Master's thesis

NTNU
Norwegian University of Science and Technology
Faculty of Engineering
Department of Marine Technology

Martin Kvisvik Larsen

Terrain-Based Navigation for Unmanned Underwater Vehicles Using Visual Simultaneous Localization and Mapping

Master's thesis in Marine Technology
Supervisor: Martin Ludvigsen
Co-supervisor: Håvard Snefjellå Løvås
June 2021

NTNU
Norwegian University of Science and Technology

Martin Kvisvik Larsen

Terrain-Based Navigation for Unmanned Underwater Vehicles Using Visual Simultaneous Localization and Mapping

Master's thesis in Marine Technology
Supervisor: Martin Ludvigsen
Co-supervisor: Håvard Snefjellå Løvås
June 2021

Norwegian University of Science and Technology
Faculty of Engineering
Department of Marine Technology

NTNU

Kunnskap for en bedre verden

NTNU Trondheim
Norwegian University of Science and Technology
Department of Marine Technology

PROJECT DESCRIPTION SHEET

Name of the candidate:	Larsen, Martin Kvisvik
Field of study:	Ocean Mapping and Visual Simultaneous Localization and Mapping
Thesis title (Norwegian):	Terrengbasert navigasjon av ubemannede undervannsfartøy ved bruk av visuell simultan lokalisering og kartlegging
Thesis title (English):	Terrain-based Navigation for Unmanned Underwater Vehicles Using Visual Simultaneous Localization and Mapping

Background

Navigation for unmanned underwater vehicles is heavily reliant on acoustic positioning systems (APSs). APSs are, however, expensive and provide low frequency navigation data with relatively low accuracy. Additionally, at greater water depths APSs are susceptible to significant time delays and reduced accuracy due to acoustic refraction. For benthic surveys, visual simultaneous localization and mapping (V-SLAM) is an alternative method of providing navigation- and bathymetry data. State of the art V-SLAM algorithms rely on inexpensive sensors, such as digital cameras and inertial measurement units, and can provide accurate navigation data at high frequency. The potential gains of adopting V-SLAM for underwater navigation are therefore significant in terms of temporal resolution, accuracy, and cost.

However, dead reckoning navigation systems, like V-SLAM, accumulate drift over time. Additionally, in underwater environments, optical sensors are susceptible to several optical phenomena, caused by the water and its constitutes, housing setup, and scene illumination. Some common underwater optical phenomena are; 1) loss of signal and change in perceived color due to light attenuation, 2) loss of contrast due to forward scattering, 3) loss of dynamic range due to backward scattering, 4) changes in the perceived size of objects due to light refraction in the housing interfaces, 5) vignetting due to uneven scene illumination. In general, underwater optical phenomena significantly reduce the robustness of photogrammetric methods, like V-SLAM. Due to the relatively low adoption of underwater V-SLAM, there is a need to identify parameters that are important for overall robustness. Additionally, there is a need to establish image processing methods to compensate for the aforementioned optical phenomena, as well as evaluate and quantify how these methods affect V-SLAM algorithms.

Work description

1. Perform a background and literature review to provide information and relevant references on:
   i. Underwater image formation and optical phenomena caused by natural waters, as well as lamp and housing setup.
   ii. Photogrammetric models and calibration methods relevant for stereo cameras.
   iii. The SLAM problem formulation, the mathematical foundation of graph-based SLAM, and the architecture and submethods of the OpenVSLAM algorithm.
   iv. The technical specifications, system topologies, and hardware configurations for relevant sensors and vehicles.

2. Collect real-world data relevant for V-SLAM by conducting a survey of the Ekne wreck site in the Trondheim fjord. Perform preparations to ensure sufficient data quality and -quantity. Specifically, gather information about the wreck site, set up an underwater housing for a Stereolabs ZED stereo camera, and create a deployable geometric calibration target.

3. Establish a model for the camera setup with the ZED stereo camera mounted inside the underwater housing. Conduct a series of experiments to calibrate the camera setup. Evaluate and discuss the obtained model parameters with regards to physical interpretation, and light refraction introduced by the housing interfaces.

4. Process navigation data from the wreck site survey to create a ground truth reference. Utilize the ground truth reference to georeference the trajectories and maps from OpenVSLAM, and evaluate the trajectories by comparing them to the ground truth reference.

5. Investigate different image processing methods to compensate for underwater optical phenomena. Evaluate how the image processing methods compensate for the different optical phenomena, as well as their effect on OpenVSLAM in terms of the extracted visual features, robustness, and accumulated drift.

6. Identify parameters that are important for the robustness of V-SLAM for underwater surveys, and discuss the validity and benefits of using V-SLAM for underwater navigation.

Specifications
The initial scope of work might be larger than anticipated. Therefore, by the approval of the supervisor, parts of the project work may be removed or reduced in size without any consequences in terms of grading. Personal contributions to problem solutions within the scope of work shall be presented by the candidate. Mathematical derivations and logical reasoning should be the primary basis of theories and conclusions.

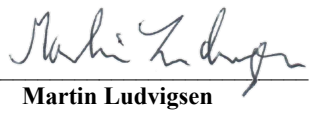
The structure of the report shall be logical and clearly outline background, results, discussions and conclusions. The language of the report should be clear and to the point and written in English. For illustrative purposes, mathematical deduction and figures should be preferred over textual explanations. The report shall contain the following elements: 1) title page, 2) abstract, 3) project description, 4) symbols and acronyms list, 5) table of contents, 6) introduction, 7) background, 8) project scope and delimitations, 9) results, 10) conclusions, 11) recommendations for further work, 12) references and optional appendices. Clearly distinctions shall be made between the original contribution of the candidate and material from other sources by using quotations and Harvard style citation.

The project work should be conducted in a manner that is in line with the NTNU code of ethics, without plagiarism and misconduct. Unless otherwise agreed, the results of the project can be freely used by NTNU in research and teaching by referencing the original work. The project report shall be submitted electronically in accordance with the specifications given by the NTNU administration, with a copy of the final revised project description included.

Start date: 15th January, 2021 **Due date:** 10th June, 2021

Supervisor: Martin Ludvigsen
Co-advisor(s): Håvard Snefjellå Løvås

Trondheim, 7th June 2021



Martin Ludvigsen
Supervisor

Abstract

This thesis investigates the robustness of visual simultaneous localization and mapping (V-SLAM) for navigation of unmanned underwater vehicles, as well as image processing methods suitable for underwater V-SLAM. A dataset is created by conducting a wreck site survey with a stereo camera mounted on a remotely operated vehicle (ROV). Two camera calibration experiments are conducted in a sea water tank, and a camera model is identified for the stereo camera by performing a camera calibration. Four different image processing methods are implemented into the V-SLAM algorithm OpenVSLAM; a bilateral filter (BLF), histogram equalization, contrast-limited adaptive histogram equalization, and a state-of-the-art convolutional neural network for underwater color correction and backscatter estimation. The visual effects of the image processing methods are identified by inspecting image histograms and similarity images. The ROV navigation data is used to estimate a ground truth reference, which is then utilized to georeference the trajectory- and map estimates from OpenVSLAM. The ground truth reference is also used to calculate the absolute trajectory error and the relative pose error (RPE) of OpenVSLAM's trajectory estimates. A comparison analysis of OpenVSLAM with various configurations of the image processing methods is then performed.

By looking at the visual feature distribution of image pyramids, the total number of visual features, and the trajectory lengths, suppression of image noise and forward scattering blur are identified to be important factors for feature matching and, consequently, the robustness of V-SLAM algorithms in underwater applications. For this purpose, the BLF is found to be a highly suitable image processing method for underwater V-SLAM. By analysing the RPE, the most significant source for accumulated drift is identified to be loss of visual features due to sudden changes in perspective. Proper maneuvering, with low altitude and without sharp turns, is identified to be an important factor for underwater V-SLAM, both in terms of robustness and accumulated drift. A well-suited camera- and lamp setup for the relevant survey is also found to be an important, practical factor for robust applications of V-SLAM in underwater environments. Evidence is also found, which indicate that the static map assumption of the full SLAM standard model is a considerable robustness factor for underwater V-SLAM, due to the large number of dynamic targets. OpenVSLAM's bag of visual words-based loop detection method is also found to be unsuited for underwater V-SLAM, due to it's sensitivity to changes in illumination.

Sammendrag

Denne avhandlingen ser nærmere på robustheten til visuell simultan lokalisering og kartlegging (V-SLAM) for navigasjon av ubemannede undervannsfarkoster, samt bildebehandlingsmetoder som er velegnede for V-SLAM under vann. Et datasett ble laget ved å gjennomføre en undersøkelse av et vrak med et stereokamera montert på en fjernstyrt undervannsfarkost (ROV). To kamerakalibreringsforsøk ble gjennomført i en tank med sjøvann, og en kameramodell ble identifisert for stereokameraet ved å foreta en kamerakalibrering. Fire forskjellige bildebehandlingsmetoder ble implementert i V-SLAM-algoritmen OpenVSLAM; et bilateralt filter (BLF), histogramutgjevning, kontrastbegrenset adaptiv histogramutgjevning, samt et nevralt nettverk for fargekorreksjon og lysspredningsestimering i undervannsbilder. De visuelle effektene av bildebehandlingsmetodene ble identifisert ved inspeksjon av bildehistogrammer og likhetsbilder. Navigasjonsdataen fra ROV-en ble brukt til å lage en sammenligningsreferanse, som ble brukt til å georeferere baneestimater og kartestimater fra OpenVSLAM. Sammenligningsreferansen ble også brukt til å beregne den absolutte banefeilen og den relative posisjonsfeilen (RPE) til OpenVSLAMs baneestimater. En sammenligningsstudie av OpenVSLAM med forskjellige konfigurasjoner av bildehandlingsmetoder ble så utført.

Ved å se på distribusjonen av visuelle kjennetegn i bildepyramider, det totale antallet visuelle kjennetegn, samt banelengder, ble filtrering av bildestøy og lysspredning identifisert som viktige faktorer for å finne overensstemmelser av visuelle kjennetegn, og følgelig robustheten til V-SLAM-algoritmer til undervannsbruk. For dette formålet ble BLF funnet til å være en høyst passende bildebehandlingsmetode for V-SLAM under vann. Ved å analysere RPE-en ble den mest signifikante kilden til akkumulert drift funnet til å være tap av visuelle kjennetegn som følge av krappe endringer i perspektiv. Nøye tilpasset manøvrering, med lav altitude og uten krappe svinger, ble identifisert til å være en viktig faktor for V-SLAM under vann, både med tanke på robusthet og akkumulert drift. Et kamera- og lysoppsett tilpasset den aktuelle undersøkelsen ble også identifisert som en viktig, praktisk faktor for robust anvendelse av V-SLAM i undervannsmiljøer. Bevis ble også funnet på at antagelsen om et statisk kart i standardmodellen for V-SLAM er en betydningsfull robusthetsfaktor for V-SLAM under vann, på grunn av det store antallet dynamiske mål. OpenVSLAMs sløyfedeteksjonsmetode ble også funnet til å være upassende for V-SLAM under vann på grunn av dens sensitivitet til endringer i belysning.

Preface

This project is the result of the work done in the 30 point course TMR4930 - Marine Technology, Master's Thesis at the Norwegian University of Science and Technology. The work in this project has been conducted between January 2021 and June 2021, and is a continuation of the work from the project thesis conducted during the Autumn of 2020.

Contents

List of Tables

List of Figures

Nomenclature

Acronyms

AOP	Appearent optical property
APS	Acoustic positioning system
ATE	Absolute trajectory error
AURLab	Applied Underwater Robotics Laboratory
BA	Bundle adjustment
BLF	Bilateral filter
BOW	Bag of words
BRIEF	Binary robust independent elementary features
CLAHE	Contrast-limited adaptive histogram equalization
CNN	Convolutional neural network
CS	Coordinate system
DL	Deep learning
DVL	Doppler velocity log
FAST	Features from accelerated segment test
FIR	Finite impulse response
FOV	Field of view
GPS	Global positioning system
HE	Histogram equalization
IMU	Inertial measurement unit
INS	Inertial navigation system
IOP	Inherent optical property
MAP	Maximum a posterior
MLE	Maximum likelihood estimation
MRS	Motion reference system
MRU	Motion reference unit
NED	North-east-down
NEES	Normalized estimation error squared
NIS	Normalized innovations squared

ORB	Oriented FAST and rotated BRIEF
OSC	Optically significant constitute
POI	Plane of incidence
RANSAC	Random sample consensus
RGB	Red-green-blue
RGBD	Red-green-blue-depth
RMSE	Root mean squared error
ROV	Remotely operated vehicle
RPE	Relative pose error
RTE	Radiant transfer equation
RWT	Rolling window threshold
SDK	Software development kit
SFM	Structure from motion
SLAM	Simultaneous localization and mapping
SNR	Signal to noise ratio
SOTA	State of the art
SSBL	Super short base line
SVD	Singular value decomposition
SVP	Single viewpoint
TBS	Trondheim Biological Station
TFIDF	Term frequency-inverse document frequency
UIENet	Underwater image enhancement network
UUV	Unmanned underwater vehicle
V-SLAM	Visual simultaneous localization and mapping
VI-SLAM	Visual inertial simultaneous localization and mapping
VNS	Visual navigation system
VO	Visual odometry
VSF	Volum scattering function

Symbols

f	Camera focal length
${}_{c}^{s}\tilde{\mathbf{K}}$	Linear camera matrix
$\mathbf{k}$	Nonlinear correction coefficients
$\mathbf{c}$	Camera principal point
$\boldsymbol{\pi}$	Camera projection function
$\mathbf{o}$	Camera projection center (aperture)
γ	Camera inverse projection scale factor
m	Camera image sensor scale difference
s	Camera image sensor shear
$\mathbf{G}$	Zhang coefficient matrix
${}^{c}\mathbf{m}$	3D landmark in the camera CS
${}^{i}\mathbf{z}$	2D point in the image plane CS
${}^{o}\mathbf{m}$	3D landmark in the object CS

${}^{r}\mathbf{z}$	2D rectified point in the image sensor CS
${}^{s}\mathbf{z}$	2D point in the image sensor CS
${}^{u}\mathbf{z}$	2D undistorted point in the image sensor CS
κ	FAST threshold
$\mathbf{H}$	Image patch
a	Beam absorption coefficient
c	Beam attenuation coefficient
B	Backscatter signal
D	Direct signal
J	Unattenuated image channel intensity
I	Image channel intensity
E	Irradiance
ϕ	Nadir angle
r	Path length
L	Radiance
Φ	Radiant power
θ	Refraction angle
μ	Refractive index
b	Beam scattering coefficient
ψ	Scattering angle
S	Sensor response
T	Total signal
λ	Wavelength of light
$\mathbf{q}$	Quaternion attitude representation
$\mathcal{C}$	Covisibility graph
$\mathbf{w}$	Graph edge
$\mathbf{e}$	Graph edge error
$\mathbf{\Omega}$	Graph edge error information matrix
$\mathbf{J}$	Graph edge error Jacobian matrix
$\mathcal{G}$	A general graph
$\mathbf{s}$	Graph state
$\mathbf{v}$	Graph vertex
$\mathbf{m}^{i}$	Landmark i vector
$\mathbf{m}$	Map set
$\mathcal{M}$	Data association set
$\mathbf{z}$	Measurement set
${}^{\mathbf{z}}\mathbf{\Sigma}$	Measurement covariance
$\mathbf{z}^{i}$	Measurement of landmark i vector
$\mathbf{u}$	Odometry vector
${}^{\mathbf{u}}\mathbf{\Sigma}$	Odometry covariance
$\mathbf{x}$	Pose vector
$\mathbf{p}$	Position vector
$\boldsymbol{\eta}$	State vector
$\mathbf{P}$	State covariance

$\mathcal{T}$	Timestamp association set
${}^{2}_{1}\mathbf{b}$	Stereo camera relative baseline vector
p_{x}	Stereo normal disparity
${}^{\tilde{}}_{c}$	
${}^{\tilde{}}_{r}\mathbf{D}$	Stereo normal disparity kernel
$\tilde{\mathbf{E}}$	The essential matrix
$\tilde{\mathbf{F}}$	The fundamental matrix
${}^{2}_{1}\mathbf{R}$	Stereo camera relative rotation matrix

Chapter 1

Introduction

1.1 Background

Unmanned underwater vehicles (UUVs) are heavily reliant on acoustic positioning systems (APSs) for position measurements. However, APSs are costly systems whose measurements suffer from large uncertainties, low accuracy, and infrequent sampling rates. To compensate for the aforementioned shortcomings, APSs are often coupled with dead reckoning navigation systems, such as inertial navigation systems (INSs), which provide high frequent navigation data, but accumulate drift over time. One alternative to INSs for dead reckoning navigation, is visual navigation systems (VNSs), which focus on utilizing cameras to provide navigation data. One of the benefits of VNSs over INSs, is that the navigation system is aware of the immediate surroundings due to the exteroceptive camera measurements. For the last decade, adaptation of VNSs has been pointed as the next big leap for underwater navigation by the marine robotics community (Dukan, 2014; Nornes, 2018). While similar photogrammetric approaches, such as structure from motion (SFM), have been used for underwater 3D reconstruction for decades, they lack the ability to provide online navigation data due to their high computational complexity. In contrast to SFM, VNSs are formulated in an iterative fashion, which allow them to provide high frequent navigation data in real-time.

VNSs are discrimated into two categories; visual odometry (VO) and visual simultaneous localization and mapping (V-SLAM). The advantage of V-SLAM over VO, is that V-SLAM algorithms estimate a map of the environment. The map estimate allows V-SLAM algorithms to refine their pose estimates, relocalize after track loss, and detect loops. Loop detection enables V-SLAM algorithms to close trajectory loops, and, consequently, correct for the accumulated drift, which would otherwise grow unbounded. For this reason, V-SLAM algorithms can provide accurate navigation data over long duration missions, and reuse map information for revisiting missions (Burguera Burguera and Bonin-Font, 2019). The iterative formulation of V-SLAM algorithms does, however, make them susceptible to

robustness issues. Some common robustness issues are high failure rates, inability to scale and perform mapping over extended periods, and performance only being representative in a limited set of environments. The effort and need to develop V-SLAM algorithms that can handle these robustness issues have, in fact, named the current V-SLAM research era the robust-perception age (Cadena et al., 2016).

Adaptation of underwater V-SLAM algorithm is, in general, low, when compared to terrestrial, urban, and aerial applications. A reason for this low adaptation is the lack of suitable datasets, since most V-SLAM algorithms require high frequency visual data, as well as camera calibration data (Ferrera et al., 2019). Moreover, the underwater environment is, in the context of V-SLAM, considered to be a harsh environment, mainly due to the optical properties of the water and its constitutes (Kim and Eustice, 2013). Light attenuation reduces the visual range, as well as the contrast and signal to noise ratio (SNR) of the acquired images, forward scattering causes objects to appear blurry, and backscatter reduces the dynamic range of cameras. In underwater photogrammetry, a common way of compensating for these optical phenomena, is to employ image processing methods, also referred to as underwater image enhancement methods. Research within the field of underwater image enhancement has, however, primarily been driven by color correction for photography and 3D reconstruction applications (Jian et al., 2021), with only a limited number of studies investigating image processing methods in the context of underwater V-SLAM (Aulinas et al., 2011). For this reason, there is a need to investigate image processing methods that can improve the robustness of underwater V-SLAM.

1.2 Objective

The objective of this project is to investigate the validity of adapting V-SLAM algorithms for underwater navigation. In this regard, factors that have a significant contribution on the robustness and drift of underwater V-SLAM algorithms, should be identified. Moreover, image processing methods that can compensate for underwater optical effects, and increase the robustness and decrease the drift of underwater V-SLAM algorithms, should be investigated. Additionally, the validity of established models and subroutines of V-SLAM algorithms should be evaluated for underwater applications.

1.3 Scope

In order to achieve the above objectives, several tasks have to performed:

- Review of relevant theory on underwater image formation, photogrammetric camera modelling and -stereo vision, as well as V-SLAM.
- Collect *in situ* stereo footage and navigation data to get a realistic and suitable data foundation for V-SLAM.
- Perform an underwater camera calibration of the stereo camera to identify a suitable camera model.

- Process the navigation data to create a ground truth reference, which can be used to evaluate the accuracy and drift of the V-SLAM algorithm OpenVSLAM.
- Implement a variety of image processing methods in the OpenVSLAM algorithm. Evaluate the effect of the image processing methods on underwater images.
- Evaluate how the image processing methods affect OpenVSLAM in terms of robustness and drift. Additionally, evaluate some of the underlying models and subroutines of OpenVSLAM.

1.4 Delimitations

V-SLAM algorithms are complex software systems that require extensive effort to develop and improve upon. For this reason, this project does not attempt to modify or improve upon OpenVSLAM, except for implementations of image processing methods in the tracking module of the algorithm.

1.5 Outline

This project is structured in five chapters. In Chapter 1 the background and outline for the project is presented. In Chapter 2 relevant literature background on underwater image formation, photogrammetric camera modelling, photogrammetric stereo vision, and V-SLAM is provided to give a theoretical foundation for the discussion of the results. In Chapter 3 the methodology that has been used to meet the project's objective is outlined, while Chapter 4 presents and discusses the results of project. Chapter 5 summarizes the results of the project and concludes on the project objective.

Chapter 2

Literature Background

Chapter 2 is in large part a continuation of the work from the project thesis, which was conducted during the autumn of 2020 (Larsen, 2020a). Section 2.2 has been reframed from underwater hyperspectral imaging to underwater image formation, while Section 2.3 remains largely unchanged. Section 2.4 has been supplemented with more information on the state of the art (SOTA) deep learning (DL) stereo vision approaches. Except from the very basic on the simultaneous localization and mapping (SLAM) problem formulation and the full SLAM standard model, Section 2.5 is exclusively the work of this project.

2.1 Notation and Coordinate Systems

2.1.1 Notation

In this project, a quite verbose notation is used for transformations, due to large number of coordinate systems (CSs), evident from Section 2.1.2. For example, transformation of the vector $\mathbf{x}$ from coordinate system b to coordinate system a is expressed as

$$ {}^{a}\mathbf{x} = {}^{a}_{b}\mathbf{H}\, {}^{b}\mathbf{x}, \tag{2.1} $$

where ${}^{b}\mathbf{x}$ is the representation of the vector $\mathbf{x}$ in coordinate system b, ${}^{a}\mathbf{x}$ is the representation in coordinate system a, and ${}^{a}_{b}\mathbf{H}$ is the transformation from b to a. Within the sections on photogrammetry, homogeneous coordinates are utilized extensively. The notation for a vector and its corresponding homogeneous representation is

$$ \mathbf{x} = \begin{bmatrix} x \\ y \\ z \end{bmatrix}, \qquad \tilde{\mathbf{x}} = \begin{bmatrix} wx \\ wy \\ wz \\ w \end{bmatrix}, \tag{2.2} $$

where $\mathbf{x}$ is the vector and $\tilde{\mathbf{x}}$ is its homogeneous representation. Transformations in homogeneous spaces follow a similar notation as $\tilde{\mathbf{H}}$.

2.1.2 Coordinate Systems

This project works both within the fields of kinematics and photogrammetry, and therefore the total number of relevant CSs is relatively high. The definitions of the relevant CSs utilized within this project are listed below and the vector notation for the CSs are shown in Table 2.1.

Coordinate System	Vector Notation
World CS	${}^{w}\mathbf{x}$
Body CS	${}^{b}\mathbf{x}$
Object CS	${}^{o}\mathbf{x}$
Camera CS	${}^{c}\mathbf{x}$
Image Plane CS	${}^{i}\mathbf{x}$
Image Sensor CS	${}^{s}\mathbf{x}$

Table 2.1: Coordinate system vector notations.

World Coordinate System

For this project, the world coordinate system (CS) is a north-east-down (NED) coordinate system defined by the UTM datum in zone 32 on the northern hemisphere. The world CS is used to express absolute positions and orientations from the navigation system of the remotely operated vehicle (ROV) SUB-Fighter 30K, as well as georeferenced V-SLAM output.

Body Coordinate System

For this project, the body CS is defined to be a 3D CS positioned in the center of the APS transponder on the ROV, with the x-axis pointing forward, the y-axis pointing to the starboard, and the z-axis pointing downward.

Object Coordinate System

The object CS is a 3D CS that is used as a local coordinate system in order to express the relative position and orientation of objects. The coordinate system is, for this project, used extensively in the context of V-SLAM, where the origin is defined as the camera position of the first keyframe.

Camera Coordinate System

The camera CS is a 3D CS that is used to describe the position and orientation of objects relative to projection center and field of view (FOV) of the camera. The origin of the camera CS is located in the projection center of the camera at any time, with its z-axis aligned with the optical axis of the camera.

Image Plane Coordinate System

The image plane CS is a 2D CS that is used to describe the projections of points onto the camera focal plane. The origin of the image plane CS is placed in the principal point of the camera, with the x- and y-axis lying in the camera focal plane.

Image Sensor Coordinate System

The image sensor CS is a 2D CS that is used to describe the projections of points into the camera image sensor. The origin of the image sensor CS origin is defined to be in the corner of the image sensor, with the x- and y-axis parallel to the axes of the image sensor.

2.2 Underwater Image Formation

2.2.1 Natural Waters and Optically Significant Constitutes

The underwater optical environment is complex, with a vast spectrum of organisms and inorganic substances interacting with the light through absorption and scattering (Mobley, 1994). Compared to the atmosphere, water bodies are composed of extreme variations of optically significant constitutes (OSCs), which vary with geographic location, season, and numerous other factors (Wozniak and Dera, 2007, p. 1-7). Some commonly referred OSCs are; 1) colored dissolved organic matter, and 2) suspended particulate matter. Water bodies and their wide array of OSCs, display a large variety of optical properties, and are collectively referred to as natural waters (Watson and Zielinski, 2013, p. 3-4).

Unlike the atmosphere, natural waters exhibit wavelength dependent attenuation, which causes them to have widely differently color. This wavelength dependency mainly stems from absorption, which is negligible in the atmosphere (Kokhanovsky, 2004; Solonenko and Mobley, 2015). The large color variation of natural waters has been the motivation behind qualitative optical classification, such as the classical Forel-Ule color scale and the more modern Jerlov water types (Jerlov, 1968).

2.2.2 Radiant Transfer in Scattering Media

Conservation of radiant energy travelling a path length r in an absorbing and scattering medium, like sea water, is expressed as

$$\Phi_i(\lambda) = \Phi_a(\lambda) + \Phi_s(\lambda) + \Phi_t(\lambda), \tag{2.3}$$

where λ is the light's wavelength, $\Phi_i(\lambda)$ is the incident radiant power, $\Phi_a(\lambda)$ is the absorbed radiant power, $\Phi_s(\lambda)$ is the radiant power scattered in all directions, and $\Phi_t(\lambda)$ is the transmitted radiant power with the same direction as the incident direction, illustrated in Figure 2.1 (Watson and Zielinski, 2013, p. 6-7). By using the radiant powers in Equation 2.3, the beam absorption coefficient $a(\lambda)$ is defined as

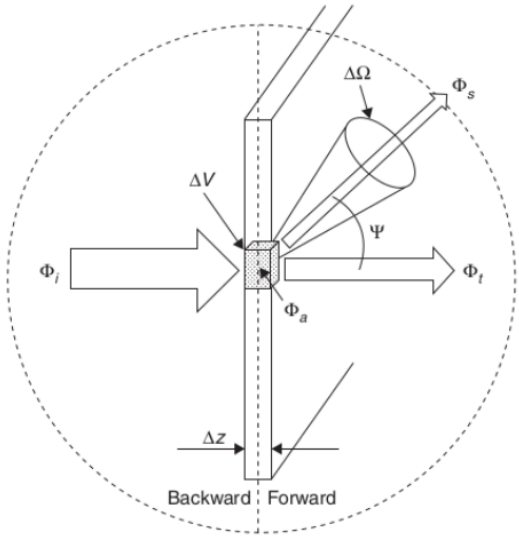


Figure 2.1: Illustration of the radiant power balance in a scattering medium. Note the difference in notation from this project. Courtesy: Watson and Zielinski (2013)

$$a(\lambda) \equiv \lim_{\Delta r \to 0} \frac{\Phi_a(\lambda)}{\Phi_i(\lambda)\Delta r}, \tag{2.4}$$

where Δr is the infinitesimal path length that the light travels through. Similarly, the volume scattering function (VSF), $\beta(\psi, \lambda)$, is defined as

$$\beta(\psi, \lambda) \equiv \lim_{\Delta r \to 0} \lim_{\Delta\Omega \to 0} \frac{\Phi_s(\psi, \lambda)}{\Phi_i(\lambda)\Delta r \Delta\Omega}, \tag{2.5}$$

where Ω is the solid angle of the scattering cone centered around the scattering angle, ψ, the angle between the incident light direction and the scattered light direction. The VSF is the fundamental property for scattering, and can be used to derive all other scattering properties. For instance, the beam scattering coefficient $b(\lambda)$, the forward scattering coefficient $b_f(\lambda)$, and the backward scattering coefficient $b_b(\lambda)$ (Watson and Zielinski, 2013, p. 7) are defined in terms of the VSF as

$$b(\lambda) = b_f(\lambda) + b_b(\lambda) = 2\pi \int_0^{\pi/2} \beta(\psi, \lambda)\sin(\psi)d\psi + 2\pi \int_{\pi/2}^{\pi} \beta(\psi, \lambda)\sin(\psi)d\psi. \tag{2.6}$$

Attenuation of a beam of radiant energy as it propagates directly from an object to an observer, is described through the beam attenuation coefficient $c(\lambda)$, defined in terms of the beam absorption coefficient $a(\lambda)$ and the beam scattering coefficient $b(\lambda)$ as

$$c(\lambda) \equiv a(\lambda) + b(\lambda). \tag{2.7}$$

The beam attenuation coefficient $c(\lambda)$, the beam absorption coefficient $a(\lambda)$, the beam scattering coefficient $b(\lambda)$, and the VSF are so-called inherent optical properties (IOPs), i.e. properties that are independent of the incident light field and only dependent on the light-carrying medium itself. Opposed to the IOPs are the apparent optical properties (AOPs), which do depend on the incident light field. The AOPs are, in general, easier to measure than the IOPs. The dependence on the incident light field do, however, make them more susceptible to variations caused by lamp setup, time of day, weather conditions, etc. An AOP example is the diffuse attenuation coefficient for spectral downwelling plane irradiance, $K_d(\lambda)$, defined as

$$K_d(\lambda) = -\frac{d(\ln(E_d(\lambda)))}{dz}, \tag{2.8}$$

where $E_d(\lambda)$ is the downward irradiance (Mobley, 1994, p. 70). Due to relatively low sensitivity to changes in illumination, the diffuse attenuation coefficient has been used extensively as a proxy for optical classification of natural waters.

Consider the scenario of an observer observing an object through a light-scattering media with ambient lighting. In this scenario, the radiance observed by the observer is governed by the general radiant transfer equation (RTE) (Mobley, 1994, p. 257), accounting for time-variants, in-homogeneities, and three-dimensional behaviour. Due to its general and complex form, the RTE is impractical and often replaced with the classical canonical RTE for a homogeneous, time-invariant, and source-free media, given by

$$\begin{aligned} L(z, \mathbf{d}, \lambda) = &\underbrace{L_0(z_0, \mathbf{d}, \lambda)e^{-c(\lambda)r}}_{\text{Object radiance}} \\ &+ \underbrace{\frac{L_s(z, \mathbf{d}, \lambda)e^{-K_d(\lambda)\cos(\phi)r}}{c(\lambda) - K_d(\lambda)\cos(\phi)}\Big[1 - e^{-[c(\lambda) - K_d(\lambda)\cos(\phi)]r}\Big]}_{\text{Path radiance}}, \end{aligned} \tag{2.9}$$

where $L(z, \mathbf{d}, \lambda)$ is the observed radiant energy reaching an underwater observer, $L_0(z_0, \mathbf{d}, \lambda)$ is the radiant energy leaving an observed object, $\mathbf{d}$ is a direction in three-dimensional space, r is the path length along $\mathbf{d}$, z is the depth, λ is the wavelength, and ϕ is the nadir angle. The nadir angle is by oceanographic convention defined as positive looking downward. $L_s(z, \mathbf{d}, \lambda)$ is the radiant path function, which describes the radiant energy gained along $\mathbf{d}$ due to scattering from all directions (Mobley, 1994, p. 260). In the horizontal scenario, i.e. $\phi = \pi/2$ and $z = z_0$, Equation 2.9 simplifies to

$$L(z, \mathbf{d}, \lambda) = L_0(z_0, \mathbf{d}, \lambda)e^{-c(\lambda)r} + \frac{L_s(z, \mathbf{d}, \lambda)}{c(\lambda)}\Big[1 - e^{-c(\lambda)r}\Big], \tag{2.10}$$

which does not depend on two attenuation coefficients, but only on the beam attenuation coefficient. A simplification in Equation 2.9 that is worth noting, is that the in-scattering radiance, also referred to as the forward scattering, has been omitted (Akkaynak and Treibitz, 2018). However, this simplification is justifiable for underwater imagery, as the forward scattering component is, in general, negligible compared to the direct signal, and therefore has a small contribution to image degradation (Schechner and Karpel, 2004).

2.2.3 Backscatter

In the case of ambient lighting propagating through a scattering medium, and that the scattered light is attenuated exponentially according to Beer-Lamberts attenuation law, the backscattered signal B is given as

$$B(r,\lambda) = \frac{b(\lambda)E(z,\lambda)}{c(\lambda)}\Big(1 - e^{-c(\lambda)r}\Big) = B^{\infty}(\lambda)\Big(1 - e^{-c(\lambda)r}\Big), \tag{2.11}$$

where E is the ambient irradiance at depth z, and B^{∞} is the backscattered signal at infinite distance, also referred to as veiling light (He et al., 2009; Akkaynak et al., 2017). The total signal T at an observer in this case is

$$T(z,\lambda) = E(z,\lambda)e^{-c(\lambda)r} + B^{\infty}(\lambda)\Big(1 - e^{-c(\lambda)r}\Big), \tag{2.12}$$

where the first term is the attenuated direct signal and the second term is the attenuated backscattered signal (Akkaynak and Treibitz, 2018).

2.2.4 Image Formation Models

The traditional image formation model for underwater red-green-blue (RGB) images with ambient illumination is based on the signal model in Equation 2.12, and the assumption that the camera response $S(\lambda)$ are delta functions, or that attenuation is wavelength independent. The traditional RGB image formation model can be expressed as

$$I_k = J_k \cdot e^{-c_k r} + B_k^{\infty} \cdot \Big(1 - e^{-c_k r}\Big), \qquad k \in \{\text{R, G, B}\}, \tag{2.13}$$

where I_k is the image intensity, J_k is unattenuated image intensity, and c_k is the wideband attenuation coefficient for image channel k (Berman et al., 2016). The invalid assumptions of the camera response, and the wavelength independent attenuation is believed to be one of the reasons for instabilities in traditional underwater image correction methods. The revised underwater image formation model, given as

$$I_k = J_k e^{-c_k^D(\cdot)r} + B_k^{\infty}\Big(1 - e^{-c_k^B(\cdot)r}\Big), \qquad k \in \{\text{R, G, B}\}, \tag{2.14}$$

seeks to improve upon these shortcomings by adopting two separate attenuation coefficients to encompass the different wavelength dependency of the direct and backscattered

signal (Akkaynak and Treibitz, 2018). In this model, the direct attenuation coefficient $c_k^D(\cdot)$ and backscatter attenuation coefficient $c_k^B(\cdot)$ are functions of the path length r, the scene reflectance R, the ambient lighting E, the sensor response S_k, the scattering coefficient b, and the attenuation coefficient c, i.e.

$$c_k^D(\cdot) = c_k^D(r, R, E, S_k, c), \qquad k \in \{\text{R, G, B}\}, \tag{2.15a}$$

$$c_k^B(\cdot) = c_k^B(E, S_k, b, c), \qquad k \in \{\text{R, G, B}\}. \tag{2.15b}$$

The revised image formation model in Equation 2.14 is the underlying model of the SOTA underwater image correction algorithm Sea-Thru (Akkaynak and Treibitz, 2019).

2.2.5 Light Refraction

Refraction is a scattering mechanism which occurs when the refractive index of the light-carrying medium changes. Underwater, the mechanism occurs when small fluctuations in the sea water changes the refractive index, also known as Einstein-Smoluchowski scattering, and when light passes through medium interfaces, such as housing ports (Mobley, 1994, p. 102-105). The effects of light beam refraction at interfaces are changes in the perceived size of objects, as well as the perceived relative direction between the objects and the observer, as seen in Figure 2.2. One of the fundamental equations for modelling of light beam refraction at interfaces is Snell's law (Hecht, 2017, p. 109). Within the plane of the incident light beam, known as the plane of incidence (POI), Snell's law is given as

$$\frac{\sin(\theta_i)}{\sin(\theta_t)} = \frac{\mu_t}{\mu_i} = \frac{\nu_i}{\nu_t}, \tag{2.16}$$

where θ_i and θ_t are angles between the light beam and the interface normal in the POI, μ_i and μ_t are the indices of refraction, and ν_i and ν_t are the speed of light in the incident- and transmitting medium, respectively. Outside the POI, in three-dimensional coordinates, Snell's law can be written in vector form as

$$\mathbf{d}_t = \frac{\mu_i}{\mu_t}\big(\mathbf{n} \times (-\mathbf{n} \times \mathbf{d}_i)\big) - \mathbf{n}\sqrt{1 - \left(\frac{\mu_i}{\mu_t}\right)^2 (\mathbf{n} \times \mathbf{d}_i)(\mathbf{n} \times \mathbf{d}_i)}, \tag{2.17}$$

where $\mathbf{d}_i$ and $\mathbf{d}_t$ are the incident- and transmitted direction of the light beam, and $\mathbf{n}$ is the unit normal of the interface. Several physical-based refraction models for underwater optical sensors have been developed based on Equation 2.16 and Equation 2.17, such as the Pinax model, and the refractive single viewpoint (SVP) model (Łuczyński et al., 2017; Telem and Filin, 2010). Additionally, studies have analysed the systematic errors introduced by excluding interface refraction when performing 3D reconstruction based on underwater imagery (Sedlazeck and Koch, 2012). The disadvantages of refractive camera models are, however, the need to measure or estimate the refractive indices of the light-carrying media, μ_i, as well as having an accurate parametrization of the interfaces through the normal vectors $\mathbf{n}$.

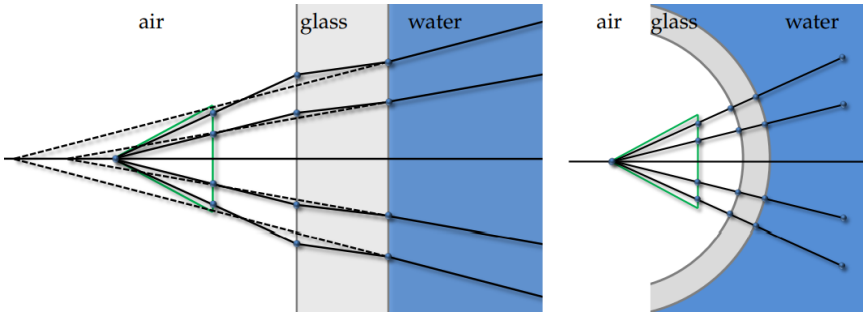


Figure 2.2: Ray diagrams for different housing configurations. Courtesy: Jordt (2014)

2.3 Photogrammetric Camera Modelling

A large portion of the material in this section has been found in Förstner and Wrobel (2016). For simplicity, the reader is referred to section 5.1-5.4 (Förstner and Wrobel, 2016, p.195-242) for the background material on homogeneous representation, and section 12.1.1-12.1.5 (Förstner and Wrobel, 2016, p.456-479) for the background material on camera modelling.

2.3.1 The General Camera Model

A camera can be modelled as a projective measurement device which maps a 3D point or landmark, ${}^{o}\mathbf{m}$, into a 2D point, or pixel, measurement on the image sensor, ${}^{s}\mathbf{z}$, through some projection function, $\boldsymbol{\pi}(\cdot)$, with additive zero mean Gaussian noise, $\mathbf{n}$. In mathematical terms, this can be expressed as

$$ {}^{s}\mathbf{z} = \boldsymbol{\pi}({}^{c}\mathbf{m}) + \mathbf{n} = \boldsymbol{\pi}({}^{o}\mathbf{x} \circ {}^{o}\mathbf{m}) + \mathbf{n}, \qquad \mathbf{n} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}), \tag{2.18} $$

where ${}^{o}\mathbf{x}$ is the pose (position and attitude) of the camera, ${}^{c}\mathbf{m}$ is the 3D point expressed in the camera coordinate system, and $\boldsymbol{\Sigma}$ is the measurement noise covariance. The expression ${}^{o}\mathbf{x} \circ {}^{o}\mathbf{m}$ is the general expression for the transformation from the object CS to the camera CS, which varies depending on the attitude representation of the camera. Due to the stochastic measurement noise, inversion of Equation 2.18 leads to an expected value for the 3D point in the camera CS

$$ {}^{c}\hat{\mathbf{m}} = E[{}^{c}\mathbf{m}] = \gamma \cdot \boldsymbol{\pi}^{-1}({}^{s}\mathbf{z}), \tag{2.19} $$

where γ is the scale of the projection, which is unobservable from a single observation, and $\boldsymbol{\pi}^{-1}({}^{s}\mathbf{z})$ is the direction from the origin of the camera CS to the estimate of the 3D point, ${}^{c}\hat{\mathbf{m}}$.

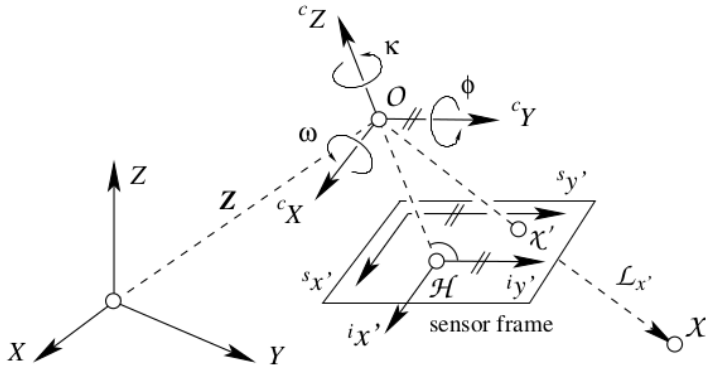


Figure 2.3: The perspective single viewpoint camera model. Note the difference in notation from this project. Courtesy: Förstner and Wrobel (2016)

2.3.2 The Perspective Single Viewpoint Camera Model

The general expression for the transformation from the object CS to the image sensor CS can, in homogeneous coordinates, be expressed as

$$ {}^{s}\tilde{\mathbf{z}} = \begin{bmatrix} {}^{s}x \\ {}^{s}y \\ 1 \end{bmatrix} = {}^{s}_{i}\tilde{\mathbf{H}}\, {}^{i}_{c}\tilde{\mathbf{P}}\, {}^{c}_{o}\tilde{\mathbf{H}} \begin{bmatrix} {}^{o}x \\ {}^{o}y \\ {}^{o}z \\ 1 \end{bmatrix} = {}^{s}_{i}\tilde{\mathbf{H}}\, {}^{i}_{c}\tilde{\mathbf{P}}\, {}^{c}_{o}\tilde{\mathbf{H}}\, {}^{o}\tilde{\mathbf{m}}, \tag{2.20} $$

where ${}^{o}\tilde{\mathbf{m}}$ is the homogeneous representation of a 3D landmark in the object CS, ${}^{s}\tilde{\mathbf{z}}$ is the homogeneous representation of the corresponding 2D point in the sensor CS. ${}^{c}_{o}\tilde{\mathbf{H}}$ is the transform from the object CS to the camera CS, ${}^{i}_{c}\tilde{\mathbf{P}}$ is the projection from camera CS onto the image plane CS, and ${}^{s}_{i}\tilde{\mathbf{H}}$ is the transformation from the image plane CS to the sensor CS. According to the linear perspective SVP camera model, the transformation in Equation 2.20 can, in homogeneous coordinates, be expressed as

$$ {}^{s}\tilde{\mathbf{z}} = \boldsymbol{\pi}({}^{o}\mathbf{x} \circ {}^{o}\tilde{\mathbf{m}}) = {}^{s}_{c}\tilde{\mathbf{K}}\, {}^{c}_{o}\mathbf{R} \begin{bmatrix} \mathbf{I}_{3\times 3} & {}^{c}_{o}\mathbf{t} \end{bmatrix} {}^{o}\tilde{\mathbf{m}} = {}^{s}_{o}\tilde{\mathbf{P}}\, {}^{o}\tilde{\mathbf{m}}, \tag{2.21} $$

where ${}^{s}_{c}\tilde{\mathbf{K}}$ is the linear camera matrix, ${}^{c}_{o}\mathbf{R}$ is the rotation from the object CS to the camera CS, and ${}^{c}_{o}\mathbf{t}$ is the translation from the object CS to the camera CS. Equation 2.21 is known as the direct linear transform and encodes the entire transformation from the object CS to the image sensor CS as one linear matrix multiplication. The linear camera matrix, ${}^{s}_{c}\tilde{\mathbf{K}}$, consists of a linear projection and an affine transformation and can be expressed as

$$ {}^{s}_{c}\tilde{\mathbf{K}} = {}^{s}_{i}\tilde{\mathbf{H}}\, {}^{i}_{c}\tilde{\mathbf{P}} = \begin{bmatrix} 1 & s & c_x \\ 0 & 1+m & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} f_x & s & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}, \tag{2.22} $$

where s is the image sensor shear coefficient, m is the image sensor scale coefficient, f is the focal length, and c_x and c_y are the x- and y-coordinates of the principal point. The parameters defining the linear camera matrix in Equation 2.22 are referred to as the intrinsic parameters, while the parameters defining the rotation and translation in Equation 2.21 are known as the extrinsic parameters.

2.3.3 Nonlinear Corrections

The linear perspective SVP camera model introduced in Section 2.3.2 is not able to model nonlinear effects. A common approach for dealing with nonlinearities is to add a nonlinear correction to the pixels in the sensor CS, a process known as undistortion. Common nonlinearities are; 1) distortion introduced by the camera lens, 2) physical imperfections of the camera lens, 3) planarity imperfections of the image sensor, 4) misalignment of the camera lens with respect to the image sensor. For underwater photogrammetry, nonlinear corrections are commonly utilized to correct for refraction introduced by underwater housings. Generally, the undistortion of the image pixels can be expressed as

$$ {}^{r}\mathbf{z} = {}^{s}\mathbf{z} + \Delta\, {}^{s}\mathbf{z}({}^{s}\mathbf{z}, \mathbf{k}), \tag{2.23} $$

where $\Delta\, {}^{s}\mathbf{z}$ is the nonlinear correction defined in terms of the image sensor coordinate ${}^{s}\mathbf{z}$ and some parameters $\mathbf{k}$. One correction method for lens distortion is the Brown radial distortion model (Brown, 1971). The distortion is modelled as an even-powered polynomial

$$ \Delta\, {}^{s}\mathbf{z}_{\text{radial}}({}^{s}\mathbf{z}, \begin{bmatrix} k_1, k_2, k_3 \end{bmatrix}^{\top}) = \begin{bmatrix} {}^{s}x(k_1 r^2 + k_2 r^4 + k_3 r^6) \\ {}^{s}y(k_1 r^2 + k_2 r^4 + k_3 r^6) \end{bmatrix}, \tag{2.24} $$

where the radius in the image censor CS is defined as

$$ r = \sqrt{({}^{s}x - c_x)^2 + ({}^{s}y - c_y)^2}. \tag{2.25} $$

Another common type of nonlinear correction is tangential distortion, also referred to as decentering distortion (Conrady, 1919). Tangential distortion corrects for distortion effects that are caused by misalignment of the camera lens with respect to the image sensor. Specifically, tangential distortion accounts for distortion effects that are present when the camera lens and image sensor are not parallel and is modelled as

$$ \Delta\, {}^{s}\mathbf{z}_{\text{tangential}}({}^{s}\mathbf{z}, \begin{bmatrix} p_1, p_2 \end{bmatrix}^{\top}) = \begin{bmatrix} 2p_1\, {}^{s}x\, {}^{s}y + p_2(r^2 + 2\, {}^{s}x^2) \\ p_1(r^2 + 2\, {}^{s}y^2) + 2p_2\, {}^{s}x\, {}^{s}y \end{bmatrix}. \tag{2.26} $$

2.3.4 Intrinsic Camera Calibration

In order the estimate the parameters of the linear camera matrix in Equation 2.22 as well as the parameters of the nonlinear corrections, such as the coefficients in Equation 2.24 and Equation 2.26, an intrinsic calibration of the camera must be performed. Zhang's method

is a popular technique for intrinsic calibration due to its flexibility in terms of nonlinearity modelling (Zhang, 2000). The method is based on a plane calibration target and exploits a simplification that follows by setting the object CS origin in the corner of the calibration target and the plane ${}^{o}z = 0$ aligned with the calibration target. Under this condition, a simplified version of Equation 2.21 can be expressed as

$$ {}^{s}\tilde{\mathbf{z}} = \begin{bmatrix} {}^{s}x \\ {}^{s}y \\ 1 \end{bmatrix} = {}^{s}_{o}\tilde{\mathbf{H}} \begin{bmatrix} {}^{o}x \\ {}^{o}y \\ 1 \end{bmatrix} = \begin{bmatrix} \mathbf{h}_1 & \mathbf{h}_2 & \mathbf{h}_3 \end{bmatrix} \begin{bmatrix} {}^{o}x \\ {}^{o}y \\ 1 \end{bmatrix} = {}^{s}_{c}\tilde{\mathbf{K}} \begin{bmatrix} \mathbf{r}_1 & \mathbf{r}_2 & {}^{c}_{o}\mathbf{t} \end{bmatrix} \begin{bmatrix} {}^{o}x \\ {}^{o}y \\ 1 \end{bmatrix}, \quad (2.27) $$

where $\mathbf{r}_1$ and $\mathbf{r}_2$ are the two first column vectors of ${}^{c}_{o}\mathbf{R}$. Zhang's method then exploits the orthonormal property of $\mathbf{r}_1$ and $\mathbf{r}_2$, to formulate the constraints

$$ \mathbf{h}_1^\top \mathbf{G} \mathbf{h}_2 = 0, \quad (2.28a) $$

$$ \mathbf{h}_1^\top \mathbf{G} \mathbf{h}_1 - \mathbf{h}_2^\top \mathbf{G} \mathbf{h}_2 = 0, \quad (2.28b) $$

where the symmetric, positive definite coefficient matrix $\mathbf{G}$ is defined as

$$ \mathbf{G} = \left({}^{s}_{c}\tilde{\mathbf{K}}^{-1} \right)^\top {}^{s}_{c}\tilde{\mathbf{K}}^{-1}. \quad (2.29) $$

Zhang's method finds the coefficient matrix $\mathbf{G}$ and, consequently, the linear camera matrix ${}^{s}_{c}\tilde{\mathbf{K}}$, by minimizing the constraints in Equation 2.28 through singular value decomposition (SVD). Since this solution does not include the nonlinear corrections, Zhang's method solves a maximum likelihood estimation (MLE) problem, where the previously obtained linear camera matrix ${}^{s}_{c}\tilde{\mathbf{K}}$ and no nonlinear corrections are used as the initial guess. The optimization problem is formulated as

$$ \underset{{}^{s}_{c}\tilde{\mathbf{K}}, \mathbf{k}, {}^{c}_{o}\mathbf{R}_n, {}^{c}_{o}\mathbf{t}_n}{\text{minimize}} \sum_{n=1}^{N} \sum_{i=1}^{I} \left\| {}^{s}\mathbf{z}_n^i - \boldsymbol{\pi}\left({}^{s}_{c}\tilde{\mathbf{K}}, \mathbf{k}, {}^{c}_{o}\mathbf{R}_n, {}^{c}_{o}\mathbf{t}_n, {}^{o}\mathbf{m}_n^i \right) \right\|^2, \quad (2.30) $$

where ${}^{s}_{c}\tilde{\mathbf{K}}$ is the linear camera matrix, $\mathbf{k}$ is the parameters defining the nonlinear corrections, ${}^{c}_{o}\mathbf{R}_n$ and ${}^{c}_{o}\mathbf{t}_n$ are the rotation and translation, respectively, between the calibration target and the camera for image n, and ${}^{o}\mathbf{m}_n^i$ is the 3D location for landmark i on the calibration target in image n.

2.4 Photogrammetric Stereo Vision

Similarly to Section 2.3, a large portion of the background material in this section has been found in Förstner and Wrobel (2016). The reader is referred to section 13.2.2, 13.2.3, and 13.2.5 for the background material on relative orientation of dependent image pairs, and

section 13.2.4 and 13.4.1 for stereo image pair triangulation (Förstner and Wrobel, 2016, p.547-606).

For the entirety of this section the relative orientation of image pairs is outlined for the case of dependent image pairs from two cameras, Camera 1 and Camera 2. The convention of using the camera CS of Camera 1 as reference and expressing the orientation of Camera 2 relative to it is used.

2.4.1 Relative Orientation of Dependent Image Pairs

Epipolar geometry, illustrated in Figure 2.4, is a mathematical model which describes the geometric relationship in image pairs. It enables efficient ways of searching for corresponding points between image pairs by reducing the search space from the entire image domain to a straight line in the ideal case.

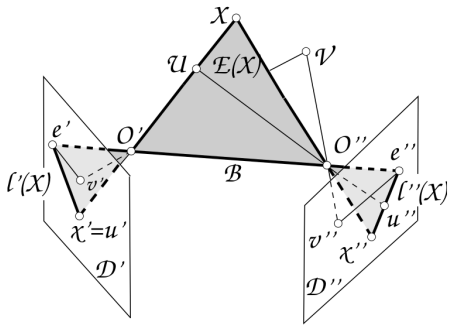


Figure 2.4: Epipolar geometry illustration. Note the difference in notation from this project. Courtesy: Förstner and Wrobel (2016)

According to the perspective SVP camera model, introduced in Section 2.3.2, light rays form straight lines and intersect through the optical center of the camera. As a consequence, the lines from a landmark ${}^{o}\mathbf{m}$ to its projected point in the sensor frames of two cameras, ${}^{s}\mathbf{z}_1$ and ${}^{s}\mathbf{z}_2$, lie in a plane. This is known as the coplanarity constraint and can, for two uncalibrated cameras, be expressed as

$$ {}^{s}\tilde{\mathbf{z}}_1^{\top} \left({}^{s}_{c}\tilde{\mathbf{K}}_1^{\top} \right)^{-1} \mathbf{S}\left({}^{2}_{1}\mathbf{b} \right) {}^{2}_{1}\mathbf{R}^{\top} \left({}^{s}_{c}\tilde{\mathbf{K}}_2 \right)^{-1} {}^{s}\tilde{\mathbf{z}}_2 \equiv {}^{s}\tilde{\mathbf{z}}_1^{\top} \tilde{\mathbf{F}} \, {}^{s}\tilde{\mathbf{z}}_2 = 0, \tag{2.31} $$

where ${}^{s}_{c}\tilde{\mathbf{K}}_1$ and ${}^{s}_{c}\tilde{\mathbf{K}}_2$ are the camera matrices of Camera 1 and Camera 2, respectively, $\mathbf{S}({}^{2}_{1}\mathbf{b})$ is the skew-symmetric matrix of the baseline vector, ${}^{2}_{1}\mathbf{R}$ is the rotation matrix from Camera 1 to Camera 2, and $\tilde{\mathbf{F}}$ is the fundamental matrix of the camera pair. In the case for two calibrated cameras, the coplanarity constraint becomes

$$ {}^{c}\tilde{\mathbf{m}}_1^{\top} \mathbf{S}({}^{2}_{1}\mathbf{b}) \, {}^{2}_{1}\mathbf{R}^{\top} \, {}^{c}\tilde{\mathbf{m}}_2 \equiv {}^{c}\tilde{\mathbf{m}}_1^{\top} \tilde{\mathbf{E}} \, {}^{c}\tilde{\mathbf{m}}_2 = 0, \tag{2.32} $$

where $\tilde{\mathbf{E}}$ is the essential matrix of the camera pair. By comparing Equation 2.31 and Equation 2.32 one can see that fundamental - and essential matrix can be related by the following expression

$$\tilde{\mathbf{E}} = {}^{s}_{c}\tilde{\mathbf{K}}_1^{\top}\tilde{\mathbf{F}}\,{}^{s}_{c}\tilde{\mathbf{K}}_2. \tag{2.33}$$

Since the fundamental and essential matrix encodes information about the extrinsic parameters of the camera pair, i.e. the baseline vector ${}^{2}_{1}\mathbf{b}$ and rotation matrix ${}^{2}_{1}\mathbf{R}$, they can be used as a mean for extrinsic calibration. A direct solution to estimation of the fundamental is the 8-point algorithm (Longuet-Higgins, 1981). Due to measurement noise and quantization errors, the coplanarity constraint cannot be satisfied exactly. Therefore, the 8-point algorithms finds the fundamental matrix by solving the following optimization problem for N pairs of corresponding image points, ${}^{s}\mathbf{z}_{1,n}$ and ${}^{s}\mathbf{z}_{2,n}$

$$\underset{\tilde{\mathbf{F}}}{\text{minimize}} \quad \sum_{n=1}^{N} {}^{s}\mathbf{z}_{1,n}\tilde{\mathbf{F}}\,{}^{s}\mathbf{z}_{2,n}, \tag{2.34a}$$

$$\text{subject to } \operatorname{rank}(\tilde{\mathbf{F}}) = 2. \tag{2.34b}$$

The optimal solution of Equation 2.34 is found by means of SVD and forcing the smallest singular value of $\tilde{\mathbf{F}}$ to be zero, ensuring that its rank is 2. The procedure for estimating the essential matrix is simila,r except for an additional constraint that the two non-zero singular values are identical. In practice, the normalized version of the 8-point algorithm is more commonly used due to its improved numerical stability (Hartley, 1997). Another approach for finding the essential matrix is the 5-point algorithm, which is considered the golden standard in the case of calibrated cameras (Nister, 2004). The algorithm is often coupled with outlier rejection through random sample consensus (RANSAC) due to its low amount of needed inliers (Fischler and Bolles, 1981).

2.4.2 Stereo Image Pair Triangulation

Given the relative orientation between two calibrated cameras, triangulation is the problem of estimating the three-dimensional coordinates of a landmark in the camera frame, ${}^{c}\mathbf{m}$, from the corresponding points in two rectified camera images, ${}^{r}\mathbf{z}_1$ and ${}^{r}\mathbf{z}_2$. The stereo normal case is an idealized case, where the cameras face the same way, the optical axis of the two cameras are parallel, and the only translation between them is an offset in the x-direction in the camera CS of Camera 1. By performing stereo image pair rectification, i.e. projecting the image pair to be in a common plane, the constraints of the stereo normal case can be satisfied approximately. In the stereo normal case, the epipolar lines are horizontal lines, and the stereo triangulation problem is given by the stereo intersection theorem as

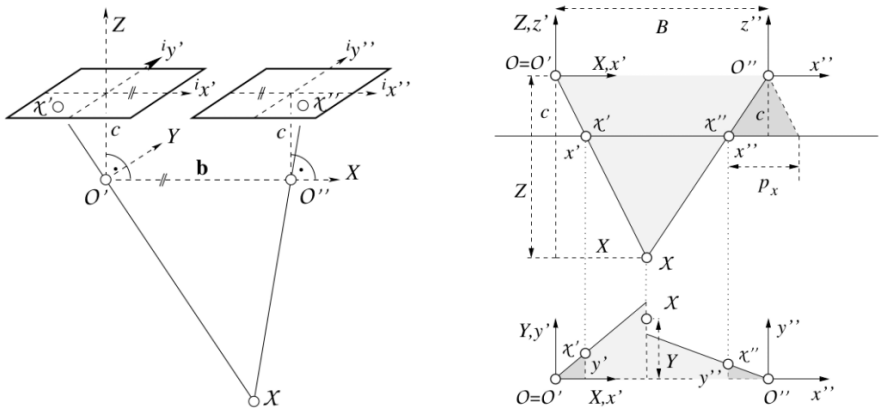


Figure 2.5: Stereo normal case illustration. Note the difference in notation from this project. Courtesy: Förstner and Wrobel (2016)

$$^{c}x_1 = {}^{r}x_1 \cdot \frac{b_x}{-p_x}, \tag{2.35a}$$

$$^{c}y_1 = \frac{{}^{r}y_1 + {}^{r}y_2}{2} \cdot \frac{b_x}{-p_x}, \tag{2.35b}$$

$$^{c}z_1 = f \cdot \frac{b_x}{-p_x}, \tag{2.35c}$$

where $^{c}x_1$, $^{c}y_1$ and $^{c}z_1$ are the x-, y- and z-coordinate of a landmark expressed in the camera frame of Camera 1, b_x is the x-component of the baseline vector ${}^{2}_{1}\mathbf{b}$, $^{r}x_1$, $^{r}y_1$, $^{r}x_2$, and $^{r}y_2$ are the x- and y-coordinate of the pixel points corresponding to the landmark expressed in the rectified sensor frame of Camera 1 and Camera 2, respectively. The quantity p_x is known as the x-disparity or x-parallax, and is defined as

$$p_x = {}^{r}x_2 - {}^{r}x_1. \tag{2.36}$$

By isolating the pixel dependent information in a vector, one can set up a depth mapping transformation on a per pixel basis. The resulting triangulation method can be expressed in homogeneous coordinates as

$$^{c}\tilde{\mathbf{m}}_1 = -p_x \begin{bmatrix} ^{c}x_1 \\ ^{c}y_1 \\ ^{c}z_1 \\ 1 \end{bmatrix} = \begin{bmatrix} b_x & 0 & 0 & 0 \\ 0 & b_x & 0 & 0 \\ 0 & 0 & b_x f & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix} \begin{bmatrix} ^{r}x_1 \\ ^{r}y_1 \\ 1 \\ p_x \end{bmatrix} = {}^{c}_{r}\tilde{\mathbf{D}} \begin{bmatrix} ^{r}\tilde{\mathbf{z}}_1 \\ p_x \end{bmatrix}, \tag{2.37}$$

where the homogeneous transformation ${}^{c}_{r}\tilde{\mathbf{D}}$ is given for a calibrated stereo camera, the homogeneous rectified image coordinates ${}^{r}\tilde{\mathbf{z}}_1$ are given for the first image and the disparity p_x is computed from the corresponding point in the second image. Using Gauss' law of propagation of uncertainty, the uncertainty of the 3D coordinates of the triangulated point is given as

$$\sigma_{^{c}x} = \frac{^{c}z}{f}\sigma_{^{r}x}, \tag{2.38a}$$

$$\sigma_{^{c}y} = \frac{\sqrt{2}}{2} \cdot \frac{^{c}z}{f}\sigma_{^{r}y}, \tag{2.38b}$$

$$\sigma_{^{c}z} = \frac{^{c}z}{p_x}\sigma_{p_x}, \tag{2.38c}$$

where $\sigma_{^{r}x}$, $\sigma_{^{r}y}$ and σ_{p_x} are the uncertainties of the rectified image coordinates and the disparity, respectively. Stereo matching, i.e. the computation of the disparity p_x, has been a research topic within the computer vision community for decades. Traditional methods have solved the stereo matching problem as a multi-stage optimization problem, generally consisting of four stages; 1) matching cost computation, 2) cost aggregation, 3) disparity selection, and 4) disparity refinement. These methods rely on matching image patches by sliding windows along epipolar lines, as well as window filters to reduce pixel matching uncertainty, disparity noise, and disparity discontinuities (Chen et al., 2015; Ma et al., 2013; Colodro-Conde et al., 2014; Werner et al., 2014). Some of the short-comings of these traditional multi-stage optimization methods are, generally, trade-off between computational complexity and accuracy, handcrafted image features, and filter parameters that need to be empirically decided per dataset. Additionally, the accuracy of the traditional methods suffer in regions of high texture, low texture, and occlusion.

Similarly to a multitude of computer vision tasks, the state-of-the-art in stereo matching has moved towards deep-learning (DL) methods to compensate for the aforementioned shortcomings. The first adaptations of DL to the stereo matching problem tended to solve one of the four stages of the multi-stage optimization, but recently the most successful approaches have been end-to-end disparity networks, which use DL to estimate the disparity map directly from rectified stereo image pairs (Laga et al., 2020). Disparity networks do, in general, consist of a sequence of smaller neural network which solve smaller, specific tasks. For instance, disparity networks commonly utilize convolutional neural networks (CNNs) pre-trained on generic image datasets, such as ResNet, for feature extraction (He et al., 2016). In addition to disparity maps, architectures have also been proposed to estimate semantic and confidence information. An example of this is DispNet3, which jointly estimates disparity and occlusion maps from rectified image pairs (Ilg et al., 2018). While supervised learning methods, such as DeepPruner and DLANet, have been a hot research topics with promising results, current methods are data-greedy and do not train well on synthetic information (Duggal et al., 2019; Yin et al., 2019). As such, self-supervised and unsupervised methods, such as SegStereo and UnsupAdpt, respectively, have gotten more attention in recent years due to the possibility to transfer learning across domains, especially from synthetic to real world data (Yang et al., 2018; Tonioni et al., 2017).

2.5 Visual Simultaneous Localization and Mapping

2.5.1 The Full SLAM Problem Formulation

The SLAM problem is formulated as simultaneously estimating a trajectory of poses $\mathbf{x}_{1:k}$ (localization), and a map $\mathbf{m}$ of the environment (mapping) from a series of exteroceptive sensor measurements $\mathbf{z}_{1:k}$, and odometry inputs $\mathbf{u}_{1:k}$. Expressed in a probabilistic manner, the full SLAM problem can be expressed as finding the probability distribution

$$p(\mathbf{x}_{1:L}, \mathbf{m}|\mathbf{z}_{1:L}) \propto \Big(\prod_{k=1}^{L} p(\mathbf{z}_k|\mathbf{x}_k, \mathbf{m}) p(\mathbf{x}_k|\mathbf{u}_k, \mathbf{x}_{k-1}) \Big) p(\mathbf{x}_0, \mathbf{m}), \tag{2.39}$$

where L is a time window over which the SLAM problem should be solved, $p(\mathbf{z}_k|\mathbf{x}_k, \mathbf{m})$ is a stochastic measurement model, $p(\mathbf{x}_k|\mathbf{u}_k, \mathbf{x}_{k-1})$ is a stochastic process model, and $p(\mathbf{x}_0, \mathbf{m})$ is a prior (Brekke, 2020, p.212). In SLAM, the measurements and map representation are, generally, sets. For this project, the measurement set and map set notations are defined as

$$\mathbf{z}_k = \{\mathbf{z}_k^1, ..., \mathbf{z}_k^{N_k}\}, \qquad \mathbf{m} = \{\mathbf{m}^1, ..., \mathbf{m}^M\}, \tag{2.40}$$

where N_k is the size of the measurement set at timestep k, M is the size of the map, $\mathbf{z}_k^i$ is measurement sample i obtained at timestep k, and $\mathbf{m}^j$ is landmark j.

2.5.2 The Full SLAM Standard Model

In the standard model for the full SLAM problem, all the probability distributions on the right-hand side of Equation 2.39 are assumed Gaussian and the map is assumed to be static. In this case, the negative logarithm of the probability distribution becomes,

$$\begin{aligned} -\ln\big(p(\mathbf{x}_{1:L}, \mathbf{m}|\mathbf{z}_{1:L})\big) \propto \|\boldsymbol{\eta}_0 - \hat{\boldsymbol{\eta}}_0\|^2_{\mathbf{P}_0} + \sum_{k=1}^{L} \big\|\mathbf{u}_k - \mathbf{f}^{-1}(\mathbf{x}_k, \mathbf{x}_{k-1})\big\|^2_{{}^{\mathbf{u}}\boldsymbol{\Sigma}_k} \\ + \sum_{k=1}^{L} \sum_{(i,j)\in\mathcal{M}_k} \big\|\mathbf{z}_k^i - \mathbf{h}(\mathbf{x}_k, \mathbf{m}^j)\big\|^2_{{}^{\mathbf{z}}\boldsymbol{\Sigma}_{ik}}, \end{aligned} \tag{2.41}$$

where $\hat{\boldsymbol{\eta}}_0$ is the prior mean, $\mathbf{P}_0$ is the prior covariance, $\mathbf{f}(\mathbf{x}_{k-1}, \mathbf{u}_k)$ is the transition function of the process model, ${}^{\mathbf{u}}\boldsymbol{\Sigma}_k$ is the odometry covariance, $\mathcal{M}_k$ is a set of matched measurement and landmark indices, $\mathbf{z}_k^i$ is a measurement associated with landmark $\mathbf{m}^j$ from pose $\mathbf{x}_k$ with covariance ${}^{\mathbf{z}}\boldsymbol{\Sigma}_{ik}$ and $\mathbf{h}(\mathbf{x}_k, \mathbf{m}^j)$ is the observation function of the measurement model. Note the notation $\|\mathbf{x} - \boldsymbol{\mu}\|^2_{\mathbf{P}}$, which is the short-hand notation for the squared Mahalanobis distance. The squared Mahalanobis distance is defined as

$$\|\mathbf{x} - \boldsymbol{\mu}\|^2_{\mathbf{P}} = (\mathbf{x} - \boldsymbol{\mu})^\top \mathbf{P}^{-1} (\mathbf{x} - \boldsymbol{\mu}), \tag{2.42}$$

where $\mathbf{x}$ is a random variable with mean $\boldsymbol{\mu}$ and covariance $\mathbf{P}$ (Brekke, 2020, p. 212). By minimizing the right-hand side of Equation 2.41 with respect to the pose trajectory $\mathbf{x}_{1:L}$ and the map $\mathbf{m}$, the maximum a posterior (MAP) estimate to the full SLAM problem is obtained.

2.5.3 Graph Optimization

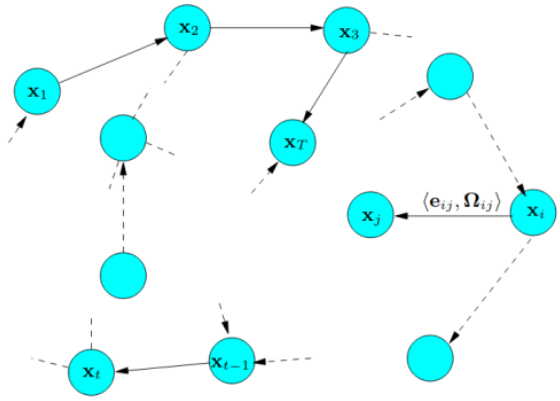


Figure 2.6: Graph representation for a nonlinear least squares pose optimization problem. Courtesy: Grisetti et al. (2010)

The SOTA for solving the full SLAM problem is to solve it as a nonlinear, directed graph optimization problem (Grisetti et al., 2010). Without specifically considering the full SLAM problem, nonlinear least squares problems, like the minimization of Equation 2.41, can be represented as a nonlinear, directed graph consisting of vertices $\mathbf{v}_i$ and $\mathbf{v}_j$, and edges $\mathbf{w}_{ij}$ connecting them, as illustrated in Figure 2.6. Considering the error function $\mathbf{e}(\mathbf{v}_i, \mathbf{v}_j, \mathbf{w}_{ij})$ for a pair of connected edges and a vertex, (i, j), in the graph $\mathcal{G}$, the error of the graph can be written as

$$f_{\mathcal{G}}(\mathbf{s}) = \sum_{(i,j)\in\mathcal{G}} \|\mathbf{e}(\mathbf{v}_i, \mathbf{v}_j, \mathbf{w}_{ij})\|^2_{\boldsymbol{\Omega}_{ij}^{-1}} = \sum_{(i,j)\in\mathcal{G}} \|\mathbf{e}_{ij}(\mathbf{s})\|^2_{\boldsymbol{\Omega}_{ij}^{-1}}, \tag{2.43}$$

where $\boldsymbol{\Omega}_{ij}$ is the error information matrix and $\mathbf{s}$ is the state of the graph. Since the error function is nonlinear, the optimal configuration of the graph, i.e. the vertices and edges which minimizes the graph error, has to be found through iteration. By considering an initial graph state $\check{\mathbf{s}}$, and small increments around it $\Delta\mathbf{s}$, the graph error function can be approximated as

$$
\begin{aligned}
f_{\mathcal{G}}(\check{\mathbf{s}} + \Delta\mathbf{s}) &\approx \sum_{(i,j)\in\mathcal{G}} \|\mathbf{e}_{ij}(\check{\mathbf{s}}) + \mathbf{J}_{ij}\Delta\mathbf{s}\|^2_{\mathbf{\Omega}_{ij}^{-1}} \\
&= \sum_{(i,j)\in\mathcal{G}} \Big[\underbrace{\mathbf{e}_{ij}(\check{\mathbf{s}})^\top \mathbf{\Omega}_{ij} \mathbf{e}_{ij}(\check{\mathbf{s}})}_{c_{ij}} + 2 \underbrace{\mathbf{e}_{ij}(\check{\mathbf{s}})^\top \mathbf{\Omega}_{ij} \mathbf{J}_{ij}}_{\mathbf{b}_{ij}} \Delta\mathbf{s} + \Delta\mathbf{s}^\top \underbrace{\mathbf{J}_{ij}^\top \mathbf{\Omega}_{ij} \mathbf{J}_{ij}}_{\mathbf{A}_{ij}} \Delta\mathbf{s} \Big] \\
&= c + 2\mathbf{b}^\top \Delta\mathbf{s} + \Delta\mathbf{s}^\top \mathbf{A} \Delta\mathbf{s},
\end{aligned}
\tag{2.44}
$$

where $\mathbf{J}_{ij}$ is the Jacobian of the error function for edge $\mathbf{w}_{ij}$, connecting vertices $\mathbf{v}_i$ and $\mathbf{v}_j$ (Kümmerle et al., 2011). By differentiation of the graph error function $f_{\mathcal{G}}(\cdot)$, the optimal increment, $\Delta\mathbf{s}^*$, can be found by solving the linear system

$$
\mathbf{A}\Delta\mathbf{s}^* = -\mathbf{b}. \tag{2.45}
$$

By adding the optimal increment $\Delta\mathbf{s}^*$ to the initial graph state $\check{\mathbf{s}}$, the starting point for the next iteration of the minimization process is obtained. The Gauss-Newton algorithm is the classical method for minimizing $f_{\mathcal{G}}(\cdot)$, but more modern solvers utilize the Levenberg-Marquardt algorithm, which is more robust due to the introduction of a damped version of the linear system

$$
(\mathbf{A} + \gamma\mathbf{I})\Delta\mathbf{s}^* = -\mathbf{b}, \tag{2.46}
$$

where γ is a dampening factor that can be set dynamically for each iteration (Levenberg, 1944).

2.5.4 Bundle Adjustment

For V-SLAM, the prior and odometry terms of Equation 2.41 are disregarded, changing the problem to a MLE problem. Additionally, the observation function $\mathbf{h}(\mathbf{x}_k, \mathbf{m}^j)$ is equal to the camera projection function outlined in Section 2.3.1, the poses $\mathbf{x}_k$ are equal to the camera poses, the measurements are equal to 2D points in the camera sensor frame ${}^s\mathbf{z}_k^i$, and the landmarks $\mathbf{m}^i$ are 3D landmarks backprojected into the object CS, ${}^o\mathbf{m}^j$. In this case, the MLE optimization problem becomes

$$
\underset{\mathbf{x}_{1:L},\mathbf{m}}{\text{minimize}} \quad \sum_{k=1}^{L} \sum_{(i,j)\in\mathcal{M}_k} \left\| {}^s\mathbf{z}_k^i - \boldsymbol{\pi}(\mathbf{x}_k, \mathbf{m}^j) \right\|^2_{\mathbf{z}\mathbf{\Sigma}_{ik}} = \|\mathbf{e}_{\pi,ik}\|^2_{\mathbf{z}\mathbf{\Sigma}_{ik}}, \tag{2.47}
$$

where $\mathbf{e}_{\pi,}$ is the reprojection error, and ${}^{\mathbf{z}}\mathbf{\Sigma}_{ik}$ is the measurement noise covariance. Equation 2.47 is known as bundle adjustment (BA) for calibrated cameras (Triggs et al., 2000). Comparing Equation 2.47 to Equation 2.43, it is evident that BA can be represented as directed, nonlinear graph where the camera poses $\mathbf{x}_k$ and landmarks $\mathbf{m}^j$ are vertices, and

the associated measurements, ${}^{s}\mathbf{z}_k^i$, are edges. Since the first implementation of BA in real-time V-SLAM with the ORB-SLAM algorithm, BA has been the golden standard for graph-based V-SLAM methods (Mur-Artal et al., 2015). In fact, tailor-engineered BA solvers have been developed specifically for the V-SLAM problem in order to increase real-time capabilities (Liu et al., 2018).

2.5.5 OpenVSLAM

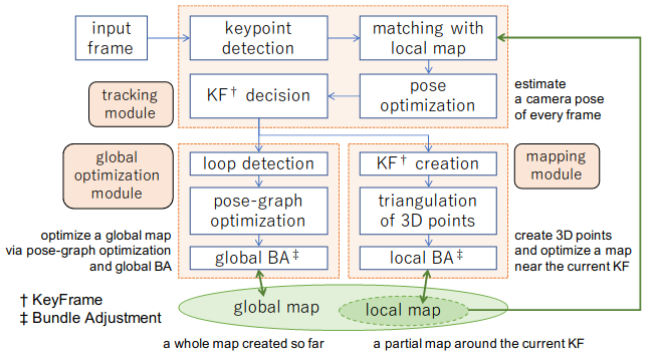


Figure 2.7: Overview of the OpenVSLAM algorithm architecture. Courtesy: Sumikura et al. (2019)

OpenVSLAM is an open-source V-SLAM framework, which won the ACM Multimedia 2019 Open Source Software Competition (Sumikura et al., 2019). OpenVSLAM is inspired by indirect, sparse graph-based V-SLAM algorithms ORB-SLAM, ORB-SLAM2, ProSLAM, and UcoSLAM (Mur-Artal et al., 2015; Mur-Artal and Tardós, 2017; Schlegel et al., 2018; Muñoz-Salinas and Medina-Carnicer, 2020). Unlike previous graph-based V-SLAM algorithms, OpenVSLAM is implemented to be versatile in terms of camera modelling, with implementations of perspective, fisheye, and equirectangular camera models for monocular, stereo, and red-green-blue-depth (RGBD) setups. OpenVSLAM also allows users to implement custom camera models from templates, a valuable feature in underwater photogrammetry due to the adaptation of refractive camera models. Additionally, OpenVSLAM provides functionality to save and load the map for offline evaluation and, potentially, map reuse. By employing the MessagePack serialization format, saved OpenVSLAM maps can be analyzed in a variety of different programming languages (Furuhashi, 2019).

The pose representation adapted by OpenVSLAM is a vector consisting of the camera position in the object CS ${}^{o}\mathbf{p}_k$, and the unit quaternion ${}^{o}_{c}\mathbf{q}_k$ representing the rotation from the camera CS to the object CS. The OpenVSLAM map representation is a set of 3D landmarks ${}^{o}\mathbf{m}^i$ represented in the object CS. By adopting the transformation and coordinate system representation utilized in this project, the pose and map are expressed as

$$\mathbf{x}_k = \begin{bmatrix} {}^{o}\mathbf{p}_k \\ {}^{o}_{c}\mathbf{q}_k \end{bmatrix}, \qquad \mathbf{m} = \{{}^{o}\mathbf{m}^1, ..., {}^{o}\mathbf{m}^M\}. \tag{2.48}$$

The OpenVSLAM architecture can be seen in Figure 2.7, and is roughly divided into three modules; 1) the tracking module, 2) the mapping module, and 3) the global optimization module. The tracking module handles frames (single images for monocular, image pairs for stereo, and image and depth map for RGBD), by performing image processing, feature detection and description, feature matching, pose estimation, and keyframe creation. Keyframes are frames which get inserted into the pose-graph, and whose pose has a certain variation from earlier keyframes. In the mapping modules, triangulated feature points from keyframes are reprojected into 3D landmarks and inserted into the map, extending it. Landmarks in close proximity to the last keyframe are refined by performing local BA. The global optimization module performs loop detection, pose-graph optimization, as well as refinement of the map by global BA.

Due to the complexity of the OpenVSLAM algorithm, only the most essential subroutines are covered in detail in this project. Specifically, the feature detection and description, and pose optimization of the tracking module, local BA of the mapping module, as well as the loop detection, pose-graph optimization, and global BA of the global optimization module are covered in more detail.

2.5.6 Feature Detection and Description

OpenVSLAM uses the oriented FAST and rotated BRIEF (ORB) feature for local image feature detection and description (Rublee et al., 2011). ORB is a combined feature detector and descriptor, utilizing oriented FAST (Rosten et al., 2010) for feature detection and rotated BRIEF (Calonder et al., 2010) for feature description. FAST performs corner detection by using a circular ring around a center pixel to perform segment tests. These segment tests considers various combinations of the states of the pixel in the circular ring. The state of a pixel j in the circular arc is determined as,

$$S_j = \begin{cases} -1, & I_j - I_c \leq -\kappa \\ 0, & \|I_j - I_c\| < \kappa \\ 1, & I_j - I_c \geq \kappa \end{cases}, \tag{2.49}$$

where κ is a threshold value, I_j and I_c are intensities for the pixel and central pixel, respectively. Since FAST generally yields quite large responses at edges and does not provide a corner measure, the ORB feature detector computes the Harris corner response at each of the detected points in order to discard edge points (Harris and Stephens, 1988). Additionally, in order to assess features on multiple scales, ORB employs a image pyramid to the images. After corner points are detected, ORB computes BRIEF descriptors for the square image patches around the corner points. The BRIEF descriptor is defined as

$$f_{\text{BRIEF}}(\mathbf{H}) = \sum_{1 \leq i \leq n} 2^{i-1} \tau(\mathbf{H}; \mathbf{z}_{1,i}, \mathbf{z}_{2,i}), \qquad \mathbf{z}_{1,i}, \mathbf{z}_{2,i} \in \mathbf{H}, \tag{2.50a}$$

$$\tau(\mathbf{H}; \mathbf{z}_1, \mathbf{z}_2) = \begin{cases} 1, & \mathbf{H}(\mathbf{z}_1) < \mathbf{H}(\mathbf{z}_2) \\ 0, & \mathbf{H}(\mathbf{z}_1) \geq \mathbf{H}(\mathbf{z}_2) \end{cases}, \tag{2.50b}$$

where $\mathbf{H}$ is an image patch of size n, $\tau(\cdot)$ is a binary test, and $\mathbf{z}_1$ and $\mathbf{z}_2$ are pixels in the image patch. In binary form, the BRIEF feature descriptor simply becomes a string of zeros and ones of length n, where each of the entries is the result from the corresponding binary test $\tau(\cdot)$. Since the BRIEF descriptor only consists of a series of binary test, it is extremely fast to compute and compare to other descriptors. However, one of the downsides of the BRIEF descriptors is that it is not rotational invariant. To account for this, ORB adopts the rotated BRIEF descriptor, which utilizes the image patch center and centroid to define the patch orientation.

2.5.7 Pose Optimization

When new frames are inserted into the OpenVSLAM algorithm, the initial pose estimate is optimized by performing pose-only BA. Specifically, this is performed by solving the following graph optimization problem

$$\underset{\mathbf{x}_k}{\text{minimize}} \quad \sum_{(i,j)\in\mathcal{M}_k} \left\|\mathbf{z}_k^i - \boldsymbol{\pi}(\mathbf{x}_k, \mathbf{m}^j)\right\|^2_{\mathbf{z}\boldsymbol{\Sigma}_{ij}}, \tag{2.51}$$

where the landmarks are fixed. The pose optimization is performed in order to improve the tracking-based pose estimate, which is, generally, less accurate and robust than the BA-based pose estimate.

2.5.8 Local Bundle Adjustment

After the pose of the last frame has been optimized and accepted as a keyframe, local BA is performed in order to filter out landmark outliers and, consequently, prune the map. The local BA is

$$\underset{\mathbf{x}_k,\mathbf{m}}{\text{minimize}} \quad \sum_{k\in\mathcal{C}^*\cup\mathcal{K}} \sum_{(i,j)\in\mathcal{M}_k} \left\|\mathbf{z}_k^i - \boldsymbol{\pi}(\mathbf{x}_k, \mathbf{m}^j)\right\|^2_{\mathbf{z}\boldsymbol{\Sigma}_{ij}}, \tag{2.52}$$

where $\mathcal{K}$ is the set of keyframes which share landmarks with the newly inserted keyframe, and $\mathcal{C}^*$ is the covisibility graph nodes connected to the keyframes in $\mathcal{K}$. The covisibility graph was first introduced by the ORB-SLAM algorithm, and is utilized as an efficient mean to get covisibility information without having to search the entire graph of poses (Mur-Artal et al., 2015). The covisibility graph consists of pose vertices and edges between the poses which have overlapping FOVs.

2.5.9 Loop Detection

To perform loop detection, OpenVSLAM adopts the bag of (visual) words (BOW) framework DBoW2 (Galvez-López and Tardos, 2012). BOW is a method to compactly describe large sets of images, as well as perform similarity comparisons between images. BOW does this by discretizing the feature descriptor space into a finite set of W descriptors. The set is referred to as a vocabulary, while the feature descriptors in the set are referred to

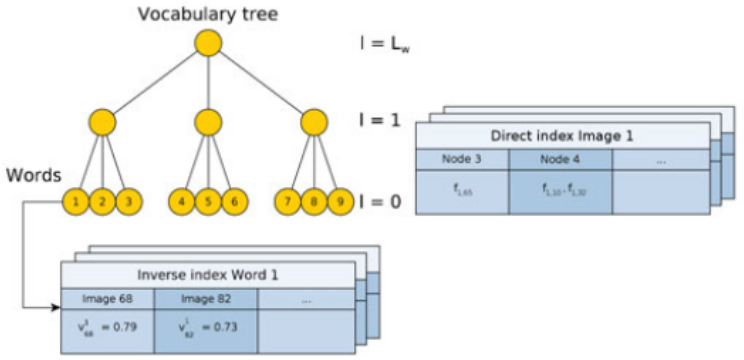


Figure 2.8: Illustration of the DBoW2 vocabulary tree, inverse indices, and direct indices. Courtesy: Galvez-López and Tardos (2012)

as words. This discretization allows an image i to be described as a histogram vector $\mathbf{v}_i$, where the occurrences of each of the visual words in the image are counted. For hierarchical BOW, the vocabulary is represented as a vocabulary tree, which is calculated off-line from a training set of images. For each image, features are extracted and the corresponding feature descriptors are calculated. The descriptors are then clustered into k_W clusters, which form the first set of nodes in the tree. This process is repeated for the descriptors associated with each node to form subsequent levels in the tree, until the tree has W leave nodes, each representing a visual word. Since frequently recurring words are ill-fit for discrimination, a weight is usually calculated for each word to emphasis strong discriminators. A common index for weighting the words is the term frequency-inverse document frequency (TFIDF),

$$\text{TFIDF}_w = \frac{n_{wi}}{n_i} \ln\Big(\frac{N}{n_w}\Big), \tag{2.53}$$

where n_{wi} is the number of occurrences of word w in image i, n_i is the number of words in image i, n_w is the number of images containing word w in the database, and N is the number of images in the database (Sivic and Zisserman, 2003). To compute the BOW vector $\mathbf{v}_i$ of a image i, one simply traverse the feature descriptors through the tree, select the node which minimizes the Hamming distance at every intermediate level, and keep count of the leave nodes that the descriptors end up on. When comparing two images, i and j, by their corresponding BOW vectors, $\mathbf{v}_i$ and $\mathbf{v}_j$, a distance measure such as the cossine distance, has to be used. The cossine distance between the vectors $\mathbf{v}_i$ and $\mathbf{v}_j$ is defined as

$$d_{\cos}(\mathbf{v}_i, \mathbf{v}_j) = 1 - \text{cossim}(\mathbf{v}_i, \mathbf{v}_j) = 1 - \frac{\mathbf{v}_i^\top \mathbf{v}_j}{\|\mathbf{v}_i\| \, \|\mathbf{v}_i\|}. \tag{2.54}$$

To query the database in an efficient manner, DBoW2 keeps an inverse index for each

word, which yields the images that contain that word, in addition to a direct index for each image, which yields the words that are in the image.

2.5.10 Pose-Graph Optimization

OpenVSLAM performs pose-graph optimization after a successful loop detection. Pose-graph optimization corrects for the accumulated drift in the loop by the following BA

$$\underset{\mathbf{x}_k}{\text{minimize}} \quad \sum_{k \in \mathcal{C}} \sum_{(i,j) \in \mathcal{M}} \left\| \mathbf{z}_k^i - \boldsymbol{\pi}(\mathbf{x}_k, \mathbf{m}^j) \right\|^2_{\mathbf{z}\boldsymbol{\Sigma}_{ij}}, \tag{2.55}$$

where $\mathcal{C}$ is the covisibility graph for all the poses in the loop. For the pose-graph optimization, the two poses which connected the loop are kept fixed.

2.5.11 Global Bundle Adjustment

After having performed pose-graph optimization, OpenVSLAM performs global BA

$$\underset{\mathbf{x}_{1:L}, \mathbf{m}}{\text{minimize}} \quad \sum_{(i,j) \in \mathcal{M}} \left\| \mathbf{z}_k^i - \boldsymbol{\pi}(\mathbf{x}_k, \mathbf{m}^j) \right\|^2_{\mathbf{z}\boldsymbol{\Sigma}_{ij}}, \tag{2.56}$$

which solves BA over the entire trajectory and map. Global BA is performed in order to refine the map after the accumulated drift has been corrected. By pruning the map for outliers and correcting for the effect of the loop closure, the global BA ensures that the map is globally consistent and that the number of landmarks is kept low, which is critical for the ability to perform long-term mapping.

Chapter 3

Method

Apart from the technical information and description of the ZED stereo camera, in Section 3.1.4, which has been adopted from Larsen (2020a), Chapter 3 is exclusively the work of this project.

3.1 Vessels, Sensors, and Systems

3.1.1 R/V Gunnerus

Figure 3.1: The NTNU research vessel, R/V Gunnerus. Courtesy: NTNU (2006)

R/V Gunnerus, seen in Figure 3.1, is a medium-sized research vessel, owned and operated by the Norwegian University of Science and Technology (NTNU) (NTNU, 2006). The ship is equipped with a large variety of sensors designed for research activities within the fields of geology, biology, chemistry, archaeology, oceanography, and aquaculture. As a part of its navigation system, the ship is equipped with a Furuno Navigator GP-

90 system consisting of a global positioning system (GPS) receiver and a display and processing unit. Additionally, the ship is equipped with a Kongsberg Seapath 300 system, combining the GPS signals from the Furuno Navigator GP-90 with measurements from a Kongsberg Seatex motion reference unit (MRU) for attitude, and position estimates. To provide position measurements to vehicles and equipment deployed under water, the ship is equipped with a Kongsberg HiPAP 500, a super short base line (SSBL) APS.

3.1.2 ROV SUB-Fighter 30K

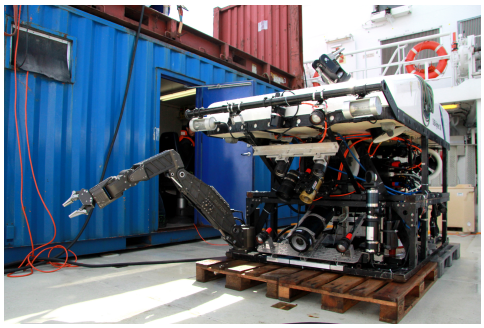


Figure 3.2: The SUB-Fighter 30K ROV. Courtesy: NTNU AURLab

The remotely operated vehicle (ROV), SUB-Fighter 30K, is a work class ROV manufactured by the Norwegian technology company Sperre AS (Dukan, 2014). The ROV can be seen in Figure 3.2. The ROV is equipped with a Kongsberg HPR acoustic transponder, which is coupled with the APS of R/V Gunnerus. In conjunction with the GPS on board the ship, the APS configuration provides position measurements to the ROV in the form of latitude, longitude, and depth. In addition to the acoustic transponder, the ROV is equipped with a set of sensors providing dead reckoning navigation data. Among others, a XSens MTi-100 inertial measurement unit (IMU), a Teledyne RDI Workhorse Navigator WHN 1200 Doppler velocity log (DVL), and a Paroscientific Digiquartz pressure sensor. The XSens MTi-100 IMU contains a gyroscope, a barometer, a magnetometer, and an accelerometer, providing heading and turn rate, pressure, magnetic field, and linear acceleration measurements, respectively. The DVL is mounted downwards, providing velocity measurements relevant for current estimation and sea bottom detection, as well as altitude measurements. The Digiquartz pressure sensor provides pressure measurements, which can be used to estimate the depth of the ROV. In addition to the navigation sensors, the ROV is equipped with one HD camera, one SD camera, a manipulator arm, two HMI lamps, and four Halogen lamps, making it suited for intrusive and non-intrusive sea bottom surveys. The technical specification for the various sensors can be found in Appendix A.

3.1.3 Navigation System Topology

Figure 3.3 shows the navigation system topology of the SUB-Fighter 30K ROV and R/V Gunnerus. In the topology diagram, the green blocks are system units, the orange are sen-

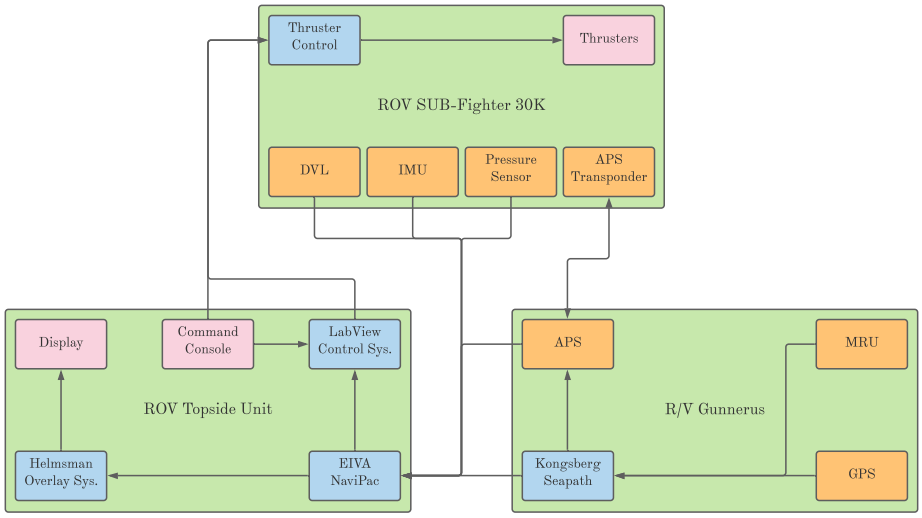


Figure 3.3: Navigation system topology and signal flow for the ROV SUB-Fighter 30K and R/V Gunnerus.

sor systems, the blue are software systems, and the pink are hardware units. On board the ship, the MRU- and GPS measurements are processed by the Kongsberg Seapath motion reference system (MRS) to estimate the ship's position and attitude. The MRS estimates is used in the APS in order to provide absolute position measurements of the APS transponder on board the ROV. The measurements from the APS, the estimates from the Kongsberg Seapath system, and the dead reckoning sensor measurements from the ROV are input to the EIVA NaviPac software, which is run on a laptop in the ROV topside unit. NaviPac relays a subset of the navigation data to the Helmans overlay system, which displays information to the ROV operators during operation. NaviPac also relays navigation data to the LabView control system, which uses the navigation data for both state estimation- and automatic control of the ROV. The ROV is operated via the command console in the topside unit, which is also used to trigger the automatic control functionalities of the control system.

3.1.4 Stereolabs ZED Stereo Camera

The Stereolabs ZED, seen in Figure 3.4, is an out-of-the-box photogrammetry sensor, that comes with a wide variety of software applications and a extensive software development kit (SDK) (Stereolabs, 2021). For instance, the sensor comes with applications for dense depth estimation, dense spatial mapping, and object detection. For photogrammetry, the ZED has several advantages, amongst others; 1) the combined casing for the two monocular cameras, 2) fully digitized camera controls, and 3) streamlining of data acquisition. The combined casing keeps the relative translation and orientation between the two cameras practically constant and allows the shutters to be synchronized with high precision,

Figure 3.4: The Stereolabs ZED stereo camera. Courtesy: Stereolabs (2021)

minimizing the relative temporal delay of image pairs. The fully digitized camera controls yield a flexible way of adjusting the camera resolution, exposure and frame rate, without the need for physical intervention. Additionally, it allows the sensor to be embedded in networking middle-wares in order to achieve full remote control of it, a crucial feature in terms of data validation and - quality. The technical specification of the ZED stereo camera can be found in Table 3.1.

Parameter	Specification
Resolution and frequency	$2 \times 2208 \times 1242$ @ 15 fps
	$2 \times 1920 \times 1080$ @ 30 fps
	$2 \times 1280 \times 720$ @ 60 fps
	$2 \times 672 \times 376$ @ 100 fps
Output Format	YUV 4:2:2
Field of View	Horizontally: 90°
	Vertically: 60°
	Diagonally: 100°
Image Sensor	1/3" 4MP CMOS
Active Array Size	2688×1520 pixels per sensor (4MP)
Focal Length	2.8 mm - f/2.0
Shutter	Electronic synchronized rolling shutter
Baseline	120 mm
Depth Range	0.5 - 25 m
Dimensions	175 x 30 x 33 mm

Table 3.1: Technical Specification for the Stereolabs ZED stereo camera. Courtesy: Stereolabs (2018)

3.1.5 Camera Setup and Software Topology

Since commercial underwater housing are not readily available for the ZED stereo camera, a custom underwater housing was set up for this project. The housing consisted of a 4 inch cylindrical acrylic tube, with the stereo camera and a NVidia Jetson TX2 microcomputer mounted inside, seen in Figure 3.5. The ZED SDK and the Sennet software (Larsen, 2020b) was installed on the Jetson TX2. The ZED SDK allowed Stereolabs software

Figure 3.5: The cylindrical underwater housing containing the ZED stereo camera.

solutions, like GPU-accelerated data compression, and automatic exposure control, to be used during data acquisition. The Sennet framework provided a network module with TCP servers and clients, a OpenGL graphics renderer, and a ImGui graphical user interface. The Jetson TX2 hosted a server, which received camera commands and camera settings from a client, hosted on a topside laptop. The client received image pairs extracted from the stereo camera by the server. The transmitted image pairs were rendered to the display by the renderer running in conjunction with the client on the topside laptop. The setup, seen in Figure 3.6, allowed the stereo camera to be remotely operated, while getting real-time visualization of the acquired image pairs.

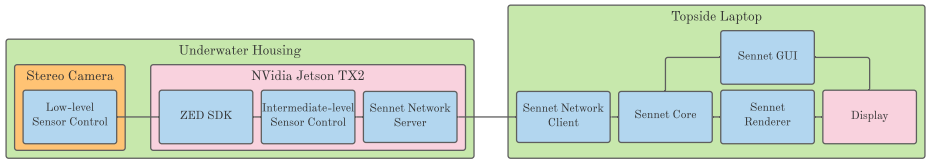


Figure 3.6: Camera setup and software topology for the ZED stereo camera.

3.2 Ekne Wreck Site Survey

The M/S Helma wreck site is located north west of Ekne in the Trondheim fjord, Norway, at depths ranging from 55 to 60 meters below sea surface. A map showing the location of the wreck site, and relevant bathymetric information can be seen in Figure 3.7. The wreck site was discovered by the Applied Underwater Robotics Laboratory (AURLab) research group at NTNU during a sea bottom survey in 2014, and was later studied with a multi-beam echo sounder, a side-scan sonar, and a stereo camera mounted on a ROV in April 2019 (King, 2020). According to historical data, the overall dimensions of the hull of M/S Helma were roughly $37.9 \times 8.4 \times 4.0$ meters.

The 22nd of January 2021, NTNU AURLab conducted a site survey of the M/S Helma wreck site with R/V Gunnerus and the SUB-Fighter 30K ROV. Due to technical difficulties, the ROV control system did not function properly, consequently, the auto functionalities of the ROV were not available throughout the survey. The survey consisted of two dives with the ROV, referred to as Dive 1 and Dive 2. Dive 1 and Dive 2 elapsed for

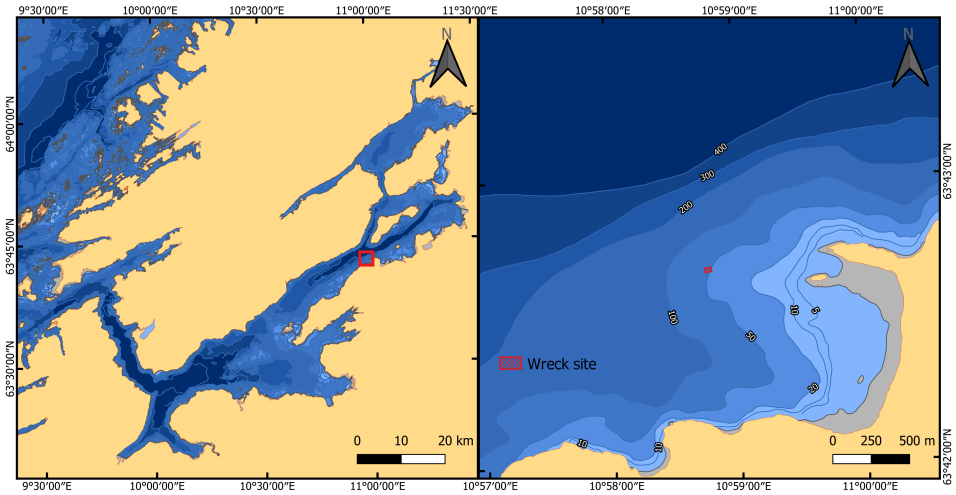


Figure 3.7: Survey map of the Ekne wreck site, created in QGIS. Map data courtesy: Kartverket (2020)

roughly 22 and 21 minutes, respectively. The survey focused on mapping the wreck site with optical sensors, mainly for the purpose of photogrammetry. The relevant payload sensors for the survey were the Stereolabs ZED stereo camera, an Ecotone scientific underwater hyperspectral imager, a omnidirectional camera, and a GoPro. For the majority of the survey, the stereo camera settings shown in Table 3.2 were used.

Setting	Value
Resolution	$2 \times 1920 \times 1080$ pixels
Target FPS	30
Self-calibration	Off
Brightness	4
Contrast	4
Hue	0
Saturation	4
Sharpness	3
Gamma	5
Exposure	Auto
Whitebalance	Auto

Table 3.2: Stereolabs ZED stereo camera settings for the Ekne wreck site survey.

For Dive 1, the omnidirectional camera was detached from the ROV as a calibration target, held by the manipulator arm, caused spatial constraints. Dive 1 consisted of general site orientation, followed by four transect lines across the midship of the wreck. The location of the transect lines were selected due to the relatively low vertical variation of the wreck, which allowed for navigation at altitudes ranging from one to two meters. This was critical

in order to minimize the effect of light beam attenuation and ensure a good signal to noise ratio for the optical sensors, especially the underwater hyperspectral imager. At the end of Dive 1, the calibration target was deployed in close proximity to the wreck site. The calibration target consisted of a planar checkerboard pattern, with 30×30 millimeters tiles, for geometrical calibration, as well as a sanded polystyrene plate, for spectral referencing. The ROV hovered over the calibration target, trying to capture it from various directions in the stereo camera images, and flew over it in order to capture it with the underwater hyperspectral imager.

For Dive 2, the omnidirectional camera was attached to the ROV, in addition to the other optical sensors utilized during Dive 1. The goal of Dive 2 was to cover as much of the wreck site as possible, especially the geometrically complex parts of the wreck, near the bow and stern. The larger vertical variations at these parts rendered navigation more difficult. Consequently, the altitude of the ROV changed more dramatically and the signal of the optical sensors were more prone to light beam attenuation, compared to Dive 1.

3.3 Camera Calibration Experiments

When performing geometrical calibration of a camera, capturing the calibration target at the edges of the image is important due to the stronger lens distortion in this region. For stereo image pairs, the calibration target has to be captured in both images in order to perform extrinsic calibration. Additionally, for underwater images, capturing the calibration target from various angles is important to reduce the correlation between different camera parameters, especially when the effects of refraction are not explicitly modelled (Shortis, 2015).

An initial attempt was made to calibrate the camera based on the image pairs of the calibration target deployed at the Ekne wreck site. However, a combination of the calibration target being placed on the flat seafloor, the 45 degree inclination angle of the stereo camera, and the roll and pitch stability of the ROV, made it hard to satisfy the aforementioned criteria. Additionally, the illumination from the HMI lamps, in combination with forward scattering in the water, caused a significant amount of glare and blur, reducing the dynamic range and contrast of image pairs of the calibration target.

In order to provide a better data foundation for calibrating the stereo camera, two auxiliary experiments were conducted in the main tank at Trondheim Biological Station (TBS), a NTNU facility for marine biology. One dataset was collect for each of the experiments, referred to as Calibration Dataset 1 and Calibration Dataset 2. During the experiments, the tank contained sea water from a depth of 100 meters in the Trondheim fjord, which was believed to be a suitable proxy for the waters at the Ekne wreck site with regards to the optical properties outlined in Section 2.2. For the experiments, the stereo camera was configured with the setup seen in Figure 3.6, with the underwater housing suspended underneath the tank footbridge. A geometrical calibration target, consisting of a checkerboard pattern with 40×40 mm tiles, was attached to a pole and submerged in front of the stereo camera, at a distance of roughly 2 meters. The calibration target was moved throughout the field of view, especially along the edges, while making sure it was visible

by both cameras. The camera settings utilized during the experiments were the same as the ones utilized during the Ekne survey, seen in Table 3.2.

3.4 Camera Calibration

In order to identify the parameters of the linear perspective SVP camera model, the non-linear corrections, and the relative orientation of the individual cameras of the ZED stereo camera, i.e. the parameters of Equation 2.22, Equation 2.24, Equation 2.26, and Equation 2.32, respectively, a camera calibration was performed. The calibration was based on the image pairs from the TBS calibration experiment. The camera calibration was conducted with Matlab's camera calibration toolbox, which is based on Zhang's method for intrinsic calibration and the 8-point algorithm for extrinsic calibration (Bouguet, 2015). For the camera calibration, the image pairs were selected such that the calibration target was exposed throughout a large portion of the FOV, and at various orientations, for the two cameras. This was done in order to more accurately capture the nonlinear distortion effects, introduced both by the housing interface and camera lenses, as well as add constraints to the MLE optimization procedure in Zhang's method (Zhang, 2000).

Due to the fixed baseline of the ZED stereo camera, the extrinsic parameters from an in-air calibration of the camera were used as validation references for the extrinsic parameters obtained from the in-water calibration. The calibration procedure was conducted by eliminating one image pair at a time, re-calibrating the cameras, and validating the obtained extrinsic parameters, until a low mean reprojection error and physically meaningful extrinsic parameters were obtained. Generally, image pairs with large reprojection errors, large differences in the mean reprojection error between the two cameras, and similar calibration target placements were eliminated. This calibration procedure was conducted for four different models, i.e. the perspective SVP camera model with two- and three coefficient radial distortion, with and without two coefficient tangential distortion.

3.5 Navigation Data Processing

To establish a comparative basis for the pose estimates produced by OpenVSLAM, the ROV navigation data from the Ekne survey was processed by means of outlier rejection and noise filtering. Due to technical issues with the ROV control system, the navigation data was not properly logged internally, and had to be extracted from the EIVA Navipack GND log files. For this reason, only a subset of the navigation data was available for post-processing. Specifically, the available navigation data were ROV position measurements from the R/V Gunnerus HiPAP APS, gyroscope roll, pitch, and yaw measurements, pressure sensor depth measurements, and DVL altitude measurements from the ROV's internal control system.

The navigation data was first inspected in order to identify the need for outlier rejection. Based on this inspection, it was established that the APS signals were the only ones that contained samples which could be considered outliers, with sporadic single position measurements one to ten meters away from the measurement tendency. In order to eliminate

the outliers, two rolling window threshold (RWT) filters were applied to the APS signals, one for the planar components and one for the depth component. In mathematical terms, the filters are defined as

$$\mu_N - t_P \cdot \sigma_N < N < \mu_N + t_P \cdot \sigma_N, \tag{3.1a}$$

$$\mu_E - t_P \cdot \sigma_E < E < \mu_E + t_P \cdot \sigma_E, \tag{3.1b}$$

$$\mu_D - t_D \cdot \sigma_D < D < \mu_D + t_D \cdot \sigma_D, \tag{3.1c}$$

where μ_N, μ_E, μ_D, and σ_N, σ_E, and σ_D are the mean and standard deviation of the APS signals within the rolling window, and t_P, and t_D are the planar and depth thresholds. Samples outside the intervals are rejected as outliers and replaced by the window mean values. The RWT filters were tuned through trial and error, until the measurement samples considered to be outliers during the initial inspection were rejected with a minimal amount of false positives.

In order to filter out high-frequency components from the navigation data, considered to be measurement noise, Hamming-windowed finite impulse response (FIR) low-pass filters were applied to the signals (Oppenheim, 1983, p. 256). The Hamming-windowed FIR low-pass filter is defined as

$$\hat{x}_n = \sum_{i=0}^{N} \left(0.54 - 0.46\cos\left(\frac{2\pi i}{N}\right)\right)\left(2f_c\mathrm{sinc}\big(2f_c i\big)\right)x_{n-i}, \tag{3.2}$$

where f_c is the filter cutoff frequency, and N is the filter order. To compensate for the time delay introduced by the FIR filter, the timestamps of the filtered signal were shifted by the analytical expression for the FIR filter time delay as

$$t_n = t_{n,\text{FIR}} - \Delta t_{\text{FIR}}, \tag{3.3}$$

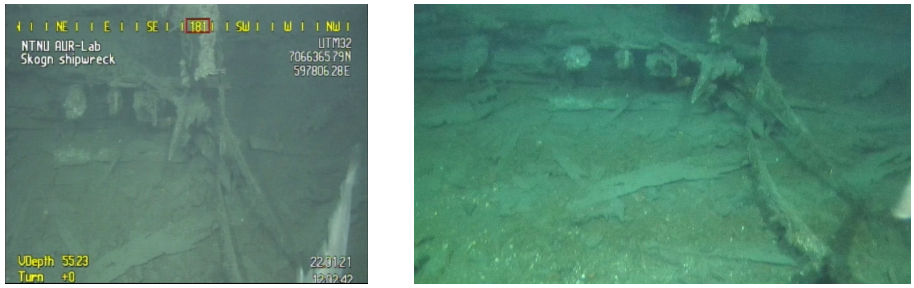
where the time delay Δt_{FIR} is given as

$$\Delta t_{\text{FIR}} = \frac{1}{2}\frac{N-1}{f_s}. \tag{3.4}$$

One FIR filter was tuned for each of the navigation data sensor systems through trial and error. The filters were tuned until the filtered signals became satisfyingly smooth, without loosing the dominant tendencies in the original signal. To avoid aliasing in the filtered signal, the cutoff frequency was kept strictly below the Nyquist frequency throughout the entire tuning process, i.e.

$$f_c < f_{NQ} = \frac{1}{2}f_s. \tag{3.5}$$

3.6 Data Synchronization



(a) **(b)**

Figure 3.8: Images of a synchronization event, a) the ROV driver camera image, b) the left image from the stereo camera.

Since the Jetson TX2 was not connected to the ROV NTP server during the Ekne survey, the ROV navigation data and the ZED stereo camera footage was not synchronized. To perform a crude synchronization, the footage from the ROV operator camera was inspected alongside the ZED stereo camera footage in order the find a correspondence between the clock of the NTP server and the TX2. 20 easily recognizable events, such as the one seen in Figure 3.8, were identified in the footage from the two cameras, 7 events from Dive 1 and 13 events from Dive 2. For each of these events, the timestamp of the ZED image pairs were extracted from the ZED SDK, while the NTP server timestamps were extracted from overlay of the ROV camera footage, under the assumption that the time delay of the overlay was negligible. The precision of the timestamps of the ROV overlay was only set to seconds. In order to get sub-second precision, the ROV camera footage timestamps were linearly interpolated between frames. Initially, the timestamps were synchronized by adding a constant bias to the TX2 timestamps, under the assumption the drift in the two CPU clocks would be negligible throughout the two dives. In later analyses, when comparing the V-SLAM trajectories to the navigation data, this methodology was found too inaccurate. Consequently, in addition to the mean bias, a bias correction was added on a per trajectory basis, as a tuning parameter. Following this methodology, the synchronization process can be expressed as

$$t^{\text{TX2}}_{k,d} \leftarrow t^{\text{TX2}}_{k,d} + \bar{t}_d + \Delta\bar{t}_s, \qquad k \in K, \quad s \in S, \quad d \in \{1, 2\}, \tag{3.6a}$$

$$\bar{t}_{b,d} = \frac{1}{N} \sum_{i=1}^{N} \left(t^{\text{ROV}}_{i,d} - t^{\text{TX2}}_{i,d} \right), \qquad i \in I, \quad d \in \{1, 2\}, \tag{3.6b}$$

where $\bar{t}_{b,d}$ is the mean constant bias for dive d, $\Delta\bar{t}_s$ is the mean bias correction for trajectory s, k is the time index for each dive, and i is a synchronization point.

3.7 Image Processing

Through visual inspection, the stereo image pairs from the Ekne wreck site survey seemingly contained a significant amount of noise. The noise was identified by sporadic white and black pixels appearing throughout the images. In addition to noise, the image pairs did, in general, have relatively low contrast, and contained a significant amount of blur along object edges, as well as a small amount of backscatter. The blur was believed to be an effect of small angle forward scattering in the water and housing interface. To improve the quality of the image pairs from the stereo camera, with respect to the robustness and accuracy of OpenVSLAM, several image processing methods were implemented in the OpenVSLAM tracking module. The implemented methods can be categorized into three categories; 1) sharpness enhancement and denoising, 2) contrast enhancement, and 3) color correction and backscatter removal.

3.7.1 Image Sharpness Enhancement and Denoising

In order to filter out the noise in the image pairs, a bilateral filter (BLF) was applied to the image pairs. The BLF is a non-linear filter which performs edge-preserving noise removal by applying a bilteral Gaussian kernel, which accounts for spatial- and radiometric (intensity) differences of pixels (Tomasi and Manduchi, 1998). The BLF has three tuning parameters; the filter diameter d_{BLF}, the spatial variance σ_{s}^2, and the radiometric variance σ_{r}^2. Larger spatial variance has the effect of smoothing larger image features, while larger radiometric variance has the effect of smoothing edges. The filter parameters were tuned by visual inspection of the images, as the resulting feature extraction and -matching in OpenVSLAM.

3.7.2 Contrast Enhancement

Since the stereo image pairs suffered from low contrast, histogram equalization (HE), a common contrast enhancement technique, was employed previous to performing BLF (Hummel, 1977). By applying HE, it was believed that the number and the quality of the visual features, extracted by OpenVSLAM, would increase. HE performs contrast enhancement by transforming the pixel intensities, making the cumulative distribution function transformed intensities into a linear function. Since this transform is given for a given image, HE is a parameter free method and therefore did not require any tuning.

Another contrast enhancement method, contrast-limited adaptive histogram equalization (CLAHE), was also implemented into OpenVSLAM (Zuiderveld, 1994). As opposed to HE, which performs one histogram equalization on the entire image, CLAHE performs several histogram equalizations, each on small image patches throughout the image. Since local histogram equalization is prone to enhancing noise, CLAHE limits the contrast on each image patch by adding a clipping limit. The clipping limit c is therefore a parameter that needs tuning, in addition to the image patch width and height, w and h, respectively. The CLAHE parameters were tuned in a similar fashion to the ones of the BLF parameters, by visually inspecting the processed image and looking at the effect on the visual features extracted by OpenVSLAM. Similarly, to HE, CLAHE was applied previous to BLF, in

order to reduce noise and enhance sharpness of edges.

3.7.3 Color Correction and Backscatter Estimation

The final method that was implemented into OpenVSLAM is a SOTA deep learning-based method for underwater image enhancement (Chen et al., 2021). The method is referred to as underwater image enhancement network (UIENet) within the scope of this project. UIENet is a CNN based on the revised underwater image formation model, outlined in Section 2.2.3 and Section 2.2.4. The model is comprised of two modules, a backscatter estimation module and a direct transmission estimation module, which estimate the backscatter and direct transmission components of a simplified version of Equation 2.14. The original model is implemented in the Python-based DL-framework PyTorch (Paszke et al., 2019), but was converted to C++ by porting it as a TorchScript. The model was then implemented in the OpenVSLAM tracking module with GPU-acceleration by adding Torch and CUDA support to OpenVSLAM.

UIENet works exclusively on RGB images, while OpenVSLAM works exclusively on grayscale images. As such, UIENet was implemented in OpenVSLAM by extracting RGB stereo image pairs from the ZED SDK, inputting them into UIENet, and then converting the processed image pairs to grayscale. Contrary to HE and CLAHE, BLF was not applied to the image pairs processed by UIENet, in order to evaluate the capabilities of an entirely physical-based underwater image enhancement model with respect to V-SLAM.

3.8 Ground Truth and Georeferencing

3.8.1 Ground Truth Reference

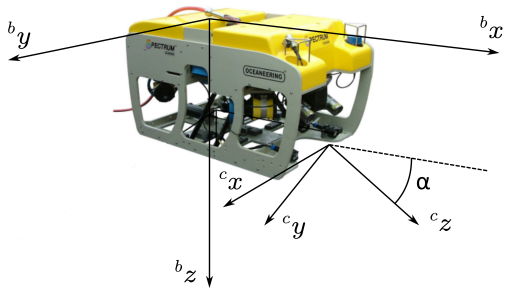


Figure 3.9: Relationship between body- and camera coordinate system. Adopted from: Dukan (2014)

In order to compare the drift and overall accuracy of different VSLAM trajectories, a ground truth reference had to be established. Due to potentially crude errors introduced

by inaccurate synchronization, the ground truth reference was based exclusively on the ROV navigation data, and physical measurements of the stereo camera mounting position relative to the APS transponder. Expressed in the ROV body CS, with the origin at the APS transponder position and the axis defined as in Figure 3.9, the relative translation from the APS transponder to the stereo camera was found to be equal to the values in Table 3.3.

Parameter	Value	Unit
${}^{b}t_x$	2.00	m
${}^{b}t_y$	-0.21	m
${}^{b}t_z$	1.40	m
α	45.00	deg

Table 3.3: Stereo camera lever arm and inclination angle.

The ground truth reference was created by utilizing the filtered APS position measurements and the filtered gyroscope roll, pitch, and yaw measurements to estimate the camera position and attitude in the world CS. To make the comparison between the ground truth reference and the OpenVSLAM trajectories easier, the attitude of the camera was represented by quaternions. Note that in this project, the Hamilton convention for quaternions is utilized, as opposed to the JPL convention, commonly utilized in visual-inertial SLAM (VI-SLAM) (Solà, 2017; Trawny and Roumeliotis, 2005). For readers unfamiliar with quaternions, a brief overview of quaternion arithmetic and attitude representation is given in Appendix B. From the filtered APS position measurements and gyroscope roll, pitch, and yaw measurements the position of the camera in the world CS was computed as

$$\begin{bmatrix} 0 \\ {}^{w}\mathbf{p}_l^{\text{GT}} \end{bmatrix} = {}^{w}_{b}\mathbf{q}_l \otimes \begin{bmatrix} 0 \\ {}^{b}\mathbf{p}^{\text{Cam}} \end{bmatrix} \otimes {}^{w}_{b}\mathbf{q}^{*}{}_l + \begin{bmatrix} 0 \\ {}^{w}\mathbf{p}_l^{\text{Trans}} \end{bmatrix}, \tag{3.7}$$

where ${}^{w}\mathbf{p}^{\text{Trans}}$ is the position of the transponder given by the filtered APS measurements, ${}^{b}\mathbf{p}^{\text{Cam}}$ is the lever arm of the stereo camera in the body CS, and ${}^{w}_{b}\mathbf{q}$ is the quaternion representing the rotation from the body CS to the world CS. The quaternion ${}^{w}_{b}\mathbf{q}$ was computed as a quaternion product of the unit quaternions representing the roll, pitch, and yaw rotations as

$${}^{w}_{b}\mathbf{q}_l = \mathbf{q}_l^{\text{Yaw}} \otimes \mathbf{q}_l^{\text{Pitch}} \otimes \mathbf{q}_l^{\text{Roll}}, \tag{3.8}$$

where $\mathbf{q}_l^{\text{Yaw}}$, $\mathbf{q}_l^{\text{Pitch}}$, and $\mathbf{q}_l^{\text{Roll}}$ were computed from the filtered gyroscope measurements matched with the corresponding filtered APS measurement. The ground truth camera attitude was computed as

$${}^{w}_{c}\mathbf{q}_l^{\text{GT}} = {}^{w}_{b}\mathbf{q}_l \otimes {}^{b}_{c}\mathbf{q}, \tag{3.9}$$

where ${}^{b}_{c}\mathbf{q}$ is the rotation aligning the camera CS with the body CS, defined in terms of the constitute rotations $\mathbf{q}_x$, $\mathbf{q}_y$, and $\mathbf{q}_z$ as

$$ {}^{b}_{c}\mathbf{q} = \mathbf{q}_x \otimes \mathbf{q}_y \otimes \mathbf{q}_z. \tag{3.10} $$

The constitute rotations were calculated in terms of the camera inclination angle α as

$$ \mathbf{q}_x = \text{Axis-Angle}(\begin{bmatrix}1 & 0 & 0\end{bmatrix}^\top, 0°) \tag{3.11a} $$

$$ \mathbf{q}_y = \text{Axis-Angle}(\begin{bmatrix}0 & 1 & 0\end{bmatrix}^\top, 90° - \alpha) \tag{3.11b} $$

$$ \mathbf{q}_z = \text{Axis-Angle}(\begin{bmatrix}0 & 0 & 1\end{bmatrix}^\top, 90°), \tag{3.11c} $$

where the Axis-Angle function is given in Equation B.18. The composition of Equation 3.11 is made under the assumption that the x-axis of the camera CS is parallel with the y-axis of the body CS.

3.8.2 Timestamp Matching

Since the output trajectories of OpenVSLAM have higher frequency than the APS measurements, and consequently the ground truth trajectories, a temporal matching problem has to be solved to find the optimal correspondence between samples. The matching problem can be expressed in terms of a constrained optimization problem as

$$ \underset{\mathcal{T}}{\text{minimize}} \sum_{(l,k)\in\mathcal{T}} \left\| t_l^{\text{GT}} - t_k^{\text{SLAM}} \right\|, \tag{3.12a} $$

$$ \text{subject to} \left\| t_l^{\text{GT}} - t_k^{\text{SLAM}} \right\| \leq \Delta t_{\text{Matching}}, \tag{3.12b} $$

where t_l^{GT} are the timestamps of the ground truth trajectories, t_k^{SLAM} are the timestamps of the OpenVSLAM trajectories, $\Delta t_{\text{Matching}}$ is a temporal matching threshold, and $\mathcal{T}$ is the ordered set of index pairs minimizing the absolute difference between the timestamps of the two trajectories. The matching problem in Equation 3.12 was solved by dynamic programming with the open-source Python library **ssdts_matching** (Palachy, 2021).

3.8.3 Optimization-Based Georeferencing

In order to compare the OpenVSLAM trajectories with the ground truth trajectories, a procedure to find the transformation from the object CS to the world CS, also known as georeferencing, had to be established. For this purpose, Umeyama's method, an algorithm commonly utilized by the VSLAM community for trajectory alignment, was used (Umeyama, 1991; Sumikura et al., 2019). Umeyama's method minimizes the squared Euclidean distance between two position trajectories under the assumption of homogeneous uncertainty. For georeferencing, the optimization problem can be expressed as

$$ \underset{{}^{w}_{o}\mathbf{q},\, {}^{w}_{o}\mathbf{t}}{\text{minimize}} \sum_{(l,k)\in\mathcal{T}} \left\| {}^{w}\mathbf{p}_l^{\text{GT}} - {}^{w}\mathbf{p}_k^{\text{SLAM}} \right\|^2, \tag{3.13a} $$

$$\begin{bmatrix} 0 \\ {}_{w}\mathbf{p}_{k}^{\text{SLAM}} \end{bmatrix} = {}_{o}^{w}\mathbf{q} \otimes \begin{bmatrix} 0 \\ {}_{o}\mathbf{p}_{k}^{\text{SLAM}} \end{bmatrix} \otimes {}_{o}^{w}\mathbf{q}^{*} + \begin{bmatrix} 0 \\ {}_{o}^{w}\mathbf{t} \end{bmatrix}, \tag{3.13b}$$

where $\mathcal{T}$ are the matched timestamp pairs between the ground truth trajectory and the OpenVSLAM trajectory, ${}^{o}\mathbf{p}_{k}^{\text{SLAM}}$ are the camera positions output by OpenVSLAM, and ${}^{w}\mathbf{p}_{l}^{\text{GT}}$ are the ground truth camera positions computed using the methodology in Section 3.8.1. The rotation obtained from solving Equation 3.13 were then used to compute the attitude of the camera in the world CS as

$${}_{c}^{w}\mathbf{q}_{k}^{\text{SLAM}} = {}_{o}^{w}\mathbf{q} \otimes {}_{c}^{o}\mathbf{q}_{k}^{\text{SLAM}}, \tag{3.14}$$

where ${}_{c}^{o}\mathbf{q}_{k}^{\text{SLAM}}$ are the camera attitudes output by OpenVSLAM. Similarly to the camera positions, the OpenVSLAM landmarks ${}^{o}\mathbf{m}^{j}$, were georeferenced as

$$\begin{bmatrix} 0 \\ {}_{w}\mathbf{m}^{j} \end{bmatrix} = {}_{o}^{w}\mathbf{q} \otimes \begin{bmatrix} 0 \\ {}_{o}\mathbf{m}^{j} \end{bmatrix} \otimes {}_{o}^{w}\mathbf{q}^{*} + \begin{bmatrix} 0 \\ {}_{o}^{w}\mathbf{t} \end{bmatrix}, \qquad \mathbf{m}^{j} \in \mathbf{m}. \tag{3.15}$$

3.9 V-SLAM Error Metrics

In order to analyze the accuracy and drift of OpenVSLAM, several error metrics had to be employed. For this purpose, the homogeneous transform $\mathbf{T}$, composed of the ground truth- and OpenVSLAM positions and attitudes, were utilized to simplify the analysis. The homogeneous transform $\mathbf{T}$, composed of the position $\mathbf{p}_k$ and quaternion attitude representation $\mathbf{q}_k$, is given as

$$\mathbf{T}_k = \mathbf{T}(\mathbf{p}_k, \mathbf{q}_k) = \begin{bmatrix} \mathbf{R}(\mathbf{q}_k) & \mathbf{p}_k \\ \mathbf{0}_{1x3} & 1 \end{bmatrix}, \tag{3.16}$$

where $\mathbf{R}(\mathbf{q}_k)$ is the rotation matrix representation of $\mathbf{q}_k$. For simplification of notation, the two utilized homogeneous transforms are expressed as

$$\mathbf{T}_{l}^{\text{GT}} = \mathbf{T}({}^{w}\mathbf{p}_{l}^{\text{GT}}, {}_{c}^{w}\mathbf{q}_{l}^{\text{GT}}), \tag{3.17a}$$

$$\mathbf{T}_{k}^{\text{SLAM}} = \mathbf{T}({}^{w}\mathbf{p}_{k}^{\text{SLAM}}, {}_{c}^{w}\mathbf{q}_{k}^{\text{SLAM}}), \tag{3.17b}$$

where ${}^{w}\mathbf{p}_{l}^{\text{GT}}$ and ${}_{c}^{w}\mathbf{q}_{l}^{\text{GT}}$, and ${}^{w}\mathbf{p}_{k}^{\text{SLAM}}$ and ${}_{c}^{w}\mathbf{q}_{k}^{\text{SLAM}}$ are computed using the methodology in Section 3.8.1 and Section 3.8.3, respectively.

3.9.1 Absolute Trajectory Error

In order to quantify the goodness of fit of the OpenVSLAM trajectories with respect to the ground truth trajectories, the absolute trajectory error (ATE) was utilized (Sturm et al., 2012). In terms of the homogeneous transforms in Equation 3.17, the ATE was calculated as

$$\text{ATE}_k = (\mathbf{T}_l^{\text{GT}})^{-1}\mathbf{T}_k^{\text{SLAM}}, \qquad (l,k) \in \mathcal{T}. \tag{3.18}$$

To evaluate the translational- and rotational components of the ATE individually, the root mean squared error (RMSE) of the components were calculated as

$$\text{ATE}_{\text{Trans.},k} = \text{RMSE}\big[\text{Translation}(\text{ATE}_k)\big], \tag{3.19a}$$

$$\text{ATE}_{\text{Rot.},k} = \text{RMSE}\big[\text{Rotation}(\text{ATE}_k)\big], \tag{3.19b}$$

where $\text{Translation}(\text{ATE}_k)$, and $\text{Rotation}(\text{ATE}_k)$ are the translational- and rotational elements of the ATE matrix, respectively.

3.9.2 Relative Pose Error

Since dead reckoning navigation systems, like V-SLAM, inevitably accumulate drift over time, it was desirable to evaluate the drift over the OpenVSLAM trajectories. To quantify the drift, the relative pose error (RPE) metric was utilized (Sturm et al., 2012). The RPE is defined as

$$\text{RPE}_k(\Delta) = \big[(\mathbf{T}_l^{\text{GT}})^{-1}\mathbf{T}_l^{\text{GT}}(\Delta)\big]^{-1}\big[(\mathbf{T}_k^{\text{SLAM}})^{-1}\mathbf{T}_k^{\text{SLAM}}(\Delta)\big], \qquad (l,k) \in \mathcal{T}, \tag{3.20}$$

where Δ is a displacement, which can either be a temporal displacement, i.e. a time shift, or a spatial displacement. In the conducted analysis, a spatial displacement was utilized to calculate the RPE. Similarly to the ATE, the translational- and rotational components of the RPE were calculated by taking the RMSE of the relevant elements of the RPE matrix

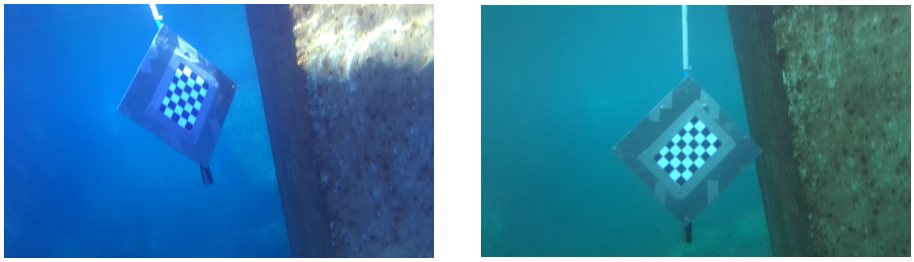
$$\text{RPE}_{\text{Trans.},k} = \text{RMSE}\big[\text{Translation}(\text{RPE}_k)\big], \tag{3.21a}$$

$$\text{RPE}_{\text{Rot.},k} = \text{RMSE}\big[\text{Rotation}(\text{RPE}_k)\big]. \tag{3.21b}$$

Chapter 4

Results and Discussion

4.1 Camera Calibration



(a) **(b)**

Figure 4.1: Example images from a) Calibration Dataset 1, and b) Calibration Dataset 2.

The parameters for the camera calibration, based on the camera calibration methodology outlined in Section 3.4, can be seen in Table 4.1 and Table 4.2. For clarification, b_x, b_y, and b_z are the x-, y-, and z-component of the stereo baseline vector ${}^{2}_{1}\mathbf{b}$, while r_x, r_y, and r_z are the XYZ Euler angles of the stereo camera rotation matrix ${}^{2}_{1}\mathbf{R}$, outlined in Section 2.4.1. The image pairs utilized in the calibration were exclusively taken from the Calibration Dataset 1 from the calibration experiment. The reason for doing this was a slight difference in the ambient illumination in Calibration Dataset 1 and Calibration Dataset 2, as seen in Figure 4.1, which had a significant effect on the calibration results. From Figure 4.2 one can see that the mean reprojection error per image pair was relatively balanced between the left and right camera, indicating that the calibration target was well exposed for both cameras and that the angle between the cameras and the calibration target was not too large.

From Table 4.1 one can see that the camera model did not include the third radial distor-

Parameter	Left, In-Air	Left, In-Water	Right, In-Air	Right, In-Water
f_x	1400.2200	1793.0321	1398.4100	1789.2000
f_y	1400.2200	1694.3191	1398.4100	1690.4034
c_x	960.3700	936.4110	924.1700	915.3692
c_y	546.3410	555.9133	523.8170	534.5125
s	0	0	0	0
k_1	-0.1725	-0.0470	-0.1712	-0.0606
k_2	0.0265	0.1164	0.0023	0.2516
k_3	0	0	0	0
p_1	0.0023	0	0.0023	0
p_2	0.0004	0	0.0004	0

Table 4.1: Intrinsic parameters of the perspective SVP model.

Parameter	In-Air	In-Water	Unit
b_x	120.0010	119.3609	mm
b_y	0.0012	0.3348	mm
b_z	0.0113	-0.0445	mm
r_x	-0.0039	0.0027	rad
r_y	0.0037	0.0076	rad
r_z	0.0003	0.0004	rad

Table 4.2: Extrinsic parameters of the stereo normal model.

tion coefficient, k_3, and omitted the tangential distortion, i.e. p_1 and p_2, completely. The third radial distortion coefficient is, practically, only needed for fisheye- and equirectangular cameras, which exert extreme distortion (Shortis, 2015). For this reason, the fact that it had a negligible impact on the reprojection error, and strongly affected the other radial distortion coefficients, it was left out of the model. The radial distortion coefficients are in fact always correlated, but the correlation is strengthened further when the model is calibrated against a planar calibration target. Additionally, tangential distortion was omitted from the model due to instabilities in the calibration procedure. Inclusion of tangential distortion in the model lead to large variations in the relative translation between the cameras, especially in the z-component of the baseline vector, b_z. A reason for this correlation can be the relatively small variation of the calibration targets orientation around the z-axis of the two camera, as seen from the reprojections in Figure 4.3 and Figure 4.4. Since Zhang's method exploits the orthonormal properties of rotation matrices, additional images, in which the calibration target lie in the same plane as in a previous image, do not add additional constraints to the optimization problem (Zhang, 2000). The lack of constraints is believed to be the reason behind the observed correlation between the tangential distortion and the baseline vector. By omitting the tangential distortion, the optimization algorithm became more stable and a better correspondence between the extrinsic parameters obtained in water and in air was achieved. As seen in Table 4.2, the differences in the translations and rotations of the calibrated extrinsic parameters are well below 1 mm and $0.5°$, when compared to their corresponding values from the Stereolabs factory calibration.

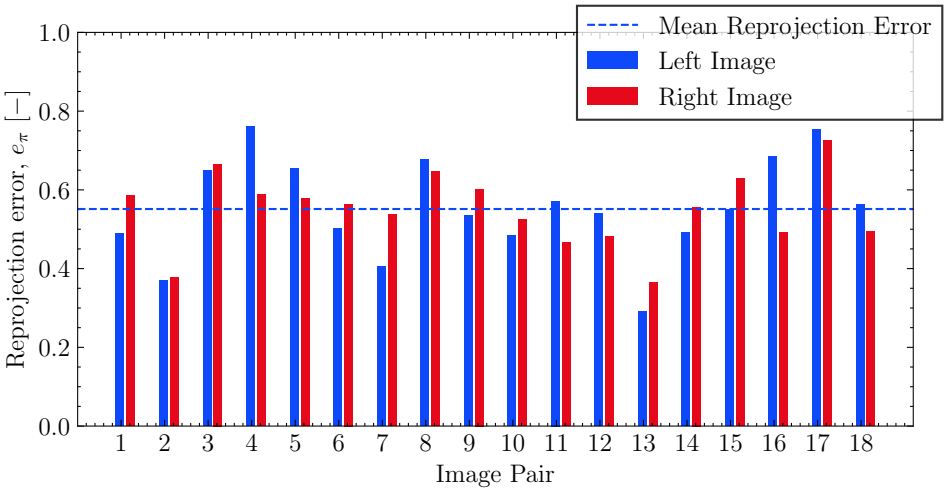


Figure 4.2: Mean reprojection errors for the calibration image pairs.

From the calibrated intrinsic parameters in Table 4.1, one can see the effects of refraction on the parameters of the perspective SVP model. Among other, one can see that the focal lengths become larger as a consequence of the perceptually smaller field of view due to refraction. For instance, planar ports have been shown to increase the focal length by up to 25%, while dome ports only increase it by 6-7%, depending on the relative translation between the focal points of the camera and port (Bruno et al., 2011). For the left camera, one can see a change in the focal length of 28% and 21%, along the cameras x- and y-axis, f_x and f_y respectively. Considering the cylindrical housing utilized in this project, these values give physical sense, as the cylinder housing has refractive properties similar to a planar port along the longitudinal axis, and similar to a dome port along the polar axis. The fact that the camera was mounted approximately 4 mm from the housing wall, which has an inner radius of 51 mm, indicates why the effects of refraction are relatively strong in the vertical direction as well. One can also see the effects of refraction on the x- and y-coordinate of the principal point, c_x and c_y respectively. The principal points for the two cameras were shifted upwards and to the left, which indicate that optical axis of the cameras were slightly misaligned with respect to the housing surface normal. Since the effects of refraction are radially symmetric only when the optical axis of the camera and the surface normal of the housing are aligned, this indicates that tangential distortion should also have been included in the camera model, which was not feasible due to the aforementioned optimization problems (Li et al., 1997).

In order to evaluate the camera model with respect to the mathematical models outlined in Section 2.3, the reprojection errors from the calibration procedure were analyzed. From Equation 2.16, one can see that refraction is dependent on the incident angle on the refractive surface. This fact would lead one to believe that the reprojection errors from the calibration would be largest in the outer edges of the images, where the incident angles and consequently refractive effects are stronger. However, from Figure 4.3 and Figure 4.4,

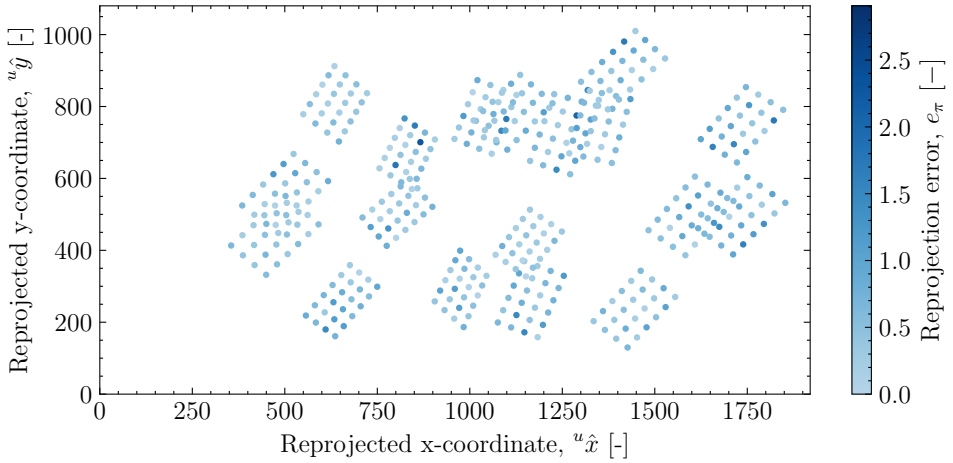


Figure 4.3: Calibration target reprojections for the left camera.

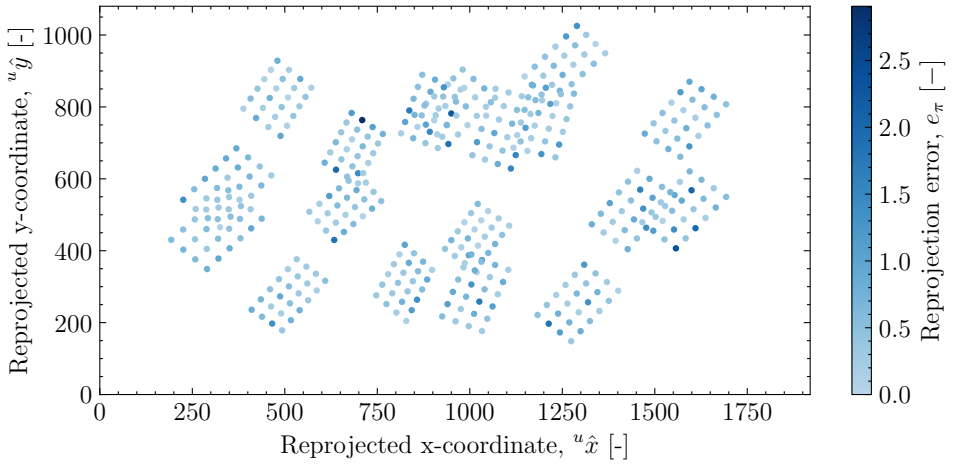


Figure 4.4: Calibration target reprojections for the right camera.

one can not see any pattern of this effect. The large reprojection errors are, seemingly, randomly distributed throughout the FOV. This was an indication that the large reprojection errors are caused by other reasons than refraction. As discussed earlier, illumination was a significant factor which affected the calibration results. It is therefore likely that, uneven illumination of the calibration target points were one of the reasons for the large reprojection errors, as the utilized dataset contained images with a significant amount of glare from reflections and wave flickering.

Based on the general camera model in Section 2.3.1, the reprojection errors should exclusively consists of zero-mean Gaussian noise. From Figure 4.5a and Figure 4.5b, one can

see that the reprojection errors did, in general, seem to fit this model, both for the left and right camera. The mean errors for both of the cameras were practically zero, and the 3σ sample covariance ellipsis, which theoretically should encompass 99.7% of the samples, did contain most of them. However, one remark that is worth noting is the slightly higher number of samples outside the 3σ-bound of the right camera than the left camera. A reason for this could, for instance, be slight differences in reflections from the calibration target for the two cameras. To get a more objective and generic evaluation of the error introduced by not employing a refractive camera model, more advanced analyses, for instance caustic analysis or 3D reconstruction of a known target at different distances, should be conducted in order to expose the distance and angle dependency of refraction (Jordt-Sedlazeck and Koch, 2013; Shortis, 2015).

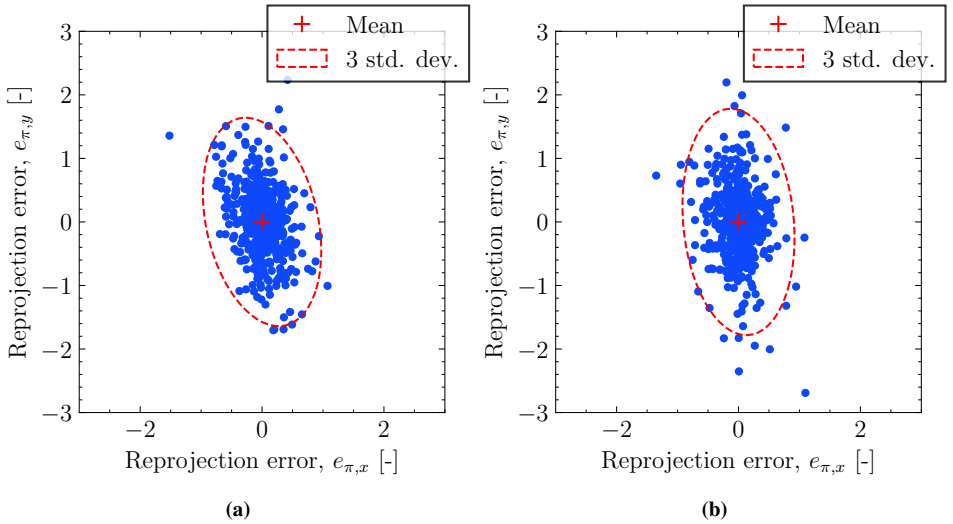


Figure 4.5: Reprojection error distributions for a) the left camera, b) the right camera.

4.2 Navigation Data Processing

Signal	Window	Threshold
-	-	-
APS, northing	10	2.4
APS, easting	10	2.4
APS, depth	10	3.0

Table 4.3: Rolling window threshold filter parameters.

The RWT filter parameters, tuned with the methodology described in Section 3.5, can be seen in Table 4.3. From the parameters one can see that the window sizes were set relatively small. This is due to the fact that the RWT does not account for tendentious changes

in the signal, which is the case when the ROV changes position. By having a too large window, the variations within the window have a higher chance of simply being changes in the ROV position, for instance seen in the depth measurements in Figure 4.6 around the 4300 second marker, when the ROV started to ascend to the surface. Additionally, from the parameters one can see that the threshold for the two planar measurements is significantly lower than the one for the depth measurement. This means that fluctuations in the depth measurements have to be more extreme, compared to the other measurements in the window, to be considered outliers. This was primarily set due to the small variations in the depth measurements, when compared to the planar measurements, as seen in Figure 4.6. The majority of the bad measurements, which are considered outliers, came from periods where the ROV was close to the surface. Physically, this makes sense as the APS measurements are more susceptible to shadow regions, and loss of line of sight closer to the surface. The majority of the false positives of the RWT filter also seem to originate from the filter not considering tendentious changes, as already pointed out. Despite the short-comings of the RWT filter, the outlier-rejected APS measurements were considered sufficiently clean for further frequency-filtering.

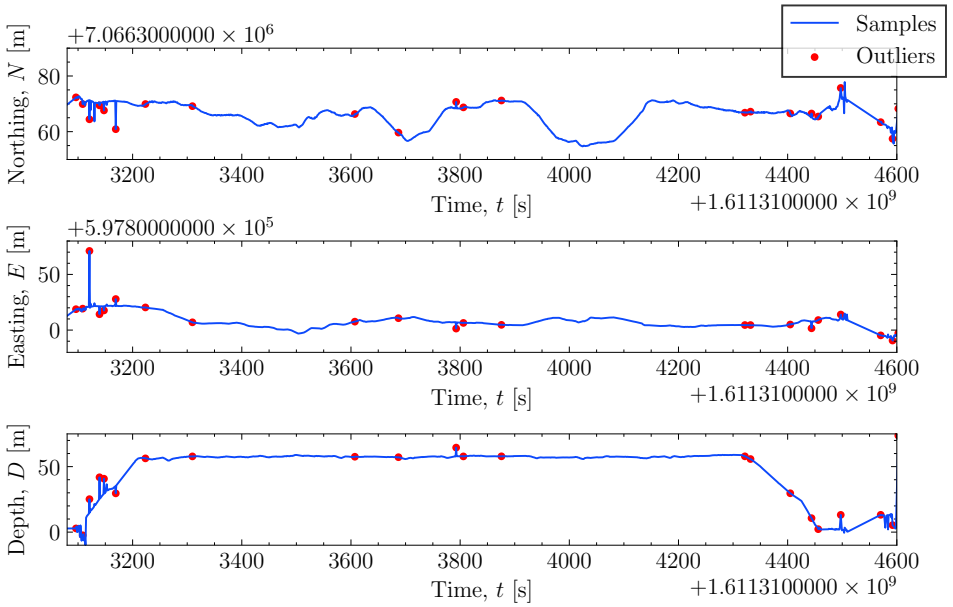


Figure 4.6: Detected outliers in the APS measurements from the RWT filter.

By using the tuning method described in Section 3.5, the FIR filter parameters seen in Table 4.4 were found to yield a satisfying filtering effect of the relevant sensor measurements. During the FIR filtering process, the sampling frequencies of the gyroscope were found to be different from the sampling frequency given in technical specifications, seen in Table A.2. This was an indication that the measurements were not raw sensor measurements, but rather measurements that had already been processed. This hypothesis was further backed by the fact that the FIR filter had little effect on the gyroscope measure-

ments. This can be seen from the seemingly identical unfiltered- and filtered gyroscope in Figure 4.7.

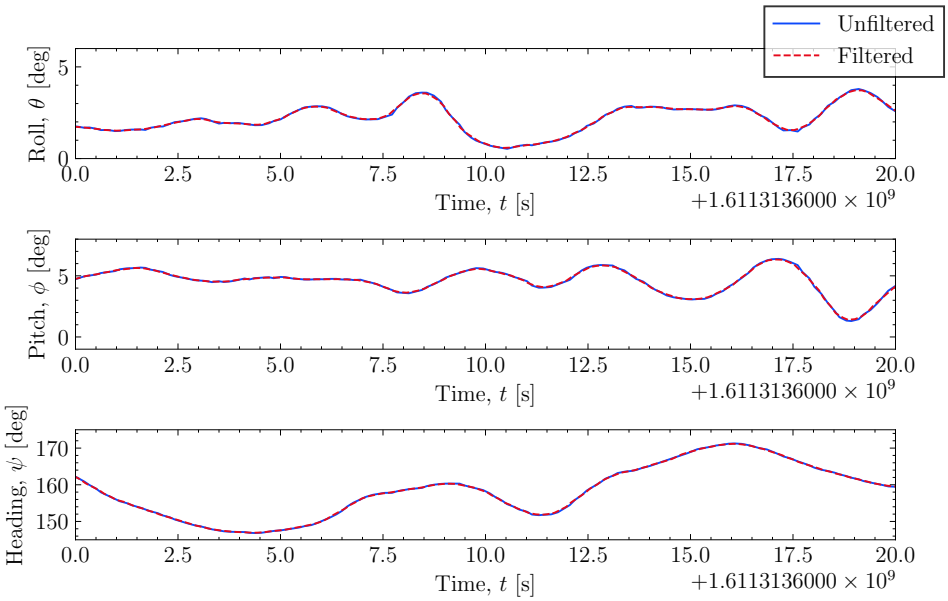


Figure 4.7: FIR filtered gyroscope roll, pitch, and heading measurements.

Unlike the gyroscope measurements, the APS position measurements were found to be particularly noisy, with sporadic fluctuations as large as 0.5 meters, as seen in Figure 4.8. The APS measurements therefore needed extensive frequency filtering in order to eliminate the high-frequency signal components. In order to get a satisfyingly smooth signal, the FIR filter order had to be set quite high, while the FIR filter cutoff frequency had to be set quite low relatively to the sampling frequency of the APS measurements, as seen in Table 4.4.

Similarly to the gyroscope, the FIR filter had little effect on the DVL measurements, which is evident from Figure 4.9. However, in contrast to the gyroscope measurements, the sampling frequency of DVL measurements coincided with the sampling frequency given in the technical specifications, seen in Table A.3. Compared to the gyroscope, the DVL is a highly advanced sensor and it is therefore likely that the sensor has some internal signal processing modules. It was later confirmed by AURLab employees that the DVL was processed internally in the ROV control system before being logged in EIVA Navipac, which explains why the frequency filtering was redundant. For further information about the DVL processing, the reader is referred to Dukan (2014). Additionally, information was provided that the gyrocompass, i.e. the one providing the heading measurements from the gyroscope, was not properly calibrated. The inaccurate heading measurements are therefore a source of error in further analyses.

Sensor	Sampling Frequency	Filter Order	Filter Cutoff Frequency
-	Hz	-	Hz
APS	1.000	8	0.100
Gyroscope	6.622	6	0.600
DVL	6.622	4	2.000

Table 4.4: FIR filter parameters for the ROV navigation data.

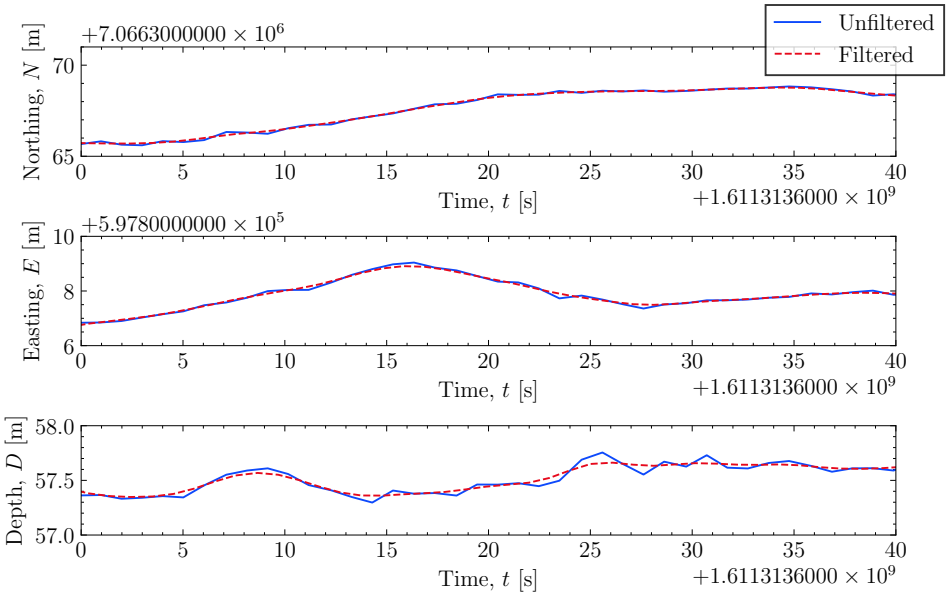


Figure 4.8: FIR filtered APS northing, easting, and depth measurements.

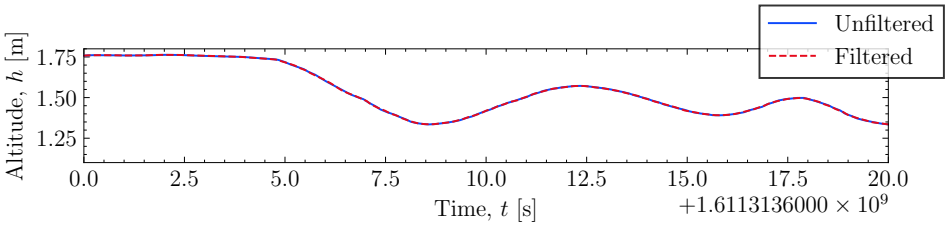


Figure 4.9: FIR filtered DVL altitude measurements.

4.3 Data Synchronization

Figure 4.10 shows the synchronization points that were identified for the two dives during the Ekne Survey, and their corresponding estimated time biases. The relatively large

spread of the estimated biases, was an indication that simply adding a constant time bias based on the methodology in Section 3.6, would be inadequate for synchronizing the ROV and Jetson TX2 CPU clocks. Table 4.5 shows the mean biases, the corresponding standard deviations, as well as the tunable bias corrections utilized for the V-SLAM trajectories. From the tunable bias corrections one can see that, in general, small positive bias corrections had to be added in order to get good correspondences between the ground truth trajectories and the OpenVSLAM trajectories.

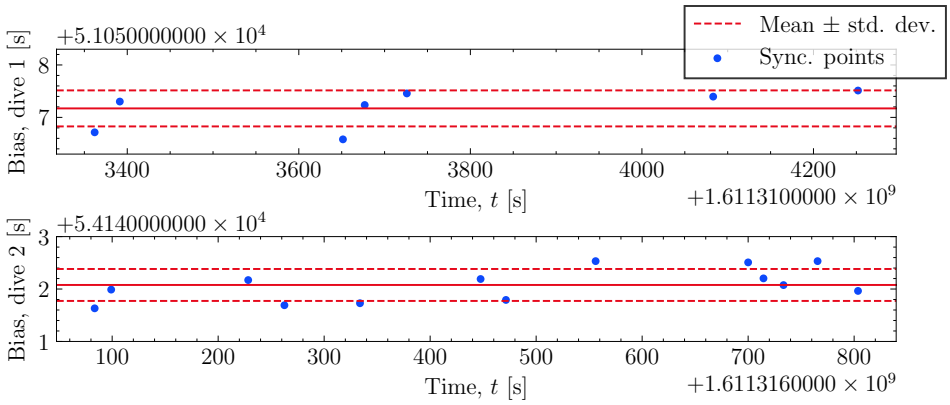


Figure 4.10: Synchronization points and estimated mean bias for Dive 1 and Dive 2.

There are several sources of error with the synchronization methodology. The assumption that the time bias between the clocks was constant is inaccurate, as CPU clocks drift over time and are highly affected by temperature. Additionally, the assumption that there was no delay in the Helmsman overlay system is questionable, as both transmitting the data from the ROV and rendering it to the topside display inherently introduces time delays. Finally, the linear interpolation of the timestamps of ROV video footage frames in order to acquire sub-second precision is highly susceptible to the video capturing software. Since the tunable bias corrections were all small positive values, it is believed that a majority of the tunable bias corrections originated from the overlay system delays. The origin of the small fluctuations in the estimated biases were, however, not identified, but it is believed that these were a combination of CPU clock drift and interpolation errors.

Trajectory	Mean Bias and Std. Dev.	Tunable Bias Correction	Unit
1	51057.171 ± 0.371	0.6	s
2	51057.171 ± 0.371	0.6	s
3	51057.171 ± 0.371	0.6	s
4	51057.171 ± 0.371	0.6	s
5	54142.115 ± 0.317	0.6	s
6	54142.115 ± 0.317	0.4	s
7	54142.115 ± 0.317	0.4	s
8	54142.115 ± 0.317	0.7	s
9	54142.115 ± 0.317	0.8	s
10	54142.115 ± 0.317	0.6	s

Table 4.5: Timestamp corrections for the synchronized V-SLAM trajectories.

4.4 Image Processing

Table 4.6 and Table 4.7 show the tuned parameters for the BLF and CLAHE. For the BLF, the filter diameter, d_{BLF}, was set to an intermediate value, as a trade-off between loss of feature awareness and computational complexity. The spatial variance, σ_{s}, was also set to a moderate level, as too small values caused noise to pass through the filter, and too large values caused loss of local texture. The radiometric variance, σ_{r}, was set relatively high, in order to suppress false edges along object edges, an artifact known to be introduced by the BLF (Kornprobst et al., 2009). For the CLAHE, the clipping limit was kept at a moderate level, as setting it too high contributed to significant amplification of the image noise, which could not be filtered out by the BLF. The CLAHE image patches were set to be quadratic and were, initially, kept relatively small. Increasing the image patch size reduced the noise amplification, but also reduced the effect of the local contrast enhancement. The image patch size was set as a trade-off between the two effects.

Parameter	Value	Description
d_{BLF}	10	Filter diameter.
σ_{r}	60	Radiometric standard deviation.
σ_{s}	20	Spatial standard deviation.

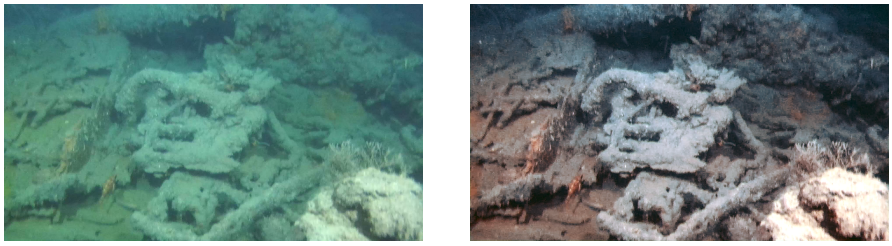
Table 4.6: Tuned BLF parameters.

Parameter	Value	Description
c	2.0	Histogram clipping limit.
w	20	Window width.
h	20	Window height.

Table 4.7: Tuned CLAHE parameters.

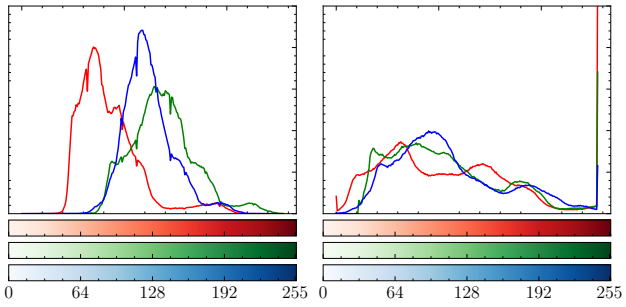
In Figure 4.11a, a raw RGB image from the Ekne wreck site is shown. From the image one can see the effects of light attenuation, outlined in Section 2.2.2, on the colors of the image.

The water molecules absorb light towards the red end of the visible spectrum, meaning that mainly light towards the green and blue parts of the spectrum manages to reach the camera. This is also visible in the RGB image histogram for the image in Figure 4.12a, where the red histogram is shifted left, towards the lower intensities, compared to the green and blue histograms. In Figure 4.11b, one can see the same RGB image processed by UIENet. The restoration of the red color channel is evidence that UIENet manages to remove a significant portion of the light attenuation effect in the image. From the corresponding RGB histogram in Figure 4.12b, one can see that in addition to the shift in the red channel, UIENet also increases the contrast of all three channels, evident by the wider channel histograms. One remark that is worth noting is the histogram peak on the right side of the histogram, at intensity level 255. This peak is abnormal, and might be due to the underlying underwater image formation model. Specifically, UIENet is based on the revised underwater image formation model, outlined in Section 2.2.4, which assumes ambient illumination. A lamp was used in this case, causing the scene illumination to be relatively uneven. This uneven illumination is not captured by the image formation model, and might be a reason for the high pixel intensities seen in the lower right corner of Figure 4.11b.



(a) **(b)**

Figure 4.11: RGB images from the Ekne wreck site; a) a raw RGB image, b) the same RGB image processed by UIENet.



(a) **(b)**

Figure 4.12: RGB image histograms; a) the unprocessed RGB image, b) the RGB image processed by UIENet.

Figure 4.13 shows the image histograms for the grayscale image processed by the various IP methods. In Figure 4.13b one can see that the BLF does not affect the contrast, evident by the similar shape of the histogram compared to the histogram of the raw grayscale image, seen in fig. 4.13a. The BLF does, however, create some small pits in the histogram, which is a consequence of the sharpness enhancing effect of the method. This pitting effect can also be seen in the other histograms where BLF has been applied to the image, i.e. Figure 4.13c and Figure 4.13d. These figures also highlight the effect of contrast enhancement by the wider histograms. When comparing the HE- and CLAHE processed histograms, the clipping limit of CLAHE is evident by the slightly lower contrast enhancement effect. A contrast enhancing effect can also be seen in fig. 4.13e for the grayscale image of the RGB image processed by UIENet.

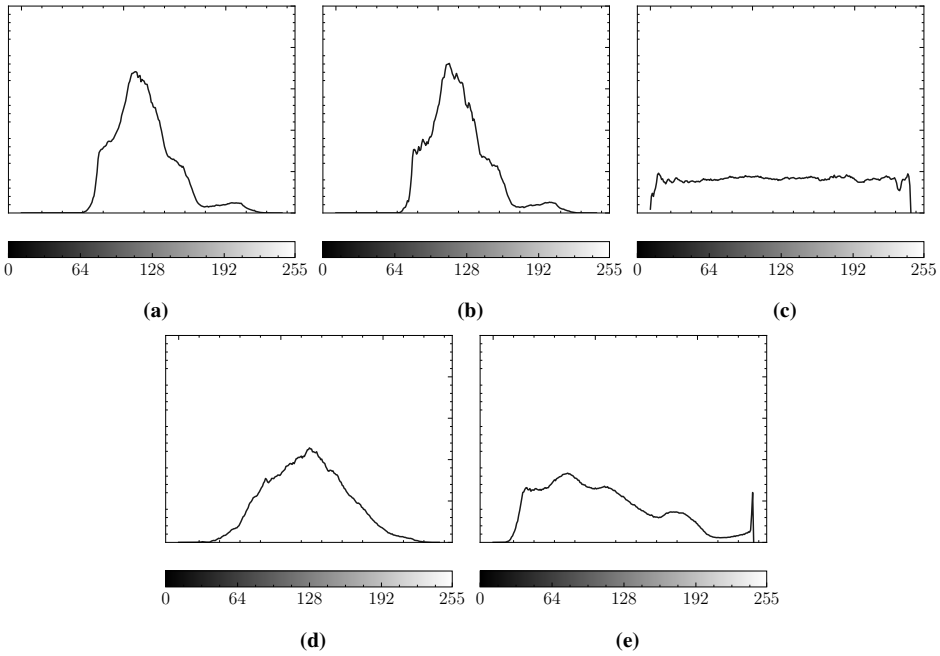


Figure 4.13: Grayscale image histograms; a) the unprocessed grayscale image, b) the grayscale image processed with BLF, c) the grayscale image processed with HE-BLF, d) the grayscale image processed with CLAHE-BLF, e) the grayscale image processed with UIENet.

In Figure 4.14 one can see the similarity images of the processed grayscale images compared to the raw grayscale image, computed with structural similarity index method (Wang et al., 2004). Darker and brighter pixels in the similarity images indicate larger and smaller changes, respectively, compared to the raw grayscale image. In Figure 4.14a the noise suppression effect of the BLF can be seen from the sporadic black pixels throughout the image. Additionally, one can see the edge sharpening effect of the filter from the narrow strips of gray pixels near object edges. This effect was found to be an effective way of suppression forward scattering, which appear as a slight blur around object edges. In Figure 4.14b one can see that the noise suppression effect of the BLF is prominent when

compared to Figure 4.14a, with only a small amount visible noise in the foreground of the scene. This could be an indication that the BLF is less effective or requires re-tuning when used in conjunction with HE. When coupled with CLAHE, the noise suppression effect of BLF is still quite prominent, as seen from the sporadic dark pixels uniformly distributed in Figure 4.14c. From Figure 4.14d one can see that UIENet performs quite sophisticated changes in the images. Since UIENet corrects for light attenuation, which is distance dependent, as seen in Equation 2.14, one would expect to see larger changes in the background of the scene than in the foreground. This is in line with what we see in the similarity image, where the darker pixels, in general, correspond to the background of the scene. Since UIENet does not get any other information than the image of the scene, it is evident that the model has learned geometric features to deduce the scene attenuation and, correspondingly, the scene depth.

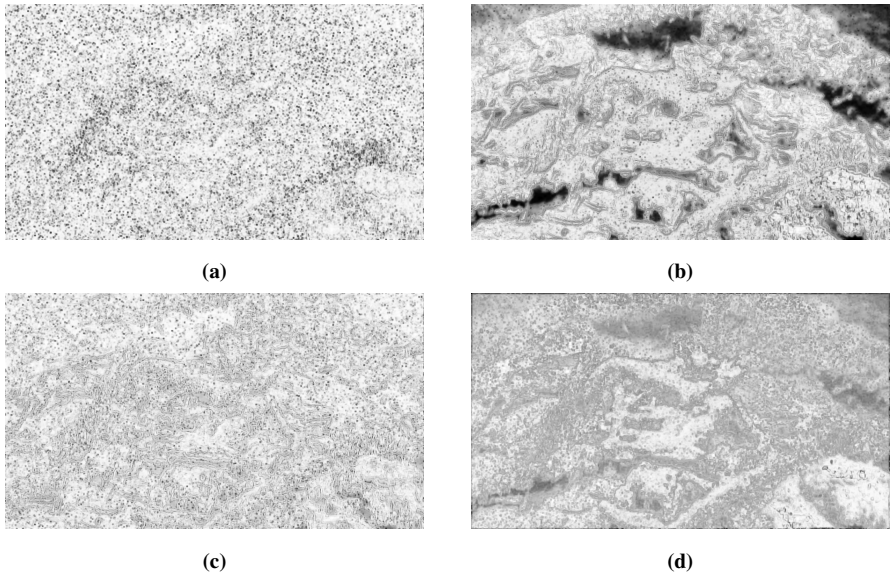


Figure 4.14: Similarity images for the grayscale image processed with; a) BLF, b) HE-BLF, c) CLAHE-BLF, and d) UIENet.

An aspect that is worth noting within the context of V-SLAM, is the processing time of the image processing methods. Figure 4.15 shows that the processing time of UIENet is close to ten times higher compared to that of the other image processing methods, despite the fact that the implementation is GPU-accelerated. This is an indication that the model architecture of UIENet is not suited for real-time V-SLAM applications, since these, in general, rely on a high frequent image data. Since real-time capabilities is one of the selling points of utilizing V-SLAM over structure-from-motion, this fact is quite detrimental for further adaptation of UIENet in underwater V-SLAM applications. However, this does not exclude DL-based underwater image processing methods from being adapted for V-SLAM. Some possibilities, that are still open research questions, are utilizing the visual depth information in conjunction with the RGB images in a DL-based underwater image

enhancement method. It is possible that this additional geometric information could reduce the needed model complexity for backscatter and direct transmission estimation, which would make the model faster. This approach is, of course, dependent on consistent visual depth estimates, which for real-time applications would require stereo vision or auxiliary distance measurements.

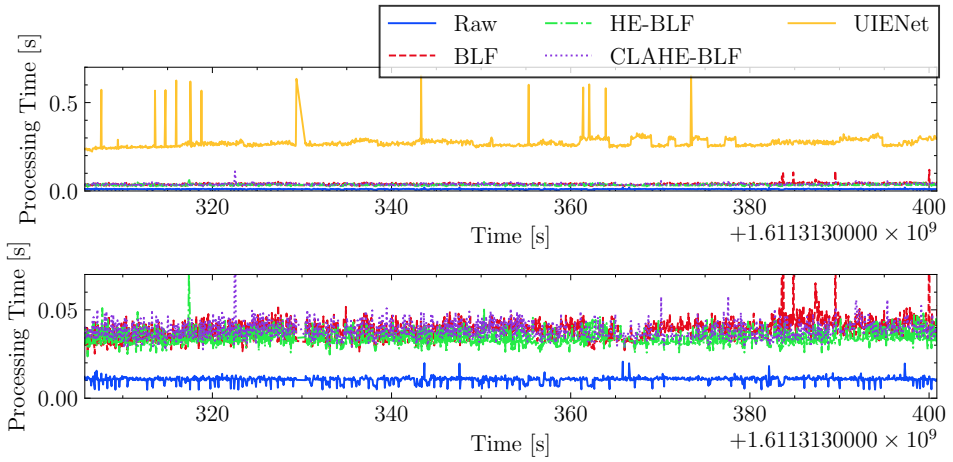


Figure 4.15: Processing times for the various image processing methods, shown at two different time scales.

4.5 Georeferencing

In total 10 different trajectories from the Ekne wreck site survey, where OpenVSLAM was able to keep track over a significant period, were identified. Figure 4.16 and Figure 4.17 show the georeferenced positions and attitude from OpenVSLAM with combined CLAHE and BLF image processing, which have been georeferenced with the methodology outlined in Section 3.8. Note that the attitudes are the XYZ Euler angles, r_x, r_y, and r_z, of the quaternion describing the rotation from the camera CS to the world CS, ${}^{w}_{c}\mathbf{q}$, and cannot be interpreted as roll, pitch, and yaw, since the axes of the camera CS are not aligned with the axes of the body CS. One can see that the georeferenced OpenVSLAM positions do, in general, correspond quite well to the calculate ground truth trajectory. The OpenVSLAM trajectories do, however, accumulate drift over time due to errors in the pose estimates. This accumulated drift is likely the reason for the discrepancy between the ground truth positions and the OpenVSLAM position estimates, which is evident in the UTM Northing estimates around the 500- and 700 second mark and the UTM Easting estimates around the 770 second mark. In Figure 4.17 one can see that the correspondences between the ground truth attitudes and the estimated attitudes, arguably, worse than the correspondences between the positions. This disprecancy might just be an effect of the georeferencing method, as Umeyama's method does not include the attitudes in the least squares optimization problem. The inclination angle of the camera is another possible

reason for the deviations in the attitude estimates. The inclination angle of the camera was measured by hand with a protractor and therefore had a high amount of uncertainty. Another remark that is worth mentioning about the georeferencing is the navigation data that was used to create the ground truth reference. The navigation data is based on filtered sensor measurements and therefore has varying degrees of accuracy and uncertainty associated with it. The uncalibrated gyrocompass, mentioned in Section 4.2, is a considerable source of uncertainty and error. For this reason, the uncertainty of the ground truth reference should be have been analyzed to see if the OpenVSLAM estimates were inside the appropriate confidence intervals. This could, for instance, have been conducted by generating the ground truth reference with a probabilistic filter, such as a Bayes filter. This could also have been used to fuse the depth measurements from the APS with the depth measurements from the pressure sensor, which was not utilized in the ground truth reference in this project.

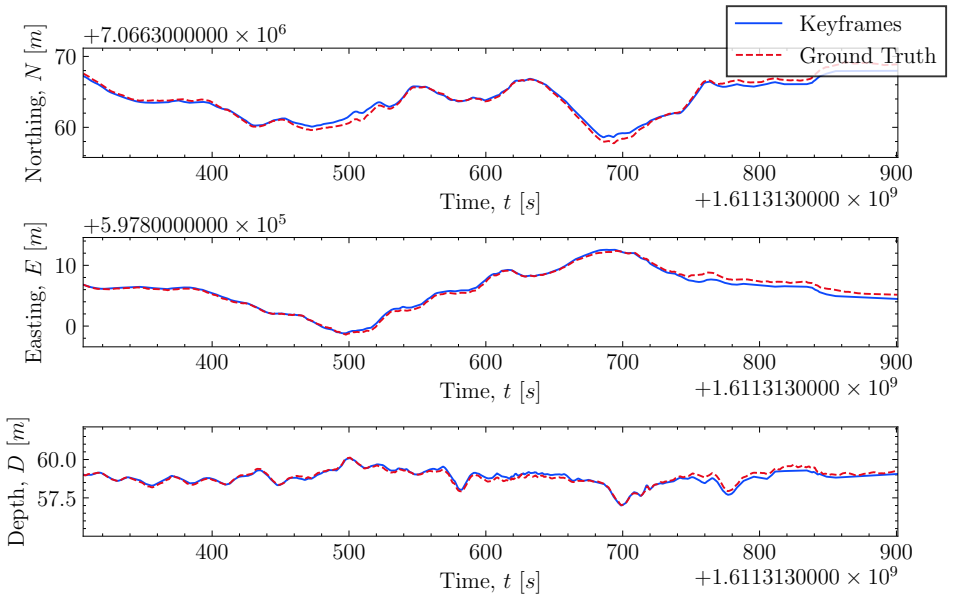


Figure 4.16: Georeferenced OpenVSLAM position estimates for Trajectory 1.

Figure 4.18 and Figure 4.19 show the survey map with the georeferenced trajectories and the extent of the georeferenced maps from Dive 1 and Dive 2, respectively. The maps are provided to give an overview of the location and length of the different trajectories. The shown trajectories and maps were generated from OpenVSLAM with the combined CLAHE and BLF image processing method. As explained in Section 3.2, Dive 1 consisted of planned transect lines across the mid-ship of the wreck site, while Dive 2 consisted of less planned maneuvering of the ROV in the more geometrically complex parts near the stern and aft. As a consequence of this difference in maneuvering, the distance between the seabed and the camera was, in general, larger for Dive 2 than for Dive 1. This meant that the stereo image pairs from Dive 2 were more prone to light attenuation, the effect of which

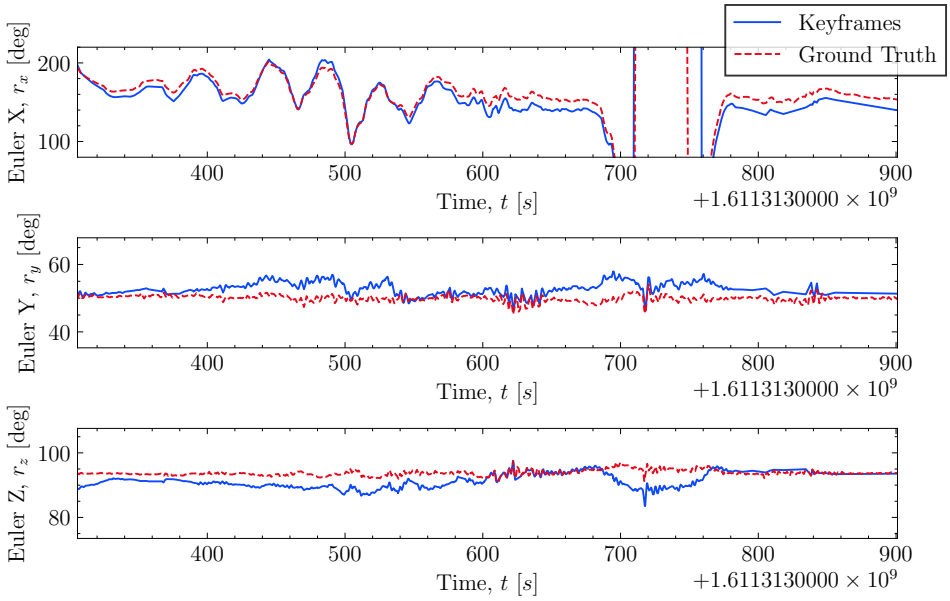


Figure 4.17: Georeferenced OpenVSLAM attitude estimates for Trajectory 1.

was shown in Figure 4.11a. Dive 2 also had multiple points were visual contact between the seabed and the stereo camera was lost completely. Based on the difference of the track lengths in Figure 4.18 and Figure 4.19, it is evident that this difference in the maneuvering of the ROV had a large impact on the robustness of OpenVSLAM and its ability to provide consistent tracks. Maintaining a minimal distance between the camera and the seabed, as well as maneuvering with consistent altitude, are therefore found to be important factors for the robustness of V-SLAM methods when used underwater. Since maneuvering of UUVs is, in general, dictated by the risk of vehicle loss, which is highly dependent on the site bathymetry, consistent maneuvering close to the seabed cannot always be guaranteed. Robust adaptation of V-SLAM for underwater navigation is therefore dependent on the bathymetry of the relevant survey site. Another way to minimize the distance between the camera and the seabed is to configure the inclination angle of the camera to be suitable for the site bathymetry. The 45 degree inclination angle that was used during the Ekne wreck site survey was considered to be too small for the relevant survey. The inclination angle was, however, set due to fact the downward facing lamps underneath the ROV were unreliable and did not providing consistent illumination. This shows that a reliable lamp setup and a camera mounting position that is suitable for the relevant site bathymetry, are practical factors that need to be considered in order to get robust setups for underwater V-SLAM.

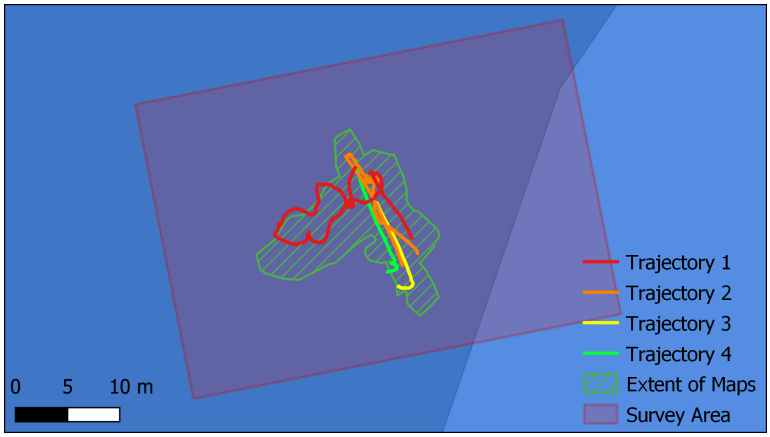


Figure 4.18: Georeferenced trajectories and extent of maps from OpenVSLAM for Dive 1.

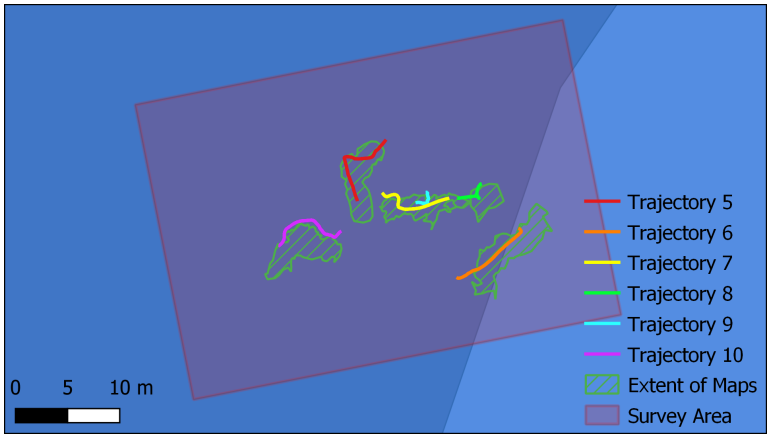


Figure 4.19: Georeferenced trajectories and extent of maps from OpenVSLAM for Dive 2.

4.6 V-SLAM Comparative Analysis

All the results in this section are generated with OpenVSLAM with various image processing methods. For simplicity, the various configurations of OpenVSLAM are at times just referred to by the acronym of the relevant image processing method.

In order to evaluate how the different image processing methods affect the feature extraction and overall performance of OpenVSLAM, a comparative analysis was performed on one of the trajectories. Trajectory 1 was chosen due to the consistent low altitude and the possibility of loop detection, and consequently, loop closing. Throughout the analysis, the tuned OpenVSLAM parameters in Table 4.8 were used. Some parameters that are worth

noting are the FAST thresholds and the minimum number of triangulated points. The FAST thresholds were initially set relatively low in order to get a sufficient amount of features extracted due to the relatively low contrast of the stereo image pairs. The minimum number of triangulated points also had to be set relatively low in order to maintain track in low-texture regions.

Parameter	Value	Unit
Maximum number of features	1000	-
Image pyramid levels	8	-
Image pyramid scale	1.2	-
FAST initial threshold	7	-
FAST backup threshold	3	-
Depth threshold factor	30	-
Minimum number of triangulated points	10	-
Baseline distance threshold	0.10	m

Table 4.8: Tuned OpenVSLAM parameters.

Since modern V-SLAM systems are multi-threaded, they exert stochastic behaviour due to race conditions. In order to compensate for this stochastic behaviour, OpenVSLAM was run 10 times on Trajectory 1 for each of the image processing methods, and then the longest track was used for further analysis. Figure 4.20 shows the distribution of the achieved track lengths for each of the image processing methods, i.e. the number of frames over which OpenVSLAM was able to keep track. From the track lengths one can clearly see a significant difference between the image processing methods. In terms of track length, OpenVSLAM with BLF, HE-BLF, and CLAHE-BLF clearly outperform OpenVSLAM with UIENet and the raw image pairs.

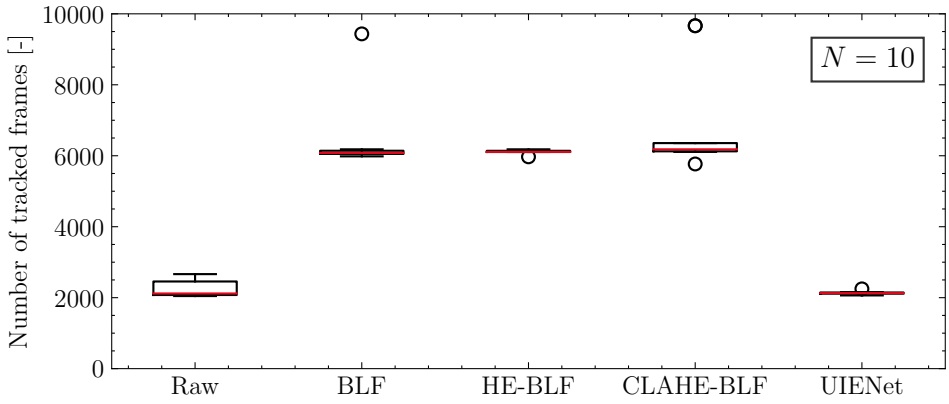


Figure 4.20: Track lengths for ten runs of Trajectory 1 for various image processing methods.

4.6.1 Feature Distributions

To evaluate how the various image processing methods affected the feature extraction and feature matching of OpenVSLAM, a statistical approach was taken. To clarify, a matched feature is an image feature which is extracted in one image and then found in the following image. In Figure 4.21 and Figure 4.22, one can see how the extracted- and matched features are distributed throughout the image pyramid of OpenVSLAM. The image pyramid is utilized by OpenVSLAM to extract features at different scales of the images, where features extracted at level one corresponds to small, detailed features, and features extracted at level eight corresponds to larger, less detailed features. From Figure 4.21 one can see that the image pyramid distribution of the extracted features are close to identical for the various image processing methods. This means that the relative relation between small-scale and large-scale features is close to equal for the extracted features for the various image processing methods.

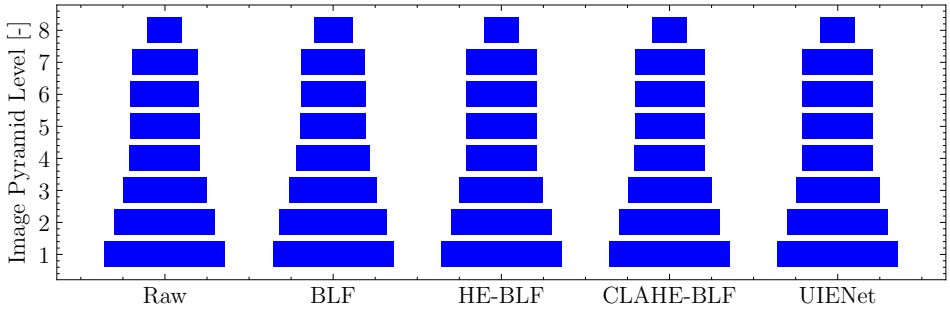


Figure 4.21: Image pyramid distribution of extracted features over Trajectory 1.

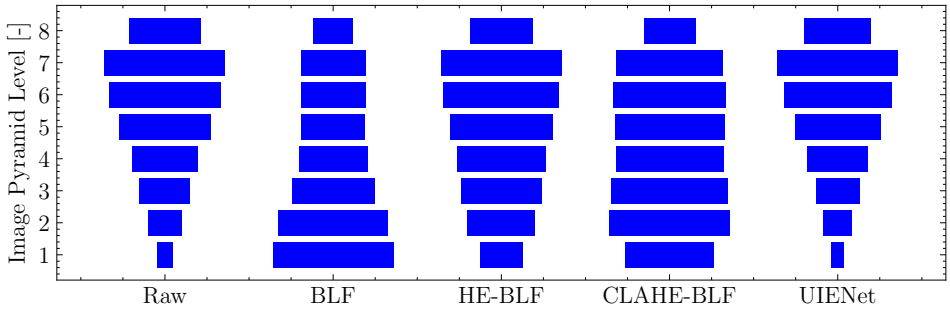


Figure 4.22: Image pyramid distribution of matched features over Trajectory 1.

However, when one examines the pyramid distribution of the matched features in Figure 4.22, the distribution is significantly different for the various image processing methods. Specifically, the top heavy pyramid distribution of the raw and UIENet image pairs means that OpenVSLAM does not find matches among small-scale features, but only among large-scale features. A similar distribution can be seen for the HE-BLF image pairs, but the distribution is not as skewed as for the raw and UIENet image pairs. For the BLF

image pairs, one can see that OpenVSLAM is able to match a lot of small-scale features, and that the distribution is relatively even for the subsequent levels. For the CLAHE-BLF image pairs the distribution is relatively even, which means that the proportion between small-scale and large-scale feature matches is equal.

To understand the reason for the distribution difference between the extracted and the matched features, one can consider the observations that was made in Section 4.4, more specifically in Figure 4.14. From the similarity images it was established that the BLF was effective at suppressing noise and forward scattering blur. This effect was also considerable when the BLF was applied in conjunction with CLAHE, but was less clear when applied in conjunction with HE. UIENet, on the other hand, did not seem to have any effect on noise suppression or reduction of forward scattering blur. All these observed image processing effects are consistent with the distribution differences. Based on these observations, one can make the proposition that suppression of image noise and forward scattering blur is important to increase the ratio between matched small-scale and large-scale features.

4.6.2 Robustness

Now that it has been established that suppression of image noise and forward scattering blur is important for the distribution of matched features, it is interesting to investigate how this affects OpenVSLAM's tracking robustness. From the track lengths in Figure 4.20, it is evident that OpenVSLAM is more robust with BLF, HE-BLF, and CLAHE-BLF, than with the raw and UIENet image pairs. From Figure 4.23, one can see that OpenVSLAM is, in general, able to reach the limit of 1000 extracted features for all image processing methods, which is due to the relatively low FAST thresholds, as seen in Table 4.8. Note that for the selected tracks, OpenVSLAM with BLF lost track at the 690 second marker, and then later relocated and regained track. For this reason, the corresponding plot lines are not representative after the 690 second marker, as seen in Figure 4.23. This means that the number of extracted features can not be the reason for the difference in tracking robustness. However, from Figure 4.24 one can see a clear difference in the amount of matched features for the different methods. Specifically, one can see that OpenVSLAM, overall, matches a significantly lower amount of features for the raw and UIENet image pairs, than for the other methods. From this observation and the discovery in Section 4.6.1, one can draw the conclusion that suppressing image noise and forward scattering blur is important to improve small-scale feature matching, and that small-scale feature matching is important to increase the total number of matched features, and, consequently, track robustness. In terms of this effect, the BLF clearly makes a significant difference on the number of features that OpenVSLAM is able to match, and consequently has a significant impact on the overall robustness.

Since BLF was applied in conjunction with HE and CLAHE, the effect of these contrast enhancing methods is still unclear. Based on the track lengths in Figure 4.20, one can conclude that the difference in terms of overall robustness is minor, since most of the runs with BLF, HE-BLF, and CLAHE-BLF loose track around frame 6100. Initially, one would think that the increased contrast would improve OpenVSLAM's ability to extract and match features at larger visual depths. To evaluate this hypothesis, consider the time

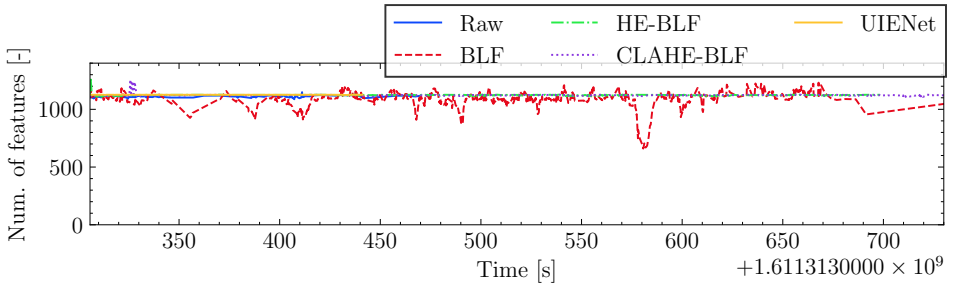


Figure 4.23: Number of extracted features for Trajectory 1.

point around the 580 second marker in Figure 4.23 and Figure 4.24. At this point in time the ROV rises up to an altitude of about 2.5 meter, as seen in the altitude measurements in Figure 4.25, which causes the visual depth within the FOV of the camera to become quite large. One can see that the number features that is extracted with BLF at this time point drops significantly. In contrast, OpenVSLAM with HE-BLF and CLAHE-BLF still reach the maximum number of extracted features of 1000. From the matched features at the same time point, one can see that, despite this difference in extracted features, OpenVSLAM with BLF is able to match more features than HE-BLF and CLAHE-BLF. Even in this situation, where one would initially believe that the contrast enhancement would improve the tracking, this does not seem to be the case. Taking this into consideration, in addition to the fact that OpenVSLAM with BLF has a higher number of matched features overall, the evidence seems to suggest that contrast enhancement is not of significant importance to improve tracking robustness.

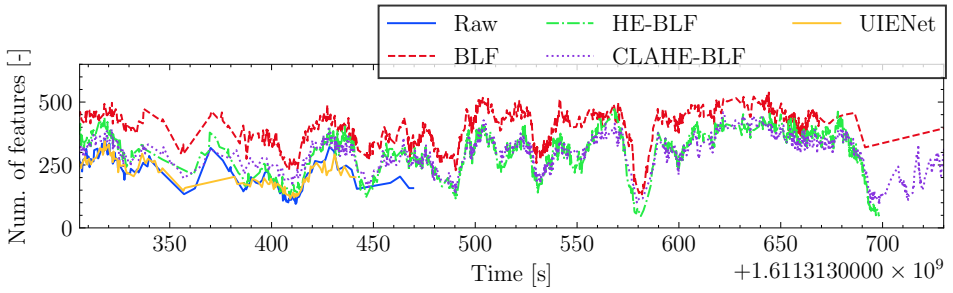


Figure 4.24: Number of matched features for Trajectory 1.

4.6.3 Absolute Trajectory Error and Relative Pose Error

In addition to the feature distributions and robustness, the standard comparison metric, ATE and RPE, were analyzed. In Figure 4.26, one can see the translational and rotational components of the ATE, defined in Equation 3.19a and Equation 3.19b, for the various image processing methods. From the translational components, one can see that there are large variations between the methods. The translational components for the raw and

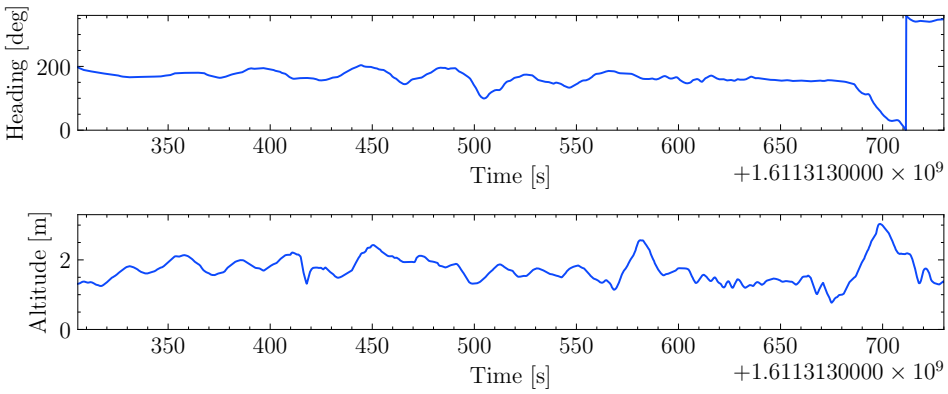


Figure 4.25: Heading and altitude measurements for Trajectory 1.

UIENet image pairs are lower than the other, up until the point where OpenVSLAM looses track with the respective image processing method. Additionally, the translational component of OpenVSLAM with BLF, HE-BLF, and CLAHE-BLF, are very similar in shape, but seem to be scaled. Since the ATE utilizes the absolute difference between the ground truth reference and the georeferenced trajectories, the ATE is highly dependent on the performance of the georeferencing methodology. The large variation in trajectory lengths in this case, means that there is a large variation in the accumulated drift in the trajectories. This difference in accumulated drift has a high effect on the georeferencing results, and consequently the ATE. For this reason, the ATE is considered to be a highly biased measure and is not considered in the comparison of the various methods.

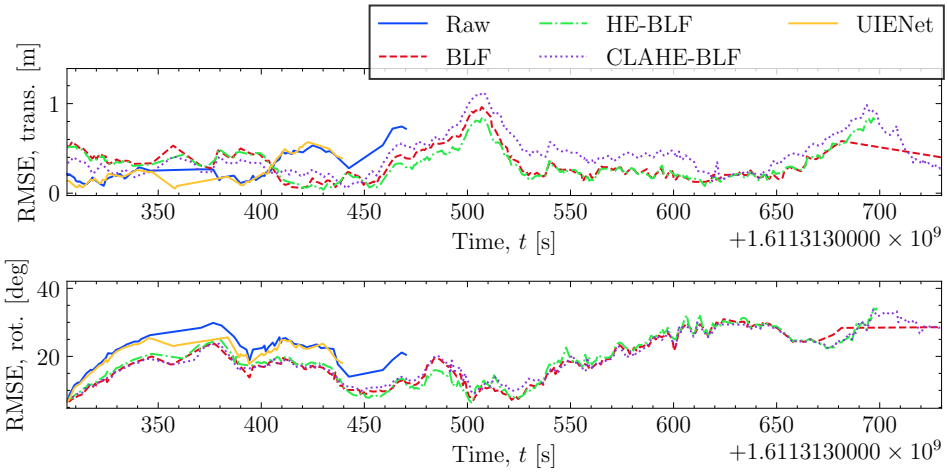


Figure 4.26: Absolute trajectory errors for Trajectory 1.

In contrast to the ATE, the RPE uses relative position differences, and is therefore more

suited to compare trajectories of various length. Figure 4.27 shows the translational and rotation RPE components, as defined in Equation 3.21a and Equation 3.21b. Note that due to the lost and later regained track of the BLF, the corresponding RPE metrics are not representative from the 650 second marker and onward. The reason for the slight time shift of this point is that the RPE compares the trajectory to a future reference point, which is displaced Δ units in space. In both the translational and rotational components one can see that for the raw and UIENet image pairs, the RPE is slightly higher than the other methods, but not by a considerable margin. The BLF, HE-BLF, and CLAHE-BLF are, generally, quite similar in terms of the RPE, with no significant different between them. An observation that is worth noting is that the peaks in the rotational RPE, at 410, 440, 490, and 525 seconds, correspond to changes in the heading, as seen in Figure 4.25. The explanation for this is that the stereo camera is a perspective camera, and therefore has a limited FOV. When the ROV turns, the camera ultimately looses sight of visual features, forcing OpenVSLAM to continually find new features to track. This means that OpenVSLAM can not track the same visual features over longer periods, which is the reason for the observed drift peaks.

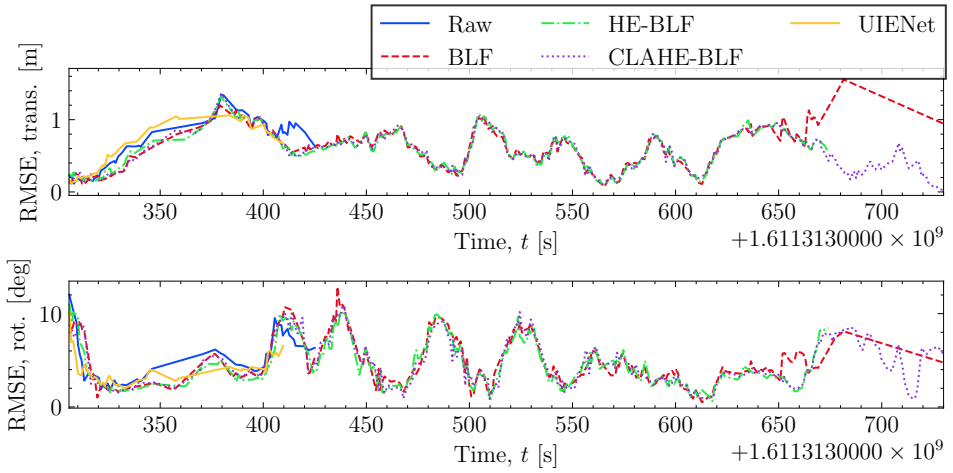


Figure 4.27: Relative pose error with $\Delta = 5$ meter for Trajectory 1.

In addition to the ATE and RPE, the normalized estimation errors squared (NEES) and normalized innovation squared (NIS) are additional metrics that would have been interesting to analyze for comparison (Fornasier et al., 2021). NEES and NIS are consistency metrics, which are used to check whether the V-SLAM method is consistent with the assumed probability distributions, such as the Gaussian assumptions in the full SLAM standard model in Equation 2.41. NEES and NIS are commonly analyzed for probabilistic filters, like the Kalman Filter, but are seldom analyzed in the context of SLAM. The reason for this is that computation of NEES and NIS requires access to the covariance matrices, which are often left out of implementations, due to the extra performance and memory savings. Due to the extra work with implementing the required changes into OpenVSLAM, this was not attempted in this project. Performing a comparative consistency analysis is, however,

highly recommended as further work for this project.

4.7 V-SLAM Qualitative Analysis

4.7.1 Dynamic Targets

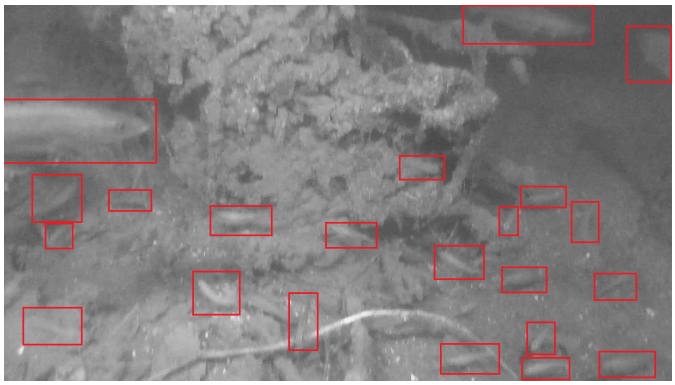
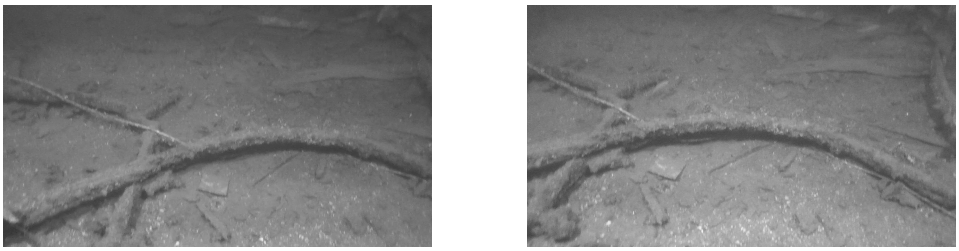


Figure 4.28: Dynamic targets highlighted by their bounding boxes. The high amount of dynamic targets causes OpenVSLAM's pose estimation to fail and the track is lost.

In addition to the quantitative comparison in Section 4.6, a qualitative evaluation of some of the underlying models of OpenVSLAM was conducted in order to assess their validity for underwater use. In Figure 4.28, an image from Trajectory 10 is shown, where the fish are marked with bounding boxes to highlight them. In this situation the ROV moves into an area with several fish schools. Since the stereo camera has visual contact with the fish schools over an extended period, in which the fish are relatively still, OpenVSLAM ultimately starts to track visual features on the fish. When the fish suddenly start to move in multiple directions, OpenVSLAM is not able to estimate a pose which resolves the change in the landmarks, and therefore looses track.

The reason why this happens, is that the standard model of the full SLAM problem, outlined in Section 2.5.2, assumes that the map is static. The normal approach of dealing with dynamic targets, which is also the approach that OpenVSLAM uses, is to reject them as outliers by using RANSAC. However, when the number of dynamic targets becomes too high, RANSAC is no longer able to reject them as outliers, and they get considered in the pose estimation. This highlights a limitation of the full SLAM standard model, which can be quite severe in certain underwater scenarios. In addition to large fish schools, another underwater scenario where static map assumption can be critical, is kelp forests moving due to currents or waves. While SLAM formulations with dynamics targets have been proposed (Sola, 2007), they do, in general, suffer from high computational complexity due to the high number of target hypotheses, and have in large part been neglected in favour of the real-time capabilities of the standard model.

4.7.2 Loop Detection



(a) (b)

Figure 4.29: Loop closure candidate images for Trajectory 1; a) Visit 1, and b) Visit 2.

As pointed out in Section 1.1, the main motivation of adopting V-SLAM over VO is the map estimate which enables loops to be detected, and, consequently, the accumulated drift to be corrected. Out of the 50 OpenVSLAM runs that was conducted on Trajectory 1 in Section 4.6, all of the runs with BLF, HE-BLF, and CLAHE-BLF managed to keep track throughout the entire trajectory loop. Out of these 30 runs, not a single loop closure was detected by the BOW-based loop detection, outlined in Section 2.5.9. Figure 4.29 shows two candidate images for loop detection, where the image in Figure 4.29b is taken approximately 5 minutes after the image in Figure 4.29a. Based on the similarity of the images, one would believe that OpenVSLAM would be able to detect the loop closure. It is possible that the image processing methods could have a negative impact on the distinction of the BOW features, or that the BOW vocabulary of OpenVSLAM is unsuited for underwater use. To distinguish why the BOW-based loop detection fails, more in-depth quantitative analyses would have to be performed. BOW-based loop detection methods have, however, been shown to be susceptible to illumination changes (Milford and Wyeth, 2012), which is a possibility why the method fails in this case, as the light source moves with the ROV. Less computational and memory intensive loop detection methods, like hash table-based approaches (Bonin-Font et al., 2014), have been proposed to replace the BOW-based approaches. The most modern approaches for loop detection are DL-based image hash approaches (Bonin-Font and Burguera, 2021), which utilize fast CNNs to create the image hashes. These approaches have proven to be more reliable, more efficient, and simpler than previous methods. However, the effects of integrating a DL-based image hash loop detection method with an underwater V-SLAM system are still open research questions. Considering the lack of performance of the BOW-based approach of this project, integration of a DL-based image hash loop detection method into OpenVSLAM is highly recommended as further work.

Chapter 5

Conclusion

5.1 Conclusion

This project investigated the validity of using V-SLAM for underwater navigation, and identified parameters that are important for the robustness and drift of underwater V-SLAM algorithms. Additionally, an evaluation of the full SLAM standard model and a BOW-based loop detection method was performed.

A dataset suitable for underwater V-SLAM was collected with a stereo camera mounted on a ROV. The dataset was used to perform a comparison analysis of the V-SLAM algorithm OpenVSLAM, with four different image processing methods. Based on the comparison analysis, filtering of image noise and forward scattering blur were found to be important factors for underwater V-SLAM robustness, due to the improved ability to match small-scale visual features. Due to this effect and the fast computational speed, the BLF was found to be a very good image processing method for underwater V-SLAM. In terms of the accumulated drift, the various image processing methods were found to have little effect on the RPE. Maneuvering with a lot of turns was, however, identified to cause a lot of peaks in the RPE and therefore a significant source of accumulated drift.

By comparing two dives with very different maneuvering patterns, proper maneuvering was found to be a very important factor for underwater V-SLAM robustness. Specifically, maneuvering with a low altitude was found to be important for the stereo camera to maintain good visual contact with the seabed, and avoiding sharp turns was found to be important to reduce motion blur. Having a properly configured camera- and lamp setup for the relevant survey site, was also identified to be crucial elements to improve visual contact and illumination, and, consequently, V-SLAM robustness.

One of the underlying models and a submethod of the OpenVSLAM algorithm were identified to be problematic for underwater V-SLAM. Specifically, the static map assumption of the full SLAM standard model was shown to be invalidated in the presence of a high

number of dynamic targets. Moreover, the BOW-based loop detection method was found to be inappropriate for underwater V-SLAM, as it is susceptible to illumination changes and, therefore, did not detect a single loop in the dataset.

5.2 Further Work

State Estimation and Georeferencing

Some recommendations for further work to this project is to improve upon the analyses that was conducted in this project. In this regard, there are several recommendations for further work that are closely related. The first is to improve upon the navigation data processing, and the establishment of the ground truth reference. A specific approach is to utilize a probabilistic filter, such as a Kalman Filter, which takes the measurement uncertainties into consideration. By employing a probabilistic filter, not only would one improve the accuracy of the ground truth reference, but one would also gain access to the ground truth covariance. Closely related to this, is to establish a georeferencing method which takes both positions and attitudes, as well as uncertainties into consideration.

Consistency Analysis

Also related to uncertainties, is to implement functionality to retrieve the covariance matrices for the pose- and landmark estimates from OpenVSLAM. Pose- and landmark estimates, and their corresponding covariance matrices can then be used to perform a consistency analysis of OpenVSLAM.

Guidance and Control

Considering the requirements that underwater V-SLAM puts on maneuvering, it is interesting to investigate guidance- and control laws for this specific purpose. For instance, one could investigate guidance laws that utilize visual information to guide the maneuvering of the UUV. The gains of such a system could be improved visual contact between the camera and the seabed, and less critical maneuvers, like sharp turns at high altitude. This could help increase the overall robustness of underwater V-SLAM algorithms.

Refractive Camera Models

A recommendation for further work within the field of photogrammetry would be to develop a refractive camera model for cylindrical housings. This refractive camera model could then be implemented in OpenVSLAM, to examine the systematic errors that are introduced by omitting refraction. This is an ambiguous project, as there has been no study of underwater refractive V-SLAM, to the author's knowledge. There are, however, studies where refractive camera models have been integrated into SFM methods with great success (Xiaorui et al., 2019). It is hard to believe that underwater V-SLAM algorithms would not see similar benefits. Adapting a refractive camera model for a flat plane housing, such as the Pinax model (Łuczyński et al., 2017), could also be an interesting project, but would require the collection of a new dataset.

Adaptive Edge-Preserving Filters

In the wake of the success of the bilateral filter in this project, a recommendation for further work is to investigate the more advanced version, the fast adaptive bilateral filter (Gavaskar and Chaudhury, 2019). The fast adaptive bilateral filter has improved on some of shortcomings of the bilateral filter, while still being able to process images at a high rate. Another image processing filter that could be interesting for underwater V-SLAM is the guided filter (He et al., 2013), which is another edge-preserving filter, just like the bilateral filter.

Deep Learing-Based Loop Detection

Perhaps the most interesting recommendation for further work, in the context of V-SLAM, is to integrate a DL-based image hashing method for loop detection in OpenVSLAM. A method that is particularly interesting is NetHALOC, a CNN which has been specifically engineered for underwater loop detection (Bonin-Font and Burguera, 2021).

Visual Inertial SLAM

The last recommendation is to investigate adaptations of VI-SLAM methods for underwater navigation. In recent years several VI-SLAM projects have been made publicly available, like the Kimera framework (Rosinol et al., 2020). By utilizing an IMU in addition to the visual information from a camera, VI-SLAM have proven to be more robust than V-SLAM, and less reliant on good visual features. There are, however, few studies of underwater VI-SLAM, so the robustness gains for underwater navigation are very much open research questions.

Bibliography

Akkaynak, D., Treibitz, T., 2018. A Revised Underwater Image Formation Model, in: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, IEEE, Salt Lake City, UT. pp. 6723–6732. URL: https://ieeexplore.ieee.org/document/8578801/, doi:10.1109/CVPR.2018.00703.

Akkaynak, D., Treibitz, T., 2019. Sea-Thru: A Method for Removing Water From Underwater Images, in: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, Long Beach, CA, USA. pp. 1682–1691. URL: https://ieeexplore.ieee.org/document/8954437/, doi:10.1109/CVPR.2019.00178.

Akkaynak, D., Treibitz, T., Shlesinger, T., Loya, Y., Tamir, R., Iluz, D., 2017. What is the Space of Attenuation Coefficients in Underwater Computer Vision?, in: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, Honolulu, HI. pp. 568–577. URL: http://ieeexplore.ieee.org/document/8099551/, doi:10.1109/CVPR.2017.68.

Aulinas, J., Carreras, M., Llado, X., Salvi, J., Garcia, R., Prados, R., Petillot, Y.R., 2011. Feature extraction for underwater visual SLAM, in: OCEANS 2011 IEEE - Spain, pp. 1–7. doi:10.1109/Oceans-Spain.2011.6003474.

Berman, D., Treibitz, T., Avidan, S., 2016. Non-local Image Dehazing. undefined URL: /paper/Non-local-Image-Dehazing-Berman-Treibitz/7b2ca78221fc59b40c122e3b230b8f552e856d12.

Bonin-Font, F., Burguera, A.B., 2021. NetHALOC: A learned global image descriptor for loop closing in underwater visual SLAM. Expert Systems 38, e12635. URL: https://onlinelibrary.wiley.com/doi/abs/10.1111/exsy.12635, doi:https://doi.org/10.1111/exsy.12635. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/exsy.12635.

Bonin-Font, F., Carrasco, P.L.N., Burguera, A.B., Codina, G.O., 2014. LSH for loop closing detection in underwater visual SLAM, in: Proceedings of the 2014 IEEE Emerging

Technology and Factory Automation (ETFA), pp. 1–4. doi:10.1109/ETFA.2014.7005245. iSSN: 1946-0759.

Bouguet, J.Y., 2015. Camera Calibration Toolbox for Matlab. URL: http://www.vision.caltech.edu/bouguetj/calib_doc/.

Brekke, E., 2020. Fundamentals of Sensor Fusion: Target tracking, navigation and SLAM. 2 ed., Unpublished.

Brown, D.C., 1971. Close-Range Camera Calibration. URL: /paper/Close-Range-Camera-Calibration-Brown/1150007b62a3c7dac99c2c8f85c63bfab74891af.

Bruno, F., Bianco, G., Muzzupappa, M., Barone, S., Razionale, A.V., 2011. Experimentation of Structured Light and Stereo Vision for Underwater 3D Reconstruction. ISPRS Journal of Photogrammetry and Remote Sensing 66, 508–518. URL: http://www.sciencedirect.com/science/article/pii/S0924271611000414, doi:10.1016/j.isprsjprs.2011.02.009.

Burguera Burguera, A., Bonin-Font, F., 2019. A Trajectory-Based Approach to Multi-Session Underwater Visual SLAM Using Global Image Signatures. Journal of Marine Science and Engineering 7, 278. URL: https://www.mdpi.com/2077-1312/7/8/278, doi:10.3390/jmse7080278. number: 8 Publisher: Multidisciplinary Digital Publishing Institute.

Cadena, C., Carlone, L., Carrillo, H., Latif, Y., Scaramuzza, D., Neira, J., Reid, I., Leonard, J.J., 2016. Past, Present, and Future of Simultaneous Localization and Mapping: Toward the Robust-Perception Age. IEEE Transactions on Robotics 32, 1309–1332. doi:10.1109/TRO.2016.2624754. conference Name: IEEE Transactions on Robotics.

Calonder, M., Lepetit, V., Strecha, C., Fua, P., 2010. BRIEF: binary robust independent elementary features, in: Proceedings of the 11th European conference on Computer vision: Part IV, Springer-Verlag, Berlin, Heidelberg. pp. 778–792.

Chen, D., Ardabilian, M., Chen, L., 2015. A Fast Trilateral Filter-Based Adaptive Support Weight Method for Stereo Matching. IEEE Transactions on Circuits and Systems for Video Technology 25, 730–743. doi:10.1109/TCSVT.2014.2361422. conference Name: IEEE Transactions on Circuits and Systems for Video Technology.

Chen, X., Zhang, P., Quan, L., Yi, C., Lu, C., 2021. Underwater Image Enhancement based on Deep Learning and Image Formation Model. arXiv:2101.00991 [eess] URL: http://arxiv.org/abs/2101.00991. arXiv: 2101.00991.

Colodro-Conde, C., Toledo-Moreo, F.J., Toledo-Moreo, R., Martínez-Álvarez, J.J., Garrigós Guerrero, J., Ferrández-Vicente, J.M., 2014. Evaluation of stereo correspondence algorithms and their implementation on FPGA. Journal of Systems Architecture 60, 22–31. URL: https://www.sciencedirect.com/science/article/pii/S1383762113002610, doi:10.1016/j.sysarc.2013.11.006.

Conrady, A.E., 1919. Decentred Lens-Systems. Monthly Notices of the Royal Astronomical Society 79, 384–390. URL: https://academic.oup.com/mnras/article/79/5/384/1078771, doi:10.1093/mnras/79.5.384. publisher: Oxford Academic.

Duggal, S., Wang, S., Ma, W.C., Hu, R., Urtasun, R., 2019. DeepPruner: Learning Efficient Stereo Matching via Differentiable PatchMatch, in: 2019 IEEE/CVF International Conference on Computer Vision (ICCV), pp. 4383–4392. doi:10.1109/ICCV.2019.00448. iSSN: 2380-7504.

Dukan, F., 2014. ROV Motion Control Systems. Ph.D. thesis. Norwegian University of Science and Technology. Trondheim, Norway. URL: /paper/ROV-Motion-Control-Systems-Dukan/a1f5c4698a599a5ccff6252edb125be16946240b.

Ferrera, M., Creuze, V., Moras, J., Trouvé-Peloux, P., 2019. AQUALOC: An underwater dataset for visual–inertial–pressure localization. The International Journal of Robotics Research 38, 1549–1559. URL: https://doi.org/10.1177/0278364919883346, doi:10.1177/0278364919883346. publisher: SAGE Publications Ltd STM.

Fischler, M.A., Bolles, R.C., 1981. Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. Technical Report. SRI INTERNATIONAL MENLO PARK CA ARTIFICIAL INTELLIGENCE CENTER. URL: https://apps.dtic.mil/docs/citations/ADA637836.

Fornasier, A., Scheiber, M., Hardt-Stremayr, A., Jung, R., Weiss, S., 2021. VINSEval: Evaluation Framework for Unified Testing of Consistency and Robustness of Visual-Inertial Navigation System Algorithms.

Furuhashi, S., 2019. MessagePack: It's like JSON. but fast and small. URL: https://msgpack.org/.

Förstner, W., Wrobel, B.P., 2016. Photogrammetric Computer Vision: Statistics, Geometry, Orientation and Reconstruction. Geometry and Computing, Springer International Publishing. URL: https://www.springer.com/gp/book/9783319115498, doi:10.1007/978-3-319-11550-4.

Galvez-López, D., Tardos, J.D., 2012. Bags of Binary Words for Fast Place Recognition in Image Sequences. IEEE Transactions on Robotics 28, 1188–1197. doi:10.1109/TRO.2012.2197158. conference Name: IEEE Transactions on Robotics.

Gavaskar, R.G., Chaudhury, K.N., 2019. Fast Adaptive Bilateral Filtering. IEEE Transactions on Image Processing 28, 779–790. doi:10.1109/TIP.2018.2871597. conference Name: IEEE Transactions on Image Processing.

Grisetti, G., Kümmerle, R., Stachniss, C., Burgard, W., 2010. A Tutorial on Graph-Based SLAM. IEEE Transactions on Intelligent Transportation Systems Magazine 2, 31–43. doi:10.1109/MITS.2010.939925.

Harris, C.G., Stephens, M., 1988. A Combined Corner and Edge Detector, in: Alvey Vision Conference, pp. 23.1–23.6. doi:10.5244/C.2.23.

Hartley, R., 1997. In defense of the eight-point algorithm. IEEE Transactions on Pattern Analysis and Machine Intelligence 19, 580–593. doi:10.1109/34.601246. conference Name: IEEE Transactions on Pattern Analysis and Machine Intelligence.

He, K., Sun, J., Tang, X., 2009. Single image haze removal using dark channel prior, in: 2009 IEEE Conference on Computer Vision and Pattern Recognition, pp. 1956–1963. doi:10.1109/CVPR.2009.5206515. iSSN: 1063-6919.

He, K., Sun, J., Tang, X., 2013. Guided Image Filtering. IEEE Transactions on Pattern Analysis and Machine Intelligence 35, 1397–1409. doi:10.1109/TPAMI.2012.213. conference Name: IEEE Transactions on Pattern Analysis and Machine Intelligence.

He, K., Zhang, X., Ren, S., Sun, J., 2016. Deep Residual Learning for Image Recognition, in: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 770–778. doi:10.1109/CVPR.2016.90. iSSN: 1063-6919.

Hecht, E., 2017. Optics. 5th, global ed., Pearson, Boston, Massachusetts.

Hummel, R., 1977. Image enhancement by histogram transformation. Computer Graphics and Image Processing 6, 184–195. URL: https://www.sciencedirect.com/science/article/pii/S0146664X77800117, doi:10.1016/S0146-664X(77)80011-7.

Ilg, E., Saikia, T., Keuper, M., Brox, T., 2018. Occlusions, Motion and Depth Boundaries with a Generic Network for Disparity, Optical Flow or Scene Flow Estimation. arXiv:1808.01838 [cs] URL: http://arxiv.org/abs/1808.01838. arXiv: 1808.01838.

Jerlov, N.G., 1968. Optical Oceanography. Elsevier. Google-Books-ID: k9EjXyVJH0UC.

Jian, M., Liu, X., Luo, H., Lu, X., Yu, H., Dong, J., 2021. Underwater image processing and analysis: A review. Signal processing. Image communication 91. doi:10.1016/j.image.2020.116088. publisher: Elsevier BV.

Jordt, A., 2014. Underwater 3D Reconstruction Based on Physical Models for Refraction and Underwater Light Propagation. Ph.D. thesis. University of Kiel. Kiel, Germany. URL: https://macau.uni-kiel.de/receive/diss_mods_00014162. accepted: 2013-11-12.

Jordt-Sedlazeck, A., Koch, R., 2013. Refractive Structure-from-Motion on Underwater Images, in: 2013 IEEE International Conference on Computer Vision, pp. 57–64. doi:10.1109/ICCV.2013.14. iSSN: 2380-7504.

Kartverket, 2020. Sjøkart - Dybdedata - Kartkatalogen. URL: https://kartkatalog.geonorge.no/metadata/sjoekart-dybdedata/2751aacf-5472-4850-a208-3532a51c529a#help-info.

Kim, A., Eustice, R.M., 2013. Real-Time Visual SLAM for Autonomous Underwater Hull Inspection Using Visual Saliency. IEEE Transactions on Robotics 29, 719–733. doi:10.1109/TRO.2012.2235699. conference Name: IEEE Transactions on Robotics.

King, B.M., 2020. Postphenomenology and Deep-Water Archaeology. Unpublished. Norwegian University of Science and Technology. Trondheim, Norway.

Kokhanovsky, A., 2004. Optical properties of terrestrial clouds. Earth-Science Reviews 64, 189–241. URL: https://www.sciencedirect.com/science/article/pii/S0012825203000424, doi:10.1016/S0012-8252(03)00042-4.

Kongsberg, 2005. HiPAP System Product Description. URL: https://www.oceanscan.net/gallery/PDFs/164268ae_HiPAP_500_and_350_product_description.pdf.

Kornprobst, P., Tumblin, J., Durand, F., 2009. Bilateral Filtering: Theory and Applications. undefined URL: /paper/Bilateral-Filtering%3A-Theory-and-Applications-Kornprobst-Tumblin/90ee89f9d9d669744fbea94ac8517afbc5825f87.

Kümmerle, R., Grisetti, G., Strasdat, H., Konolige, K., Burgard, W., 2011. G2o: A general framework for graph optimization, in: 2011 IEEE International Conference on Robotics and Automation, pp. 3607–3613. doi:10.1109/ICRA.2011.5979949. iSSN: 1050-4729.

Laga, H., Jospin, L.V., Boussaid, F., Bennamoun, M., 2020. A Survey on Deep Learning Techniques for Stereo-based Depth Estimation. IEEE Transactions on Pattern Analysis and Machine Intelligence , 1–1doi:10.1109/TPAMI.2020.3032602. conference Name: IEEE Transactions on Pattern Analysis and Machine Intelligence.

Larsen, M.K., 2020a. Georeferencing of Underwater Hyperspectral Scan Lines Using Stereo-Based Visual Simultaneous Localization and Mapping. Project Thesis. Norwegian University of Science and Technology. Trondheim, Norway.

Larsen, M.K., 2020b. markvilar/Sennet. URL: https://github.com/markvilar/Sennet.

Levenberg, K., 1944. A method for the solution of certain non-linear problems in least squares. Quarterly of Applied Mathematics 2, 164–168. URL: https://www.ams.org/qam/1944-02-02/S0033-569X-1944-10666-0/, doi:10.1090/qam/10666.

Li, R., Tao, C., Curran, T.A., Smith, R.G., 1997. Digital Underwater Photogrammetric System for Large Scale Underwater Spatial Information Acquisition. Marine Geodesy 20, 163–173. URL: https://doi.org/10.1080/01490419709388103, doi:10.1080/01490419709388103. publisher: Taylor & Francis _eprint: https://doi.org/10.1080/01490419709388103.

Liu, H., Chen, M., Zhang, G., Bao, H., Bao, Y., 2018. ICE-BA: Incremental, Consistent and Efficient Bundle Adjustment for Visual-Inertial SLAM, in: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 1974–1982. doi:10.1109/CVPR.2018.00211. iSSN: 2575-7075.

Longuet-Higgins, H.C., 1981. A computer algorithm for reconstructing a scene from two projections. Nature 293, 133–135. URL: https://www.nature.com/articles/293133a0, doi:10.1038/293133a0. number: 5828 Publisher: Nature Publishing Group.

Ma, Z., He, K., Wei, Y., Sun, J., Wu, E., 2013. Constant Time Weighted Median Filtering for Stereo Matching and Beyond, in: 2013 IEEE International Conference on Computer Vision, pp. 49–56. doi:10.1109/ICCV.2013.13. iSSN: 2380-7504.

Milford, M.J., Wyeth, G.F., 2012. SeqSLAM: Visual route-based navigation for sunny summer days and stormy winter nights, in: 2012 IEEE International Conference on Robotics and Automation, pp. 1643–1649. doi:10.1109/ICRA.2012.6224623. iSSN: 1050-4729.

Mobley, C., 1994. Light and Water: Radiative Transfer in Natural Waters. Academic Press.

Mur-Artal, R., Montiel, J.M.M., Tardós, J.D., 2015. ORB-SLAM: A Versatile and Accurate Monocular SLAM System. IEEE Transactions on Robotics 31, 1147–1163. doi:10.1109/TRO.2015.2463671. conference Name: IEEE Transactions on Robotics.

Mur-Artal, R., Tardós, J.D., 2017. ORB-SLAM2: An Open-Source SLAM System for Monocular, Stereo, and RGB-D Cameras. IEEE Transactions on Robotics 33, 1255–1262. doi:10.1109/TRO.2017.2705103. conference Name: IEEE Transactions on Robotics.

Muñoz-Salinas, R., Medina-Carnicer, R., 2020. UcoSLAM: Simultaneous localization and mapping by fusion of keypoints and squared planar markers. Pattern Recognition 101, 107193. URL: http://www.sciencedirect.com/science/article/pii/S0031320319304923, doi:10.1016/j.patcog.2019.107193.

Nister, D., 2004. An efficient solution to the five-point relative pose problem. IEEE Transactions on Pattern Analysis and Machine Intelligence 26, 756–770. doi:10.1109/TPAMI.2004.17. conference Name: IEEE Transactions on Pattern Analysis and Machine Intelligence.

Nornes, S.M., 2018. Guidance and Control of Marine Robotics for Ocean Mapping and Monitoring. NTNU. URL: https://ntnuopen.ntnu.no/ntnu-xmlui/handle/11250/2560596. accepted: 2018-09-04T07:45:37Z ISSN: 1503-8181.

NTNU, 2006. R/V Gunnerus - Research vessel - NTNU. URL: https://www.ntnu.edu/gunnerus/gunnerus.

Oppenheim, A.V., 1983. Signals and systems. Prentice-Hall signal processing series, Prentice-Hall, Englewood Cliffs, N.J.

Palachy, S., 2021. shaypal5/ssdts_matching. URL: https://github.com/shaypal5/ssdts_matching. original-date: 2017-07-13T15:58:52Z.

Paroscientific, 2005. Digiquartz Submersible Depth Sensors. URL: https://seatronics-group.com/wp-content/uploads/2020/03/Paroscientific%20Digiquartz%20Depth%20Sensors%20-%20Data%20Sheet.pdf.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Köpf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., Chintala, S., 2019. PyTorch: An Imperative Style, High-Performance Deep Learning Library. arXiv:1912.01703 [cs, stat] URL: http://arxiv.org/abs/1912.01703. arXiv: 1912.01703.

Rosinol, A., Abate, M., Chang, Y., Carlone, L., 2020. Kimera: an Open-Source Library for Real-Time Metric-Semantic Localization and Mapping. arXiv:1910.02490 [cs] URL: http://arxiv.org/abs/1910.02490. arXiv: 1910.02490.

Rosten, E., Porter, R., Drummond, T., 2010. Faster and Better: A Machine Learning Approach to Corner Detection. IEEE Transactions on Pattern Analysis and Machine Intelligence 32, 105–119. doi:10.1109/TPAMI.2008.275. conference Name: IEEE Transactions on Pattern Analysis and Machine Intelligence.

Rublee, E., Rabaud, V., Konolige, K., Bradski, G., 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. iSSN: 2380-7504.

Schechner, Y., Karpel, N., 2004. Clear underwater vision. undefined URL: /paper/Clear-underwater-vision-Schechner-Karpel/dec3de4ae1cb82c75189eb98b5ebb9a1a683f334.

Schlegel, D., Colosi, M., Grisetti, G., 2018. ProSLAM: Graph SLAM from a Programmer's Perspective. 2018 IEEE International Conference on Robotics and Automation (ICRA) , 3833–3840URL: http://arxiv.org/abs/1709.04377, doi:10.1109/ICRA.2018.8461180. arXiv: 1709.04377.

Sedlazeck, A., Koch, R., 2012. Perspective and Non-perspective Camera Models in Underwater Imaging – Overview and Error Analysis, in: Dellaert, F., Frahm, J.M., Pollefeys, M., Leal-Taixé, L., Rosenhahn, B. (Eds.), Outdoor and Large-Scale Real-World Scene Analysis, Springer, Berlin, Heidelberg. pp. 212–242. doi:10.1007/978-3-642-34091-8_10.

Shortis, M., 2015. Calibration Techniques for Accurate Measurements by Underwater Camera Systems. Sensors 15, 30810–30826. URL: https://www.mdpi.com/1424-8220/15/12/29831, doi:10.3390/s151229831. number: 12 Publisher: Multidisciplinary Digital Publishing Institute.

Sivic, Zisserman, 2003. Video Google: a text retrieval approach to object matching in videos, in: Proceedings Ninth IEEE International Conference on Computer Vision, pp. 1470–1477 vol.2. doi:10.1109/ICCV.2003.1238663.

Sola, J., 2007. Towards Visual Localization, Mapping and Moving Objects Tracking by a Mobile Robot: a Geometric and Probabilistic Approach. PhD Thesis. Université de Toulouse. Toulouse. URL: http://ethesis.inp-toulouse.fr/archive/00000528/.

Solonenko, M.G., Mobley, C., 2015. Inherent optical properties of Jerlov water types. undefined URL: /paper/Inherent-optical-properties-of-Jerlov-water-types.-Solonenko-Mobley/608daad56084d6e1fdb7dd2c0f7d941bde7e10dd.

Solà, J., 2017. Quaternion kinematics for the error-state Kalman filter. arXiv:1711.02508 [cs] URL: http://arxiv.org/abs/1711.02508. arXiv: 1711.02508.

Stereolabs, 2018. ZED Camera and SDK Overview. URL: https://cdn.stereolabs.com/assets/datasheets/zed-camera-datasheet.pdf.

Stereolabs, 2021. Stereolabs - Capture the World in 3D. URL: https://www.stereolabs.com/.

Sturm, J., Engelhard, N., Endres, F., Burgard, W., Cremers, D., 2012. A benchmark for the evaluation of RGB-D SLAM systems, in: 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 573–580. doi:10.1109/IROS.2012.6385773. iSSN: 2153-0866.

Sumikura, S., Shibuya, M., Sakurada, K., 2019. OpenVSLAM: A Versatile Visual SLAM Framework. Proceedings of the 27th ACM International Conference on Multimedia , 2292–2295URL: http://arxiv.org/abs/1910.01122, doi:10.1145/3343031.3350539. arXiv: 1910.01122.

Teledyne, 2013. Workhorse Navigator Doppler Velocity Log. URL: https://ocean-innovations.net/OceanInnovationsNEW/Teledyne%20RD%20Instruments/navigator_datasheet_lr.pdf.

Telem, G., Filin, S., 2010. Photogrammetric Modeling of Underwater Environments. ISPRS Journal of Photogrammetry and Remote Sensing 65, 433–444. URL: https://www.sciencedirect.com/science/article/pii/S0924271610000444, doi:10.1016/j.isprsjprs.2010.05.004.

Tomasi, C., Manduchi, R., 1998. Bilateral filtering for gray and color images. undefined URL: /paper/Bilateral-filtering-for-gray-and-color-images-Tomasi-Manduchi/bfeaf424a2ea6ca4702d545c6e959e2caeb68e9b.

Tonioni, A., Poggi, M., Mattoccia, S., Di Stefano, L., 2017. Unsupervised Adaptation for Deep Stereo, in: 2017 IEEE International Conference on Computer Vision (ICCV), pp. 1614–1622. doi:10.1109/ICCV.2017.178. iSSN: 2380-7504.

Trawny, N., Roumeliotis, S., 2005. Indirect Kalman Filter for 3 D Attitude Estimation. URL: /paper/Indirect-Kalman-Filter-for-3-D-Attitude-Estimation-Trawny-Roumeliotis/2c8e95bc331024105cbde6f6918cda8493f263c8.

Triggs, B., McLauchlan, P.F., Hartley, R.I., Fitzgibbon, A.W., 2000. Bundle Adjustment — A Modern Synthesis, in: Triggs, B., Zisserman, A., Szeliski, R. (Eds.), Vision Algorithms: Theory and Practice, Springer, Berlin, Heidelberg. pp. 298–372. doi:10.1007/3-540-44480-7_21.

Umeyama, S., 1991. Least-squares estimation of transformation parameters between two point patterns. IEEE Transactions on Pattern Analysis and Machine Intelligence 13, 376–380. doi:10.1109/34.88573. conference Name: IEEE Transactions on Pattern Analysis and Machine Intelligence.

Wang, Z., Bovik, A., Sheikh, H., Simoncelli, E., 2004. Image quality assessment: from error visibility to structural similarity. IEEE Transactions on Image Processing 13, 600–612. doi:10.1109/TIP.2003.819861. conference Name: IEEE Transactions on Image Processing.

Watson, J., Zielinski, O., 2013. Subsea Optics and Imaging. Woodhead Publishing Series in Electronic and Optical Materials. 1st ed., Woodhead Publishing. OCLC: 865332549.

Werner, M., Stabernack, B., Riechert, C., 2014. Hardware implementation of a full HD real-time disparity estimation algorithm. IEEE Transactions on Consumer Electronics 60, 66–73. doi:10.1109/TCE.2014.6780927. conference Name: IEEE Transactions on Consumer Electronics.

Wozniak, B., Dera, J., 2007. Light Absorption in Sea Water. volume 33. doi:10.1007/978-0-387-49560-6. journal Abbreviation: Light Absorption in Sea Water Publication Title: Light Absorption in Sea Water.

Xiaorui, Q., Atsushi, Y., Hajime, A., 2019. Underwater Structure from Motion for Cameras Under Refractive Surfaces. Journal of Robotics and Mechatronics 31, 603–611. URL: https://search.proquest.com/docview/2465809566/abstract/35EFBA2D527A46C7PQ/1, doi:http://dx.doi.org/10.20965/jrm.2019.p0603. num Pages: 603-611 Place: Tokyo, Japan Publisher: Fuji Technology Press Co. Ltd.

Xsens, 2018. MTi 100-series. URL: https://cdn2.hubspot.net/hubfs/3446270/Downloads/Leaflets/mti-100-series.pdf?__hstc=81749512.5a39540bd282e2ad97258c978a63a051.1618670100670.1618670100670.1622574643912.2&__hssc=81749512.2.1622574643912&__hsfp=2566586354&hsCtaTracking=ab734ab9-10ca-422d-9070-5dac538c0770%7Cada1539f-8ae0-447d-bda1-a2a6b61edeab.

Yang, G., Zhao, H., Shi, J., Deng, Z., Jia, J., 2018. SegStereo: Exploiting Semantic Information for Disparity Estimation. arXiv:1807.11699 [cs] URL: http://arxiv.org/abs/1807.11699. arXiv: 1807.11699.

Yin, Z., Darrell, T., Yu, F., 2019. Hierarchical Discrete Distribution Decomposition for Match Density Estimation, in: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 6037–6046. doi:10.1109/CVPR.2019.00620. iSSN: 2575-7075.

Zhang, Z., 2000. A flexible new technique for camera calibration. IEEE Transactions on Pattern Analysis and Machine Intelligence 22, 1330–1334. doi:10.1109/34.888718. conference Name: IEEE Transactions on Pattern Analysis and Machine Intelligence.

Zuiderveld, K., 1994. Contrast limited adaptive histogram equalization, in: Graphics gems IV. Academic Press Professional, Inc., USA, pp. 474–485.

Łuczyński, T., Pfingsthorn, M., Birk, A., 2017. The Pinax-model for Accurate and Efficient Refraction Correction of Underwater Cameras in Flat-Pane Housings. Ocean Engineering 133, 9–22. URL: http://www.sciencedirect.com/science/article/pii/S0029801817300434, doi:10.1016/j.oceaneng.2017.01.029.

Appendices

Appendix A

Technical Information

Technical Specifications of Sensors and Sensor Systems

Parameter	Specification	Unit
Angular accuracy, 20 dB S/N	0.06	deg
Angular accuracy, 10 dB S/N	0.10	deg
Angular accuracy, 0 dB S/N	0.30	deg
Range accuracy, 20 dB S/N	0.10	m
Range accuracy, 10 dB S/N	0.15	m
Range accuracy, 0 dB S/N	0.20	m
Coverage	± 100	deg

Table A.1: Technical specifications for the Kongsberg HiPAP 500 system. Courtesy: Kongsberg (2005)

Parameter	Specification	Unit
Frequency	415	Hz
Initial bias error	0.2	deg/s
Bias stability	10.0	deg/h
Noise density	0.01	deg/s$\sqrt{\text{Hz}}$

Table A.2: Technical specifications for the XSens MTi-100 IMU gyroscope. Courtesy: Xsens (2018)

Parameter	Specification	Unit
Maximum ping frequency	7	Hz
Std. dev. at 1 m/s	0.3	cm/s
Std. dev. at 3 m/s	0.5	cm/s
Std. dev. at 5 m/s	0.7	cm/s
Long-term accuracy	0.2	%
Long-term std. dev.	0.1	cm/s
Minimum altitude	0.5	m
Maximum altitude	25	m

Table A.3: Technical specifications for the Teledyne RDI Workhorse Navigator DVL. Courtesy: Teledyne (2013)

Parameter	Specification	Unit
Pressure signal frequency	37-42	kHz
Accuracy	0.01	%

Table A.4: Technical specifications for the Paroscientific Digiquartz pressure sensor. Courtesy: Paroscientific (2005)

Appendix B

Mathematical Preliminaries

Quaternions

Quaternion Definitions and Properties

Quaternion Fundamentals

For the background material on quaternion arithmetic and unit quaternion rotation representation, the reader is referred to Solà (2017).

A quaternion $\mathbf{q}$ is a four-dimensional number consisting of a real scalar part η, and a hyper-imaginary part $\boldsymbol{\epsilon}$. The quaternion $\mathbf{q}$ can be expressed in terms of its real part and the hyper-imaginary units, $\mathbf{i}$, $\mathbf{j}$, and $\mathbf{k}$, as

$$\mathbf{q} = \begin{bmatrix} \eta \\ \boldsymbol{\epsilon} \end{bmatrix} = \begin{bmatrix} \eta \\ \epsilon_1 \\ \epsilon_2 \\ \epsilon_3 \end{bmatrix} = \eta + \epsilon_1 \mathbf{i} + \epsilon_2 \mathbf{j} + \epsilon_3 \mathbf{k}, \tag{B.1}$$

where the hyper-imaginary units are connected through Sir William Rowan Hamilton's famous fundamental property of quaternions,

$$\mathbf{i}^2 = \mathbf{j}^2 = \mathbf{k}^2 = \mathbf{ijk} = -1. \tag{B.2}$$

Quaternion Sums

Addition of quaternions is straight-forward, and defined as

$$\mathbf{q}_a + \mathbf{q}_b = \begin{bmatrix} \eta_a \\ \boldsymbol{\epsilon}_a \end{bmatrix} + \begin{bmatrix} \eta_b \\ \boldsymbol{\epsilon}_b \end{bmatrix} = \begin{bmatrix} \eta_a + \eta_b \\ \boldsymbol{\epsilon}_a + \boldsymbol{\epsilon}_b \end{bmatrix}. \tag{B.3}$$

Quaternion Product

The quaternion product is were the arithmetics become more complex. The product of two quaternions $\mathbf{q}_a$ and $\mathbf{q}_b$ is defined as

$$\mathbf{q}_a \otimes \mathbf{q}_b = \begin{bmatrix} \eta_a \eta_b + \boldsymbol{\epsilon}_a^\top \boldsymbol{\epsilon}_b \\ \eta_b \boldsymbol{\epsilon}_1 + \eta_a \boldsymbol{\epsilon}_b + \boldsymbol{\epsilon}_a \times \boldsymbol{\epsilon}_b \end{bmatrix}. \tag{B.4}$$

From eq. B.4 one can see that the quaternion product is non-commutative, i.e.

$$\mathbf{q}_a \otimes \mathbf{q}_b \neq \mathbf{q}_b \otimes \mathbf{q}_a, \tag{B.5}$$

due to non-commutativity of the vector cross product of the hyper-imaginary parts. The quaternion product is, however, associative

$$(\mathbf{q}_a \otimes \mathbf{q}_b) \otimes \mathbf{q}_c = \mathbf{q}_a \otimes (\mathbf{q}_b \otimes \mathbf{q}_c), \tag{B.6}$$

and distributive

$$\mathbf{q}_a \otimes (\mathbf{q}_b + \mathbf{q}_c) = \mathbf{q}_a \otimes \mathbf{q}_b + \mathbf{q}_a \otimes \mathbf{q}_c. \tag{B.7}$$

Quaternion Identity

Utilizing the quaternion product definition in eq. B.4, one can see that the identity quaternion $\mathbf{q}_1$, with the following property

$$\mathbf{q} \otimes \mathbf{q}_1 = \mathbf{q}_1 \otimes \mathbf{q} = \mathbf{q}, \tag{B.8}$$

is defined as

$$\mathbf{q}_1 = \begin{bmatrix} 1 \\ \mathbf{0} \end{bmatrix}. \tag{B.9}$$

Quaternion Conjugate

Similarly to imaginary numbers, the conjugate of a quaternion $\mathbf{q}$ is defined as

$$\mathbf{q}^* = \begin{bmatrix} \eta \\ -\boldsymbol{\epsilon} \end{bmatrix}, \tag{B.10}$$

with the following properties

$$\mathbf{q} \otimes \mathbf{q}^* = \mathbf{q}^* \otimes \mathbf{q} = \eta^2 + \epsilon_1^2 + \epsilon_2^2 + \epsilon_3^2 = \begin{bmatrix} \eta^2 + \epsilon_1^2 + \epsilon_2^2 + \epsilon_3^2 \\ \mathbf{0} \end{bmatrix} \tag{B.11}$$

and

$$(\mathbf{q}_a \otimes \mathbf{q}_b)^* = \mathbf{q}_b^* \otimes \mathbf{q}_a^*. \tag{B.12}$$

Quaternion Norm

The norm of a quaternion is defined as

$$\|\mathbf{q}\| = \sqrt{\mathbf{q} \otimes \mathbf{q}^*} = \sqrt{\eta^2 + \|\boldsymbol{\epsilon}\|^2} = \sqrt{\eta^2 + \epsilon_1^2 + \epsilon_2^2 + \epsilon_3^2}. \tag{B.13}$$

$$\|\mathbf{q}_a \otimes \mathbf{q}_b\| = \|\mathbf{q}_a\| \, \|\mathbf{q}_b\| \tag{B.14}$$

Quaternion Inverse

Now that the identity quaternion $\mathbf{q}_1$ has been established, the property which would define the inverse of a quaternion $\mathbf{q}$, would be that the product of the quaternion and its inverse is equal to the identity quaternion, i.e.

$$\mathbf{q} \otimes \mathbf{q}^{-1} = \mathbf{q}^{-1} \otimes \mathbf{q} = \mathbf{q}_1. \tag{B.15}$$

This relation in addition to the definition of the quaternion product leads to the following expression of the inverse quaternion

$$\mathbf{q}^{-1} = \frac{\mathbf{q}^*}{\|\mathbf{q}\|^2}. \tag{B.16}$$

Quaternion Rotation Representation

Unit quaternions are useful for attitude representation, due to the afore-mentioned properties, the fact that multiple consecutive rotations can be expressed as a single quaternion product, and that they do not suffer from gimbal locking. A unit quaternion can rotate a three-dimensional vector $\mathbf{v}$ to $\mathbf{v}'$ as

$$\begin{bmatrix} 0 \\ \mathbf{v}' \end{bmatrix} = \mathbf{q} \otimes \begin{bmatrix} 0 \\ \mathbf{v} \end{bmatrix} \otimes \mathbf{q}^*, \quad \|\mathbf{q}\| = 1. \tag{B.17}$$

Since quaternion are challenging to compose directly, a common approach is to define them in terms of an axis and an angle. Using this approach, a unit quaternion $\mathbf{q}$ can be composed by a rotation α around an axis, defined by the unit vector $\mathbf{d}$, as

$$\mathbf{q} = \text{Axis-Angle}(\mathbf{d}, \alpha) = \begin{bmatrix} \eta \\ \boldsymbol{\epsilon} \end{bmatrix} = \begin{bmatrix} \cos(\frac{\alpha}{2}) \\ \mathbf{d} \cdot \sin(\frac{\alpha}{2}) \end{bmatrix}, \quad \|\mathbf{d}\| = 1. \tag{B.18}$$

Combined rotations can also be expressed with unit quaternions in a straight forward fashion, depending on the frame of reference. Consider two consecutive rotations applied to an object, expressed by the two unit quaternions $\mathbf{q}_a$ and $\mathbf{q}_b$, respectively. In an absolute reference frame, the resulting rotation is expressed in terms of the unit quaternion $\mathbf{q}_c$, as

$$\mathbf{q}_c = \mathbf{q}_b \otimes \mathbf{q}_a. \tag{B.19}$$

In the frame of reference of the rotated object the same resulting rotation $\mathbf{q}_c$ is expressed by reversing the order of the two constituent rotations in the quaternion product, i.e.

$$\mathbf{q}_c = \mathbf{q}_a \otimes \mathbf{q}_b. \tag{B.20}$$

Appendix C

Data and Source Code

Ekne Wreck Site Video Sequences

Sequence	Duration	YouTube Link
"Ekne Wreck Site 01"	14:18	Link
"Ekne Wreck Site 02"	4:13	Link
"Ekne Wreck Site 03"	4:05	Link
"Ekne Wreck Site 04"	2:16	Link
"Ekne Wreck Site 05"	4:18	Link
"Ekne Wreck Site 06"	3:13	Link
"Ekne Wreck Site 07"	0:58	Link
"Ekne Wreck Site 08"	1:07	Link

Source Code Repositories

`https://github.com/markvilar/Sennet`

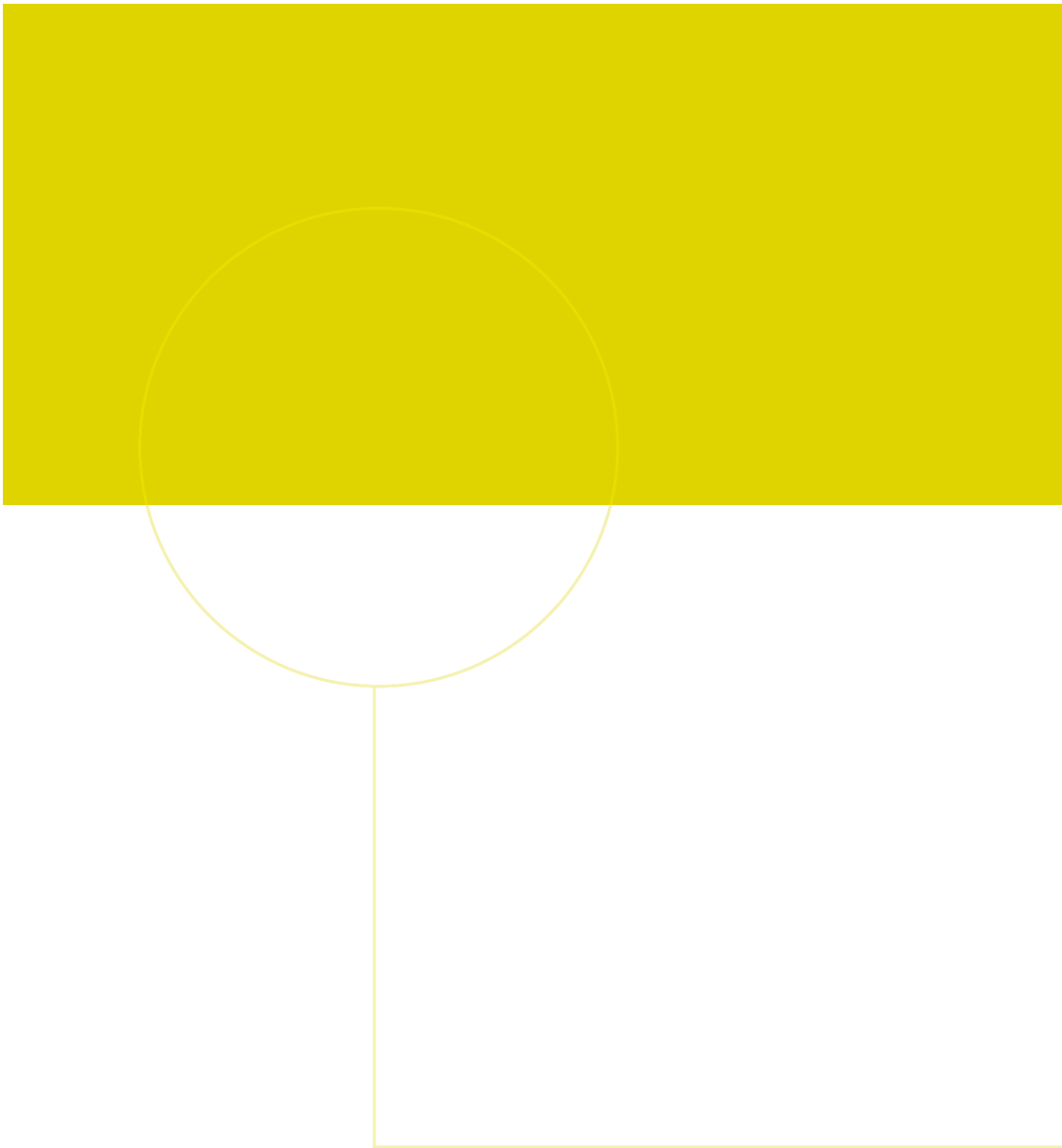
`https://github.com/markvilar/Sennet-ZED`

`https://github.com/markvilar/Cardinal`

`https://github.com/markvilar/Focal`

`https://github.com/markvilar/Trajectory`

https://github.com/markvilar/openvslam



NTNU

Norwegian University of
Science and Technology