}\mathbf{v}$, ${}^{w}\mathbf{a}$ and $d$ during a part of the whole Cable 2 tracking sequence: $t \in [12, 27]$ s. The tracked point on the mobile end has index 8.

Table 3.2 gives the mean, minimum, maximum, median and standard deviation $\sigma$ of indicators $\gamma$, $e_{\mathcal{P}}$ and $e_{\mathcal{C}}$ for Cable 1's full tracking sequence. The same results for Cable 2's full sequence are displayed in Table 3.3. Only the values computed when $N > 6$ are considered reliable and included in these statistics.

TABLE 3.2: Results for Cable 1

| | **$\gamma$ (°)** | **$e_{\mathcal{P}}$ (mm)** | **$e_{\mathcal{C}}$ (mm)** |
|---|---|---|---|
| mean | 2.67 | 0.60 | 12.6 |
| min | $2.94 \times 10^{-5}$ | 0.08 | 0.19 |
| max | 29.1 | 4.55 | 286 |
| median | 1.70 | 0.45 | 4.38 |
| $\sigma$ | 3.08 | 0.42 | 27.67 |

TABLE 3.3: Results for Cable 2

| | **$\gamma$ (°)** | **$e_{\mathcal{P}}$ (mm)** | **$e_{\mathcal{C}}$ (mm)** |
|---|---|---|---|
| mean | 16.52 | 1.14 | 52.2 |
| min | $6.81 \times 10^{-4}$ | 0.10 | 0.30 |
| max | 82.0 | 3.65 | 500 |
| median | 11.3 | 1.09 | 19.2 |
| $\sigma$ | 15.1 | 0.46 | 69.1 |

On the one hand, the results in Table 3.2 show that Cable 1's shape stays very close to the catenary shape along the entire sequence. The mean, median and standard deviation of $e_{\mathcal{P}}$ are very small with respect to the cable length, with an order of magnitude of $5 \times 10^{-4}$ m (cable length of approximately 3.5m). This validates the assumption of a planar cable. The mean, median and standard deviation of angle $\gamma$ are also small, with an order of magnitude of 1°, showing that this plane can be well approximated as vertical. The maximum value for $\gamma$ in the whole sequence is about 30°, but the small standard variation of about 3° indicates that such an important angle is rare. This extremum corresponds to a fast lateral movement of the cable. Finally, $e_{\mathcal{C}}$ and their variations are small on the scale of Cable 1's length, with less than 1% of its length, indicating that it can be well approximated by a catenary shape in the cable plane. Even the maximal value of $e_{\mathcal{C}}$ represents only approximately 10% of Cable 1's length. Cable 1 can then be well approximated by the vertical catenary model, even with its endpoints moving. In addition, the small maximal values and standard deviation of $e_{P}$ and $e_{C}$ show that even if the cable's plane may be inclined from the vertical, it can then be well approximated by an inclined catenary curve.

On the other hand, the results in Table 3.3 show that the vertical catenary model is less accurate for Cable 2. $e_{\mathcal{P}}$ and its variations have the same order of magnitude than those of Cable 1. This indicates that the cable can be considered planar most of the time, but the mean value and variations of $\gamma$ show that this plane cannot be considered vertical in the general case. The maximum value of $\gamma$ shows that the cable's plane even came close to the horizontal plane at some point in the sequence. The mean, median and standard deviation of $e_{\mathcal{C}}$ are about four times higher than those of Cable 2, showing that even in the cable's plane, Cable 2 is much further from the vertical catenary model than Cable 1. Qualitatively, the changes in direction of Cable 2's end point movements create inflection points on the cable, making its shape more different from a catenary at these moments. This does not occur with Cable 1 during the tracking sequence. One can notice that the mean, median and standard deviation of $e_C$ are however still small with respect to the cable length. This shows that even a light cable like Cable 2 can be modelled by an inclined catenary curve as a first estimate. Such an approximation is nevertheless less accurate than modeling Cable 1 as a catenary.

Finally, these results show that the catenary curve is a good approximation of an underwater cable with moving endpoints if it has a 'heavy enough' linear mass. Cables with lower linear mass may be better approximated by an inclined catenary by assuming that the cable's plane may be inclined at an angle $\gamma$ from the vertical, but with less accuracy. It is then a question of design to guarantee that the cable conforms to a catenary curve in a vertical plane. It must also be a compromise with keeping the cable light enough not to impede the robot's motion. In addition, it is necessary to take into account the water flow generated by the thrusters which can move the part of the cable located in their vicinity. The cable should thus be placed out of the thruster's flow as much as possible. Avoiding movements with abrupt changes of direction will also help to keep the model valid.

#### 3.2.1.3 Conclusions

This section introduced the catenary model for underwater weighing cables in a vertical or inclined plane. Validity of these two models was evaluated based on the motion tracking of two underwater cables with different linear mass while moving one of their endpoints. The tracking data was then used to estimate the plane that best contained the cable, and the catenary shape that was the closest to the projection of the cable in this plane. Different metrics were used to characterize the catenary likelihood of the two cables. The results show that the catenary equation still gives a good shape approximation for underwater cables having a large enough linear mass, even with their endpoints moving dynamically. The shape of cables with too light weight to neglect hydrodynamics can be approximated using an inclined catenary model, but with less accuracy.

Thus, the selection of such a cable when designing an underwater tethered system can simplify the cable's shape estimation by modelling it by a catenary curve, while maintaining a good accuracy. The choice of the cable has to be a compromise between a cable heavy enough to conform into a catenary and light enough not to impede the robot's motion. Avoiding abrupt changes of direction in the ROV's motion will also contribute in keeping the model valid.

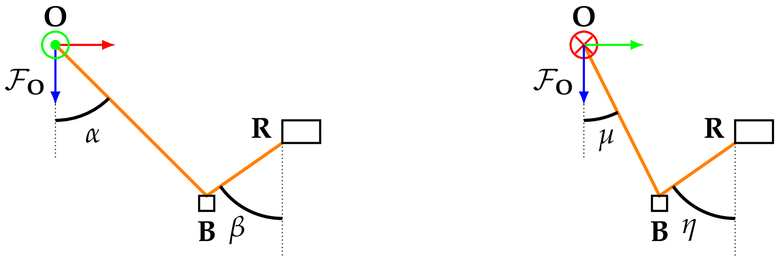
### 3.2.2 Straight-line model

This section considers a system composed of a ROV tethered to a fixed point or another vehicle and whose cable is constrained into a piecewise linear shape by the addition of a sliding buoy or ballast. While a fixed ballast or buoy placed on a cable can only stretch one part of it, in the current system, these elements can move freely on the cable and therefore always find their position at the lowest or highest point corresponding to a minimum potential energy, where it stretches both parts of the cable simultaneously.

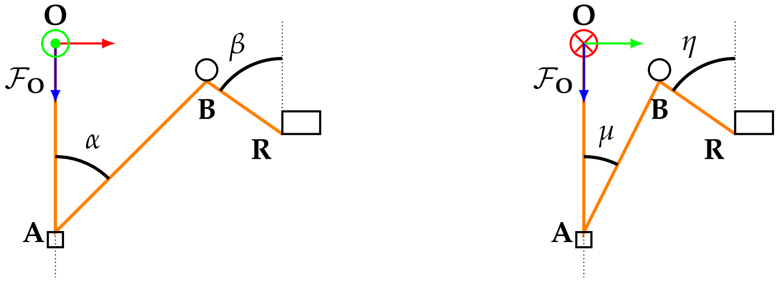
In this system, the cable can be modeled by a straight line shape which can be expressed from cable segments three-dimensional orientation and from the relative depth of the cable attachment points. The position of the ROV with respect to the other cable attachment point is therefore a function of these parameters. The specificity of this model is to express the position of the ROV as a function of the shape of its cable, which can be deduced from a small number of model parameters. The cable can therefore be used as a localization system.

The assumptions, application scope and model equations have been developed by Christophe Viel, CNRS-LabSTICC, Brest, France, and are not a contribution of this thesis. There are are nonetheless essential to introduce further developments based on this straight-line model in Sections 3.3 and 3.4 which belong to the contributions of this thesis. These assumptions, scope and model are therefore given in Appendix B.

We consider a system composed of a ROV tethered to another vehicle or to a fixed point by its umbilical, where these attachment points are denoted **R** and **O** respectively. The first part of the cable from **O** is constrained to be vertical down to an anchor placed in **A**, and a sliding buoy or ballast is placed along the cable between **A** and **R**, in **B**. The position **R** of the ROV can be deduced from cable length $L_t$ between **A** and **R**, anchor depth $l_0$, relative depth ${}^{\mathbf{O}}z_{\mathbf{R}}$ between **R** and **O** and a set of four angles $\alpha$, $\beta$, $\mu$ and $\eta$ represented in Figure 3.7 for the sliding buoy and ballast configurations.



(A) Sliding ballast in plane $(\mathbf{O}, \mathbf{x_O}, \mathbf{z_O})$ (B) Sliding ballast in plane $(\mathbf{O}, \mathbf{y_O}, \mathbf{z_O})$



(C) Sliding buoy in plane $(\mathbf{O}, \mathbf{x_O}, \mathbf{z_O})$ (D) Sliding buoy in plane $(\mathbf{O}, \mathbf{y_O}, \mathbf{z_O})$

FIGURE 3.7: Single sliding element system. Figures 3.7a and 3.7b present a single-ballast configuration with $\mathbf{A} = \mathbf{O}$, and Figures 3.7c and 3.7d present a single-buoy configuration.

## 3.3 Model parameters measurement

In order to use the cable models developed in Section 3.2 for the the online state estimation of underwater tethered robotic systems, model parameters must be measured from embedded sensors. This is the focus of the current section. Section 3.3.1 described cable attachment point depth measurement, which is used for both models. Sections 3.3.2 and 3.3.3 present parameter measurements for the catenary model and straight-line model respectively.

Regarding the catenary model, the following assumptions are made in accordance with conclusions of Section 3.2.1:

1. The cable is heavy enough to neglect the hydrodynamic forces applied to it;
2. Cable ends motion is free from sudden changes of direction;

so that the cable can be modeled by a vertical catenary. The inclined model is not used in this section, and the vertical catenary model is simply referred to as 'catenary'. Note that these assumptions are not necessary for th straight-line model.

### 3.3.1 Cable attachment point depth measurement

The depth of the attachment point of the cable on the ROV can be measured using a water pressure sensor and an IMU embedded in the ROV. Let $\mathbf{P}_a$ denote the attachment point. Depending on the model and on the cable end considered, $\mathbf{P}_a$ can be equal to $\mathbf{P}_{a_1}$ or $\mathbf{R}$, or to $\mathbf{P}_{a_2}$ or $\mathbf{O}$ if these points are placed on a ROV. The pressure sensor and the IMU are placed in $\mathbf{P}_d$ and $\mathbf{P}_{i_r}$ respectively. The IMU frame is denoted $\mathcal{F}_{i_r}$. An additional frame $\mathcal{F}_d$ is defined with origin $\mathbf{P}_d$ and same orientation as $\mathcal{F}_{i_r}$. This system is represented in Figure 3.8.

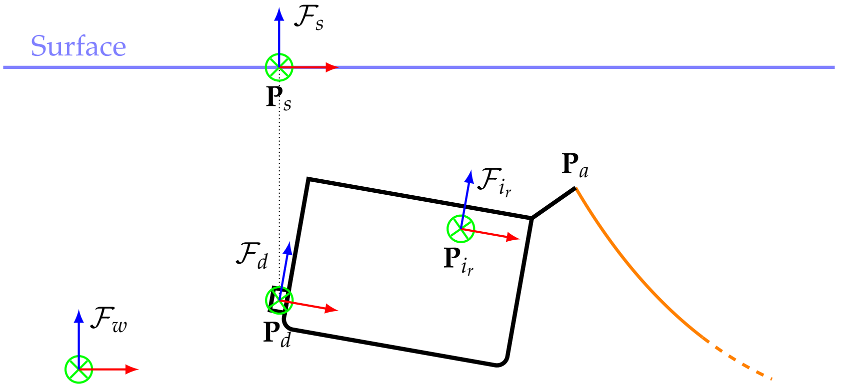


FIGURE 3.8: System involved in attachment point depth measurement

Let $P_{\mathbf{P_d}}$ denote the water pressure measured at $\mathbf{P}_d$.

$$P_{\mathbf{P}_d} = P_{\text{water}} + P_{\text{athm}} + P_b \tag{3.22}$$

where $P_{\text{athm}}$ is the atmospheric pressure, $P_b$ is a sensor bias, and $P_{\text{water}}$ is the pressure exerted by the water column, with:

$$P_{\text{water}} = \rho_{\text{water}} * g * d_{\mathbf{P}_d} \tag{3.23}$$

$\rho_{\text{water}}$ is the volumetric mass of water, $g$ is the value of gravity on Earth and $d_{\mathbf{P}_d}$ is the depth of measurement point $\mathbf{P}_d$ with respect to the surface. Let us define $P_0 = P_{\text{athm}} + P_b$, which is the pressure returned by the pressure sensor at zero depth. One can get:

$$P_{\text{water}} = \rho_{\text{water}} * g * d_{\mathbf{P}_d} - P_0 \tag{3.24}$$

hence:

$$d_{\mathbf{P}_d} = \frac{P_{\mathbf{P}_d} - P_0}{\rho_{\text{water}} * g} = \frac{P_{\mathbf{P}_d}}{\rho_{\text{water}} * g} + c_2 \tag{3.25}$$

where $c_2 = \frac{-P_0}{\rho_{\text{water}} * g}$.

The position ${}^d\mathbf{P}_a$ of $\mathbf{P}_a$ in frame $\mathcal{F}_d$ is fixed, and calibrated. The embedded IMU of the robot measures its orientation ${}^w\mathbf{R}_{r_i}$ with respect to a world frame $\mathcal{F}_w$ with its z-axis vertical, upwards. Depth at $\mathbf{P}_{a_1}$ can be deduced from $d_{\mathbf{P}_d}$ and ${}^w\mathbf{R}_{r_i}$. Let us define frame $\mathcal{F}_s$ with origin the surface point located vertically from $\mathbf{P}_d$ and with same orientation as $\mathcal{F}_w$, as illustrated in Figure 3.8.

Depth at $\mathbf{P}_a$ is then:

$$d_{\mathbf{P}_a} = -{}^s z_{\mathbf{P}_a} \tag{3.26}$$

where ${}^s z_{\mathbf{P}_a}$ is the $z$-coordinate of $\mathbf{P}_a$ in $\mathcal{F}_s$. In homogeneous coordinates, one gets:

$${}^s\overline{\mathbf{P}}_a = {}^s\mathbf{T}_d \, {}^d\overline{\mathbf{P}}_a \tag{3.27}$$

where ${}^s\mathbf{T}_d$ is the rigid transformation between $\mathcal{F}_d$ et $\mathcal{F}_s$. According to the definition of frames $\mathcal{F}_d$, $\mathcal{F}_s$, $\mathcal{F}_{r_i}$ and $\mathcal{F}_w$ one gets:

$${}^s\mathbf{T}_d = \left[\begin{array}{c|c} {}^s\mathbf{R}_d & \begin{matrix} 0 \\ 0 \\ d_{\mathbf{P}_d} \\ 1 \end{matrix} \end{array}\right] = \left[\begin{array}{c|c} {}^w\mathbf{R}_{r_i} & \begin{matrix} 0 \\ 0 \\ d_{\mathbf{P}_d} \\ 1 \end{matrix} \end{array}\right] \tag{3.28}$$

and finally:

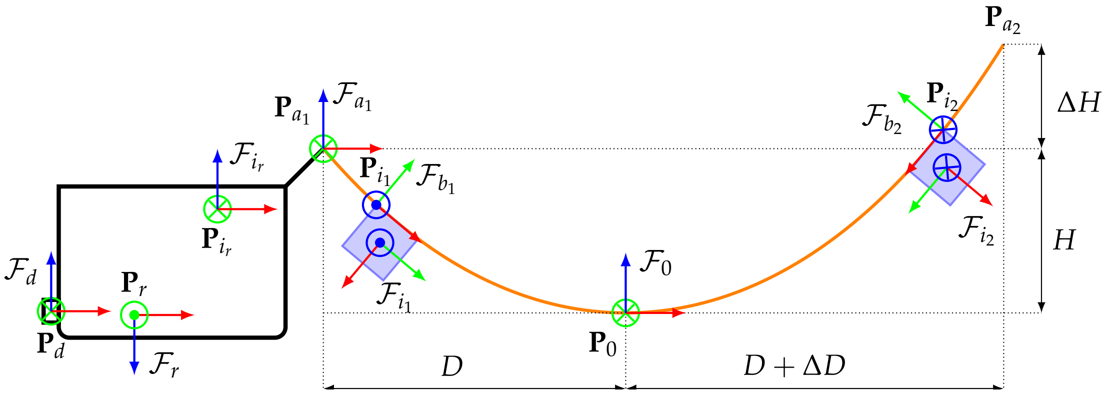
$$d_{\mathbf{P}_a} = -\begin{bmatrix} 0 & 0 & 1 & 0 \end{bmatrix} {}^s\mathbf{T}_d \, {}^d\overline{\mathbf{P}}_a \tag{3.29}$$

### 3.3.2 Catenary parameters measurement

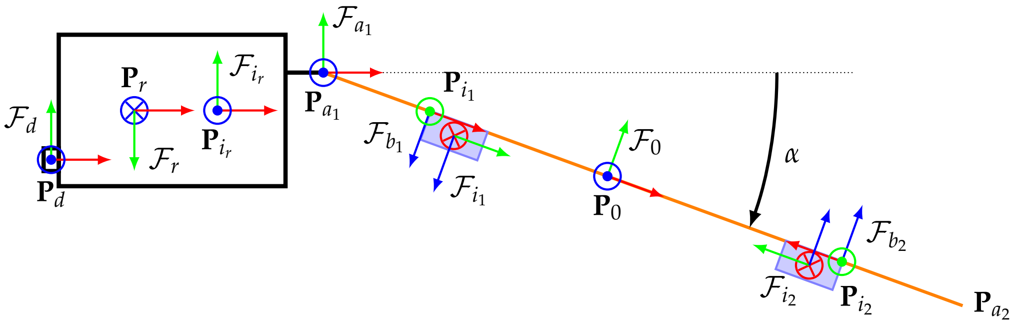
We consider a subsystem of an underwater robot chain which includes a ROV and the section of hanging cable in front of it, which is modeled by a catenary curve. As explained in Section 3.2.1.1, the three-dimensional catenary shape of the cable is defined by three parameters $\{H, \Delta H, \alpha\}$, where $\Delta H$ can is the difference of attachment points depth and can therefore be measured as explained in Section 3.3.1.

Parameters $H$ and $\alpha$ are measured using IMUs placed tangent to the cable, near its ends. This system is represented in Figure 3.9, with the same notations as in Section 3.2.1.1 and 3.3.1. $\mathcal{F}_{i_1}$ and $\mathcal{F}_{i_2}$ are the frames of the IMUs placed on each side of the cable at fixation points $\mathbf{P}_{i_1}$ and $\mathbf{P}_{i_2}$ respectively. Two additional frames $\mathcal{F}_{b_1}$ and $\mathcal{F}_{b_2}$ are defined, with respective origins $\mathbf{P}_{i_1}$ and $\mathbf{P}_{i_2}$, and $x$-axes tangent to the catenary. IMUs are attached to the cable along this axis, around which the sensor can rotate when the friction of the water on the IMU is strong enough to induce a twist in the cable.

The estimation of the $H$ and $\alpha$ uses the IMU embedded onboard the robot and one or both of the cable IMUs. All IMUs are assumed to measure their orientation with respect to the same world frame $\mathcal{F}_w$ with $z$-axis vertical, upwards. The yaw



(A) Side view



(B) Top view

FIGURE 3.9: Subsystem composed of a ROV and its front catenary cable with notation, frame definitions, and 3D parameters.

angle $\alpha$ is determined from the rotation matrix ${}^{r}\mathbf{R}_{b_1}$ or ${}^{r}\mathbf{R}_{b_2}$ with:

$$ {}^{r}\mathbf{R}_{b_k} = ({}^{w}\mathbf{R}_{i_r}\,{}^{i_r}\mathbf{R}_{r})^{T}\,{}^{w}\mathbf{R}_{i_k}\,{}^{i_k}\mathbf{R}_{b_k}, \quad k \in \{1,2\} \tag{3.30} $$

where ${}^{i_k}\mathbf{R}_{b_k}$ and ${}^{i_r}\mathbf{R}_{r}$ are constant calibrated matrices, and ${}^{w}\mathbf{R}_{r}$ and ${}^{w}\mathbf{R}_{i_k}$ are measured by the robot and the catenary IMUs, respectively. $\alpha$ is then such that:

$$ {}^{r}\mathbf{R}_{b_k} = \mathbf{R}_{z}(\alpha)\,\mathbf{R}_{y}(\theta)\,\mathbf{R}_{x}(\phi) \tag{3.31} $$

where $\mathbf{R}_x$, $\mathbf{R}_y$ and $\mathbf{R}_z$ denote elemental rotations around the $x$, $y$ and $z$-axis respectively (see Appendix A.1.1.2).

$\forall\, k \in \{1,2\}$ we define $\beta_k$ as the angle between the cable tangent at $\mathbf{P}_k$, which corresponds to the $x$-axis of $\mathcal{F}_{b_k}$, and the horizontal. Figure 3.10 shows the two cable tangent angles $\beta_k$, at points $\mathbf{P}_{i_1}$ and $\mathbf{P}_{i_2}$ that are placed at distances $L_1$ and $L_2$ from extremities. $R_1$ is the curvilinear distance along the cable from $\mathbf{P}_{i_1}$ to $\mathbf{P}_0$:

$$ R_1 = (L - (L_1 + L_2))\frac{g(\beta_1,\beta_2)}{1 + g(\beta_1,\beta_2)} \tag{3.32} $$

with

$$ g(\beta_1,\beta_2) = \frac{|\tan(\beta_1)|}{|\tan(\beta_2)|} \tag{3.33} $$

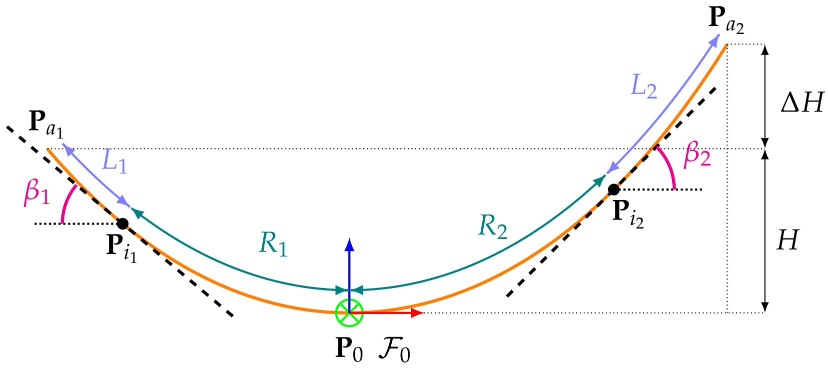


FIGURE 3.10: Introducing catenary tangent angles $\beta_1$ and $\beta_2$.

The catenary parameter $C$ can be obtained with:

$$C = \frac{1}{R_1}|\tan(\beta_1)| \tag{3.34}$$

Equation 3.3 is used to recover $H$ as the only positive root of the second degree polynomial $a_2\ H^2 + a_1\ H + a_0 = 0$ where:

$$\begin{aligned} a_2 &= 4C^2(\Delta H^2 - L^2) \ < \ 0 \\ a_1 &= 4C(\Delta H^2 - L^2)(C\Delta H + 2) \ < \ 0 \\ a_0 &= [C(L^2 - \Delta H^2) - 2\Delta H]^2 > 0 \end{aligned}$$

The catenary parameters can also be estimated with a single tangent, e.g. at $\mathbf{P}_{i_1}$ by approximating $R_1$ with:

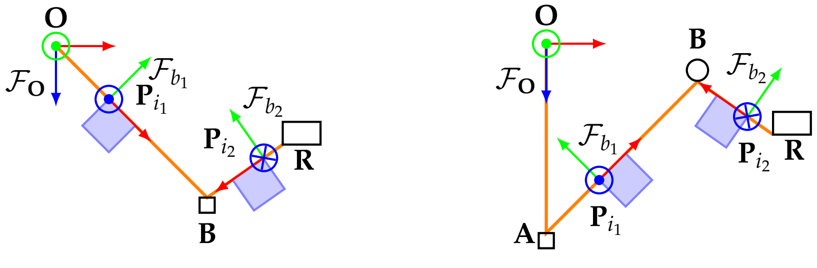
$$R_1 = \frac{(L - \Delta H)}{2} - L_1 \tag{3.35}$$

where the difference of length between $L_1$ and $L_2$ is approximated by the difference of their altitude, what is valid only if segments $\mathbf{P}_{i_1}\mathbf{P}_{a_1}$ and $\mathbf{P}_{i_2}\mathbf{P}_{a_2}$ are close to the vertical or if $\Delta H << L$. Symmetrically, $R_2$ is approximated using the tangent at $\mathbf{P}_{i_2}$. It is worth noticing that an exact expression of $H$ can de derived from a single tangent, but it involves solving a fourth degree equation and using additional conditions to select the correct solution between the roots of the equation, such that the lowest point lies between the attachment points.

### 3.3.3 Straight-line model angle parameters measurement

Similarly to Section 3.3.2, the orientation of a cable segment is measured using an IMU placed along it, tangent to the cable. Angles $(\alpha, \mu)$ and $(\beta, \eta)$ can be deduced from the 3D orientation of segments **AB** and **BR** respectively.

This set-up is illustrated in Figure 3.11 for the single sliding element configurations described in Sections 3.2.2. All IMUs are assumed to give their orientation with respect to the axis of reference frame $\mathcal{F}_\mathbf{O}$. Similarly to Section 3.3.2, IMU$k$, $k \in \{1, 2\}$ is positioned at $\mathbf{P}_{i_k}$. IMU frames are denoted $\mathcal{F}_{i_k}$ and $\forall k \in \{1, 2\}$ frame $\mathcal{F}_{b_k}$ is defined fixed with respect to $\mathcal{F}_{i_k}$, with its $x$-axis tangent to the cable. Frames $\mathcal{F}_{i_k}, k \in \{1, 2\}$ are not represented in Figure 3.11 for clarity reasons.



(A) Single ballast system (B) Single buoy system

FIGURE 3.11: IMU-measurement example configurations

## 3.4 Experiments

This section presents the experimental evaluation of underwater tethered system state estimation using the models developed in Section 3.2 and model parameters measurements described in Section 3.3. Section 3.4.1 focuses on the catenary shape estimation of an underwater cable connecting a pair of robots in a robot chain framework, and Section 3.4.2 focuses on the cable-based ROV localization using a straight line cable model.

### 3.4.1 Catenary shape estimation of an underwater cable for tethered robots

The catenary shape estimation method using the vertical catenary model described in Section 3.2.1 and model parameters measurements detailed in Section 3.3.2 is implemented on an underwater robotic system. Evaluations are conducted in two times, first in an airborne set-up, and in a pool in a second time. This inertial-based catenary shape estimation method is compared with a visual one from a previous work (Laranjeira et al., 2019) where cable shape estimation is used as input for a cable-shape-based visual servoing. This visual-based method is summarized in Appendix C. Section 3.4.1.1 describes the robotic system used as well as the evaluation methodology. Sections 3.4.1.2 and 3.4.1.3 present the experiments in the air and in a pool respectively. Section 3.4.1.4 provides a conclusion and a discussion on the proposed catenary shape estimation of an underwater cable for tethered robots.

#### 3.4.1.1 Robotic system and evaluation methodology

The system is composed of a BlueROV2 tethered to a fixed point by a 1.50 m long 70 g/m red negatively buoyant cable made out of colored ballasted cord (Fig. 3.16). The robot embeds a camera, an IMU and a water pressure sensor. The cable is equipped with a pair of PhidgetSpatial Precision 3/3/3 High Resolution IMUs, ensuring they did not affect much the visibility of the cable from the embedded camera. The robot and the cable are tracked with a motion capture (mocap) system used in the air and in water. The process of ground truth recording using this mocap system is detailed in Appendix D.

The following catenary estimation methods are compared and evaluated:

(i) optimized vision-based (*vision-based optim.*) (Laranjeira et al., 2019);

(ii) only vision-based initial guess (*initial-guess*) (Laranjeira et al., 2019);

(iii) the proposed IMU-based method (*IMU-based*).

Regarding the IMU-based method, the estimation of $H$ is compared with one or both IMUs and the estimation of $\alpha$ using each one of the two cable IMUs separately. *IMU-based-1* refers to using only IMU 1, *IMU-based-2* refers to using only IMU 2, and *IMU-based* refers to using both. Since $\alpha$ estimation implies only one cable IMU, there is no *IMU-based* estimation for this parameter. The mean, median and standard deviation ($\sigma$) of the errors on $H$ and $\alpha$ are compared for each method. These errors are denoted $e_H$ and $e_\alpha$ respectively.

#### 3.4.1.2 Air configuration

A first series of experiments is set up out of water to test the estimations in a controlled environment where the robot has no pitch or roll, and the system is not disturbed by the hydrodynamics. As shown in Figure 3.12, the robot is placed on a cart and moved around in two-dimensions. The system is equipped with reflective passive markers to be tracked by a 6 camera optical motion capture system. A statistical analysis is computed on a 117$s$ sequence.

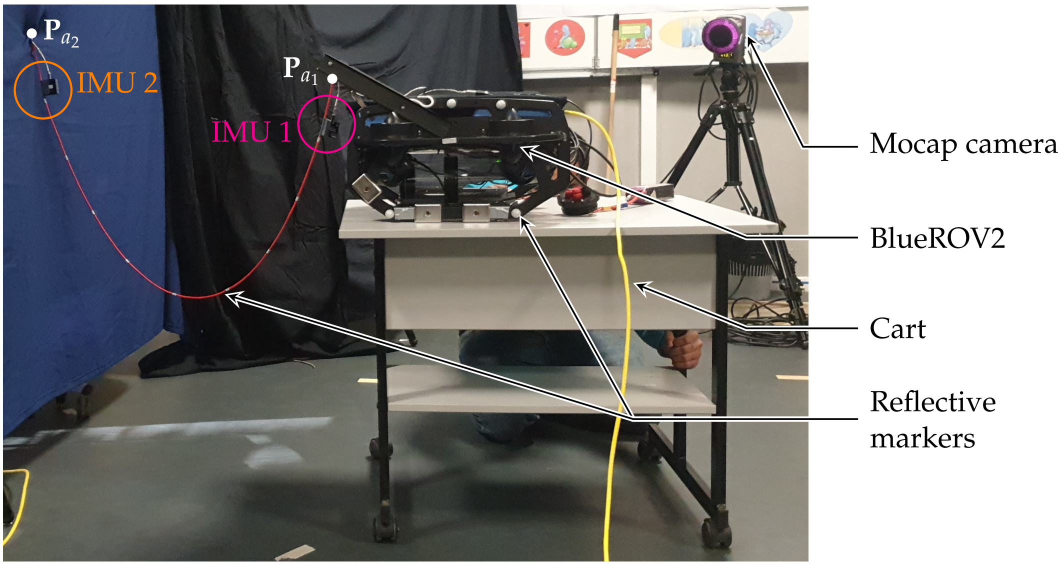


FIGURE 3.12: Airborne system.

The *vision-based optim.* method fails to converge for 11.5 % of the sequence's duration, *i.e.* the optimization does not reach a catenary shape having a sufficiently low cost within the number of allowed iterations. In the following, the results are only evaluated when converged. Figure 3.13 shows the robot's movements in a representative 15$s$ sub-sequence of the full air sequence, including the trajectories of points $\mathbf{P}_{a_1}$ and $\mathbf{P}_{a_1}$ and robot's forward axis $\mathbf{x}_r$, which is the $x$-axis of frame $\mathcal{F}_r$. The robot's motion is a composition of a lateral motion and a motion towards $\mathbf{P}_{a_2}$. Figure 3.14 shows $H$ and $\alpha$ parameters estimated by the three methods during the sub-sequence.

While the estimation of $\alpha$ overlaps the ground truth with all methods, the estimation of $H$ by the *vision-based optim.* method shows important errors from $t = 29s$ to 40$s$. This coincides with small angles $\alpha \in [-10, 10]$ degrees, *i.e.* one of the singularities where the tether plane is aligned with the optical axis and projects as a line. The error peak of the *initial-guess* method at $t = 37.4s$ is due to an outlier below the

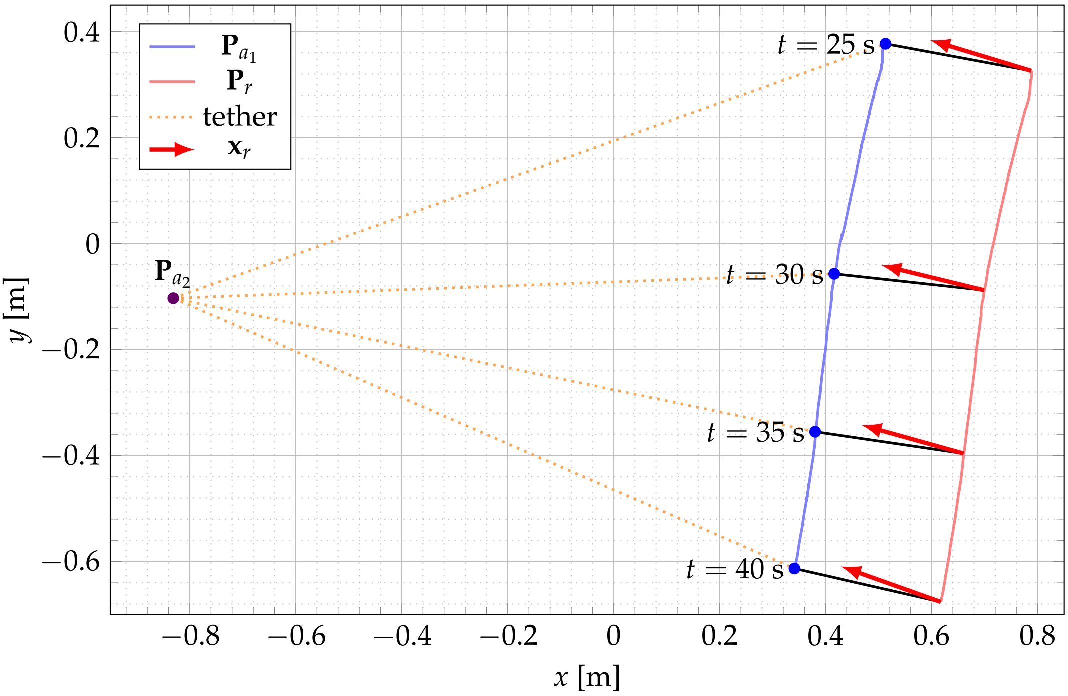


FIGURE 3.13: Trajectory of the robot projected on a horizontal plane

tether that shifts the lowest point detection. Figure 3.15 illustrates *vision-based optim.* and *initial-guess* methods. Figure 3.15a shows that both methods give overlapping results. In Fig. 3.15b, one can observe that the optimization algorithm refines the *initial-guess* result in terms of the reprojection error. Figures 3.15c, 3.15d and 3.15f show examples of bad estimations for *vision-based optim.* when $\alpha$ is close to zero. Figure 3.15e shows how an outlier below the cable disturbs the *initial-guess* method. Finally, we notice that *IMU-based* methods using only one IMU or both show similar results in the estimation of $H$ in this setup.

When considering the whole sequence, all *IMU-based* methods greatly improve the accuracy of $H$: the error is 5 to 10 times less than the *vision-based optim.* and *initial-guess* estimations (Tab. 3.4).

TABLE 3.4: Airborne results.

| | **Method** | **Visual-based optim.** | **Initial guess** | **IMU-based 1** | **IMU-based 2** | **IMU-based** |
|---|---|---|---|---|---|---|
| $e_H$ (m) | mean | 0.0425 | 0.0165 | 0.0039 | 0.0038 | **0.0033** |
| | median | 0.0264 | 0.0312 | **0.0022** | 0.0031 | 0.0026 |
| | $\sigma$ | 0.0402 | 0.0180 | 0.0131 | 0.0032 | **0.0030** |
| $e_\alpha$ (°) | mean | 2.0331 | 3.1022 | 2.7004 | **2.0313** | $\emptyset$ |
| | median | 1.9092 | 2.8962 | 2.6741 | **1.7106** | |
| | $\sigma$ | 1.1038 | 1.5256 | **1.4499** | 1.5098 | |

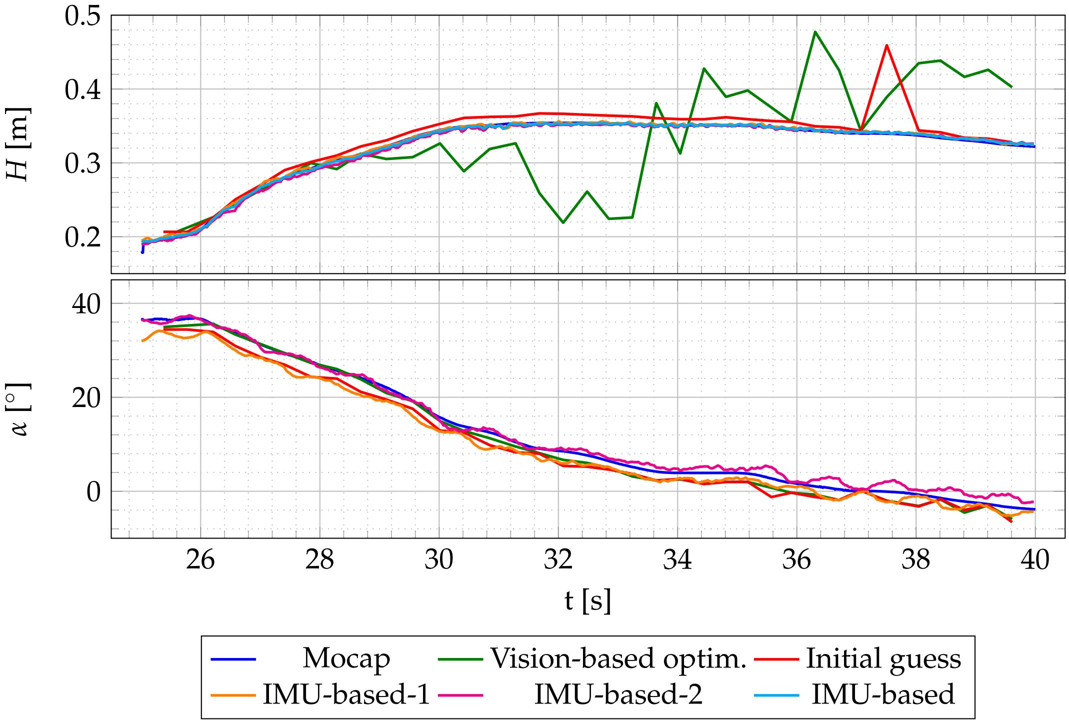


FIGURE 3.14: Estimation of $H$ and $\alpha$ in the air, with fixed $\Delta H$.

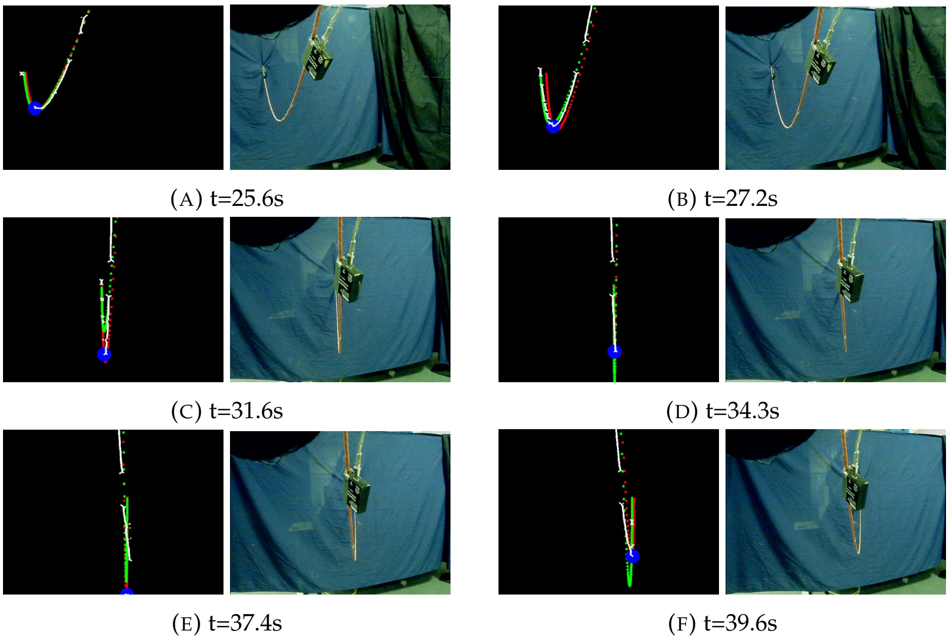

(A) t=25.6s
(B) t=27.2s
(C) t=31.6s
(D) t=34.3s
(E) t=37.4s
(F) t=39.6s

FIGURE 3.15: Detected cable points (white), with lowest point (blue) and estimated catenary projections for *vision-based optim.* (green) and *initial-guess* (red) methods.

#### 3.4.1.3 Underwater set-up

A similar set-up is deployed in an underwater environment, in the pool of the Cephismer, Marine Nationale, France. Similarly to the previous airborne set-up, the robot and the cable are equipped with reflective markers and tracked with a 5-cameras underwater motion capture system. Cable IMUs are sealed in waterproof housings (Fig. 3.16). The experiments were conducted in a steel made pool, preventing the use of magnetometer data in the IMU measurements integration. This leads to a drift of the yaw angle measurement around the vertical axis, which was characterized on a static sequence and corrected. The following results are based on a 92$s$ sequence where the robot pitches and rolls. The hydrodynamic effects and the thrusters' flow distort the catenary model. Note that, as explained in Section 3.4.1.1, there is no two-IMU based estimation of $\alpha$.

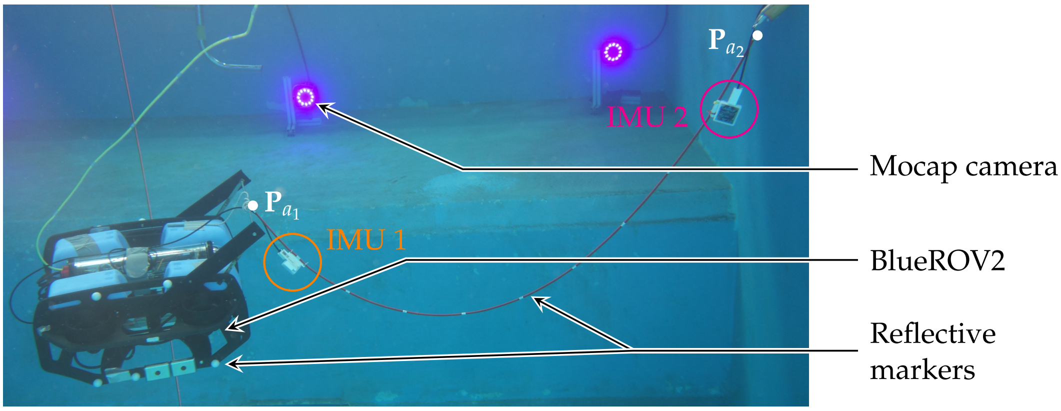


FIGURE 3.16: Underwater system.

The tether moves out of the camera's field of view during 2% of the sequence's duration. In the following, the vision-based methods (*vision-based optim.* and *initial-guess*) are analyzed only when the cable is inside the field of view. 45.3% failure is observed with the *vision-based optim.* method and 0.3% failure with the *initial-guess* method. Figures 3.17 shows the trajectories applied during a representative sub-sequence, for $t \in [24, 40]$ s. The trajectories of points $\mathbf{P}_{a_1}$ and $\mathbf{P}_{a_1}$ are represented, as well as the orientation of the robot's forward axis $\mathbf{x}_r$, which is the $x$-axis of frame $\mathcal{F}_r$. For $t \in [24, 29]$s, the robot moves towards and backwards with yaw variations and minor $y$, $z$, roll and pitch variations. For $t \in [19, 40]$s, the robot dives with minor $x$, $y$, roll and yaw variations. The robot's pitch changes when diving. Figure 3.18 gives the estimations of $H$ and $\alpha$ during the sub-sequence. Gaps in the curves correspond to estimation failures. Table 3.5 presents a statistical analysis of the full sequence.

TABLE 3.5: Waterborne results.

| | **Method** | **Visual-based optim.** | **Initial guess** | **IMU-based 1** | **IMU-based 2** | **IMU-based** |
|---|---|---|---|---|---|---|
| $e_H$ (m) | mean | 0.1514 | 0.1787 | 0.0835 | **0.0333** | 0.0438 |
| | median | 0.1261 | 0.1612 | 0.0621 | **0.0299** | 0.0406 |
| | $\sigma$ | 0.1188 | 0.1108 | 0.0978 | **0.0234** | 0.0266 |
| $e_\alpha$ (°) | mean | 16.545 | 16.042 | **11.342** | 13.270 | $\emptyset$ |
| | median | 11.030 | 11.825 | **7.7013** | 11.013 | |
| | $\sigma$ | 19.655 | 14.890 | **10.608** | 10.692 | |

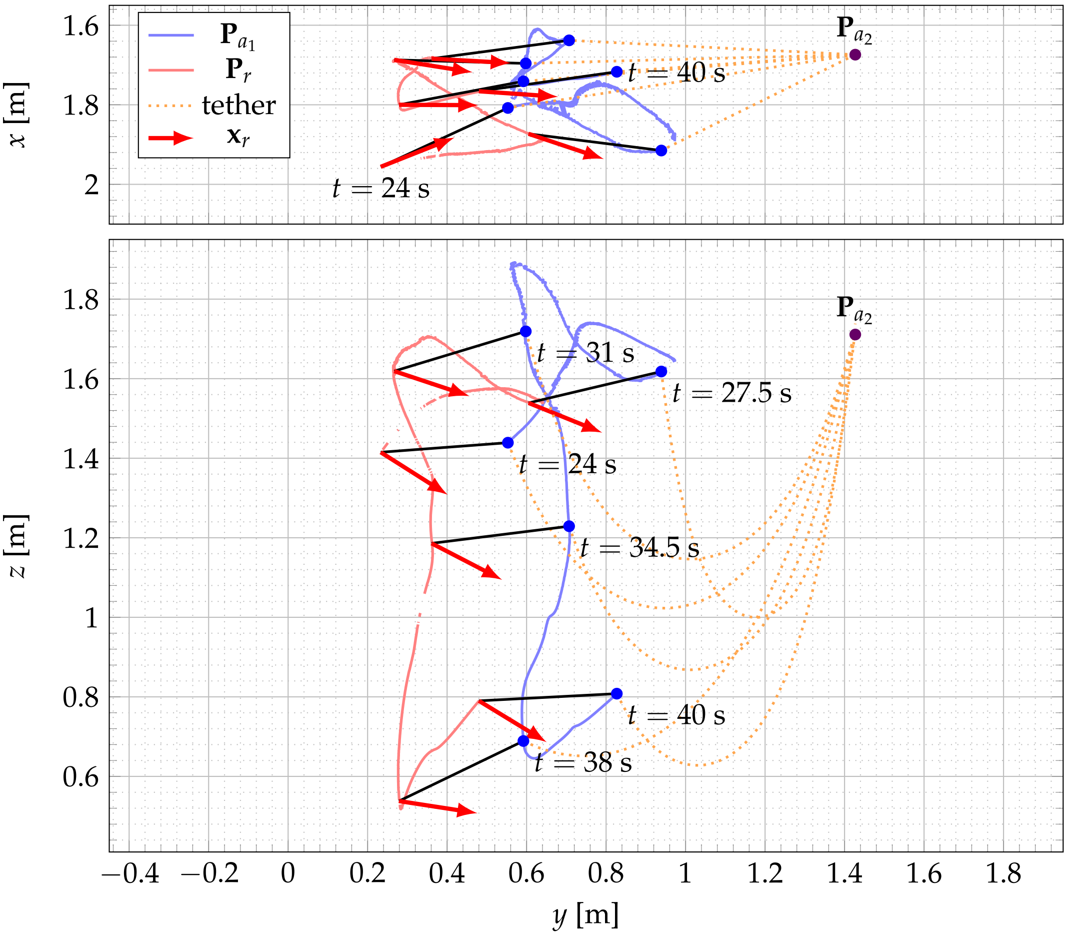


FIGURE 3.17: Trajectory of the robot projected on vertical and horizontal planes

First, the very important failure rate of the *vision-based optim.* method can be noticed. In the underwater sequence, the color based detection fails at detecting the farthest cable points due to water absorbance, as shown in Fig. 3.19. Furthermore, the cable projection in the image is deformed with regard to vision based expectations due to roll, pitch, and hydrodynamics, including the currents created by the thrusters. The error in the model makes the vision based optimization fail to fit the cable shape projection properly. The *initial-guess* method gives more robust results with regard to the deformations since it only considers the lowest point. Figure 3.19 illustrates the behaviour of the *vision-based optim.* and *initial-guess* methods. In Fig. 3.19a both methods fit the points quite well, whereas in Fig. 3.19f, the *vision-based optim.* method clearly refines the *initial-guess* in terms of reprojection error. In Figs. 3.19b, 3.19d and 3.19f, the *vision-based optim.* method suffers from the farthest part of the cable not being detected correctly because of color absorption. Figure 3.19c clearly shows that the deformations of the cable projection in the image make the projection model unsuitable for the *vision-based optim.* and *initial-guess* estimations.

Considering the estimation of $H$, IMU-based methods perform 3 times better than vision-based methods, and show a smaller dispersion. The estimation from the IMU 1 is less accurate due to the thrusters' flow that deforms the rope. For

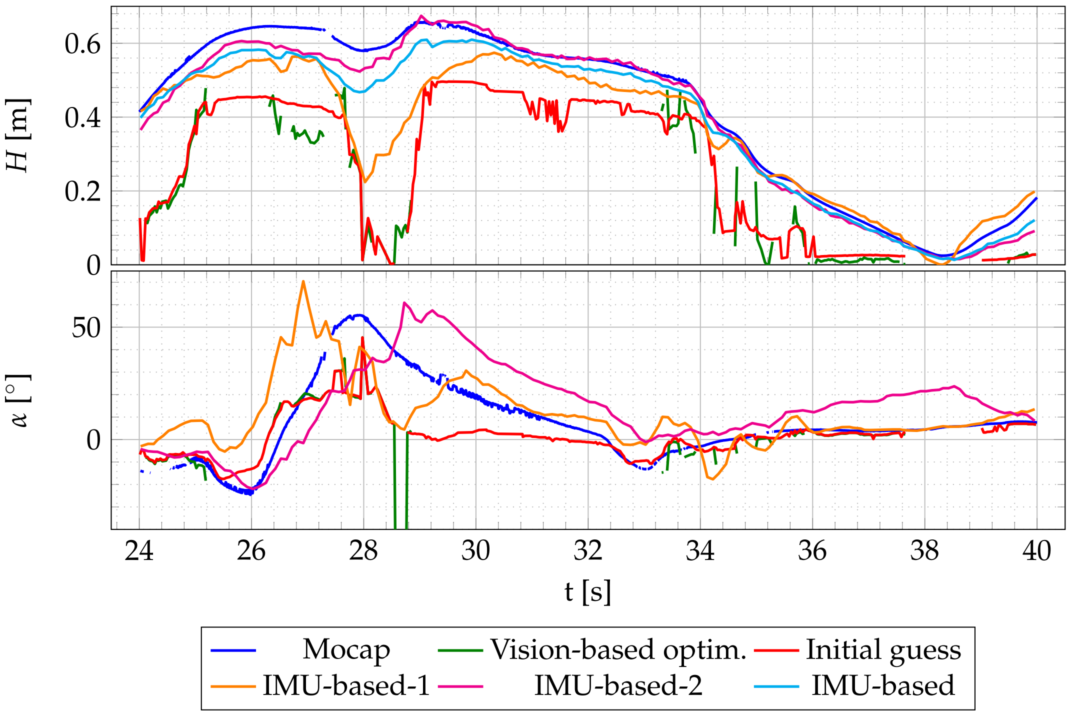


FIGURE 3.18: Estimation of $H$ and $\alpha$ in the pool.

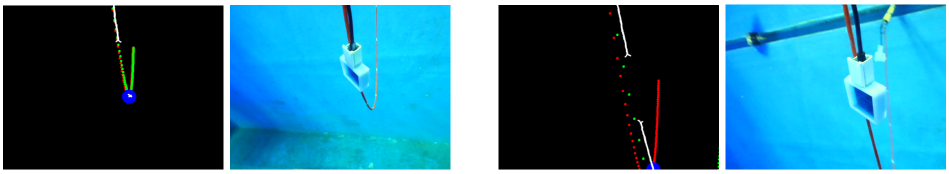
instance, in Fig. 3.18, one can see an important error for the *IMU-based-1* method at $t \in [27, 29]$s, which coincides with a backwards movement of the robot that propels water onto IMU 1 (Figure 3.17).

Looking at $\alpha$, all methods show similar estimation performances, but with different sources of errors. The *initial-guess* method is impacted by the quality of the tether detection, since it needs the cable's lowest projection point in the image to be properly detected, but this error does not propagate over time. *Vision-based optim.* method suffers from the detection and the deformation of the cable's projection, and eventually shows poor performances compared to the others due to its important failure rate. *IMU-based* methods' errors are partly due to the hydrodynamics of the system, with IMU 1 being disturbed by the robot's thrust, and IMU 2's movement being impacted by the propagation of the deformations along the cable. There may also be yaw integration issues of IMUs in the absence of a magnetometer, in an indoor environment. During the sequence, the yaw angle was corrected with shift and drift calibration assuming constant drift, which needs to be updated over time. A too sharp motion could also lead to bad yaw integration. In outdoor environments, the method will be more robust through the use of magnetometers.

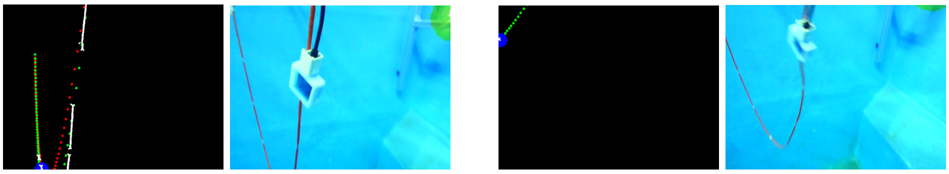
As a conclusion, it results from this analysis that combining visual and inertial measurements in estimating $\alpha$ could lead to a much more robust estimation of this parameter.

#### 3.4.1.4 Conclusions

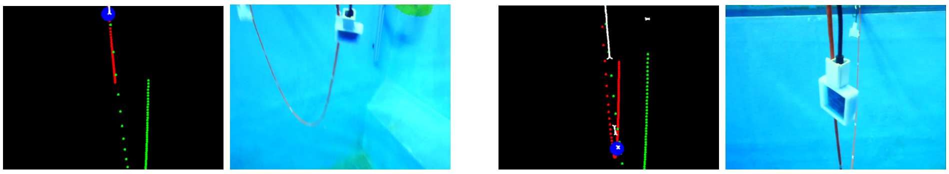
The current section evaluated a new catenary shape estimation for a negatively buoyant underwater cable based on inertial measurements of one or two tangents near the attachment points, where the model was presented in Section 3.2.1 and model parameters estimation was described in Section 3.3.2. This estimation was



(A) t=23.96s (B) t=25.53s



(C) t=27.56s (D) t=28.02s



(E) t=28.65s (F) t=33.40s

FIGURE 3.19: Detected tether points (white), with lowest point (blue) and estimated catenary projections for *vision-based optim.* (green) and *initial-guess* (red) methods in the image plane.

compared to vision-based estimation from previous work in experiments carried out in the air and in a pool, using a motion tracking system for ground truth. It was shown that the new method is significantly more accurate and robust in estimating the catenary sag $H$, with an error less than 10 cm and often less than 5 cm. In addition, approximating $H$ with a single IMU measurement gives the same order of accuracy than computing the exact analytical solution given for two IMUs. On the other hand, all the estimation methods evaluated have the same order of accuracy in estimating the cable angle $\alpha$, including the visual initial guess.

Although it has not been investigated in the current thesis, fusing visual and inertial methods might be an interesting future work for improving estimation robustness. Indeed, IMU integration may drift on long periods whereas vision will not, and unlike vision IMUs are always available. The IMU-based computation is also more robust to cable deformation. The visual and inertial approaches may also be complementary in estimating the relative cable-robot orientation when the thrusters' flow impacts the IMU orientation. Beyond the scope of the current thesis, the current IMU-based cable shape estimation may also be used as a control input to regulate the robot chain's shape.

With regards to the problem of underwater robot chain localization, our IMU-based cable shape estimation provides cable 3D localization information, which can be used to localize the robots relative to each other. Distance between a pair of consecutive robots can be estimated robustly with our method from $H$. However, our precision of about ten centimeters may be important compared to other localization methods, and more specifically compared to VSLAM-based approaches. In addition, estimation of $\alpha$ is still quite imprecise, and robot cable-based pose estimation accuracy may decrease very quickly when propagating from pair to pair along the robot chain. Future works may thus investigate how the currently roughly accurate but robust inter-robot localization and a more accurate but maybe less robust VSLAM based localization strategies may be complementary in locating an underwater robot chain.

### 3.4.2 Cable-based ROV localization using a straight line cable model

The cable-based ROV localization strategy using the straight-line cable model introduced in Section 3.2.2 and model parameters measurements described in Section 3.3.3 is implemented on an underwater robotic system. This experimental study focuses on a single sliding ballast configuration, which is studied for a ROV tethered to a fixed point or to a secondary ROV in the context of an underwater robot chain. Section 3.4.2.1 presents the evaluation of the proposed localization method for a ROV tethered to a fixed point. Section 3.4.2.2 studies the influence of IMU positioning along the cable on localization accuracy. Section 3.4.2.3 focuses on a pair of ROVs connected together in a robot chain context. A conclusion is provided in Section 3.4.2.4

#### 3.4.2.1 *Single-ballast* system towards a fixed point

The proposed cable-based localization system is first evaluated for a ROV tethered to a fixed point, in the pool of the Cephismer, Marine Nationale, France. The robotic system used is composed of a BlueROV2 from BlueRobotics tethered to a fixed point by its Fathom Slim[1] communication tether with a 240 g sliding ballast, such that $l_0 = 0$ and $L_t = 3$ m. The sliding ballast is made of a neutrally buoyant pulley to which a mass is fixed, as illustrated in Figure 3.20.

FIGURE 3.20: Sliding ballast composed of a pulley and two masses of 120 g each.

The BlueROV2 embeds a pressure sensor and an IMU. As in the experiments depicted in Section 3.4.1.3, the cable is equipped with a pair of PhidgetSpatial Precision 3/3/3 High Resolution IMUs sealed in waterproof housing. As in Section 3.4.1.3, a mocap system is installed in the pool in order to track reflective markers placed on the ROV, the cable and the pulley and provide a ground truth of the state of the system. This set-up is illustrated in Figure 3.21.

---

[1] https://bluerov-solutions.com/produkt/fathom-slim-rov-tether-rov-ready/

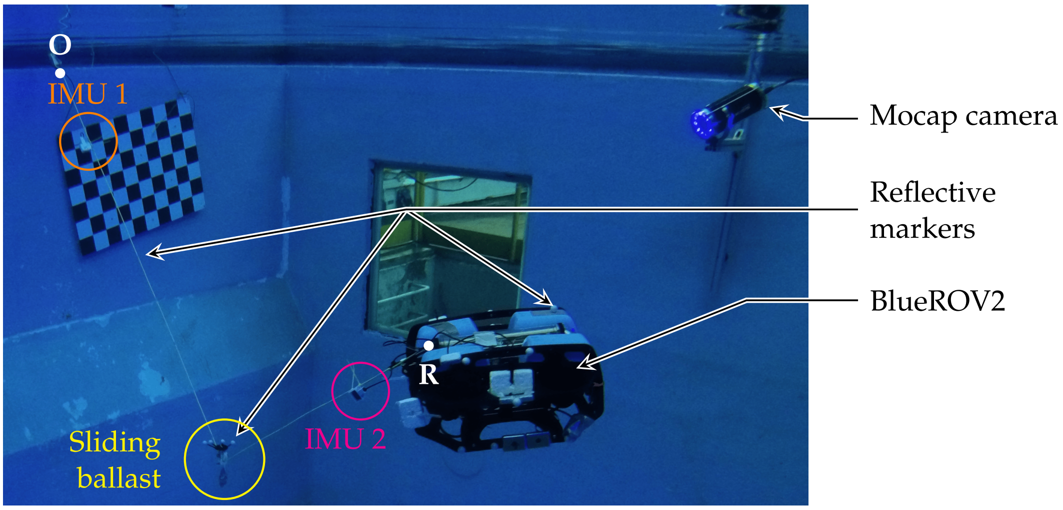


FIGURE 3.21: Experimental system

The distances between **O** and $\mathbf{P}_{i_1}$ and **R** and $\mathbf{P}_{i_2}$ (see Figure 3.11) are denoted respectively $l_{i_1}$ and $l_{i_2}$. IMUs are placed such that $l_{i_1} = 0.4$ m and $l_{i_2} = 0.2$ m. These positions are selected arbitrarily to be far enough from the cable ends to make sure angle measurement is not disturbed by the cable's stiffness, but close enough to the ends not to block the displacement of the sliding ballast. Three ROV localization methods are compared:

- the proposed *IMU-based* ROV localization using model parameter measurements from the embedded sensors
- the *ground truth* ROV localization provided by the mocap tracking of the system
- the localization computed using mocap measurements of model parameters instead of embedded sensor, which correspond to ideal parameters measurements. This localization is referred to as *mocap-based* ROV localization.

Embedded sensor measurements and mocap measurements are put in the same theoretical referential $\mathcal{F}_O$ by a change of coordinate frame for comparison purpose. The data is processed offline, including a synchronization of clock between the IMU and mocap measurements, as well as the transformation between mocap and sensor frames.

Two motion sequences are recorded, indexed 1 and 2. Sequence 1 is a short sequence featuring very slow displacements with small range, while Sequences 2 is a four-minute sequence showing ROV displacements closer to a real use case, with higher speed, motion variation and displacement range. ROV localization results and error with respect to mocap *ground-truth* are represented in Figures 3.22 and 3.23 for Sequences 1 and 2 respectively. The error $e$ is the distance between the ground truth and the estimated ROV position. In addition, we define the average distance error $E_d$ as the distance between the estimation of the position of **R** in $\mathcal{F}_O$ computed using our model and the position measured by the mocap system averaged over a test sequence.

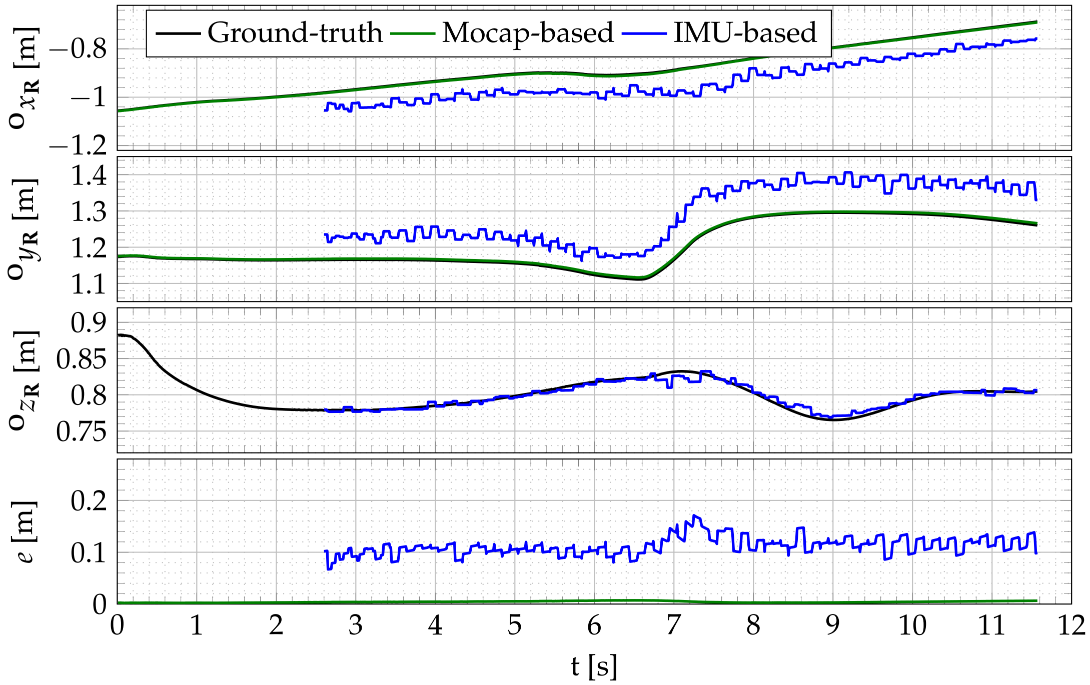


FIGURE 3.22: *IMU-based*, *mocap-based* and *ground truth* ROV position and positioning error for Sequence 1 ($[l_{i_1}, l_{i_2}] = [0.20{,}0.40]$ m).

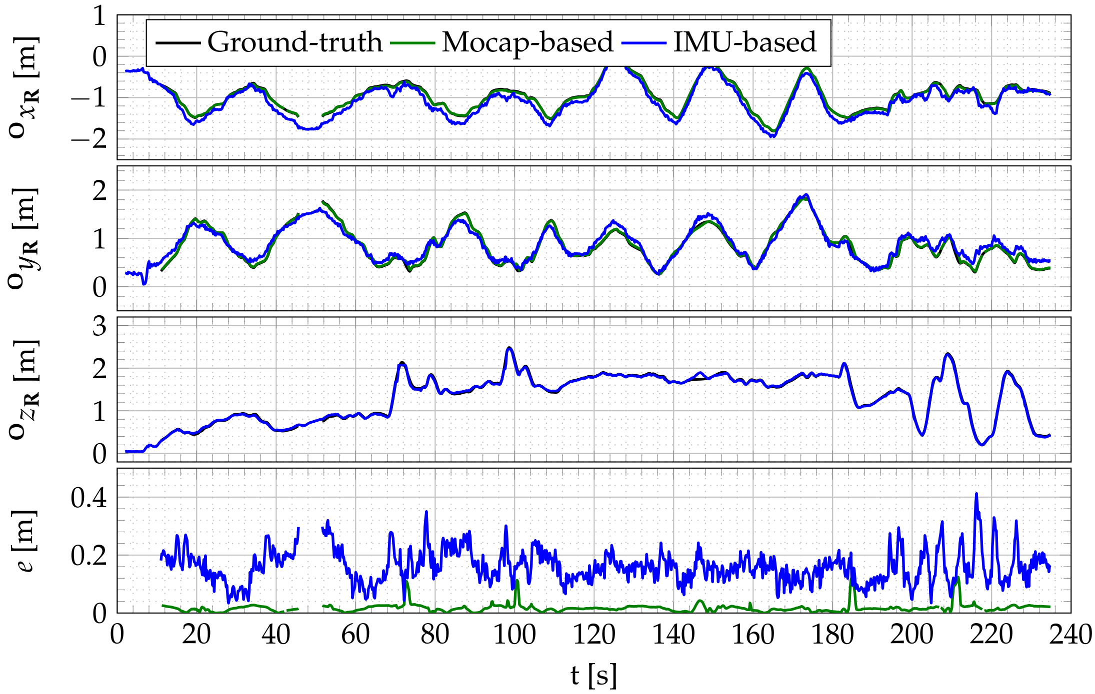


FIGURE 3.23: *IMU-based*, *mocap-based* and *ground truth* ROV position and positioning error for Sequence 2 ($[l_{i_1}, l_{i_2}] = [0.20{,}0.40]$ m).

One can see that the results of *mocap-based* ROV localization are particularly close to the *ground truth* in Sequence 1, with an error $E_d$ of about 0.15 m only. *IMU-based*

ROV localization shows a slightly lower accuracy but the corresponding results are still very close to those of the *mocap-based* ROV localization, meaning that the cable angles are estimated very accurately by the IMUs. This result shows that the proposed localization method can lead to a very accurate ROV position estimation if the ROV is close to stationary. ROV localization error is higher for Sequences 2 to 4. The error is about 0.20 m for the *IMU-based* approach, which is thus acceptable for the tested configuration. The *mocap-based* approach results in a 0.02 m error, showing that the accuracy of the method is directly proportional to the accuracy of the angle measurement. These results demonstrate the validity of the theoretical model and the possibility of obtaining a fairly accurate position even when the ROV is in motion, since the umbilical remains stretched by the ballast during the displacement.

#### 3.4.2.2 Influence of IMU positionning

Due to the stiffness of the cable, local cable orientation may vary slightly along the cable and be more or less representative of the global cable segment orientation. The influence of the positions of the IMUs along the cable is therefore examined by reproducing the experiment described in Section 3.4.2.1 with different values of $l_{i_1}$ and $l_{i_2}$. Two positions are compared for each IMU, with $l_{i_1} = 0.40$ m or 0.78 m and $l_{i_2} = 0.20$ m or 0.40 m, resulting in four distinct $\{l_{i_1}, l_{i_2}\}$ configurations. Sequences 1 and 2 studied in Section 3.4.2.1 feature the $\{l_{i_1} = 0.40\text{ m}, l_{i_2} = 0.20\text{ m}\}$ configuration. Three additional sequences indexed from 3 to 5 are recorded in order to test the three remaining $\{l_{i_1}, l_{i_2}\}$ configurations. Similarly to Sequence 2, Sequences 3, 4 and 5 involve motion close to a real ROV use case, with speed and motion variations and large displacement range.

The position of the ROV is evaluated according to the *IMU-based* ROV localization to *mocap-based* localization. The ROV localization results and error with respect to mocap *ground-truth* are represented in Figures 3.24, 3.25 and 3.26. The mean, median and standard deviation ($\sigma$) of ROV position error for each $\{l_{i_1}, l_{i_2}\}$ configuration are reported in Table 3.6, based on Sequences 2, 3, 4 and 5. Because Sequence 1 is the only quasi-static sequence, it is omitted from this analysis.

TABLE 3.6: Statistics on ROV position error for each $\{l_{i_1}, l_{i_2}\}$ configuration for *IMU-based* estimation

| | **Seq. 2** | **Seq. 3** | **Seq. 4** | **Seq. 5** |
|---|---|---|---|---|
| $\{l_{i_1}, l_{i_2}\}$ (m) | $\{0.40, 0.20\}$ | $\{0.78, 0.20\}$ | $\{0.78, 0.40\}$ | $\{0.40, 0.40\}$ |
| mean (m) | **0.161** | 0.173 | 0.171 | 0.201 |
| median (m) | 0.156 | 0.168 | **0.150** | 0.206 |
| $\sigma$ (m) | **0.057** | 0.090 | 0.089 | 0.069 |

One can observe that the position of the IMUs appears to have little impact on the results. Indeed, all configurations lead to the same error order of magnitude, between 0.15 m and 0.20 m. These results show that the *IMU-based* method proposed in this work is therefore flexible with respect to the installation of IMUs on the umbilical, whose position can be chosen so as to obstruct the *sliding ballast* as little as possible.

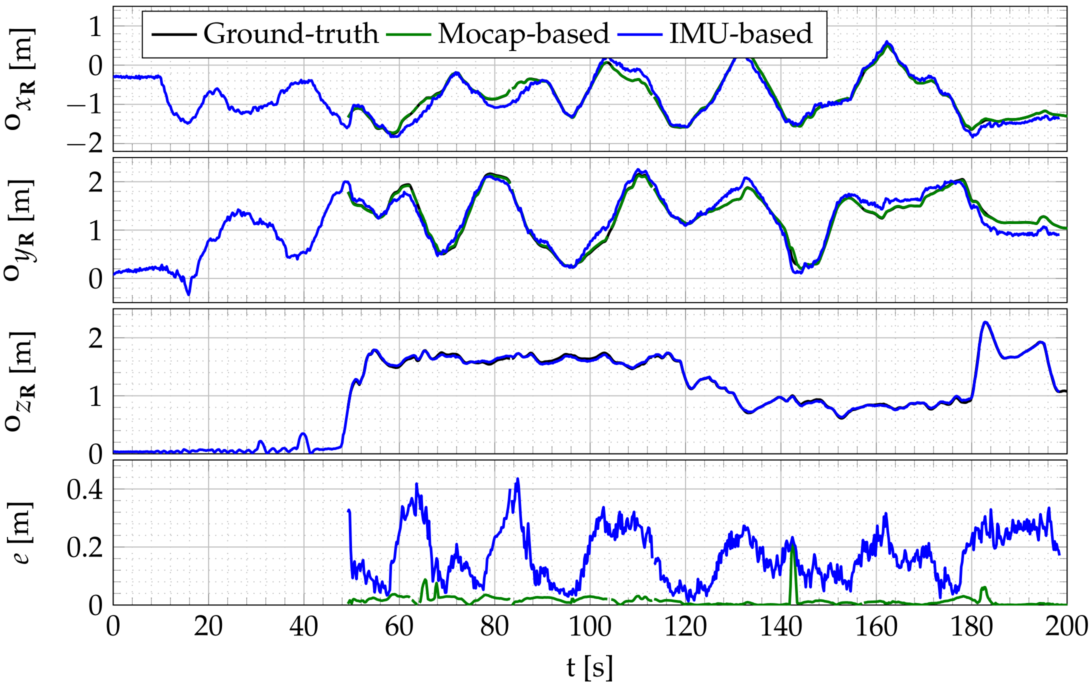


FIGURE 3.24: *IMU-based*, *mocap-based* and *ground truth* ROV position and positioning error for Sequence 3 ($[l_{i_1}, l_{i_2}] = [0.78, 0.20]$ m).

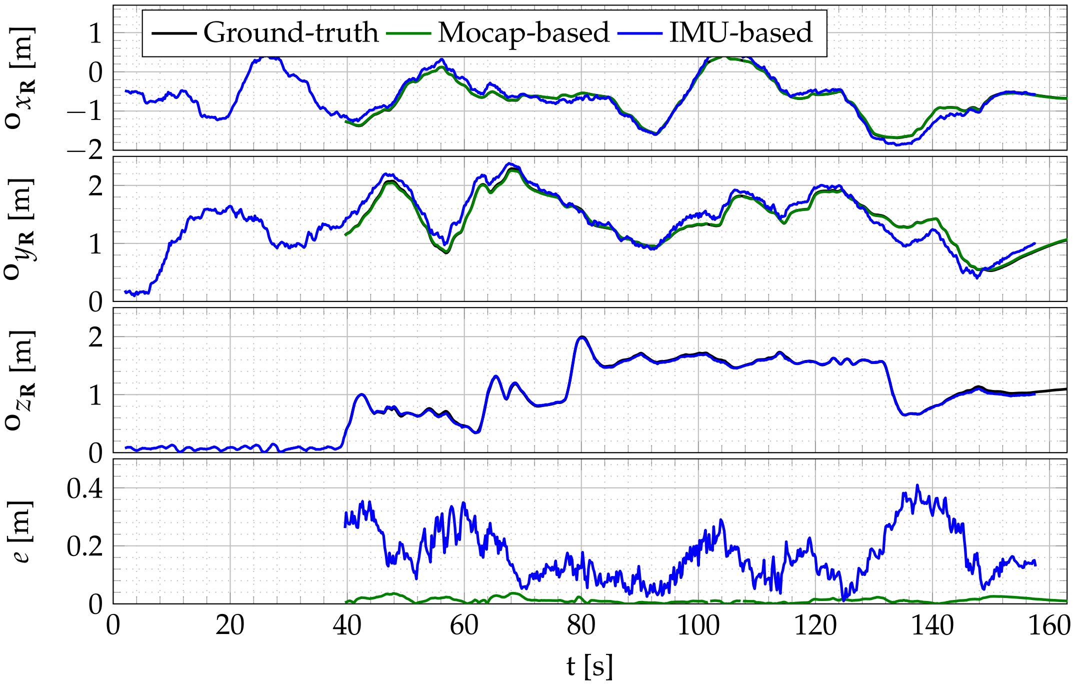


FIGURE 3.25: *IMU-based*, *mocap-based* and *ground truth* ROV position and positioning error for Sequence 4 ($[l_{i_1}, l_{i_2}] = [0.78, 0.40]$ m).

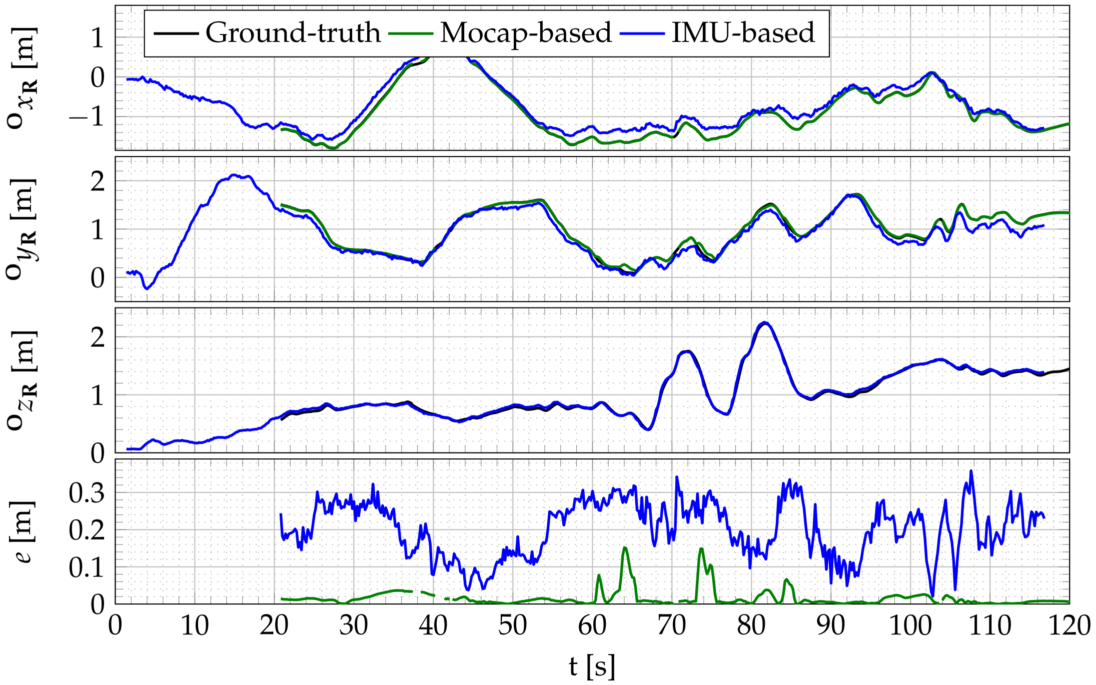


FIGURE 3.26: *IMU-based*, *mocap-based* and *ground truth* ROV position and positioning error for Sequence 5 ($[l_{i_1}, l_{i_2}] = [0.40, 0.40]$ m).

#### 3.4.2.3 IMU-based localization in a chain of two ROVs

Lastly, a system composed of two BlueROVs tethered together by a cable equipped with a sliding ballast is considered. In this experiment, the cable which connects the robot is simulated by a 3 m long cord. As in the previous experiments, a pair of IMUs are placed along the cable. The system is deployed in the pool of the Ifremer, La Seyne-sur-mer, France, which is larger and deeper, making it easier to deploy a two-robot system. However, this pool does not allow the installation of the underwater mocap system. Instead, we take advantage of the embedded cameras of the ROV to generate a Structure from Motion (SfM) localization baseline. This baseline is assumed to be far less precise than the mocap ground truth used in the previous sequences because of the accuracy of the SfM itself, but also and mostly because transferring the SfM estimated pose to the attachment points involve calibrations which introduce important sources of imprecision. However, a precision of a few centimeters is to be expected, making this SfM baseline still suitable for evaluating our approach.

The system is represented in Figure 3.27, and pictured in Figure 3.28. $\forall k \in \{1, 2\}$, Robot $k$ embeds an IMU in $\mathbf{P}_{i_{r,k}}$, a camera in $\mathbf{P}_{c_k}$ and a pressure sensor in $\mathbf{P}_{d_k}$. Frames $\mathcal{F}_{i_{r,k}}$, $\mathcal{F}_{c_k}$ and $\mathcal{F}_{d_k}$ are the frames associated to these sensors, respectively. Frame $\mathcal{F}_{\mathbf{O}}$ is defined with its $x$-axis vertical, downwards, and with same yaw angle as $\mathcal{F}_{\mathbf{i_{r,1}}}$. In line with the conclusions of Section 3.4.2.2, the position of the IMUs along the cable is chosen close to the attachment points, in order to obstruct the *sliding ballast* as little as possible.

The pose of the camera frames $\mathcal{F}_{c_1}$ and $\mathcal{F}_{c_2}$ is estimated using the SfM software Colmap (Schönberger et al., 2016) with respect to a reference frame $\mathcal{F}_{w_c}$, which is defined arbitrarily. The position of attachment points $\mathbf{O}$ and $\mathbf{R}$ in frames $\mathcal{F}_{c_1}$ and

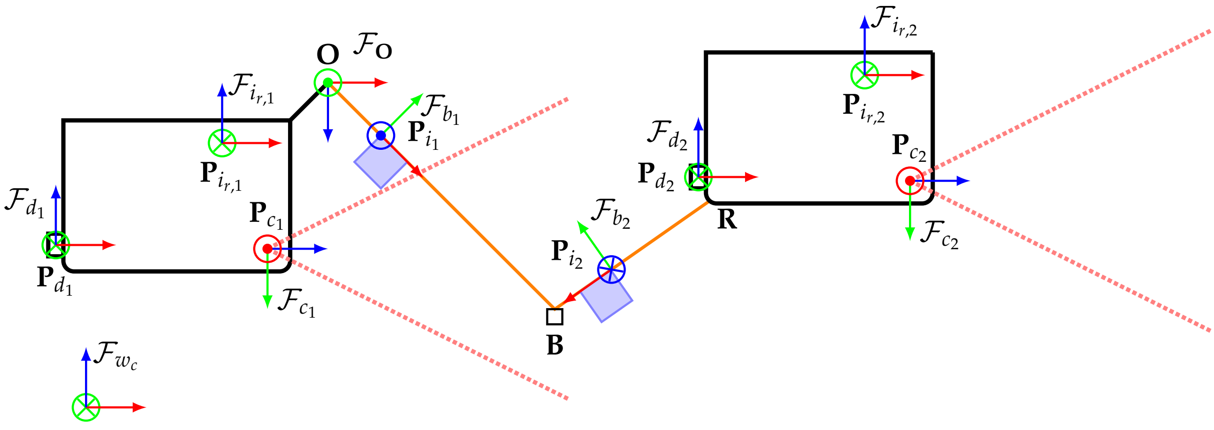


FIGURE 3.27: Two-agent robot chain with a sliding ballast

$\mathcal{F}_{c_2}$ respectively is measured at the beginning of the experiment, in order to determine their relative position in $\mathcal{F}_{w_c}$ according to the SfM reconstruction, along the sequence. Since the transformation between frames $\mathcal{F}_{w_c}$ and $\mathcal{F}_{\mathbf{O}}$ is unknown at any time of the sequence, we use the distance between points **O** and **R** to characterize the estimation error.

FIGURE 3.28: Robotic system in the pool, featuring two BlueROV2 linked by a cable on which are placed a sliding ballast and two IMUs. robots move around a fake reef which is used to compute the SfM baseline.

A 60 s sequence is recorded. The distance $d$ between attachment points**O** and **R** computed from the SfM reconstruction and from our localization method and the corresponding error are reported in Figure 3.29, as well as the distance error $e_d$ defined as the absolute difference between the reference and estimated distance. One can see that the accuracy is about 0.20 m, which is the same order of magnitude as for the system studied in Sections 3.4.2.1 and 3.4.2.2, featuring a sliding ballast system tethered to a fixed point. Similarly, we can conclude that the current cable-based ROV localization system can also be used in estimating the relative localization of a pair of ROVs tethered by a cable equipped with a sliding ballast, with an accuracy about 20 cm here, which represent about 7% of the cable length in the studied configuration.

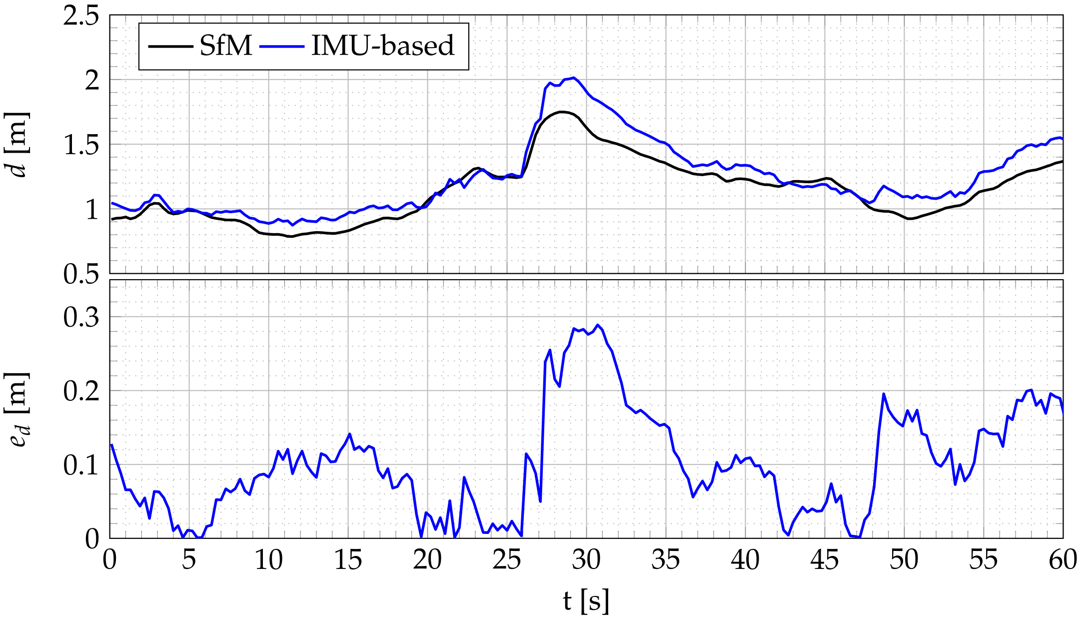


FIGURE 3.29: Relative distance estimation result.

### 3.4.2.4 Conclusions

The current section presented the experimental validation of a method to estimate the position of a ROV by observing the shape of its umbilical, using the straight-line model developed in Section 3.2.2 and model parameter measurement from embedded sensors including two IMUs placed on the cable and an IMU and a water pressure sensor placed in the ROV, as described in Section 3.3.3. By equipping the umbilical with moving ballast, the cable takes a predictable shape with straight lines, simple to model. By measuring the angles at the ends of the cable and the depth of the ROV, the shape of the umbilical can be reconstructed from these straight line models and therefore the position of the ROV can be determined. This section investigated cable angle measurement using IMUs placed along the umbilical, and depth measurement using the embedded pressure sensor of a ROV and its IMU. The sliding element also allows avoiding entanglement of the cable with itself or with the surrounding obstacles, without motorization or TMS, easy to set up. In addition, the cable model is still valid in the presence of currents if the umbilical is taut.

The proposed IMU-based cable-based localization method was evaluated in a pool in two scenarios featuring a 3 m long cable: a ROV tethered to a fixed point, and a ROV tethered to another ROV, in the context of a robot chain. In the fixed point scenario, a motion capture system was used to provide the ground truth state of the system. Motion capture measurements of model parameters were also used to compute a cable model-based ROV localization with ideal model parameter measurements. With ideal parameter measurements, the model allowed locating the ROV with an accuracy of about 10 cm, embedded sensor-based measurements only lead to an accuracy of about 20 cm. These results demonstrate the interest of the proposed cable-based localization method for operational applications, but also that its accuracy is directly impacted by the accuracy of parameters measurements, including cable angles and cable attachment point depth. An analysis of the influence of measurement errors is available in (Viel et al., 2023).

## 3.5 Conclusions and perspectives

This chapter focused on the state estimation of a ROV connected to another vehicle, based on an estimation of the shape of its umbilical. The problem of cable shape estimation was addressed by modeling the cable as a function of the relative depth of its attachment points, which can be measured using a pressure sensor and an embedded IMU, and the local 3D orientation of the cable in one or two points, which can be measured by IMUs placed on the cable. In addition, if the shape of the tether is known, the tether can be turned into an advantage by being converted into a localization system which can be used when other systems are unusable or as a redundant system to another location system.

Two models were considered. First, the catenary model was investigated for modeling an underwater non-extensible weighted cable, even with moving end points. This model is a physical, quasi-static model, which does not require constraining the cable into a specific shape. The validity of this model for heavy enough underwater non-extensible weighing cable with moving ends was demonstrated experimentally, allowing to develop an inertial-measurement-based catenary shape estimation of underwater cables for tethered robots. The second model investigated was a straight line model derived from the work of (Viel, 2022a), which involves constraining the cable into a piecewise linear shape by the addition of sliding buoys or ballasts on it. A cable-based ROV localization method was developed based on this second model.

These two cable-based state estimation strategies were evaluated in a pool and compared to a motion capture gold standard, for a robot tethered to a fixed point and to provide an inter-robot relative localization in the context of a robot chain. These experiments demonstrated that both of these models can locate the cable and its end point with an order of accuracy of 20 cm for a length of 1.5 to 3 m.

The main asset of the cable-based tethered system state estimation methods studied in the current chapter, and in particular of the proposed cable-based ROV localization strategy, is that they are available even when navigating in open water, where the seabed cannot be observed by the embedded sensors. They may therefore be particularly useful for a ROV or a chain of ROVs traveling between the surface and the working depth. A limitation of these approaches for system localization is that they do not provide any information about the surrounding of the system, while such information is still essential in order to control such a system in the presence of obstacles. A cable-based perception of the environment may be studied by using contact detection between the cable and surrounding obstacles, like in (McGarey et al., 2017). Such approach in the underwater domain may although present safety issues, since the cable may get stuck in the obstacles. Contact-free localization and mapping strategies using additional exteroceptive acoustic or optical may be an interesting option in order to limit the risk of entanglements while locating the system with respect to its environment. In addition, exteroceptive localization may be fused to cable-based state-estimation in order to improve the accuracy but also the robustness in the knowledge of the system's state. An exteroceptive feedback should also prevent inter-robot localization error propagation in robot chains composed of more than two robots.

The work presented in Chapter 3 led to the following publications:

**Journal papers:**

- ROV localization based on umbilical angle measurement.
  C. Viel, J. Drupt, C. Dune, V. Hugel,
  Ocean Engineering, Volume 269, 2023, 113570, ISSN 0029-8018.

**Conference papers:**

- Inertial-measurement-based catenary shape estimation of underwater cables for tethered robots.
  J. Drupt, C. Dune, A. I. Comport, S. Seillier and V. Hugel,
  2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Kyoto, Japan, 2022, pp. 6867-6872.
  Video 1: `https://www.youtube.com/watch?v=TKLLVTSUN8s`
  Video 2: `https://www.youtube.com/watch?v=x-SXut75vHk`

- Estimation de forme de câble pesant pour la localisation de robots sous-marins encordés : comparaison d'une approche visuelle à une nouvelle approche inertielle.
  J. Drupt, C. Dune, A. I. Comport and V. Hugel,
  ORASIS 2023, Carqueiranne, France, 2023.

- An augmented catenary model for underwater tethered robots.
  M. Filliung, J. Drupt, C. Peraud, C. Dune, N. Boizot, A. I. Comport and V. Hugel,
  Submitted to 2024 IEEE International Conference on Robotics and Automation (ICRA).

**Workshops:**

- "Validity of the catenary model for moving submarine cables with negative buoyancy.
  J. Drupt, C. Dune, A. I. Comport and V. Hugel,
  3rd workshop on RObotic MAnipulation of Deformable Objects: challenges in perception, planning and control for Soft Interaction (ROMADO-SI), Best Paper Award, Kyoto, Japan, 2022.
  Video: `https://www.youtube.com/watch?v=kmcs9xKf3KQ`

In addition, a journal paper on ROV localization through IMU-based angle measurement of an umbilical equipped with sliding ballast is currently being written, in order to be submitted to the IEEE Journal of Oceanic Engineering.

# Chapter 4

# Visual-based localization

## 4.1 Introduction

As discussed in Section 3.5, controlling a robot chain in the presence of obstacles not only involves a knowledge of the system's state but also of the location of the system with respect to the surrounding obstacles. Consequently, this chapter focuses on simultaneous localization and mapping for a chain of underwater vehicles.

As explained in Section 2.4, the most common exteroceptive sensors for underwater are acoustic sensors and cameras, where cameras are low-cost, lightweight and low power sensor compared to acoustic sensors, such that even the smallest and most affordable underwater robots embed at least a monocular camera. Cameras can provide a rich feedback on their surroundings but they do not interfere with each other nor with their environment because they are passive sensors, unlike acoustic ones. In particular, acoustic sensors may not be suitable in cluttered environments because of multiple reflections of the acoustic waves on non-target objects. In addition, visual simultaneous localization and mapping (VSLAM) algorithms are very efficient, accurate and widely used in airborne applications. VSLAM may therefore be a solution for underwater localization. More specifically, we focus on monocular VSLAM as it is the most usual instrumentation, which does not require extrinsic calibration of multiple sensors.

Underwater conditions are, though, particularly challenging, because of selective color absorption, backscattering and suspended particles. In addition, the embedded lights required for deep sea missions invalidate the lambertian assumption of airborne VSLAM. Lastly, most airborne VSLAM systems are designed for highly structured urban or industrial environments, whereas underwater vessels often operate in natural, less structured environments.

In this chapter, monocular VSLAM is first investigated for a single underwater robot. Six monocular VSLAM algorithms are first evaluated and compared on single agent datasets in underwater conditions, highlighting the interest of the multimap approach of ORB-SLAM3 (Campos et al., 2021) for underwater applications. In a second time, a multi-ROV scenario is considered, where all robots can communicate in real time with a central server through their cable. A new multi-agent and multi-map fully centralized VSLAM algorithm is derived from ORB-SLAM3 and compared to the state-of-the-art multi-agent VSLAM algorithm on underwater scenarios.

The contributions of this chapter include:

- A qualitative benchmark of six monocular, single agent VSLAM algorithms on underwater scenarios, namely ORB-SLAM (Mur-Artal et al., 2015), ORB-SLAM (Huang et al., 2020), ORB-SLAM3 (Campos et al., 2021), DSO (Engel et al., 2018), LDSO (Gao et al., 2018) and DSM (Zubizarreta et al., 2020)

- MAM$^3$SLAM a new multi-agent and multi-map fully centralized monocular VSLAM algorithm derived from ORB-SLAM3 (Campos et al., 2021)
- A benchmark of multi-agent monocular VSLAM algorithms on underwater two-agent scenarios, including CCM-SLAM (Schmuck et al., 2019), ORB-SLAMM (Daoud et al., 2018) and MAM$^3$SLAM
- The collection of underwater two-agent datasets for multi-agent monocular VSLAM benchmarking.

Section 4.2 presents the underwater evaluation of state-of-the-art monocular VSLAM algorithms on underwater scenarios, and Section 4.3 presents the work on multi-agent VSLAM, including MAM$^3$SLAM and the multi-agent monocular VSLAM benchmark. Finally, Section 4.4 provides a conclusion and discusses the perspectives from this chapter.

## 4.2 Underwater evaluation of monocular VSLAM

This Section focuses on monocular VSLAM algorithms for underwater robotic applications. A reliable VSLAM algorithm for such applications should be robust enough to underwater visual deformations to perform correct data associations, but also — ideally — implement an efficient SLAM recovery algorithm in order to handle tracking failures. In addition, such VSLAM algorithm should also be robust to occasionally fast motion which may cause loss of sight of the surrounding visual features. Indeed, while underwater vehicle motion is usually constrained for photogrammetric surveys in order to be slow and with the camera always facing the seabed or the surrounding objects, other kinds of missions may benefit from a more free trajectory. Efficient SLAM recovery may also be a solution in such case.

In line with previous studies (Quattrini Li et al., 2017; Hidalgo et al., 2018; Joshi et al., 2019), we focus on ORB-SLAM (Mur-Artal et al., 2015) and DSO (Engel et al., 2018)-based works, as discussed in Section 2.4. Section 4.2.1 recaps the algorithms under study. Section 4.2.2 introduces the methodology used in the benchmark, including test datasets selection and evaluation criteria. Finally, evaluation results are presented and discussed in Section 4.2.3, leading to a conclusion in Sections 4.2.4.

### 4.2.1 Algorithms under study

As discussed in Section 2.4.8, ORB-SLAM (Mur-Artal et al., 2015) is reported as robust to underwater visual conditions but subject to initialization difficulties and critical relocalization failure after tracking loss, hence the interest in ORB-SLAM-based works featuring improved SLAM failure recovery functionalities. Section 4.2 focuses on the works ORB-SLAM (Huang et al., 2020) and ORB-SLAM Atlas (Elvira et al., 2019) as released in ORB-SLAM3 (Campos et al., 2021). Note that, in the following, “ORB-SLAM3” simply refers to the ORB-SLAM Atlas implementation of ORB-SLAM3.

Conversely, DSO is described as able to cope with poorly textured environment and blur, but still limited by the absence of place recognition functionalities for loop closing and map reuse *via* relocalization, leading respectively to map inconsistency and a lack of robustness to bad data association, and to critical SLAM failure in case of tracking loss. Section 4.2 focuses on DSO-based works which implement long-term data association, what may limit these failures. Namely, the focus is made on

LDSO (Loop Closure Direct Sparse Odometry) (Gao et al., 2018) and DSM (Direct Sparse Mapping) (Zubizarreta et al., 2020).

While a detailed description of the VSLAM works under study is available in Section 2.4, Table 4.1 recaps the main features of these algorithms. ORB-SLAM Atlas (Elvira et al., 2019) and ORB-SLAM (Huang et al., 2020) are expected to allow the SLAM to recover more efficiently from tracking loss than ORB-SLAM, whereas LDSO (Gao et al., 2018) and DSM (Zubizarreta et al., 2020) are expected to produce less tracking loss than DSO due to better mapping performances through long term data association handling.

TABLE 4.1: Overview of the algorithms under study

| | **Front-end** | **Loop closing** | **SLAM recovery** |
|---|---|---|---|
| ORB-SLAM<br>(Mur-Artal et al., 2015) | Indirect, sparse | DBoW2 | Relocalization (DBoW2) |
| ORB-SLAM3<br>(Campos et al., 2021) | Indirect, sparse | DBoW2 | Relocalization (DBoW2)<br>If failure: new map creation with DBoW2 map matching and merging. |
| ORB-SLAM<br>(Huang et al., 2020) | Indirect, sparse | DBoW2 | New map creation.<br>Backwards SLAM for map matching and merging. |
| DSO<br>(Engel et al., 2018) | Direct, sparse | ∅ | ∅ |
| LDSO<br>(Gao et al., 2018) | Direct, sparse | DBoW3 | ∅ |
| DSM<br>(Zubizarreta et al., 2020) | Direct, sparse | Direct, using pose prior | ∅ |

### 4.2.2 Methodology

The evaluation was conducted using eight underwater datasets featuring various visual conditions. Section 4.2.2.1 explains the dataset selection and describes the eight chosen sequences. The evaluation method, including comparison criteria, is detailed in Section 4.2.2.2.

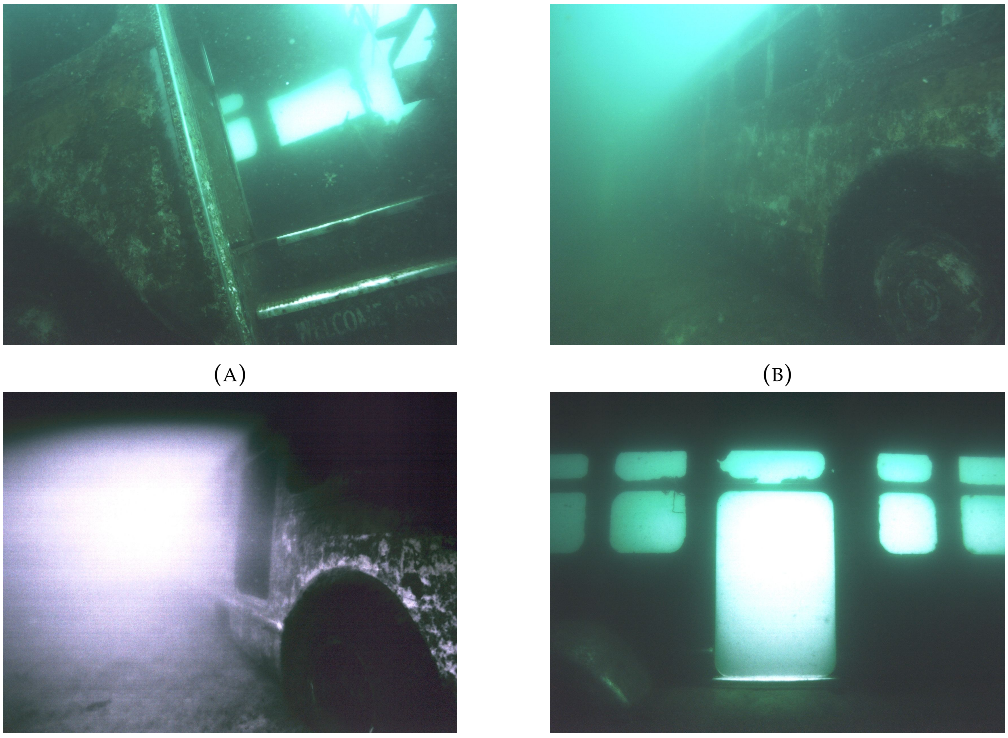
#### 4.2.2.1 Datasets

Airborne VSLAM evaluation can rely on standard, public datasets recorded in different environments, featuring several sequences in the similar environments and visual conditions with various trajectories of gradual difficulty (Geiger et al., 2012; Burri et al., 2016; Schubert et al., 2018). However, there is no equivalent in the underwater field at the time of writing, because of the important cost and resources required for acquiring such data. Previous works on VSLAM benchmark under underwater conditions released their evaluation datasets, which are composed of heterogeneous sequences recorded in completely different environments from one to another, with various lighting conditions and camera settings (Quattrini Li et al., 2017; Joshi et al., 2019). Such heterogeneous datasets are particularly interesting for comparing VSLAM methods under very different conditions, but are not suitable for a detailed evaluation under specific conditions. (Ferrera et al., 2019) released AQUALOC, an underwater visual-inertial-pressure dataset. Similarly to standard aerial datasets, it is composed of several gradually more difficult sequences recorded in similar environments, on three different marine sites. All theses sequences, however, show quite similar conditions by featuring man-made objects lying on a planar sandy area and involving only slow camera motion As a result, the AQUALOC dataset only represents a small portion of the wide variety of underwater environments and visual conditions. The underwater environment can be highly structured, such as caves, shipwrecks or offshore infrastructure, or highly unstructured, such as natural underwater reliefs or sandbanks. In addition, because lighting conditions and scene color aspect depend on the depth, the distance to the scene and the presence of embedded lights, these conditions can also be very different between two underwater scenes. The presence of turbidity, suspended particles or mobile objects or creatures is another factor in the variability of underwater scenes aspect.

The generation of ground truth trajectories relative to underwater datasets is more difficult than for aerial datasets. Whereas airborne datasets' ground truth commonly rely on laser scans or, sometimes, motion capture systems in smaller scale indoor environments, such systems are not available in the sea. In (Joshi et al., 2019), the output trajectory of a visual-inertial-SONAR-depth SLAM (Rahman et al., 2019) is used as a reference for visual and visual-inertial SLAM evaluation, but this can only apply to data acquired with a very specific sensor system. In AQUALOC (Ferrera et al., 2019), the offline Structure-from-Motion Colmap is used to compute a reference trajectory. These two strategies assume that the use of more sensors or time and computational resources will lead to a more reliable state estimation than real-time visual-only SLAM. While the absence of a ground truth only allows a coarse comparison between VSLAM approaches, it has been shown that such qualitative evaluations are already sufficient to discriminate most monocular VSLAM works in underwater fields (Joshi et al., 2019). This is why the present work only considers qualitative criteria to evaluate localization and mapping accuracy. In addition, the no-need in ground truth allows diversifying the test sequences. Evaluation criteria will be detailed in Section 4.2.2.2.

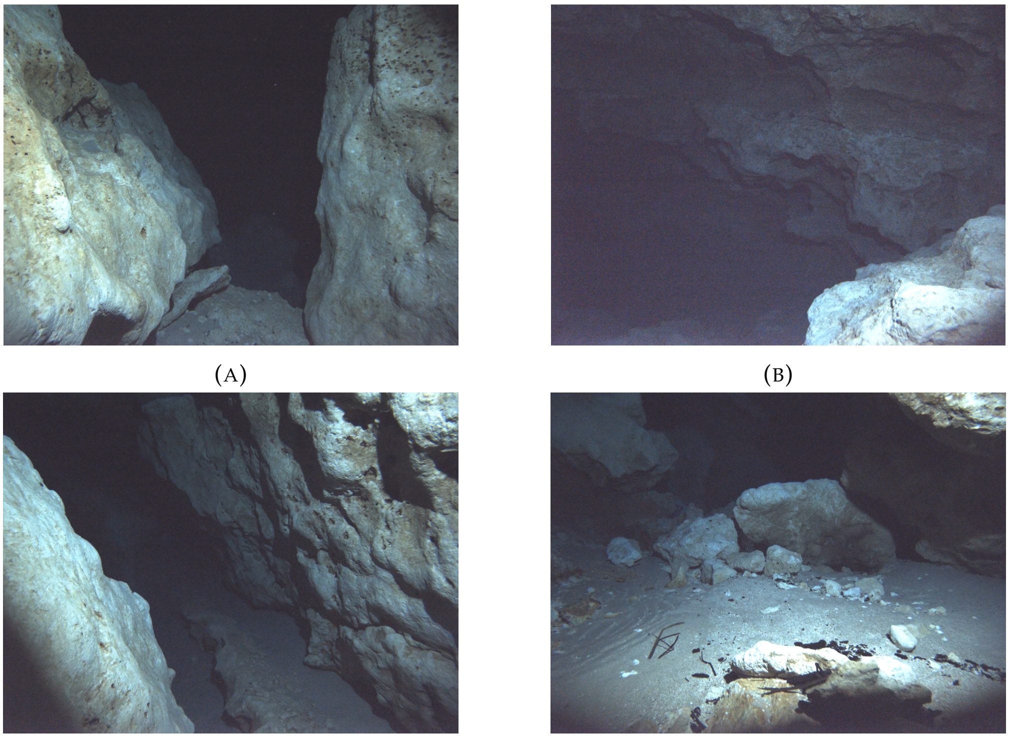
Section 4.2 aims at qualitatively characterizing the performances of VSLAM methods in the underwater field. Therefore, evaluations are conducted on a selection of eight datasets chosen to have different environments and visual conditions. The main characteristics of these datasets are recapped in Table 4.2.

The *Bus* dataset (Joshi et al., 2019) (Figure 4.1) is recorded in quite turbid water, by a forward facing RGB camera. The camera slowly turns around a sunken bus, hence several loop closures. The camera enters inside the bus during a small part of the trajectory (Figure 4.1d), and one side of the bus is poorly illuminated (Figure 4.1c). Therefore, this sequence features important visual conditions variations along the camera's trajectory, which is the main difficulty of this sequence.



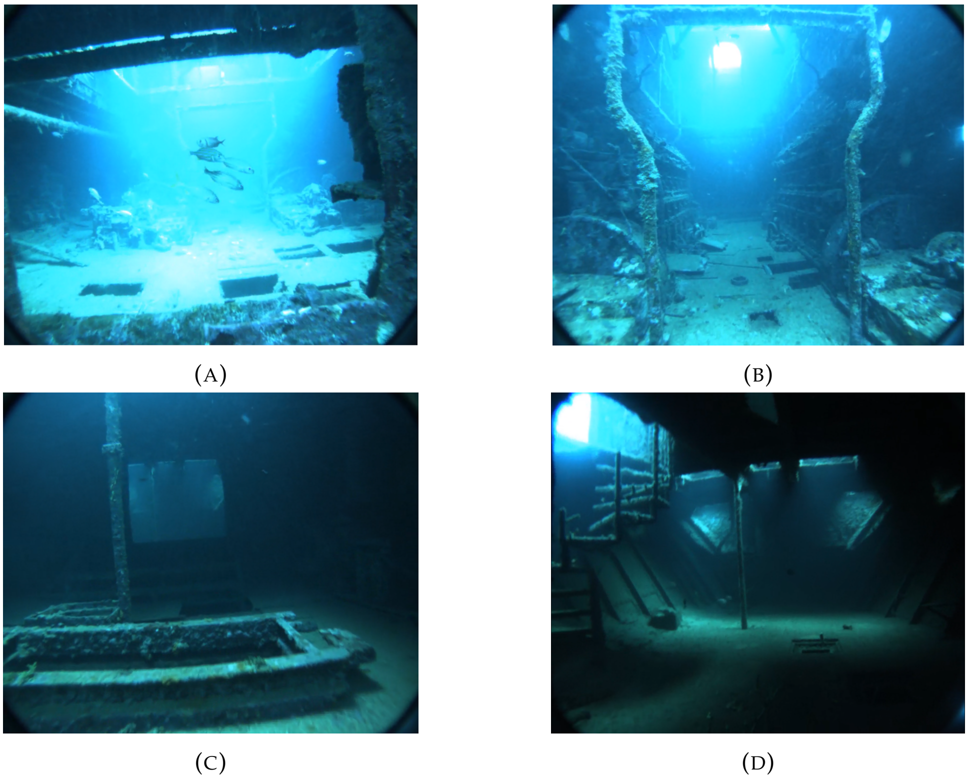
(A) (B) (C) (D)

FIGURE 4.1: *Bus* dataset (Joshi et al., 2019)



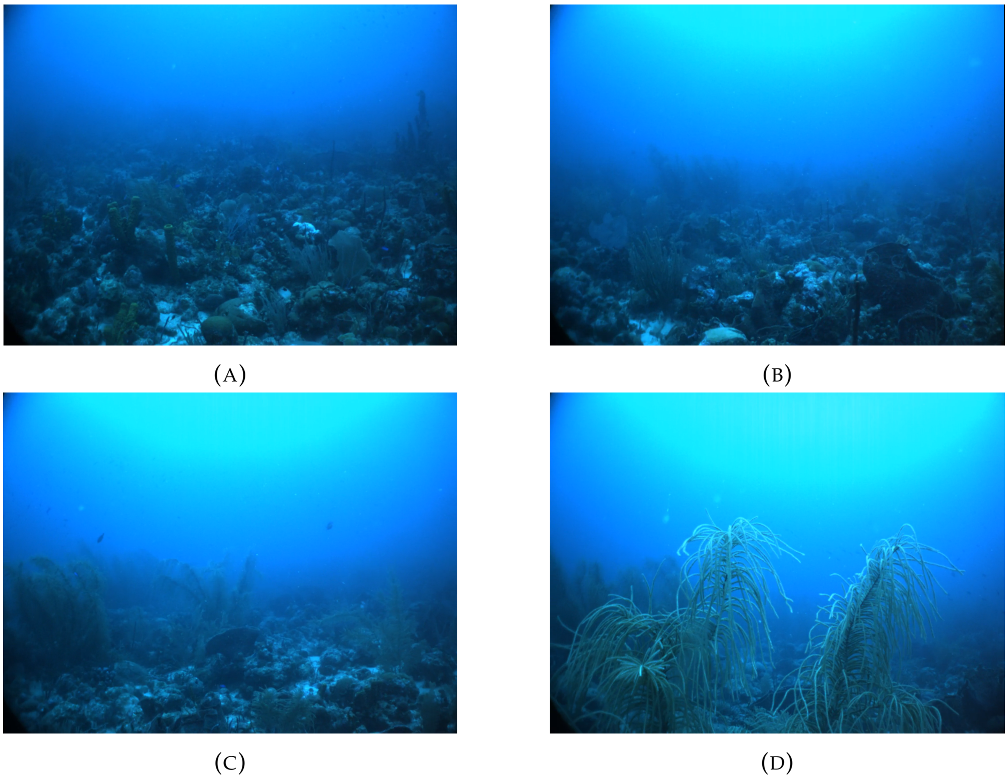
(A) (B) (C) (D)

FIGURE 4.2: *Cave* dataset (Joshi et al., 2019)



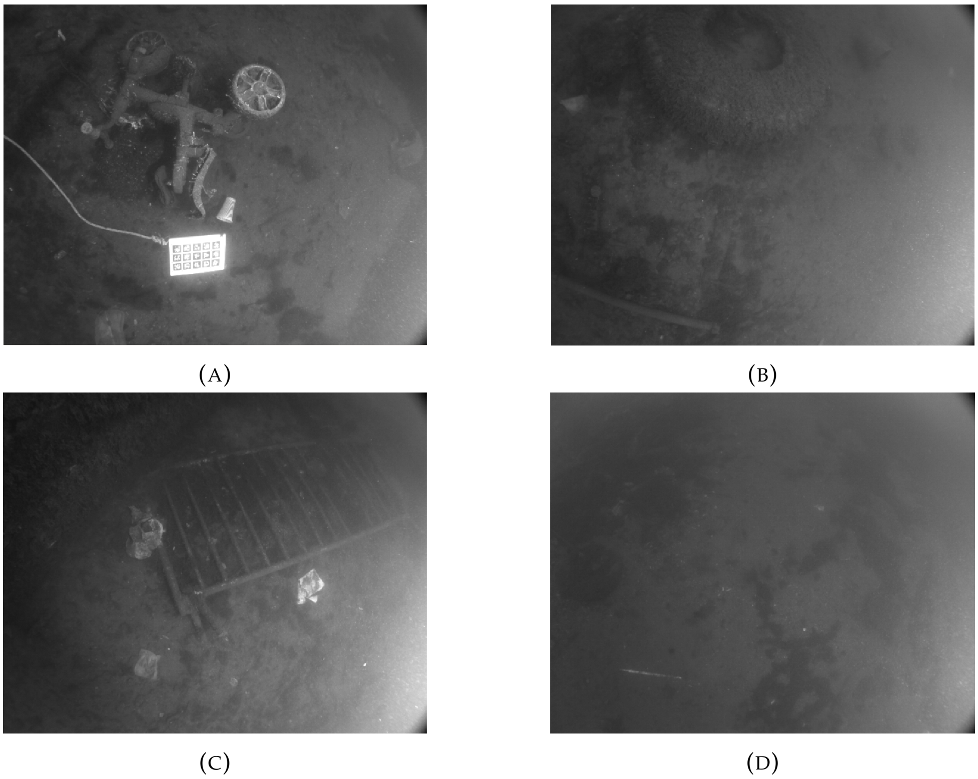
(A) (B) (C) (D)

FIGURE 4.3: *A/In* dataset (Quattrini Li et al., 2017)



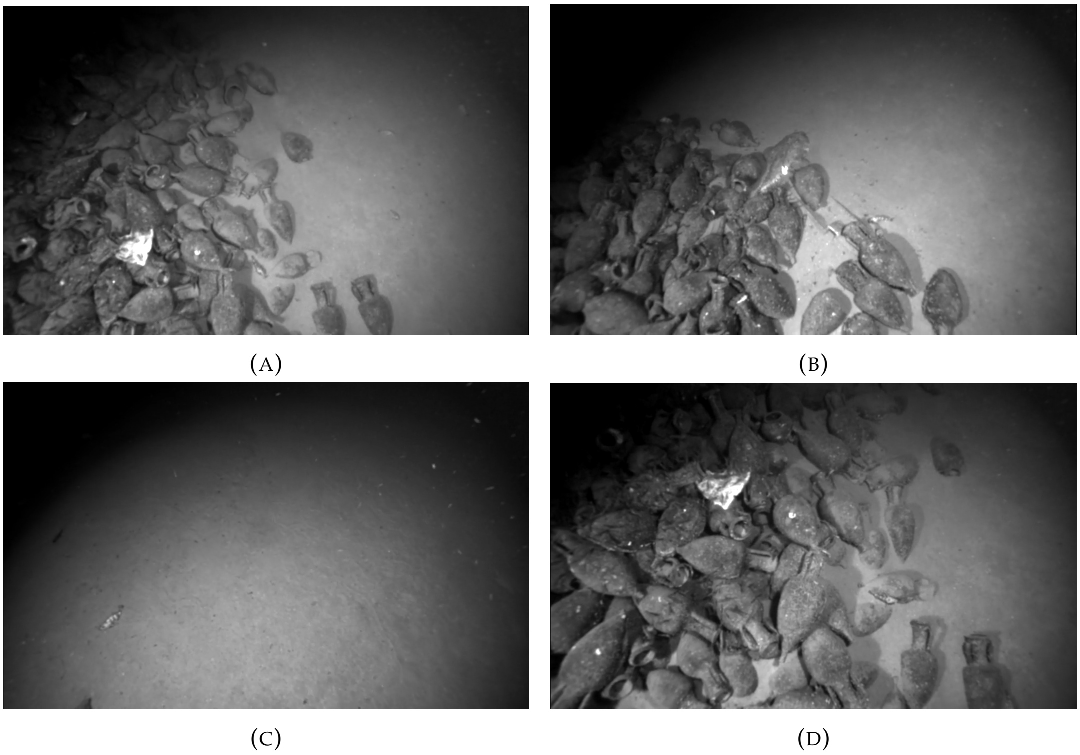
(A) (B) (C) (D)

FIGURE 4.4: *A/Out – Line #1* dataset (Quattrini Li et al., 2017)


(A) (B)

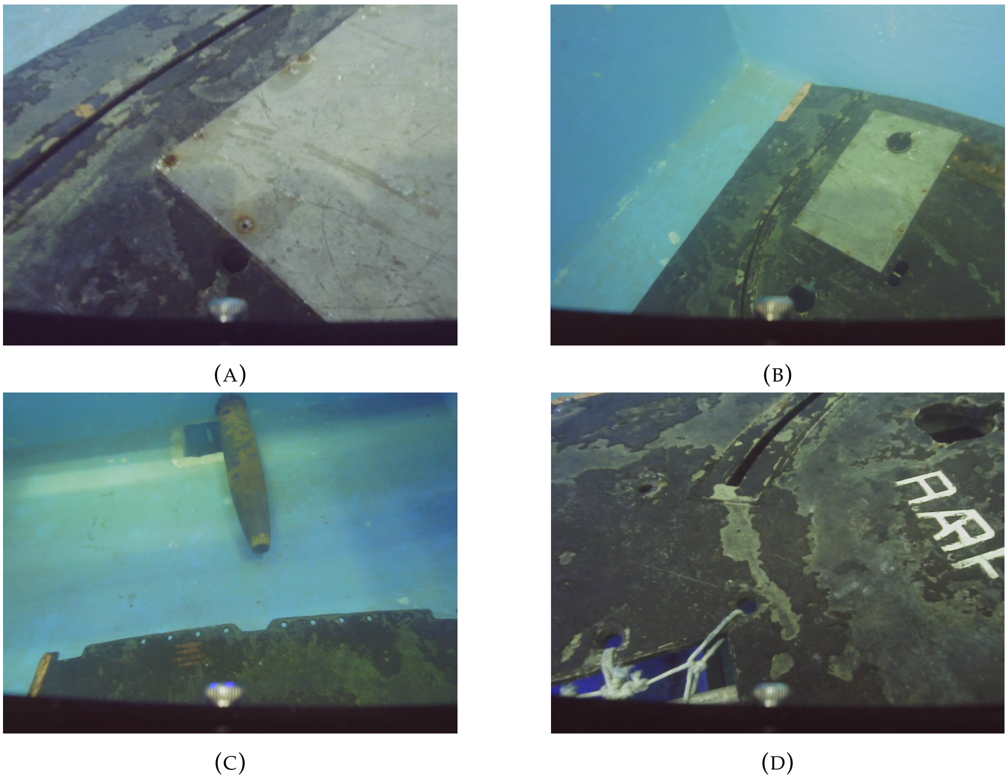
(C) (D)

FIGURE 4.5: *Aqualoc Harbor (AH) #01* dataset (Ferrera et al., 2019)


(A) (B)

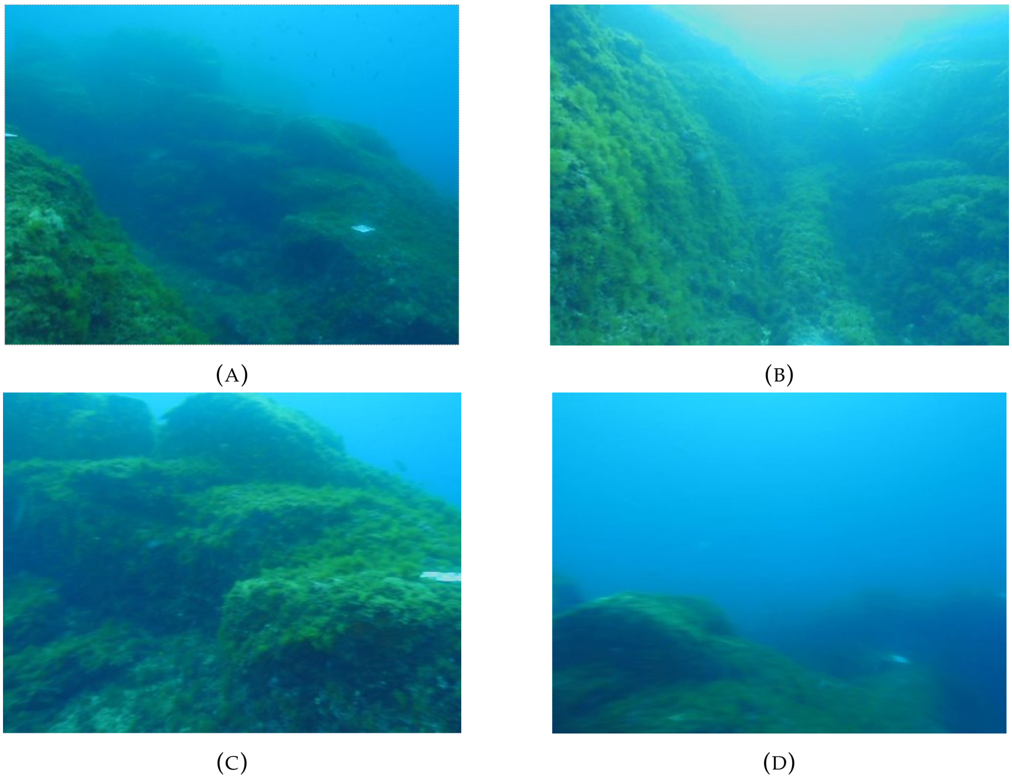
(C) (D)

FIGURE 4.6: *Aqualoc Archaeo (AA) #09* dataset (Ferrera et al., 2019)



(A) (B) (C) (D)

FIGURE 4.7: *Cephismer* dataset



(A) (B) (C) (D)

FIGURE 4.8: *St-Raphael* dataset

TABLE 4.2: Main characteristics of the evaluation datasets

| | **Camera** | **Duration** | **Embedded light** | **Loop closure(s)** | **Depth** |
|---|---|---|---|---|---|
| *Bus*<br>(Joshi et al., 2019) | RGB<br>1200×1600 pixels<br>12.5 Hz | 584 s | ∅ | ✓ | 20 m |
| *Cave*<br>(Joshi et al., 2019) | RGB<br>1200×1600 pixels<br>12.5 Hz | 709 s | ✓ | ✓ | 20 m |
| *A/In*<br>(Quattrini Li et al., 2017) | RGB<br>640×776 pixels<br>15 Hz | 88 s | ∅ | ∅ | Unknown |
| *A/Out – Line #1*<br>(Quattrini Li et al., 2017) | RGB<br>640×776 pixels<br>4 Hz | 53 s | ∅ | ∅ | Unknown |
| *AQUALOC Harbor #01*<br>(Ferrera et al., 2019) | Grayscale<br>512×640 pixels<br>20 Hz | 289 s | ✓ | ✓ | 3 m |
| *AQUALOC Archaeo #09*<br>(Ferrera et al., 2019) | Grayscale<br>608×968 pixels<br>20 Hz | 349 s | ✓ | ✓ | 380 m |
| *Cephismer* | RGB<br>480×640 pixels<br>5 Hz | 156 s | ∅ | ✓ | 1.5 m |
| *St-Raphael* | RGB<br>480×640 pixels<br>20 Hz | 472 s | ∅ | ✓ | 20 m |

The *Cave* dataset (Joshi et al., 2019) (Figure 4.2) is recorded in an underwater cave, with an embedded light source. It shows a natural mineral-only environment in clear water. The RGB camera is facing forward, and its motion is slow. The sequence includes several loop closures.

The *A/In* dataset (Quattrini Li et al., 2017) (Figure 4.3) is recorded inside a shipwreck, with a forward facing RGB camera. Environment structure is thus closer to standard airborne indoor VSLAM evaluation datasets. The sequence mainly consists in a forward travelling and does not include any loop closure. Water is globally clear, but parts of the sequence feature suspended particles and fishes.

The *A/Out – Line #1* dataset (Quattrini Li et al., 2017) (Figure 4.4) shows a coral reef, including some mobile elements like seaweeds, fishes and suspended particles. The RGB camera is facing forward. No loop closure is included.

The *Aqualoc Harbor #01* dataset (Figure 4.5) is recorded with a downwards looking grayscale fisheye camera, and shows large man-made objects lying on the sand. The sequence includes a loop closure, which is marked by an apriltag target (Figure 4.5a).

The *Aqualoc Archaeo #09* dataset (Figure 4.6) involves a grayscale camera which is slightly tilted downwards. The sequence features amphora hills with high texture, but also low textured sandy areas (Figure 4.6c). Images show turbidity and backscattering. The sequence includes loop closures. Both *Aqualoc Harbor #01* and *Aqualoc Archaeo #09* use an embedded light source.

Both *Cephismer* and *Saint-Raphael* datasets are new datasets recorded by the embedded RGB camera of a BlueROV2, in challenging visual conditions. The *Cephismer* dataset (Figure 4.7) is recorded in a pool. The camera is slightly tilted downwards, and a small portion of the housing appears in its field of view. The sequence features fast motion, including pure rotations, around submarine spare parts. It is recorded at a low frame rate, with the camera sometimes facing poorly textured areas. This dataset is thus particularly challenging. It also includes several loop closures.

In the *Saint-Raphael* dataset (Figure 4.8), the ROV's camera is facing forward. The sequence is recorded at shallow depth, in the Mediterranean Sea, in turbid water. The sequence includes fast motions and pure rotations, and the camera sometimes happens to face plain water. Several loop closures are included. This dataset is also very challenging.

Contrary to the other sequences, the *Cephismer* and *Saint-Raphael* datasets feature fast motion, and occasional loss of view of the surrounding objects and of the seafloor. These sequences are therefore representative from a ROV use case where ROV motion is not specifically constrained to simplify visual perception.

#### 4.2.2.2 Evaluation method

Sequences are processed with ORB-SLAM (Mur-Artal et al., 2015), ORB-SLAM3 (Campos et al., 2021), ORB-SLAM (Huang et al., 2020), DSO (Engel et al., 2018), LDSO (Gao et al., 2018) and DSM (Zubizarreta et al., 2020). Since ORB-SLAM, ORB-SLAM and DSM do not support fisheye camera models, these three approaches are not evaluated on the *Aqualoc Harbor #01* dataset. When possible, evaluations are conducted in real-time conditions, using ROS middleware. The only approaches that do not implement real-time processing are LDSO and DSM. The evaluation of these two methods are therefore non-real-time in the present work.

The parameters of each VSLAM method are tuned manually for each dataset, following all available documentation provided by the authors. For each VSLAM approach, the following criteria are evaluated:

- *Ability to track the complete trajectory*, evaluated by the percentage of the sequence duration for which a localization is computed. One can notice that this does not take into account the reliability of the estimated pose. This criteria is common with the work (Joshi et al., 2019).
- *Loop closure detection and handling capability*, evaluated qualitatively by a color-coded rating which indicates wether the SLAM system manages to detect and process the main sequence's loops (*green*), only a few of them (*yellow*), or none of them (*red*).
- *Localization and mapping consistency*, also evaluated qualitatively by a color-coded rating. A *green* one means that the SLAM outputs consistent trajectory and map on almost all the sequence duration. A *yellow* one indicates consistent trajectory and map on more than half of the sequence, and an *orange* one corresponds to consistent trajectory and map during less than half of the sequence.

Lastly, a *red* mark indicates that the SLAM is unable to initialize or produces completely inconsistent outputs.

In the works (Quattrini Li et al., 2017; Joshi et al., 2019), loop closure detection and handling capability and localization and mapping consistency are evaluated together by a qualitative mark indicating failure, partial failure, partial success or success. We chose to separate these two qualitative criteria and make more explicit the definition of the four possible color-coded ratings.

### 4.2.3 Results

Evaluations are carried out in real-time on a computer with an Intel i7-10610U CPU @ 1.80GHz × 8, 16 GB RAM, running Ubuntu 18.04 and ROS Melodic. In order to take into account the non-deterministic behavior of multithreaded applications, the reported observations are based on the median out of 3 runs per SLAM for each dataset. Results are reported in Table 4.3, including the percentage of the sequence for which a localization is computed, and qualitative loop closure capability and consistency marks as defined in Section 4.2.2.2. ORB-SLAM3 being a multimap SLAM system, the final number of disconnected maps is also indicated.

TABLE 4.3: Results

| | | **ORB-SLAM** | **ORB-SLAM3** | **Dual-SLAM** | **DSO** | **LDSO** | **DSM** |
|---|---|---|---|---|---|---|---|
| Bus | % localized | 38.08 | 64.03 (1 map) | 33.83 | 15.74 | 21.25 | 14.65 |
| | Loop closure | 🟨 | 🟩 | 🟨 | ∅ | 🟥 | 🟥 |
| | Consistency | 🟨 | 🟨 | 🟨 | 🟧 | 🟧 | 🟥 |
| Cave | % localized | 79.81 | 99.96 (1 map) | 99.78 | 7.89 | 2.97 | 46.26 |
| | Loop closure | 🟩 | 🟩 | 🟩 | ∅ | 🟥 | 🟥 |
| | Consistency | 🟩 | 🟩 | 🟩 | 🟧 | 🟧 | 🟧 |
| A/In | % localized | 91.23 | 99.62 (1 map) | 99.62 | 99.03 | 99.00 | 99.60 |
| | Loop closure | ∅ | ∅ | ∅ | ∅ | ∅ | ∅ |
| | Consistency | 🟩 | 🟩 | 🟩 | 🟩 | 🟩 | 🟨 |
| A/Out | % localized | 0.0 | 86.07 (1 map) | 0.0 | 92.55 | 99.0 | 99.0 |
| | Loop closure | ∅ | ∅ | ∅ | ∅ | ∅ | ∅ |
| | Consistency | 🟥 | 🟩 | 🟥 | 🟨 | 🟩 | 🟥 |
| AH | % localized | ∅ | 99.71 (1 map) | ∅ | 77.73 | 79.08 | ∅ |
| | Loop closure | ∅ | 🟩 | ∅ | ∅ | 🟥 | ∅ |
| | Consistency | ∅ | 🟩 | ∅ | 🟨 | 🟨 | ∅ |
| AA | % localized | 99.79 | 99.51 (1 map) | 99.72 | 88.07 | 99.0 | 82.50 |
| | Loop closure | 🟩 | 🟩 | 🟩 | ∅ | 🟥 | 🟨 |
| | Consistency | 🟩 | 🟩 | 🟩 | 🟨 | 🟨 | 🟨 |
| Cephismer | % localized | 3.88 | 66.72 (5 maps) | 3.88 | 28.50 | 42.15 | 41.03 |
| | Loop closure | 🟥 | 🟨 | 🟥 | ∅ | 🟥 | 🟥 |
| | Consistency | 🟥 | 🟨 | 🟥 | 🟥 | 🟧 | 🟥 |
| St-Raphael | % localized | 6.81 | 64.46 (12 maps) | 3.43 | 9.51 | 9.56 | 0.05 |
| | Loop closure | 🟥 | 🟧 | 🟥 | ∅ | 🟥 | 🟥 |
| | Consistency | 🟧 | 🟨 | 🟧 | 🟥 | 🟥 | 🟥 |

First, one can notice that none of the tested methods manages to completely process all sequences with visually consistent trajectory and map. In addition, DSO-based approaches give particularly poor performances compared to ORB-SLAM-based ones.

One can see from Table 4.3 that ORB-SLAM fails to process important parts of the test sequences. This is caused by important initialization delays (ORB-SLAM even fails to initialize on the *A/In* sequence) and tracking failure recovery disability, hence the need for SLAM recovery strategies. Whereas ORB-SLAM's failure recovery strategy seems inefficient in the test datasets, leading to similar performances than ORB-SLAM, one can see that ORB-SLAM3 significantly improves the localization capabilities from ORB-SLAM. Indeed, ORB-SLAM3 outputs a localization on longer sequence portions, due to faster initialization and new map creation in case of SLAM failure, which allows keeping the SLAM running. ORB-SLAM3's initialization's algorithm is the same as ORB-SLAM but with a different library, which may lead to faster computation and explain the improved initialization capabilities of ORB-SLAM3. ORB-SLAM3's tracking failure handling strategy with new map initialization is particularly interesting.

In the *Bus* dataset, this strategy allows covering a more important portion of the sequence than ORB-SLAM, since the system does not have to wait to reach an already mapped area to keep running. In difficult sequences leading to repetitive tracking failures, like in the *Cephismer* and *St-Raphael* datasets, this multimap approach results in disconnected trajectory parts and submaps, which cover an important portion of the full video sequence. Finally, all ORB-SLAM-based approaches show the same good loop closure detection and handling capabilities, which appears to be robust to underwater conditions. However, the same place recognition module fails to detect most map overlaps when running ORB-SLAM3 on the *Cephismer* and *St-Raphael* scenarios, failing to fuse these maps into a global one in these particularly difficult sequences.

Whereas ORB-SLAM based approaches, and namely ORB-SLAM3, manage to consistently process the majority of the duration of all test sequences, DSO-based methods prove to be far less robust to underwater visual conditions. Similarly to (Joshi et al., 2019), we observe that DSO is able to process at least partially some of the sequences and produce a quite realistic map of the environment during this time interval. The best DSO results are observed for the most structured environments, and in particular for the *A/In*, *Aqualoc Harbor* and *Aqualoc Archaeo* datasets which feature man-made objects. Images from these datasets show quite clear object contours, which are more adequate for DSO's tracking, which relies on close-to-contours pixel patches.

DSO also suffers from local tracking inconsistencies, drift, and the incapacity to recover from tracking failure that may happen soon after initialization like in the *Cave*, *Cephismer* and *St-Raphael* datasets. As expected, these limits highlight the interest of extending DSO with loop closing capabilities in order to reduce or correct these local tracking inconsistencies and lead to a more reliable mapping of the environment, for more robustness. However, both LDSO and DSM's loop closing implementations on a DSO basis show important limitations in the underwater field. LDSO fails to detect loops, and shows similar performances than DSO in terms of delay before failure and SLAM consistency. It is also very slow compared to DSO, and requires up to several seconds to process a single image. LDSO's loop detection process is very close to the one of ORB-SLAM, which manages to detect and process most loop closures. This difference in loop detection success with very close detection methods might be caused by the bad triangulation of map points used for loop detection, which prevents any geometric consistency validation. Finally, DSM also fails to give better performances than DSO on the evaluated underwater scenarios. It is also extremely slow, up to dozens of seconds per frame. In addition, DSM's loop detection strongly relies on a good pose prior, resulting in loop detection failure in all sequences because of localization drift. In the *Cave* dataset, bad pose prior even leads to false loop detections, which decrease SLAM performances. On the other hand, in the *Aqualoc Archaeo* dataset, DSM manages to detect accurate covisibilities between recent but first disconnected KF, hence the yellow loop closing color-coded rating. Successfully including older covisible KF with complementary points of view in the tracking window leads to a less important trajectory drift than DSO and LDSO on this sequence, as represented in Figure 4.9. Trajectories are given with an unknown scale. They are aligned by a Umeyama $Sim(3)$ alignment with respect to the most complete trajectory, namely the one given by ORB-SLAM3. One can see the drift of DSO-based approaches (DSO, LDSO, DSM) compared to ORB-SLAM-based ones (ORB-SLAM, ORB-SLAM3, ORB-SLAM). Note that ORB-SLAM and ORB-SLAM estimated trajectories almost completely overlap in Figure 4.9. The trajectories obtained on all datasets for all SLAM algorithms on their run with median localization percentage among the three test runs are represented in Appendix E.

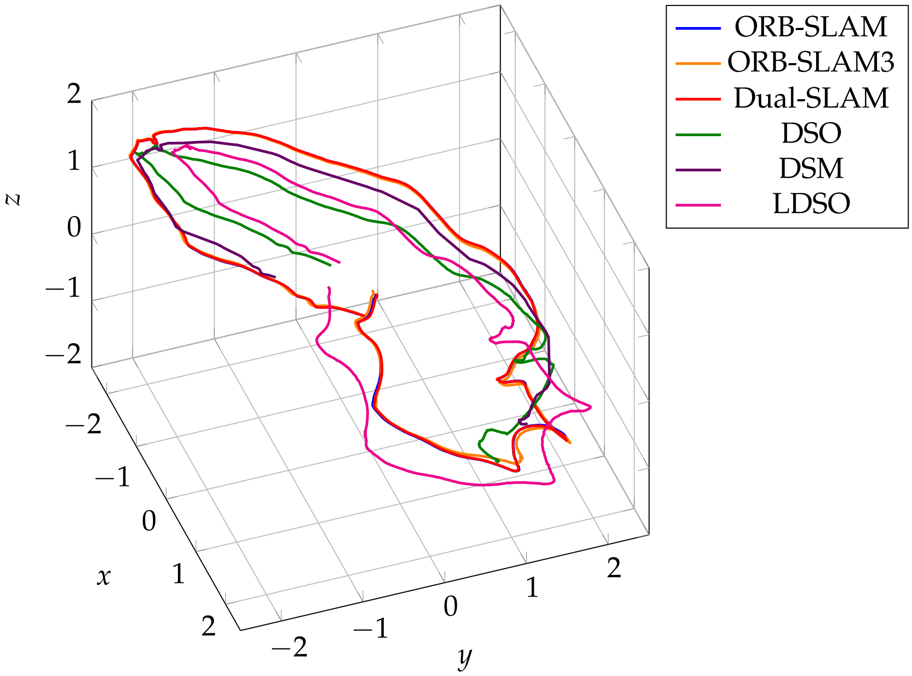


FIGURE 4.9: Trajectories estimated by the six SLAM algorithms on the *Aqualoc Archaeo* dataset.

### 4.2.4 Conclusions

Section 4.2 provides a qualitative underwater evaluation of recent monocular developments on DSO (Engel et al., 2018) and ORB-SLAM (Mur-Artal et al., 2015). The main underwater VSLAM challenges appear to be the handling of tracking loss and place recognition. We demonstrated that DSO's recent loop closure extensions DSM (Zubizarreta et al., 2020) and LDSO (Gao et al., 2018) are not robust enough to tracking failure. However, the recent multimap extension ORB-SLAM3 (Campos et al., 2021) of ORB-SLAM seems very promising and allows keeping computing a localization and a map even after a tracking loss with relocalization failure. While ORB-SLAM3 appears to be already quite robust to underwater challenging visual conditions, it may although be improved by investigating a more robust place recognition algorithm for map fusion. The other ORB-SLAM multimap SLAM recovery extension evaluated, ORB-SLAM (Huang et al., 2020), does not demonstrate important robustness improvement compared to ORB-SLAM in the scenarios evaluated.

Finally, ORB-SLAM3 (Campos et al., 2021) appears to be the most robust monocular VSLAM solution in handling underwater scenarios, at present time. Even though its map overlap detection algorithm may be improved, ORB-SLAM3 can already be a solution for underwater visual-based, monocular, localization and mapping in scenarios avoiding fast motions and featuring textured areas.

## 4.3 MAM$^3$SLAM underwater-robust multi-agent and multi-map VSLAM

This section investigates multi-agent monocular VSLAM for the localization of multiple underwater ROVs deployed in a common area. This scenario corresponds to a chain of ROVs, but can also apply to other multi-ROV applications including archaeological exploration and recovery missions. These applications need individual but also inter-robot localization within their environment in order to operate safely. Underwater multi-agent SLAM is generally addressed in the case of AUVs which have limited communication through the water medium (Mangelson et al., 2018; Özkahraman et al., 2022). In contrast, multi-ROV systems can communicate in real time with a central server, fitting the standard centralized or partially decentralized communication configuration of airborne multi-agent VSLAM systems.

In accordance with the observations of Section 4.2, which highlights the robustness of ORB-SLAM3 (Campos et al., 2021) to underwater conditions, Section 4.3.1 introduces MAM$^3$SLAM (multi-agent and multi-map monocular SLAM), a new fully centralized multi-agent and multi-map monocular VSLAM framework based on ORB-SLAM3. While the main motivation is underwater multi-agent tasks, which correspond to a particulalry challenging environment,, this work equally applies for airborne applications. MAM$^3$SLAM is evaluated and compared to the state-of-the-art multi-agent VSLAM on four two-agent scenarios, including one standard airborne dataset and three new underwater datasets recorded in a pool and in the sea. Section 4.3.2 presents the new underwater datasets and their collection. Section 4.3.3 describes the evaluation methodology, and Section 4.3.4 presents and discusses the results, before concluding in Section 4.3.1.

### 4.3.1 MAM$^3$SLAM algorithm

MAM$^3$SLAM builds on the ORB-SLAM Atlas (Elvira et al., 2019) implementation provided by ORB-SLAM3 (Campos et al., 2021). MAM$^3$SLAM algorithm is represented in Figure 4.10. A comparable representation of ORB-SLAM Atlas is provided in Section 2.4.6.

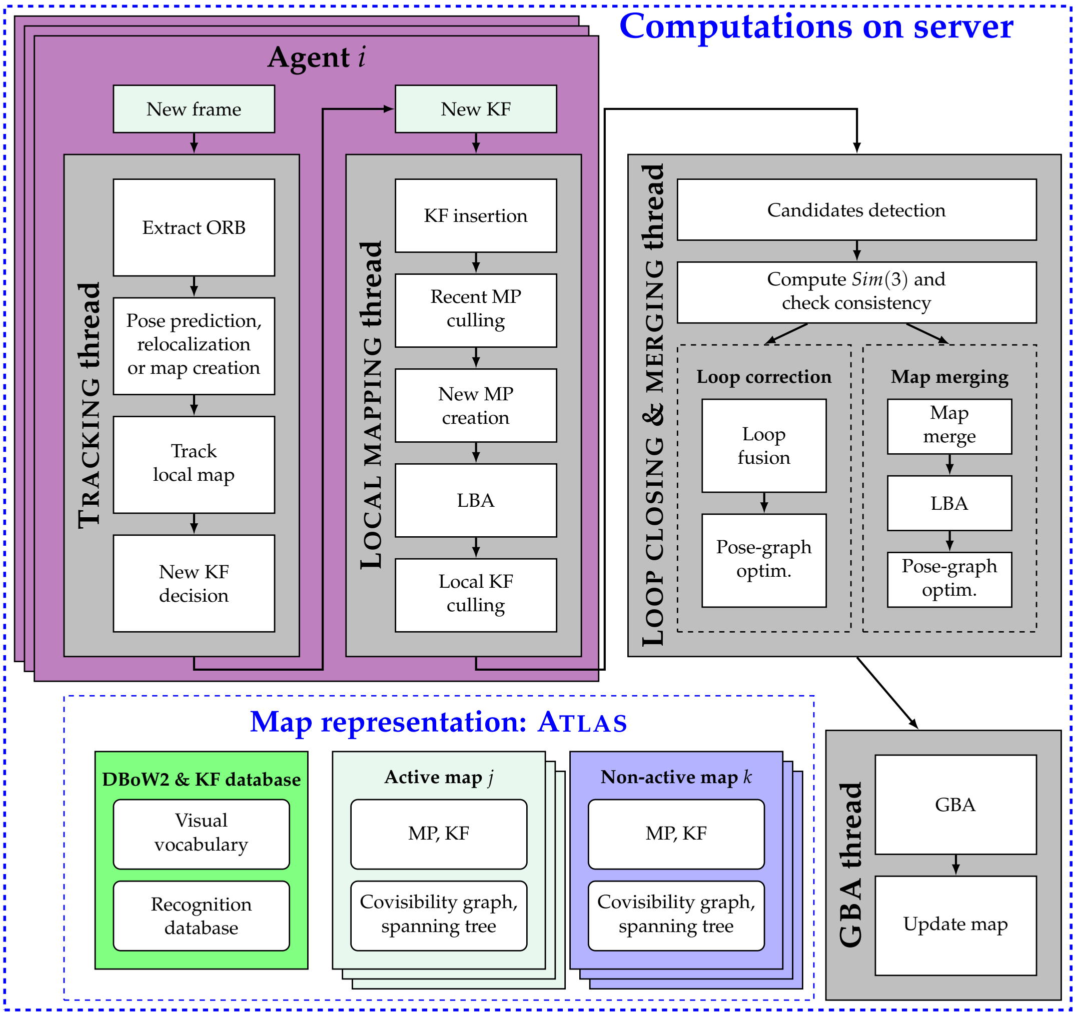


FIGURE 4.10: MAM$^3$SLAM multi-map representation and workflow

#### 4.3.1.1 Overall architecture

MAM$^3$SLAM is a central multi-agent SLAM, meaning all computations are performed on a central server. Agents only send frames to the server. A tracking thread and a local mapping thread are run for each agent, on the central server. Loop closing and map merging tasks are conducted by one unique, common thread in the multi-agent system. A system of $n$ agents then runs on $2n + 1$ threads, including $n$ tracking threads (one per agent), $n$ local mapping threads (one per agent) and 1 single, common loop closing and map merging thread. Similarly to ORB-SLAM Atlas, an additional global bundle adjustment thread is launched after a loop correction or map merging. Differences between ORB-SLAM Atlas and MAM$^3$SLAM are the following:

- The main contribution lies in creating multi-agent VSLAM instances, and making maps a shared resource between multiple agents which was not implemented in ORB-SLAM Atlas. To this end, multiple agents can localize in the same map and access and update its data.
- Multi-threading support was implemented to allow concurrent map access and update, hence the need to protect the map from concurrent modification.

- A new KF insertion algorithm was implemented for agents locating on the same map. This involves protection from concurrent modifications.

#### 4.3.1.2 Shared multi-map resources

Maps are initialized individually by the server for each agent but are shared among them using the multi-map representation proposed in ORB-SLAM Atlas and denoted as the *Atlas*. KFs from all maps are stored in a common database, to which all agents contribute. When a new KF is inserted, a place recognition query is performed over the KF database. If a match is found, a loop closing or a map merging operation is performed, depending on whether the matched KF belongs to the same map or not. This process is inspired by ORB-SLAM3, since having multiple agent inputs does not modify the intrinsic behavior of the Atlas. Two matched maps $\mathcal{M}_i$ and $\mathcal{M}_j$ become a single map $\mathcal{M}_k$ that includes all KFs from the original $\mathcal{M}_i$ and $\mathcal{M}_j$, and merges their map point observations. All the data originally contained in $\mathcal{M}_i$ and $\mathcal{M}_j$ is thus made available for tracking and local mapping via $\mathcal{M}_k$. Making the Atlas a central resource for multiple agents allows fusing maps created by different agents, enabling any agent to reuse the map KFs and map points created by another agent in its own tracking process. If lost, agents can also relocalize in any map, even if they never contributed to it. Furthermore, several agents can localize on the same map, enabling relative localization estimation and collaborative map incrementation.

A map is called *active* if at least one agent is currently localizing in it. Active maps are being updated by new KF insertions and local mapping. Contrary to ORB-SLAM3 where map fusion can only happen between the current active map and an old released map, MAM$^3$SLAM allows the fusion of two active maps. This operation is, however, fully consistent with the ORB-SLAM3 merging algorithm and does not affect the SLAM. When several agents localize on the same map, the map is protected from concurrent updates from agents' local mapping threads by a mutex.

#### 4.3.1.3 New KF insertion

In ORB-SLAM3, a new KF is inserted if (i) less than 90% of the map points of the current reference KF are visible in the current frame and if either (ii) more than 1 second passed since the last insertion or (iii) the local mapping is idle. The local mapping thread periodically checks if some new KFs are to be inserted and processes them one by one within a given period. However, in a multi-agent scenario, two agents on the same map may slow down their local mapping due to the need to wait for the mutex and thus delay the insertion of new KFs. Therefore, in MAM$^3$SLAM, KF insertion is modified so that the local mapping inserts all KFs into an insertion queue at each iteration. If many KFs are inserted, however, this may reduce the idle time of the local mapping, resulting in a decreasing number of new KF creation. As a result, in MAM$^3$SLAM, the KF insertion criterion of ORB-SLAM3 is modified to force KF insertion if more than five consecutive frames satisfy criterion (i) but not (ii) or (iii).

### 4.3.2 Underwater multi-agent datasets collection

To the best of our knowledge, there is no public, multi-agent, underwater VSLAM datasets at the time of writing. Multi-agent VSLAM datasets can be either recorded using multiple devices or simulated by playing simultaneously several video sequences recorded in the same environment. The first solution features real multi-agent cases, where agents may see each other. Such datasets can also integrate dynamic changes in the environment, which will be observed synchronously by the multiple agents. However, they represent a higher acquisition cost. The second solution is less realistic with respect to a real multi-robot application. It is impossible for the simulated agents to observe each other, and possible dynamic changes in the environments will be observed out of sync among the simulated agents. These datasets are although far simpler to acquire, because they only need the deployment of a single device, or can be realized from existing sequences recorded in the same static environment. For instance, in (Schmuck et al., 2019), several sequences of the EuRoC Machine Hall dataset (Burri et al., 2016) are used to simulate multiple agents. In (Daoud et al., 2018), the same thing is performed with the KITTI odometry dataset (Geiger et al., 2012).

As explained in Section 4.2.2.1, only few public underwater VSLAM datasets are available. Unfortunately, none of them includes sequences recorded in the exact same environment, or which may be divided in several disconnected sequences with spatial overlap. Consequently, there was a need for creating underwater multi-agent VSLAM datasets. Because of the operational cost of deploying multiple underwater robots simultaneously, we privileged playing multiple agents from multiple videos acquired on the same site in the absence of major aspect changes in the environment. Three two-agent datasets were collected this way, featuring different environments and challenges, denoted *Ifremer tank*, *Cephismer tank* and *Sea diving*, and illustrated in Figures 4.11, 4.12 and 4.13 respectively.

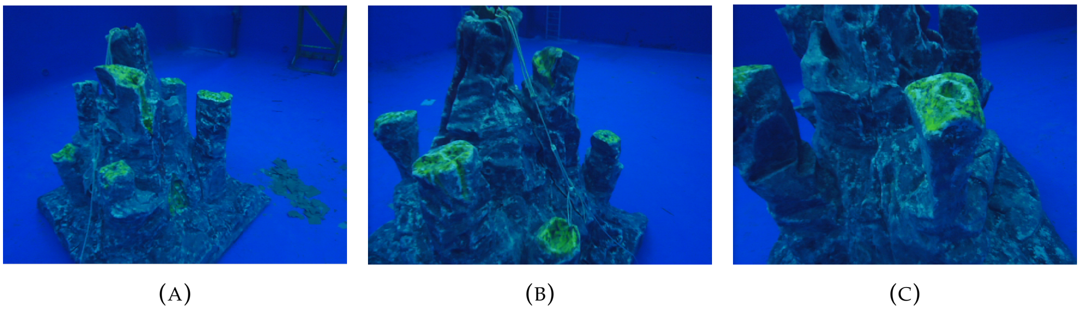

(A) (B) (C)

FIGURE 4.11: Overview of the *Ifremer tank* dataset

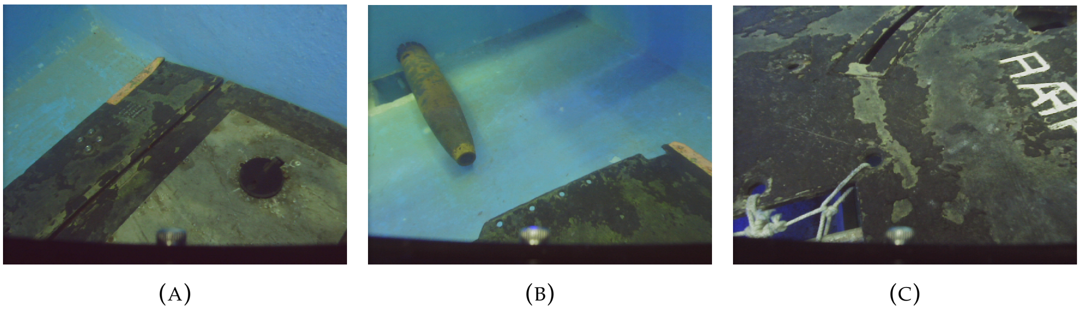

(A) (B) (C)

FIGURE 4.12: Overview of the *Cephismer tank* dataset

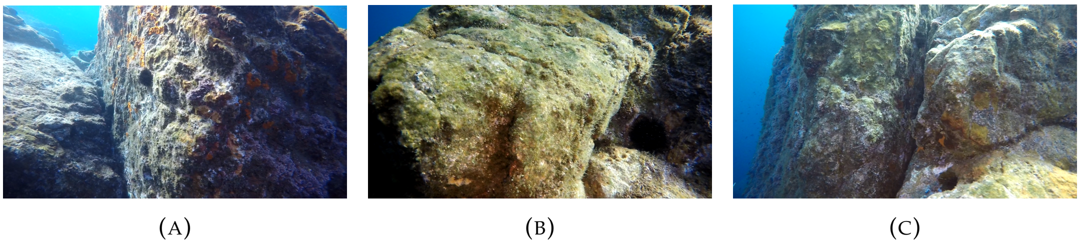

(A) (B) (C)

FIGURE 4.13: Overview of the *Sea diving* dataset

The *Ifremer* and *Cephismer tank* datasets are both recorded in a pool using the embedded camera of a BlueROV2. The *Ifremer tank* is a quite easy sequence featuring slow motion around a highly textured artificial marine reef. The agents move around the reef with different radii and depths (Figure 4.14a). This dataset allows evaluating the SLAM with easy underwater visual conditions, and allows validating the proposed collaborative mapping approach.

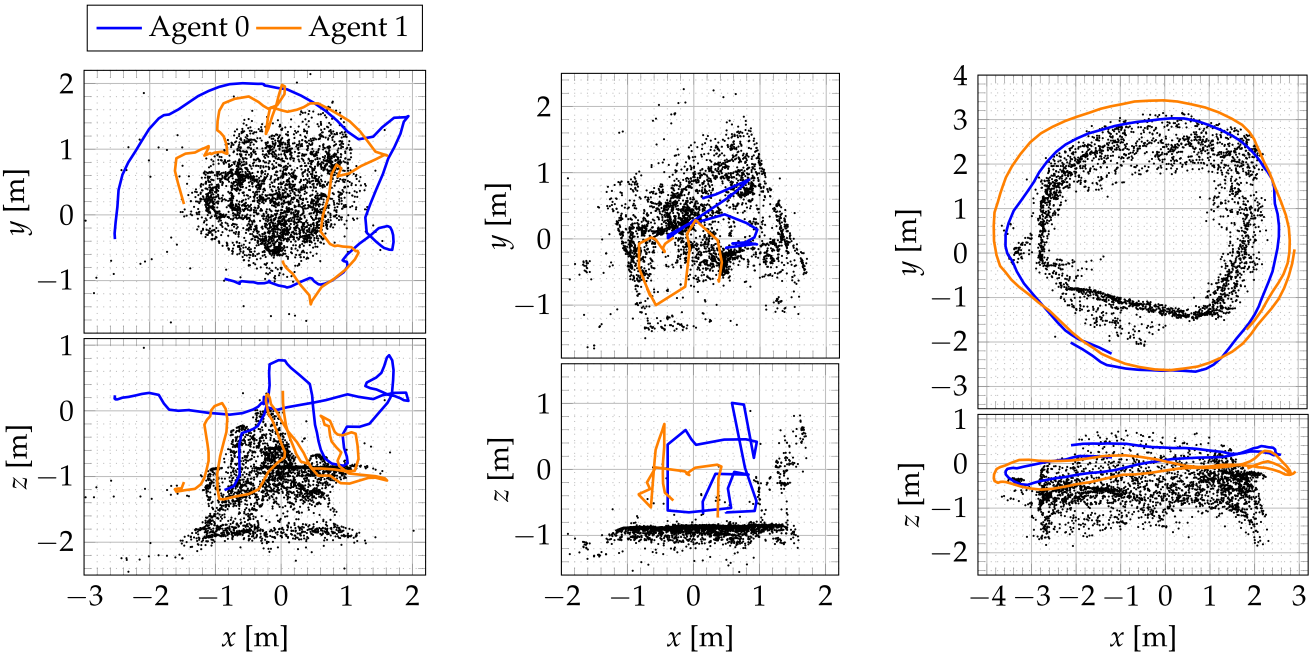
The *Cephismer tank* dataset features fast motion, including pure rotations, around submarine spare parts. It is recorded at low frame rate (5 Hz), and the camera sometimes faces poorly textured areas. This dataset is thus particularly difficult. Agents' trajectories are represented in Figure 4.14b. Agent 1's sequence is easier than Agent 0's, with more global scene views and less motion blur. The aim of this dataset is to evaluate the robustness of the SLAM in particularly difficult visual conditions leading to repeated tracking failure, and to show the interest of collaborative mapping to improve individual localization when one of the agents has difficult visual conditions.

Finally, the *Sea diving* dataset extends the evaluation to a real underwater field scenario. It is recorded by divers with a GoPro in the Mediterranean Sea, at shallow depth, in clear water. Suspended particles, fish, and a second diver appear in some frames. Different agent recordings are made moving slowly around a rock with very closed trajectories (Figure 4.14c).

Generating a ground truth is particularly challenging in underwater environments. Similarly to (Ferrera et al., 2019), a comparative baseline for SLAM evaluation is computed for all of our new underwater datasets using the Structure-from-Motion software Colmap (Schönberger et al., 2016), which performs an offline reconstruction of the scene with exhaustive matching between images, and outputs an accurate estimation of the camera trajectory. This output is not a ground truth, but is still a fair reference for online SLAM evaluation. The scaling factor is retrieved from the known dimensions of the submerged objects. The main characteristics of the three datasets are summarized in Table 4.4. Trajectories of the agents are represented in Figure 4.14, with respect to the surrounding objects.

TABLE 4.4: Underwater multi-agent datasets description. All of them are recorded with an RGB camera and between 1 and 5 m depth, without embedded lights.

| | Camera | Duration | Description |
|---|---|---|---|
| *Ifremer tank* | 480×640 pixels<br>10 Hz | 100 s | Textured fake reef in pool. Slow motion. |
| *Cephismer tank* | 480×640 pixels<br>5 Hz | 75 s | Submarine spare parts, in pool. Some poorly textured areas (walls, floor). Fast motion and motion blur. Agent 1's sequence easier than Agent 0's. |
| *Sea diving* | 380×640 pixels<br>8 Hz | 100 s | Around a rock, at sea, at shallow depth, in clear water. Presence of suspended particles, fishes and a fellow diver. |



(A) *Ifremer tank* (B) *Cephismer tank* (C) *Sea diving*

FIGURE 4.14: Agent trajectories in the underwater sequences. The black points are a sub-sample of the SfM reconstruction and give an idea of the position of the surrounding objects with respect to the agents' trajectories.

### 4.3.3 Multi-agent VSLAM evaluation methodology

In order to evaluate MAM$^3$SLAM and compare its performances with respect to state-of-the-art multi-agent and multimap VSLAM, a choice of datasets and evaluation criteria is necessary. Section 4.3.3.1 presents the competing multi-agent works evaluated, Section 4.3.3.2 indicates the evaluation datasets used and Section 4.3.3.3 introduces evaluation criteria.

#### 4.3.3.1 Competing works

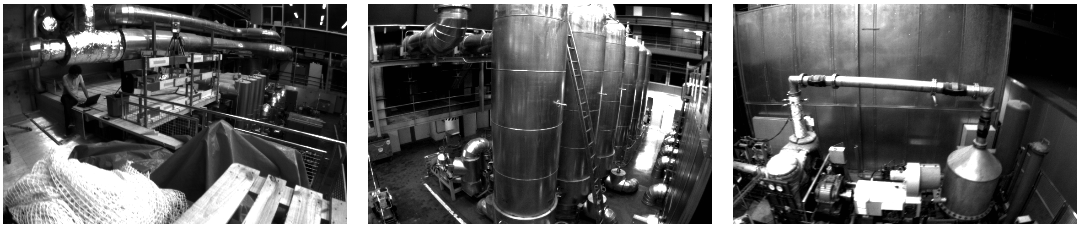
In accordance with the state-of-the-art in multi-agent VSLAM presented in Section 2.5, we select ORB-SLAMM (Daoud et al., 2018) and CCM-SLAM (Schmuck et al., 2019) for conducting a comparative evaluation with MAM$^3$SLAM While ORB-SLAMM is the only state-of-the-art both multi-agent and multimap VSLAM to the best of our knowledge, CCM-SLAM is the most complete state-of-the-art real-time multi-agent VSLAM framework at the time of writing. Workflow representations of these two works is available in Figures 2.39 and 2.40 from Section 2.5, and can be compared to Figure 4.10. A comparison between competing works is provided in Table 4.5.

TABLE 4.5: Competing algorithms. Unless otherwise stated, tracking (T), local mapping (LM), loop closing (LC), place recognition (PR) and GBA operations are the same as the one implemented in ORB-SLAM. Map merging is abbreviated into MM.

| | **Agents** | **Server** | **Map merging** | **Recovery** |
|---|---|---|---|---|
| MAM$^3$SLAM | Send frames to the server | 1 T thread per agent<br>1 LM thread per agent<br>1 LC & MM thread<br>(1 LC GBA thread) | Similar to ORB-SLAM3's MM | Multi-map |
| ORB-SLAMM (Daoud et al., 2018) | Send frames to the server | 1 T thread per agent<br>1 LM thread per agent<br>1 LC thread per agent<br>(1 LC GBA thread per agent)<br>1 inter-map PR thread | Close to ORB-SLAM's LC but with less geometric consistency checks | Multi-map |
| CCM-SLAM (Schmuck et al., 2019) | 1 T thread<br>1 LM thread<br>+ send new KF to the server | 1 inter-map PR thread<br>(1 MM GBA thread)<br>+ send local KF to the agents | Similar to ORB-SLAM's LC + GBA | ∅ |

#### 4.3.3.2 Datasets

In order to evaluate MAM$^3$SLAM on a standard VSLAM benchmark, a first evaluation is conducted using the *Machine Hall (MH)* sequences from the *EuRoC* MAV Dataset (Burri et al., 2016). In a second part, the underwater evaluation is performed on the three new underwater datasets introduced in Section 4.3.2. The *EuRoC MH* sequences are captured in the same industrial environment with the same lighting conditions and are provided with a ground truth position from a Leica Total Station. Similarly to (Schmuck et al., 2019), sequences `MH_02` and `MH_03` were used to simulate two agents. The agents start from a close position, but do not overlap much after takeoff, each agent exploring a different part of the hall. An overview of this dataset is given in Figure 4.15, as well as a brief summary in Table 4.6 and the corresponding trajectories per agent in Figure 4.16.



(A) (B) (C)

FIGURE 4.15: Overview of the *EuRoC MH* dataset

TABLE 4.6: *EuRoC MH* dataset description.

| | **Camera** | **Duration** | **Description** |
|---|---|---|---|
| *EuRoC MH*<br>(Burri et al., 2016) | Grayscale<br>480×752 pixels<br>20 Hz | 135 s | Flying drone in an industrial hall. |

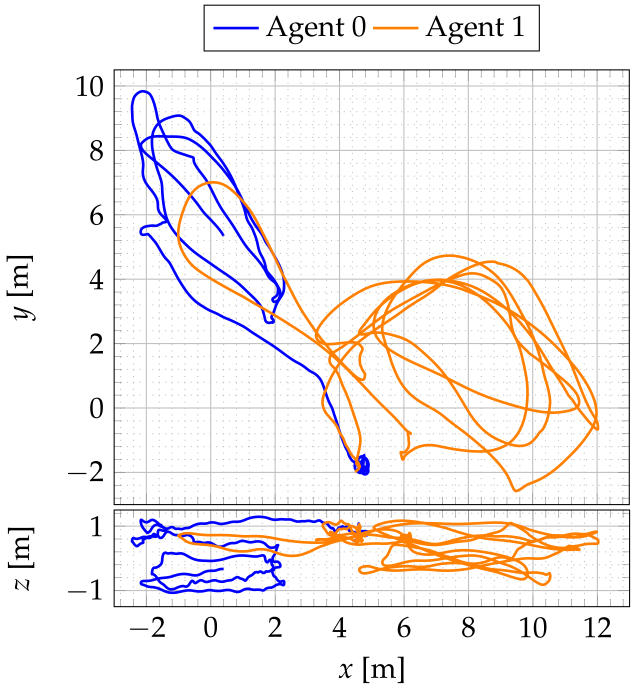


FIGURE 4.16: Agent trajectories on the *EuRoC MH* dataset

#### 4.3.3.3 Evaluation criteria

Two evaluations are conducted: localization and mapping performances, and real-time performances. In related multi-agent VSLAM works (Daoud et al., 2018; Schmuck et al., 2019), localization and mapping performances are only evaluated by the RMS Absolute Position Error (APE) on each agent's trajectory. However, multi-agent SLAM also aims at estimating inter-agent relative poses and in mapping their environment collaboratively. This is why additional comparison metrics are introduced here, in order to evaluate both individual agent localization, inter-agent relative localization and collaborative mapping.

*Individual localization* is evaluated by the RMS APE, RMS Relative Position Error (RPE) between two consecutive frames and the percentage of localization failure with respect to the number of frames.

*Relative localization* is characterized by the percentage of the sequence for which the agents are localized on the same map, and by the RMS Absolute Relative Position Error (ARPE) between the two agents. If the system fails to merge all maps into a single one, these metrics are given both as a statistic over all maps and for the map with the largest number of KFs, denoted as the *main map*.

*Collaborative mapping* is first evaluated by the number $N_{maps}$ of unmerged maps at the end of the sequence. Additional metrics are computed on the main map. Agent contribution and map size are characterized by the number of KFs created by each agent. The map's spatial cover $L_{map}$ is computed as the sum of the edges of the minimum spanning tree among the positions of the KF in the map, and evaluates the size of the mapped area. $L_{map}$ is computed using the ground truth pose of the KF to avoid introducing a bias due to poor estimation of the KF pose. The RMS APE on the pose of the KFs of the main map is also computed.

Given an error metric $e$ on a set of $N$ poses $\mathbf{T}_j \in SE(3)$ defined on $n$ disconnected maps denoted $\mathcal{M}_i, i \in \{0...n-1\}$, the global RMS associated to $e$ is defined by:

$$e_{RMS} = \sqrt{\frac{\sum_{i=1}^{n} \sum_{\mathbf{T}_j \in \mathcal{M}_i} e(\mathbf{T}_j)^2}{N}} \tag{4.1}$$

where $\mathbf{T}_j \in \mathcal{M}_i$ denotes that pose $\mathbf{T}_j$ is defined in map $\mathcal{M}_i$

In order to compare the real time performances of the algorithms evaluated, the durations of the tracking ($T$), local mapping ($LM$) and place recognition ($PR$) operations are recorded. Place recognition includes inter and intra map loop detection, loop fusion and map merging. Let $t_{op}$, $op \in \{T, LM, PR\}$ denote the total duration of the $op$ operation summed for all iterations for all agents on a sequence. Let $N_i$, $i \in \mathbb{N}$ define the total number of frames of Agent $i$ during the same sequence. Since the different datasets evaluated have different durations and framerates, the run time of each operation $op$ is characterized by the quantity:

$$\overline{t_{op}} = \frac{t_{op}}{\sum_i N_i} \tag{4.2}$$

Because the tracking operation is triggered by a new incoming frame, $\overline{t_T}$ is also the average duration of a tracking operation.

### 4.3.4 Evaluation results

A quantitative evaluation of MAM$^3$SLAM is performed on two-agent scenarios on the four datasets described in Section 4.3.3.2, including one aerial dataset and three underwater datasets. MAM$^3$SLAM is compared to multi-agent ORB-SLAMM (Daoud et al., 2018) and to CCM-SLAM (Schmuck et al., 2019). In addition, ORB-SLAM3 (Campos et al., 2021) is used as a reference for individual localization performance, as it represents the current state-of-the-art for monocular VSLAM. Evaluations are carried out in real-time on a computer with an Intel i7-10610U CPU @ 1.80GHz × 8, 16 GB RAM, running Ubuntu 18.04 and ROS Melodic.

#### 4.3.4.1 Localization and mapping evaluation

A localization and mapping performance evaluation is performed according to the metrics described in Section 4.3.3.3. Given the non-deterministic behavior of multithreaded applications, all localization and mapping evaluation metrics are given as their median values among 5 runs. The results are reported in Tables 4.7, 4.8, 4.9 and 4.10. If there are more than one map in the system, the RMS APE and RPE on the main map are indicated in parentheses in the table. Since ORB-SLAMM does not completely fuse the maps in the system but only computes an alignment, the total number of maps in the system is shown in parentheses in the table. Estimated and reference trajectories are aligned using the $Sim(3)$ Umeyama alignment of the EVO library (Grupp, 2017) in order to compute the error metrics. In Tables 4.7, 4.8, 4.9 and 4.10, the best result for each evaluation criterion among the multi-agent SLAM works is shown in bold. If ORB-SLAM3 (Campos et al., 2021) gives a better result than the multi-agent approaches, ORB-SLAM3's result is also indicated in bold.

Figure 4.17 shows the trajectories estimated by MAM$^3$SLAM on one of the five runs, for the four datasets. Estimated and reference trajectories overlap particularly well for the *EuRoC MH*, *Ifremer tank* and *Sea diving* datasets.

TABLE 4.7: Results on the *EuRoC MH* dataset

| | | **MAM$^3$SLAM** | **ORB-SLAMM** | **CCM-SLAM** | ***ORB-SLAM3*** |
|---|---|---|---|---|---|
| #0 | RMSAPE (m) | **0.021** | 1.237 | 0.055 | ***0.016*** |
| | RMSRPE (m) | 0.015 | 0.147 | **0.008** | ***0.005*** |
| | failure rate (%) | 0.72 | **0.00** | **0.00** | ***0.00*** |
| #1 | RMSAPE (m) | **0.034 (0.033)** | 2.624 | 0.152 | ***0.027*** |
| | RMSRPE (m) | **0.011 (0.011)** | 1.436 | 0.135 | ***0.010*** |
| | failure rate (%) | 1.55 | 1.48 | **0.00** | ***0.00*** |
| % same map | | 99.38 | **100.00** | **100.00** | ∅ |
| RMSARPE (m) | | **0.025** | 2.841 | 0.135 | ∅ |
| $N_{maps}$ | | 2 | **1 (4)** | 1 | ∅ |
| KF per agent (#0,#1) | | 243, 249 | 306, 479 | 281, 360 | ∅ |
| $L_{map}$ (m) | | 135.292 | **150.224** | 145.347 | ∅ |
| KF RMSAPE (m) | | **0.031** | 2.162 | 0.159 | ∅ |

CCM-SLAM fails to initialize on the *Ifremer* and *Cephismer tank* datasets, denoting a lack of robustness.

MAM$^3$SLAM demonstrates the best individual localization performances among the multi-agent approaches evaluated, with the lowest RMS APE and RPE in all sequences, reaching centimetric accuracy on the *EuRoC MH* and *Ifremer tank* datasets. CCM-SLAM's performances are close to MAM$^3$SLAMs but still with a lower accuracy, showing up to ten times higher RMS APE and RPE on *EuRoC MH*'s Agent 1 and *Sea diving*'s both agents. ORB-SLAMM shows particularly poor individual localization performances, with RMS APE and RPE exceeding one meter in *EuRoC MH*

TABLE 4.8: Results on the *Ifremer tank* dataset

| | | **MAM$^3$SLAM** | **ORB-SLAMM** | **CCM-SLAM** | ***ORB-SLAM3*** |
|---|---|---|---|---|---|
| #0 | RMSAPE (m) | **0.026** | 0.489 | ∅ | ***0.012*** |
| | RMSRPE (m) | **0.026** | 0.461 | ∅ | *0.038* |
| | failure rate (%) | **0.00** | **0.00** | 100.00 | ***0.00*** |
| #1 | RMSAPE (m) | **0.037** | 0.208 | ∅ | ***0.011*** |
| | RMSRPE (m) | **0.038** | 0.213 | ∅ | ***0.030*** |
| | failure rate (%) | **0.00** | 3.16 | 100.00 | ***0.00*** |
| % same map | | **100.00** | **100.00** | ∅ | ∅ |
| RMSARPE (m) | | **0.025** | 2.841 | ∅ | ∅ |
| $N_{maps}$ | | **1** | **1 (5)** | ∅ | ∅ |
| KF per agent (#0,#1) | | 101, 128 | 89, 125 | ∅ | ∅ |
| $L_{map}$ (m) | | **32.086** | 29.366 | ∅ | ∅ |
| KF RMSAPE (m) | | **0.012** | 0.509 | ∅ | ∅ |

TABLE 4.9: Results on the *Cephismer tank* dataset

| | | **MAM$^3$SLAM** | **ORB-SLAMM** | **CCM-SLAM** | ***ORB-SLAM3*** |
|---|---|---|---|---|---|
| #0 | RMSAPE (m) | **0.125 (0.142)** | 0.217 (0.327) | ∅ | *0.204 (0.241)* |
| | RMSRPE (m) | 0.139 **(0.152)** | **0.127** (0.158) | ∅ | *0.143 (0.159)* |
| | failure rate (%) | **27.78** | 33.33 | 100.00 | ***24.18*** |
| #1 | RMSAPE (m) | **0.115 (0.026)** | 0.286 (0.174) | ∅ | ***0.074 (0.022)*** |
| | RMSRPE (m) | **0.043 (0.021)** | 0.099 (0.066) | ∅ | *0.050 (0.025)* |
| | failure rate (%) | 23.08 | **18.85** | 100.00 | *26.28* |
| % same map | | **13.92** | 0.00 | ∅ | ∅ |
| RMSARPE (m) | | **0.204** | ∅ | ∅ | ∅ |
| $N_{maps}$ | | **4** | 6 (6) | ∅ | ∅ |
| KF per agent (#0,#1) | | 74, 30 | 41, 0 | ∅ | ∅ |
| $L_{map}$ (m) | | **8.412** | 2.855 | ∅ | ∅ |
| KF RMSAPE (m) | | **0.107** | 0.327 | ∅ | ∅ |

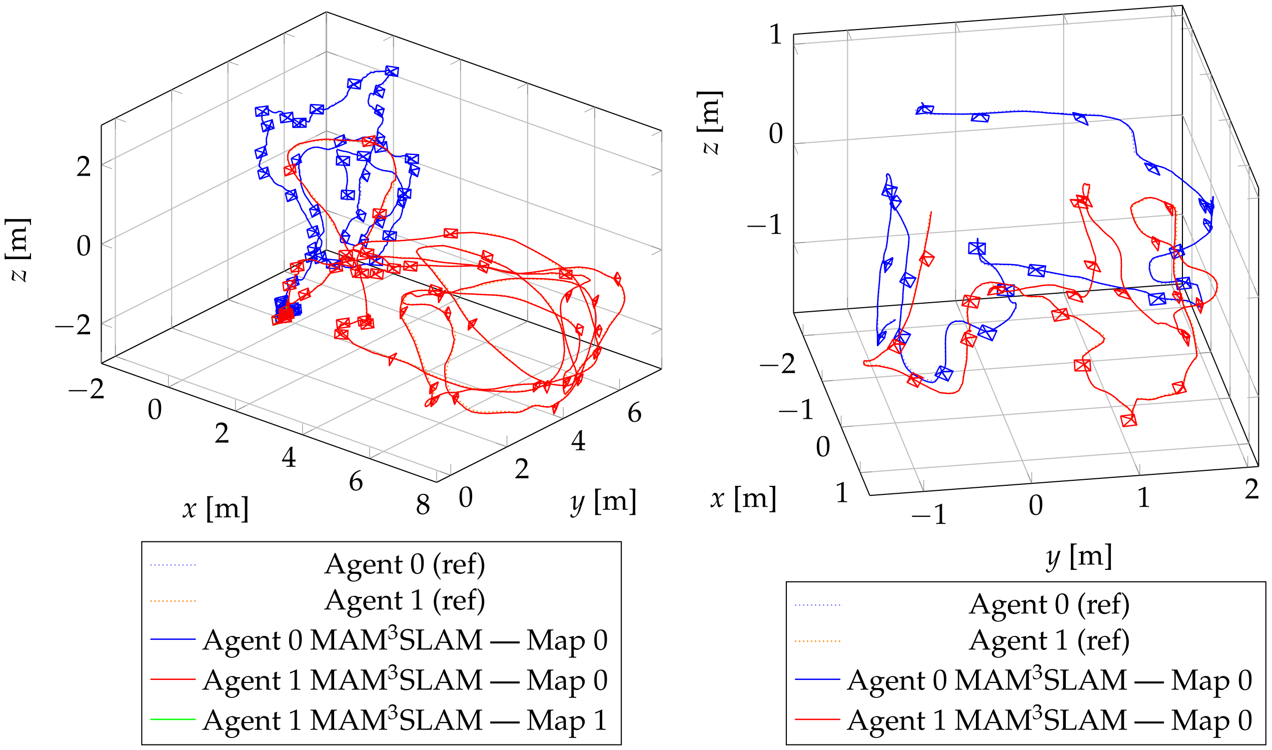
and RMS APE 50 times higher than all other approaches on *Sea diving*'s Agent 1. This is mainly caused by bad inter-map transformation estimation caused by the lack of robustness of their inter-map place recognition and map merging algorithm compared to the other approaches. An example of ORB-SLAMM map misalignment is provided in Figure 4.18, were trajectories estimated by ORB-SLAMM are aligned with respect to the reference trajectories using Umeyama alignment. In Figure 4.18a, trajectories estimated on different maps are aligned together using the inter-map transformation estimated by ORB-SLAMM, and the resulting trajectory is aligned with respect to the reference ones. In Figure 4.18b, trajectories estimated on different maps are aligned individually with respect to the reference, without using the inter-map transformation estimate from ORB-SLAMM. It can be observed that ORB-SLAMM produces inconsistent inter-map alignments, which deteriorate significantly the accuracy of the global trajectory. The same phenomenon happens on all datasets.

TABLE 4.10: Results on the *Sea diving* dataset

| | | **MAM$^3$SLAM** | **ORB-SLAMM** | **CCM-SLAM** | ***ORB-SLAM3*** |
|---|---|---|---|---|---|
| #0 | RMSAPE (m) | **0.032** | 0.582 (0.485) | 0.037 | ***0.026*** |
| | RMSRPE (m) | **0.034** | 0.264 (0.135) | 0.151 | ***0.033*** |
| | failure rate (%) | 5.41 | 0.75 | **0.00** | ***0.00*** |
| #1 | RMSAPE (m) | 0.031 | 1.203 (1.428) | **0.028** | ***0.021*** |
| | RMSRPE (m) | **0.041** | 0.112 (0.125) | 0.144 | ***0.040*** |
| | failure rate (%) | **0.00** | 1.88 | **0.00** | ***0.00*** |
| | % same map | **100.00** | 6.00 | **100.00** | ∅ |
| | RMSARPE (m) | **0.033** | 2.088 | 0.050 | ∅ |
| | $N_{maps}$ | **1** | 5 (6) | **1** | ∅ |
| | KF per agent (#0,#1) | 269, 285 | 21, 187 | 199, 202 | ∅ |
| | $L_{map}$ (m) | **39.556** | 16.235 | 36.053 | ∅ |
| | KF RMSAPE (m) | 0.063 | 0.433 | 0.049 | ∅ |

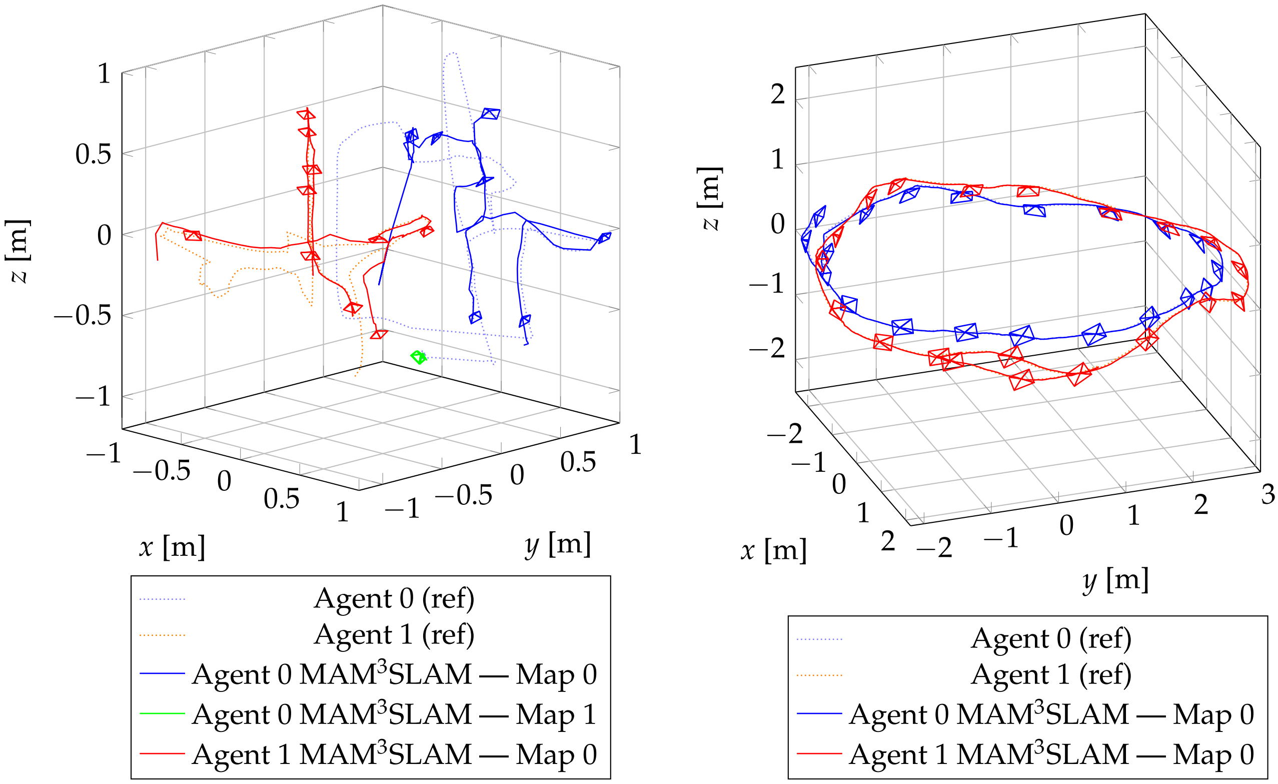
One can notice that even if MAM$^3$SLAM and CCM-SLAM show similar performances, ORB-SLAM3 produces the most accurate individual localization on the *EuRoC MH*, *Ifremer tank* and *Sea diving* datasets. This can be explained by the computing resources available since more parallel computations are required by multi-agent approaches. This can also explain the non-zero failure rates observed for multi-agent approaches on these three datasets. However, MAM$^3$SLAM outperforms ORB-SLAM3 in individual localization on the *Cephismer tank*, highlighting the interest of the collaborative map construction and map sharing for particularly difficult sequences. In this dataset, Agent 1's sequence is easier than Agent 0's. As a result, ORB-SLAM3 outputs 2 times higher RMS APE and RPE on Agent 0 than on Agent 1. However, in MAM$^3$SLAM Agent 0's localization relies on more complete and reliable mapping data through the collaborative scene mapping, resulting in agent performance smoothing, reaching a similar accuracy of about 10 cm for both agents.

MAM$^3$SLAM also outperforms the other evaluated methods on inter-agent relative localization. All methods localize the two agents on the same map almost 100% of the *EuRoC MH* and *Ifremer tank* datasets, but MAM$^3$SLAM outperforms the other approaches on relative pose estimation with centimetric precision. It is also the only SLAM approach capable of localizing the two agents on the same map for some periods of time on the *Cephismer tank* dataset, with a fair 20 cm accuracy. Finally, on the *Sea diving* dataset, MAM$^3$SLAM and CCM-SLAM localize the agents in the same map 100 % of the sequence and with a similar relative localization accuracy, whereas, ORB-SLAMM fails to fuse all maps of the system, resulting in a very small sequence percentage for which a relative localization can be computed.



(A) *EuRoC MH*

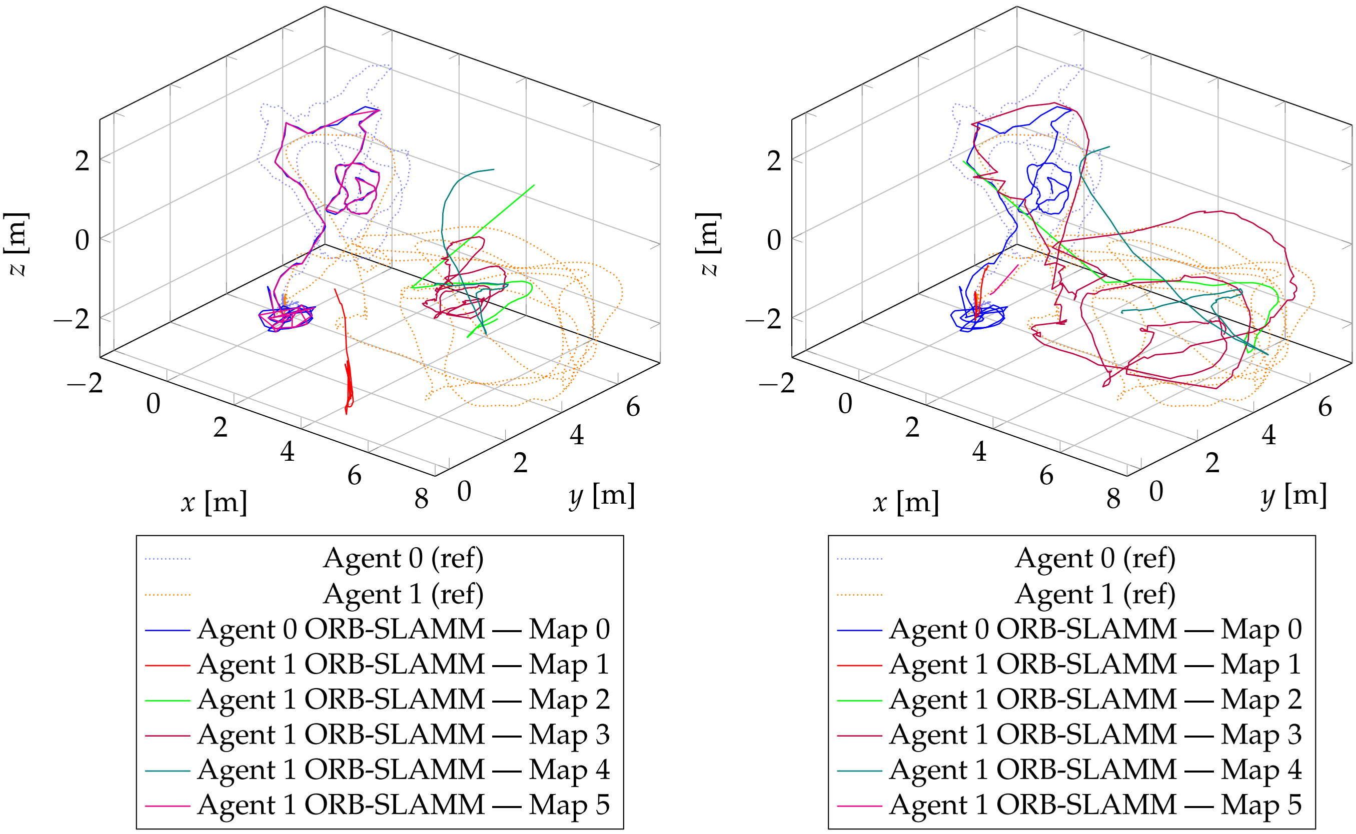
(B) *Ifremer tank*



(C) *Cephismer tank*

(D) *Sea diving*

FIGURE 4.17: Trajectories estimated by MAM$^3$SLAM (solid) and reference trajectories (dashed) on the test datasets, after Umeyama alignment, showing estimated camera poses for every 5s.



(A) Using ORB-SLAMM's map alignment

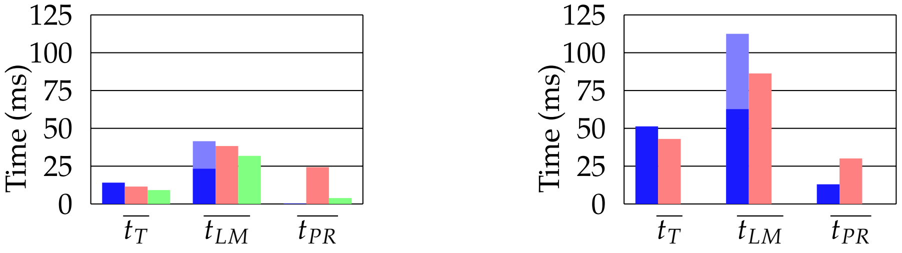
(B) Without using ORB-SLAMM's map alignment

FIGURE 4.18: Trajectories estimated by ORB-SLAMM on the *EuRoC MH* dataset (plain) and ground truth trajectories (dashed). Trajectories are estimated over 5 maps, which are aligned by the system. Parts of the trajectory from different maps are represented in a different color.

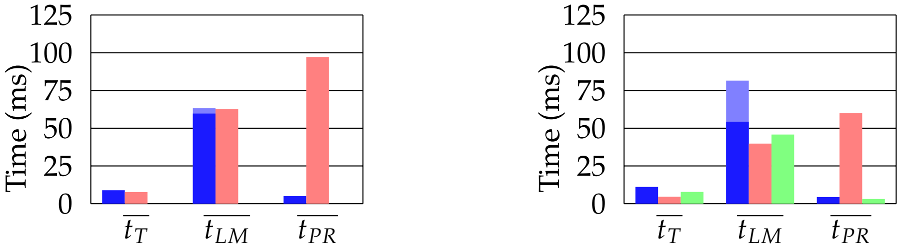
Considering collaborative mapping, ORB-SLAMM produces inaccurate map alignments and sometimes fails to detect map overlap, as with the *Cephismer tank* and the *Sea diving* datasets. Map merging is far more robust in MAM$^3$SLAM and CCM-SLAM, which output a common, single map in most sequences. It is, however, worth noticing that CCM-SLAM's agents cannot initialize a new map, limiting to two the maximum number of maps in the system for two agents. One can see that MAM$^3$SLAM produces two maps on the *EuRoC MH* dataset, but localizes the agents on the same map during almost all the sequence. This is explained because one of the agents fails to track the scene soon after initialization and relocalizes in the map initialized by the other agent, which is incremented collaboratively along the sequence. The second map is thus very small, with a median size of 3 KF. Finally, in the general case, MAM$^3$SLAM produces a more accurate main map than the other approaches for an equivalent number of KF per agent and map size $L_{map}$.

### 4.3.4.2 Computing time evaluation

A computing time evaluation is performed as described in Section 4.3.3.3. Figure 4.19 compares the quantities $\overline{t_T}$, $\overline{t_{LM}}$ and $\overline{t_{PR}}$ obtained in processing each one



(A) *EuRoC MH* (B) *Ifremer tank*



(C) *Cephismer tank* (D) *Sea diving*

🟦 MAM$^3$LAM 🟦 Mutex wait in MAM$^3$LAM's LM
🟥 ORB-SLAMM 🟩 CCM-SLAM

FIGURE 4.19: Computing time indicators $\overline{t_T}$, $\overline{t_{LM}}$ and $\overline{t_{PR}}$

of the test datasets. As explained in Section 4.3.3.3, these quantities characterize the total computational time dedicated to tracking ($T$), local mapping ($LM$) and place recognition ($PR$) operations averaged per frame. The component of $\overline{t_{LM}}$ which corresponds to the mutex wait is also indicated.

One can see that tracking durations are quite similar from an algorithm to another. For the *EuRoC MH*, *Cephismer tank* and *Sea diving* datasets, tracking duration represent about 10 ms per frame per agent. Tracking duration per frame is much higher for the *Ifremer tank* dataset, what can be explained by the high number of ORB features tracked (5,000 ORB features per frame are extracted, *vs* 1000 for *EuRoC MH* and *Sea diving*). 5000 ORB features per frame are extracted for the *Cephismer tank* dataset too, but most of them are not matched correctly and thus unused for pose computation, hence a lower computational duration.

LM total computational duration is higher for MAM$^3$SLAM. One can see that a significant proportion of this duration corresponds to wait at mutex lock. In the evaluated configurations, the maximum duration of local mapping operations divided by the number of incoming frames is of about 110 ms. This indicates that, in this case, the system is able to create a new KF every 9 s, what is still a satisfying ratio. However, this maximum KF insertion rate will decrease with the a higher number of agents, since more agents will increase the total mutex wait duration. Our system is thus limited to a small number of agents. One can notice that the total duration of mutex wait in local mapping is very low for the *Cephismer tank* dataset. This is due to the agents being located on the same map simultaneously more rarely and the low framerate reducing the probability of a simultaneous KF creation intent by the two agents.

Lastly, PR total computational duration is significantly higher for ORB-SLAMM than for CCM-SLAM and MAM$^3$SLAM. Indeed, ORB-SLAMM's inter map place recognition consists in a loop over all KF from all maps which is repeated continuously, while CCM-SLAM and MAM$^3$SLAM only query the KF database at new KF creation, what prevents unnecessary queries if the maps and KF are idle.

### 4.3.5 Conclusions

Section 4.3 introduced MAM$^3$SLAM, a new fully centralized multi-agent and multimap monocular VSLAM framework based on ORB-SLAM Atlas (Elvira et al., 2019) and ORB-SLAM3 (Campos et al., 2021), as a solution for underwater multi-ROV localization. Three new underwater two-agent datasets including two in a pool and one in the sea were acquired, and several multi-agent evaluation criteria were introduced in order to realize a comparative evaluation of MAM$^3$SLAM with respect to the state-of-the-art multi-agent VSLAM CCM-SLAM (Schmuck et al., 2019) and ORB-SLAMM (Daoud et al., 2018).

Four two-agent scenarios were evaluated, including a standard aerial dataset and our three new underwater datasets.

A run time analysis demonstrated that MAM$^3$SLAMs place recognition computation time is significantly lower than competing works, and its tracking computational time is of the same order of magnitude. This analysis also shows that MAM$^3$SLAMs local mapping takes significantly more time than the one of competing approaches, because of the mutex wait imposed to prevent concurrent map updates from different agents when they localize with respect to the same map, and some low level software engineering would be required to extend MAM$^3$SLAM to more than two agents in future works. This extension would be a prerequisite for using MAM$^3$SLAM in a real-life underwater robot chain configuration.

Even though the current implementation did not focus on handling this scalability problem and left it for future works, our two-agent MAM$^3$SLAM proved to outperform all other evaluated approaches, namely CCM-SLAM (Schmuck et al., 2019) and ORB-SLAMM (Daoud et al., 2018), in terms of individual localization, relative localization and mapping accuracy, and demonstrates good robustness to poor visual conditions. MAM$^3$SLAM therefore proves to be a reliable multi-agent VSLAM approach for multi-ROV localization, reaching a centimetric accuracy on two out of three underwater evaluation datasets.

In light with these results, MAM$^3$SLAM seems a promising solution for multiple ROV localization, for instance in an underwater robot chains. Some low level software developments are nonetheless required to extend this approach to a higher number of agents, while keeping the overall high level principle. In addition, future works may focus on improving the robustness of the system by fusing multiple inputs in order to increase the robustness to underwater visual conditions and estimate the scale of the environment. In particular, in the specific context of a robot chain, inter-agent pose constraints might be used at different levels in the SLAM pipeline, including scale estimation, inter-map place recognition and pose estimation optimization. The system may also be extended to stereo camera input and fuse inertial and eventually pressure measurements.

## 4.4 Conclusion and perspectives

This chapter focuses on the reliability of monocular VSLAM algorithms for locating a chain of underwater ROV with respect to their environment. The problem of underwater monocular VSLAM is initially addressed for a single vehicle. In line with previous works which highlight the interest of DSO (Engel et al., 2018) and ORB-SLAM (Mur-Artal et al., 2015) for underwater scenarios but also their limitations in performing accurate long-term data associations, a comparative evaluation of promising recent works based on these algorithms is performed. Namely, LDSO (Gao et al., 2018) and DSM (Zubizarreta et al., 2020) extend DSO with place recognition capabilities, and ORB-SLAM3 (Campos et al., 2021) and ORB-SLAM (Huang et al., 2020) extend ORB-SLAM with multi-map SLAM recovery strategies. A qualitative evaluation based on eight underwater datasets featuring different visual conditions, including two new datasets, demonstrate that ORB-SLAM3 outperforms other approaches and is robust to underwater visual conditions, and that even though its place recognition algorithm may be improved ORB-SLAM3 can already be a solution for underwater localization in scenarios avoiding fast motions and featuring textured areas.

In accordance with these observations, we introduce MAM$^3$SLAM a monocular, multi-map and multi-agent, centralized VSLAM algorithm based on ORB-SLAM3, as a solution for multi-ROV localization. MAM$^3$SLAM is evaluated and compared to state-of-the-art works ORB-SLAMM (Daoud et al., 2018) and CCM-SLAM (Schmuck et al., 2019) on two-agent scenarios, on one airborne dataset and three new underwater datasets including two in pool and one at sea. Individual localization, relative localization and mapping accuracy are evaluated, and a runtime evaluation is conducted. Results show that MAM$^3$SLAM is able to run in real time on the test scenarios, even though it is limited to a small number of agents, and that it significantly outperforms the competing works in terms of robustness and accuracy, reaching a centimetric accuracy on two out of three underwater evaluation datasets.

This chapter shows that a monocular VSLAM algorithm, and in particular MAM$^3$SLAM can be a solution for locating a chain of a small number of underwater ROV with respect to their environment. Compared to the cable-based localization strategies investigated in Chapter 3, the assets of this VSLAM approach are the following:

- its accuracy, with individual and relative position errors of only a few centimeters when avoiding fast motions, in scenes featuring textured areas. Cable based localization error is at least ten times higher and will increase significantly with the number of connected vehicles.
- an online mapping of the environment, including obstacles.

However, our cable-based localization strategy has assets too compared to VSLAM:

- our state estimation of the cable is always available, even when navigating in plain water, where the sea bed cannot be observed by the embedded cameras
- it is far more robust than VSLAM. While VSLAM can fail to track environment points and never recover from this failure, the only way cable-based localization may completely fail is if the cable is significantly distorted from the model. While this may happen to the catenary model in the presence of currents, the straight line model is robust to such disturbances.
- monocular VSLAM returns a pose and a map with an unknown scale factor, while the scale is given by cable length in cable-based localization algorithms

- cable-based localization can give the location of the ROV with respect to a surface point with known GPS coordinates, leading to an absolute localization of the underwater vehicle
- in MAM$^3$SLAM, agents are not located in the same map until the place recognition module detects map overlap, while a cable based approach directly provides an inter-agent relative pose

As a result, VSLAM and cable-based localization strategies are very complementary, and an interesting future work would be to combine them in a common localization framework. A first option would be to use cable-based localization as a backup localization solution and be used temporarily instead of VSLAM if the VSLAM fails. This option although involves to align the scales of the two localization methods. A second option may be to integrate cable shape knowledge more tightly in the VSLAM framework in order to not only compute the scale of the scene, but also increase SLAM robustness. Cable shape knowledge may be integrated in pose estimation in order to make it more robust to challenging visual conditions, or be exploited for recovering from tracking failure. Relative localization information from cable-based localization can also be used to fuse information from multiple agents efficiently in a common reference frame.

The work presented in Chapter 4 led to the following publication:

**Conference paper:**

- Qualitative evaluation of state-of-the-art DSO and ORB-SLAM-based monocular visual SLAM algorithms for underwater applications.
  J. Drupt, C. Dune, A. I. Comport and V. Hugel,
  OCEANS 2023, Limerick, Ireland, 2023, pp. 1-7.

In addition, a journal paper on MAM$^3$SLAM for multi-ROV underwater localization and mapping is currently being written, in order to be submitted to Ocean Engineering.

## Chapter 5

# Conclusion and perspectives

This thesis focused on underwater robot chains as a solution for managing the tethers of ROVs. In this system, intermediary cable management robots are placed along the cable in order to control its three-dimensional shape and limit the mechanical impact of the cable on the ROV, including the risk of entanglements and energy-intensive drag forces. All the robots in the chain are then connected together by the communication cable, which can also provide them energy supply.

The current thesis investigated specifically the problem of localizing an underwater robot chain with embedded sensors measurements. Robot chain localization includes localization of the robots and of the cable which links them with respect to their environment. While underwater robot localization with respect to its environment can be estimated from the robot's embedded sensors, presence of an umbilical can be taken as an advantage to localize the robot at its end point, as it can be deduced from cable 3D state knowledge. Two categories of localization approaches were thus studied, namely *tether-based state estimation* and *VSLAM-based localization*.

## 5.1 Tether-based state estimation

The tether has been investigated as a localization system for the ROVs it connects. If the three-dimensional shape of the tether linking a pair of vehicles can be estimated, the relative location of the two vehicles can be deduced from the position of the tether's end points. We proposed to use a simple cable model parameterized by cable ends relative depth and one or two local cable tangents. These parameters were measured using IMUs placed along the cable and embedded pressure sensors.

Two categories of cable configuration have been studied, both involving a non-extensible cable. First, the catenary model has been investigated for modeling an underwater, non-extensible, slack, weighting cable. The catenary model is a physical, quasi-static model defined in a vertical plane. It does not require constraining the cable into a specific shape. Its validity for an underwater cable with moving ends was evaluated experimentally, demonstrating that the model well describes heavy enough cables even with mobile ends. A new catenary shape estimation approach was therefore introduced for a negatively buoyant underwater cable, using inertial measurements of one or two cable tangents near the attachment points. An experimental validation demonstrated that, for a cable of length 1.5 m, this method can estimated the sag of the cable within a few centimeters and the orientation of the cable plane within a dozen degrees. In a second time, a cable stretched by a system of mobile buoys and ballast modeled by straight lines was investigated, in line with the work of Christophe Viel (CNRS-LabSTICC, ENSTA Bretagne, France) with whom we collaborated on this work. Cable shape estimation based on local tangent measurements using IMUs was studied for locating the ROV at the cable end point.

Evaluations in pool demonstrated that the accuracy of this approach in cable and ROV localization is of about twenty centimeters, given a cable length of 3 m.

We demonstrated that the shape of the cable of a ROV can be estimated using a cable model and one or two local tangent measurement using IMUs, as well as robot depth knowledge using pressure measurements, and that the cable can be used as a localization system for determining the relative position of the vehicles at its end points. A strong asset of this cable-based localization strategy is that it is available even when navigating in open water, where the seabed cannot be observed by the embedded sensors. A limitation is however a limited accuracy and the propagation of errors along multiple tethered vehicles within a robot chain. In addition, the cable-based system state estimation methods studied in the current work do not provide any exteroceptive information about the surroundings of the system, which are nonetheless crucial for robot chain control and navigation.

## 5.2 VSLAM-based localization

Because the safe navigation and control of an underwater robot chain involves localization with respect to a previously unknown environment, we focused on simultaneous localization and mapping (SLAM) for such systems. While some works address the SLAM problem from a cable-based environment perception (McGarey et al., 2017), such strategies require the cable to enter in contact with obstacles, what may be unwise in term of operational safety for an underwater system. Consequently, we focused on a SLAM strategy using common underwater exteroceptive sensors. Usual underwater exteroceptive sensors include acoustic sensors and cameras, where cameras are lower cost, power and weight, such that even the smallest and most affordable underwater robots embed at least an underwater camera. In addition, cameras do not suffer from interferences nor multiple reflections from non-target objects in cluttered environments, contrary to accoustic sensors. These arguments lead us to focus on monocular VSLAM for underwater.

We first studied underwater monocular VSLAM for a single robot. In accordance with previous studies (Quattrini Li et al., 2017; Joshi et al., 2019; Hidalgo et al., 2018), we focused on ORB-SLAM (Mur-Artal et al., 2015) and DSO (Engel et al., 2018)-based works designed to improve their long-term data association capabilities. Namely, we studied LDSO (Gao et al., 2018) and DSM (Zubizarreta et al., 2020) which extend DSO with place recognition capabilities, and ORB-SLAM3 (Campos et al., 2021) and Dual-SLAM (Huang et al., 2020) which extend ORB-SLAM with multi-map SLAM recovery strategies. The six aforementioned algorithms were evaluated on eight underwater datasets featuring different visual conditions, including two new datasets. Results highlighted the robustness of ORB-SLAM3 to underwater visual conditions and the interest of its multi-map SLAM recovery algorithm.

In line with these observations, we introduced MAM$^3$SLAM a monocular, multi-map and multi-agent, centralized VSLAM algorithm based on ORB-SLAM3, as a solution for multi-ROV localization. MAM$^3$SLAM was evaluated and compared to state-of-the-art works ORB-SLAMM (Daoud et al., 2018) and CCM-SLAM (Schmuck et al., 2019) on three new underwater two-agent datasets, demonstrating a better individual localization, relative localization and mapping accuracy, with an order of magnitude of less than one centimeter in two out of the three underwater evaluation datasets. Although a runtime evaluation of MAM$^3$SLAM showed that some low level software engineering work may be required to increase SLAM scalability, MAM$^3$SLAM proved to be a promising solution for underwater-robust multiple ROV localization, including in the context of a robot chain.

## 5.3 Perspectives

Table 5.1 recaps the assets and drawbacks of cable-based localization and monocular VSLAM for the localization of an underwater robot chain.

TABLE 5.1: Assets and drawbacks of cable-based localization and monocular VSLAM

| | **Assets** | **Drawbacks** |
|---|---|---|
| Cable-based localization | • Always available, even in open water where the sea bed cannot be observed<br>• Far more robust than VSLAM<br>• Known scale factor<br>• Can give the location of the ROV with respect to a surface point with known GPS coordinates<br>• Directly provides an inter-agent relative pose | • Coarse accuracy compared to VSLAM, about 10 to 20 cm<br>• Fails if the cable is significantly distorted from the model |
| Monocular VSLAM | • Individual and relative localization accuracy down to only a few centimeters errors<br>• Online mapping of the environment, including obstacles | • Unavailable in open water<br>• Risk of SLAM failure in case of fast motion or in low textured areas<br>• Unknown scale factor<br>• Requires place recognition to provide an inter-agent relative pose<br>• Does not give cable shape |

It can be observed from this comparison that cable-based localization and monocular VSLAM may me complementary for localizing an underwater robot chain with respect to its environment.

- The two approaches can therefore provide different information about the system's state. Indeed, cable-based localization can give a geo-referenced position while VSLAM cannot, and VSLAM computes an online mapping of the environment, what has not been considered using the cable in order to prevent entanglements (see Section 5.2).

- They are not efficient in the same conditions for robot localization. If the robot moves slowly enough in a textured environment, VSLAM will be available and give a localization estimate far more accurate than cable-based algorithms. Conversely, cable-based approaches can give a quite coarse but robot localization estimate in open water, or in a scenario where the SLAM would have failed.

- Generally speaking, cable-based localization and VSLAM do not have the same causes of failure.

- VSLAM does not give any information about cable location, except for its attachment points on the robots. A knowledge of cable shape is therefore necessary, using either a proprioceptive cable (see Section 2.3.3) or a model derived from measurable physical parameters.

In light of these complementary aspects, future works may investigate a fusion between cable-based localization and monocular VSLAM. A first option would be to use cable-based localization only for the functionalities VSLAM does not implement, including state estimation in open water, geo-referenced positioning, scale estimation, relative localization between the robots before they happen to be located on the same map, and optionally cable shape estimation if a cable shape cannot be deduced directly from robot localization. Cable-based localization may also be a backup solution in case of VSLAM failure. A second option may be to integrate cable-based state estimation and VSLAM more tightly, in a tethered-VSLAM framework. For instance, an initial guess for inter-agent map fusion may be computed from their cable-based relative pose, and cable-based pose estimate may be exploited for recovering from tracking failure. Conversely, VSLAM robot localization may help detecting if the cable does not conform anymore to the expected model, what may, for instance, help detecting entanglements or measure currents. Therefore, many works are still to be done in the continuation of the research axis of the current thesis.

# Appendix A

# Mathematical background and system modeling

## A.1 Rotations

The three-dimensional attitude of a rigid body can be given according to several representations. The most common ones are rotation matrices, Euler angles, vector-axis representation and quaternions. In the following, let us denote $\mathcal{F}_a = (\mathbf{O}_a, \mathbf{x}_a, \mathbf{y}_a, \mathbf{z}_a)$ and $\mathcal{F}_b = (\mathbf{O}_b, \mathbf{x}_b, \mathbf{y}_b, \mathbf{z}_b)$ two three-dimensional frames, and $\mathbf{u}$ a three-dimensional vector. $\forall\, i \in \{a, b\}$, ${}^{i}\mathbf{u}$ denotes the coordinates of vector $\mathbf{u}$ in frame $\mathcal{F}_i$.

### A.1.1 Rotation matrix

A rotation matrix is defined as a matrix whose multiplication with a vector rotates the vector while preserving its length. The special orthogonal group of $3 \times 3$ rotation matrices is denoted $SO(3)$, such that $\forall\, \mathbf{R} \in SO(3)$, $det(\mathbf{R}) = 1$ and $\mathbf{R}$ is invertible, with $\mathbf{R}^{-1} = \mathbf{R}^{T}$.

#### A.1.1.1 Coordinate transformation between two frames

The attitude of a rigid body can be encoded by a rotation matrix. Let us define ${}^{a}\mathbf{R}_b$ the unique rotation matrix which represents the orientation of frame $\mathcal{F}_a$ with respect to frame $\mathcal{F}_b$. The coordinates of vector $\mathbf{u}$ in frame $\mathcal{F}_a$ can be deduced from its coordinates in $\mathcal{F}_b$ by the relation:

$$ {}^{a}\mathbf{u} = {}^{a}\mathbf{R}_b\, {}^{b}\mathbf{u} \tag{A.1} $$

where the columns of ${}^{a}\mathbf{R}_b$ are the coordinates of the unit vectors of frame $\mathcal{F}_b$ in frame $\mathcal{F}_a$, *i.e.*:

$$ {}^{a}\mathbf{R}_b = \begin{bmatrix} {}^{a}\mathbf{x}_b & {}^{a}\mathbf{y}_b & {}^{a}\mathbf{z}_b \end{bmatrix} \tag{A.2} $$

#### A.1.1.2 Elemental rotations

An elemental rotation is a rotation about one single coordinate axis. Elemental rotations by an angle $\alpha$ about unit axes $\mathbf{x}$, $\mathbf{y}$ and $\mathbf{z}$ of a coordinate system are respectively:

$$\mathbf{R}_x(\alpha) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\alpha) & \sin(\alpha) \\ 0 & -\sin(\alpha) & \cos(\alpha) \end{bmatrix} \tag{A.3}$$

$$\mathbf{R}_y(\alpha) = \begin{bmatrix} \cos(\alpha) & 0 & -\sin(\alpha) \\ 0 & 1 & 0 \\ \sin(\alpha) & 0 & \cos(\alpha) \end{bmatrix} \tag{A.4}$$

$$\mathbf{R}_z(\alpha) = \begin{bmatrix} \cos(\alpha) & \sin(\alpha) & 0 \\ -\sin(\alpha) & \cos(\alpha) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{A.5}$$

#### A.1.1.3 Composition

Let us define frame $\mathcal{F}_c = (\mathbf{O}_c, \mathbf{x}_c, \mathbf{y}_c, \mathbf{z}_c)$. The rotation matrix ${}^a\mathbf{R}_c$ between frames $\mathcal{F}_a$ and $\mathcal{F}_c$ can be deduced from the rotation matrices ${}^a\mathbf{R}_b$ and ${}^b\mathbf{R}_c$ respectively between frames $\mathcal{F}_b$ and $\mathcal{F}_a$ and frames $\mathcal{F}_c$ and $\mathcal{F}_b$ by the relation:

$${}^a\mathbf{R}_c =^c \mathbf{R}_b \, {}^b\mathbf{R}_c \tag{A.6}$$

### A.1.2 Euler angles

Euler angles are one of the most common orientation representations. They are based on the decomposition of the rotation into elemental rotations. Let $i, j, k \in \{x, y, z\}^3$ where $i \neq j$ and $j \neq k$. The tuple $(i, j, k)$ then defines an Euler sequence, denoted *ijk*. Any such *ijk* sequence can be defined according to an intrinsic or extrinsic rotations convention, resulting in twelve possible Euler angles conventions. Usually, in an intrinsics or extrinsic *xyz* or *zyx* Euler convention, the angles about the $x$, $y$ and $z$ axes are denoted *roll*, *pitch* and *yaw* respectively.

#### A.1.2.1 Conventions by intrinsic rotations

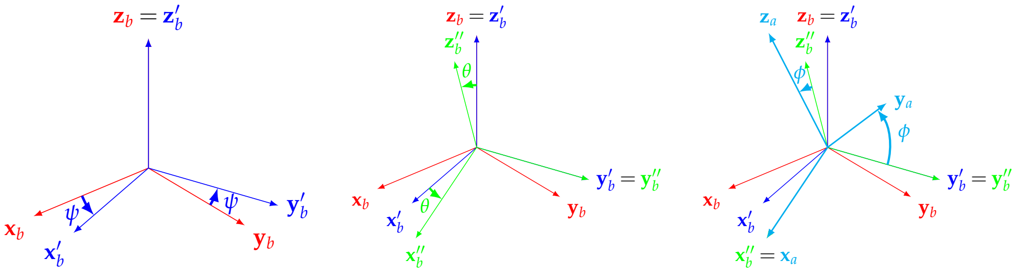
Let us define $\{\phi, \theta, \psi\} \in \mathbb{R}^3$, the set of Euler angles giving the orientation of frame $\mathcal{F}_a$ in frame $\mathcal{F}_b$ in intrinsic *ijk* convention. Rotating $\mathcal{F}_b$ to $\mathcal{F}_a$ can then be decomposed into the following, ordered, operations:

1. Rotate frame $\mathcal{F}_b$ of angle $\psi$ around its $k$-axis. The rotated frame is denoted $\mathcal{F}_b'$.
2. Rotate frame $\mathcal{F}_b'$ of angle $\theta$ around its $j$-axis. The rotated frame is denoted $\mathcal{F}_b''$.
3. Rotate frame $\mathcal{F}_b''$ of angle $\phi$ around its $i$-axis. The rotated frame now has the same orientation as $\mathcal{F}_a$.

An example of such operations in intrinsic *xyz* convention is given in Figure A.1. The decomposition of any three-dimensional rotation matrix ${}^a\mathbf{R}_b \in SO(3)$ in *ijk* convention is given by:

$${}^a\mathbf{R}_c = \mathbf{R}_i(\phi)\, \mathbf{R}_j(\theta)\, \mathbf{R}_k(\psi) \tag{A.7}$$

Angles $\{\phi, \theta, \psi\}$ are called Euler angles in *ijk* convention. The rotation is then represented by a set of three Euler angles in a given decomposition convention.



(A) Step 1 (B) Step 2 (C) Step 3

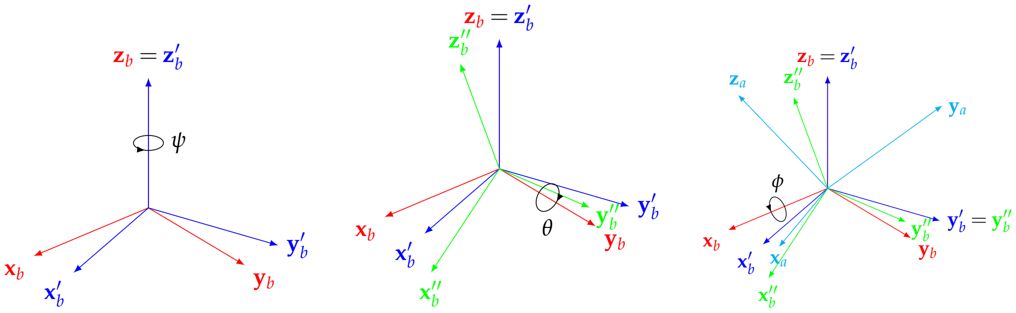
FIGURE A.1: Euler intrinsic $xyz$ rotations from $\mathcal{F}_b$ to $\mathcal{F}_a$

#### A.1.2.2 Conventions by extrinsic rotations

Let us define $\{\phi, \theta, \psi\} \in \mathbb{R}^3$, the set of Euler angles giving the orientation of frame $\mathcal{F}_a$ in frame $\mathcal{F}_b$ in extrinsic $ijk$ convention. Rotating $\mathcal{F}_b$ to $\mathcal{F}_a$ can then be decomposed into the following, ordered, operations:

1. Rotate frame $\mathcal{F}_b$ of angle $\psi$ around its $k$-axis. The rotated frame is denoted $\mathcal{F}'_b$.
2. Rotate frame $\mathcal{F}'_b$ of angle $\theta$ around the $j$-axis of $\mathcal{F}_b$. The rotated frame is denoted $\mathcal{F}''_b$.
3. Rotate frame $\mathcal{F}''_b$ of angle $\phi$ around the $i$-axis of $\mathcal{F}_b$. The rotated frame now has the same orientation as $\mathcal{F}_a$.

An example of such operations in extrinsic $xyz$ convention is given in Figure A.2. Note that equation A.7 is not valid for extrinsic conventions.



(A) Step 1 (B) Step 2 (C) Step 3

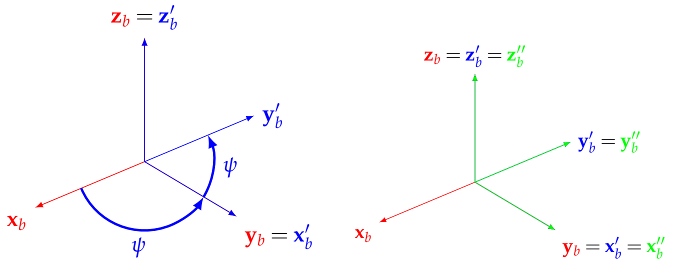
FIGURE A.2: Euler extrinsic $xyz$ rotations from $\mathcal{F}_b$ to $\mathcal{F}_a$

#### A.1.2.3 Discussion

While Euler angles are widely used, they have two main limitations.

The first limitation is practical, since the co-existence of multiple conventions, but also of conflicting notation conventions. For instance, extrinsic conventions are sometimes written in uppercase to differentiate them from intrinsic conventions, but the exact opposite notation convention is also sometimes employed. In addition, the order of the Euler sequence is sometimes inverted from what we presented in this section, with an $ijk$ sequence denoting an opposite order of applying the three steps of Euler rotations. Consequently, Euler angles can be confusing because of the coexistence of several conventions from a work to another.

The second limitation is mathematical, from the definition of Euler angles, and is known as gimbal lock. It can be defined as the loss of one degree of freedom when two of the three rotation axis involved in the Euler angles definition happen to be aligned. An example is given in Figure A.3 for an intrinsic $xyz$ convention. With $\psi = 90^\circ$ and $\theta = 0^\circ$, axis $\mathbf{y}_b''$ gets aligned with $\mathbf{x}_b$.



(A) Step 1

(B) Step 2

FIGURE A.3: Gimbal lock in Euler intrinsics $xyz$ rotations from $\mathcal{F}_b$ to $\mathcal{F}_a$, with $\psi = 90^\circ$ and $\theta = -90^\circ$.

### A.1.3 Axis-angle representation

Alternatively, a rotation can be represented by the combination of a three-dimensional unit vector $\mathbf{u}$ and an angle $\theta$. The corresponding rotation is then a rotation of angle $\theta$ around axis $\mathbf{v}$. This $(\mathbf{u}, \theta)$ representation is called *angle-axis* representation. An interest of this representation is that, like rotation matrices, it is unique.

### A.1.4 Quaternion

Quaternions can be used to describe a three-dimensional rotation by a set of four coordinates which encode a rotation axis and a rotation angle. In terms of mathematics, the quaternion number system extends the complex numbers. Quaternions are usually represented in the form:

$$q = w + xi + yj + zk \tag{A.8}$$

where $\{w, x, y, z\} \in \mathbb{R}^4$ and $i^2 = j^2 = k^2 = ijk = -1$. Quaternions can also be written as four-dimensional vectors, with:

$$\mathbf{q} = \begin{bmatrix} x & y & z & w \end{bmatrix}^T \tag{A.9}$$

Let $\mathbf{u}$ define a three dimensional unit vector. Let ${}^a\mathbf{u} = \begin{bmatrix} {}^a u_x & {}^a u_y & {}^a u_z \end{bmatrix}^T$ be the coordinates of $\mathbf{u}$ in frame $\mathcal{F}_a$. A rotation of angle $\theta$ around axis $\mathbf{u}$ can be represented by the quaternion:

$$q = \cos\left(\frac{\theta}{2}\right) + \sin\left(\frac{\theta}{2}\right)\left({}^a u_x i +^a u_y j +^a u_z k\right) \tag{A.10}$$

Such quaternion is called a *unit quaternion*. The definition of a rotation by a unit quaternion is unique.

### A.1.5 Conclusion

The three-dimensional orientation of a rigid body can be assimilated to a rotation. It can be defined according to several representations, where the most common ones are rotation matrices, Euler angles, axis-angle representation and quaternions. The rotation matrix representation is unique, but involves nine coordinates for defining only three degrees of freedom. Euler angles are widely used, but are confusing because of the coexistence of several conventions from a work to another, and because of the gimbal lock phenomenon. Axis-angle representation and quaternions are both unique, but are not as intuitive as rotation matrices and Euler angles. In this work, we will use mostly rotation matrices.

## A.2 Rigid transformations

The three-dimensional state of a rigid body is given by its orientation and its position with respect to a reference frame. Let us define a frame $\mathcal{F}_b$ associated to a rigid body, and a reference frame $\mathcal{F}_w$. The rigid transformation between frames $\mathcal{F}_b$ and $\mathcal{F}_w$ is the combination of a translation between their origins $\mathbf{P}_b$ and $\mathbf{P}_w$ respectively and by their relative orientation. Let ${}^b\mathbf{P}_w$ denote the coordinates of $\mathbf{P}_w$ in $\mathcal{F}_b$ and let ${}^b\mathbf{R}_w$ be the rotation matrix between the two frames. Finally, let us define a three-dimensional point $\mathbf{X}$. The coordinates ${}^b\mathbf{X}$ of $\mathbf{X}$ in frame $\mathcal{F}_b$ can be deduced from its coordinates in ${}^w\mathbf{X}$ in $\mathcal{F}_w$ by the relation:

$$ {}^b\mathbf{X} = {}^b\mathbf{R}_w \, {}^w\mathbf{X} + {}^b\mathbf{P}_w \tag{A.11} $$

### A.2.1 Homogeneous coordinates

Let ${}^a\mathbf{X}$ denote the coordinates of a point $\mathbf{X}$ in a frame $\mathcal{F}_a$. Its homogeneous coordinates in $\mathcal{F}_a$ are defined as:

$$ \overline{{}^a\mathbf{X}} = \begin{bmatrix} {}^a\mathbf{X} \\ 1 \end{bmatrix} \tag{A.12} $$

### A.2.2 Homogeneous transformation matrix

A rigid transformation, or pose, in three-dimensional Euclidian space can be uniquely defined by the 4 × 4 matrix:

$$ \mathbf{T} = \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0}^T & 1 \end{bmatrix} \tag{A.13} $$

where $\mathbf{R} \in SO(3)$ and $\mathbf{t} \in \mathbb{R}^3$. The group of rigid transformations in three-dimensional Euclidian space is denoted $SE(3)$.

Equation A.11 can be re-written using homogeneous coordinates and the matrix ${}^b\mathbf{T}_w \in SE(3)$ such that:

$$ {}^b\mathbf{T}_w = \begin{bmatrix} {}^b\mathbf{R}_w & {}^b\mathbf{P}_w \\ \mathbf{0}^T & 1 \end{bmatrix} \tag{A.14} $$

as:

$$\overline{{}^{b}\mathbf{X}} =^{b} \mathbf{T}_{w} \overline{{}^{w}\mathbf{X}} \tag{A.15}$$

### A.2.3 Composition

Given three three-dimensional frames $\mathcal{F}_a$, $\mathcal{F}_b$ and $\mathcal{F}_c$, the rigid transformation ${}^{a}\mathbf{T}_c \in SE(3)$ between frames $\mathcal{F}_a$ and $\mathcal{F}_c$ can be deduced from the rigid transformations ${}^{a}\mathbf{T}_b$ and ${}^{b}\mathbf{T}_c$ respectively between frames $\mathcal{F}_b$ and $\mathcal{F}_a$ and frames $\mathcal{F}_c$ and $\mathcal{F}_b$ by the relation:

$${}^{a}\mathbf{T}_c =^{c} \mathbf{T}_b \, {}^{b}\mathbf{T}_c \tag{A.16}$$

## A.3 Camera modeling

A camera is a physical system which projects incoming light rays onto a set of photosensitive detectors, or pixels. Measurements from these pixels constitute an image. Given the incident angle of the incoming light rays, the geometry of the resulting image is a function of the internal geometry of the camera, which is usually defined by a set of parameters, named *intrinsic parameters*. Some applications may require a knowledge of the pose of the camera with respect to other sensors or frames: this pose is known as the *extrinsic parameters*. The current section focuses on image formation modeling according to the camera's intrinsic parameters.

### A.3.1 Pinhole model

The mathematical relationship between the coordinates of a 3D point and its projection onto the image of a camera is usually described using a simplified camera model, named *pinhole model*. In an ideal pinhole camera, the light from a scene passes through a very small aperture, or pinhole, and projects an inverted image on an *image plane*, where the photometric senors are located. The small aperture is considered a point, named *optical center*. This model is illustrated in Figure A.4. The optical axis is defined as perpendicular to the image place and passing through the optical center $\mathbf{P}_c$. The optical axis intersects the image plane in $\mathbf{O}$, which is called the *principal point*. The camera frame $\mathcal{F}_c$ is defined with origin $\mathbf{P}_c$ and $z$-axis aligned with the optical axis point in the opposite direction of the image plane. The focal length $f$ is the distance between the optical center $\mathbf{P}_c$ and the image plane along the optical axis.

Let $\mathbf{P}$ be a three dimensional point of coordinates ${}^{c}\mathbf{P} = \begin{bmatrix} X & Y & Z \end{bmatrix}^T$ in $\mathcal{F}_c$. The projection of $\mathbf{P}$ in the image plane is denoted $\mathbf{p}$ and has coordinates $\begin{bmatrix} x & y \end{bmatrix}^T$ in the image plane. According to Thales' theorem, one can get:

$$\begin{cases} x &= \frac{fX}{Z} \\ y &= \frac{fY}{Z} \end{cases} \tag{A.17}$$

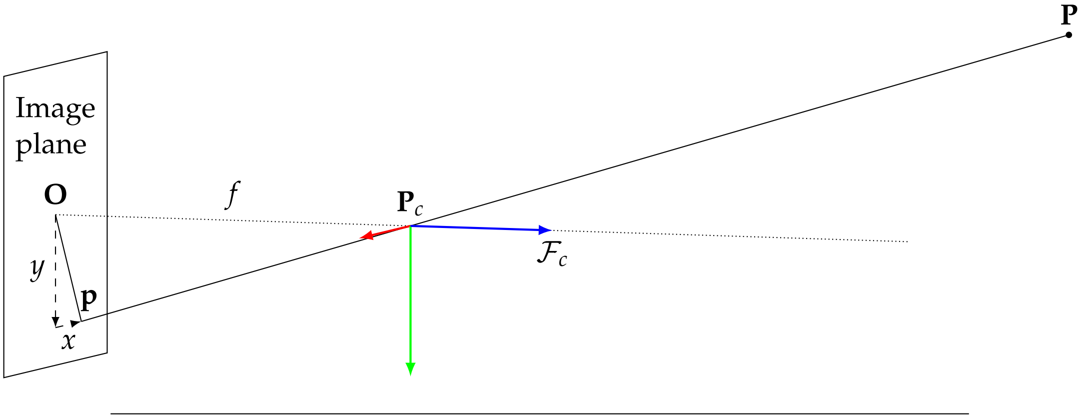


FIGURE A.4: Pinhole model

In homogeneous coordinates, one gets:

$$\overline{\mathbf{p}} = \frac{1}{Z} \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} {}^{c}\overline{\mathbf{P}} \tag{A.18}$$

$$= \frac{1}{Z} \mathbf{K}_f \, \mathbf{\Pi}_0 \, {}^{c}\overline{\mathbf{P}} \tag{A.19}$$

where:

$$\mathbf{K}_f = \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{A.20}$$

and:

$$\mathbf{\Pi}_0 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \tag{A.21}$$

### A.3.2 Pixel coordinates

Digital cameras output an image composed of discrete measurements from a pixel grid. Coordinates of image points are given in terms of discrete pixel coordinates, with respect to a two-dimensional frame $\mathcal{F}_\pi$ whose origin $\mathbf{P}_\pi$ is the image upper-left corner, as represented in Figure A.5. The $x$ and $y$-axis of $\mathcal{F}_\pi$ have the same direction of those of the camera frame $\mathcal{F}_c$.

Let $\overline{\mathbf{p}}' = \begin{bmatrix} u & v & 1 \end{bmatrix}^T$ be the pixel coordinates of $\overline{\mathbf{p}}$ and let $\begin{bmatrix} u_0 & v_0 & 1 \end{bmatrix}^T$ be the pixel coordinates of the principal point $\mathbf{O}$. Let $(w, h)$ and $(s_x, s_y)$ denote the width and height of the image in pixels and in meters respectively. One gets:

$$\overline{\mathbf{p}}' = \begin{bmatrix} \frac{w}{s_x} & s_\theta & u_0 \\ 0 & \frac{h}{s_y} & v_0 \\ 0 & 0 & 1 \end{bmatrix} \overline{\mathbf{p}} \tag{A.22}$$

$$= \mathbf{K}_s \, \overline{\mathbf{p}} \tag{A.23}$$

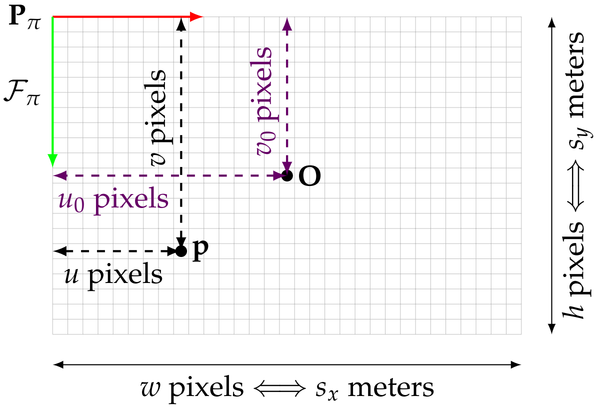


FIGURE A.5: Pixel coordinates

where $s_\theta$ accounts for a possible non-orthogonality of the pixel lines and columns on the sensor. Usually, lines and columns are orthogonal and thus $s_\theta = 0$, leading to:

$$\mathbf{K}_s = \begin{bmatrix} \frac{w}{s_x} & 0 & u_0 \\ 0 & \frac{h}{s_y} & v_0 \\ 0 & 0 & 1 \end{bmatrix} \tag{A.24}$$

### A.3.3 Calibration matrix

According to Sections A.3.1 and A.3.2, the coordinates in pixels of the projection of a three-dimensional point $\mathbf{P}$ in the image are given by the relation:

$$\overline{\mathbf{p}}' = \frac{1}{Z}\,\mathbf{K}_s\,\mathbf{K}_f\,\mathbf{\Pi}_0\,{}^c\mathbf{P} \tag{A.25}$$

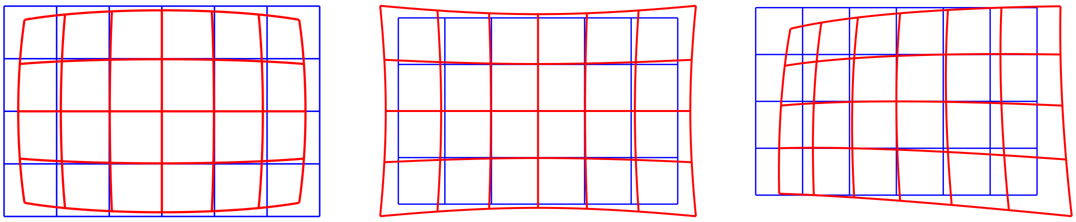
$$= \frac{1}{Z}\,\mathbf{K}\,\mathbf{\Pi}_0\,{}^c\mathbf{P} \tag{A.26}$$

where ${}^c\mathbf{P}$ the coordinates of $\mathbf{P}$ in the camera frame $\mathcal{F}_c$ with $z$-coordinate $Z$ and with $\mathbf{K}_s$, $\mathbf{K}_f$ and $\mathbf{\Pi}_0$ defined in Sections A.3.1 and A.3.2. The matrix $\mathbf{K} = \mathbf{K}_s\,\mathbf{K}_f$ is called *calibration matrix*. The calibration matrix is parameterized by the following eight intrinsic parameters:

- focal length $f$;
- image width $w$ and height $h$ in pixels;
- image width $s_x$ and height $s_y$ in meters;
- pixel coordinates $\begin{bmatrix} u_0 & v_0 \end{bmatrix}^T$ of the principal point $\mathbf{O}$;
- possible flaw of orthogonality parameter $s_\theta$ if pixel columns and lines on the sensor or non-orthogonal.

### A.3.4 Distortion correction

However, real cameras involve lenses which usually have some distortion, which needs to be corrected. Image distortion is generally assumed to be a combination of radial and tangential distortion. The effect on the image is illustrated in Figure A.6.



(A) Barrel radial distortion ($k_1 < 1$)

(B) Pin-cushion radial distortion ($k_1 > 1$)

(C) Tangential distortion

FIGURE A.6: Image distortion examples for regularly spaced values of $x'$ and $y'$ (see Equation A.27). The blue grid shows zero distortion.

Radial distortion is usually modeled by six coefficients $k_i$, $i \in \{1 \ldots 6\}$ and tangential distortion is commonly modelled by two coefficients $(p_1, p_2)$. Given a 3D point $\mathbf{P}$ of coordinates ${}^c\mathbf{P} = \begin{bmatrix} X & Y & Z \end{bmatrix}^T$ in camera frame $\mathcal{F}_c$ and its projection $\mathbf{p}$ in the image, the coordinates $\begin{bmatrix} x & y \end{bmatrix}^T$ of $\mathbf{p}$ in the image are given by the relation:

$$\begin{cases} x' &= \frac{fX}{Z} \\ y' &= \frac{fY}{Z} \\ x &= x' \frac{1+k_1r^2+k_2r^4+k_3r^6}{1+k_4r^2+k_5r^4+k_6r^6} + 2p_1x'y' + p_2(r^2 + 2x'^2) \\ y &= y' \frac{1+k_1r^2+k_2r^4+k_3r^6}{1+k_4r^2+k_5r^4+k_6r^6} + p_1(r^2 + 2y'^2) + 2p_2x'y' \end{cases} \tag{A.27}$$

where $r^2 = x'^2 + y'^2$. Distortion parameters are also intrinsic parameters which need to be calibrated in order to know the relationship between the 3D position of a point in the camera frame and its projection in the image.

### A.3.5 Camera intrinsic calibration

The intrinsic calibration of a camera consists in the estimation of the intrinsic parameters $\{f, \frac{w}{s_x}, \frac{h}{s_y}, s_\theta, u_0, v_0, k_1, \ldots, k_6, p_1, p_2\}$. This calibration is performed using a target with a pattern of known shape and dimension. The intrinsic parameters are deduced from multiple views of the calibration target with different viewing angles. If the camera is to be used underwater, this calibration is usually performed underwater in order to include the refraction of light rays between water, camera housing and camera in the calibrated distortion parameters.

# Appendix B

# Straight-line model

The current section introduces the straight line model, where Section B.1 lists the assumptions considered and described the application scope of the system under study, and Section B.2 gives the equations of the model. These assumptions, scope and model have been developed by Christophe Viel, CNRS-LabSTICC, Brest, France, and are not a contribution of this thesis. They are nonetheless essential to introduce further developments based on this straight-line model in Sections 3.3 and 3.4 which belong to the contributions of this thesis.

## B.1 Assumptions and application scope

The following assumptions are considered:

**(A1)** The ratio of mass to buoyancy of the umbilical is negligible compared to the ballasts' weight and the buoys' buoyancy used in the configuration, or currents applied on ballast/buoys;

**(A2)** The length variation of the umbilical is negligible compared to its length, considered constant;

**(A3)** When the umbilical is taut, its geometry can be assimilated to straight lines between defined points, here the ballasts, the buoys, the boat and the ROV;

**(A4)** The motion of anchors and of the cable end point which is not located on the ROV are assumed not to be affected by the action of the umbilical and the ROV, and so can be considered motionless;

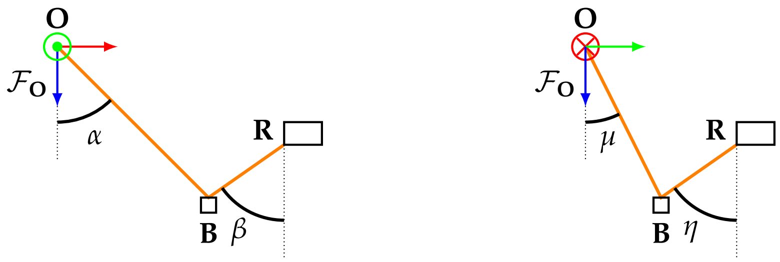
**(A5)** ROV motion and cable length are such that the umbilical remains taut. Constraints to have this assumption valid are discussed in (Viel, 2022a).

Due to the umbilical modeled by straight lines, with a negligible length variation, the scope of application of the system is for ROVs with an umbilical shorter than 50 m. The umbilical must also be flexible and allow the sliding elements to move freely on the cable. Applications of this system include:

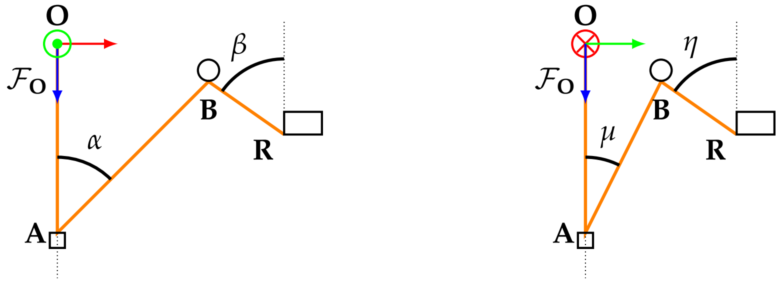
- The exploration of shallow water from a boat with a depth of less than 50 m;
- The umbilical between a ROV and its cage in case of deep exploration. The cable between the boat and the cage, then, does not need to respect the assumptions described in the current section;
- A chain of ROVs connected with the same umbilical. The model is applicable for each section of cable between each pair of ROVs.

## B.2 Model equations

Let us consider a system composed of a ROV tethered to another vehicle or to a fixed point by its umbilical. These attachment points are denoted **R** and **O** respectively. A reference frame $\mathcal{F}_\mathbf{O}$ is defined with origin **O** and its z-axis vertical, downwards. The first part of the cable from **O** is constrained to be vertical down to an anchor placed in **A**, and a sliding buoy or ballast is placed along the cable between **A** and **R**. The position of the sliding element is denoted **B**. Depth of the anchor in **A** depends on the intended application and the sliding element in use. A non-zero **A** depth is in fact necessary if the sliding element is a buoy. The system is represented in Figure B.1 in both sliding buoy and sliding ballast configuration.



(A) Sliding ballast in plane $(\mathbf{O}, \mathbf{x_O}, \mathbf{z_O})$ (B) Sliding ballast in plane $(\mathbf{O}, \mathbf{y_O}, \mathbf{z_O})$



(C) Sliding buoy in plane $(\mathbf{O}, \mathbf{x_O}, \mathbf{z_O})$ (D) Sliding buoy in plane $(\mathbf{O}, \mathbf{y_O}, \mathbf{z_O})$

FIGURE B.1: Single sliding element system. Figures B.1a and B.1b present a single-ballast configuration with $\mathbf{A} = \mathbf{O}$, and Figures B.1c and B.1d present a single-buoy configuration.

An umbilical of length $l$ is thus divided in three parts: segment **OA** of length $l_0$ between the first cable end and the anchor, segment **AB** of length $l_1$ between the anchor and the sliding element, and segment **BR** of length $l_2$ between the sliding element and the cable attachment point on the ROV. Cable length between anchor **A** and attachment point **R** is fixed and equal to $L_t = l_1 + l_2$. Let us define angles $\alpha$, $\beta$, $\mu$ and $\eta$ as follows:

- $\alpha$ is the angle between the $z$-axis of $\mathcal{F}_\mathbf{O}$ and segment **AB** around the $y$-axis of $\mathcal{F}_\mathbf{O}$, in plane $(\mathbf{O}, \mathbf{x_O}, \mathbf{z_O})$;
- $\beta$ is the angle between segment **BR** and the $z$-axis of $\mathcal{F}_\mathbf{O}$ around the $y$-axis of $\mathcal{F}_\mathbf{O}$, in plane $(\mathbf{O}, \mathbf{x_O}, \mathbf{z_O})$;
- $\mu$ is the angle between segment **AB** and the $z$-axis of $\mathcal{F}_\mathbf{O}$ around the $x$-axis of $\mathcal{F}_\mathbf{O}$, in plane $(\mathbf{O}, \mathbf{y_O}, \mathbf{z_O})$;
- $\eta$ is the angle between the $z$-axis of $\mathcal{F}_\mathbf{O}$ and segment **BR** around the $x$-axis of $\mathcal{F}_\mathbf{O}$, in plane $(\mathbf{O}, \mathbf{y_O}, \mathbf{z_O})$.

These angles are represented in Figure B.1. In addition, we define $l_{1x}$ and $l_{1y}$ the length of the projection of vector **AB** on planes $(\mathbf{O}, \mathbf{x_O}, \mathbf{z_O})$ and $(\mathbf{O}, \mathbf{y_O}, \mathbf{z_O})$ respectively. $l_{2x}$ and $l_{2y}$ are defined similarly for vector **BR**. Finally, let $s_b$ be defined such that $s_b = 1$ if the sliding element is a ballast and $s_b = -1$ if the sliding element is a buoy.

The coordinates of **R** can be expressed in frame $\mathcal{F}_{\mathbf{O}}$ as

$$\begin{cases} {}^{\mathbf{O}}x_{\mathbf{R}} = & l_{1x}\sin(\alpha) + l_{2x}\sin(\beta) \\ {}^{\mathbf{O}}y_{\mathbf{R}} = & l_{1y}\sin(\mu) + l_{2y}\sin(\eta) \end{cases} \tag{B.1}$$

and

$$\begin{cases} {}^{\mathbf{O}}z_{\mathbf{R}} = & l_0 + s_b l_{1x}\cos(\alpha) - s_b l_{2x}\cos(\beta) \\ {}^{\mathbf{O}}z_{\mathbf{R}} = & l_0 + s_b l_{1y}\cos(\mu) - s_b l_{2y}\cos(\eta) \end{cases} \tag{B.2}$$

where

$$l_1^2 = l_{1x}^2 + \sin(\mu)^2 l_{1y}^2 \tag{B.3}$$

$$l_2^2 = l_{2x}^2 + \sin(\eta)^2 l_{2y}^2 \tag{B.4}$$

$$L_t = l_1 + l_2 \tag{B.5}$$

From these equations, one can get

$$l_{1x}^2 = \frac{l_1^2}{\left(1 + \tan(\mu)^2\cos(\alpha)^2\right)} \tag{B.6}$$

$$l_{1y}^2 = \frac{l_1^2}{\left(\sin(\mu)^2 + \left(\frac{\cos(\mu)}{\cos(\alpha)}\right)^2\right)} \tag{B.7}$$

$$l_{2x}^2 = \frac{l_2^2}{\left(1 + \tan(\eta)^2\cos(\beta)^2\right)} \tag{B.8}$$

$$l_{2y}^2 = \frac{l_2^2}{\left(\sin(\eta)^2 + \left(\frac{\cos(\eta)}{\cos(\beta)}\right)^2\right)} \tag{B.9}$$

$l_1$ and $l_2$ can be expressed as a function of $L_t$, ${}^{\mathbf{O}}z_{\mathbf{R}}$, $\alpha$, $\beta$, $\mu$ and $\eta$ as follows:

$$l_1 = L_t - l_2 \tag{B.10}$$

$$l_2 = \frac{\left(\frac{L_t\cos(\alpha)}{a_1} - s_b\left({}^{\mathbf{O}}z_{\mathbf{R}} - l_0\right)\right)}{\left(\frac{\cos(\alpha)}{a_1} + \frac{\cos(\beta)}{a_2}\right)} \tag{B.11}$$

where

$$a_1 = \sqrt{1 + \tan(\mu)^2\cos(\alpha)^2} \tag{B.12}$$

and

$$a_2 = \sqrt{1 + \tan(\eta)^2\cos(\beta)^2} \tag{B.13}$$

Consequently, the coordinates of **R** in $\mathcal{F}_{\mathbf{O}}$ can be deduced from $L_t$, ${}^{\mathbf{O}}z_{\mathbf{R}}$, $\alpha$, $\beta$, $\mu$, $\eta$ and fixed length $l_0$.

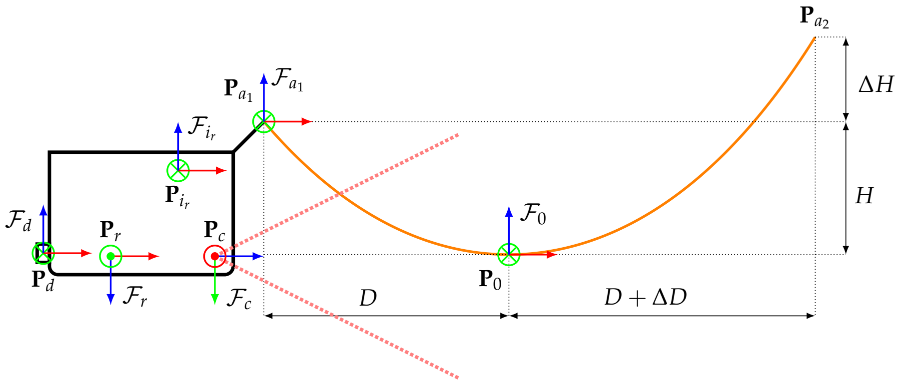
# Appendix C

# Visual-based catenary shape estimation (Laranjeira et al., 2019)

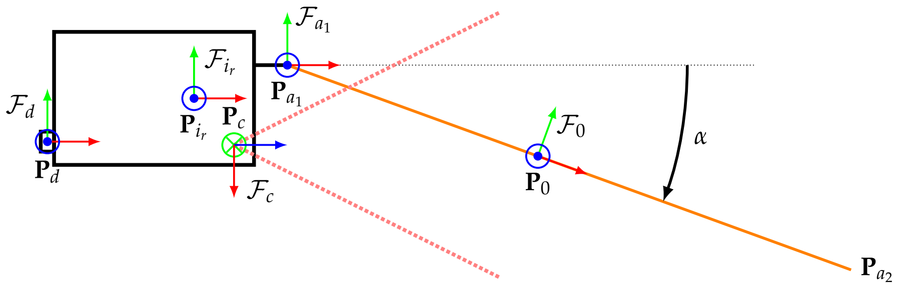
The current section describes the visual-based catenary shape estimation of underwater cables for tethered robots from (Laranjeira et al., 2019), which is compared to our IMU-based approach is Section 3.4.1. Section C.1 presents the model and the assumptions. Section C.2 explains the cable detection process in the image and Section C.3 describes the model parameters estimation process.

## C.1 System modeling and assumptions

Just like in Section 3.3.2, we consider a subsystem of a robot chain composed of a robot and the section of cable in front of it, as represented in Figure C.1.



(A) Side view



(B) Top view

FIGURE C.1: Subsystem under study

$\mathcal{F}_r$ is the robot frame in SNAME convention. $\mathbf{P}_{a_1}$ and $\mathbf{P}_{a_2}$ are the cable attachment points and $\mathbf{P}_0$ is the cable's lowest point. Frame $\mathcal{F}_{a_1}$ is defined with origin $\mathbf{P}_{a_1}$ and axes aligned $\mathcal{F}_r$. The cable is modeled by a catenary of catenary frame $\mathcal{F}_0$ (see Section 3.2.1). In addition, the robot embeds a camera in $\mathbf{P}_c$, oriented towards the cable, and a pressure sensor in $\mathbf{P}_d$. Let $\mathcal{F}_c$ denote the camera frame, with origin $\mathbf{P}_c$. As detailed in Section 3.2.1, the three-dimensional catenary shape of the cable is fully described by the three parameters $\{H, \Delta H, \alpha\}$.

In addition, the following assumptions are made:

1. the robot remains horizontal ;
2. the axes of the camera frame $\mathcal{F}_c$ are aligned with those of the robot frame $\mathcal{F}_r$ ;
3. cable color is uniform and distinct from the robot's environment.

## C.2 Cable detection

The cable is detection in the image from the camera by a color segmentation in the HSV space (Crevier, 1993), with threshold values set manually. To this end, an orange cable is used in (Laranjeira et al., 2019). In our experiments reported in Section 3.4.1, we selected a red cable. Red is known to be quickly absorbed by the water, but the cable always remained close enough to the camera during our experiments to allow the choice of this color. Cable detection is illustrated in Figure C.2, for the red cable used in Section 3.4.1.3.

(A) Input image

(B) Segmented cable

FIGURE C.2: Cable detection in the image

## C.3 Model parameters estimation

Parameter $\Delta H$ is measured using the robot's embedded pressure sensor. While (Laranjeira et al., 2019) assumed that the robot stays horizontal, we preferred using the methodology described in Section 3.3.1 for this step in order to take into account small pitch and roll variations in estimating $\Delta H$.

In (Laranjeira et al., 2019), the estimation of parameters $H$ and $\alpha$ is performed under the assumption that points $\mathbf{P}_c$, $\mathbf{P}_{a_1}$ and $\mathbf{P}_{a_2}$ are not aligned, what is called the non-degenerate case. Because it only involved a minor change in the algorithm, we modified the algorithm provided by (Laranjeira et al., 2019) in order to handle the degenerate case, with points $\mathbf{P}_c$, $\mathbf{P}_{a_1}$ and $\mathbf{P}_{a_2}$ aligned. These two cases are detailed in Sections C.3.1 and C.3.2 respectively.

### C.3.1 Non-degenerate case

In the non-degenerate case, (Laranjeira et al., 2019) estimates parameters $H$ and $\alpha$ is performed in two steps, including an initial guess and an optimized estimation, described in Sections C.3.1.1 and C.3.1.2 respectively.

#### C.3.1.1 Initial guess

The initial guess for parameters $\{H, \alpha\}$ is computed by assuming that the lowest cable point observed in the image is the projection of $\mathbf{P}_0$. Actually, these points do not match exactly due to perspective projection, but they are expected to be close enough to make the initial guess consistent. Nonetheless, if the bottom of the cable moves out of the field of view, the lowest point depth is underestimated.

Let $\begin{bmatrix} x_B & y_b \end{bmatrix}^T$ denote the image coordinates of the lowest cable point observed in the image (see Appendix A.3 for more details on image coordinates). If this point coincides with the projection of $\mathbf{P}_0$ and under the zero roll and pitch assumptions, one can get:

$$H = y_b({}^cZ_{a_1} + D\cos(\alpha)) - {}^cY_{a_1} \tag{C.1}$$

where ${}^cY_{a_1}$ and ${}^cZ_{a_1}$ are the $y$ and $z$-coordinates of $\mathbf{P}_{a_1}$ in $\mathcal{F}_c$, and $D$ is displayed on Figure C.1.

In the non-degenerate case, $x_B\cos(\alpha) + \sin(\alpha) \neq 0$. As a result, it can be shown that:

$$D = \frac{{}^cX_{a_1} - x_B^c Z_{a_1}}{x_B\cos(\alpha) + \sin(\alpha)} \tag{C.2}$$

and $\{H, \alpha\}$ are such that:

$$g(\alpha) = C(\alpha)H(\alpha) - \cosh(C(\alpha)D(\alpha)) + 1 \qquad = 0 \tag{C.3}$$

where $C$ is defined in (3.3).

$\alpha$ is then estimated numerically by the bisection method, with $\alpha$ value bounded as follows, due to the geometry of the system:

$$\begin{cases} \alpha < \arctan(-x_B) & \text{if } {}^cX_{a_1} - x_B{}^cZ_{a_1} > 0 \\ \alpha > \arctan(-x_B) & \text{if } {}^cX_{a_1} - x_B{}^cZ_{a_1} < 0 \end{cases} \tag{C.4}$$

In addition, if $y_B$ and $\xi_{\max} +^c Y_{a_1} - y_B^c Z_{a_1}$ have the same sign, with:

$$\xi_{\max} = \max(0, -\Delta H) \tag{C.5}$$

then:

$$\begin{cases} \alpha > \arctan\left(\frac{y_B({}^cX_{a_1} - x_B{}^cZ_{a_1})}{\xi_{\max} + {}^cY_{a_1} - y_B{}^cZ_{a_1}} - x_B\right) & \text{if } y_B({}^cX_{a_1} - x_B{}^cZ_{a_1}) > 0 \\ \alpha < \arctan\left(\frac{y_B({}^cX_{a_1} - x_B{}^cZ_{a_1})}{\xi_{\max} + {}^cY_{a_1} - y_B{}^cZ_{a_1}} - x_B\right) & \text{if } y_B({}^cX_{a_1} - x_B{}^cZ_{a_1}) < 0 \end{cases} \tag{C.6}$$

Finally, if $y_B$ et $\xi_{\min} +^c Y_{a_1} - y_B^c Z_{a_1}$ have the same sign with:

$$\xi_{\min} = \min(\frac{L - \Delta H}{2}, -L) \tag{C.7}$$

then:

$$\begin{cases} \alpha > \arctan\left(\frac{y_B({}^cX_{a_1} - x_B^c Z_{a_1})}{\zeta_{\min} + {}^cY_{a_1} - y_B^c Z_{a_1}} - x_B\right) & \text{if } y_B({}^cX_{a_1} - x_B{}^cZ_{a_1}) > 0 \\ \alpha < \arctan\left(\frac{y_B({}^cX_{a_1} - x_B^c Z_{a_1})}{\zeta_{\min} + {}^cY_{a_1} - y_B^c Z_{a_1}} - x_B\right) & \text{if } y_B({}^cX_{a_1} - x_B{}^cZ_{a_1}) < 0 \end{cases} \tag{C.8}$$

$H$ is then deduced using (C.1) and (C.2). Note that these bounds are slightly different from the work (Laranjeira et al., 2019) because they have been recomputed in order to take into account a non-zero value of ${}^cX_{a_1}$.

#### C.3.1.2 Optimized estimation

Assuming zero pitch and roll angles, the projection of the catenary in the image is given as:

$$y(a,b,x) = \frac{1}{{}^cZ}[-\frac{\cosh(C\zeta - CD) - 1}{C} + aH_{\max} +^c Y_{a_1}] \tag{C.9}$$

where $\begin{bmatrix} x & y \end{bmatrix}^T$ denote the image coordinates of a cable point projection, and with $a = \frac{H}{H_{\max}}$ and $b = \sin(\alpha)$. $H_{\max}$ is defined arbitrarily as a maximum value allowed for $H$. $\zeta$ and ${}^cZ$ are defined as follows:

$$\zeta = \frac{{}^cX_{a_1} - x^cZ_{a_1}}{b + x\sqrt{1 - b^2}} \tag{C.10}$$

and

$${}^cZ = \frac{{}^cX_{a_1}\sqrt{1 - b^2} + b^cZ_{a_1}}{b + x\sqrt{1 - b^2}} \tag{C.11}$$

Initial value of parameters $\{a, b\}$ are computed from the $\alpha$ and $H$ parameters estimated with the initial guess depicted in Section C.3.1.1. Values of $\{a, b\}$ are then optimized using a Gauss-Newton algorithm that minimizes the function

$$\Gamma(a,b) = \sum_i r_i(a,b)^2 \tag{C.12}$$

with $r_i(a,b) = y_i - y(a,b,x_i)$, where $\begin{bmatrix} x_i & y_i \end{bmatrix}^T$ are the observed image coordinates of a cable point detected using the method described in Section C.2.

### C.3.2 Degenerate case

Our estimation method to handle the degenerate case used the lowest cable point observed in the image, juste like the initial guess of the non-degenerate case. In this case, one gets:

$$\alpha = \arctan\left(-\frac{{}^cX_{a_1}}{{}^cZ_{a_1}}\right) \tag{C.13}$$

Because (C.1) is still valid, it can be shown that $H$ is such that:

$$\begin{aligned} f(H) &= y_B - \frac{H +^c Y_{a_1}}{{}^cZ_{a_1} + \frac{1}{C(H)}\operatorname{arccosh}(C(H)H + 1)\cos(\alpha)} && \text{(C.14)} \\ &= 0 && \text{(C.15)} \end{aligned}$$

and that $H$ is the only root of $f$. This equation is solved using the bisection method.

# Appendix D

# Motion capture ground truth recording

This chapter details how the robotic system was tracked using a mocap system in Sections 3.2.1.2, 3.4.1 and 3.4.2. Because of their high accuracy, mocap measurements were used as a ground truth for validation embedded sensor based state estimation approaches. Section D.1 describes the motion capture system used in this work. Section D.2 present the experimental set-up for tracking our robotic system. Section D.3 explains how these mocap measurements were aligned with respect to the system's embedded sensors in order to be used as a ground truth, including temporal and frame alignment.

## D.1 Motion capture system description

This work uses two Qualisys motion capture systems[1]: an aerial one in Section 3.4.1.2 and an underwater one otherwise. While the cameras and connection cables are different from these two systems (see Table D.1 and Figure D.1), they work in a similar way in allowing the three-dimensional tracking of small markers which can be placed on a system which can be a rigid or deformable body.

TABLE D.1: Motion capture cameras

| | **Oqus 400** | **Miqus m5u** |
|---|---|---|
| Field | Airborne | Underwater |
| Max. field of view | $70° \times 70°$ | $51° \times 51°$ |
| Sensor resolution | 3 MP | 4 MP |
| Max. capture distance with 19 mm markers | 35 m | 17 m |
| Max. framerate | 1100 fps | 180 fps |
| Wavelength range | Infrared | Ultraviolet |

The mocap cameras emit light rays close to visible light, which are reflected by the markers and allow their detection in the camera frames. Markers observations from multiple cameras are then fused in order to triangulate their position. Two kinds of markers are used in the current work: 19 mm spherical markers, and pieces of reflective tape (Figure D.2). These markers both reflect infrared and ultraviolet radiations and are thus compatible with both airborne and underwater camera systems.

---

[1] https://www.qualisys.com/

(A) Oqus 400 (airborne) (B) Miqus m5u (underwater)

FIGURE D.1: Qualisys mocap cameras used. Courtesy of Qualisys.

(A) Spherical marker (B) Reflective tape

FIGURE D.2: Qualisys reflective markers used. Courtesy of Qualisys.

This process requires a preliminary calibration of the intrinsic and extrinsic parameters of the mocap cameras. This calibration involves two targets: one is fixed, and the other one is moved around the volume covered by the field of view of the cameras. The fixed calibration target is used to define the origin and axis of a reference frame $\mathcal{F}_{w,q}$ in which mocap measurements are given, as represented in Figure D.3. The fixed target is usually placed on the floor, or at the bottom of the pool, in order to have $\mathcal{F}_{w,q}$'s $z$-axis vertical, upwards. Target dimensions are given by the constructor with a $10^{-4}$ m precision. Calibration is deemed valid only if the residual in mobile target tracking is below 3 mm. The accuracy of the tracking of reflective markers after calibration is of about 1 cm.

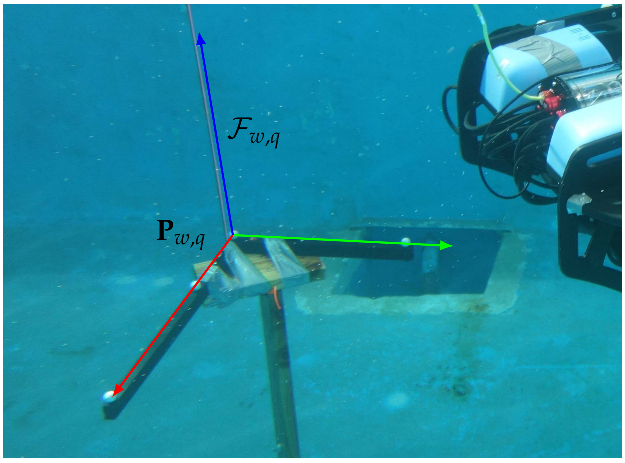


FIGURE D.3: Mocap reference frame $\mathcal{F}_{w,q}$, wih origin $\mathbf{P}_{w,q}$

System calibration and parameter tuning, marker observation matching from different camera views and marker trajectory tracking over time are performed by the Qualisys Track Manager (QTM) software from Qualisys, which also provides post-processing tools.

It is worth noticing that these mocap systems can be disturbed by additional light sources with overlapping range with respect to the light emitted by the active cameras. For instance, the underwater system is highly disturbed by the sunlight and is thus not suitable for outdoor, shallow depth tracking during daytime.

## D.2 Experimental tracking set-up

### D.2.1 Camera positioning

For each experiment, the volume in which the objects to be tracked will be placed, or *operational volume* is defined. Mocap cameras are then placed in order to have this volume covered by their field of view. Any point of the operational volume needs to be in the field of at least three mocap cameras. Cameras are placed on modular supports in order to adapt their placement to the configuration required. For an airborne set-up, cameras are simply placed on tripods. For an underwater configuration, camera supports can be positioned on the edges or at the bottom of the pool. Figure D.4 shows an example of mocap camera positioning in the Cephismer pool.

(A) (B)

FIGURE D.4: Examples of motion capture camera positioning in the Cephismer pool.

### D.2.2 Robotic system tracking

Experiments from Chapter 4 require the tracking of cables, and rigid bodies, namely a BlueROV but also a pulley in Sections 3.4.2. The QTM software allows to track a rigid body, or *6-DOF body* from a set of minimum three markers placed on it. A mobile frame $\mathcal{F}_{6DOF,q}$ associated to the rigid body is then defined. Its pose with respect to the mocap reference frame $\mathcal{F}_{w,q}$ can be computed by the QTM software anytime at least three markers of the 6-DOF body are tracked simultaneously. The origin and axis of $\mathcal{F}_{6DOF,q}$ can be set-up manually from the markers points of the 6-DOF body. In addition, marker points of the 6-DOF body which may not have been detected and tracked correctly at certain times are reconstructed automatically from the 6-DOF body pose. Both the ROV and the pulley were equipped with spherical markers. Frames associated to the ROV and pulley 6DOF bodies are denoted $\mathcal{F}_{r,q}$ and $\mathcal{F}_{b,q}$ respectively. Lastly, cable tracking is performed by tracking tape markers regularly placed along the cable. Figure D.5 shows the robotic system involved in Section 3.4.1.3's experiments and its tracking in $\mathcal{F}_{w,q}$ by the mocap system.

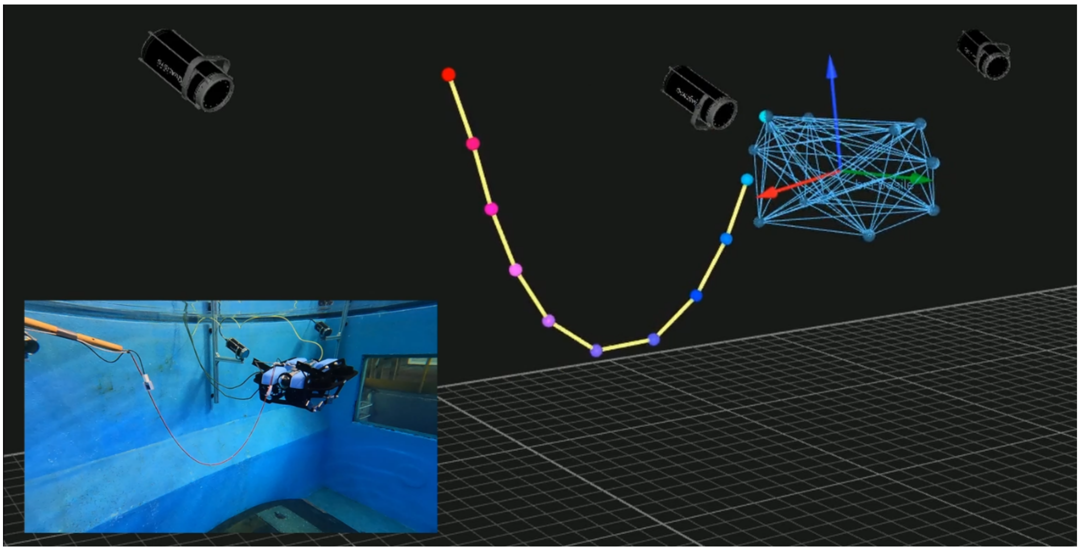


FIGURE D.5: Motion tracking of Section 3.4.1.3's robotic system

## D.3 Alignment with respect to embedded sensors

In Sections 3.4.1 and 3.4.2, mocap data is used as a ground truth for evaluating embedded sensor based robotic system state estimation. To this end, embedded sensor measurements and mocap data have to be synchronized both temporally and spatially, in order to be expressed with the same timeline and in the same frame.

### D.3.1 Frames, notations and assumptions

We consider a robotic system including a robot tracked by a motion capture system. The robot embeds a water pressure sensor in $\mathbf{P}_d$ and an IMU in $\mathbf{P}_{i_r}$ as illustrated in Figure D.6. $\mathcal{F}_{w,q}$ is the Qualisys reference frame. The robot embeds a water pressure sensor in $\mathbf{P}_d$ and an IMU in $\mathbf{P}_{i_r}$. The IMU frame is denoted $\mathcal{F}_{i_r}$, and the IMU reference frame is $\mathcal{F}_{w,i}$ with a vertical, upward $z$-axis. The following assumptions are considered:

- **(A1)** The water pressure sensor is calibrated such that the depth of $\mathbf{P}_d$ can be computed from the pressure measurement
- **(A2)** The IMU is able to measure the orientation of $\mathcal{F}_{i_r}$ with respect to $\mathcal{F}_{w,i}$ with a negligible error during the mocap recording duration
- **(A3)** The embedded sensors are already temporally synchronized and their relative position is known, such that the position of $\mathbf{P}_d$ in $\mathcal{F}_{i_r}$ is known

**(A4)** $\mathcal{F}_{i_r}$ has its $x$-axis, $y$-axis and $z$-axis aligned with the robots's forward, lateral and upward axis respectively

**(A5)** The mocap markers placed on the robot are such that pairs of markers are aligned with the robots's forward, lateral and upward axis

**(A6)** The expression of $\mathbf{P}_d$ from robot marker points is known, and thus the coordinates of $\mathbf{P}_d$ in $\mathcal{F}_{r,q}$ are known

**(A7)** $\mathcal{F}_{w,q}$ has been defined using the static calibration target such that its $z$-axis is vertical, upward. Consequently, the rotation between $\mathcal{F}_{w,q}$ and $\mathcal{F}_{w,i}$ is a unit rotation around their common $z$-axis.

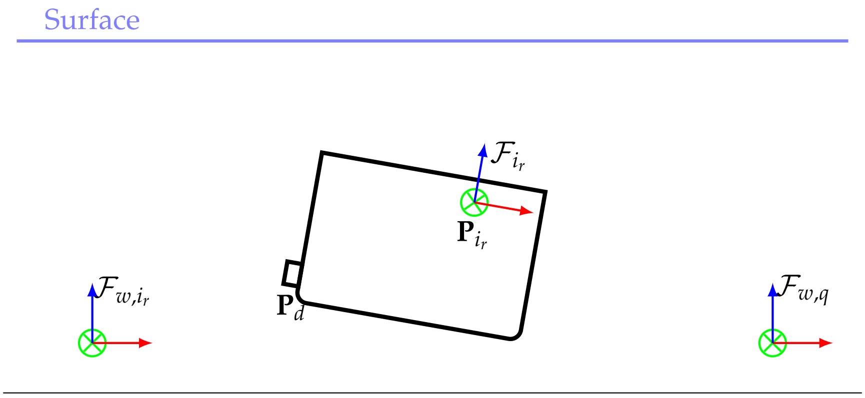


FIGURE D.6: System and frames

### D.3.2 Temporal alignment

Because their are recorded by different computers, embedded sensor data and mocap measurements have no reason to be temporally synchronized. Consequently, at the same global time, mocap and embedded sensor data will have respective timestamps $t_q$ and $t_r$ such that:

$$t_q = t_r + \Delta t \tag{D.1}$$

$\Delta t$ is estimated by aligning depth measurement realized from the water pressure sensor and from the mocap recording. Let us denote $d_{\mathbf{P}_d,r}$ and $d_{\mathbf{P}_d,q}$ the measurements of the depth of point $\mathbf{P}_d$ from the robot's pressure sensor and the mocap system respectively. As explained in Section 3.3.1:

$$d_{\mathbf{P}_d,r} \frac{P_{\mathbf{P}_d} - P_0}{\rho_{\text{water}} * g} \tag{D.2}$$

where $P_0$ is assumed to have been calibrated. The coordinates of $\mathbf{P}_d$ in $\mathcal{F}_{w,q}$ are given by:

$${}^{w,q}\mathbf{P}_d = {}^{w,q}\mathbf{T}_{r,q}\,{}^{r,q}\mathbf{P}_d \tag{D.3}$$

where ${}^{w,q}\mathbf{T}_{r,q}$ is measured by the mocap and ${}^{r,q}\mathbf{P}_d$ is assumed to be known, from Section D.3.1. Then:

$$d_{\mathbf{P}_d,q} = {}^{w,q}z_d + c_1 \tag{D.4}$$

where ${}^{w,q}z_d$ is the $z$-coordinate of ${}^{w,q}\mathbf{P}_d$ and $c_1$ is a constant which accounts for the depth of the origin of $\mathcal{F}_{w,q}$. $\Delta t$ is then tuned in order to have $d_{\mathbf{P}_d,r}(t_r + \Delta t) - {}^{w,q}z_d(t_q) = c1$.

### D.3.3 Frame alignment

This section describes the calibration of the rotation ${}^{w_i}\mathbf{R}_{w,q}$ between the reference frames $\mathcal{F}_{w,q}$ and $\mathcal{F}_{w,i}$ of the Qualisys and of the IMU. Because the experiments presented in this work only involve relative measurements between points of the system and orientation measurements, calibration of ${}^{w_i}\mathbf{R}_{w,q}$ is sufficient for mocap and embedded sensor measurements alignment. This transformation is such that:

$$ {}^{w_i}\mathbf{R}_{w,q} = \mathbf{R}_z(\psi) \tag{D.5} $$

and is estimated using the relation:

$$ {}^{w_i}\mathbf{R}_{w,q} = {}^{w_i}\mathbf{R}_{i_r}(t)\,{}^{i_r}\mathbf{R}_{r,q}\,{}^{r,q}\mathbf{R}_{w,q}(t) \tag{D.6} $$

at any time $t$ of the sequence, where ${}^{w_i}\mathbf{R}_{i_r}$ is measured by the embedded IMU and ${}^{r,q}\mathbf{R}_{w,q}$ is measured by the Qualisys mocap system.

Marker positioning on the robot is given in Figure D.7. The unit vectors of frame $\mathcal{F}_{i_r}$ can be expressed from points $\mathbf{P}_{lbb}$ (*left-bottom-back*), $\mathbf{P}_{rbf}$ (*right-bottom-front*) and $\mathbf{P}_{rbb}$ (*right-bottom-back*) by the relation:

$$ \begin{cases} {}^{r,q}\mathbf{x}_{i_r} & = \frac{\mathbf{P}_{rbb}\mathbf{P}_{rbf}}{\|\mathbf{P}_{rbb}\mathbf{P}_{rbf}\|} \\ {}^{r,q}\mathbf{y}_{i_r} & = \frac{\mathbf{P}_{rbb}\mathbf{P}_{lbb}}{\|\mathbf{P}_{rbb}\mathbf{P}_{lbb}\|} \\ {}^{r,q}\mathbf{z}_{i_r} & = {}^{r,q}\mathbf{x}_{i_r} \times {}^{r,q}\mathbf{y}_{i_r} \end{cases} \tag{D.7} $$

where $\times$ denotes the vector cross product and $\|\cdot\|$ is the Euclidian norm. Finally:

$$ {}^{i_r}\mathbf{R}_{r,q} = {}^{r,q}\mathbf{R}_{i_r}^{T} \tag{D.8} $$

$$ = \left[{}^{r,q}\mathbf{x}_{i_r} \quad {}^{r,q}\mathbf{y}_{i_r} \quad {}^{r,q}\mathbf{z}_{i_r}\right]^{T} \tag{D.9} $$

(A) Right side (B) Left side

FIGURE D.7: BlueROV equipped with mocap markers

$\psi(t) \in ]-\pi, \pi]$ is computed as the yaw of ${}^{w_i}\mathbf{R}_{i_r}(t)\,{}^{i_r}\mathbf{R}_{r,q}\,{}^{r,q}\mathbf{R}_{w,q}(t)$ in the $xyz$ intrinsic Euler convention (see Appendix A.1.2). $\psi$ is finally taken as the mean value of

$\psi(t)$ over the sequence, assuming $\psi(t)$ values are far enough from $-\pi$ or $\pi$ to have remain continuous over time.

## Appendix E

# Underwater evaluation of monocular VSLAM: trajectories

The current section is a complement to Section 4.2. It illustrates the trajectories obtained for the different VSLAM algorithms compared in Section 4.2 on the eight underwater test datasets considered. For each dataset and each algorithm, the trajectory represented corresponds to the run with the median localization percentage among the three test runs. Trajectories are given with an unknown scale, and in a random coordinate system. They are aligned by a Umeyama $Sim(3)$ alignment with respect to the most complete trajectory, namely the one given by ORB-SLAM3.

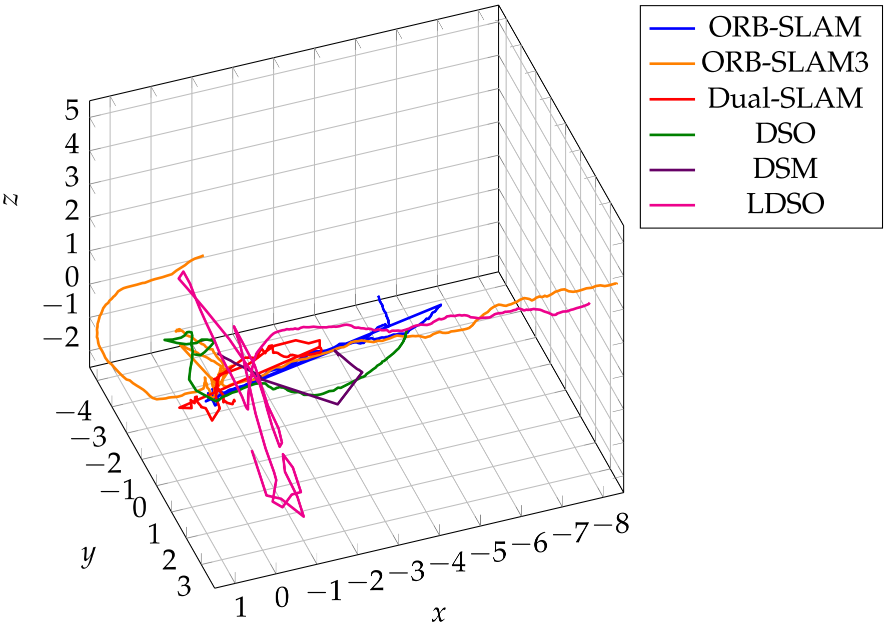


FIGURE E.1: Trajectories estimated by all evaluated SLAM systems on the *Bus* dataset.

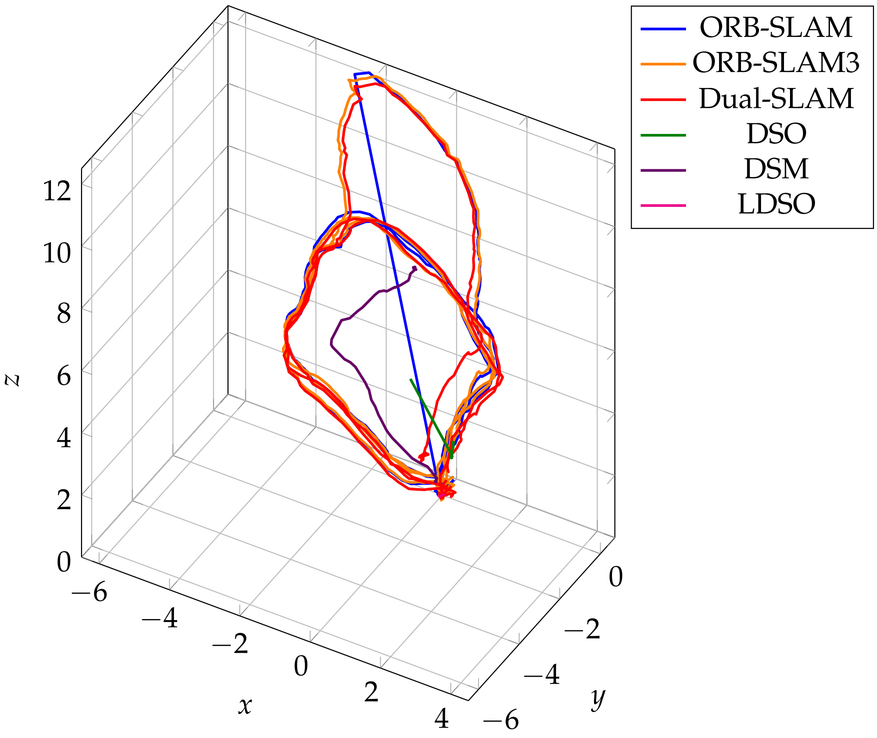


FIGURE E.2: Trajectories estimated by all evaluated SLAM systems on the *Cave* dataset.

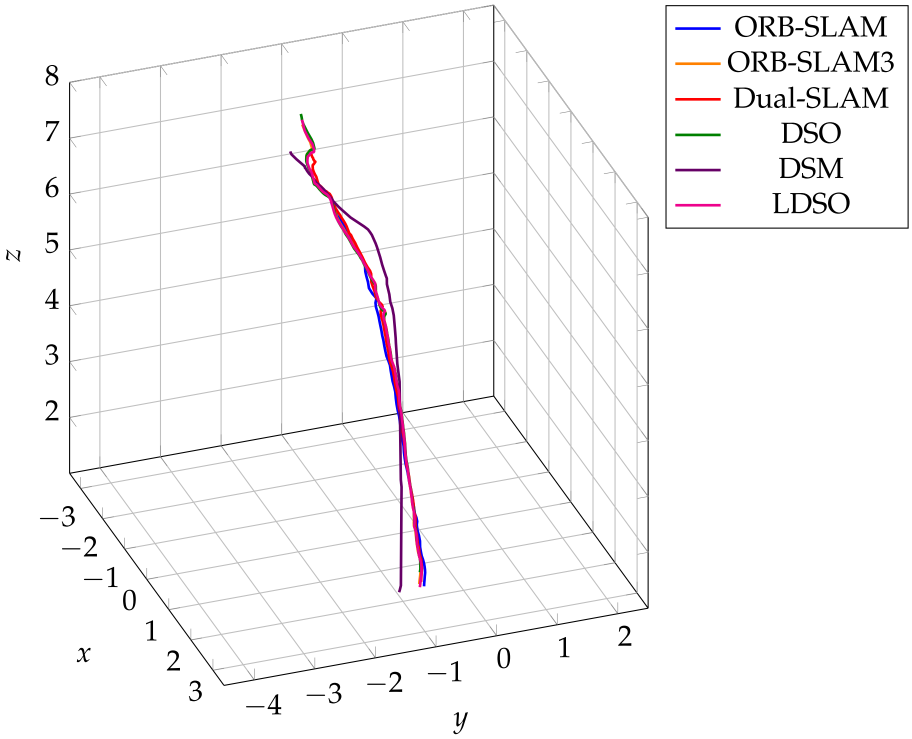


FIGURE E.3: Trajectories estimated by all evaluated SLAM systems on the *A/In* dataset.

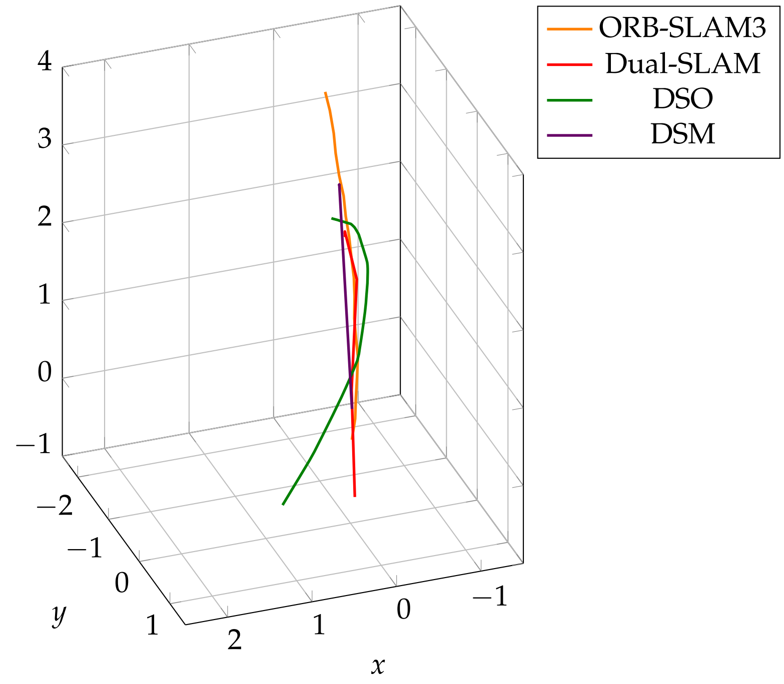


FIGURE E.4: Trajectories estimated by all evaluated SLAM systems on the *A/Out* dataset. ORB-SLAM, DSM and LDSO fail.

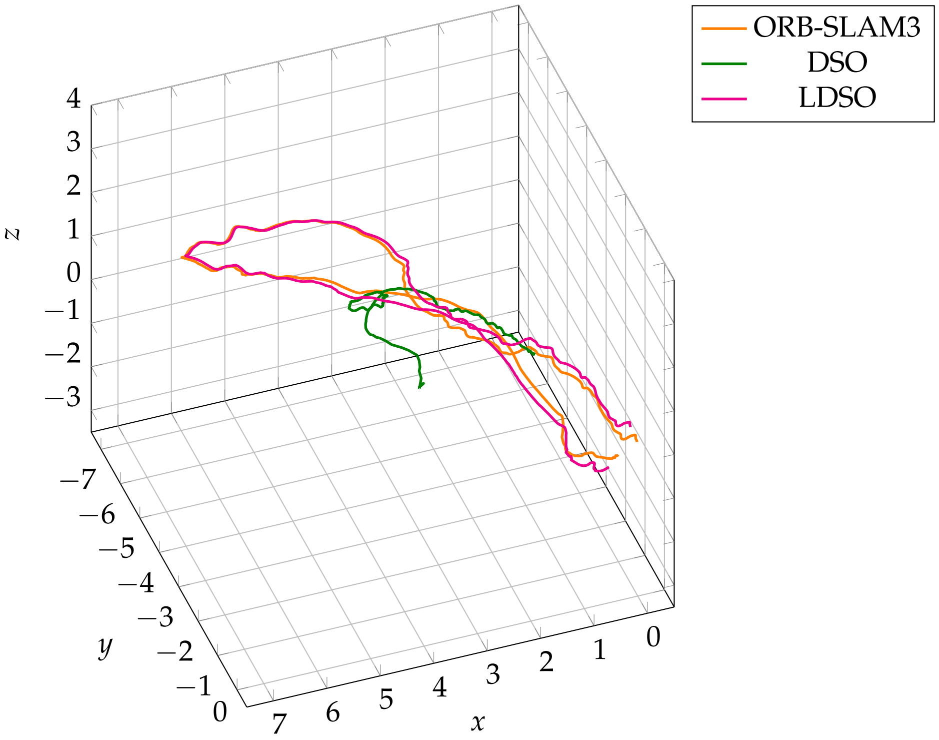


FIGURE E.5: Trajectories estimated by all evaluated SLAM systems on the *Aqualoc Harbor* dataset. DSM fails to initialize. ORB-SLAM and Dual-SLAM are not evaluated on this dataset because they do not support fisheye cameras.

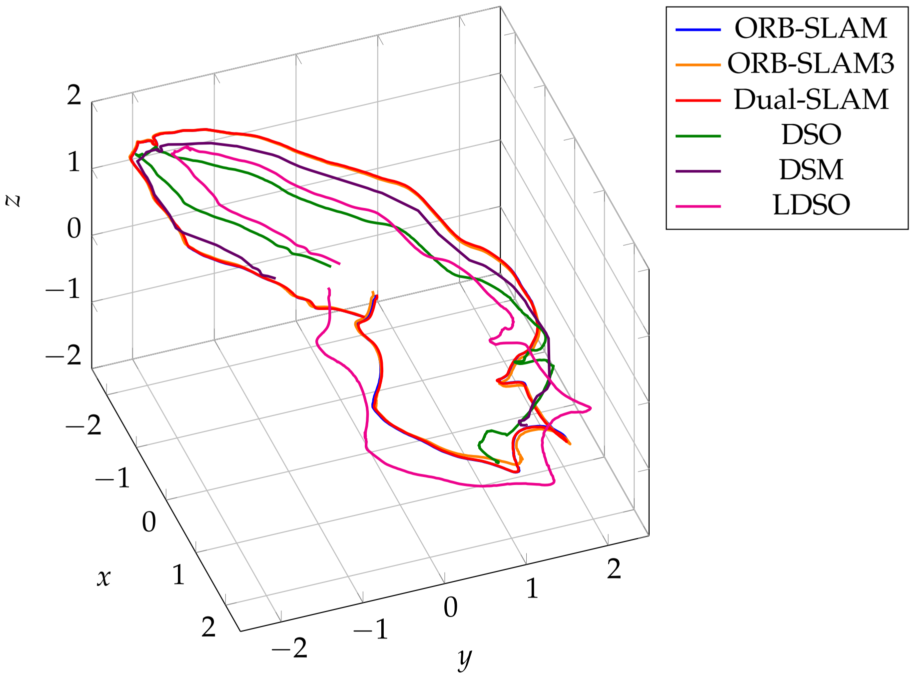


FIGURE E.6: Trajectories estimated by all evaluated SLAM systems on the *Aqualoc Archaeo* dataset.

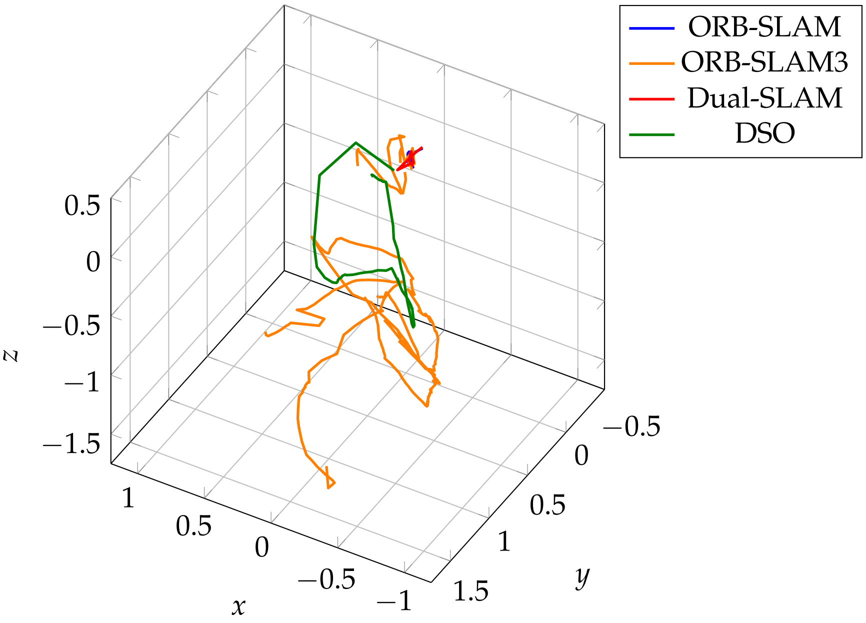


FIGURE E.7: Trajectories estimated by all evaluated SLAM systems on the *Cephismer* dataset.

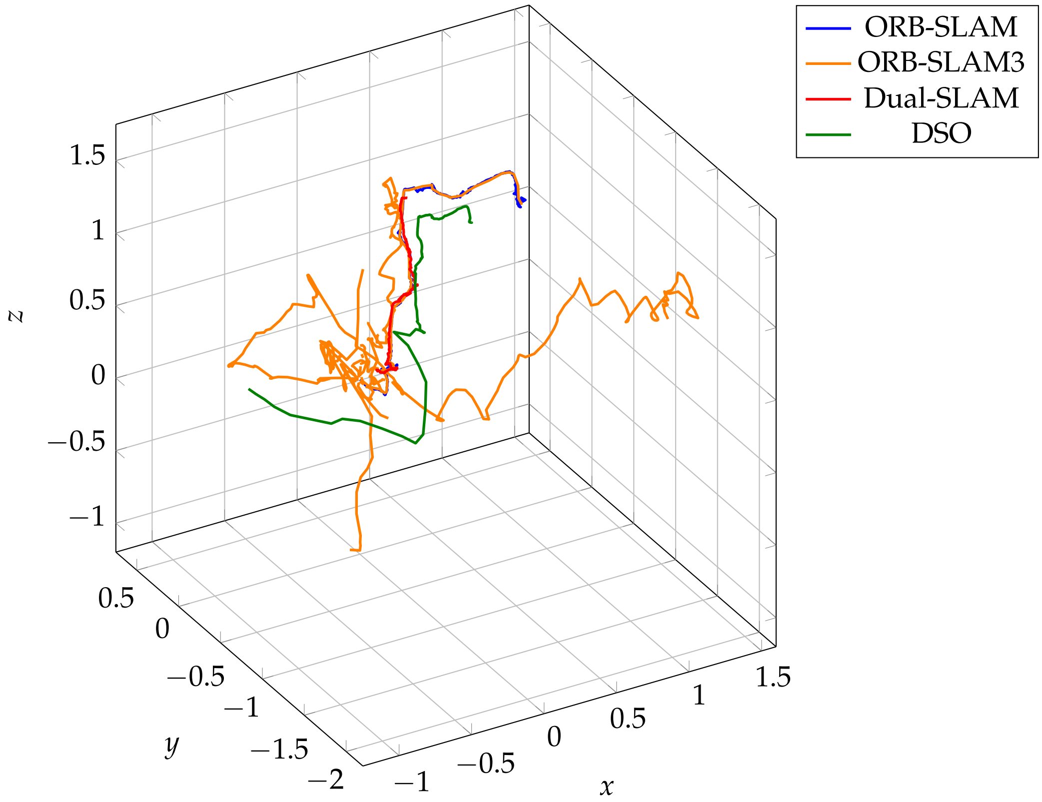


FIGURE E.8: Trajectories estimated by all evaluated SLAM systems on the *St-Raphael* dataset.

# Appendix F

# Résumé en français

## F.1 Introduction

Les opérations sous-marines représentent un défi tant pour l'exploration scientifique que pour les applications civiles et militaires. Ces missions ne peuvent cependant pas toujours être réalisées par des humains, pour des raisons physiques où liées à leur dangerosité, ce qui a conduit à l'essors de la robotique sous-marine. Les robots sous-marins sont toutefois, eux aussi, confrontés à d'importants défis. En raison de l'absorption rapide des ondes électromagnétiques dans un rayon de quelques mètres, la communication en temps réel entre un véhicule sous-marin et un opérateur en surface est limitée. Ces contraintes ont conduit à deux stratégies opposées dans la conception des robots sous-marins, qui correspondent à deux familles de véhicules :

- les véhicules sous-marins autonomes (*Autonomous Underwater Vehicles* ou AUVs), conçus pour fonctionner de manière totalement autonome ;
- les véhicules téléopérés (*Remotely Operated Vehicles* ou ROVs), reliés par câble à une station de surface, afin d'être téléopérés.

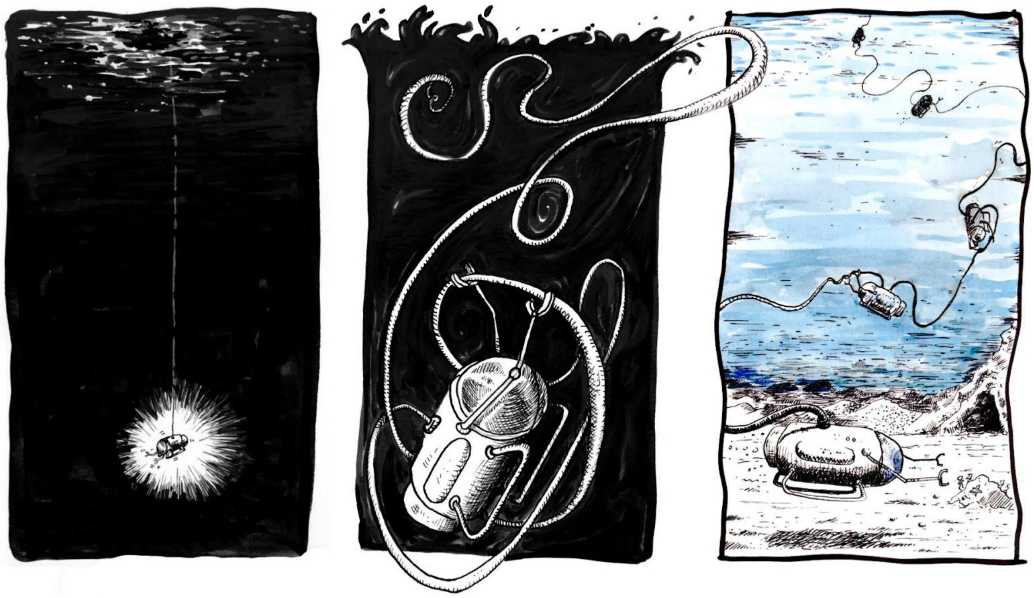


FIGURE F.1: Vue d'artiste d'une cordée de robots sous-marins, Pierre Straumann

La gestion des câbles des ROV est l'un des axes de recherche du laboratoire COSMER, Université de Toulon, France, et a conduit à l'introduction du concept de cordée de robots sous-marins, où la forme globale du câble est contrôlée par un ensemble de robots intermédiaires placés le long de celui-ci, et dont une vue d'artiste est présentée en Figure F.1. Ce concept a été initialement introduit dans le cadre du travail de doctorat de Matheus Laranjeira (Laranjeira et al., 2017; Laranjeira et al., 2018; Laranjeira et al., 2019; Laranjeira, 2019; Laranjeira et al., 2020). Dans la continuité de ces travaux, cette thèse porte sur la localisation d'une cordée de ROVs à l'aide de capteurs embarqués, et a été préparée au sein du laboratoire COSMER en collaboration avec le CNR-I3S, Sophia-Antipolis, France.

Cette problématique de localization d'une cordée de robots sous-marins a été envisagée à travers deux axes :

- une localisation proprioceptive qui exploite une estimation de la conformation tri-dimensionnelle du câble, dont peut être déduit le positionnement des robots encordés ;
- une localisation extéroceptive et multi-agents de la cordée par rapport à son environnement, en utilisant des techniques visuelles de localisation et cartographie simultanées.

Les contributions de cette thèse sont les suivantes:

- l'évaluation expérimentale du modèle de chaînette appliqué à des câbles coulants, immergés ;
- une méthode d'estimation de la forme d'un câble coulant reliant deux robots sous-marins à partir du modèle de chaînette et de mesures inertielles ;
- une méthode de localisation d'un ROV à partir de la forme de son câble, dans le cadre d'une collaboration avec Christophe Viel du CNRS LabSTICC, ENSTA Bretagne ;
- une évaluation comparative de méthodes visuelles de localisation et cartographie simultanées en milieu sous-marin ;
- un nouvel algorithme de localisation et cartographie simultanées multi-agent, à partir de données visuelles, son évaluation et sa comparaison à des algorithmes de l'état de l'art sur des jeux de données sous-marins acquis dans le cadre de cette thèse.

Ces contributions ont mené aux publications listées ci-après :

**Articles de journaux :**

- ROV localization based on umbilical angle measurement.
  C. Viel, J. Drupt, C. Dune, V. Hugel,
  Ocean Engineering, Volume 269, 2023, 113570, ISSN 0029-8018.

**Articles de congrés :**

- Inertial-measurement-based catenary shape estimation of underwater cables for tethered robots.
  J. Drupt, C. Dune, A. I. Comport, S. Seillier and V. Hugel,

2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Kyoto, Japan, 2022, pp. 6867-6872.
Vidéo 1 : `https://www.youtube.com/watch?v=TKLLVTSUN8s`
Vidéo 2 : `https://www.youtube.com/watch?v=x-SXut75vHk`

- Estimation de forme de câble pesant pour la localisation de robots sous-marins encordés : comparaison d'une approche visuelle à une nouvelle approche inertielle.
  J. Drupt, C. Dune, A. I. Comport and V. Hugel,
  ORASIS 2023, Carqueiranne, France, 2023.
- An augmented catenary model for underwater tethered robots.
  M. Filliung, J. Drupt, C. Peraud, C. Dune, N. Boizot, A. I. Comport and V. Hugel,
  Soumis à 2024 IEEE International Conference on Robotics and Automation (ICRA).

***Workshops* :**

- "Validity of the catenary model for moving submarine cables with negative buoyancy.
  J. Drupt, C. Dune, A. I. Comport and V. Hugel,
  3rd workshop on RObotic MAnipulation of Deformable Objects: challenges in perception, planning and control for Soft Interaction (ROMADO-SI), Best Paper Award, Kyoto, Japan, 2022.
  Vidéo : `https://www.youtube.com/watch?v=kmcs9xKf3KQ`

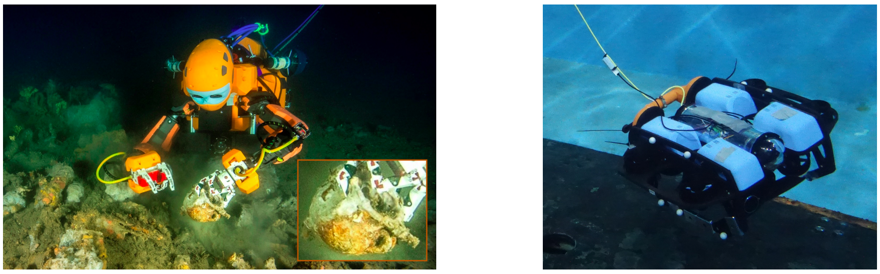
## F.2 État de l'art

Cette thèse s'intéresse à la localisation d'une cordée de ROVs à l'aide de capteurs embarqués. Ce sujet implique à la fois d'estimer la configuration de la cordée, à savoir la localisation relative des robots et la forme des portions de câble qui les relient, mais aussi le positionnement de la cordée par rapport à son environnement, et enfin, éventuellement, sa localisation géo-référencée.

### F.2.1 Robots et câbles sous-marins

Les robots sous-marins (*Unmanned Underwater Vehicles*, ou UUVs) peuvent être classés en trois catégories : ROVs, AUVs, et ROVs hybride (*Hybrid ROVs* or HROVs), ces dernier pouvant être utilisés comme des ROVs ou AUVs selon le contexte opérationnel (Christ et al., 2014; Creuze, 2014; Brignone et al., 2015). La Figure F.2 présente quelques exemples d'UUVs. Les ROVs sont utilisés pour toutes les missions qui ne peuvent pas être complètement automatisées ou qui présentent des enjeux importants en termes de sûreté de fonctionnement.

La présence du câble est néanmoins lourde de conséquences sur les ROVs : il risque en effet de s'emmêler lors de la plongée ou dans les obstacles environnants, et exerce une gêne mécanique. Des méthodes ont donc été développées pour limiter l'impact du câble sur le ROV (Christ et al., 2014), incluant des méthodes passives impliquant l'ajout de masses ou de bouées sur le câble, et des méthodes actives mettant en scène des systèmes mécatroniques permettant de réguler la longueur et parfois la forme du câble. En particulier, le contrôle de la forme du câble par l'ajout de robots intermédiaires dédiés le long de celui-ci a rencontré un intérêt croissant ces dernières années sous le nom de cordée de robots sous-marins (Laranjeira et al., 2020).



(A) Ocean One (ROV), Université de Stanford. Source : (Khatib et al., 2016)

(B) BlueROV2 (ROV), Blue Robotics

(C) UlyX (AUV), Ifremer. Source : Ifremer

(D) Seasam (HROV), Delair Marine. Source : Delair Marine

FIGURE F.2: Exemples d'UUVs

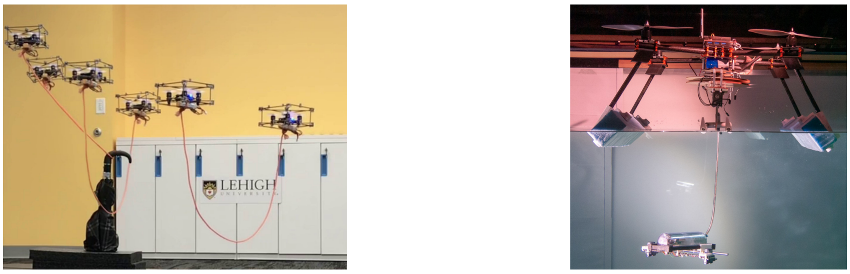
Le GPS étant inutilisable sous l'eau, la localisation des UUVs repose essentiellement sur leurs capteurs embarqués. Les capteurs les plus répandus en robotique sous-marine incluent les capteurs acoustiques, les centrales inertielles (*Inertial Measurement Units*, ou IMUs), les capteurs de pression et les caméras (Creuze, 2014; Cong et al., 2021; Massot-Campos et al., 2015). Ces trois derniers types de capteurs présentent habituellement des prix et des dimensions significativement moins élevés que les capteurs acoustiques, et sont donc présents sur l'immense majorité des UUVs, dont les modèles les plus petits et abordables. Les capteurs extéroceptifs usuels sont les SONARs et les caméras. Outre leurs prix et dimensions attractifs, les caméras sont particulièrement intéressantes dans des environnements confinés, où les SONARs subissent des effets de réverbération du signal émis qui bruitent leur mesure. La perception visuelle en milieu sous-marin est néanmoins compliquée par l'absorption sélective des rayons lumineux par l'eau, les phénomènes de réfraction sur les caissons étanches qui habritent les capteurs, mais aussi des phenomènes de diffusion, par la turbidité de l'eau, ou encore l'impact d'un éventuel éclairage embarqué sur l'aspect visuel des alentours (Wang et al., 2019).

### F.2.2 Estimation d'état pour les systèmes robotiques encordés

Les systèmes robotiques encordés sont présents à la fois en robotique marine et sous-marine, terrestre et aérienne (Laranjeira et al., 2020; D'Antonio et al., 2021; McGarey et al., 2016; Debruyn et al., 2020). Leur déploiement requiert une estimation de leur état, dont la localisation des câbles et des robots. L'estimation de la forme du câble peut être réalisée à partir de différents modèles et capteurs. La Figure F.3 présente quelques exemples de systèmes encordés.

(A) ROV. Source : Ifremer

(B) *Tethered Robotic Explorer* (TRex).
Source : (McGarey et al., 2017)



(C) Transport d'objets déformables.
Source : (D'Antonio et al., 2021)

(D) Système hybride air-eau.
Source : (Debruyn et al., 2020)

FIGURE F.3: Exemples de systèmes robotique encordés

Les modèles de câbles peuvent être classés en deux catégories principales : les modèles dynamiques (Hong et al., 2020; Meng et al., 2020) et les modèles quasi-statiques (D'Antonio et al., 2021; Viel, 2022b). Les modèles dynamiques sont les plus complets et sont dérivés des forces qui s'appliquent au câble, compte tenu de ses propriétés physiques. Les modèles quasi-statiques sont toutefois plus simples, et généralement préférés pour estimer la forme d'un câble en temps réel. Le modèle quasi-statique le plus répandu est le modèle caténaire (Laranjeira et al., 2020; D'Antonio et al., 2021), qui correspond à la forme d'un câble suspendu, homogène, non-élastique uniquement soumis à son propre poids, avec ses extrémités fixes. Cette définition peut être étendue à tout câble soumis à une force verticale proportionnelle à sa masse linéique, comme un câble dans un liquide, également soumis à la poussée d'Archimède.

Certains câbles en fibre optique ou recouverts d'IMUs sont conçus comme des capteurs proprioceptifs, capables de mesurer leur forme (Yu et al., 2013; Frank et al., 2013). Ceux-ci sont néanmoins assez rares en pratique. La forme d'un câble peut également être estimée en associant un modèle et des mesures locales, nécessitant une instrumentation légère du câble par des IMUs, capteurs de courbure ou encore capteurs d'angle ou de tension (McGarey et al., 2017; Merlet, 2018; Tortorici et al., 2023). Les mesures peuvent également provenir d'un capteur extérieur au câble, par exemple une caméra (Laranjeira et al., 2020; Jin et al., 2022; Smolentsev et al., 2023).

Enfin, certains travaux utilisent la forme du câble pour déduire la localisation des robots encordés, et vont jusqu'à utiliser les emmêlements du câble dans les obstacles environnants pour en réaliser une cartographie (McGarey et al., 2017). Dans un tel contexte, le câble n'est plus une source de gène mais bien un atout pour l'estimation d'état du système global.

### F.2.3 SLAM visuel monoculaire en milieu sous-marin

La localisation d'un système robotique par rapport aux obstacles d'un environment initialement inconnu nécessite une cartographie en ligne de celui-ci, et un calcul de localisation au sein de la carte simultanément construite. Ce problème est connu sous le non de SLAM (*Simultaneous Localization and mapping*). Les algorithmes SLAM sous-marins reposent généralement sur la fusion de plusieurs capteurs. Toutefois, l'omniprésence de caméras monoculaires, même sur les systèmes les plus petits et abordable, couplée à la richesse des informations visuelles dans la perception tri-dimensionnelle conduisent à s'intéresser au SLAM visuel (*Visual SLAM*, ou VSLAM) pour la localisation et cartographie simultanée en milieu sous-marin. Les algorithmes de VSLAM utilisent uniquement une perception visuelle pour résoudre le problème du SLAM. Étant donné que les robots les plus légers embarquent une unique caméra monoculaire, on s'intéressera plus spécifiquement au VSLAM monoculaire.

Alors que de nombreux travaux s'intéressent au VSLAM aérien, la problème du VSLAM sous-marin est moins étudié. En outre, la perception visuelle sous-marine reste très différente de son pendant aérien de part l'absorption des couleurs, l'effet des éclairages embarqués, les phénomènes de réfraction et de diffusion, la présence de particules en suspension ou encore la présence d'environnements très peu structurés. Des études (Joshi et al., 2019) montrent cependant que, parmi les algorithmes de VSLAM monoculaires de l'état de l'art en libre accès, ORB-SLAM (Mur-Artal et al., 2015) et DSO (Engel et al., 2018) sont les seules travaux présentant une certaine robustesse à cet environnement, sous certaines limites.

Les algorithmes de VSLAM peuvent être classées selon plusieurs critères :

- les approches directes utilisent les caractéristiques photométrique des images, tandis que les approches indirectes utilisent des caractéristiques de plus haut niveau ;
- les méthodes denses utilisent tous les points de l'image, alors que les méthodes éparses ne s'appuient que sur une sélection de points ;
- le traitement des fermetures de boucles, les méthodes dépourvues de cette fonctionnalité étant communément appelées 'odométries visuelles' ;
- la présence d'un module de récupération, qui permet de continuer à calculer une localisation et construire une carte malgré un échec du calcul de pose courante.

ORB-SLAM (Mur-Artal et al., 2015) est une méthode indirecte, éparse, qui implémente la fermeture de boucles et une stratégie de récupération par une relocalisation dans la carte, qui s'avère cependant souvent inefficace lors des tests sous-marins (Joshi et al., 2019), et devient inutile si le système sort de la zone déjà cartographiée. Des travaux récents proposent une récupération par l'initialisation d'une nouvelle carte, qui sera fusionnée par la suite avec la précédente (Daoud et al., 2018; Elvira et al., 2019; Huang et al., 2020). DSO (Engel et al., 2018) est une odométrie visuelle directe et éparse. Sous l'eau, DSO produit parfois des estimations incohérentes qui ne sont jamais corrigées au cours du temps faute d'association de données à long-terme, conduisant à des résultats parfois incohérents. Des travaux récents se sont cependant intéressés à l'intégration d'association de données à long-terme dans DSO (Gao et al., 2018; Zubizarreta et al., 2020), qui s'apparentent à une détection et gestion de fermeture de boucle. Ces développements reposant sur ORB-SLAM (Mur-Artal et al., 2015) et DSO (Engel et al., 2018) pourraient donc présenter une meilleure robustesse au contexte sous-marin.

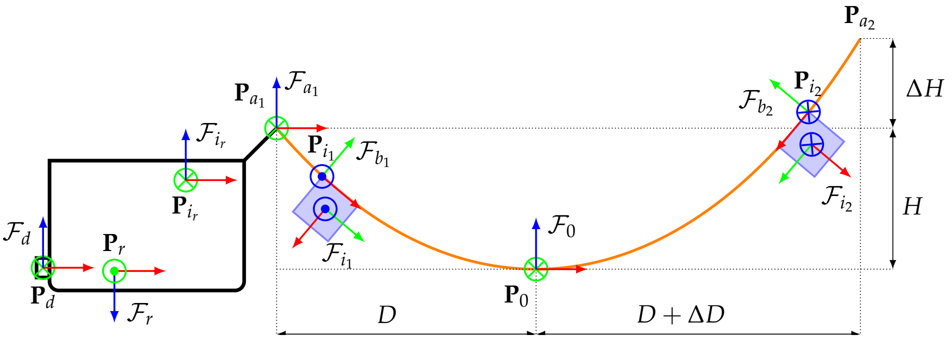
### F.2.4 SLAM multi-agent en milieu sous-marin

Certaines applications requièrent le déploiement de plusieurs robots simultanément (Cho et al., 2018; Laranjeira et al., 2020; Özkahraman et al., 2022). Il est alors nécessaire de les localiser l'un par rapport à l'autre, mais aussi par rapport à leur environnement, menant à un problème de SLAM multi-agents. Les méthodes de SLAM visuel multi-agent les plus récentes d'appuient sur ORB-SLAM (Mur-Artal et al., 2015) et effectuent tout ou une partie des calculs sur un serveur central, avec lequel les agents communiquent (Li et al., 2018; Daoud et al., 2018; Schmuck et al., 2019). Ce schéma de communication peut être mis en œuvre en milieu sous-marin pour un système multi-ROV.

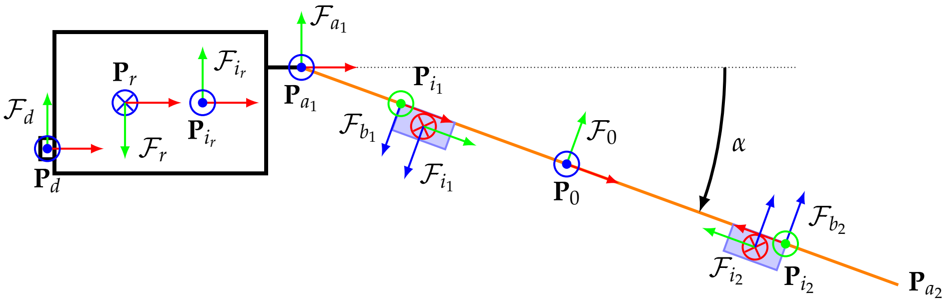
## F.3 Estimation d'état d'un système robotique sous-marin encordé à partir de la forme prise par son câble

### F.3.1 Estimation de forme de câble à partir du modèle de chaînette

L'état d'un système robotique encordé peut être caractérisé par une estimation de la forme de son câble. Un câble coulant reliant deux robots sous-marins encordés peut-être modélisé par une chaînette.



(A) Vue de côté



(B) Vue du dessus

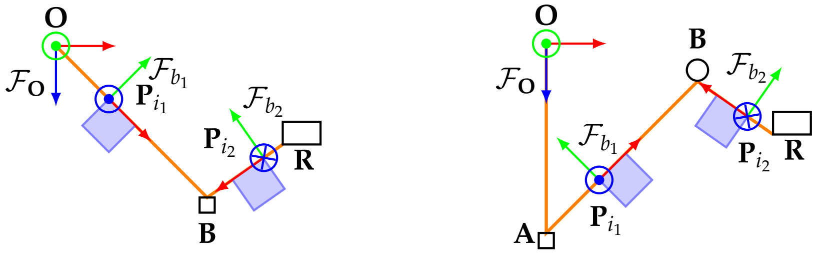
FIGURE F.4: Système robotique et paramétrage considérés

La validité de cette modélisation est vérifiée expérimentalement en conditions dynamiques, montrant que la modélisation d'une chaînette est adaptée à un câble suffisamment lourd, en l'absence de changement brusques dans la direction du mouvement de ses extrémités. Conformément à ces conclusions, le modèle de chaînette est ensuite utilisé pour estimer la forme d'un câble coulant reliant deux robots sous-marins consécutifs, au sein d'une cordée. Le système robotique considéré et le paramétrage de la forme de chaînette sont indiqués sur la Figure F.4. La chaînette est entièrement caractérisée par les paramètres $\{H, \Delta H, \alpha\}$. Ces paramètres sont mesurés expérimentalement à partir des capteurs embarqués : une ou deux IMUs fixées près des extrémités du câble ainsi qu'un capteur de pression et un IMU supplémentaire, placés dans le robot.

Cette méthode d'estimation de forme de câble est validée expérimentalement, d'abord dans l'air, puis en bassin, et comparée à une méthode visuelle issue de précédents travaux (Laranjeira et al., 2020). La méthode proposée démontre de meilleures robustesse et précision quant à l'estimation des paramètres de chaînette, avec une erreur d'environ 5 cm sur $H$ et 12° sur $\alpha$ pour un câble de 1.5 m.

### F.3.2 Localisation d'un ROV à partir de son câble conformé en zigzags

En collaboration avec Christophe Viel du CNRS LabSTICC, ENSTA Bretagne, France, une méthode de localisation d'un ROV à partir d'une caractérisation de la forme de sont câble est développée. Le câble est contraint en forme de zigzags par l'adjunction de bouées ou de masses libres de coulisser le long du câble. Les paramètres du modèle établi par Christophe Viel sont estimés à l'aide de capteurs embarqués consistant en deux IMUs fixées près des extrémités du câble ainsi qu'un capteur de pression et un IMU supplémentaire, placés dans le robot. Le conformation du câble et son instrumentation sont illustrés par la Figure F.5.



(A) Masse glissante (B) Bouée glissante

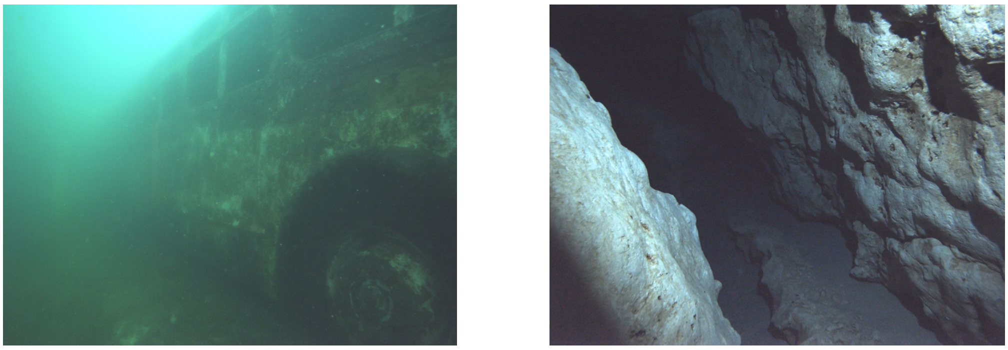
FIGURE F.5: Câble en zigzags équipé d'IMUs

La méthode de localisation proposée est validée expérimentalement en bassin pou un ROV relié à un point fixe et pour une paire de ROV en cordée, démontrant une précision de l'ordre d'une quinzaine de centimètre pour un câble de 3 m.

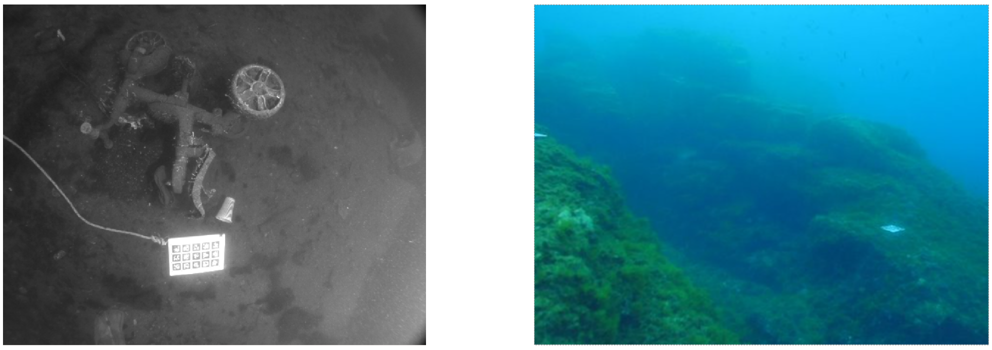
## F.4 Localisation visuelle pour un système robotique sous-marin

### F.4.1 Évaluation d'algorithmes de VSLAM en milieu sous-marin

Six algorithmes de VSLAM en libre accès s'appuyant sur ORB-SLAM (Mur-Artal et al., 2015) et DSO (Engel et al., 2018) sont évalués et comparés sur huit séquences visuelles sous-marines présentant des environnements et conditions visuelles variées. Les six algorithmes évalués sont ORB-SLAM (Mur-Artal et al., 2015), ORB-SLAMM (Daoud et al., 2018), Dual-SLAM (Huang et al., 2020), DSO (Engel et al., 2018), LDSO (Gao et al., 2018) et DSM (Zubizarreta et al., 2020), où ORB-SLAMM et Dual-SLAM proposent des stratégies de récupération multi-cartes sur une architecture similaire à celle d'ORB-SLAM et où LDSO et DSM introduisent des associations de données sur le long terme dans un algorithme similaire à celui de DSO. La Figure F.6 donne un aperçu de quatre séquences test parmi les huit utilisées.



(A) *Bus* (Joshi et al., 2019) (B) *Cave* (Joshi et al., 2019)



(C) *AQUALOC Harbor #01* (Ferrera et al., 2019) (D) *St-Raphael*

FIGURE F.6: Vues extraites de quatre séquences test parmi les huit

Cette évaluation montre que les méthodes développées à partir de DSO sont très peu robustes aux conditions visuelles sous-marines, produisant des résultats souvent incohérents du fait de mauvaises associations de données sur le long terme. Si Dual-SLAM ne semble pas présenter de meilleures performances qu'ORB-SLAM, ORB-SLAM Atlas présente une robustesse accrue particulièrement intéressante de par son algorithme de récupération, permettant de poursuivre la localisation et la cartographie malgré des conditions visuelles particulièrement compliquée. La principale limite de cet algorithme reste toutefois de ne pas toujours réussir à fusionner toutes les cartes locales qu'il crée.

### F.4.2 MAM$^3$SLAM: VSLAM multi-agent robuste au milieu sous-marin

Étant donnée la robustesse d'ORB-SLAM Atlas (Elvira et al., 2019) aux conditions de visibilités sous-marines relevée lors des précédents tests, un nouvel algorithme de VSLAM multi-agent centralisé est construit à partir d'ORB-SLAM Atlas afin de pouvoir être appliqué à des scénarios mettant en scène plusieurs ROVs opérant simultanément sur le même lieu. La Figure F.7 donne une représentation schématisée du principe de ce nouvel algorithme, nommé MAM$^3$SLAM Plus précisément, MAM$^3$SLAM s'appuie sur l'implémentation d'ORB-SLAM Atlas publiée dans la librairie ORB-SLAM3 (Campos et al., 2021). Pour des raisons de lisibilité, la Figure F.7 comporte quelques abréviations : 'proc.' pour 'processus', 'optim.' pour 'optimisation', 'KF' pour 'trame-clef' ('*KeyFrame*'), 'MP' pour 'point de la carte' ('*Map point*'), 'BD' pour 'base de données' et 'covis.' pour 'covisibilité'.

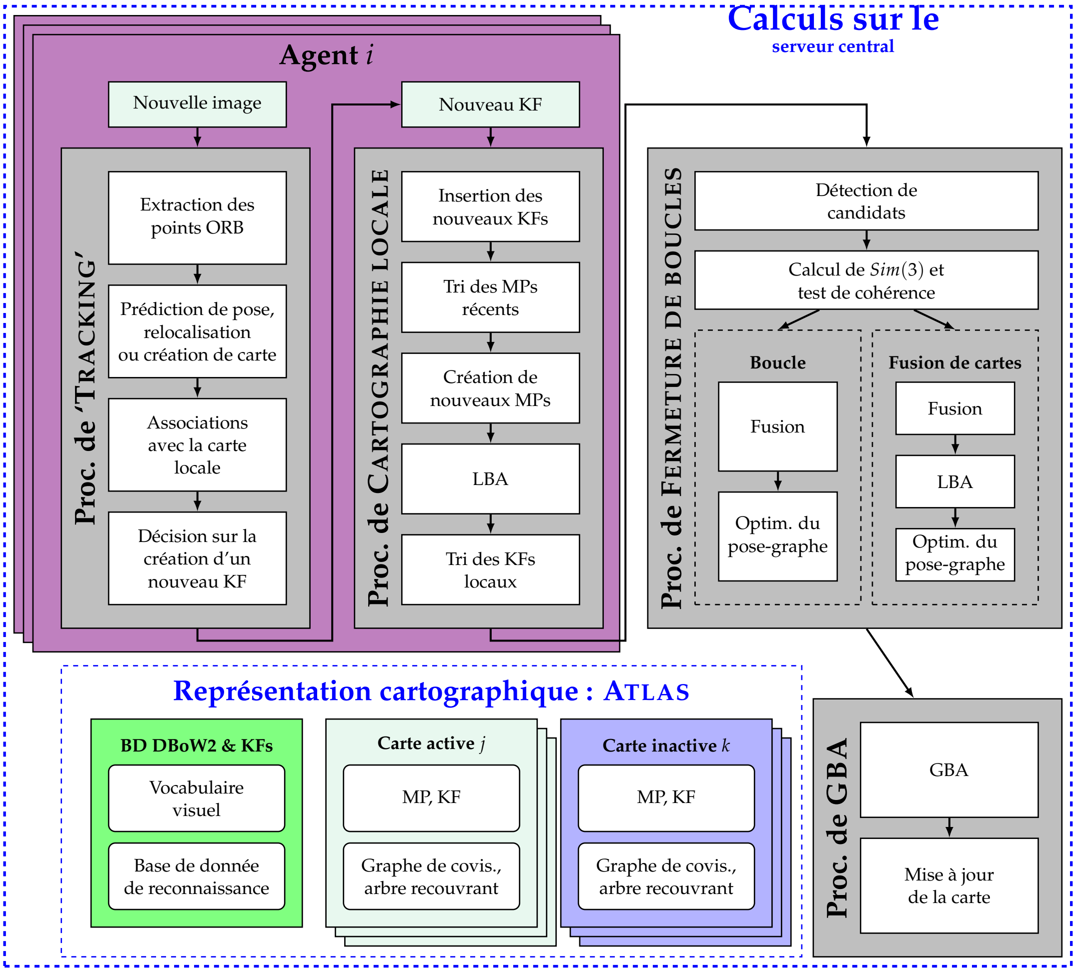


FIGURE F.7: Schéma de l'algorithme de MAM$^3$SLAM

MAM$^3$SLAM est évalué et comparé aux algorithmes de l'état de l'art ORB-SLAMM (Daoud et al., 2018) et CCM-SLAM (Schmuck et al., 2019) sur quatre séquences à deux agents, dont une aérienne issue du jeux de données EuRoc (Burri et al., 2016) et trois nouvelles séquences sous-marines acquises dans le cadre de ce travail de validation. MAM$^3$SLAM montre de meilleures robustesse et précision que les approches concurrentes, avec une erreur moyenne de l'ordre du centimètre sur trois des quatre séquences test. Bien qu'un travail logiciel sur l'algorithme plus approfondi reste nécessaire pour envisager l'extension de MAM$^3$SLAM à plus de deux agents simultanés, MAM$^3$SLAM semble donc une solution de localisation et cartographie simultanées prometteuse pour des systèmes multi-ROVs, dont font partie les cordées de robots sous-marins.

## F.5 Conclusion générale et perspectives

La Tableau F.1 récapitule les avantages et inconvénients de l'estimation d'état à partir du câble et du VSLAM monoculaire pour la localisation d'une cordée de robots sous-marins.

TABLE F.1: Avantages et inconvénients de l'estimation d'état à partir du câble et du VSLAM monoculaire

| | **Avantages** | **Inconvénients** |
|---|---|---|
| Câble | • Toujours disponible, même en pleine eau où le fond marin ne peut être observé<br>• Beaucoup plus robuste que le VSLAM<br>• Facteur d'échelle connu<br>• Peut donner la position des ROVs par rapport à un point de surface de coordonnées GPS connues<br>• Donne directement une position relative inter-agents | • Précision médiocre par rapport au VSLAM, environ 10 à 20 cm<br>• Échec si le câble est significativement déformé par rapport au modèle |
| VSLAM monoculaire | • Localisation individuelle et relative précise jusqu'à quelques centimètres<br>• Cartographie en ligne | • Indisponible en pleine eau<br>• Risque d'échec du SLAM en cas de mouvements rapides ou dans des environnements peu texturés<br>• Facteur d'échelle inconnu<br>• Nécessite une reconnaissance de lieux pour fournir une position relative inter-agents<br>• Ne donne pas la forme du câble |

Localisation par câble et le VSLAM monoculaire peuvent donc être complémentaires pour localiser une cordée de robots sous-marins par rapport à son environnement.

- Les deux approches peuvent donc fournir des informations différentes sur l'état du système. En effet, le câble peut donner une position géo-référencée alors que le VSLAM ne le peut pas, et le VSLAM calcule une cartographie en ligne de l'environnement, ce qui n'a pas été envisagé en utilisant le câble afin d'éviter que le câble ne se coince.

- Elles ne sont pas efficaces dans les mêmes conditions. Si le robot se déplace suffisamment lentement dans un environnement texturé, le VSLAM pourra donner une estimation de localisation beaucoup plus précise que la localisation par câble. Inversement, le câble peut permettre de localiser les robots dans des conditions qui ne permettent pas le fonctionnement du VSLAM (pleine eau ou conditions visuelles trop difficiles), bien qu'avec une précision moindre.

- De manière générale, la localisation par câble et le VSLAM n'ont pas les mêmes causes d'échec.

- Le VSLAM ne donne aucune information sur l'état du câble, à l'exception de ses points d'attache sur les robots. Une connaissance de la forme du câble est donc nécessaire, en utilisant soit un câble proprioceptif, soit un modèle dérivé de paramètres physiques mesurables.

De futurs travaux pourraient s'intéresser à une fusion entre la localisation par câble et le VSLAM monoculaire. Une première option consisterait à utiliser la localisation par câble uniquement pour les fonctionnalités que le VSLAM ne met pas en œuvre, notamment la localisation en pleine eau, ou géo-référencé, l'estimation de l'échelle de l'environnement cartographié et des mouvements, la localisation relative entre les robots avant que le VSLAM ne les localise sur la même carte et, éventuellement, l'estimation de la forme du câble si celle-ci ne peut être déduite directement de la pose relative des robots. La localisation par câble peut également s'avérer une solution de secours en cas de défaillance du VSLAM. Une deuxième option pourrait être de fusionner plus étroitement l'estimation d'état par câbles et le VSLAM. Par exemple, une estimation initiale pour la fusion des cartes inter-agents pourrait être calculée à partir de leur position relative estimée à l'aide d'une mesure de forme du câble, et la localisation par câble câble pourrait aider à la récupération en cas d'échec du SLAM. Inversement, la localisation du robot par VSLAM pourrait permettre de détecter si le câble sort du modèle, permettant, par exemple, de détecter les blocages du câble ou de mesurer les courants. Par conséquent, de nombreux travaux sont envisageable dans le cadre de la poursuite de l'axe de recherche de la présente thèse.

# Bibliography

Akkaynak, Derya and Tali Treibitz (2019). “Sea-Thru: A Method for Removing Water From Underwater Images”. In: *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1682–1691. DOI: `10.1109/CVPR.2019.00178`.

Alam, Khairul, Tapabrata Ray, and Sreenatha G. Anavatti (2014). “Design and construction of an autonomous underwater vehicle”. In: *Neurocomputing* 142. SI Computational Intelligence Techniques for New Product Development, pp. 16–29. ISSN: 0925-2312. DOI: `https://doi.org/10.1016/j.neucom.2013.12.055`. URL: `https://www.sciencedirect.com/science/article/pii/S0925231214005219`.

Alonso-Mora, Javier, Ross Knepper, Roland Siegwart, and Daniela Rus (2015). “Local motion planning for collaborative multi-robot manipulation of deformable objects”. In: *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 5495–5502. DOI: `10.1109/ICRA.2015.7139967`.

Bechlioulis, Charalampos P., Fotis Giagkas, George C. Karras, and Kostas J. Kyriakopoulos (2019). “Robust Formation Control for Multiple Underwater Vehicles”. In: *Frontiers in Robotics and AI* 6. ISSN: 2296-9144. DOI: `10.3389/frobt.2019.00090`. URL: `https://www.frontiersin.org/articles/10.3389/frobt.2019.00090`.

Bellavia, Fabio, Marco Fanfani, and Carlo Colombo (Dec. 2015). “Selective visual odometry for accurate AUV localization”. In: *Autonomous Robots* 41.1, pp. 133–143. DOI: `10.1007/s10514-015-9541-1`. URL: `https://doi.org/10.1007/s10514-015-9541-1`.

Boittiaux, Clémentin, Ricard Marxer, Claire Dune, Aurélien Arnaubec, Maxime Ferrera, and Vincent Hugel (2023). *SUCRe: Leveraging Scene Structure for Underwater Color Restoration*. arXiv: `2212.09129 [cs.CV]`.

Brignone, L., E. Raugel, J. Opderbecke, V. Rigaud, R. Piasco, and S. Ragot (2015). “First sea trials of HROV the new hybrid vehicle developed by IFREMER”. In: *OCEANS 2015 - Genova*, pp. 1–7. DOI: `10.1109/OCEANS-Genova.2015.7271682`.

Brizard, T. (2014). *Jet-pump-based autonomous underwater vehicle and method for coupling to ocean bottom during marine seismic survey*. US Patent 20140251199-A1.

Buckham, Bradley J. (1997). “Dynamics Modelling of Low-Tension Tethers for Submerged Remotely Operated Vehicles”. PhD thesis. Canada: University of Victoria.

Burri, Michael, Janosch Nikolic, Pascal Gohl, Thomas Schneider, Joern Rehder, Sammy Omari, Markus W Achtelik, and Roland Siegwart (2016). “The EuRoC micro aerial vehicle datasets”. In: *The International Journal of Robotics Research* 35.10, pp. 1157–1163. DOI: `10.1177/0278364915620033`. eprint: `https://doi.org/10.1177/0278364915620033`. URL: `https://doi.org/10.1177/0278364915620033`.

Campos, Carlos, Richard Elvira, Juan J. Gómez Rodríguez, José M. M. Montiel, and Juan D. Tardós (2021). “ORB-SLAM3: An Accurate Open-Source Library for Visual, Visual–Inertial, and Multimap SLAM”. In: *IEEE Transactions on Robotics* 37.6, pp. 1874–1890. DOI: `10.1109/TRO.2021.3075644`.

Carreras, Marc, Juan David Hernández, Eduard Vidal, Narcís Palomeras, David Ribas, and Pere Ridao (2018). "Sparus II AUV—A Hovering Vehicle for Seabed Inspection". In: *IEEE Journal of Oceanic Engineering* 43.2, pp. 344–355. DOI: 10.1109/JOE.2018.2792278.

Castle, Robert, Georg Klein, and David W. Murray (2008). "Video-rate localization in multiple maps for wearable augmented reality". In: *2008 12th IEEE International Symposium on Wearable Computers*, pp. 15–22. DOI: 10.1109/ISWC.2008.4911577.

Cheng, Feng, Benxin Chi, Nathaniel J. Lindsey, T. Craig Dawe, and Jonathan B. Ajo-Franklin (Mar. 2021). "Utilizing distributed acoustic sensing and ocean bottom fiber optic cables for submarine structural characterization". In: *Scientific Reports* 11.1. DOI: 10.1038/s41598-021-84845-y. URL: https://doi.org/10.1038/s41598-021-84845-y.

Cho, Younggun and Ayoung Kim (2018). "Channel invariant online visibility enhancement for visual SLAM in a turbid environment". In: *Journal of Field Robotics* 35.7, pp. 1080–1100. DOI: https://doi.org/10.1002/rob.21796. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/rob.21796. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/rob.21796.

Christ, Robert D and Robert L Wernli Sr (2014). "The ocean environment". In: *The ROV Manual*. Elsevier, pp. 21–52.

Cohen, Nadav and Itzik Klein (2022). "BeamsNet: A data-driven approach enhancing Doppler velocity log measurements for autonomous underwater vehicle navigation". In: *Engineering Applications of Artificial Intelligence* 114, p. 105216. ISSN: 0952-1976. DOI: https://doi.org/10.1016/j.engappai.2022.105216. URL: https://www.sciencedirect.com/science/article/pii/S0952197622003013.

Cong, Yang, Changjun Gu, Tao Zhang, and Yajun Gao (2021). "Underwater robot sensing technology: A survey". In: *Fundamental Research* 1.3, pp. 337–345. ISSN: 2667-3258. DOI: https://doi.org/10.1016/j.fmre.2021.03.002. URL: https://www.sciencedirect.com/science/article/pii/S2667325821000522.

Creuze, Vincent (2014). "Robots marins et sous-marins Perception, modélisation, commande". In: *Techniques de l'ingénieur Robotique* base documentaire : TIP661WEB.ref. article : s7783. fre. DOI: 10.51257/a-v1-s7783. URL: https://www.techniques-ingenieur.fr/base-documentaire/automatique-robotique-th16/applications-en-robotique-42623210/robots-marins-et-sous-marins-s7783/.

Crevier, Daniel (1993). "Hue-based segmentation of color images". In: *Proceedings of Canadian Conference on Electrical and Computer Engineering*, 1250–1253 vol.2. DOI: 10.1109/CCECE.1993.332483.

Cunningham, Alexander, Vadim Indelman, and Frank Dellaert (2013). "DDF-SAM 2.0: Consistent distributed smoothing and mapping". In: *2013 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 5220–5227. DOI: 10.1109/ICRA.2013.6631323.

Cunningham, Alexander, Manohar Paluri, and Frank Dellaert (2010). "DDF-SAM: Fully distributed SLAM using Constrained Factor Graphs". In: *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 3025–3030. DOI: 10.1109/IROS.2010.5652875.

Daoud, Hayyan Afeef, Aznul Qalid Md. Sabri, Chu Kiong Loo, and Ali Mohammed Mansoor (Apr. 2018). "SLAMM: Visual monocular SLAM with continuous mapping using multiple maps". In: *PLOS ONE* 13.4. Ed. by Antonio Agudo, e0195878. DOI: 10.1371/journal.pone.0195878. URL: https://doi.org/10.1371/journal.pone.0195878.

Davison, Andrew J., Ian D. Reid, Nicholas D. Molton, and Olivier Stasse (2007). "MonoSLAM: Real-Time Single Camera SLAM". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 29.6, pp. 1052–1067. DOI: `10.1109/TPAMI.2007.1049`.

Debruyn, Diego, Raphael Zufferey, Sophie F. Armanini, Crystal Winston, André Farinha, Yufei Jin, and Mirko Kovac (2020). "MEDUSA: A Multi-Environment Dual-Robot for Underwater Sample Acquisition". In: *IEEE Robotics and Automation Letters* 5.3, pp. 4564–4571. DOI: `10.1109/LRA.2020.3001534`.

Deutsch, Isaac, Ming Liu, and Roland Siegwart (2016). "A framework for multi-robot pose graph SLAM". In: *2016 IEEE International Conference on Real-time Computing and Robotics (RCAR)*, pp. 567–572. DOI: `10.1109/RCAR.2016.7784092`.

Digumarti, Sundara Tejaswi, Gaurav Chaurasia, Aparna Taneja, Roland Siegwart, Amber Thomas, and Paul Beardsley (2016). "Underwater 3D capture using a low-cost commercial depth camera". In: *2016 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pp. 1–9. DOI: `10.1109/WACV.2016.7477644`.

Dragicevic, Pierre and Yvonne Jansen (2012). *List of Physical Visualizations*. www.dataphys.org/list. Last accessed 2023-10-04.

Dubois, Rodolphe, Alexandre Eudes, and Vincent Frémont (2019). "On Data Sharing Strategy for Decentralized Collaborative Visual-Inertial Simultaneous Localization And Mapping". In: *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 2123–2130. DOI: `10.1109/IROS40897.2019.8967617`.

Duncan, Roger G., Mark E. Froggatt, Stephen T. Kreger, Ryan J. Seeley, Dawn K. Gifford, Alexander K. Sang, and Matthew S. Wolfe (Apr. 2007). "High-accuracy fiber-optic shape sensing". In: *Sensor Systems and Networks: Phenomena, Technology, and Applications for NDE and Health Monitoring 2007*. Ed. by Kara J. Peters. SPIE. DOI: `10.1117/12.720914`. URL: `https://doi.org/10.1117/12.720914`.

D'Antonio, Diego S., Gustavo A. Cardona, and David Saldaña (2021). "The Catenary Robot: Design and Control of a Cable Propelled by Two Quadrotors". In: *IEEE Robotics and Automation Letters* 6.2, pp. 3857–3863. DOI: `10.1109/LRA.2021.3062603`.

D'Antonio, Diego S. and David Saldaña (2022). "Folding Knots Using a Team of Aerial Robots". In: *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 3372–3377. DOI: `10.1109/IROS47612.2022.9981363`.

Eidsvik, Ole Alexander and Ingrid Schjølberg (2016). "Time Domain Modeling of ROV Umbilical using Beam Equations". In: *IFAC-PapersOnLine* 49.23. 10th IFAC Conference on Control Applications in Marine SystemsCAMS 2016, pp. 452–457. ISSN: 2405-8963. DOI: `https://doi.org/10.1016/j.ifacol.2016.10.447`. URL: `https://www.sciencedirect.com/science/article/pii/S2405896316320353`.

Elvira, Richard, Juan D. Tardós, and J.M.M. Montiel (2019). "ORBSLAM-Atlas: a robust and accurate multi-map system". In: *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 6253–6259. DOI: `10.1109/IROS40897.2019.8967572`.

Engel, Jakob, Vladlen Koltun, and Daniel Cremers (2018). "Direct Sparse Odometry". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 40.3, pp. 611–625. DOI: `10.1109/TPAMI.2017.2658577`.

Eustice, Ryan, Hanumant Singh, John Leonard, Matthew Walter, and Robert Ballard (2005). "Visually Navigating the RMS Titanic with SLAM Information Filters". In: *Proceedings of Robotics: Science and Systems*. Cambridge, USA. DOI: `10.15607/RSS.2005.I.008`.

Eustice, Ryan M., Hanumant Singh, John J. Leonard, and Matthew R. Walter (2006). "Visually Mapping the RMS Titanic: Conservative Covariance Estimates for SLAM Information Filters". In: *The International Journal of Robotics Research* 25.12, pp. 1223–1242. DOI: `10.1177/0278364906072512`. eprint: `https://doi.org/10.1177/0278364906072512`. URL: `https://doi.org/10.1177/0278364906072512`.

Ferrera, Maxime (2019). "Monocular Visual-Inertial-Pressure fusion for Underwater localization and 3D mapping". PhD thesis. Montpellier, France: University of Montpellier.

Ferrera, Maxime, Vincent Creuze, Julien Moras, and Pauline Trouvé-Peloux (2019). "AQUALOC: An underwater dataset for visual–inertial–pressure localization". In: *The International Journal of Robotics Research* 38.14, pp. 1549–1559. DOI: `10.1177/0278364919883346`. eprint: `https://doi.org/10.1177/0278364919883346`. URL: `https://doi.org/10.1177/0278364919883346`.

Ferrera, Maxime, Alexandre Eudes, Julien Moras, Martial Sanfourche, and Guy Le Besnerais (2021). "OV$^2$SLAM: A Fully Online and Versatile Visual SLAM for Real-Time Applications". In: *IEEE Robotics and Automation Letters* 6.2, pp. 1399–1406. DOI: `10.1109/LRA.2021.3058069`.

Forster, Christian, Zichao Zhang, Michael Gassner, Manuel Werlberger, and Davide Scaramuzza (2017). "SVO: Semidirect Visual Odometry for Monocular and Multicamera Systems". In: *IEEE Transactions on Robotics* 33.2, pp. 249–265. DOI: `10.1109/TRO.2016.2623335`.

Frank, J.E., R. Geiger, D. R. Kraige, and A. Murali (2013). *Smart tether system for underwater navigation and cable shape measurement*. US Patent 8437979-B2.

Galvez-López, Dorian and Juan D. Tardos (2012). "Bags of Binary Words for Fast Place Recognition in Image Sequences". In: *IEEE Transactions on Robotics* 28.5, pp. 1188–1197. DOI: `10.1109/TRO.2012.2197158`.

Gao, Xiang, Rui Wang, Nikolaus Demmel, and Daniel Cremers (2018). "LDSO: Direct Sparse Odometry with Loop Closure". In: *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 2198–2204. DOI: `10.1109/IROS.2018.8593376`.

Geiger, Andreas, Philip Lenz, and Raquel Urtasun (2012). "Are we ready for autonomous driving? The KITTI vision benchmark suite". In: *2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3354–3361. DOI: `10.1109/CVPR.2012.6248074`.

Ghader, Bilal, Claire Dune, Eric Watelain, and Vincent Hugel (2023). "Skeleton-based Visual Recognition of Diver's Gesture". In: *OCEANS 2023 - Limerick*, pp. 1–5. DOI: `10.1109/OCEANSLimerick52467.2023.10244628`.

Grupp, Michael (2017). *evo: Python package for the evaluation of odometry and SLAM.* `https://github.com/MichaelGrupp/evo`.

Hall, Matthew (2020). "MoorDyn V2: New Capabilities in Mooring System Components and Load Cases". In: *Volume 9: Ocean Renewable Energy*. URL: `https://api.semanticscholar.org/CorpusID:234519235`.

Hartley, Richard and Andrew Zisserman (2004). *Multiple View Geometry in Computer Vision*. 2nd ed. Cambridge University Press. DOI: `10.1017/CBO9780511811685`.

Hidalgo, Franco, Chris Kahlefendt, and Thomas Bräunl (2018). "Monocular ORB-SLAM Application in Underwater Scenarios". In: *2018 OCEANS - MTS/IEEE Kobe Techno-Oceans (OTO)*, pp. 1–4. DOI: `10.1109/OCEANSKOBE.2018.8559435`.

Hong, Sung Min, Kyoung Nam Ha, and Joon-Young Kim (2020). "Dynamics Modeling and Motion Simulation of USV/UUV with Linked Underwater Cable". In: *Journal of Marine Science and Engineering* 8.5. ISSN: 2077-1312. DOI: `10.3390/jmse8050318`. URL: `https://www.mdpi.com/2077-1312/8/5/318`.

Huang, Huajian, Wen-Yan Lin, Siying Liu, Dong Zhang, and Sai-Kit Yeung (2020). "Dual-SLAM: A framework for robust single camera navigation". In: *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 4942–4949. DOI: `10.1109/IROS45743.2020.9341513`.

Islam, Md Jahidul, Youya Xia, and Junaed Sattar (2020). "Fast Underwater Image Enhancement for Improved Visual Perception". In: *IEEE Robotics and Automation Letters* 5.2, pp. 3227–3234. DOI: `10.1109/LRA.2020.2974710`.

Jin, Shiyu, Wenzhao Lian, Changhao Wang, Masayoshi Tomizuka, and Stefan Schaal (2022). "Robotic Cable Routing with Spatial Representation". In: *IEEE Robotics and Automation Letters* 7.2, pp. 5687–5694. DOI: `10.1109/LRA.2022.3158377`.

Joshi, Bharat, Sharmin Rahman, Michail Kalaitzakis, Brennan Cain, James Johnson, Marios Xanthidis, Nare Karapetyan, Alan Hernandez, Alberto Quattrini Li, Nikolaos Vitzilaios, and Ioannis Rekleitis (2019). "Experimental Comparison of Open Source Visual-Inertial-Based State Estimation Algorithms in the Underwater Domain". In: *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 7227–7233. DOI: `10.1109/IROS40897.2019.8968049`.

Katzschmann, Robert K., Joseph DelPreto, Robert MacCurdy, and Daniela Rus (2018). "Exploration of underwater life with an acoustically controlled soft robotic fish". In: *Science Robotics* 3.16, eaar3449. DOI: `10.1126/scirobotics.aar3449`. eprint: `https://www.science.org/doi/pdf/10.1126/scirobotics.aar3449`. URL: `https://www.science.org/doi/abs/10.1126/scirobotics.aar3449`.

Khatib, Oussama, Xiyang Yeh, Gerald Brantner, Brian Soe, Boyeon Kim, Shameek Ganguly, Hannah Stuart, Shiquan Wang, Mark Cutkosky, Aaron Edsinger, Phillip Mullins, Mitchell Barham, Christian R. Voolstra, Khaled Nabil Salama, Michel L'Hour, and Vincent Creuze (2016). "Ocean One: A Robotic Avatar for Oceanic Discovery". In: *IEEE Robotics and Automation Magazine* 23.4, pp. 20–29. DOI: `10.1109/MRA.2016.2613281`.

Klein, Georg and David Murray (2007). "Parallel Tracking and Mapping for Small AR Workspaces". In: *2007 6th IEEE and ACM International Symposium on Mixed and Augmented Reality*, pp. 225–234. DOI: `10.1109/ISMAR.2007.4538852`.

Lagarias, Jeffrey C., James A. Reeds, Margaret H. Wright, and Paul E. Wright (1998). "Convergence Properties of the Nelder–Mead Simplex Method in Low Dimensions". In: *SIAM Journal on Optimization* 9.1, pp. 112–147. DOI: `10.1137/S1052623496303470`. eprint: `https://doi.org/10.1137/S1052623496303470`. URL: `https://doi.org/10.1137/S1052623496303470`.

Laranjeira, Matheus (2019). "Visual Servoing on Deformable Objects: An Application to Tether Shape Control". PhD thesis. Toulon, France: University of Toulon.

Laranjeira, Matheus, Claire Dune, and Vincent Hugel (2017). "Catenary-based visual servoing for tethered robots". In: *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 732–738. DOI: `10.1109/ICRA.2017.7989090`.

— (Oct. 2018). "Local Vision-Based Tether Control for a Line of Underwater Robots". In: *IEEE IROS 2018*. Madrid, Spain. URL: `https://hal.science/hal-04123118`.

— (2019). "Embedded Visual Detection and Shape Identification of Underwater Umbilical for Vehicle Positioning". In: *OCEANS 2019 - Marseille*, pp. 1–9. DOI: `10.1109/OCEANSE.2019.8867548`.

— (2020). "Catenary-based visual servoing for tether shape control between underwater vehicles". In: *Ocean Engineering* 200, p. 107018. ISSN: 0029-8018. DOI: `https://doi.org/10.1016/j.oceaneng.2020.107018`. URL: `https://www.sciencedirect.com/science/article/pii/S0029801820300949`.

Leibniz, Gottfried Wilhelm (1691). “De linea in quam flexile se pondere proprio curvat”. In: *Acta eruditorum,(Juin), Lipsia*.

Leonardos, Spyridon and Kostas Daniilidis (2017). “A game-theoretic approach to Robust Fusion and Kalman filtering under unknown correlations”. In: *2017 American Control Conference (ACC)*, pp. 2568–2573. DOI: `10.23919/ACC.2017.7963339`.

Li, Fu, Shaowu Yang, Xiaodong Yi, and Xuejun Yang (2018). “CORB-SLAM: A Collaborative Visual SLAM System for Multiple Robots”. In: *Collaborative Computing: Networking, Applications and Worksharing*. Ed. by Imed Romdhani, Lei Shu, Hara Takahiro, Zhangbing Zhou, Timothy Gordon, and Deze Zeng. Cham: Springer International Publishing, pp. 480–490. ISBN: 978-3-030-00916-8.

Liu, Huimin, Zhenjie Wang, Rui Shan, Kaifei He, and Shuang Zhao (2020). “Research into the integrated navigation of a deep-sea towed vehicle with USBL/DVL and pressure gauge”. In: *Applied Acoustics* 159, p. 107052. ISSN: 0003-682X. DOI: `https://doi.org/10.1016/j.apacoust.2019.107052`. URL: `https://www.sciencedirect.com/science/article/pii/S0003682X18310193`.

Madgwick, Sebastian O. H., Andrew J. L. Harrison, and Ravi Vaidyanathan (2011). “Estimation of IMU and MARG orientation using a gradient descent algorithm”. In: *2011 IEEE International Conference on Rehabilitation Robotics*, pp. 1–7. DOI: `10.1109/ICORR.2011.5975346`.

Mallios, Angelos, Pere Ridao, David Ribas, Marc Carreras, and Richard Camilli (2016). “Toward Autonomous Exploration in Confined Underwater Environments”. In: *Journal of Field Robotics* 33.7, pp. 994–1012. DOI: `https://doi.org/10.1002/rob.21640`. eprint: `https://onlinelibrary.wiley.com/doi/pdf/10.1002/rob.21640`. URL: `https://onlinelibrary.wiley.com/doi/abs/10.1002/rob.21640`.

Mallios, Angelos, David Romagós, and Pere Ridao (Jan. 2009). “Localization Advances in the Unstructured Underwater Environment.” In: pp. 111 –116.

Mangelson, Joshua G., Ram Vasudevan, and Ryan M. Eustice (Oct. 2018). “Communication Constrained Trajectory Alignment For Multi-Agent Inspection via Linear Programming”. In: *OCEANS*. Charleston, SC, USA: MTS/IEEE. DOI: `10.1109/oceans.2018.8604775`.

Massot-Campos, Miquel and Gabriel Oliver-Codina (2015). “Optical Sensors and Methods for Underwater 3D Reconstruction”. In: *Sensors* 15.12, pp. 31525–31557. ISSN: 1424-8220. DOI: `10.3390/s151229864`. URL: `https://www.mdpi.com/1424-8220/15/12/29864`.

McGarey, Patrick, Kirk MacTavish, François Pomerleau, and Timothy D Barfoot (2017). “TSLAM: Tethered simultaneous localization and mapping for mobile robots”. In: *The International Journal of Robotics Research* 36.12, pp. 1363–1386. DOI: `10.1177/0278364917732639`. URL: `https://doi.org/10.1177/0278364917732639`.

McGarey, Patrick, François Pomerleau, and Timothy D. Barfoot (2016). “System Design of a Tethered Robotic Explorer (TReX) for 3D Mapping of Steep Terrain and Harsh Environments”. In: *Field and Service Robotics: Results of the 10th International Conference*. Ed. by David S. Wettergreen and Timothy D. Barfoot. Cham: Springer International Publishing, pp. 267–281. ISBN: 978-3-319-27702-8. DOI: `10.1007/978-3-319-27702-8_18`. URL: `https://doi.org/10.1007/978-3-319-27702-8_18`.

Meng, Yan, Xuesong Xu, and Min Zhao (2020). “Dynamics calculation of complex deep-sea cable system based on hybrid optimization algorithm”. In: *Ocean Engineering* 200, p. 107041. ISSN: 0029-8018. DOI: `https://doi.org/10.1016/j.`

oceaneng.2020.107041. URL: https://www.sciencedirect.com/science/article/pii/S0029801820301153.

Merci, Aurélien, Cédric Anthierens, Nadège Thirion-Moreau, and Yann Le Page (2023). "A simulator of underwater glider missions for path planning". In: *Ocean Engineering* 269, p. 113514. ISSN: 0029-8018. DOI: https://doi.org/10.1016/j.oceaneng.2022.113514. URL: https://www.sciencedirect.com/science/article/pii/S0029801822027974.

Merlet, J.-P. (2018). "An Experimental Investigation of Extra Measurements for Solving the Direct Kinematics of Cable-Driven Parallel Robots". In: *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 6947–6952. DOI: 10.1109/ICRA.2018.8460901.

Michael, Nathan, Shaojie Shen, Kartik Mohta, Vijay Kumar, Yoshito Nagatani Keijiand Okada, Seiga Kiribayashi, Kazuki Otake, Kazuya Yoshida, Kazunori Ohno, Eijiro Takeuchi, and Satoshi Tadokoro (2014). "Collaborative Mapping of an Earthquake Damaged Building via Ground and Aerial Robots". In: *Field and Service Robotics: Results of the 8th International Conference*. Ed. by Kazuya Yoshida and Satoshi Tadokoro. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 33–47. ISBN: 978-3-642-40686-7. DOI: 10.1007/978-3-642-40686-7_3. URL: https://doi.org/10.1007/978-3-642-40686-7_3.

Ming, Aiguo, Takahiro Ichikawa, Wenjing Zhao, and Makoto Shimojo (2014). "Development of a sea snake-like underwater robot". In: *2014 IEEE International Conference on Robotics and Biomimetics (ROBIO 2014)*, pp. 761–766. DOI: 10.1109/ROBIO.2014.7090423.

Mur-Artal, Raúl, J. M. M. Montiel, and Juan D. Tardós (2015). "ORB-SLAM: A Versatile and Accurate Monocular SLAM System". In: *IEEE Transactions on Robotics* 31.5, pp. 1147–1163. DOI: 10.1109/TRO.2015.2463671.

Mur-Artal, Raúl and Juan D. Tardós (2017). "ORB-SLAM2: An Open-Source SLAM System for Monocular, Stereo, and RGB-D Cameras". In: *IEEE Transactions on Robotics* 33.5, pp. 1255–1262. DOI: 10.1109/TRO.2017.2705103.

Murphy, Robin R., Karen L. Dreger, Sean Newsome, Jesse Rodocker, Brian Slaughter, Richard Smith, Eric Steimle, Tetsuya Kimura, Kenichi Makabe, Kazuyuki Kon, Hisashi Mizumoto, Michinori Hatayama, Fumitoshi Matsuno, Satoshi Tadokoro, and Osamu Kawase (2012). "Marine heterogeneous multirobot systems at the great Eastern Japan Tsunami recovery". In: *Journal of Field Robotics* 29.5, pp. 819–831. DOI: https://doi.org/10.1002/rob.21435. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/rob.21435. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/rob.21435.

Murtra, Andreu Corominas and Josep M. Mirats Tur (2013). "IMU and cable encoder data fusion for in-pipe mobile robot localization". In: *2013 IEEE Conference on Technologies for Practical Robot Applications (TePRA)*, pp. 1–6. DOI: 10.1109/TePRA.2013.6556377.

Nister, D. and H. Stewenius (2006). "Scalable Recognition with a Vocabulary Tree". In: *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*. Vol. 2, pp. 2161–2168. DOI: 10.1109/CVPR.2006.264.

Nokin, M. (1998). "Sea trials of the deep scientific system VICTOR 6000". In: *IEEE Oceanic Engineering Society. OCEANS'98. Conference Proceedings (Cat. No.98CH36259)*. Vol. 3, 1573–1577 vol.3. DOI: 10.1109/OCEANS.1998.726337.

Özkahraman, Özer and Petter Ögren (2022). "Collaborative Navigation-Aware Coverage in Feature-Poor Environments". In: *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 10066–10073. DOI: 10.1109/IROS47612.2022.9981547.

Pairet, Èric, Juan David Hernández, Marc Carreras, Yvan Petillot, and Morteza Lahijanian (2022). "Online Mapping and Motion Planning Under Uncertainty for Safe Navigation in Unknown Environments". In: *IEEE Transactions on Automation Science and Engineering* 19.4, pp. 3356–3378. DOI: `10.1109/TASE.2021.3118737`.

Pecheux, Nicolas, Vincent Creuze, Frédéric Comby, and Olivier Tempier (2023). "Self Calibration of a Sonar-Vision System for Underwater Vehicles: A New Method and a Dataset". In: *Sensors* 23.3. ISSN: 1424-8220. DOI: `10.3390/s23031700`. URL: `https://www.mdpi.com/1424-8220/23/3/1700`.

Quattrini Li, Alberto, A. Coskun, S. M. Doherty, S. Ghasemlou, A. S. Jagtap, M. Modasshir, S. Rahman, A. Singh, M. Xanthidis, J. M. O'Kane, and I. Rekleitis (2017). "Experimental Comparison of Open Source Vision-Based State Estimation Algorithms". In: *2016 International Symposium on Experimental Robotics*. Ed. by Dana Kulić, Yoshihiko Nakamura, Oussama Khatib, and Gentiane Venture. Cham: Springer International Publishing, pp. 775–786. ISBN: 978-3-319-50115-4.

Rahman, Sharmin, Alberto Quattrini Li, and Ioannis Rekleitis (2019). "SVIn2: An Underwater SLAM System using Sonar, Visual, Inertial, and Depth Sensor". In: *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1861–1868. DOI: `10.1109/IROS40897.2019.8967703`.

Raugel, Ewen, Jan Opderbecke, Marie Claire Fabri, Lorenzo Brignone, and Vincent Rigaud (2019). "Operational and scientific capabilities of Ariane, Ifremer's hybrid ROV". In: *OCEANS 2019 - Marseille*, pp. 1–7. DOI: `10.1109/OCEANSE.2019.8867102`.

Renda, Federico, Frédéric Boyer, Jorge Dias, and Lakmal Seneviratne (2018). "Discrete Cosserat Approach for Multisection Soft Manipulator Dynamics". In: *IEEE Transactions on Robotics* 34.6, pp. 1518–1533. DOI: `10.1109/TRO.2018.2868815`.

Riazuelo, L., Javier Civera, and J.M.M. Montiel (2014). "C2TAM: A Cloud framework for cooperative tracking and mapping". In: *Robotics and Autonomous Systems* 62.4, pp. 401–413. ISSN: 0921-8890. DOI: `https://doi.org/10.1016/j.robot.2013.11.007`. URL: `https://www.sciencedirect.com/science/article/pii/S0921889013002248`.

Rosten, Edward and Tom Drummond (2006). "Machine Learning for High-Speed Corner Detection". In: *Computer Vision – ECCV 2006*. Ed. by Aleš Leonardis, Horst Bischof, and Axel Pinz. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 430–443. ISBN: 978-3-540-33833-8.

Rublee, Ethan, Vincent Rabaud, Kurt Konolige, and Gary Bradski (2011). "ORB: An efficient alternative to SIFT or SURF". In: *2011 International Conference on Computer Vision*, pp. 2564–2571. DOI: `10.1109/ICCV.2011.6126544`.

Sahoo, Avilash, Santosha K. Dwivedy, and P.S. Robi (2019). "Advancements in the field of autonomous underwater vehicle". In: *Ocean Engineering* 181, pp. 145–160. ISSN: 0029-8018. DOI: `https://doi.org/10.1016/j.oceaneng.2019.04.011`. URL: `https://www.sciencedirect.com/science/article/pii/S0029801819301623`.

Salumäe, Taavi, Rasmus Raag, Jaan Rebane, Andres Ernits, Gert Toming, Mart Ratas, and Maarja Kruusmaa (2014). "Design principle of a biomimetic underwater robot U-CAT". In: *2014 Oceans - St. John's*, pp. 1–5. DOI: `10.1109/OCEANS.2014.7003126`.

Salvi, Joaquim, Yvan Petillot, and Elisabet Batlle (2008). "Visual SLAM for 3D large-scale seabed acquisition employing underwater vehicles". In: *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1011–1016. DOI: `10.1109/IROS.2008.4650627`.

Schmuck, Patrik and Margarita Chli (2019). “CCM-SLAM: Robust and efficient centralized collaborative monocular simultaneous localization and mapping for robotic teams”. In: *Journal of Field Robotics* 36.4, pp. 763–781. DOI: `https://doi.org/10.1002/rob.21854`. eprint: `https://onlinelibrary.wiley.com/doi/pdf/10.1002/rob.21854`. URL: `https://onlinelibrary.wiley.com/doi/abs/10.1002/rob.21854`.

Schönberger, Johannes L. and Jan-Michael Frahm (2016). “Structure-from-Motion Revisited”. In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4104–4113. DOI: `10.1109/CVPR.2016.445`.

Schubert, David, Thore Goll, Nikolaus Demmel, Vladyslav Usenko, Jörg Stückler, and Daniel Cremers (2018). “The TUM VI Benchmark for Evaluating Visual-Inertial Odometry”. In: *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1680–1687. DOI: `10.1109/IROS.2018.8593419`.

Silveira, Luan, Felipe Guth, Paulo Drews-Jr, Pedro Ballester, Matheus Machado, Felipe Codevilla, Nelson Duarte-Filho, and Silvia Botelho (2015). “An Open-source Bio-inspired Solution to Underwater SLAM”. In: *IFAC-PapersOnLine* 48.2. 4th IFAC Workshop on Navigation, Guidance and Controlof Underwater VehiclesNGCUV 2015, pp. 212–217. ISSN: 2405-8963. DOI: `https://doi.org/10.1016/j.ifacol.2015.06.035`. URL: `https://www.sciencedirect.com/science/article/pii/S2405896315002748`.

Sladen, Anthony, Diane Rivet, Jean-Paul Ampuero, Louis De Barros, Yann Hello, Gaëtan Calbris, and Patrick Lamare (June 2019). “Distributed sensing of earthquakes and ocean-solid Earth interactions on seafloor telecom cables”. In: DOI: `10.31223/osf.io/ekrfy`. URL: `https://doi.org/10.31223/osf.io/ekrfy`.

Smolentsev, Lev, Alexandre Krupa, and François Chaumette (2023). “Shape visual servoing of a tether cable from parabolic features”. In: *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 734–740. DOI: `10.1109/ICRA48891.2023.10161101`.

Soylu, Serdar, Bradley J. Buckham, and Ron P. Podhorodeski (2010). “Dynamics and control of tethered underwater-manipulator systems”. In: *OCEANS 2010 MTS/IEEE SEATTLE*, pp. 1–8. DOI: `10.1109/OCEANS.2010.5664366`.

Sundaresan, Priya, Jennifer Grannen, Brijen Thananjeyan, Ashwin Balakrishna, Michael Laskey, Kevin Stone, Joseph E. Gonzalez, and Ken Goldberg (2020). “Learning Rope Manipulation Policies Using Dense Object Descriptors Trained on Synthetic Depth Data”. In: *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 9411–9418. DOI: `10.1109/ICRA40945.2020.9197121`.

Torres-Méndez, Luz A. and Gregory Dudek (2005). “Color Correction of Underwater Images for Aquatic Robot Inspection”. In: *Energy Minimization Methods in Computer Vision and Pattern Recognition*. Ed. by Anand Rangarajan, Baba Vemuri, and Alan L. Yuille. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 60–73. ISBN: 978-3-540-32098-2.

Tortorici, Ornella (2021). “Conception et contrôle automatique d’un ombilical instrumenté pour robots sous-marins”. PhD thesis. Toulon, France: University of Toulon.

Tortorici, Ornella, Cedric Anthierens, Vincent Hugel, and Herve Barthelemy (2019). “Towards active self-management of umbilical linking ROV and USV for safer submarine missions”. In: *IFAC-PapersOnLine* 52.21. 12th IFAC Conference on Control Applications in Marine Systems, Robotics, and Vehicles CAMS 2019, pp. 265–270. ISSN: 2405-8963. DOI: `https://doi.org/10.1016/j.ifacol.2019.12.318`. URL: `https://www.sciencedirect.com/science/article/pii/S2405896319322037`.

Tortorici, Ornella, Cédric Anthierens, and Vincent Hugel (2023). “A New Flex-Sensor-Based Umbilical-Length Management System for Underwater Robots”. In: *2023 European Conference on Mobile Robots (ECMR)*, pp. 1–6. DOI: 10.1109/ECMR59166.2023.10256299.

Vargas, Elizabeth, Raluca Scona, Jonatan Scharff Willners, Tomasz Luczynski, Yu Cao, Sen Wang, and Yvan R. Petillot (2021). “Robust Underwater Visual SLAM Fusing Acoustic Sensing”. In: *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2140–2146. DOI: 10.1109/ICRA48506.2021.9561537.

Viel, Christophe (2022a). “Self-Management of ROV Umbilical Using Sliding Buoys and Stop”. In: *IEEE Robotics and Automation Letters* 7.3, pp. 8061–8068. DOI: 10.1109/LRA.2022.3187267.

— (2022b). “Self-management of the umbilical of a ROV for underwater exploration”. In: *Ocean Engineering* 248, p. 110695. ISSN: 0029-8018. DOI: https://doi.org/10.1016/j.oceaneng.2022.110695. URL: https://www.sciencedirect.com/science/article/pii/S0029801822001494.

Viel, Christophe, Juliette Drupt, Claire Dune, and Vincent Hugel (2023). “ROV localization based on umbilical angle measurement”. In: *Ocean Engineering* 269, p. 113570. ISSN: 0029-8018. DOI: https://doi.org/10.1016/j.oceaneng.2022.113570. URL: https://www.sciencedirect.com/science/article/pii/S0029801822028530.

Wang, Yan, Wei Song, Giancarlo Fortino, Li-Zhe Qi, Wenqiang Zhang, and Antonio Liotta (2019). “An Experimental-Based Review of Image Enhancement and Image Restoration Methods for Underwater Imaging”. In: *IEEE Access* 7, pp. 140233–140251. DOI: 10.1109/ACCESS.2019.2932130.

Wu, Jiaming, Shunyuan Xu, Hua Liao, Chenghua Ma, Xianyuan Yang, Haotian Wang, Tian Zhang, and Xiangxi Han (2021). “Maneuverability and Hydrodynamics of a Tethered Underwater Robot Based on Mixing Grid Technique”. In: *Journal of Marine Science and Engineering* 9.6. ISSN: 2077-1312. DOI: 10.3390/jmse9060561. URL: https://www.mdpi.com/2077-1312/9/6/561.

Wu, Yilin, Wilson Yan, Thanard Kurutach, Lerrel Pinto, and Pieter Abbeel (July 2020). “Learning to Manipulate Deformable Objects without Demonstrations”. In: *Robotics: Science and Systems XVI*. Robotics: Science and Systems Foundation. DOI: 10.15607/rss.2020.xvi.065. URL: https://doi.org/10.15607/rss.2020.xvi.065.

Xu, Shida, Tomasz Luczynski, Jonatan Scharff Willners, Ziyang Hong, Kaicheng Zhang, Yvan R. Petillot, and Sen Wang (2021). “Underwater Visual Acoustic SLAM with Extrinsic Calibration”. In: *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 7647–7652. DOI: 10.1109/IROS51168.2021.9636258.

Yan, Mengyuan, Yilin Zhu, Ning Jin, and Jeannette Bohg (2020). “Self-Supervised Learning of State Estimation for Manipulating Deformable Linear Objects”. In: *IEEE Robotics and Automation Letters* 5.2, pp. 2372–2379. DOI: 10.1109/LRA.2020.2969931.

Yu, Son-Cheol, Junku Yuh, and Jinwhan Kim (2013). “Armless underwater manipulation using a small deployable agent vehicle connected by a smart cable”. In: *Ocean Engineering* 70, pp. 149–159. ISSN: 0029-8018. DOI: https://doi.org/10.1016/j.oceaneng.2013.06.006. URL: https://www.sciencedirect.com/science/article/pii/S002980181300245X.

Yuan, Fei, Fengqi Xiao, Kaihan Zhang, Yifan Huang, and En Cheng (2021). “Noise reduction for sonar images by statistical analysis and fields of experts”. In: *Journal of Visual Communication and Image Representation* 74, p. 102995. ISSN: 1047-3203.

DOI: https://doi.org/10.1016/j.jvcir.2020.102995. URL: https://www.sciencedirect.com/science/article/pii/S104732032030211X.

Zhang, Jun, Viorela Ila, and Laurent Kneip (2018). "Robust Visual Odometry in Underwater Environment". In: *2018 OCEANS - MTS/IEEE Kobe Techno-Oceans (OTO)*, pp. 1–9. DOI: 10.1109/OCEANSKOBE.2018.8559452.

Zhang, Song, Shili Zhao, Dong An, Jincun Liu, He Wang, Yu Feng, Daoliang Li, and Ran Zhao (2022). "Visual SLAM for underwater vehicles: A survey". In: *Computer Science Review* 46, p. 100510. ISSN: 1574-0137. DOI: https://doi.org/10.1016/j.cosrev.2022.100510. URL: https://www.sciencedirect.com/science/article/pii/S1574013722000442.

Zhu, Jihong, David Navarro-Alarcon, Robin Passama, and Andrea Cherubini (2021). "Vision-based manipulation of deformable and rigid objects using subspace projections of 2D contours". In: *Robotics and Autonomous Systems* 142, p. 103798. ISSN: 0921-8890. DOI: https://doi.org/10.1016/j.robot.2021.103798. URL: https://www.sciencedirect.com/science/article/pii/S092188902100083X.

Zhu, Yongyun, Tao Zhang, Shouquan Xu, Hyo-Sang Shin, Peijuan Li, Bonan Jin, Liang Zhang, Chengcheng Weng, and Yao Li (2020). "A Calibration Method of USBL Installation Error Based on Attitude Determination". In: *IEEE Transactions on Vehicular Technology* 69.8, pp. 8317–8328. DOI: 10.1109/TVT.2020.2995599.

Zou, Danping and Ping Tan (2013). "CoSLAM: Collaborative Visual SLAM in Dynamic Environments". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 35.2, pp. 354–366. DOI: 10.1109/TPAMI.2012.104.

Zou, Danping, Ping Tan, and Wenxian Yu (2019). "Collaborative visual SLAM for multiple agents:A brief survey". In: *Virtual Reality and Intelligent Hardware* 1.5. 3D Vision, pp. 461–482. ISSN: 2096-5796. DOI: https://doi.org/10.1016/j.vrih.2019.09.002. URL: https://www.sciencedirect.com/science/article/pii/S2096579619300634.

Zubizarreta, Jon, Iker Aguinaga, and Jose Maria Martinez Montiel (2020). "Direct Sparse Mapping". In: *IEEE Transactions on Robotics* 36.4, pp. 1363–1370. DOI: 10.1109/TRO.2020.2991614.