Visual localization for deep-sea long-term monitoring

Clémentin Boittiaux

ÉCOLE DOCTORALE 548 — MER ET SCIENCES

Ifremer

COSMER & LIS

THÈSE présentée par :

Clémentin Boittiaux

soutenue le : 14 décembre 2023

pour obtenir le grade de Docteur en Automatique, signal, productique, robotique

Visual localization for deep-sea long-term monitoring

THÈSE dirigée par :

Professeur HUGEL Vincent	Directeur de thèse, Université de Toulon

JURY :

Docteure BERGER Marie-Odile	Rapportrice, INRIA Nancy Grand Est
Professeur LEPETIT Vincent	Rapporteur, École des Ponts ParisTech
Professeur DEMONCEAUX Cédric	Examinateur, Université de Bourgogne
Docteur SATTLER Torsten	Examinateur, Czech Technical University
Professeur HUGEL Vincent	Directeur de thèse, Université de Toulon
Docteure DUNE Claire	Co-encadrante, Université de Toulon
Docteur ARNAUBEC Aurélien	Co-encadrant, Ifremer
Professeur MARXER Ricard	Co-encadrant, Université de Toulon

INVITÉ :

Docteur OPDERBECKE Jan	Directeur de l'unité SM, Ifremer

Acknowledgments

Je tiens à remercier tout d'abord Dr Marie-Odile Berger et Pr Vincent Lepetit pour avoir accepté de rapporter cette thèse. Je souhaiterais également remercier Pr Cédric Demonceaux et Dr Torsten Sattler pour avoir accepté de participer à mon jury de thèse en tant qu'examinateurs.

Je remercie mon directeur de thèse Vincent Hugel, pour m'avoir aiguillé dans mes travaux de recherche, ainsi que mes encadrants de thèse, Aurélien Arnaubec, Claire Dune et Ricard Marxer, pour m'avoir apporté le support nécessaire pendant ces trois années formatrices. Je remercie l'Ifremer pour avoir financé entièrement cette thèse.

Un merci très particulier à Maxime pour sa contribution majeure à mes travaux de thèse et ses conseils avisés et son humour à peu près subtil. Un merci de Cyrille pour moi pour l'avoir supporté tous les jours avec le sourire. Un grand merci à mes compagnons de thèse Juliette et Bilal pour avoir ensoleillé ce long séjour dans le sud.

Merci à l'unité SM et à tout le personnel d'Ifremer pour votre bonne humeur qui m'a motivée à réaliser cette thèse pendant trois années. Merci Jan, Lorenzo, Marie-Edith et Anne-Gaëlle. Merci aux laboratoires COSMER et LIS pour votre accueil.

Thank you again Dr Sattler and the computer vision and robotics teams at CIIRC CTU for their warm welcome.

Merci Pr Éric Marchand et l'équipe Rainbow de l'INRIA pour leur accueil et leurs conseils.

Thank you Dr Derya Akkaynak and Dr David Nakath for our short but colorful discussions.

Merci à ma famille, mes parents, mes grands-parents et mes amis qui ont pu suivre mes aventures ces trois dernières années et se montrer patient pour nos retrouvailles. Merci à Émilie d'avoir été à mes côtés pendant tout ce temps.

Enfin, merci à tous mes stagiaires, collègues, camarades doctorants et amis avec qui j'ai pu apprendre, voyager, snorkeler, surfer, grimper et tant d'autres.

Abstract

This thesis explores the challenge of localizing underwater vehicles within previously explored environments in long-term scenarios, where significant scene appearance changes may have occurred. Typically, underwater vehicle positioning relies on fusing measurements from acoustic and inertial sensors. While these sensors deliver precise relative pose estimations, their absolute position estimates exhibit notable biases, resulting in position offsets spanning tens of meters between different dives. This limitation impedes the practical use of autonomous underwater vehicles for tasks requiring high precision, like mapping a precise area of interest. In response, this thesis investigates the use of visual observations made by underwater vehicles to enhance absolute positioning accuracy. The underwater environment introduces unique sources of variability absent in terrestrial environments. Consequently, the first contribution of this thesis is a novel dataset designed for benchmarking long-term visual localization algorithms in deep-sea conditions. Another obstacle inherent to underwater images is that their suffer from low contrast and loss of colors because of light propagation in the water medium. To address this issue, the second contribution of this work introduces two underwater color restoration methods, specifically designed to mitigate these phenomena and recover clear images. Independent of the underwater environment, the third contribution of this thesis is a novel loss function tailored for camera pose regression within the context of deep learning applications. This is an important aspect to consider when training visual localization networks. Finally, this thesis concludes with a benchmark of several visual localization methods on the proposed dataset. The obtained results show that applying our underwater color restoration method improves visual localization performance. This work also identifies the major problem encountered by visual localization methods on the proposed underwater dataset, and presents an approach to improve the accuracy of visual localization techniques by making the most of a limited size dataset.

Résumé

Cette thèse explore le problème de la localisation de véhicules sous-marins dans des environnements déjà explorés. Elle s'inscrit dans le cadre de la surveillance des grands fonds à long terme. Ainsi, l'environnement visité peut avoir subi des changements significatifs entre plusieurs visites. Traditionnellement, la localisation de véhicules sous-marins repose sur la fusion de mesures provenant de capteurs acoustiques et inertiels. Alors que ces capteurs fournissent des estimations précises de pose relatives, leurs estimations de position absolue présentent des biais importants, entraînant des décalages de position de plusieurs dizaines de mètres entre différentes plongées. Cette limitation entrave considérablement l'utilisation des véhicules sous-marins autonomes pour des tâches exigeant un haut degré de précision, telles que la cartographie de zones d'intérêt spécifiques. En réponse, cette thèse explore l'utilisation des observations visuelles faites par les véhicules sous-marins pour obtenir une localisation absolue plus précise. Le milieu sous-marin introduit diverses sources de variabilité qui sont absentes dans le domaine terrestre. Par conséquent, la première contribution de cette thèse est la création d'un nouveau jeu de données spécialement conçu pour évaluer des algorithmes de localisation visuelle à long terme dans les conditions des grands fonds océaniques. De plus, un autre défi inhérent aux images sous-marines est leur faible contraste et la perte de couleurs dus à la propagation de la lumière dans l'eau. Pour remédier à ce problème, la deuxième contribution de cette thèse présente deux nouvelles méthodes de restauration des couleurs des images sous-marines spécifiquement conçues pour atténuer ces phénomènes et restituer des images claires. Indépendamment du milieu sous-marin, la troisième contribution de cette thèse est la proposition d'une nouvelle fonction de coût, conçue pour la régression de pose de caméra dans un contexte d'applications à l'apprentissage profond. Il s'agit d'un aspect important pour l'entraînement des réseaux de neurones dédiés à la localisation visuelle. Enfin, cette thèse se termine par une évaluation de plusieurs méthodes de localisation visuelle sur le nouveau jeu de données proposé. Les résultats obtenus montrent que l'application de notre technique de restauration des couleurs d'images sous-marines améliore sensiblement les performances de localisation visuelle. Ce travail identifie également le principal problème rencontré par les méthodes de localisation visuelle sur le jeu de données sous-marin proposé, et présente une approche visant à améliorer l'efficacité des techniques de localisation visuelle en exploitant au mieux un jeu de données de taille limitée.

Contents

List of Figures

List of Tables

Acronyms

AUV Autonomous Underwater Vehicle.

BoW Bag of Words.

CNN Convolutional Neural Network.

DoF Degrees of Freedom.

DVL Doppler Velocity Log.

GNN Graph Neural Networks.

GNSS Global Navigation Satellite System.

IMU Inertial Measurement Unit.

INS Inertial Navigation System.

ROV Remotely Operated Vehicle.

SLAM Simultaneous Localization and Mapping.

USBL Ultra Short Baseline.

VLAD Vector of Locally Aggregated Descriptors.

Glossary

Altitude Altitude and absolute altitude are referenced with respect to the Earth's surface, using the WGS 84 reference ellipsoid as a standardized model for the Earth's shape. The WGS 84 reference ellipsoid provides a precise and consistent framework for representing altitude, ensuring uniformity in global positioning measurements[1].

Global descriptor A vector characterizing an entire image.

ICP Iterative Closest Point. An iterative algorithm used for aligning two sets of 3D points by estimating rotation, translation and sometimes scale.

Local feature A representation of a 2D point in an image consisting of the pixel coordinates of that point and a local descriptor characterizing that point based on its surrounding context within the image.

NeRF Neural Radiance Fields. Neural networks encoding the radiance emitted from various positions and directions within a scene.

PnP Perspective-n-Point. Problem of estimating the pose of a camera given its calibration matrix and 2D-3D correspondences.

PSNR Peak Signal-to-Noise Ratio. A metric quantifying an image quality by comparing it to a reference image. It measures the ratio between the maximum value of the reference image and the mean square error between the two images. It is expressed in dB.

RANSAC RANdom SAmple Consensus. A robust statistical algorithm used to estimate model parameters from noisy data.

SfM Structure-from-Motion. An algorithm that estimates camera poses, intrinsics and a 3D point cloud of a scene given a set of images.

[1] `https://earth-info.nga.mil/index.php?dir=wgs84&action=wgs84`

SSIM Structural Similarity Index Measure. A metric quantifying an image perceived quality by comparing it to a reference image. In contrast with PSNR, it considers not only pixel values but also the structural and luminance information in the images.

UCIQE Underwater Color Image Quality Evaluation. A no-reference metric introduced by Yang and Sowmya (2015). It weights images properties such as sharpness and contrast to evaluate the quality of restored underwater images.

UIQM Underwater Image Quality Measure. Similar to UCIQE, it is a no-reference metric introduced by Panetta et al. (2016) weighting colorfulness, sharpness and contrast metrics to evaluate the quality of restored underwater images.

Chapter 1

Introduction

Contents

1.1 Context

The contemporary world faces a multitude of global challenges, such as climate change and environmental degradation. To overcome these challenges and pave the way for a better future, it is crucial to gain a deeper understanding of the world we inhabit. Yet, while it covers the majority of our planet's surface, the ocean remains the least explored and understood environment. Long-term monitoring of this realm is essential to understand the function of the ocean in climate change and improve the detection of geological hazards' early warnings to better prevent their impact (Ruhl et al., 2011).

To explore the ocean's depths, we rely heavily on underwater vehicles, which serve as our eyes and hands in this elusive domain. However, deep-sea exploration with Remotely Operated Vehicles (ROVs) is often prohibitively expensive

and time-consuming as it requires the ship to be mobilized for the whole duration of the dive. This necessitates the development of autonomous systems to efficiently carry out these exploratory tasks using less time and resources. Central to the success of Autonomous Underwater Vehicles (AUVs) is their ability to accurately determine their location within their environment, a process known as localization. However, due to the high attenuation and limited penetration of electromagnetic waves in water, Global Navigation Satellite Systems (GNSS) do not function in underwater environments. Moreover, while acoustic sensors have traditionally been employed for underwater localization, they are not always readily available and may lack the necessary precision for certain critical applications, such as mapping a precise area. In contrast, the visual observations captured by underwater vehicles offer a promising complement, potentially enabling a more accurate localization.

1.2 Underwater navigation

As previously mentioned, existing underwater localization methods may not be available or accurate enough to carry certain autonomous tasks. In order to understand why, this section presents a brief overview of the sensors used by Ifremer to localize underwater vehicles (Ferrera, 2019) and makes an overview of their limitations in the context of autonomous robots.

1.2.1 Sensors for underwater navigation

Several methods and sensing modalities are combined to obtain complementary cues for localization. Their availability, accuracy and precision varies significantly and may depend on the position of the vehicle during a dive.

Pressure sensor. The pressure sensor records the pressure experienced by the vehicle as it operates. In an underwater environment, the pressure is directly correlated with the depth of the water column above the vehicle. Consequently, by solely analyzing the measured pressure, it is possible to retrieve the immersion level of the vehicle. Subsequently, the vehicle's absolute altitude can then be estimated by taking into account the height of the tide.

Doppler Velocity Log. A Doppler Velocity Log (DVL) is an acoustic navigation device used to estimate velocity with respect to the seabed[1] (Figure 1.1). It operates by emitting four acoustic beams in different directions. It then utilizes the

[1] https://www.nortekgroup.com/knowledge-center/wiki/new-to-subsea-navigation

Doppler effect to analyze the frequency shift of the echoes reflected off the seabed and recover the vehicle's velocity. In the case where a DVL is not within range of the bottom, it may estimate the velocity relative to the surrounding water as an alternative — this is referred to as water track.

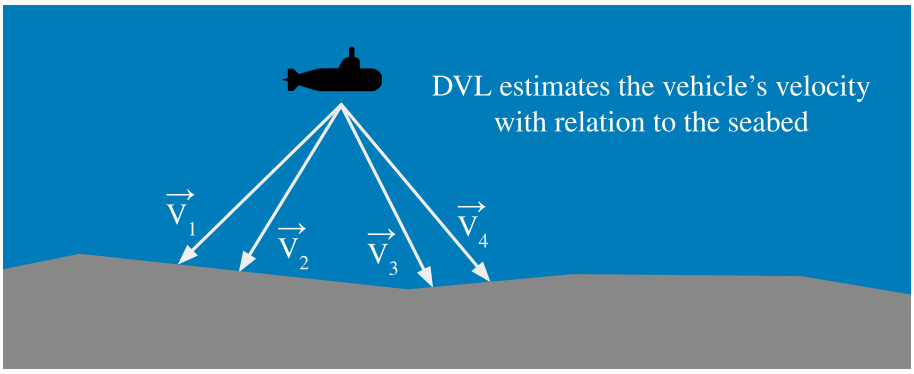


Figure 1.1: **Doppler Velocity Log.**

Inertial Navigation System. An Inertial Navigation System (INS) is a device embedding an Inertial Measurement Unit (IMU) and a computational unit (Figure 1.2). In the scope of this thesis, we will consider that IMUs consist of 3-axes accelerometers and 3-axes fiber-optic gyroscopes. The accelerometers provide information about the linear acceleration of the vehicle in the 3D space. The gyroscopes provide information about the angular velocity of the vehicle. Using its computational unit, the INS integrates the IMU's measurements to compute an estimate of the vehicle's position, orientation and speed. One downside is that this integration accumulates errors over time, leading to a drift in the vehicle's pose estimation. To alleviate this effect, the vehicle has to rely on other sensors, such as the DVL that can provide an estimate of its velocity.

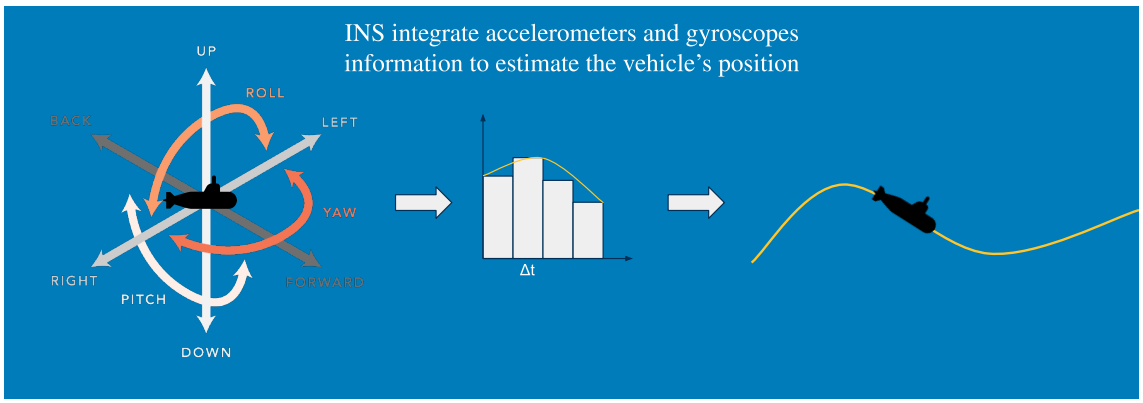


Figure 1.2: **Inertial Navigation System.**

Ultra Short Baseline. An Ultra-Short Baseline (USBL) acoustic positioning system is a technology used to determine the absolute position of an underwater vehicle (Figure 1.3). It operates by leveraging acoustic signals transmitted between an array of transceivers, often mounted on the ship, and a transponder attached to the underwater vehicle. It works based on the principle of measuring the time it takes for the acoustic signal to travel from the transceivers to the transponder and back. The transceivers emit a signal which propagates through the water and reaches the transponder. Upon receiving the signal, the transponder responds by sending a reply signal back to the transceivers. By precisely measuring the time of flight for the signals, the USBL system can calculate the range between the transceiver and each transponder. This range information, along with the known positions of the transceivers, is then used to triangulate and determine the position of the underwater vehicle.

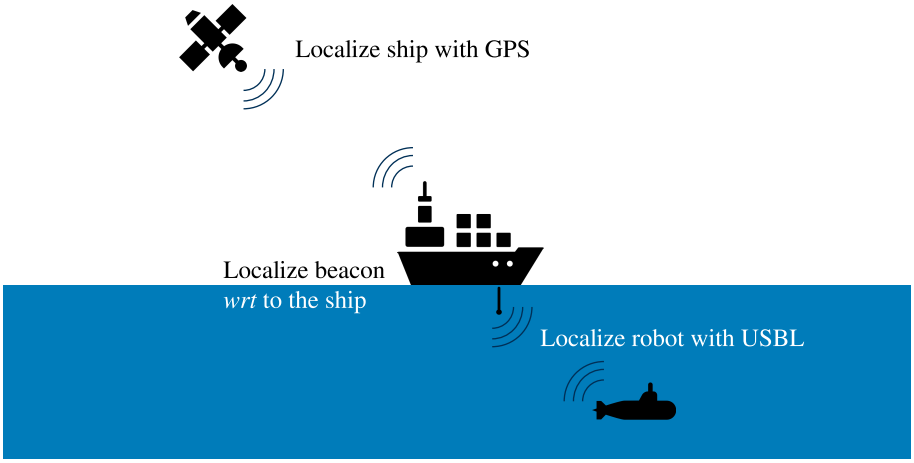


Figure 1.3: **Ultra Short Baseline** acoustic positioning system.

1.2.2 Absolute positioning error

The absolute position of the vehicle mostly relies on the estimate provided by the USBL, as this system offers the sole means of absolute positioning. To gain a perspective on the accuracy of this pose estimate, we examine the Posidonia USBL datasheet from iXblue[2], which is incorporated in Ifremer's vehicles. According to the datasheet, in conjunction with a high-performance INS, the USBL achieves an accuracy of 0.06% relative to the slant range of the vehicle. To put this position accuracy into context, let us consider an idealized scenario: the vehicle is situated at a depth of 6,000 meters directly beneath the ship, thereby minimizing the slant range for this specific depth. Additionally, let us disregard any calibration noise originating from our sensors. In this optimal scenario, the position uncertainty

[2] `https://www.ixblue.com/store/posidonia/`

amounts to 3.6 meters. Employing the three-sigma rule of thumb, this translates to a positional margin of ± 10.8 meters with a confidence level of 99.7%. Essentially, this signifies that our vehicle's position can be estimated within a range of 21.6 meters with a confidence level of 99.7%. In practice, the pose estimate provided by the USBL is fused with the measurements of other sensors to provide accurate relative pose estimates of the vehicle. However, the vehicle's absolute pose is still affected by an offset due to the USBL uncertainty. This error alone suffices to motivate the need for a more precise underwater vehicle localization.

1.2.3 Limitations for autonomous vehicles

In practice, AUVs might not have access to USBL. In such cases, the robot is only aware of its position when it is at the surface before its descent. Once the AUV begins its dive, it must rely solely on the INS, DVL and depth sensor to estimate its position. In this scenario, the DVL operates in water track mode, where its estimation of the vehicle's velocity can be greatly influenced by the water current. As a result, there is a significant drift in the pose estimation when integrating the INS and DVL information. This drift becomes more pronounced as the vehicle travels thousands of meters before reaching the seabed, resulting in highly inaccurate absolute pose estimation.

In the case where AUVs are equipped with USBL, the vehicles absolute position estimates are still affected by an important offset, as previously mentioned. This offset may impair their capability to perform specific tasks, such as mapping a precise area.

1.3 A practical use case scenario

To provide a clear understanding of the specific problem we aim to tackle, it is beneficial to illustrate a use case scenario. Consider a pilot teleoperating a ROV during an oceanographic campaign. The ROV is exploring an unknown site located at a depth of 2,000 meters. During the dive, the ROV acquires some images of the site and records the sensors' data. Once the exploration is finished, the ROV reaches the surface and is picked up by the ship. The data it acquired are then processed and analyzed — scientists agree that there is an interest in monitoring the evolution of this site over the next years. Therefore, one year later, they release an AUV in the ocean above the site of interest with one mission: dive onto the site, map a specific predetermined area and return safely to the surface to be picked up by the ship.

Where does this thesis step in? Considering the previously mentioned absolute positioning uncertainty, by the time it reaches the seabed, the AUV's absolute position estimate may deviate significantly from its actual location. Yet, in order to map precisely the correct area, it needs to accurately determine its position with relation to the previously mapped area. To achieve this, the AUV must rely on the data acquired during the previous dive. More specifically, using onboard sensors and previously acquired data, it needs to recognize the observed site of interest and then retrieve its accurate pose in order to map the correct predetermined area of interest. While some of the previously mentioned navigation sensors might help in providing an estimate of the robot's absolute pose, cameras have proven to have the capability of localizing a robot within its environment very accurately (Sattler et al., 2017; Sarlin et al., 2019; Brachmann and Rother, 2022). This thesis focuses on the methods that rely only on the visual observations of the vehicle to estimate its accurate pose, a process known as visual localization. A multi-modal approach that relies on the multiple sensors that the vehicle embeds remains in the scope of future work.

1.4 Challenges

Addressing the visual localization task in deep-sea environments presents two primary challenges in contrast to solving the same task in terrestrial environments: specificities of deep-sea images and environment, and data scarcity.

1.4.1 Deep-sea images and environment

Beyond a depth of 1,000 meters, we enter the aphotic zone, where no significant light from the surface reaches such depths. As a result, deep-sea vehicles must incorporate an artificial lighting system to illuminate the surrounding scene. This introduces a continuous variation in the lighting conditions of the scene. Additionally, the propagation of light through water is influenced by various physical phenomena, such as scattering. Consequently, underwater images suffer from reduced contrast and shifting colors based on the distance the light has traveled before reaching the sensor (see Figure 1.4). Furthermore, the elements present in the deep-sea environment, such as hydrothermal vents, contrast significantly with the structures typically encountered in terrestrial applications. These elements are also subject to diverse sources of variability, including events like boulder collapses and shifts in marine populations. These differences pose significant challenges for existing computer vision methods, which often struggle to handle underwater images. These difficulties are amplified when images are acquired during different visits that have a substantial temporal gap between them.

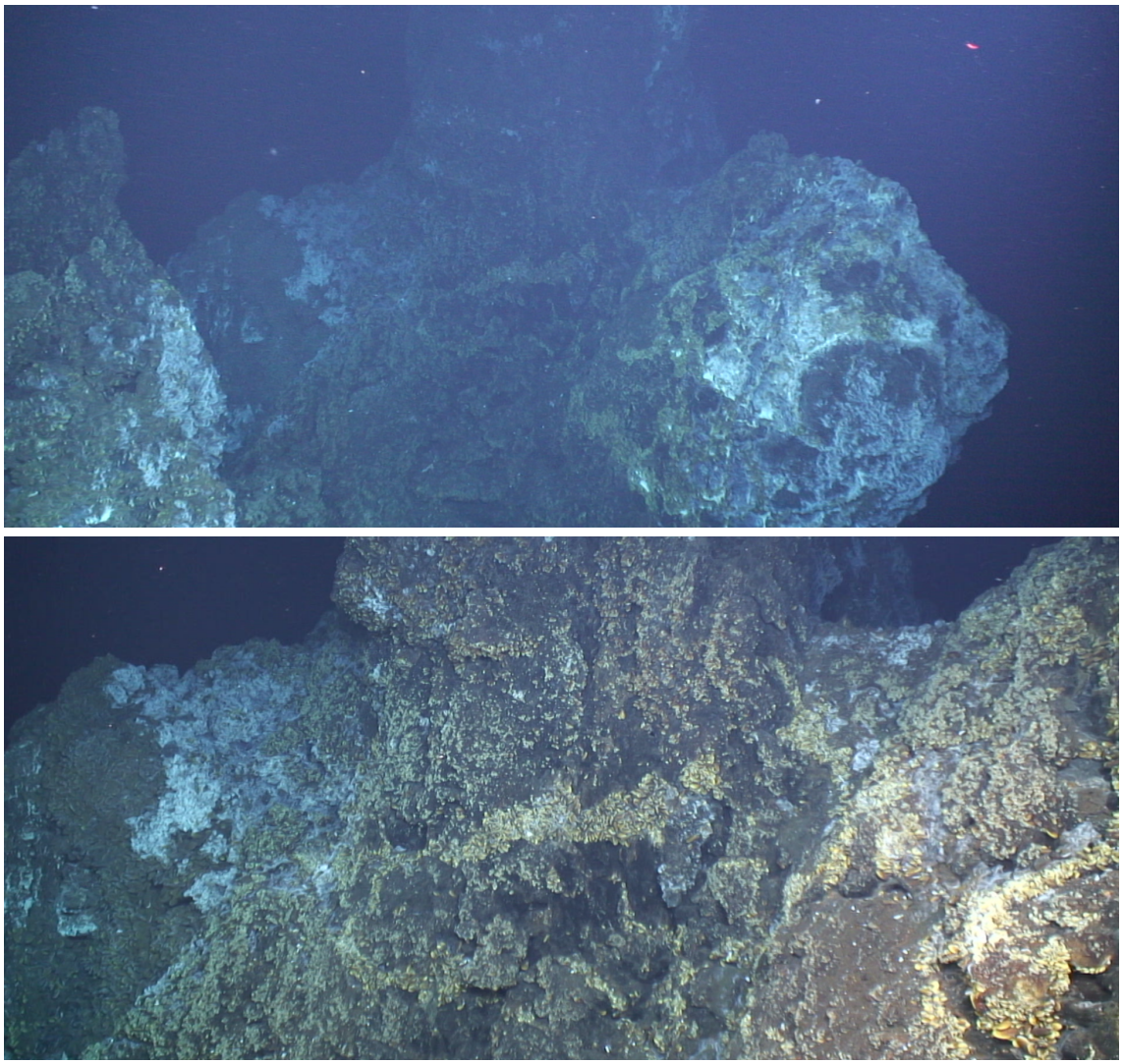


Figure 1.4: **Observing two underwater images** of the same region acquired at different distances reveals that the scene's appearance shifts as the vehicle moves. Because of the artificial lights embedded on the vehicle and light propagation in the water medium, the appearance of the scene strongly depends on the position of the camera.

1.4.2 Data scarcity

Obtaining data in the deep-sea demands substantial efforts and requires extensive equipment. There are only a limited number of underwater vehicles capable of operating at depths up to 6,000 meters. Deploying these vehicles often involves large ships and requires the continuous involvement of the ship's crew throughout the dive. Moreover, not all institutes and organizations share their data publicly, and the collected data may not always be suitable for our specific application. Our application has specific constraints for video recording, such as the inability to use the camera's zoom, as it would temporarily modify the camera's intrinsics parameters.

This data scarcity poses two problems for solving the visual localization task. Firstly, without sufficient data for testing our algorithms, it is difficult to evaluate their accuracy. Secondly, with the emergence of deep learning, many contemporary methods rely on data-driven approaches, and models optimized for terrestrial data may not generalize well to underwater images. Moreover, the limited availability of data makes it challenging to adapt existing solutions to the specificities of underwater images.

1.5 Contributions

This thesis investigates visual localization in the context of long-term deep-sea monitoring. As such, it makes the following contributions:

- We introduce a deep-sea dataset for long-term visual localization. It can be used for the evaluation of localization methods in an underwater environment as well as for training or fine-tuning neural networks that are used in the context of visual localization.
- We propose two new methods for restoring the colors of underwater images, transforming them to appear as if they were captured in-air. The objective is to enable the use of neural networks pretrained on terrestrial data to be effectively applied to underwater images.
- We present a novel loss function that defines the error between two camera poses. This loss function holds a particular significance in the context of deep learning applied to visual localization, as it forms the foundation for how localization neural networks can learn.
- We evaluate several visual localization methods on the newly established dataset. This benchmark includes preprocessing underwater images with the proposed underwater color restoration approaches, as well as training a visual localization network using the suggested loss function.

Publications. These contributions were published in the following articles:

Clémentin Boittiaux, Ricard Marxer, Claire Dune, Aurélien Arnaubec, and Vincent Hugel (2022). Homography-based loss function for camera pose regression. *IEEE Robotics and Automation Letters* and selected for an oral presentation at the *IEEE International Conference on Robotics and Automation*.

Clémentin Boittiaux, Claire Dune, Maxime Ferrera, Aurélien Arnaubec, Ricard Marxer, Marjolaine Matabos, Loïc Van Audenhaege, and Vincent Hugel (2023).

Eiffel tower: A deep-sea underwater dataset for long-term visual localization. *The International Journal of Robotics Research*.

Clémentin Boittiaux, Claire Dune, Aurélien Arnaubec, Ricard Marxer, Maxime Ferrera, and Vincent Hugel (2023). Long-term visual localization in deep-sea underwater environment. In *ORASIS*.

Maxime Ferrera, Aurélien Arnaubec, Clémentin Boittiaux, Inès Larroche and Jan Opderbecke (2023). Vision-based 3D Reconstruction for Deep-Sea Environments: Practical Use for Surveys and Inspection. In *OCEANS*.

Clémentin Boittiaux, Ricard Marxer, Claire Dune, Aurélien Arnaubec, Maxime Ferrera, and Vincent Hugel (2024). SUCRe: Leveraging scene structure for underwater color restoration. Accepted at *3DV*.

1.6 Outline

This thesis is organized as follows:

Chapter 2: An overview of the underwater visual localization problem. This chapter provides the essential background information that serves as a foundation for the entire thesis. It encompasses two primary aspects: the distinctive characteristics of underwater imagery and the visual localization challenge. In the context of underwater images, it delves into the physical phenomena that affect them and introduces models employed to describe these phenomena. It also provides an overview of the methods developed to mitigate these phenomena and recover unaltered images. Concerning visual localization, it outlines a global definition of the problem and conducts a review of common methods related to this problem.

Chapter 3: Building a deep-sea dataset. This chapter introduces a novel underwater dataset designed specifically for evaluating deep-sea long-term visual localization. The dataset comprises images captured during four separate visits to the same hydrothermal vent edifice that span over a five-year period. To establish reference camera poses for evaluating visual localization techniques, we design a Structure-from-Motion (SfM) pipeline that leverages the vehicule's navigation data, and relies on point cloud registration techniques. Additionally, we conduct a thorough analysis of the dataset to gain insights into the significant changes observed over the years. This analysis provides valuable information about the types of changes that can occur in this environment, allowing us to identify potential challenges that visual localization algorithms may encounter.

Chapter 4: Underwater image color restoration. This chapter presents two new methods designed to restore the colors of underwater images, as if they were captured on the surface without the influence of water-induced light propagation effects. These methods take into account the impact of absorption and scattering phenomena, which are strongly influenced by the camera's position relative to the scene. To accurately model and invert these phenomena, the proposed methods make use of the 3D information about the observed scene. More precisely, they rely on SfM outcomes. Subsequently, we empirically validate these approaches using synthetic and real-world datasets, encompassing shallow water and deep-sea scenarios.

Chapter 5: Pose regression for deep learning. We make a focus on the neural networks that can directly predict poses from input images. In particular, we focus on the loss functions that embed the error between two camera poses to perform deep learning-based camera pose regression. Existing loss functions are either difficult-to-tune multi-objective functions or present unstable reprojection errors that rely on ground truth 3D scene points and require a two-step training. To deal with these issues, we introduce a novel loss function which is based on a multiplane homography integration. This new function does not require prior initialization and only depends on physically interpretable hyperparameters.

Chapter 6: Underwater visual localization. This chapter builds upon the research conducted in previous chapters to evaluate a variety of visual localization techniques using the deep-sea dataset we created. Through our analysis, we identify steps within the visual localization process that have room for improvement. We demonstrate that restoring the colors of underwater images can improve visual localization results when using algorithms initially designed for terrestrial environments. Furthermore, our analysis identifies the image retrieval localization step as the primary weakness in underwater localization. We then show that we can improve this step by fine-tuning a neural network pretrained on terrestrial images using a few thousands underwater images.

Chapter 2

An overview of the underwater visual localization problem

Contents

2.1 Underwater images

This thesis revolves around the utilization of computer vision algorithms in underwater environments. Within this context, this section describes the effects of water on the formation of images, aiming to gain a deeper understanding of the underlying sources of variability specific to the underwater setting. Because these effects can potentially deteriorate the performance of computer vision algorithms (Ancuti et al., 2017; Berman et al., 2021), this section also explores how these effects can be modeled and makes a review of methods designed to compensate their impact.

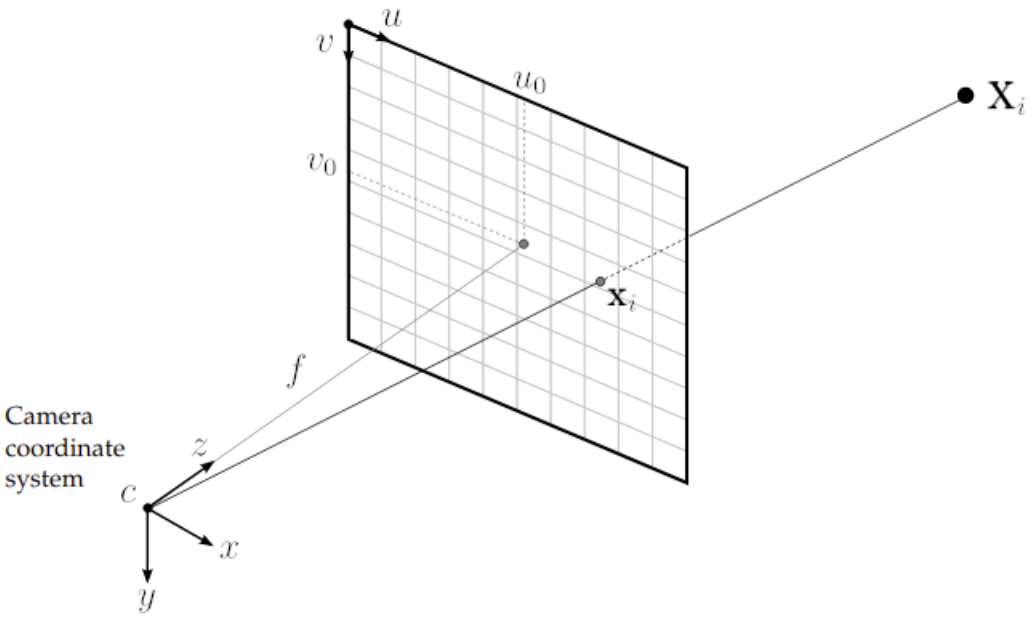


Figure 2.1: **The pinhole camera model.** Adaptation of an illustration extracted from the documentation of the Kornia library (Riba et al., 2020).

Within the context of visual localization, two key aspects need to be considered when working in underwater environments. The first concerns the optical characteristics arising from the changes in light direction due to water, glass and air mediums. These characteristics invalidate the conventional pinhole camera model, which forms the basis of numerous computer vision algorithms. The second is the effect of water on light propagation, that leads to low contrast and color distorted images. Section 2.1.1 details the issues arising from refraction and strategies to circumvent its impact. Section 2.1.2 details how water alters light propagation. Section 2.1.3 presents how to model these alterations in underwater images. Finally, Section 2.1.4 makes a review of algorithms that aim to eliminate these effects and recover the images as if they were captured in the air.

2.1.1 Optical model

Many 3D computer vision algorithms rely on the pinhole camera model (Hartley and Zisserman, 2003) illustrated in Figure 2.1. This model describes how elements in a 3D scene are projected onto the 2D image plane of a camera. In practice, this projection process can be mathematically represented through a homogeneous calibration matrix. This calibration matrix, $\boldsymbol{K} \in \mathbb{R}^{3\times3}$, encapsulates the intrinsic parameters of the camera:

$$\boldsymbol{K} = \begin{bmatrix} f & 0 & u_0 \\ 0 & f & v_0 \\ 0 & 0 & 1 \end{bmatrix}, \tag{2.1}$$

with f the focal length expressed in pixels, and (u_0, v_0) the optical center of the camera expressed in pixels. For a given 3D point

$$\mathbf{X}_i = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \tag{2.2}$$

in the scene, its corresponding homogeneous 2D pixel coordinate $\tilde{\mathbf{x}}_i$ is computed through the application of the calibration matrix:

$$\tilde{\mathbf{x}}_i = \boldsymbol{K}\,\mathbf{X}_i = \begin{bmatrix} fX + Zu_0 \\ fY + Zv_0 \\ Z \end{bmatrix}. \tag{2.3}$$

Following this, the projection function $\pi(\cdot) : \mathbb{R}^3 \to \mathbb{R}^2$, converts this homogeneous coordinate into a conventional Cartesian coordinate, finalizing the process of mapping a 3D point in the scene to its corresponding 2D pixel location on the image:

$$\mathbf{x}_i = \pi(\tilde{\mathbf{x}}_i) = \begin{bmatrix} \frac{fX}{Z} + u_0 \\ \frac{fY}{Z} + v_0 \end{bmatrix}. \tag{2.4}$$

In practice, real-world camera images often experience distortion, meaning that pixels aren't projected onto the image plane in a perfectly linear fashion. To account for this effect, two primary types of distortions are commonly modeled: radial and tangential. Radial distortion is usually characterized using parameters such as k_1 and k_2, which quantify the extent of distortion as it radiates outward from the image center. Tangential distortion is described using parameters like p_1 and p_2, which account for shifts in the image caused by the camera lens not being perfectly aligned with the image sensor. Distortion is computed directly in the camera (x, y) plane, meaning that the de-homogenization process is computed first:

$$\begin{bmatrix} X' \\ Y' \end{bmatrix} = \begin{bmatrix} X/Z \\ Y/Z \end{bmatrix}. \tag{2.5}$$

The distorted coordinates are then computed on X' and Y':

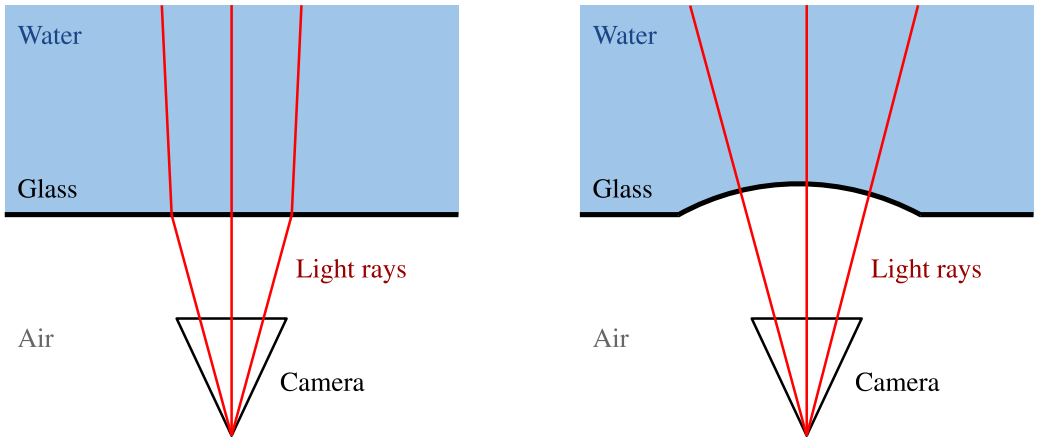
$$\begin{bmatrix} X'' \\ Y'' \end{bmatrix} = \begin{bmatrix} X'(1 + k_1 r^2 + k_2 r^4) + 2p_1 X'Y' + p_2(r^2 + 2X'^2) \\ Y'(1 + k_1 r^2 + k_2 r^4) + p_1(r^2 + 2Y'^2) + 2p_2 X'Y' \end{bmatrix}, \tag{2.6}$$

with

$$r^2 = X'^2 + Y'^2. \tag{2.7}$$

Finally, the 2D pixel coordinate is obtained using the camera intrinsic parameters:

$$\mathbf{x}_i = \begin{bmatrix} fX'' + u_0 \\ fY'' + v_0 \end{bmatrix}. \tag{2.8}$$



(a) Flat glass plane. (b) Dome port.

Figure 2.2: **Effects of refraction on a pinhole camera.** A properly centered dome port camera system is essential to compensate refraction when passing through the air-glass-water mediums. Without a dome port, the linear projection assumption of the pinhole model becomes invalid due to the refraction. In contrast, with a centered dome port, rays pass through these mediums perpendicularly, attenuating refraction and enabling the use of the pinhole model.

However, in underwater settings, cameras are encased within underwater housings positioned behind a glass window for protection. This arrangement results in light rays reaching the sensor after passing through water, then glass, and finally air, leading to refraction, as illustrated in Figure 2.2a. This refraction invalidates the pinhole camera model, as it creates distortion patterns that are dependent on the distance of the observed scene (She et al., 2019; Menna et al., 2020). To alleviate this effect, a commonly used approach is to place the camera behind a spherical glass structure known as a dome port. This dome port ensures that the light rays pass perpendicularly through the air-glass-water mediums, thus preventing refraction, as depicted in Figure 2.2b. While this approach requires to precisely position the camera at the center of the sphere to function optimally, practical experience has demonstrated that given a well-configured dome port camera setup, it is sufficient to approximate the system as a pinhole camera model with standard distortion parameters (She et al., 2022), such as the ones presented above.

Finally, it is important to note that reality is more complex than what is depicted in Figure 2.2. Not all light rays that contribute to a pixel's observation pass through the dome port perpendicularly. The degree of refraction experienced by these rays varies with their wavelengths, inducing chromatic aberration on the edge of the image. The camera's pinhole is never perfectly aligned with the center of the dome port. To address these challenges, Ifremer devised a specialized camera housing comprising a dome port and two lenses, one of which is

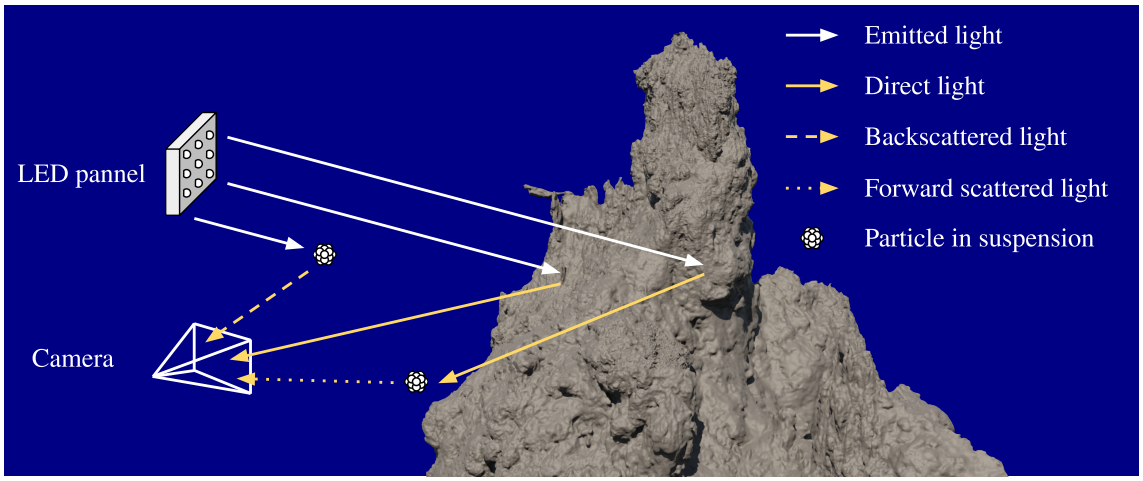


Figure 2.3: **Light propagation under water.** Some of the emitted light is absorbed and converted into other forms of energy as it travels through water. Some photons collide with particles in suspension on their way to the sensor, acting as source of light and inducing scattering.

wavelength-sensitive. This configuration effectively mitigates these effects, aiming to replicate the image produced by a thin lens in air.

2.1.2 Light propagation under water

In the underwater environment, the behavior of light is significantly influenced by both color absorption and scattering phenomena (Duntley, 1963; McGlamery, 1980; Jaffe, 1990; Akkaynak et al., 2017; Akkaynak and Treibitz, 2018), illustrated by Figure 2.3. Absorption refers to the process by which light energy is converted into other forms of energy as it interacts with water molecules and dissolved substances. Different wavelengths of light are absorbed to different degrees. Longer wavelengths, such as red and infrared, are absorbed more strongly than shorter wavelengths like blue and green. This phenomenon is responsible for the gradual loss of color and contrast with increasing distance traveled by the light. Scattering, on the other hand, occurs when light encounters particles suspended in the water. These particles can be small suspended solids, phytoplankton, or other impurities. When light interacts with these particles, it changes direction and scatters in various angles. This scattering effect can lead to reduced visibility, blurring of images, and the diffusion of light, causing underwater scenes to appear hazy or less distinct. Similarly to absorption, this phenomenon varies with wavelengths and depends on the distance traveled by the light. As illustrated in Figure 2.3, there are two primary forms of scattering encountered in practice: forward scatter and backscatter. In contrast to forward scatter, backscattered light does not carry any information about the observed scene (Akkaynak and Treibitz, 2018). In most instances, especially when dealing with deep-sea images captured by

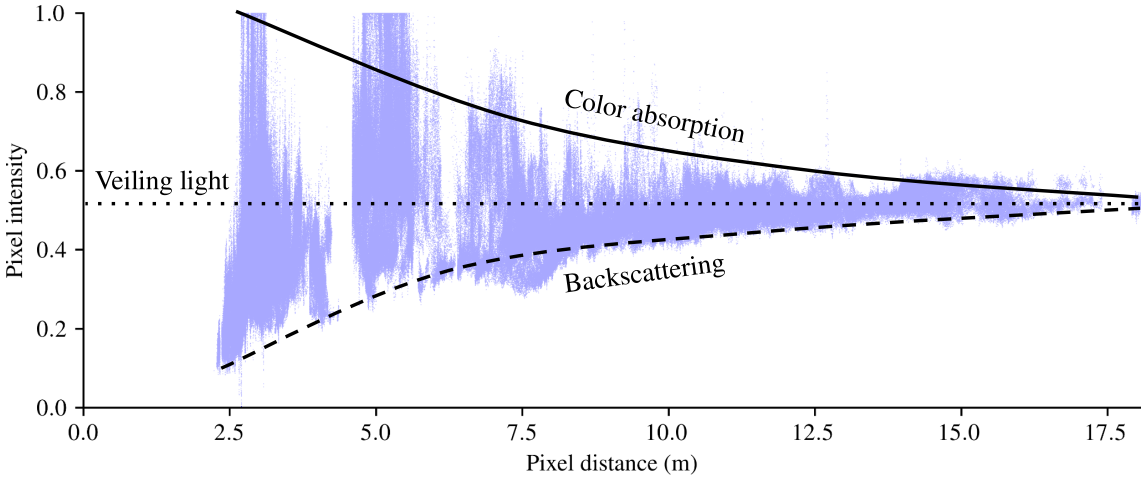


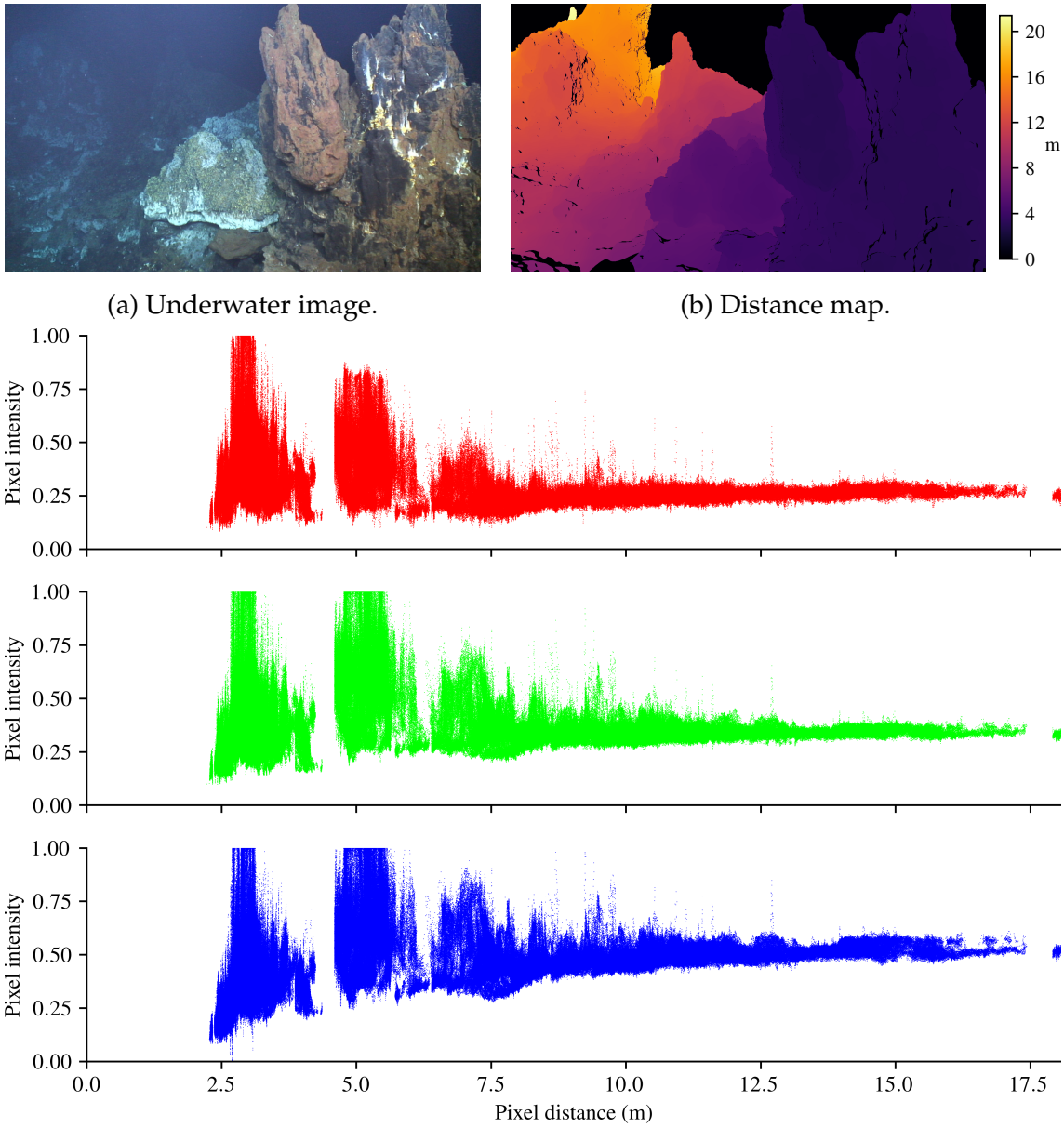
Figure 2.4: **Pixel intensity vs. pixel distance** on the blue channel of a real-world image. Scattering phenomenon adds light. Absorption phenomenon absorbs light. If no object is in sight, the observed color tends towards the veiling light, usually a dark blue in the open ocean.

systems equipped with their own artificial lights, backscatter predominates over forward scatter. As a result, this thesis primarily focuses on this phenomenon.

In order to observe these phenomena in a practical context, we examine the relationship between pixel intensities and their corresponding observation distances. Figures 2.4 and 2.5 effectively illustrate the effects of color absorption and backscattering in underwater images by plotting the intensity of pixels against their distance of observation. Using this representation, Figure 2.4 outlines the effects of color absorption and backscattering on the blue channel of a real-world image. This visualization illustrates how absorption leads to a decline in pixel intensity, whereas backscattering introduces an additional source of light intensity. Both of these factors depend on the pixels' distance of observation and cause pixel intensities to converge towards a specific intensity known as veiling light. In open oceans, this veiling light typically manifests as a deep blue shade. In Figure 2.5, similar plots on red, green and blue color channels illustrate that different wavelengths have different absorption rates, with longer wavelengths like red being absorbed more quickly than shorter ones like blue.

2.1.3 Image formation model

Now that we have seen the main effects that occur in underwater images, we will see how these effects can be modeled in the resulting images. Indeed, a key component of methods that aim to retrieve the appearance of underwater images without the disturbing effects of water is their underwater image formation model that describes how the colors of the observed scene are affected by the water medium.



(a) Underwater image.

(b) Distance map.

(c) Pixel intensity vs. pixel distance on red, green and blue channels.

Figure 2.5: **Impact of observation distance on pixel intensity.** This figure shows an underwater image along with its corresponding depth map obtained using COLMAP SfM (Schönberger and Frahm, 2016) and OpenMVS (Cernea, 2020). Plotting pixel intensities against their distance of observation highlights the absorption and scattering phenomena encountered in underwater scenarios. Comparison between the different color channels highlights how longer wavelenghts, like red, are absorbed more rapidly than shorter ones, like blue.

Variable	Description	Type
$\boldsymbol{I}$	underwater images	$\mathbb{R}^{N\times H\times W\times C}$
$\boldsymbol{J}$	restored images	$\mathbb{R}^{N\times H\times W\times C}$
$\boldsymbol{z}$	distance maps of images	$\mathbb{R}^{N\times H\times W}$
B	veiling light	$\mathbb{R}^{C}$
β	color absorption coefficient	$\mathbb{R}^{C}$
γ	backscatter coefficient	$\mathbb{R}^{C}$
i	image index	$[1..N]$
c	color channel index	$[1..C]$
p	pixel index	$[1..H\times W]$

Table 2.1: **Underwater image formation model variables.** We use subscripts to index specific images, color channels and pixels, *e.g.*, $\boldsymbol{I}_{i,c,p}$ is the intensity of pixel p in the channel c of image i. For single-view applications, the image index i is discarded. Bold symbols are used for variables encoding spatial information such as images or distance maps. H and W are the height and width of images and C is the number of color channels, $C = |\{R, G, B\}|$.

A revised underwater image formation model for computer vision. Many underwater color restoration methods (Chiang and Chen, 2012; Berman et al., 2017, 2021) rely on the underwater image formation model introduced by Schechner and Karpel (2005) to model backscatter and color absorption in natural light conditions. The model describes that pixel intensities are driven by the following equation:

$$\boldsymbol{I}_{c,p} = \boldsymbol{J}_{c,p}e^{-\alpha_c \boldsymbol{z}_p} + B_c(1 - e^{-\alpha_c \boldsymbol{z}_p}), \tag{2.9}$$

where $\alpha \in \mathbb{R}^C$ is the wavelength-dependent coefficient weighting the distance dependency of color absorption and backscatter. The other variables are described in Table 2.1. This model is a simplification of the more complete Jaffe-McGlamery model McGlamery (1980); Jaffe (1990). This simplification facilitates the practical application of the Jaffe-McGlamery model to computer vision algorithms. Nevertheless, Akkaynak et al. (2017); Akkaynak and Treibitz (2018) further revised the model presented in Eq. (2.9) to account for differences between backscatter and absorption coefficients. The relation between pixel intensities and distance of observation is rewritten:

$$\boldsymbol{I}_{c,p} = \boldsymbol{J}_{c,p}e^{-\beta_c \boldsymbol{z}_p} + B_c(1 - e^{-\gamma_c \boldsymbol{z}_p}), \tag{2.10}$$

where β and γ are the absorption and backscatter coefficients defined in Table 2.1. With its two exponential components, one for absorption and one for backscatter, this model describes the curves depicted in Figure 2.4.

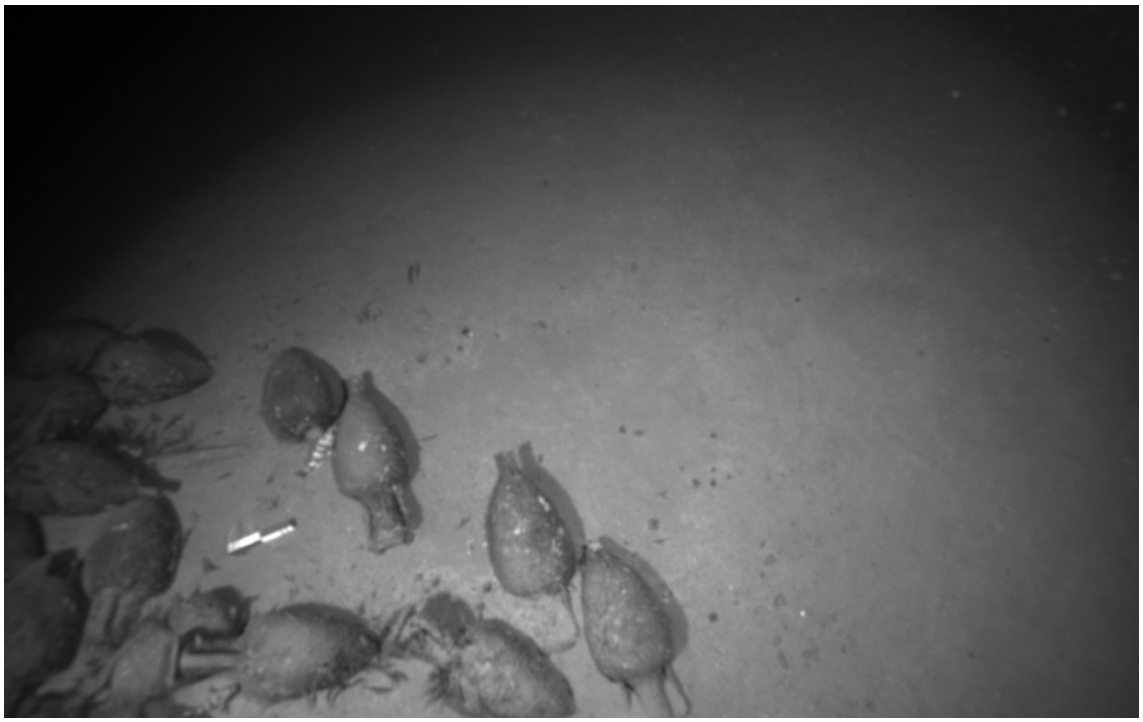


Figure 2.6: **Vignetting effect**. The ROV's artificial lighting system creates a halo on the seabed. This image was taken from the AQUALOC dataset (Ferrera et al., 2019).

Deep-sea lighting. The model described above was developed specifically for natural light conditions. However, in the deep-sea environment, where some of our target scenarios take place, there is no light from the surface. As a result, underwater exploration vehicles need to carry their own artificial lighting systems to illuminate the surroundings. Consequently, the light source becomes an integral part of the exploring robot, causing the illumination of scenes to change as the robot maneuvers. Additionally, as illustrated in Figure 2.6, this lighting system often results in uneven distribution of brightness across the image, commonly referred to as vignetting.

Previous research has addressed the challenge of modeling this phenomenon. An artificial light can be conceptualized as an inverted pinhole camera, projecting a light pattern onto the scene rather than projecting the scene onto the camera view. This light pattern can either be an RGB image Nakath et al. (2021) or expressed as a function (Arnaubec et al., 2015; Bryson et al., 2015). On the one hand, using an RGB image to represent the light pattern offers the advantage of being able to depict various light patterns. On the other hand, it is more challenging to optimize compared to using a function, as it involves a greater number of parameters in the form of an entire image.

2.1.4 Underwater color restoration

Underwater color restoration algorithms aim to rectify underwater images, making them appear as if they were captured in the air, trying to compensate for the effects of water on light propagation. Numerous approaches have been explored to address this challenge using various methods. Some studies have approached the problem from a pure image processing perspective, while others have employed physics-based models, such as those outlined in Eqs. (2.9) and (2.10). Additionally, the required inputs can differ across methods. While many approaches concentrate on restoring the color of individual images, others incorporate 3D scene information as input, such as depth maps or SfM results. Due to the wavelength dependency of the phenomena that affect underwater images, underwater color restoration methods are usually performed on each channel independently. We here present the rational of image processing and physics-based methods that will be evaluated alongside those proposed in Chapter 4.

Image processing methods. Methods that do not rely on any underwater light propagation model mainly aim to enhance the visual appearance of underwater images. To this end, some methods have focused on combining different variants of an input underwater image, like a white balanced or a contrast-enhanced version of the image. This has been done using either statistical analysis (Ancuti et al., 2012) or by learning weight maps using neural networks (Li et al., 2020). Others have leveraged Generative Adversarial Networks to learn a mapping from the underwater domain to an underwater-enhanced domain (Islam et al., 2020; Liu et al., 2022).

Physics-based methods. Methods that rely on underwater light propagation models usually encounter the problem of constraining an underdetermined optimization problem. A notable state-of-the-art approach addressing this specific issue is the *Sea-thru* method. It requires high dynamic range images in a raw file format and their corresponding distance maps (Akkaynak and Treibitz, 2019). The method consists in inverting the underwater image formation model described by Eq. (2.10). With the help of the distance information, the problem has $|\boldsymbol{I}_c|$ equations and $|\boldsymbol{I}_c| + |k|$ unknowns, with $k = \{\beta_c, B_c, \gamma_c\}$ the set of parameters of the image formation model. Given there are more unknowns than observations, the problem is underdetermined and requires additional assumptions to constrain the optimization. For example, an extreme trivial solution can be found with $\boldsymbol{J}_c \rightarrow \boldsymbol{I}_c$, $\beta_c \rightarrow 0$ and $B_c \rightarrow 0$. To tackle this, *Sea-thru* relies on a distance-based alternative to the dark channel prior (He et al., 2010) to retrieve B_c and γ_c, and an illuminant map estimation (Ebner and Hansen, 2013) to retrieve β_c. $\boldsymbol{J}_c$ is then retrieved from Eq. 2.10 using these parameters. On an-

other hand, Berman et al. (2021) propose an underwater image color restoration method based on the principle of Haze-Lines (Berman et al., 2016). Relying on the model outlined in Eq. (2.9), they constrain the estimation of parameters by assuming that colors of the restored image are well approximated by a few hundred distinct colors.

2.2 Visual localization

As mentioned in Chapter 1, improving the localization capability of underwater vehicles is essential for enabling AUVs to perform critical tasks necessary for the continuous monitoring of deep-sea environments, such as conducting a precise survey of a site of interest. Over the past decade, several studies have demonstrated that the visual data collected by robots present a promising solution for achieving accurate vehicle localization (Sattler et al., 2012a; Brachmann et al., 2017; Piasco et al., 2019b; Sarlin et al., 2021; Panek et al., 2022). In this section, we delve into the specificities of the visual localization problem, introduce relevant multi-view geometry concepts, and provide an overview of existing methods employed for visual localization.

2.2.1 Problem definition

Camera pose. Before diving into the definition of the visual localization problem, it is essential to detail the concept of camera pose. A camera pose is a transformation that maps points from a given reference frame, or world coordinate system, to the camera frame. It consists in a rotation component and a translation component, respectively depicting the camera's orientation and position in 3D space. A pose has six degrees of freedom, three for the rotation and three for the translation. Poses are usually represented using a three by three rotation matrix $\boldsymbol{R} \in SO(3)$ and a translation vector $\boldsymbol{t} \in \mathbb{R}^3$. Let ${}^{w}\mathbf{X}$ be a 3D point expressed in the world coordinate system. Let $[{}^{c}\boldsymbol{R}_{w},\, {}^{c}\boldsymbol{t}_{w}]$ be the 6DoF pose of a camera mapping points from the reference to the camera frame. The operation mapping a 3D point ${}^{w}\mathbf{X}$ from the world coordinate system to the camera frame is:

$$ {}^{c}\mathbf{X} = {}^{c}\boldsymbol{R}_{w}\, {}^{w}\mathbf{X} + {}^{c}\boldsymbol{t}_{w}. \tag{2.11} $$

This 6DoF pose can also be represented as a four by four homogeneous transformation matrix:

$$ {}^{c}\boldsymbol{T}_{w} = \begin{bmatrix} {}^{c}\boldsymbol{R}_{w} & {}^{c}\boldsymbol{t}_{w} \\ \mathbf{0}_{1\times 3} & 1 \end{bmatrix} \tag{2.12} $$

For this homogeneous representation, we define the operator $\odot$ that performs the same mapping operation described by Eq. (2.11) more succinctly:

$$ {}^{c}\boldsymbol{T}_{w} \odot {}^{w}\mathbf{X} = {}^{c}\boldsymbol{R}_{w}\, {}^{w}\mathbf{X} + {}^{c}\boldsymbol{t}_{w}. \tag{2.13} $$

This operator has priority over matrix multiplication.

Visual localization. In essence, visual localization consists in determining the precise 6DoF pose of a camera within a known environment based on its acquired image (Sattler, 2013). This challenge finds application in various fields, including robot localization for AUVs or self-driving cars (Maddern et al., 2017), augmented and virtual reality experiences (Sarlin et al., 2022), loop closure in Simultaneous Localization and Mapping (SLAM) (Ferrera et al., 2021), and SfM (Schönberger et al., 2017). Typically, the camera's 6DoF pose is expressed within a specific coordinate system. The process of defining this coordinate system is explored in detail in Section 2.2.2. Furthermore, as the visual localization problem has numerous applications, it has been addressed using various approaches, each relying on different kinds of input data. These diverse strategies are comprehensively outlined in Section 2.2.4.

2.2.2 Reference camera poses

As previously discussed, visual localization consists in retrieving the 6DoF pose of a camera in a specified reference frame. In practice, this reference frame is established in relation to reference camera poses. These reference camera poses can be generated using various algorithms dedicated to this purpose (Brachmann et al., 2021). As detailed in Section 2.2.3, the selection of a particular reference algorithm is not neutral, and significantly influences the outcomes of visual localization methods. Here, we describe some of the reference methods used to compute these reference camera poses.

2.2.2.1 Structure-from-Motion

One of the most popular methods to generate visual localization reference poses is incremental SfM (Li et al., 2012; Kendall et al., 2015; Sun et al., 2017; Sattler et al., 2018; Arnold et al., 2022). SfM takes as input a collection of images and leverages multi-view geometry principles (Hartley and Zisserman, 2003) to estimate camera poses, intrinsics and a representation of the scene in the form of a point cloud. Incremental SfM workflows (Schönberger and Frahm, 2016) typically comprise several distinct phases, which we outline here in the order in which they are executed.

Local feature extraction. The initial step in SfM involves the extraction of local features from each image. These local features consist of two fundamental components: a keypoint, which represents a 2D point in the image denoted in pixel coordinates, and a visual descriptor, which is a vector characterizing the keypoint based on its surrounding context within the image. These local features can be either crafted manually (Lowe, 2004; Bay et al., 2006; Tola et al., 2010; Calonder et al., 2010; Rublee et al., 2011) or learned through deep learning techniques (DeTone et al., 2018; Ono et al., 2018; Revaud et al., 2019; Dusmanu et al., 2019; Tyszkiewicz et al., 2020).

When crafting features manually, various criteria come into play during their selection, including considerations such as scale, orientation and illumination invariance, computational efficiency, and the dimension of descriptor vectors. Garcia-Fidalgo and Ortiz (2015) have conducted a comprehensive survey of local feature descriptors that find application in localization and mapping algorithms.

Conversely, learned local features are characterized through neural networks. Schonberger et al. (2017) provide a comparative analysis between handcrafted and learned features, and demonstrate that advanced handcrafted features may still perform on par or better than some learned features depending on the specific application.

Image retrieval. The next phase in the SfM process involves finding images that observe the same scene elements. This is usually achieved by computing and comparing global descriptors. In contrast to the local descriptors used in the previous step, these global descriptors characterize the entire image. A common approach for computing these global descriptors involves the aggregation of local descriptors extracted from the image. Approaches for image retrieval vary in terms of whether they are manually crafted (Sivic and Zisserman, 2003; Philbin et al., 2007; Jégou et al., 2010; Oliva and Torralba, 2001; Benbihi et al., 2020) or learned (Arandjelovic et al., 2016; Radenović et al., 2016; Gordo et al., 2016).

A popular method for image retrieval is the Bag of Words (BoW) approach (Sivic and Zisserman, 2003; Philbin et al., 2007), which is inspired by text document research. Given a set of images, BoW extracts local features, such as those mentioned above, for all these images. These features are then grouped into clusters to obtain a compact representation. Each cluster's center is referred to as a visual word. For a given image, BoW assigns each of the image's local features to its closest visual word. The image is then represented by a global descriptor vector whose length is equal to the number of visual words. Each dimension of this vector is proportional to the frequency of the associated visual word within the image. Another noteworthy descriptor closely related to BoW is the Vector of Locally Aggregated Descriptors (VLAD) (Jégou et al., 2010). Unlike BoW, VLAD

does not perform a hard assignment of local features to their nearest visual word. Instead, it performs soft assignment by storing the distance between local descriptors and their nearest visual word.

In recent years, neural networks have emerged as powerful tools for image representation across various applications. Consequently, numerous methods have focused on employing neural networks to either extract or aggregate local features into global descriptors. One such method, NetVLAD (Arandjelovic et al., 2016), extends the VLAD method by simultaneously learning dense local features and the assignment of visual words, using Convolutional Neural Networks (CNN) for this purpose.

Feature matching. Feature matching consists in pairing two sets of local features extracted from different images. In SfM, this step is performed for each input image between the input image and its previously retrieved similar images.

The most straightforward method for addressing the feature matching challenge is brute force matching. The following description of this method is inspired by Ferrera (2019). Let us consider two images, $\boldsymbol{I}_1$ and $\boldsymbol{I}_2$. During brute force matching, the system computes the descriptor distances between all local features extracted from both $\boldsymbol{I}_1$ and $\boldsymbol{I}_2$. The most likely matches are recovered by looking for the pairs with the lowest descriptor distance. To enhance the accuracy of matches and reduce incorrect pairings, the computation of descriptor distances is carried out twice — from $\boldsymbol{I}_1$ to $\boldsymbol{I}_2$ and from $\boldsymbol{I}_2$ to $\boldsymbol{I}_1$. Given that a perfect one-to-one correspondence between features in both images is highly improbable, different outcomes may emerge from these two calculation steps. As a result, the most plausible matches are those with the lowest distances in both directions. To further fortify the matching process, an additional step involves examining the second lowest distance for each pair. A pair is only retained if the difference between the lowest matching distance and the second lowest distance exceeds a predefined threshold (Lowe, 2004). This method effectively filters out ambiguous matches, ensuring that only distinctive ones are retained.

Contemporary work has also tackled the matching problem by leveraging deep learning techniques such as Transformers (Vaswani et al., 2017) and Graph Neural Networks (GNN) (Gilmer et al., 2017). While some methods solve an optimal matching problem between two sets of sparse local features (Sarlin et al., 2020; Lindenberger et al., 2023; Edstedt et al., 2023a), others focused on extracting and matching dense local features (Sun et al., 2021; Chen et al., 2022a; Edstedt et al., 2023b).

Incremental reconstruction. The reconstruction phase of SfM involves a cyclic process of image registration and bundle adjustment. Image registration en-

tails incorporating a new image into the existing model. This is often achieved through a pose estimation procedure followed by local optimization. This pose estimation task centers around solving the Perspective-n-Point (PnP) problem, typically within a RANdom SAmple Consensus (RANSAC) scheme. The PnP problem aims to determine the camera's position and orientation based on its intrinsic parameters and a set of correspondences between 3D points and their 2D projections. To solve this problem, a minimum of three correspondences is required (Gao et al., 2003). This minimal setting is referred to as the P3P problem. However, in this scenario, up to four valid sets of position and orientation can be obtained, necessitating a fourth point to resolve the correct pose. Several algorithms have been proposed to directly estimate a pose solution from four or more correspondences (Lepetit et al., 2009; Kneip et al., 2014). In practice, 2D-3D pairs are subject to considerable noise and contain a substantial number of outliers. To address this challenge, the PnP problem is often initialized within a RANSAC scheme. RANSAC is a robust statistical algorithm used to estimate model parameters from noisy data (Fischler and Bolles, 1981). It iteratively samples minimal subsets of data points (three or four 2D-3D pairs in this context), fits a model to each subset, and identifies the model with the best consensus with the data, making it resilient to outliers and noise. After obtaining an initial pose and a set of inliers using RANSAC, the final pose is refined through a least squares optimization process applied to the remaining inliers. All registered images are then refined using bundle adjustment. Bundle adjustment is an optimization problem that simultaneously optimizes the poses of cameras and the position of 3D scene points observed by these cameras. In some applications, this optimization may involve other parameters, such as camera intrinsics. This optimization task is accomplished by minimizing the distance between the projection of 3D points in multiple camera views and their corresponding detected 2D keypoints coordinates. A more comprehensive exploration of bundle adjustment is presented in Section 3.4.3, where we formulate it to incorporate known position priors.

2.2.2.2 Visual Simultaneous Localization and Mapping

An alternative approach to acquiring reference camera poses and scene representation is through visual SLAM (Shotton et al., 2013; Glocker et al., 2013). Visual SLAM is similar to SfM but is tailored for real-time operation (Mur-Artal et al., 2015; Ferrera et al., 2021), and predominantly uses frames from sequential video acquisition, whereas SfM handles unordered sets of images. In visual SLAM, compromises are often made, like conducting bundle adjustment locally rather than on all images. Additionally, SLAM algorithms may integrate additional sources of information, such as depth measurements obtained from RGB-D cameras (Newcombe et al., 2011).

2.2.2.3 Motion capture systems

An alternative method for acquiring accurate 6DoF reference camera poses is to employ motion capture systems. These systems use a set of calibrated cameras to detect and triangulate markers attached to a moving object. Nielsen et al. (2019) used a motion capture system to gather underwater data within a pool, with the goal of evaluating the performance of a visual localization algorithm (Kendall et al., 2015) in underwater scenarios.

2.2.2.4 Leveraging multiple sensors from augmented reality devices

Sarlin et al. (2022) design a full pipeline to estimate ground truth trajectories using data acquired with augmented reality devices such as the HoloLens. To achieve this, they leverage images, 3D LiDAR, inertial and radio data.

2.2.3 Pseudo ground truth

We have just discussed various approaches for generating ground truth reference camera poses using different data inputs. Nevertheless, it is essential to recognize that these generated references do not provide an absolute ground truth of each camera's position with infinite precision. Despite rigorous efforts (Sarlin et al., 2022), all of these methods, remain approximations with varying accuracy with respect to the real pose, and most importantly they will be the source of biases.

The different methods used to define the ground truth reference frames are significant factors to consider in the context of visual localization, as visual localization approaches are designed to determine 6DoF images poses with respect to these reference frames. Interestingly, this aspect has only recently gained substantial attention. In a recent study, Brachmann et al. (2021) demonstrated that visual localization methods that optimize a similar cost function as the reference algorithm are more adept at reproducing the local minima and imperfections of the generated ground truth. Consequently, depending on the application, a visual localization algorithm may derive greater benefit from minimizing one specific objective metric over another. Another consequence of this discovery is that the evaluation of visual localization algorithms should always be interpreted in conjunction with the specific method employed to construct the ground truth for benchmark datasets.

2.2.4 A review of visual localization methods

This section makes a review of existing visual localization methods.

Retrieval-based. Retrieval-based approaches share strong similarities with SfM techniques (Sattler et al., 2012b; Taira et al., 2018; Sarlin et al., 2019; Humenberger et al., 2022). They take as input a SfM ground truth database containing images that observe the same scene as the query image. The SfM database already contains local features extracted from database images along with their corresponding 2D-3D correspondences. Retrieval-based methods proceed through three fundamental stages. First, they use image retrieval techniques to identify images within the database that closely resemble the query image. Then, they perform feature matching between the local features extracted from the query image and those from the retrieved images. Finally, they employ a pose estimation algorithm like PnP/RANSAC to determine the query image's 6DoF pose.

Direct matching. Direct matching methods also require a SfM ground truth database as input (Irschara et al., 2009; Sattler et al., 2012a, 2017). However, they differ in that there is no image retrieval step involved. Instead, 2D-3D correspondences are directly obtained from the pool of 3D points provided by the SfM database and their corresponding local features. Subsequently, the query image's DoF pose is determined using algorithms such as PnP/RANSAC.

Scene coordinates regressors. Instead of relying on an external explicit database, scene coordinates regressors encode scene coordinates, or 3D points, directly within a neural network's weights (Brachmann et al., 2017; Brachmann and Rother, 2018, 2022). This neural network is trained per scene in an end-to-end manner, from the image to the pose, to learn the scene coordinates corresponding to image patches. The training process still requires a ground truth database acquired through SfM or RGB-D SLAM. During training, they employ a differentiable version of the PnP/RANSAC scheme to estimate the image's 6DoF pose and enable the backpropagation of pose errors throughout the entire network. A query image's 6DoF pose can then be retrieved from a single network inference.

Absolute pose regressors. Similarly to scene coordinates regressors, these approaches implicitly encode the scene's information in the weights of a neural network (Kendall et al., 2015; Kendall and Cipolla, 2017; Shavit et al., 2021). This network is trained for each individual scene in an end-to-end manner to directly estimate a 6DoF pose from an input image. Training this network only necessitates pairs of images along with their associated ground truth poses. However, Sattler et al. (2019) have revealed that these systems lack the capability to accurately model 3D geometry principles. As a result, these methods only have a limited accuracy that aligns more closely with image retrieval performance than precise 6DoF pose estimation.

Structure-based. Some studies have concentrated on localizing images using the scene's structural information. Panek et al. (2022) argue that storing all the images and local features of a SfM ground truth database is impractical. Instead, they rely only on a reconstructed 3D mesh to dynamically render images and conduct feature matching between the query image and these rendered images. In contrast, Piasco et al. (2019a) leverage learned depth maps to derive a local 3D point cloud for a given query image. Subsequently, they employ the Iterative Closest Point (ICP) method, an iterative algorithm for aligning two sets of 3D points, to align the query image's local 3D point cloud with the global point cloud of the scene, which was initially generated using SfM or RGB-D SLAM.

NeRF. Moreau et al. (2022) train an absolute pose regressor on a set of both real-world images and synthesized images obtained with Neural Radiance Fields (NeRF) (Mildenhall et al., 2020; Martin-Brualla et al., 2021). They show that by exposing the network to a more extensive range of viewpoints, they significantly improve visual localization performance. Nevertheless, the applicability of this approach in underwater settings is somewhat restricted due to NeRF's difficulties in modeling radiance fields within scattering media. Nonetheless, recent research efforts have made considerable progress in adapting NeRF techniques to underwater scenarios (Sethuraman et al., 2022; Levy et al., 2023).

Map-free. The objective of map-free localization is to determine the 6DoF camera pose of a query image using just a single reference image along with its camera intrinsics. This problem scenario is primarily driven by augmented reality applications, where a comprehensive scene representation might not be accessible as an input for localization. However, using only one image does not provide the necessary information to retrieve the scale of the scene. To address this challenge, Arnold et al. (2022) employ 2D feature matching (Lowe, 2004; DeTone et al., 2018; Sarlin et al., 2020; Sun et al., 2021) to estimate the relative pose between the query and reference images, up to a scale factor. This scale factor is subsequently determined by leveraging deep single-image depth estimation models (Liu et al., 2019; Ranftl et al., 2021).

2.3 Conclusion

Within this chapter, we have delved into key notions for long-term visual localization in underwater settings. Specifically, we have presented the physical phenomena influencing the formation of underwater images, encompassing refraction and light propagation under water. We introduced image formation models

designed to account for underwater light propagation and discussed methods developed to mitigate their effects, which range from model-based to statistical and deep learning approaches. Shifting the focus to visual localization, we provided a description of the problem in its general form, and offered insights into the mathematical foundations and common strategies employed to address it. Notably, we discussed the algorithms frequently utilized for generating reference camera poses and conducted a survey of existing visual localization methods, classifying them into distinct categories.

Chapter 3

Building a deep-sea dataset

Contents

3.1 Introduction

This thesis focuses on visual localization in the context of deep-sea long-term monitoring. As discussed in Section 2.1, underwater images present distinct sources of technical and environmental variability that are not encountered in terrestrial environments. Consequently, existing terrestrial datasets are inadequate for evaluating long-term localization performance in underwater scenarios. Therefore, there is a need for a deep-sea visual localization dataset that accounts for the specific difficulties faced in this environment, such as those described in Section 2.1. Such a dataset should cover all the factors that could impair visual localization algorithms in this environment. More specifically, it should be acquired over an extended period of time and employ diverse acquisition systems to encompass the full range of topological, environmental, and robotic equipment variations inherent to deep-sea operations.

Over the past decade, the problem of visual localization has received significant attention, primarily driven by the increasing interest in self-driving cars. Within this context, there has been a particular focus on addressing the challenges posed by dynamic and changing environments. Consequently, research efforts have primarily centered on enhancing the robustness of visual localization in terrestrial settings (Benbihi, 2020). Most of the available datasets for evaluating visual localization methods consist of terrestrial data (Griffith et al., 2017; Sattler et al., 2018). These datasets encompass a wide array of environmental variations, including day-night transitions, seasonal changes, diverse weather conditions, alterations in natural landscapes, and occlusions caused by vehicles or pedestrians. Yet, they lack some specific characteristics of underwater images, such as absorption and scattering.

To address the specific challenges of deep-sea environments, we introduce in this chapter a new underwater dataset to benchmark deep-sea long-term visual localization. In this context, this chapter makes the following contributions:

- We make a review of existing visual localization datasets, and outline the requirements for building a deep-sea dataset. Subsequently, we describe the data acquisition process that was employed for creating this dataset.

- We describe how we estimate reference camera poses for the dataset using SfM, and delve into the challenges encountered when applying SfM to deep-sea data, especially when dealing with images from different visits. To tackle these challenges, we conduct a detailed study of each step in the SfM process to identify potential difficulties. As a result, we propose a novel pipeline that facilitates the creation of a unified reference model, consolidating the camera poses from all visits into a common reference frame.

- We conduct an analysis of the proposed dataset to provide insights about the major changes observed throughout the years. This analysis shows that the constructed dataset captures long-term naturally occurring transformations, such as chimney collapse and population shifts, making it highly valuable for evaluating visual localization algorithms.

This chapter is organized as follows. First, Section 3.2 makes a review of existing visual localization datasets. Then, Section 3.3 outlines the data collection process. Subsequently, Section 3.4 outlines the SfM pipeline to generate the reference camera poses. Finally, Section 3.5 analyses the long-term changes captured by the dataset.

3.2 Existing datasets

The benchmark datasets commonly used for evaluating visual localization algorithms primarily consist of terrestrial scenes. Given the large interest in the problem, there exists a wide range of datasets available that encompass diverse environments and offer distinct challenges. Aachen Day-Night, RobotCar Seasons, CMU Seasons (Sattler et al., 2018) and Cambridge (Kendall et al., 2015) datasets were acquired in outdoor environments while 7-Scenes (Shotton et al., 2013) and 12-Scenes (Valentin et al., 2016) datasets were captured indoor. However, the availability of similar datasets in underwater environments is limited due to the high cost of data collection.

Existing underwater datasets (Mallios et al., 2017; Ferrera et al., 2019) primarily focus on supporting the development of underwater SLAM algorithms. For example, the AQUALOC dataset (Ferrera et al., 2019) offers synchronized monochromatic underwater images along with inertial and depth data from three different sites off the coast of Corsica. It includes a harbor site at a depth of approximately four meters and two archaeological sites at depths of 270 meters and 380 meters. Although the dataset contains sequences with different trajectories, all the visits were conducted on the same day, limiting the representation of various environmental changes that can occur under water. For instance, variations in turbidity and changes in marine populations and sedimentation are not fully captured. Nielsen et al. (2019) built an underwater dataset within a controlled pool environment. They also provide ground truth camera poses obtained with an underwater motion capture system. Additionally, there are underwater datasets available for different tasks like image enhancement and color restoration (Akkaynak and Treibitz, 2019; Li et al., 2020; Berman et al., 2021), but they do not provide image sequences of the same site covering long periods of time.

3.3 Data collection

Context. Since 2010, the EMSO-Azores deep-sea observatory situated on the Mid-Atlantic Ridge has facilitated the continuous monitoring of the hydrothermal vent field known as *Lucky Strike* (Figure 3.1). During annual maintenance cruises (Cannat and Sarradin, 2010), a ROV operated by Ifremer, named Victor6000 (Figure 3.2), has been deployed to investigate the evolution of hydrothermal circulation and associated fauna communities over multiple years (Matabos et al., 2022). Within the explored area, particular focus has been given to the hydrothermal vent edifice named Eiffel Tower, located at a depth of 1700 meters below the surface. Since its discovery in 1992 (Langmuir et al., 1993), four dives were dedicated to the 3D reconstruction of this vent, taking place in 2015, 2016,

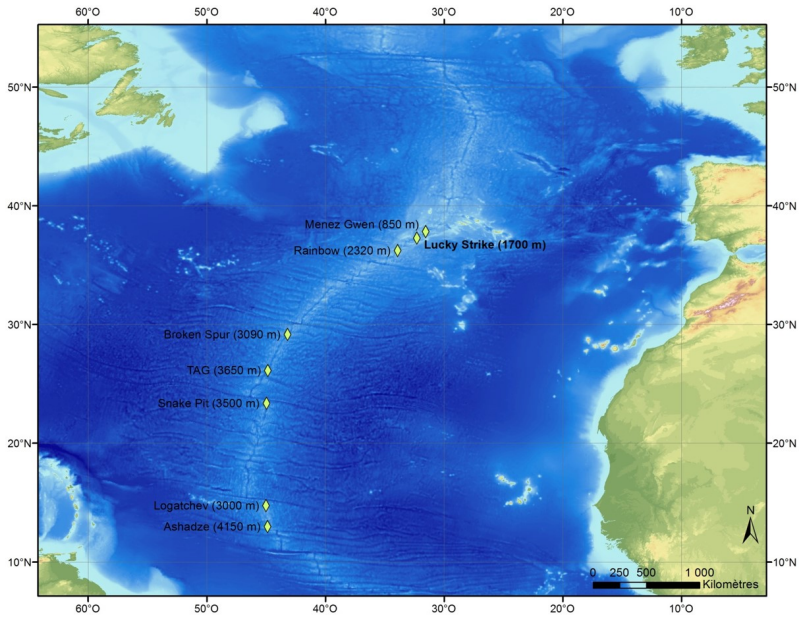


Figure 3.1: **Location of the Lucky Strike vent field** on the Mid-Atlantic Ridge (Sources: Esri, GEBCO, NOAA, National Geographic, DeLorme, HERE, Geonames.org).

Year	Camera	Resolution	Frame rate
2015	SONY FCB-H11	1920x1080 px	25 fps
2016	SONY FCB-H11	1920x1080 px	25 fps
2018	SONY FCB-H11	1920x1080 px	25 fps
2020	DeepSea Apex 4K Power & Light	3840x2160 px	30 fps

Table 3.1: **Camera settings** for the four dives. A different camera was employed for the 2020 dive.

2018, and 2020. These dives have enabled quantitative monitoring of the distribution and dynamics of the vent community (Girard et al., 2020), and also provide valuable data for our target application. As such, the data acquired during these four visits will serve as foundation for constructing our deep-sea visual localization dataset.

Acquisition requirements. To construct the reference model using SfM, specific acquisition requirements must be met during each visit. *Trajectories:* The edifice must be fully observed and tracked continuously to ensure that there are sufficient overlapping views for SfM processing. *Lighting:* Adequate lighting is necessary to illuminate the scene while minimizing the vignetting effect caused by the artificial lighting system. *Refraction:* To preserve the linear projection assumption of the pinhole model, the camera must be equipped with a custom dome port that corrects the refraction induced by the air-glass-water mediums (see Section 2.1.1).

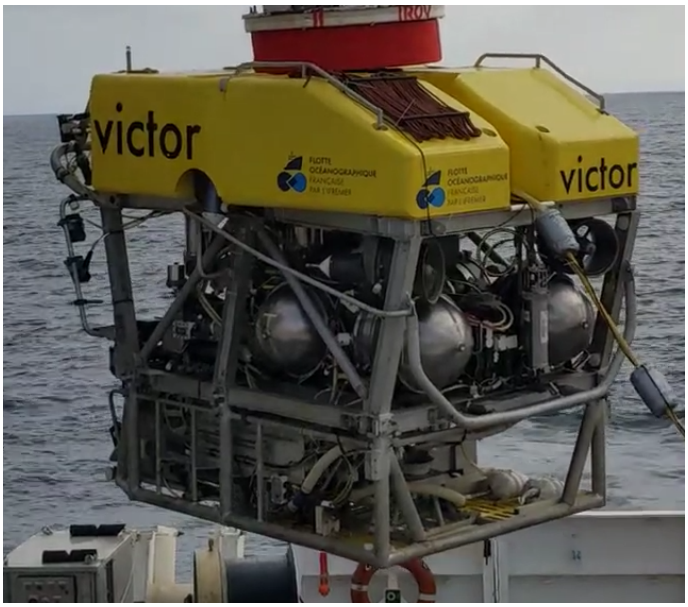


Figure 3.2: **The ROV Victor6000** operated by Ifremer.

Zoom: Using the zoom should be avoided to maintain the consistency of the pinhole model. *Navigation:* The ROV's navigation data should be synchronized with the video recording system to ensure that each frame is associated with a known position prior, which is helpful for the image pairing and bundle adjustment steps in SfM.

Acquisition setup. To meet the specified acquisition requirements, the following setup is used. *Navigation:* The ROV Victor6000 embeds an USBL, INS, a DVL and a depth sensor described in Section 1.2. These sensors are fused similarly to Guerrero-Font et al. (2016) to compute the navigation data. These navigation data provide estimates of the ROV's latitude, longitude and altitude at a frequency of 1 Hz, all synchronized with the video recording. *Cameras:* Over the years, the vehicle's camera was replaced. Details about the camera specifications used for each year can be found in Table 3.1. Ifremer designed and equipped each camera with customized housing and special lenses that alleviate refraction effects. As these lenses significantly mitigate camera distortion, the cameras were calibrated under water using a simple second-order radial distortion model, as defined on Section 2.1.1. *Lighting:* The ROV was equipped with an artificial lighting system comprising 12 LED panels, each delivering 20,000 lumens.

Acquired data. Following the acquisition setup and requirements listed above, the ROV conducted four dives into the Eiffel Tower vent in 2015, 2016, 2018 and 2020, recording a total of 17 hours of synchronized video and navigation data. Navigation data were recorded at a frequency of 1 Hz in a text format similar to:

$timestamp1 \quad lat_1 \quad lon_1 \quad alt_1$
$timestamp2 \quad lat_2 \quad lon_2 \quad alt_2$
...

The acquired data satisfies all the requirements listed in Section 3.1 for evaluating visual localization in deep-sea environments, covering the same site over an extended period using different recording equipment.

3.4 Building a reference model

Benchmarking datasets for visual localization necessitates reference camera poses for every image. These poses can be obtained through various approaches. The most commonly utilized methods, SfM and depth-based SLAM, enable the acquisition of both camera poses and scene geometry. However, underwater applications face challenges with depth-based SLAM due to the absorption of infrared light in water, making it more difficult to set up. Underwater LiDAR systems, for instance, demand additional preparation, and sonar data can be prone to noise. An alternative approach for recovering camera poses without the scene's geometry is to use motion capture systems. Nonetheless, motion capture are impractical in deep-sea missions due to the challenges in deploying such a system over vast areas spanning hundreds of square meters at depths exceeding a thousand meters. Consequently, SfM emerges as a practical solution for estimating camera poses and a point cloud of the scene in deep-sea environments.

Input data. From the video samples acquired at each dive, we extract one image every three seconds for a total of 18,082 images. From the navigation data, we interpolate linearly for each image its estimated latitude, longitude and altitude.

Objective. For all the extracted images, our goal is to retrieve their 6DoF camera poses in a common reference frame, consistent across the four different years. More specifically, we aim to build a unified reference model using SfM, by ensuring that all SfM processing steps described in Section 2.2.2 can effectively handle underwater data from different years.

Challenges. When working with underwater data, traditional methods used in the SfM process encounter limitations at every stage of the reconstruction. *Image retrieval* methods encounter difficulties in pairing images from different acquisitions. *Feature matching* algorithms struggle to generalize to some topological and environmental changes. *Bundle adjustment* can benefit from incorporating navigation data as prior knowledge. Sections 3.4.1 to 3.4.3 provide detailed explanations of the methods employed to overcome each of these difficulties.

3.4.1 Image retrieval

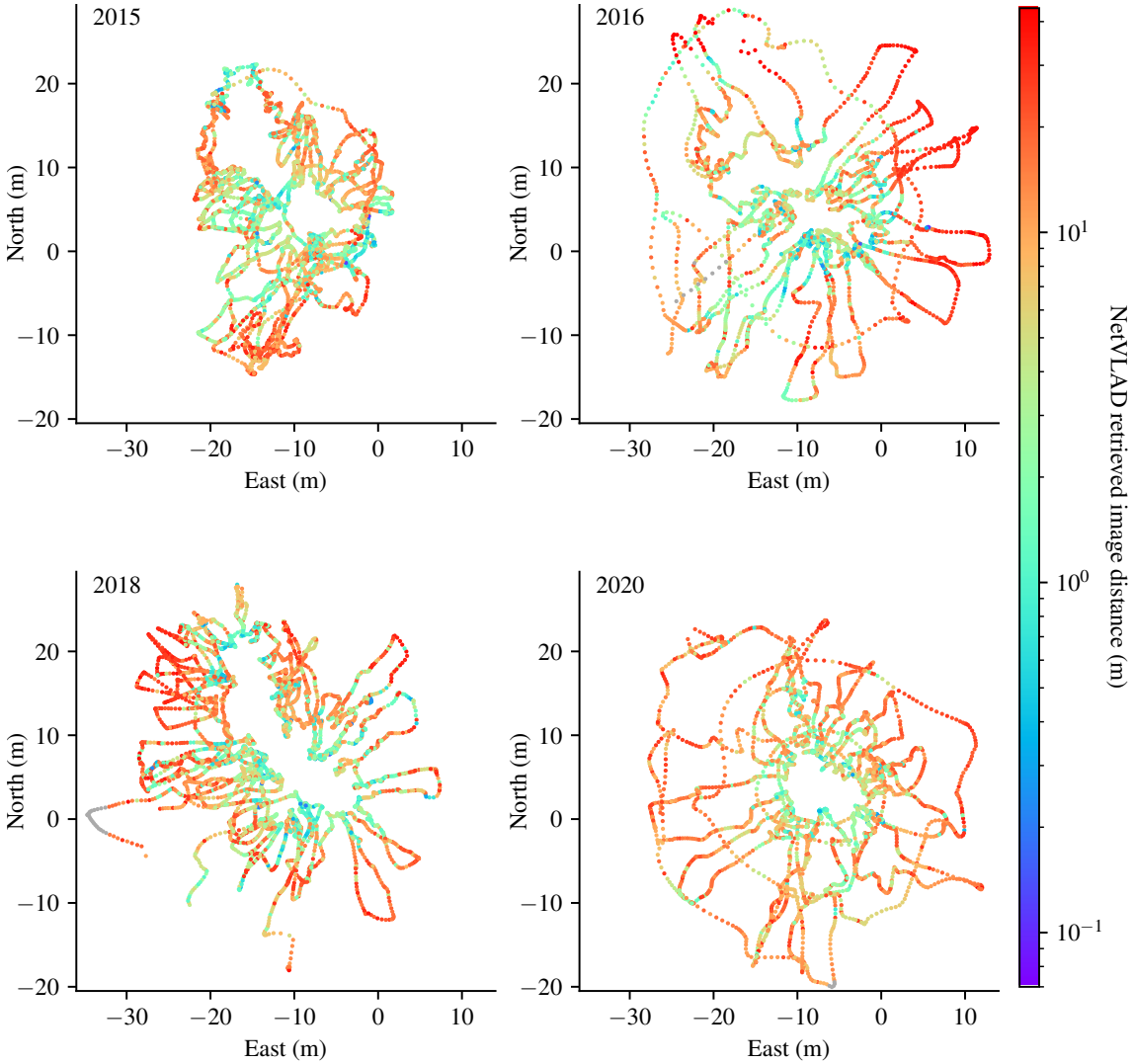


Figure 3.3: **NetVLAD performance on cross-years image retrieval.** We compute the 4096-dimensional NetVLAD descriptor of every image. For each query image, we select its retrieved image as the one with the closest NetVLAD descriptor amongst images of different years. We display the query image's east and north position colored with its spatial distance to the retrieved image. Images that have no retrieval candidates within a five meters distance are rendered grey.

As described in Section 2.2.2, a primary step in the SfM procedure involves determining which images should be paired together. Indeed, as the target model comprises thousands of images, it is impractical to consider every possible image pair combination for local feature matching. This image pairing step can be accomplished through various approaches.

One intuitive approach is to make use of the available navigation data and only pair images that are spatially close to each other. However, this approach is only applicable for pairing images within the same visit year. As discussed in

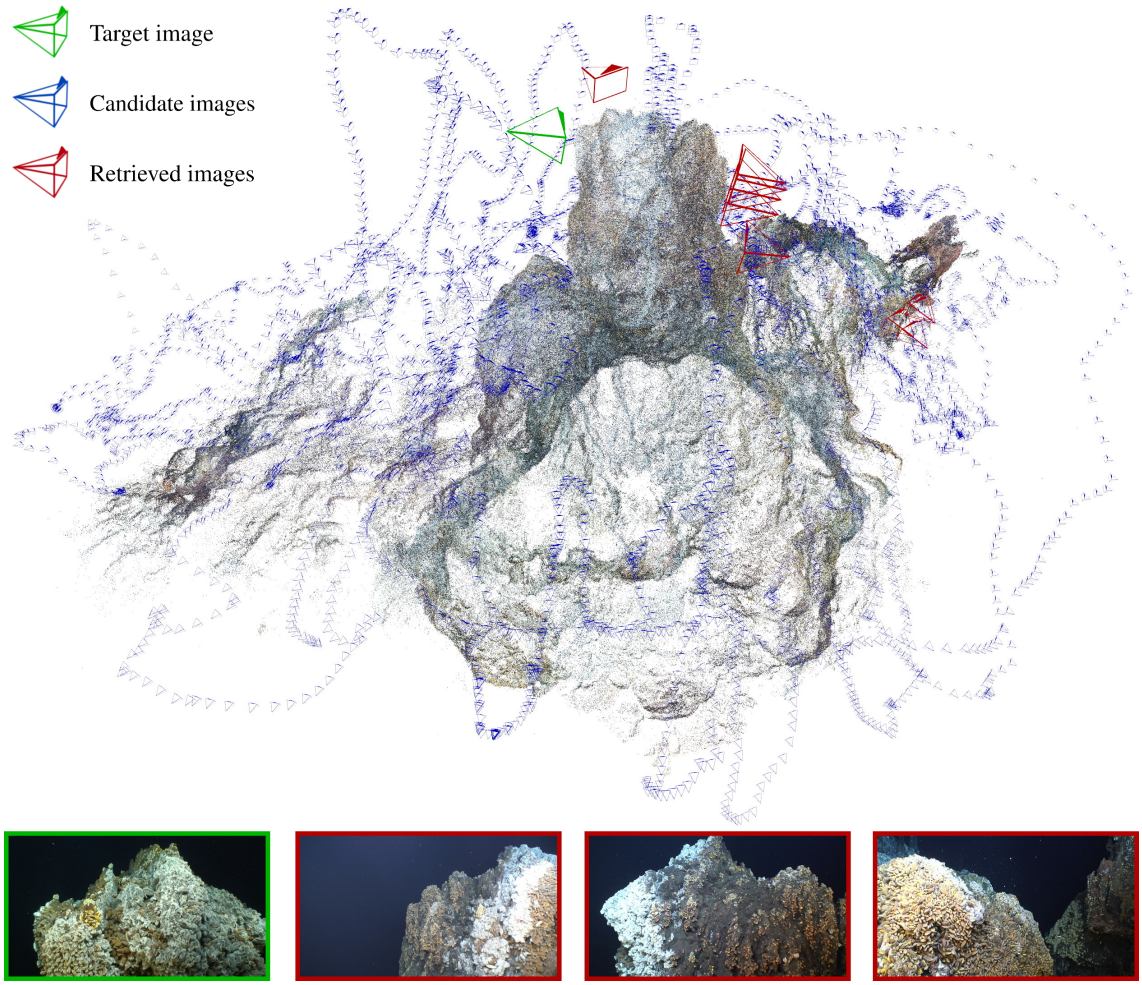


Figure 3.4: **Image retrieval** with NetVLAD (Arandjelovic et al., 2016). Our goal is to identify candidate images from the 2015 dive (in blue) that are similar to a target image from the 2016 visit (in green). We show in red the candidate images whose NetVLAD descriptors are closest to the descriptor of the target image.

Section 1.2, the navigation data are only consistent within each visit due to the error of the USBL, which can exhibit offsets of several meters between the frames of the different visits. As a result, this approach does not enable the pairing of images from different acquisitions.

Another solution is to use local or global image descriptors. For instance, we can compute local descriptors for each image and employ methods like BoW to pair images that have close features (Sivic and Zisserman, 2003). Alternatively, we can use methods that generate a single global descriptor for each image, such as NetVLAD (Arandjelovic et al., 2016), to pair images with similar global descriptors. However, as illustrated in Figures 3.3 and 3.4, while these approaches may yield satisfactory results for pairing images of the same visit year, they exhibit poor performance when pairing images of different dives. In Figure 3.3, we show the distance between query images and their top cross-years retrieved

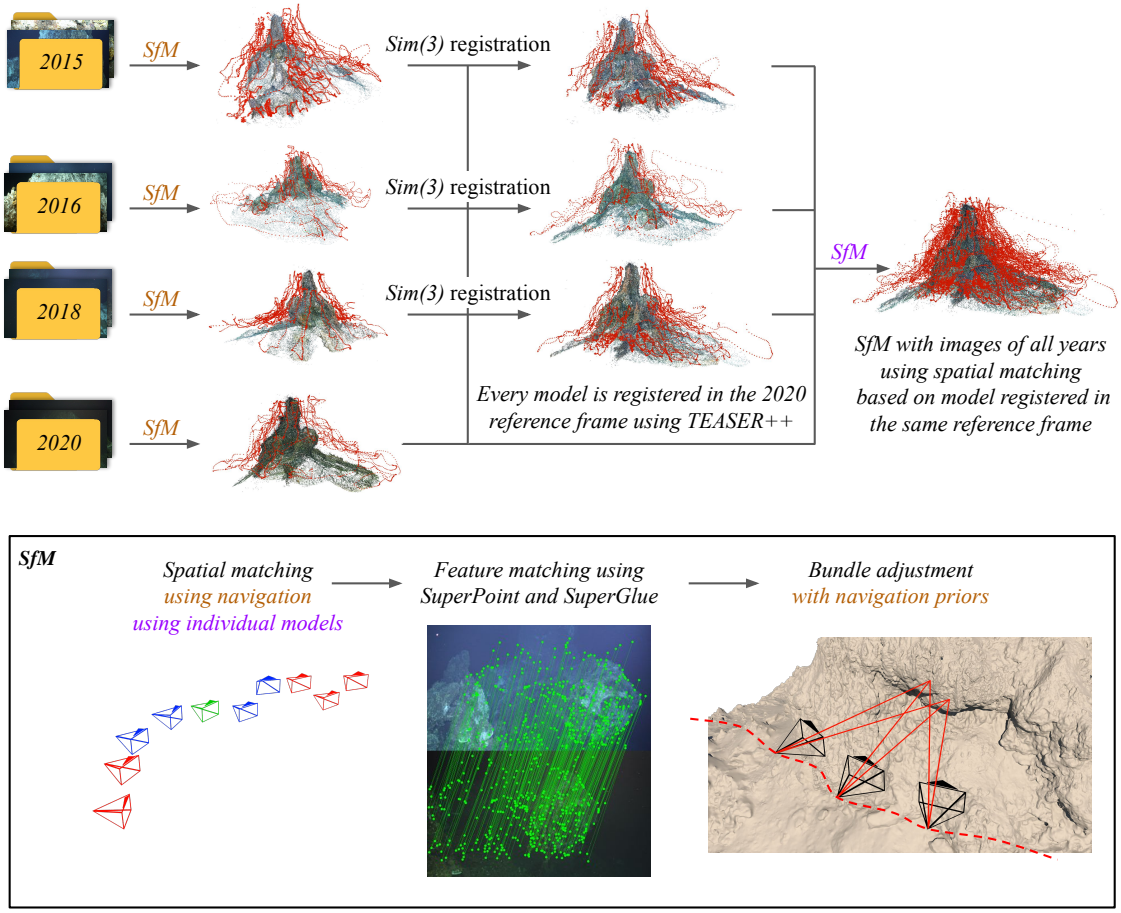


Figure 3.5: **Structure-from-Motion pipeline.** Initially, independent models are constructed for each year. These models are then aligned to a common reference frame, specifically the 2020 reference frame, using TEASER++ (Yang et al., 2021) and ICP (Zhou et al., 2018). Finally, a global model encompassing images of all years is generated using spatial matching based on the camera poses of individual models that now share a common reference frame.

images using NetVLAD. Unfortunately, most retrieved images are situated at a distance of more than five meters from their query image. There also appears to be a spatial disparity in retrieval performance, with regions on the vent border exhibiting lower performance compared to those at the vent's center. Nevertheless, images observing the chimney at the vent's center are rarely matched with images on a sub-metric scale. Figure 3.4 shows a query image and its retrieved images using NetVLAD. Although multiple candidate images could have been relevant matches, most of the retrieved images are located far from the target image. Similarly to Figure 3.3, there seems to be an ambiguity in determining which side of the top of the vent was actually observed.

To overcome this problem, we adopt the SfM pipeline outlined in Figure 3.5. We first build separate models for each year. For each individual model, we perform image retrieval based on the interpolated navigation data, specifically pair-

ing images that are within a proximity of 10 meters and 45 degrees. We then carry out feature matching, described in Section 3.4.2, followed by bundle adjustment, detailed in Section 3.4.3. These steps result in individual point clouds of the vent for each visit year. To bring these point clouds into a unified reference frame, we need to determine the transformations in $Sim(3)$ that align them. The $Sim(3)$ group represents a 6DoF pose, and adds an extra degree of freedom — the scale factor, denoted s. This scale factor allows us to account for changes in size or scale when describing transformations. In other words, we seek to find three transformations:

$$ {}^{B}\boldsymbol{T}_{A} = \begin{bmatrix} s\,{}^{B}\boldsymbol{R}_{A} & {}^{B}\boldsymbol{t}_{A} \\ \mathbf{0}_{1\times 3} & 1 \end{bmatrix} \in Sim(3), \tag{3.1} $$

such as the projection

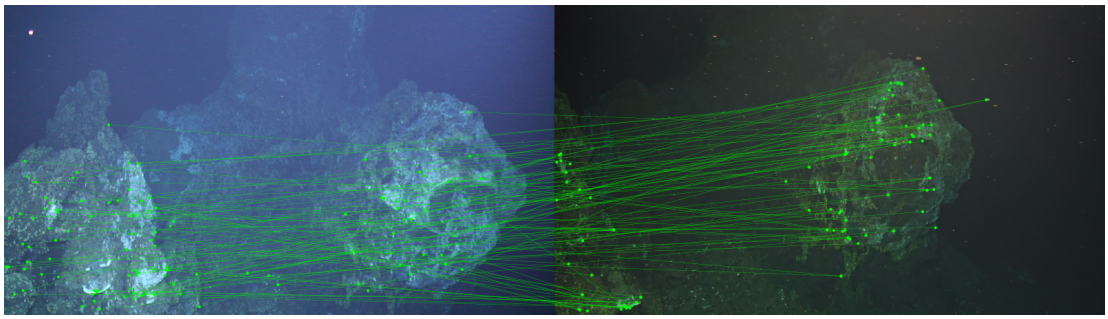
$$ {}^{B}\mathbf{X} = {}^{B}\boldsymbol{T}_{A} \odot {}^{A}\mathbf{X} \tag{3.2} $$

maps the 3D point ${}^{A}\mathbf{X}$ from frame A to frame B. Using TEASER++ (Yang et al., 2021), a robust 3D point cloud registration method similar to the ICP algorithm, we retrieve ${}^{2020}\boldsymbol{T}_{2015}$, ${}^{2020}\boldsymbol{T}_{2016}$ and ${}^{2020}\boldsymbol{T}_{2018}$, the respective transformations from 2015, 2016 and 2018 models to the 2020 reference frame. As suggested by the authors, we further refine these transformations using the ICP algorithm (Zhou et al., 2018). By applying their respective transformation to each model, we obtain an initial coarse estimation of the camera poses from each dive in the same reference frame. Using these camera poses, we once again pair images that are spatially close, and perform feature matching and bundle adjustment to generate a reference global model that encompasses images from all visit years. The final step involves retrieving the scale, orientation, and location of the unified model. To achieve this, we employ Umeyama's algorithm (Umeyama, 1991) to find the transformation ${}^{\text{nav}}\boldsymbol{T}_{\text{global}} \in Sim(3)$ that aligns the camera poses of the 2020 images in the global model with their interpolated navigation.

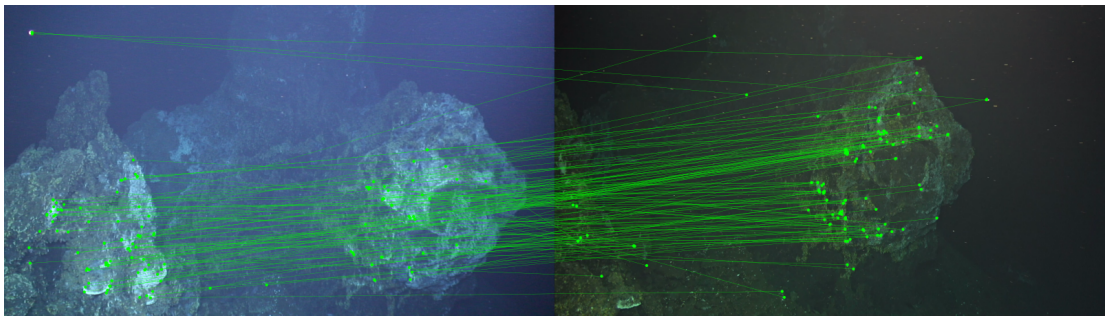
On a related topic, Appendix B develops the lie algebra to perform the $Sim(3)$ alignment between camera poses and their associated depth priors, similarly to Umeyama's algorithm, but using only vertical position priors. This development proves valuable in underwater applications, where complete camera position priors might not always be available, in contrast to depth measurements which require only a pressure sensor.

3.4.2 Image matching

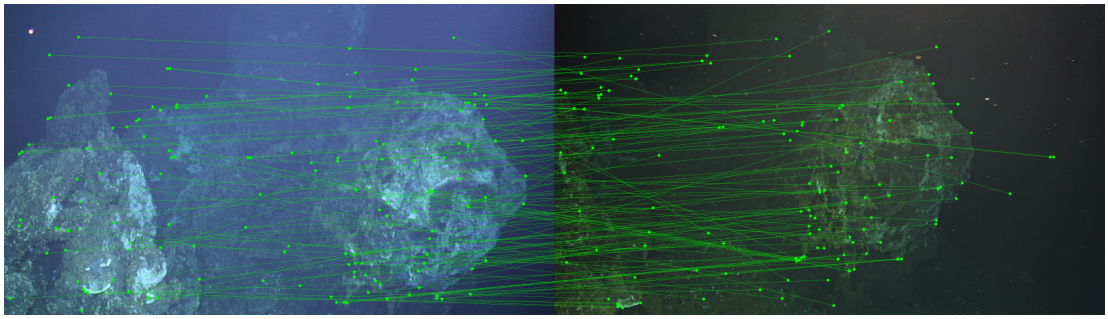
To ensure consistency in the estimated camera poses and scene geometry across different years during the construction of the SfM model, it is necessary to not only match images from the same visit but also match local features in images



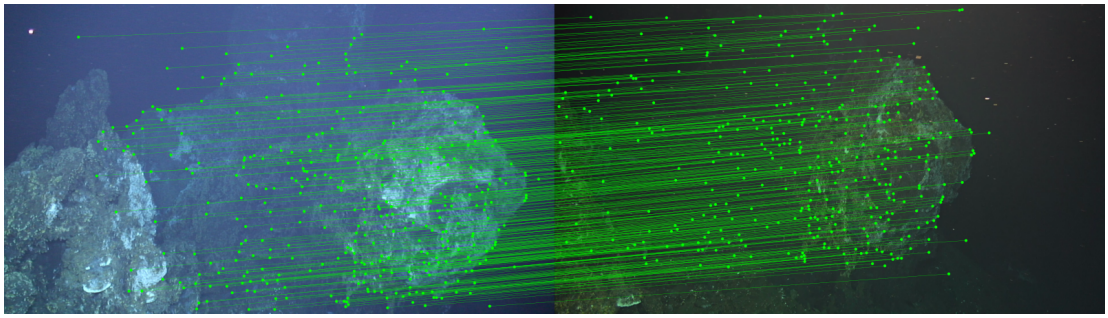
(a) Brute force matching using SIFT descriptors (Lowe, 1999).



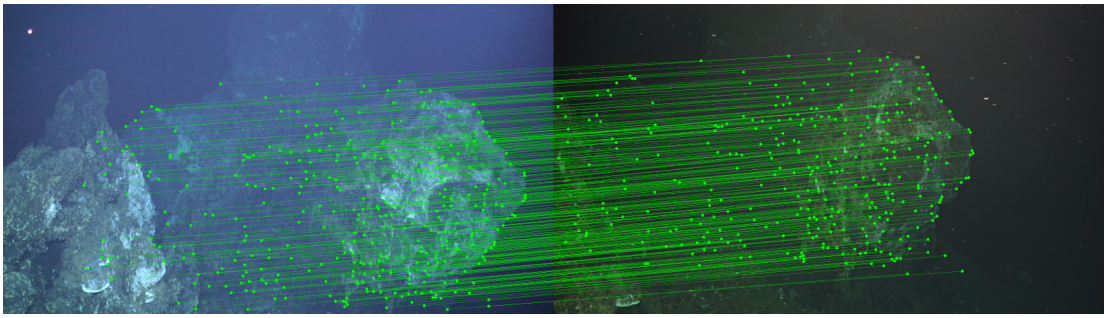
(b) Brute force matching using ORB descriptors (Rublee et al., 2011).



(c) Brute force matching using SuperPoint descriptors (DeTone et al., 2018).



(d) Matching with SuperGlue (Sarlin et al., 2020) using SuperPoint descriptors.



(e) Matching with LightGlue (Lindenberger et al., 2023) using SuperPoint descriptors.

Figure 3.6: **Feature matching** between cross-year images using different methods.

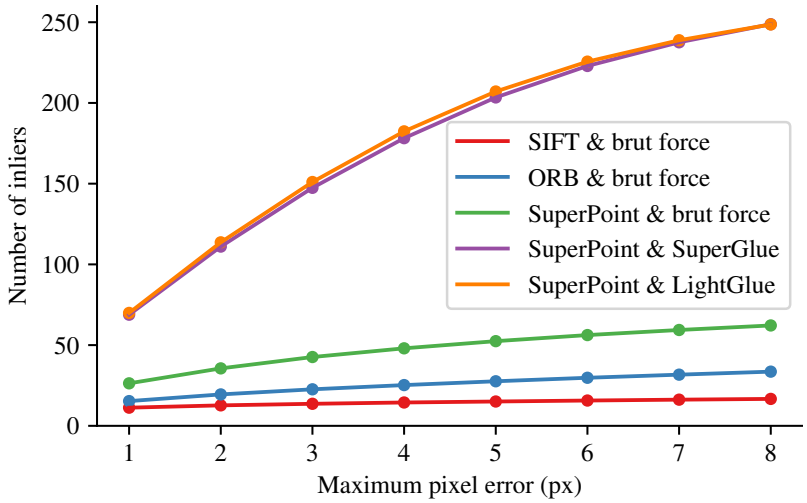


Figure 3.7: **Number of feature matching inliers** between images of different visit years using different matching methods. We first create a database of image pairs using the unified model. We pair every image of the 2020 dive with its spatially closest image amongst the 2015, 2016 and 2018 dives. We filter image pairs that are not within a radius of five meters and twenty degrees. We then perform feature matching on the retrieved image pairs. Using the camera calibration matrices, we filter the resulting matches by computing their essential matrix within a RANSAC scheme with a given pixel distance threshold. This figure reports the number of inliers resulting from this operation using different pixel thresholds.

from different visits. However, there is no guarantee that feature matching algorithms will generalize well to the significant changes present in deep-sea environments. Therefore, in order to identify the most suitable feature matching algorithm for our application, it is essential to conduct benchmarking experiments using various algorithms. Benchmarking feature matching algorithms relies on quantitative metrics computed from ground truth data. For instance, the HPatches dataset (Balntas et al., 2017) provides image pairs of planar scenes along with their corresponding homography. This dataset enables the calculation of various standard metrics, such as the precision and recall of estimated matches compared to ground truth matches within a reprojection error threshold of three pixels (Sarlin et al., 2020; Lindenberger et al., 2023). Unfortunately, we do not have access to similar datasets with ground truth information to benchmark these feature matching algorithms in an underwater scenario.

To alleviate this concern, we rely on a practical quantitative and qualitative survey of different feature matching methods, encompassing different descriptors and feature matchers. We evaluate three different local descriptors commonly used in SfM and visual localization pipelines: SIFT (Lowe, 1999), ORB (Rublee et al., 2011) and SuperPoint (DeTone et al., 2018), with brute force matching. We also investigate the performance SuperPoint in conjunction with two

feature matching networks: SuperGlue (Sarlin et al., 2020) and LightGlue (Lindenberger et al., 2023). Figure 3.6 provides examples of these feature matching methods on two images extracted from the 2015 and 2020 dives. Figure 3.7 shows the number of filtered matches obtained using these different feature matching methods on a set of image pairs from different visits. Our findings indicate that handcrafted features like SIFT and ORB struggle to generalize effectively in the face of cross-years changes. In contrast, SuperPoint's learned descriptors demonstrate improved generalization capabilities, albeit with noticeable levels of noise in the feature matching results. A real improvement is observed by combining these learned descriptors with feature matching neural networks, such as SuperGlue and the more recent LightGlue approach.

Since LightGlue was not available at the time of creating the dataset, the pipeline described in Section 3.4.1 and illustrated by Figure 3.5 uses SuperPoint & SuperGlue for matching local features in image pairs.

3.4.3 Bundle adjustment with position priors

Following the feature matching step, we obtain 2D keypoint pairs measurements. In the SfM pipeline, the camera poses and a 3D point cloud of the scene are retrieved through bundle adjustment by maximizing the likelihood of observing these measurements (Hartley and Zisserman, 2003). This section details the bundle adjustment problem and provides an alternative cost function to incorporate known position priors $\bar{\boldsymbol{t}}_i \in \mathbb{R}^3$ derived from the navigation data in the optimization procedure.

We note C_i the image with camera pose $[\boldsymbol{R}_i, \boldsymbol{t}_i]$ and camera matrix $\boldsymbol{K}_i$. We note $\overline{\mathbf{x}}_{i,j}$ the measured 2D keypoint coordinates resulting from the observation of 3D point $\mathbf{X}_j$ in image C_i. Let

$$\mathbf{x}_{i,j} = \pi\left(\boldsymbol{K}_i(\boldsymbol{R}_i\mathbf{X}_j + \boldsymbol{t}_i)\right) \tag{3.3}$$

be the projection of 3D point $\mathbf{X}_j$ in C_i image plane. Given a set of observations $\overline{\mathbf{x}}_{i,j}$, we aim to estimate the set of camera poses, intrinsics and 3D scene coordinates:

$$\theta = \{\boldsymbol{R},\, \boldsymbol{t},\, \boldsymbol{K},\, \mathbf{X}\}. \tag{3.4}$$

Traditionally, in bundle adjustment, we make the assumption that these observations are measured with a zero-mean Gaussian noise:

$$\mathbf{x}_{i,j} \sim \mathcal{N}(\overline{\mathbf{x}}_{i,j},\, \boldsymbol{\Sigma}^{\mathrm{x}}). \tag{3.5}$$

An estimation of the parameters is then obtained through the maximum likelihood estimator:

$$\hat{\theta}_{\mathrm{MLE}} = \arg\max_{\theta} \prod_i \prod_j p(\mathbf{x}_{i,j} \mid \theta), \tag{3.6}$$

where $p(\mathbf{x}_{i,j} \mid \theta)$ is the probability of observing $\mathbf{x}_{i,j}$ given θ parameters. In the context of our specific scenario, we have access to navigation data that provide valuable prior information regarding the camera poses. More specifically, we leverage the position measurements of the vehicle $\bar{\boldsymbol{t}}_i$ and their associated uncertainties $\boldsymbol{\Sigma}_i^t$. We make the assumption that these measurements were acquired with a zero-mean Gaussian noise:

$$\boldsymbol{t}_i \sim \mathcal{N}(\bar{\boldsymbol{t}}_i,\, \boldsymbol{\Sigma}_i^t). \tag{3.7}$$

To leverage this new source of information, we formulate the bundle adjustment problem using the maximum a posteriori estimator with $p(\theta) = p(\boldsymbol{t}_i)$, since we only have prior information for the camera's position:

$$\hat{\theta}_{\text{MAP}} = \arg\max_{\theta} \prod_i \prod_j \Big[p(\mathbf{x}_{i,j} \mid \theta)\Big] \prod_i \Big[p(\boldsymbol{t}_i)\Big] \tag{3.8}$$

with

$$p(\mathbf{x}_{i,j}|\theta) = \frac{1}{2\pi\sqrt{\det(\boldsymbol{\Sigma}^{\text{x}})}} \exp\left(-\frac{1}{2}(\mathbf{x}_{i,j} - \overline{\mathbf{x}}_{i,j})^T\,(\boldsymbol{\Sigma}^{\text{x}})^{-1}(\mathbf{x}_{i,j} - \overline{\mathbf{x}}_{i,j})\right) \tag{3.9}$$

and

$$p(\boldsymbol{t}_i) = \frac{1}{\sqrt{8\pi^3\det(\boldsymbol{\Sigma}_i^t)}} \exp\left(-\frac{1}{2}(\boldsymbol{t}_i - \bar{\boldsymbol{t}}_i)^T\,(\boldsymbol{\Sigma}_i^t)^{-1}(\boldsymbol{t}_i - \bar{\boldsymbol{t}}_i)\right). \tag{3.10}$$

This is equivalent to minimizing the negative logarithm:

$$\hat{\theta}_{\text{MAP}} = \arg\min_{\theta} -\sum_i \sum_j \Big[\log(p(\mathbf{x}_{i,j} \mid \theta))\Big] - \sum_i \Big[\log(p(\boldsymbol{t}_i))\Big] \tag{3.11}$$

with

$$\log(p(\mathbf{x}_{i,j}|\theta)) = -\frac{1}{2}(\mathbf{x}_{i,j} - \overline{\mathbf{x}}_{i,j})^T\,(\boldsymbol{\Sigma}^{\text{x}})^{-1}(\mathbf{x}_{i,j} - \overline{\mathbf{x}}_{i,j}) - \log\left(2\pi\sqrt{\det(\boldsymbol{\Sigma}^{\text{x}})}\right) \tag{3.12}$$

and

$$\log(p(\boldsymbol{t}_i)) = -\frac{1}{2}(\boldsymbol{t}_i - \bar{\boldsymbol{t}}_i)^T\,(\boldsymbol{\Sigma}_i^t)^{-1}(\boldsymbol{t}_i - \bar{\boldsymbol{t}}_i) - \log\left(\sqrt{8\pi^3\det(\boldsymbol{\Sigma}_i^t)}\right). \tag{3.13}$$

By eliminating the constant terms that do not affect the minimization, we can rewrite Eq. (3.11) as:

$$\begin{aligned}\hat{\theta}_{\text{MAP}} = \arg\min_{\theta} \sum_i \sum_j \Big[&(\mathbf{x}_{i,j} - \overline{\mathbf{x}}_{i,j})^T\,(\boldsymbol{\Sigma}^{\text{x}})^{-1}(\mathbf{x}_{i,j} - \overline{\mathbf{x}}_{i,j})\Big] \\ &+ \sum_i \Big[(\boldsymbol{t}_i - \bar{\boldsymbol{t}}_i)^T\,(\boldsymbol{\Sigma}_i^t)^{-1}(\boldsymbol{t}_i - \bar{\boldsymbol{t}}_i)\Big].\end{aligned} \tag{3.14}$$

Equation (3.14) describes the cost function minimized during bundle adjustment that incorporates known position estimates as priors during the optimization.

In Section 3.4.1, we outlined that we use bundle adjustment to build the individual models and the unified model. For individual models, we make use of the available navigation data and perform bundle adjustment using Eq. (3.14). As for the unified model, considering that navigation data is inconsistent across different years due to the USBL bias, we employ the traditional bundle adjustment cost function described by Eq. (3.6).

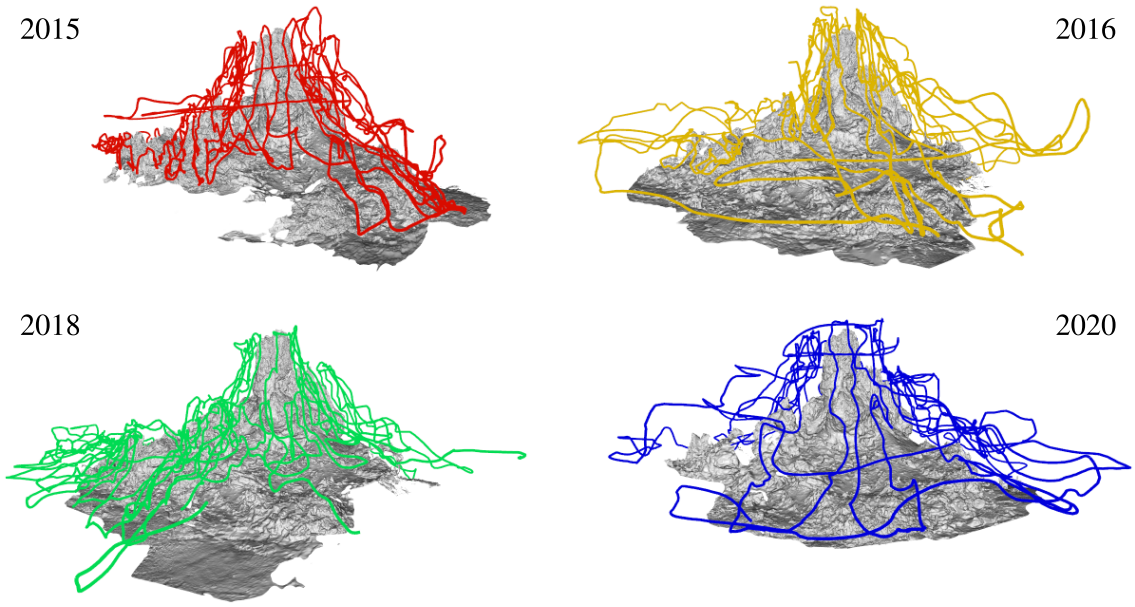


Figure 3.8: **Trajectories** followed by the ROV during the four different visits. The figure displays unified camera positions on the aligned individual models.

Model	Nb. of images	Nb. of 3D points	Mean track length	Mean obs. per image	Mean reproj. error
2015	4,914	525,522	8.48	906.4	1.35 px
2016	3,699	520,320	5.85	823.5	1.32 px
2018	5,493	618,421	7.09	798.1	1.31 px
2020	3,976	464,331	8.35	975.5	1.33 px
Global	18,082	1,971,726	8.24	898.7	1.39 px

Table 3.2: **Reconstruction statistics.** For each model, we report the number of registered images, the number of triangulated 3D points, the mean track length (number of images in which a 3D point is observed), the mean number of 2D observations per image as well as the mean reprojection error in pixels.

3.4.4 Model statistics

Figure 3.8 illustrates the estimated camera poses in the unified model. To assess the reliability of the proposed ground truth, Table 3.2 presents statistics on the obtained reconstructions. Additionally, Table 3.3 reports the percentage of 3D points matched between each year in the global model. While the majority of 3D points observations are confined within the same year, a significant proportion of them were successfully matched across different years, ensuring consistency in the scene geometry and camera poses between the various visits.

Observation year	2015	2016	2018	2020
2015	65.9%	10.7%	15.2%	8.12%
2016	20.3%	50.9%	16.2%	12.6%
2018	16.1%	9.05%	63.1%	11.8%
2020	9.79%	8.00%	13.5%	68.8%

Table 3.3: **Percentage of 3D points triangulated using cross-years observations.** For each row, given all the 3D observations of the specified year, we report the percentage of these observations that were triangulated using 2D observations of other years. Consequently, rows add up to 100%.

3.5 Characterizing changes across years

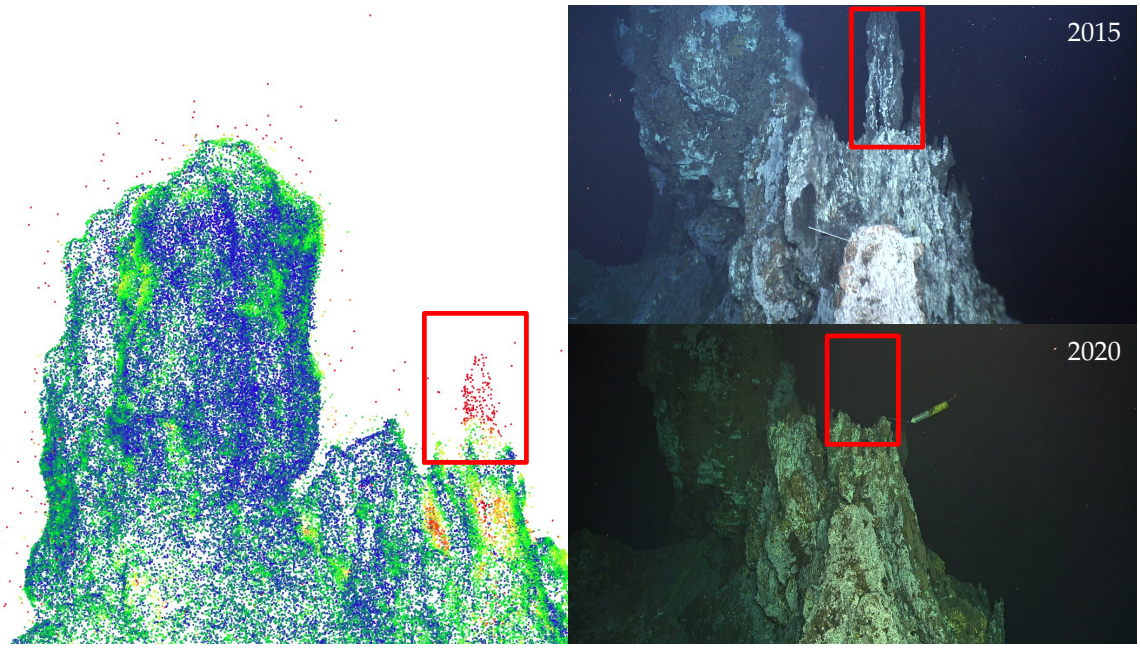


Figure 3.9: **Illustration of a topological modification.** The left image shows a point cloud distance between 2015 and 2020 models. We notice a piece from 2015 missing in 2020. This modification is visible in the images on the right.

This dataset presents significant appearance changes across visits, posing substantial challenges for the visual-based localization task. These alterations encompass various aspects, such as topographic variations, environmental shifts, and modifications in the ROV's equipment. During the observation period, several changes were detected and measured, indicating alterations in the local geomorphology of the edifice over the years (Van Audenhaege et al., 2023). Chimney collapse, outcrop/boulder detachment or slide resulted in a loss of material. Meanwhile, areas where the vent is active grew through mineral accretion creating new outcrops, flanges and spires. Material build-up was twice as important

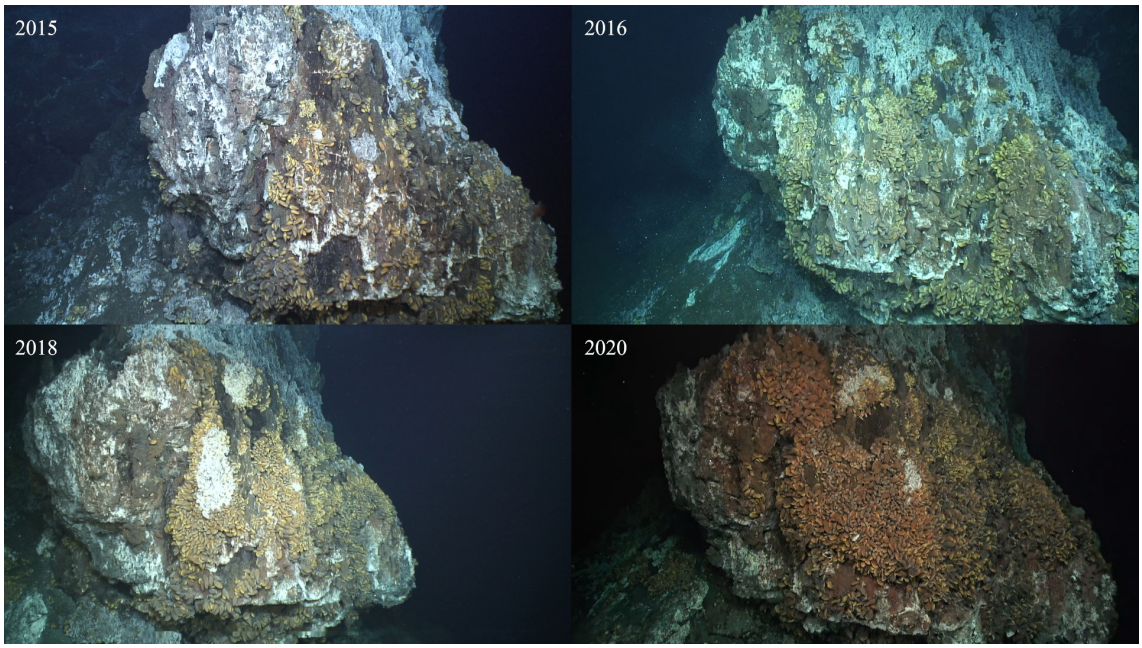
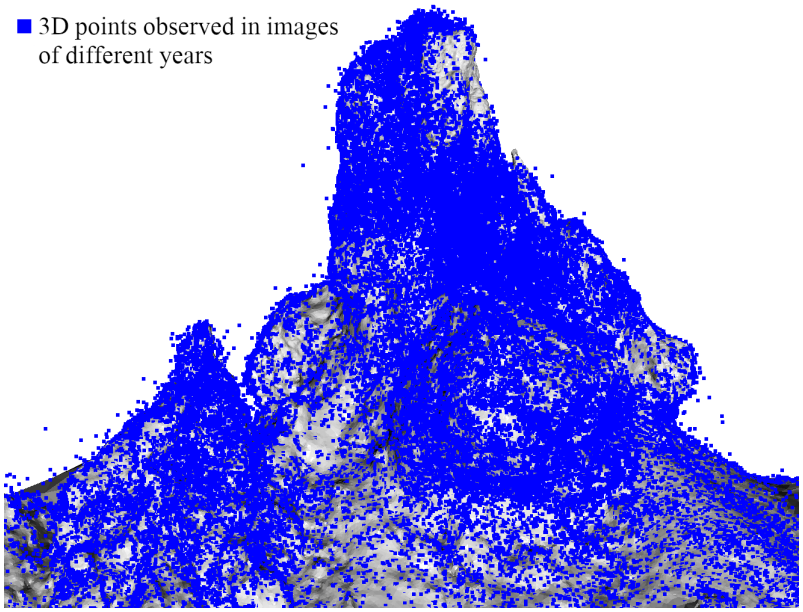


Figure 3.10: **Evolution of the south-east façade of the vent.** A growth in the mussels' population significantly alters the visual aspect of the scene, making it difficult to match specific 2D points.

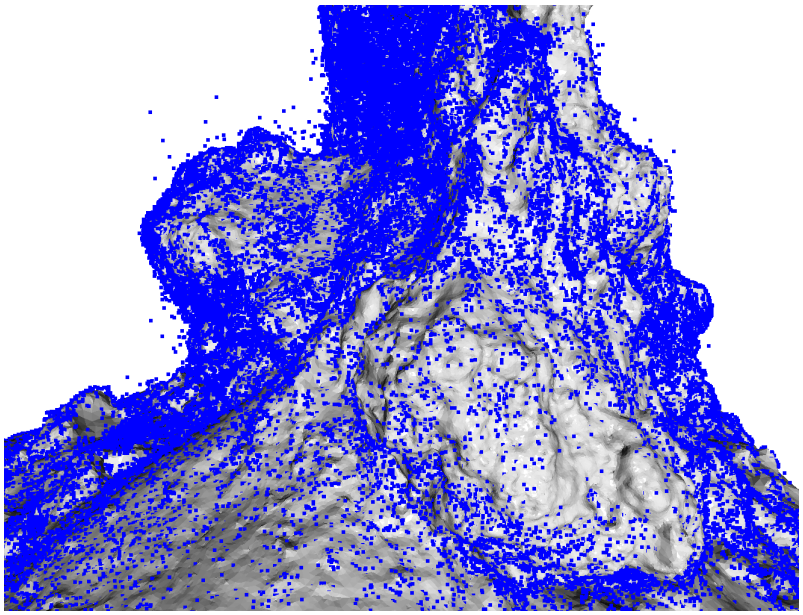
as the loss, suggesting that the volume of the edifice's volume is increasing over time. While these changes can be locally drastic and affect the registration of 3D models over the years, they represent only 5% of the total surface and are localized in areas of active venting. Furthermore, variations in hydrothermal activity result in distinct mineralization processes, leading to color changes in deposits depending on the temperature and chemical composition of the fluid.

Figure 3.9 reveals a modification in the topography of the scene. A chimney visible during the 2015 dive is missing in 2020. Additionally, a temperature sensor, which was not observed in 2015, was deployed in the vicinity in 2020.

Biological changes were more important and mainly localized in areas of topographic changes. They result from mussel populations that grow and migrate to colonize newly created habitats (143.97 m^2 from 2015 to 2020) (Audenhaege et al., 2022). Moreover, mussels are dynamically reoriented on a daily basis. Over the period from 2015 to 2020, there was an overall disappearance of white microbial mats across the entire edifice (-72.85 m^2). Although these changes do not affect the general topography of the structure, they strongly modify the color and texture of the model. Figure 3.10 illustrates how the growth in mussels population over the years has altered both the topography of the scene and the colors of the south-east façade of the vent. Due to these organic modifications, matching 3D points between different years on the chimney are scarce, making it challenging to match specific 2D points. The impact of these biological phenomena on the global 3D reconstruction are illustrated in Figure 3.11. While most of the vent's



(a) West façade.

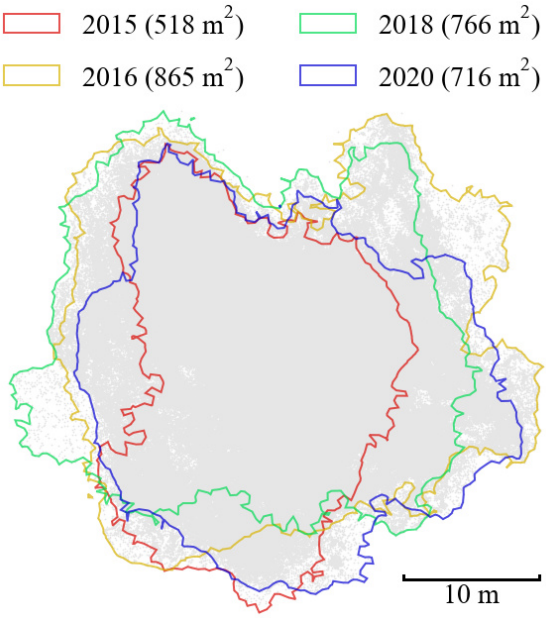


(b) South-east façade.

Figure 3.11: **Distribution of 3D points** that are triangulated between images of different years on the Eiffel Tower edifice. 3D points resulting from cross-years triangulation are more scarce on the south-east façade due to biological changes.

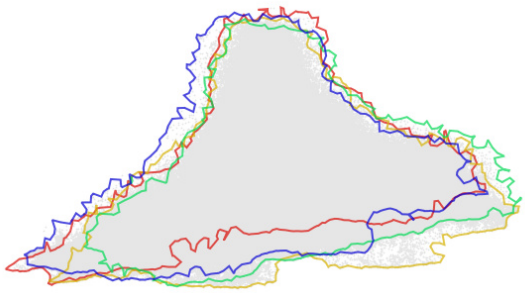
surface is covered by 3D points triangulated using cross-years observations, this specific area suffers from this source of variability, and the model mostly relies on matches between images of the same year.

The vehicle explored uneven regions over different years, with the 2015 dive covering the least ground compared to other years. This is depicted in Figure 3.12, which showcases the area explored by the ROV for each year.



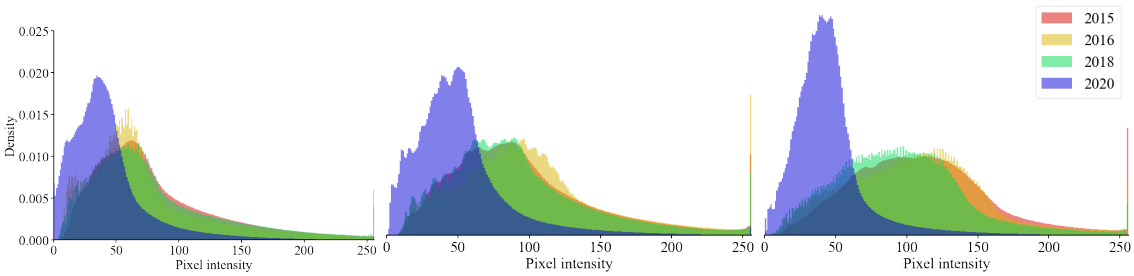
(a) East-North View.

(b) East-Up View.



(c) North-Up View.

Figure 3.12: **Area covered by the ROV** during the different dives.



(a) Red channel. (b) Green channel. (c) Blue channel.

Figure 3.13: **Comparison of pixel intensity histograms** for each year on each color channel.

In Figure 3.13, we compare the histograms of pixel intensities in images from each year. The red channel generally exhibits lower pixel intensities compared

to the green and blue channels. This is easily explained by the wavelength-dependent light attenuation caused by the water medium (Akkaynak and Treibitz, 2019; Berman et al., 2021). Additionally, a discernible shift in pixel intensity is noticeable for the 2020 dive, which can likely be attributed to the camera change.

3.6 Conclusion

In this chapter, we have developed a novel dataset specifically designed for evaluating long-term visual localization algorithms in deep-sea environments. This dataset contains 18,082 images captured during four distinct visits to the Eiffel Tower hydrothermal vent spanning from 2015 to 2020. Throughout the construction of this dataset, we encountered various challenges closely related to the visual localization problem.

We investigated the difficulties associated with image retrieval methods, particularly in pairing underwater images with significant appearance changes. We conducted a comprehensive benchmark of feature matching algorithms to identify the most robust ones capable of handling environmental variations effectively. We introduced a maximum a posteriori formulation of the bundle adjustment problem that incorporates known camera pose estimates as priors. This formulation not only enables to retrieve the model's scale and orientation but also guides the convergence of the registered images during the optimization.

Following our findings, we built a unified model that embeds images from all visits. We then provided reconstruction metrics to validate its applicability. Lastly, we conducted a comprehensive survey of the changes that have occurred over the years, aiming to gain a deeper understanding of the different sources of variability encountered in this environment. This analysis represents a key step in identifying potential factors that may impair the performance of visual localization algorithms in this challenging underwater setting.

The work presented in this chapter led to a publication in *The International Journal of Robotics Research* (Boittiaux et al., 2023b).

Chapter 4

Underwater image color restoration

Contents

4.1 Introduction

In Section 2.1, we have seen that due to the alteration of light propagation in the water medium, underwater images suffer from low contrast and distorted colors, mainly due to absorption and scattering phenomena. Consequently, when

building an underwater dataset for visual localization in Section 3.4, we identified challenges in every step of the SfM pipeline. To better understand and alleviate the issues encountered when applying computer vision algorithms on underwater images, this chapter introduces two new methods that aim to restore the colors of underwater images as if they were acquired at the surface, without the effect of water on light propagation. Because absorption and scattering phenomena heavily depend on the position of the camera relative to the scene, the presented methods take as input 3D information about the scene.

The contributions of this chapter are the following:

- Building upon the success of the *Sea-thru* method presented in Section 2.1.4, we propose a new optimization scheme for estimating the parameters of the underwater image formation model present in Eq. (2.10) from a single image and its corresponding distance map. It relies on a single assumption, *i.e.*, pixel intensities are normally distributed within each color channel of the restored image.

- We then introduce SUCRe (for Structured Underwater Color Restoration), a multi-view method that proposes to make full use of the 3D information given by SfM models. It simultaneously estimates the parameters of the underwater image formation model described in Eq. (2.10) alongside the restored image by tracking points in multiple images to retrieve their intensities at different distances to the scene. We demonstrate that it alleviates some of the main issues encountered when using a single image.

- We validate experimentally the developed approaches on both synthetic and real-world datasets in natural light and deep-sea scenarios. We perform extensive objective quantitative evaluation on two datasets containing reference ground truth data: synthesized underwater images (Zwilgmeyer et al., 2021) and color charts captured under water (Akkaynak and Treibitz, 2019). The applicability and ecological validity of our method is confirmed with qualitative analysis on images from two deep dive surveys: the Eiffel Tower edifice presented in Chapter 3 and a similar private dataset visiting a submarine wreck.

This chapter is organized as follows. First, Section 4.2 introduces Gaussian *Sea-thru*, which is an alternative optimization to the *Sea-thru* method. Then, Section 4.3 proposes a novel method, called SUCRe, for multi-view underwater color restoration. Finally, Section 4.4 present experiments to illustrate the efficiency of the proposed methods.

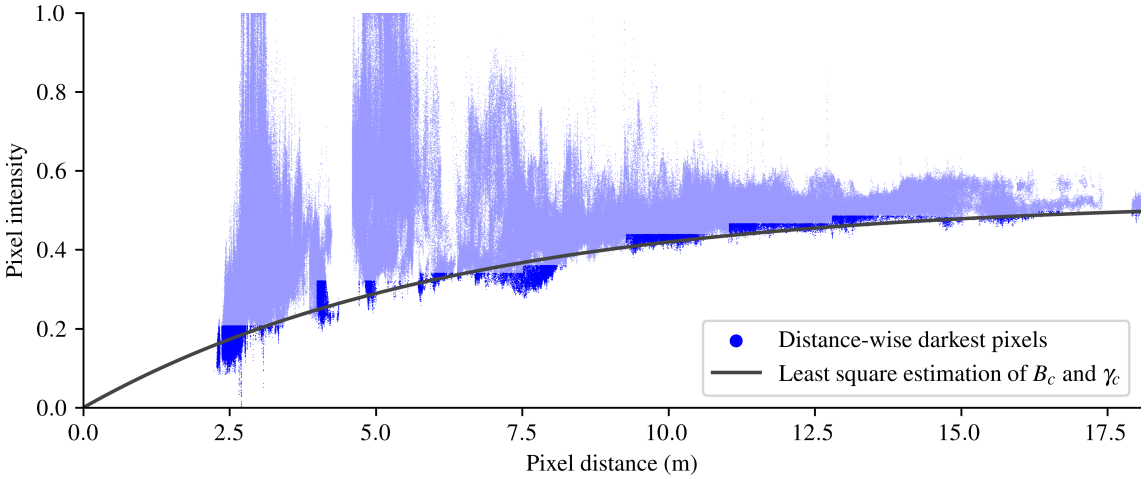


Figure 4.1: **Distance-dependent dark channel prior** on the blue channel.

4.2 Gaussian prior for underwater color restoration

4.2.1 Motivation

As we can recall from Section 2.1.4, the *Sea-thru* method aims at inverting the following underwater image formation model in order to retrieve an unattenuated version of the image:

$$\boldsymbol{I}_{c,p} = \boldsymbol{J}_{c,p} e^{-\beta_c \boldsymbol{z}_p} + B_c(1 - e^{-\gamma_c \boldsymbol{z}_p}), \tag{4.1}$$

with $\boldsymbol{I}_{c,p}$ the acquired underwater image, $\boldsymbol{J}_{c,p}$ the restored image without the effects of water, $\boldsymbol{z}_p$ the distance of pixel p to the scene, B_c the veiling light, β_c the absorption coefficient and γ_c the backscatter coefficient. To achieve this, *Sea-thru* requires only a single image and its distance map. Still in Section 2.1.4, we have seen that inverting this model from a single image and its corresponding distance map is an underdetermined problem.

To cope with this issue, *Sea-thru* relies on two additional hypotheses to constrain the problem. First, it estimates both the veiling light B_c and the backscatter coefficient γ_c using a distance-dependent alternative of the dark channel prior (He et al., 2010) illustrated in Figure 4.1. Pixel distances are split into 10 evenly spaced parts. For each of these parts, the authors select the darkest pixels, in the bottom 1% of pixel intensities. They make the assumption that these pixels are observations of dark scene elements. This means that, for these pixels, $\boldsymbol{J}_{c,p} \to 0$, and hence $\boldsymbol{I}_{c,p} \to B_c(1 - e^{-\gamma_c \boldsymbol{z}_p})$. In this setting, the authors estimate both B_c and γ_c in a least squares manner. They then rely on LSAC (Ebner and Hansen, 2013), a costly illuminant map estimation algorithm, to retrieve β_c. As a final processing step, the authors make use of the Gray World Hypothesis to estimate the white point of the image and perform a white balance. This hypothesis assumes that the average color of the observed scene should be gray.

4.2.2 Method

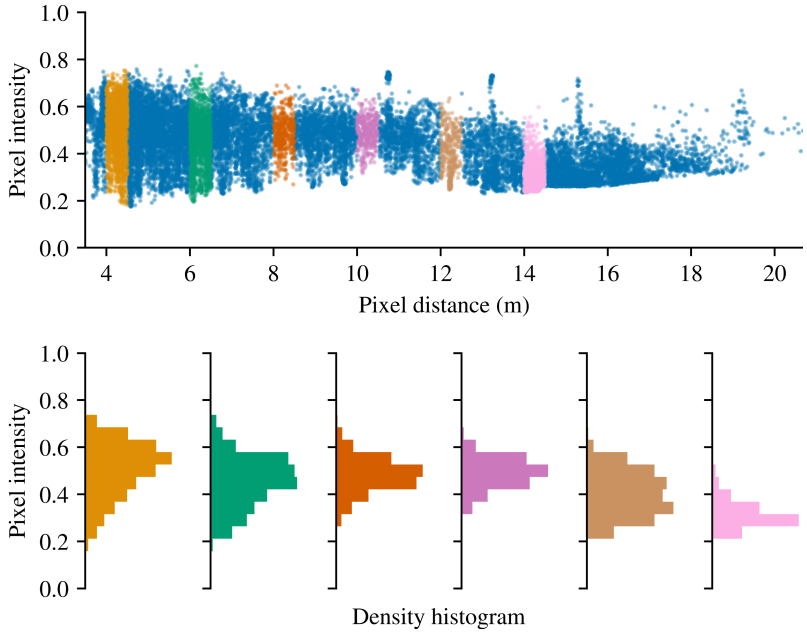


Figure 4.2: **Pixel intensities appear to follow a normal distribution** whose mean and standard deviation depend on the distance.

Since *Sea-thru* uses the Gray World Hypothesis to estimate the white point of their restored image, we might as well rely on a similar assumption from the start. Looking at Figure 4.2, we can notice that pixel intensities of underwater images appear to follow a normal distribution whose mean and standard deviation depend on their viewing distance. Encouraged by this observation and leveraging the assumption made by the Gray World Hypothesis, we introduce an alternative optimization method to *Sea-thru* that relies on the single assumption that pixel intensities are normally distributed within each channel of the restored image, *i.e.*, we use a Gaussian prior over each channel of the restored image:

$$\boldsymbol{J}_{c,p} \sim \mathcal{N}(\mu_c, \sigma_c^2), \tag{4.2}$$

where μ_c and σ_c are the channel-wise mean and standard deviation of the restored image pixel intensities. By shifting and scaling the parameters of Eq. (4.2) according to Eq. (4.1), we can deduce that pixel intensities of the acquired image also follow a normal distribution:

$$\boldsymbol{I}_{c,p} \sim \mathcal{N}(\boldsymbol{m}_{c,p}, \boldsymbol{s}_{c,p}^2), \tag{4.3}$$

where

$$\boldsymbol{m}_{c,p} = \mu_c e^{-\beta_c \boldsymbol{z}_p} + B_c(1 - e^{-\gamma_c \boldsymbol{z}_p}) \tag{4.4}$$

and

$$\boldsymbol{s}_{c,p} = \sigma_c e^{-\beta_c \boldsymbol{z}_p} \tag{4.5}$$

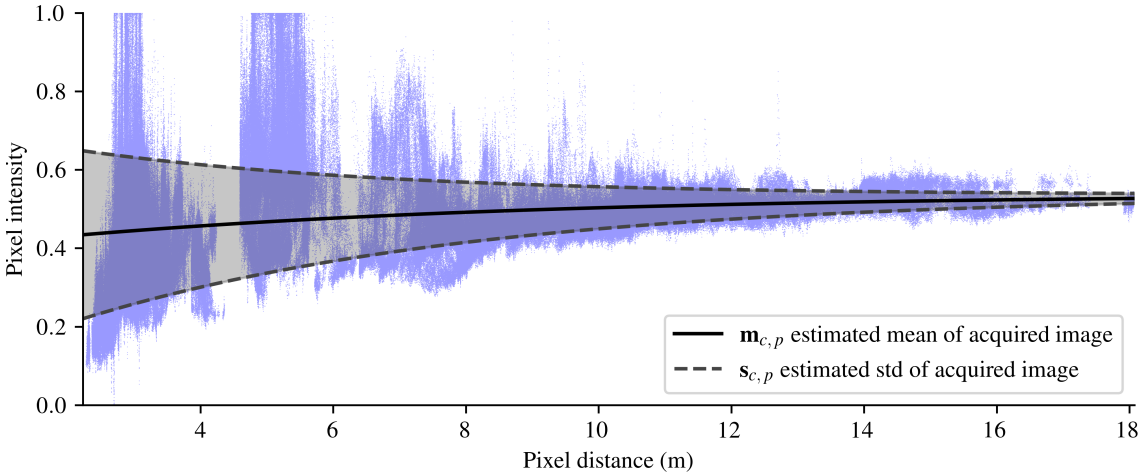


Figure 4.3: **Underwater image formation model parameters estimation using Gaussian *Sea-thru*.** We assume that pixel intensities are normally distributed in the restored image. From Eq. (2.10), we deduce that pixel intensities of the acquired image also follow a normal distribution with mean $\boldsymbol{m}_{c,p}$ and standard deviation $\boldsymbol{s}_{c,p}$ that depend on the pixels distances to the scene. We illustrate the estimated parameters on a deep-sea image and its distance map.

are respectively the channel-wise mean and standard deviation of pixel intensities of the acquired underwater image. Additional information regarding this calculation is available in Appendix C. Following the assumption made in Eq. (4.2), Eqs. (4.3) to (4.5) describe that pixel intensities in the acquired image also follow a normal distribution, whose parameters depend on the distance of the pixel to the scene, as illustrated by Fig. 4.3.

We then retrieve all the parameters of the underwater image formation model in a single optimization procedure using the Maximum Likelihood estimator. From Eq. (4.3) and the normal distribution probability density function, we can express the likelihood of observing $\boldsymbol{I}_c$:

$$\hat{B}_c, \hat{\beta}_c, \hat{\gamma}_c = \underset{B_c,\beta_c,\gamma_c}{\arg\max} \prod_p \left[\frac{1}{\boldsymbol{s}_{c,p}\sqrt{2\pi}} \exp\left(-\frac{(\boldsymbol{I}_{c,p} - \boldsymbol{m}_{c,p})^2}{2\boldsymbol{s}_{c,p}^2} \right) \right]. \tag{4.6}$$

This is equivalent to estimating the negative logarithm of Eq. (4.6):

$$\hat{B}_c, \hat{\beta}_c, \hat{\gamma}_c = \underset{B_c,\beta_c,\gamma_c}{\arg\min} \sum_p \left[\log(\boldsymbol{s}_{c,p}\sqrt{2\pi}) + \frac{(\boldsymbol{I}_{c,p} - \boldsymbol{m}_{c,p})^2}{2\boldsymbol{s}_{c,p}^2} \right]. \tag{4.7}$$

An estimation of the restored image is then retrieved using Eq. (4.1):

$$\hat{\boldsymbol{J}}_{c,p} = \left(\boldsymbol{I}_{c,p} - \hat{B}_c(1 - e^{-\hat{\gamma}_c \boldsymbol{z}_p}) \right) e^{\hat{\beta}_c \boldsymbol{z}_p}. \tag{4.8}$$

In summary, we have derived an estimation procedure to estimate the parameters of the underwater image formation model described by Eq. (4.1) based on

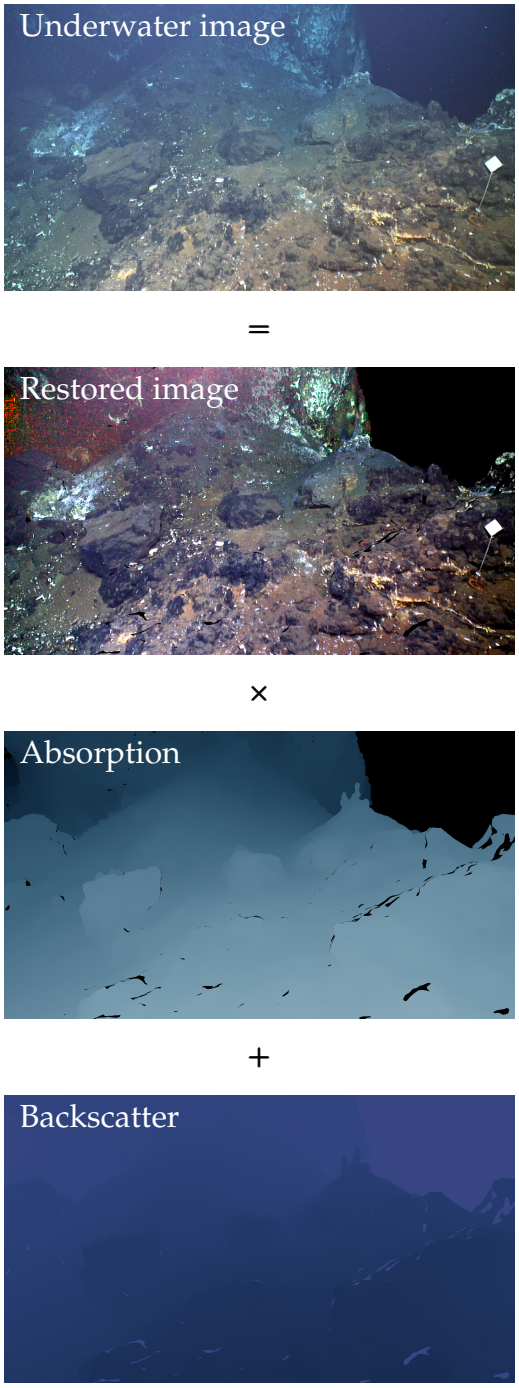


Figure 4.4: **Estimated absorption and backscatter** using Gaussian *Sea-thru*.

the single assumption that pixel intensities follow a Gaussian distribution in the restored image. This procedure involves fewer assumptions when compared to the *Sea-thru* method. It is also simpler to implement as it only relies on the minimization of a scalar function. In Figure 4.4, we illustrate graphically the image formation model of Eq. (4.1) using absorption and backscatter parameters estimated with Gaussian *Sea-thru*. To conclude, in simple terms, Gaussian *Sea-thru* stretches the pixel intensity histogram illustrated in Figure 4.3 in a distance-wise manner based on the image formation model described by Eq. (4.1).

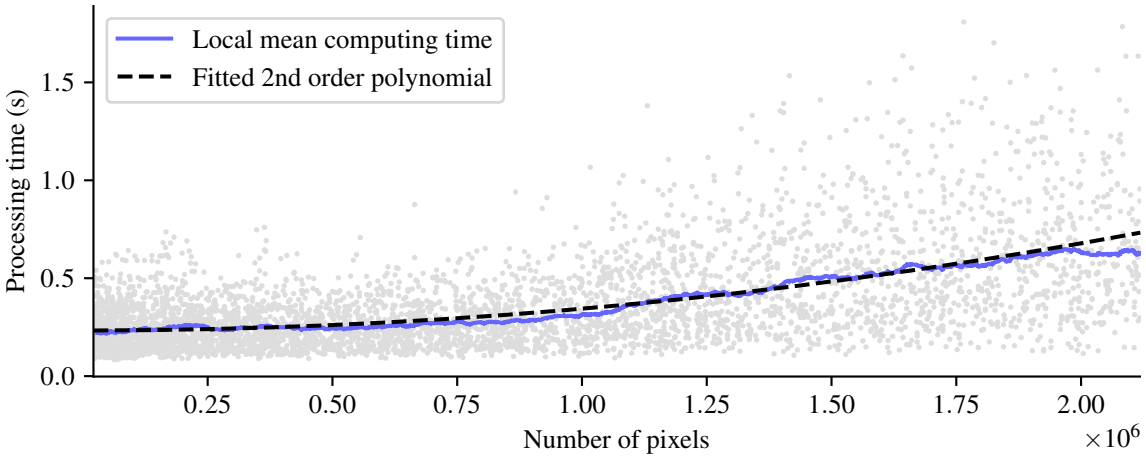


Figure 4.5: **Gaussian *Sea-thru* processing time** on a RTX A5000 GPU using the simplex algorithm.

4.2.3 Implementation

Our method is applied on each color channel independently. Each resulting component is then normalized to obtain the final restored image.

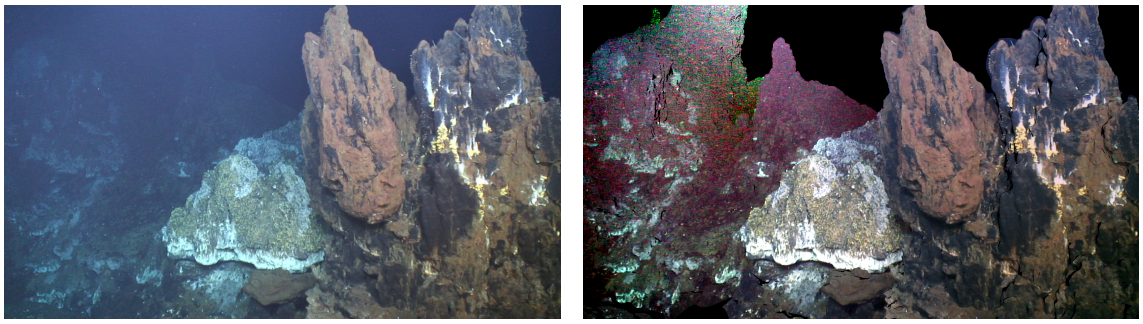
Initialization. Similarly to *Sea-thru*, our approach requires only a coarse initialization. We initialize the parameters with $B_c = \beta_c = \gamma_c = 0.1$, as if water had a very light effect on the image formation.

Optimization. The parameters of Eq. (4.1) image formation model have practical realistic bounds: $0 < B_c < 1$ and $0 < \beta_c, \gamma_c < 5$. In order to incorporate these constraints into the estimation procedure, we minimize Eq. (4.7) using a bounded optimization algorithm. We tested successfully different optimization methods: limited-memory BFGS, trust region constrained and simplex. Over the course of our experiments, we found that the simplex algorithm (Nelder and Mead, 1965) provided the most effective minimization of the objective function.

Normalization. As a final processing step, we normalize each color channel by performing a simple histogram stretching. This is achieved by clipping the pixel intensities of each color channel between their lowest 1% and top 99% values, and then normalize the resulting channels between 0 and 1.

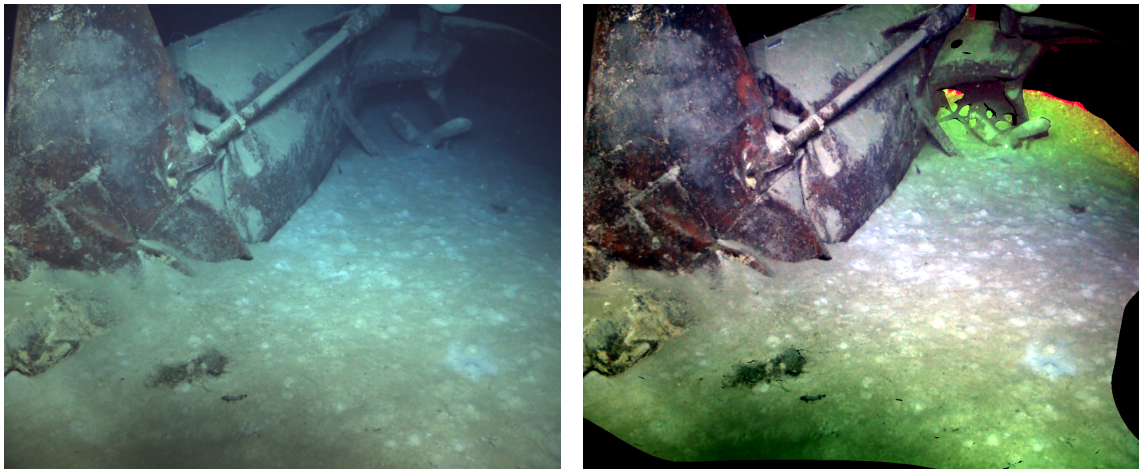
Processing time. The processing time of Gaussian *Sea-thru* mainly depends on the method used to perform the optimization. Using the simplex algorithm, Figure 4.5 shows that the method appears to have a processing time in $O(|\boldsymbol{I}_c|^2)$, with $|\boldsymbol{I}_c|$ the number of pixels. Restoring an underwater image with a resolution of 1920 by 1080 pixels takes on average 0.75 second on a RTX A5000 GPU.

4.2.4 Limitations



(a) Underwater image. (b) Gaussian *Sea-thru*.

Figure 4.6: **Quantization limitations of Gaussian *Sea-thru*.** Images are stored in an 8-bits format. Because of this, there is no sufficient information in the image to recover distant areas, like in the top-left corner of the image.



(a) Underwater image. (b) Gaussian *Sea-thru*.

Figure 4.7: **Vignetting limitations of Gaussian *Sea-thru*.** The image formation model we use in Gaussian *Sea-thru* does not include the vignetting effect created by artificial lighting systems. Because of this, borders of the image deviate from the model, resulting in distorted colors.

Quantization. In our experiments , we apply our method to 8-bits images, in contrast with *Sea-thru* that processes raw high dynamic range images. As illustrated in Figure 4.6, this 8-bits quantization introduces limitations in the restoration process. Due to quantization, there is no information left on distant areas of the underwater image. Hence, it is not possible to retrieve the unattenuated pixel intensities of these areas.

Vignetting. In Figure 4.7, the lights embedded by the deep-sea vehicle induces a strong vignetting effect (see Section 2.1.3). Since this effect is not included in the underwater image formation model, applying Gaussian *Sea-thru* to the image results on distorted color on the edges.

4.3 Leveraging scene structure

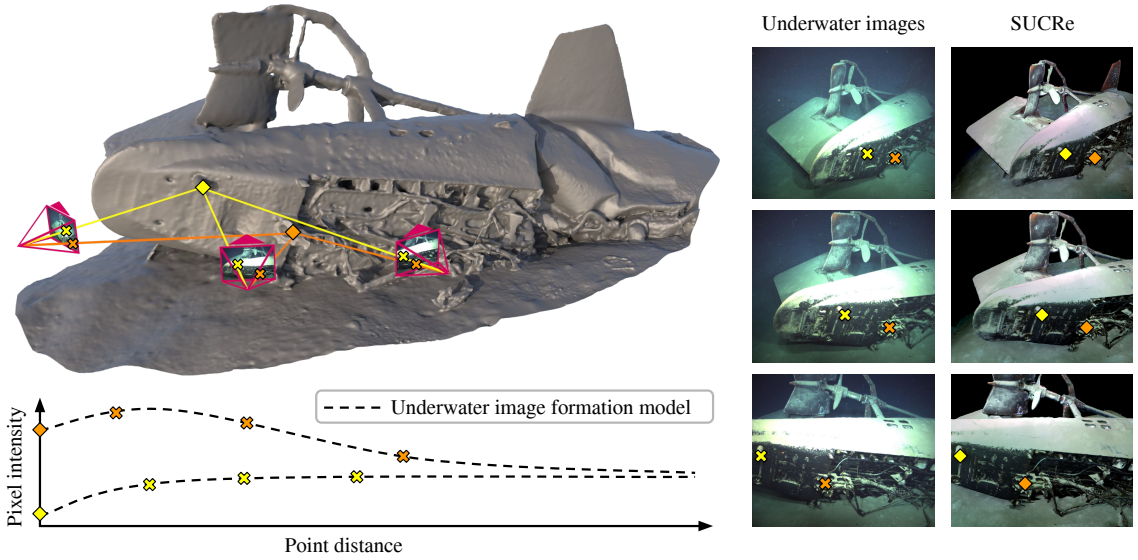


Figure 4.8: **SUCRe pipeline.** We use camera poses, intrinsics and depth maps resulting from a SfM to geometrically pair pixels between different views. We project pixels from one view to another, enabling us to pair points in low contrast areas. We then simultaneously estimate the parameters of an underwater image formation model along with the restored image. This figure illustrates our method on a real-world deep-sea dive at a submarine wreck.

4.3.1 Motivation

Distance maps used in *Sea-thru* or Gaussian *Sea-thru* are usually estimated using SfM. Yet, these methods do not exploit all the 3D information given by the SfM. This section introduces a novel approach, named SUCRe, that overcomes the limitations of single-view underwater image color restoration methods by leveraging multiple observations of the scene, thus eliminating the need for additional assumptions. Moreover, we may obtain from other views closer observations of points, adding information that is not available in the image to be restored. Also, using multiple views of the same scene allows to virtually increase the dynamic range of the observed pixels. Our method takes as input underwater images together with their corresponding camera poses, intrinsics and depth maps.

Figure 4.8 illustrates the core idea of our method. By pairing pixels in multiple images, we are able to follow the intensity evolution of points at different distances and estimate the parameters of an underwater image formation model and pixel intensities at a hypothetical distance of zero meter, implying a lack of disturbance by the water medium.

4.3.2 Method

The proposed method relies on three components. First, based on multiple images of the same scene, we compute camera poses, intrinsics and depth map using SfM and multi-view stereo pipelines. Then, we use this information to pair pixels geometrically between images. Finally, we simultaneously estimate the parameters of an underwater image formation model along with the restored image in a single optimization procedure.

SfM pipeline. To obtain the inputs for SUCRe, we first compute a SfM using the pipeline described in Section 3.4 for individual models. More specifically, i) we pair images spatially using the navigation data, ii) we use SuperPoint (DeTone et al., 2018) and SuperGlue (Sarlin et al., 2020) to perform feature matching, iii) we perform bundle adjustment using navigation priors. Then, by using the resulting SfM as input, we compute a dense 3D mesh of the scene with the OpenMVS software (Cernea, 2020). Finally, depth maps are obtained by ray-casting the images onto the 3D mesh.

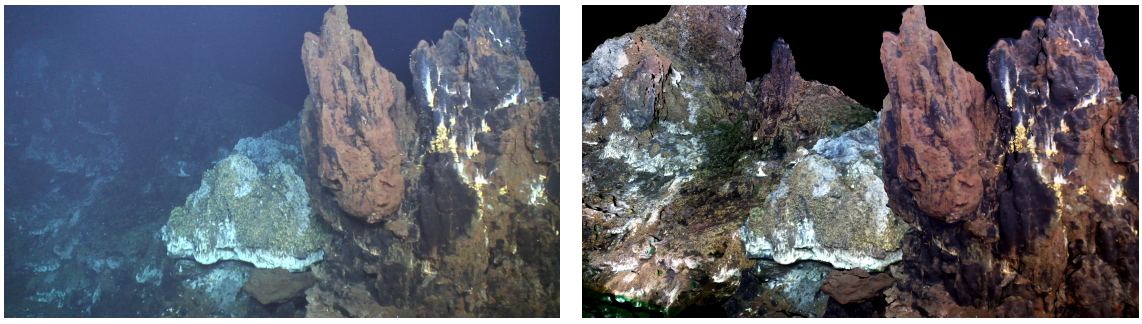
Dense multi-view pixel pairing. The second step of our approach is to pair pixels in a dense manner between different views. This is accomplished by projecting pixel coordinates from one view to another using the depth maps as well as the poses of the cameras and their intrinsics parameters. Let $\mathbf{x}_1$ be the homogeneous coordinates of a pixel in image view i_1 with depth $d_1 \in \mathbb{R}$ and homogeneous pose matrix ${}^{w}\boldsymbol{T}_{i_1} \in SE(3)$. Let $\boldsymbol{K} \in \mathbb{R}^{3\times3}$ be the intrinsic calibration matrix of images i_1 and i_2. The projection of $\mathbf{x}_1$ in image view i_2 with pose ${}^{w}\boldsymbol{T}_{i_2}$ can be obtained by:

$$\mathbf{x}_2 = \pi\left(\boldsymbol{K}\left({}^{i_2}\boldsymbol{T}_{w}\,{}^{w}\boldsymbol{T}_{i_1}\right) \odot \left(\boldsymbol{K}^{-1}\, d_1\, \mathbf{x}_1\right)\right). \tag{4.9}$$

We then back-project $\mathbf{x}_2$ in i_1 view using i_2 depth map:

$$\mathbf{x}'_1 = \pi\left(\boldsymbol{K}\left({}^{i_1}\boldsymbol{T}_{w}\,{}^{w}\boldsymbol{T}_{i_2}\right) \odot \left(\boldsymbol{K}^{-1}\, d_2\, \mathbf{x}_2\right)\right), \tag{4.10}$$

where d_2 is the depth of $\mathbf{x}_2$ in image view i_2. The pixels in both images ($\mathbf{x}_1$, $\mathbf{x}_2$) are only paired if $\mathbf{x}'_1$ and $\mathbf{x}_1$ land on the same pixel coordinate, *i.e.*, pixels are matched to each other in both directions, from i_1 to i_2 and from i_2 to i_1. This ensures that each pixel has only one match in both images. It also filters out points occluded by



(a) Underwater image. (b) SUCRe.

Figure 4.9: **Applying SUCRe from an image of the Eiffel Tower dataset.** Pixels without depth information are left blank.

the structure. This geometry-based approach allows us to robustly pair pixels in scenarios where feature matching algorithms fail, *e.g.*, in low contrast areas where most image signal has been attenuated like in the top left corner of Figure 4.6.

Multi-view optimization. With multiple observations of the same pixel $\boldsymbol{J}_{c,p}$, our problem becomes well-posed. In SUCRe, we formulate the underwater image formation model described by Eq. (4.1) in a multi-view setting:

$$\boldsymbol{I}_{i,c,p} = \boldsymbol{J}_{c,p} e^{-\beta_c \boldsymbol{z}_{i,p}} + B_c(1 - e^{-\gamma_c \boldsymbol{z}_{i,p}}). \tag{4.11}$$

Parameters of Eq. (4.11) are then estimated by fitting the model in a least squares manner. We note our residuals:

$$\boldsymbol{r}_{i,c,p} = \boldsymbol{I}_{i,c,p} - \boldsymbol{J}_{c,p} e^{-\beta_c \boldsymbol{z}_{i,p}} - B_c(1 - e^{-\gamma_c \boldsymbol{z}_{i,p}}). \tag{4.12}$$

We aim to find the parameters that minimize the cost function:

$$\mathcal{L}(\boldsymbol{J}_c, B_c, \beta_c, \gamma_c) = \sum_i \sum_p \boldsymbol{r}_{i,c,p}^2. \tag{4.13}$$

More precisely, we look for:

$$\hat{\boldsymbol{J}}_c, \hat{B}_c, \hat{\beta}_c, \hat{\gamma}_c = \underset{\boldsymbol{J}_c, B_c, \beta_c, \gamma_c}{\arg\min}\ \mathcal{L}(\boldsymbol{J}_c, B_c, \beta_c, \gamma_c). \tag{4.14}$$

Because some pixels in low contrast areas are matched with closer observations, we are able to retrieve their color despite insufficient information about them on the image being restored. This can be observed in the top left corner of the image presented in Figure 4.9.

4.3.3 A partial closed-form solution

The core of the proposed approach is an optimization problem. Our goal is to estimate the parameters of an underwater image formation model that fit observations of paired pixel intensities and distances. Yet, because this optimization

process involves estimating an entire image, employing Jacobian-based optimization techniques, such as the Levenberg-Marquardt algorithm, becomes intricate. For instance, considering 10 million observations, restoring an image with a resolution of 1920 by 1080 pixels would require generating a Jacobian matrix of approximate dimensions of 10 million by 2 million, demanding a staggering 80 terabytes of memory resources. To overcome this problem, this section reveals that when solving Eq. (4.14), we can actually represent $\boldsymbol{J}_{c,p}$ and B_c in terms of β_c and γ_c. This new representation enables us to perform the optimization procedure solely on the β_c and γ_c parameters.

The first step in attaining this representation involves expressing $\boldsymbol{J}_{c,p}$ in terms of B_c, β_c and γ_c. To achieve this, we adopt the following notations for the sake of simplicity:

$$\boldsymbol{b}_{i,c,p} = 1 - e^{-\gamma_c \boldsymbol{z}_{i,p}} \tag{4.15}$$

and

$$\boldsymbol{D}_{i,c,p} = \boldsymbol{I}_{i,c,p} - B_c \boldsymbol{b}_{i,c,p}. \tag{4.16}$$

To retrieve $\boldsymbol{J}_{c,p}$, we need to find where the partial derivative of the cost function defined in Eq. (4.13) with respect to $\boldsymbol{J}_{c,p}$ is equal to zero:

$$\frac{\partial}{\partial \boldsymbol{J}_{c,p}} \mathcal{L}(\boldsymbol{J}_c, B_c, \beta_c, \gamma_c) = 0 \tag{4.17}$$

$$\Rightarrow \frac{\partial}{\partial \boldsymbol{J}_{c,p}} \sum_i \sum_p \Big[\boldsymbol{D}_{i,c,p} - \boldsymbol{J}_{c,p} e^{-\beta_c \boldsymbol{z}_{i,p}} \Big]^2 = 0 \tag{4.18}$$

$$\Rightarrow \sum_i \Big[\frac{\partial}{\partial \boldsymbol{J}_{c,p}} \left(\boldsymbol{D}_{i,c,p} - \boldsymbol{J}_{c,p} e^{-\beta_c \boldsymbol{z}_{i,p}} \right)^2 \Big] = 0 \tag{4.19}$$

$$\Rightarrow \sum_i \Big[-2 e^{-\beta_c \boldsymbol{z}_{i,p}} \left(\boldsymbol{D}_{i,c,p} - \boldsymbol{J}_{c,p} e^{-\beta_c \boldsymbol{z}_{i,p}} \right) \Big] = 0 \tag{4.20}$$

$$\Rightarrow 2 \sum_i \Big[\boldsymbol{J}_{c,p} e^{-2\beta_c \boldsymbol{z}_{i,p}} - \boldsymbol{D}_{i,c,p} e^{-\beta_c \boldsymbol{z}_{i,p}} \Big] = 0 \tag{4.21}$$

$$\Rightarrow \sum_i \Big[\boldsymbol{J}_{c,p} e^{-2\beta_c \boldsymbol{z}_{i,p}} \Big] = \sum_i \Big[\boldsymbol{D}_{i,c,p} e^{-\beta_c \boldsymbol{z}_{i,p}} \Big] \tag{4.22}$$

$$\Rightarrow \boldsymbol{J}_{c,p} = \frac{\sum_i \Big[\boldsymbol{D}_{i,c,p} e^{-\beta_c \boldsymbol{z}_{i,p}} \Big]}{\sum_i \Big[e^{-2\beta_c \boldsymbol{z}_{i,p}} \Big]}. \tag{4.23}$$

Then, to retrieve B_c, we first develop Eq. (4.23) using the notation defined in Eq. (4.16). This way, we can isolate B_c:

$$\boldsymbol{J}_{c,p} = \boldsymbol{\nu}_{c,p} - B_c \boldsymbol{\xi}_{c,p} \tag{4.24}$$

with

$$\boldsymbol{\nu}_{c,p} = \frac{\sum_i \left[\boldsymbol{I}_{i,c,p} e^{-\beta_c \boldsymbol{z}_{i,p}} \right]}{\sum_i \left[e^{-2\beta_c \boldsymbol{z}_{i,p}} \right]} \tag{4.25}$$

and

$$\boldsymbol{\xi}_{c,p} = \frac{\sum_i \left[\boldsymbol{b}_{i,c,p} e^{-\beta_c \boldsymbol{z}_{i,p}} \right]}{\sum_i \left[e^{-2\beta_c \boldsymbol{z}_{i,p}} \right]}. \tag{4.26}$$

After injecting the closed form of $\boldsymbol{J}_{c,p}$ from Eq. (4.24) in the cost function, we need to find where the partial derivative of Eq. (4.13) with respect to B_c is equal to zero:

$$\frac{\partial}{\partial B_c} \mathcal{L}(\boldsymbol{J}_c, B_c, \beta_c, \gamma_c) = 0 \tag{4.27}$$

$$\Rightarrow \frac{\partial}{\partial B_c} \sum_i \sum_p \left[\boldsymbol{I}_{i,c,p} - (\boldsymbol{\nu}_{c,p} - B_c \boldsymbol{\xi}_{c,p}) e^{-\beta_c \boldsymbol{z}_{i,p}} - B_c \boldsymbol{b}_{i,c,p} \right]^2 = 0. \tag{4.28}$$

For the sake of simplicity let us rewrite Eq. (4.28) as:

$$\frac{\partial}{\partial B_c} \sum_i \sum_p \left[\boldsymbol{\zeta}_{i,c,p} - B_c \boldsymbol{\eta}_{i,c,p} \right]^2 = 0 \tag{4.29}$$

with

$$\boldsymbol{\zeta}_{i,c,p} = \boldsymbol{I}_{i,c,p} - \boldsymbol{\nu}_{c,p} e^{-\beta_c \boldsymbol{z}_{i,p}} \tag{4.30}$$

and

$$\boldsymbol{\eta}_{i,c,p} = \boldsymbol{b}_{i,c,p} - \boldsymbol{\xi}_{c,p} e^{-\beta_c \boldsymbol{z}_{i,p}}. \tag{4.31}$$

We can now develop Eq. (4.29) to express B_c in terms of β_c and γ_c:

$$\sum_i \sum_p \left[\frac{\partial}{\partial B_c} \left(\boldsymbol{\zeta}_{i,c,p} - B_c \boldsymbol{\eta}_{i,c,p} \right)^2 \right] = 0 \tag{4.32}$$

$$\Rightarrow 2 \sum_i \sum_p \left[\boldsymbol{\zeta}_{i,c,p} \boldsymbol{\eta}_{i,c,p} - B_c \boldsymbol{\eta}_{i,c,p}^2 \right] = 0 \tag{4.33}$$

$$\Rightarrow \sum_i \sum_p \left[B_c \boldsymbol{\eta}_{i,c,p}^2 \right] = \sum_i \sum_p \left[\boldsymbol{\zeta}_{i,c,p} \boldsymbol{\eta}_{i,c,p} \right] \tag{4.34}$$

$$\Rightarrow B_c = \frac{\sum_i \sum_p \left[\boldsymbol{\zeta}_{i,c,p} \boldsymbol{\eta}_{i,c,p} \right]}{\sum_i \sum_p \left[\boldsymbol{\eta}_{i,c,p}^2 \right]}. \tag{4.35}$$

Equations (4.24) and (4.35) allow to express the cost function with only β_c and γ_c parameters. Consequently, to enable the use of the Levenberg-Marquardt algorithm, the Jacobian matrix is only computed for β_c and γ_c parameters. This Jacobian matrix can either be obtained by injecting Eqs. (4.24) and (4.35) in Eq. (4.12) and derive the residuals, or by using an auto-differentiation framework.

In conclusion, we have demonstrated that it is possible to perform Jacobian-based optimization procedures on only two parameters, hence avoiding the need to perform the optimization procedure on an entire image.

4.3.4 Modeling artificial lights

The underwater image formation model described by Eq. (4.1) and expressed within a multi-view setting in Eq. (4.11) was designed for natural light environments. Because of this, the model does not incorporate some specificities of deep-sea imaging. Mostly, it does not model the vignetting effect created by artificial lighting systems. Consequently, we here extend Eq. (4.11) to incorporate artificial lighting systems into the image formation model:

$$\boldsymbol{I}_{i,c,p} = \boldsymbol{l}_{i,p} \left(\boldsymbol{J}_{c,p} e^{-\beta_c \boldsymbol{z}'_{i,p}} + B_c (1 - e^{-\gamma_c \boldsymbol{z}'_{i,p}}) \right), \tag{4.36}$$

where $\boldsymbol{l}_{i,p}$ is a scalar attenuating pixel intensities based on the positions of their 3D observations and $\boldsymbol{z}'_{i,p}$ is the distance traveled by the light as it travels from the light source through the scene to reach the camera. Similarly to previous work (Arnaubec et al., 2015; Bryson et al., 2015; Nakath et al., 2021), we represent each light source as an invert pinhole camera. Instead of projecting the scene into the camera view, we use this camera to virtually project light onto the scene.

Our artificial light's model consists of the relative 6DoF pose between the light source and the camera, and the light pattern to be projected on the scene. Here, we choose to represent the light pattern as a zero-centered multivariate normal, as it requires only a few parameters to estimate and serves as a suitable approximation for the halo effect created by the artificial light. In practice, the relative light's pose ${}^{l}\boldsymbol{T}_{c}$ is represented in $\mathfrak{se}(3)$ using six scalars $\{\omega_{lx}, \omega_{ly}, \omega_{lz}, \rho_{lx}, \rho_{ly}, \rho_{lz}\}$, such as:

$${}^{l}\boldsymbol{T}_{c} = \exp\left(\begin{bmatrix} 0 & -\omega_{lz} & \omega_{ly} & \rho_{lx} \\ \omega_{lz} & 0 & -\omega_{lx} & \rho_{ly} \\ -\omega_{ly} & \omega_{lx} & 0 & \rho_{lz} \\ 0 & 0 & 0 & 0 \end{bmatrix} \right). \tag{4.37}$$

The covariance matrix of the zero-centered multivariate normal Σ^l is represented using three scalars $\{\sigma_{lu}, \sigma_{lv}, \kappa_l\}$, such as:

$$\boldsymbol{\Sigma}^l = \begin{bmatrix} \sigma_{lu}^2 & \kappa_l \\ \kappa_l & \sigma_{lv}^2 \end{bmatrix}. \tag{4.38}$$

The light's attenuation factor $\boldsymbol{l}_{i,p}$ is then computed as follows. Let ${}^{c}\mathbf{x}_{i,p}$ be the homogeneous coordinates of a pixel in the camera view. The pixel has an associated depth d and the camera has a calibration matrix $\boldsymbol{K}$. Without a loss of generality,

we assume the calibration matrix of the light source is the identity matrix. We first project the pixel in 3D space into the light's camera frame:

$$^{l}\mathbf{X}_{i,p} = {}^{l}\boldsymbol{T}_{c} \odot \left(\boldsymbol{K}^{-1}\, d\, {}^{c}\mathbf{x}_{i,p}\right). \tag{4.39}$$

We then project it into the light's camera view using the light's identity calibration matrix:

$$^{l}\mathbf{x}_{i,p} = \pi\left({}^{l}\mathbf{X}_{i,p}\right). \tag{4.40}$$

We now compute $\boldsymbol{l}_{i,p}$ using the multivariate normal probability density function, but normalize it so that the maximum value is equal to one:

$$\boldsymbol{l}_{i,p} = \exp\left(-\frac{1}{2}\, {}^{l}\mathbf{x}_{i,p}\, (\boldsymbol{\Sigma}^{l})^{-1}\, {}^{l}\mathbf{x}_{i,p}\right). \tag{4.41}$$

Finally, $\boldsymbol{z}'_{i,p}$ is obtained as the sum of the distance between the pixel and the scene and the distance between the scene and the light:

$$\boldsymbol{z}'_{i,p} = \boldsymbol{z}_{i,p} + \|{}^{l}\mathbf{X}_{i,p}\|. \tag{4.42}$$

The restored image and the parameters of the image formation model are then estimated similarly to Eq. (4.14) in a least squares manner. In this case, we note our residuals:

$$\boldsymbol{r}^{l}_{i,c,p} = \boldsymbol{I}_{i,c,p} - \boldsymbol{l}_{i,p}\left(\boldsymbol{J}_{c,p} e^{-\beta_c \boldsymbol{z}_{i,p}} - B_c(1 - e^{-\gamma_c \boldsymbol{z}_{i,p}})\right). \tag{4.43}$$

We aim to find the parameters that minimize the following cost function:

$$\hat{\boldsymbol{J}}_c, \hat{B}_c, \hat{\beta}_c, \hat{\gamma}_c, {}^{l}\hat{\boldsymbol{T}}_c, \hat{\boldsymbol{\Sigma}}^{l} = \underset{\boldsymbol{J}_c, B_c, \beta_c, \gamma_c, {}^{l}\boldsymbol{T}_c, \boldsymbol{\Sigma}^{l}}{\arg\min} \sum_{i} \sum_{p} \left(\boldsymbol{r}^{l}_{i,c,p}\right)^{2}. \tag{4.44}$$

In conclusion, Eqs. (4.36) to (4.42) present a model that incorporate the vignetting effect created by artificial lighting systems into an existing underwater image formation model using only nine parameters.

4.3.5 Implementation

Initialization. Our method was tested using various initialization techniques described below. Yet, practical experiments have shown the method to be robust even in the case of a rough initialization.

- *Local initialization:* β_c and γ_c are initialized using Gaussian *Sea-thru*. $\boldsymbol{J}_c$ and B_c are then initialized using Eqs. (4.24) and (4.35).
- *Global initialization:* From all the images, the veiling light B_c is initialized as the mean intensity of pixels without depth information. Indeed, pixels

with no depth mostly result from very distant observations. In rarer instances, this lack of depth information may be due to holes in the mesh. In practice, these rare instances have a negligible impact in comparison to the presence of distant observations and are smoothed out during the computation of the mean intensity. The backscatter coefficient γ_c is initialized using the distance-dependent dark channel prior described by Figure 4.1 on all the images. The absorption coefficient β_c is initialized using a similar assumption we call the bright channel prior. Instead of selecting the bottom 1% of pixel intensities, we select the top 99% of pixel intensities. We then estimate simultaneously the maximum possible value for $\boldsymbol{J}_c$, we call $J\text{max}_c$, along with β_c to fit these bright pixel intensities.

- *Coarse:* We initialize all parameters as if the water had almost no effect on light propagation. Thus, we set $\boldsymbol{J}_c = \boldsymbol{I}_c$ and $B_c = \beta_c = \gamma_c = 0.1$. This is the method we use when including the artificial lighting model, as other initialization methods discard the vignetting parameters. Artificial lights are initialized with ${}^l\boldsymbol{T}_c = \mathbf{I}_{4\times4}$ and $\boldsymbol{\Sigma}_l = \mathbf{I}_{2\times2}$, *i.e.*, the light is superimposed with the camera and projects a light pattern whose intensity follows a centered multivariate normal with identity covariance matrix.

Empirically, we use global initialization, as it needs to be computed only once for a given dataset. Nevertheless, neither local nor global initializations enable the initialization of parameters related to artificial lights. Consequently, we use coarse initialization when employing the model that includes artificial lights.

Optimization. Our method was tested using various optimization schemes.

- *Levenberg-Marquardt:* Using Eqs. (4.24) and (4.35), we implemented the optimization procedure described by Eq. (4.14) using the Levenberg-Marquardt algorithm, by estimating only β_c and γ_c. We determined the initial damping parameter and the damping factor through trial and error. We initialize the damping parameter to 0.1, and we set the damping factor to 10.

- *Simplex:* Similarly to our Levenberg-Marquardt implementation, we relied on Eqs. (4.24) and (4.35) to solve Eq. (4.14) using the simplex algorithm (Nelder and Mead, 1965). As it gives the possibility of specifying bounds, we set physically realistic bounds: $0 < \beta_c, \gamma_c < 5$.

- *Gradient descent:* We also propose an optimization procedure that does not rely on Eqs. (4.24) and (4.35). To optimize the many parameters involved in jointly estimating the restored image along with the underwater image formation model parameters, we use gradient descent (Paszke et al., 2019)

with an Adam optimizer (Kingma and Ba, 2015). Each step of the gradient descent is computed using all matched observations by minimizing the function described by Eq. (4.14). Specifically, we perform 200 optimization steps with a learning rate of 0.05. In particular, this is the method we use when including the artificial lights model, as we did not compute a closed-form estimation of $\boldsymbol{J}_c$ for the model described by Eq. (4.36).

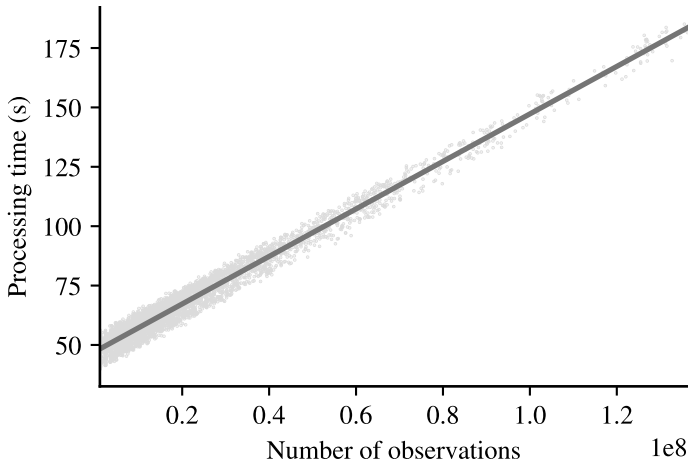


Figure 4.10: **SUCRe processing time.** Scatter plot illustrating the relation between the number of observations in an image and the processing time of restoring it. Each point corresponds to an image of the 2015 Eiffel Tower dive. The number of observations is the total number of pixels that have been paired to another pixel in the processed image. The processing time of the optimization procedure follows a very consistent linear increase *w.r.t.* the number of observations. The intercept of approximately 50 seconds is due to the pairing step that computes pixel pairs for all 4,875 images in the dataset. The optimization was performed using gradient descent (Paszke et al., 2019), minimizing Eq. (4.14).

Processing time. The processing time of our approach can be divided into two main components: i) pairing pixels between the image to be restored and every other images, ii) the optimization procedure described by Eq. (4.14). The pairing step depends on the size of the images and the number of candidate matching images in the dataset. As illustrated by Figure 4.10, the optimization step evolves linearly with the number of matched observations. To restore an image from a dataset comprising 4,875 images with a resolution of 1920 by 1080 pixels, our approach takes about 50 seconds to compute pixel pairs and 1 minute and 40 seconds for the optimization procedure using 100 millions matched pixel observations, for a total of 2.5 minutes processing time. All computations were performed with 32 Intel Xeon Gold 6226R CPU threads and a RTX A5000 GPU.

4.3.6 Limitations

SfM quality. As our approach heavily relies on SfM information, its performance depends entirely on the quality of the SfM. Incorrect depth maps or camera poses will lead to invalid pixel pairs, deteriorating the optimization procedure.

Static scene. In practice, our method relies on SfM information. Consequently, non-static objects are often excluded during the SfM pipeline, and more generally, it is not possible to perform dense pixel pairing of these objects using the geometric method outlined in Section 4.3.2. As a result, our method is currently only applicable on static elements of the scene.

Underwater image formation model. The image formation model described by Eq. (4.1) is based on a minimal set of three parameters. Representing all the factors influencing light propagation in the observed scene would naturally entail a much larger number of parameters. The model used here is a simplification primarily addressing absorption and backscattering phenomena. It assumes, for instance, that the observed scene is viewed through a water volume with consistent salinity, temperature, absorption, and diffusion properties. However, in reality, these parameters experience local variations, such as chimneys releasing hydrothermal fluids or the vehicle lifting off sediment from the seabed.

Vignetting parameters redundancy. There might be a theoretical redundancy between the roll parameter ω_{lz} of the light's 6DoF pose and the covariance parameter κ_l of the light pattern. Indeed, both parameters only have a 2D rotation effect in the light's projection plane.

4.4 Experiments

In this section, we evaluate the performance of the developed approaches on four different datasets, encompassing both deep-sea and natural light environments.

4.4.1 Benchmark datasets

We evaluate our method on four distinct datasets. Two of these datasets, Varos (Zwilgmeyer et al., 2021) and *Sea-thru* D5 (Akkaynak and Treibitz, 2019), offer respectively reference images and color charts with known colors, enabling the calculation of quantitative metrics. The remaining two datasets showcase the real-life practical applicability of our methods for two sites of interest, *i.e.*, the Eiffel Tower hydrothermal vent presented in Chapter 3 and a submarine wreck.

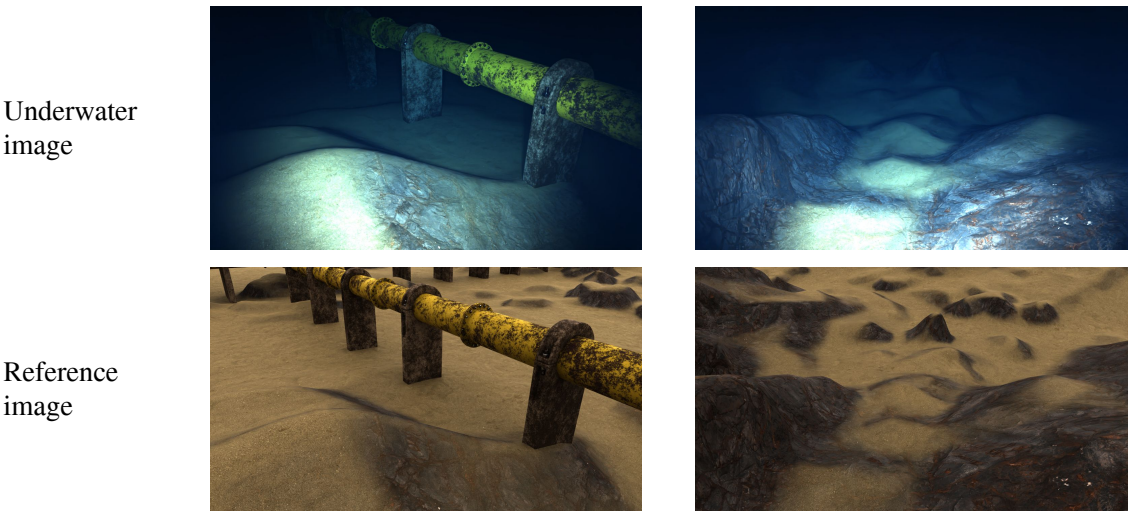


Figure 4.11: **Varos dataset.** The Varos dataset offers simulated underwater images along with reference images rendered with uniform lighting.

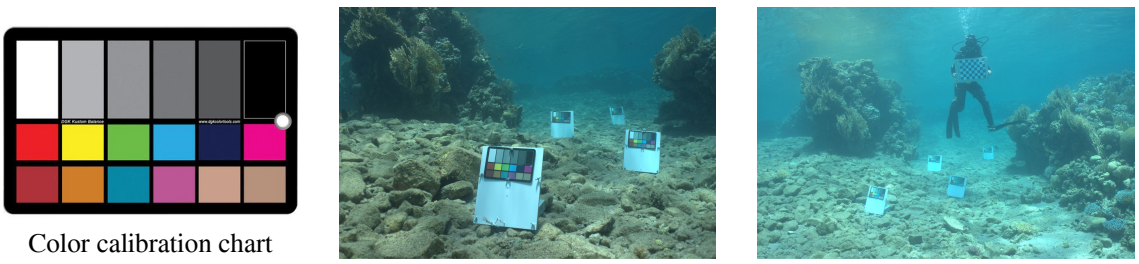


Figure 4.12: ***Sea-thru* D5 dataset.** The *Sea-thru* D5 dataset provides images in a raw file format with four color calibration charts dispatched across the scene.

Varos. Varos is a synthetic deep-sea dataset embedding 4,715 images that were rendered with Blender within a simulated underwater setting (Zwilgmeyer et al., 2021). The dataset benefits from Blender's ray tracing technology, which enables the simulation of scattering and attenuation effects that are commonly observed in underwater scenes. A notable feature of Varos is its provision of reference images under uniform lighting conditions, which are useful for assessing the accuracy of color restoration techniques. These reference images offer a consistent baseline for comparison and facilitate the measurement of standard metrics such as PSNR and SSIM. Examples of Varos underwater images along with their corresponding reference image are provided in Figure 4.11.

***Sea-thru* D5.** The *Sea-thru* D5 dataset is composed of 43 raw images captured under natural light conditions, along with their corresponding distance maps obtained through SfM (Akkaynak and Treibitz, 2019). The scene contains four color calibration charts with known patterns positioned throughout the scene. These charts, illustrated in Figure 4.12, serve as ground truth for computing metrics used to evaluate the performance of color restoration algorithms.

Eiffel Tower. We evaluate our methods on the 2015 dive of the Eiffel Tower dataset presented in Chapter 3.

Submarine wreck. Similar to the Eiffel Tower dataset, the submarine wreck is a private Ifremer dataset comprising 4,595 images extracted from a ROV's video feed during a dive to a submarine wreck at a depth of around 1,150 meters. Due to the depth, the ROV was equipped with an artificial lighting system to illuminate the wreck.

4.4.2 Quantitative evaluation

Metrics. Evaluating the performance of underwater color restoration methods is a challenging task. Ground truth restored colors are generally unavailable for real-world underwater images. Nevertheless, synthetic datasets like Varos or images featuring color charts, as found in the *Sea-thru* D5 dataset can be employed to provide reference values. These references enable the computation of so called full-reference metrics (Li et al., 2020). Additionally, proposed no-reference metrics can be employed to assess factors such as contrast and saturation (Yang and Sowmya, 2015; Panetta et al., 2016). However, recent literature has raised doubts about the ability of these no-reference metrics to accurately measure the correction of physical phenomena such as attenuation and scattering (Li et al., 2020; Jiang et al., 2022).

Method	PSNR ↑	SSIM ↑	UCIQE ↑	UIQM ↑
Acquired image	10.71	0.39	0.60	1.40
Fusion (Ancuti et al., 2012)	10.25	0.35	0.51	2.10
Water-Net (Li et al., 2020)	11.20	0.38	0.54	1.96
FUnIE-GAN (Islam et al., 2020)	11.02	0.35	**0.62**	2.51
Haze-Lines (Berman et al., 2021)	9.64	0.36	0.57	2.00
TACL (Liu et al., 2022)	10.02	0.36	0.44	**2.52**
Gaussian *Sea-thru*	10.15	0.39	0.52	1.88
SUCRe	**12.13**	**0.42**	0.32	1.99

Table 4.1: **Restoration evaluation on Varos dataset.** We report PSNR, SSIM, UCIQE and UIQM on pairs of underwater and reference images illustrated in Figure 4.11.

In this study we rely on six different metrics: PSNR, SSIM, UCIQE, UIQM, CIEDE2000 and $\bar{\psi}$ error. PSNR and SSIM (higher is better) are full-reference measures of image similarity that are particularly useful when entire ground truth restored images are available. UCIQE (Yang and Sowmya, 2015) and UIQM (Panetta et al., 2016) (higher is better) are commonly used no-reference metrics for evaluating the visual quality of restored underwater images. The CIEDE2000 (ΔE_{00}) formula (lower is better) was developed by the International Commission on Illumination to evaluate color differences (Sharma et al., 2005) and is commonly used in underwater color restoration (Ancuti et al., 2017; Li et al., 2021). We hereby compute it between the restored and expected color patches of the *Sea-thru* D5 dataset. The $\bar{\psi}$ error (lower is better) was introduced by Berman et al. (2021) and is designed specifically for images with color charts of known colors distributed throughout the scene. For a given color chart in an image, the $\bar{\psi}$ error is defined as the average angle in RGB space between grayscale patches and a pure gray color. We redefine the error to take into account all twelve color patches in the color calibration charts used in the *Sea-thru* D5 dataset:

$$\bar{\psi} = \frac{1}{12} \sum_{p \in P} \cos^{-1} \left(\frac{\boldsymbol{J}_p \cdot \boldsymbol{E}_p}{\|\boldsymbol{J}_p\| \cdot \|\boldsymbol{E}_p\|} \right), \tag{4.45}$$

where P is a set containing pixel indices of the twelve color patches in the given color chart and $\boldsymbol{E}_p$ denotes the expected RGB values of the color patch with pixel index p.

All quantitative results for the SUCRe method refer to images that were obtained by minimizing Eq. (4.14), not encompassing the vignetting effect of deep-sea images.

Method	$\bar{\psi}\downarrow$	$\bar{\psi}$ std $\downarrow$	$\Delta E_{00}\downarrow$	ΔE_{00} std $\downarrow$
Acquired image	37.14	3.72	36.93	3.68
Fusion (Ancuti et al., 2012)	29.85	6.38	30.60	6.34
Water-Net (Li et al., 2020)	29.12	4.11	31.49	5.89
FUnIE-GAN (Islam et al., 2020)	32.91	3.63	35.55	5.07
Haze-Lines (Berman et al., 2021)	25.80	7.14	28.85	6.89
TACL (Liu et al., 2022)	29.28	4.27	30.50	4.93
Gaussian *Sea-thru*	27.55	3.68	30.64	5.46
SUCRe	**21.45**	**2.63**	**22.56**	**2.84**

Table 4.2: **Restoration evaluation on *Sea-thru* D5 dataset.** We report the ΔE_{00} color difference and $\bar{\psi}$ error in degrees between the restored color charts and their reference illustrated in Figure 4.12. As a way to evaluate the stability of restoration results, we also report these metrics' standard deviation for all restored color charts.

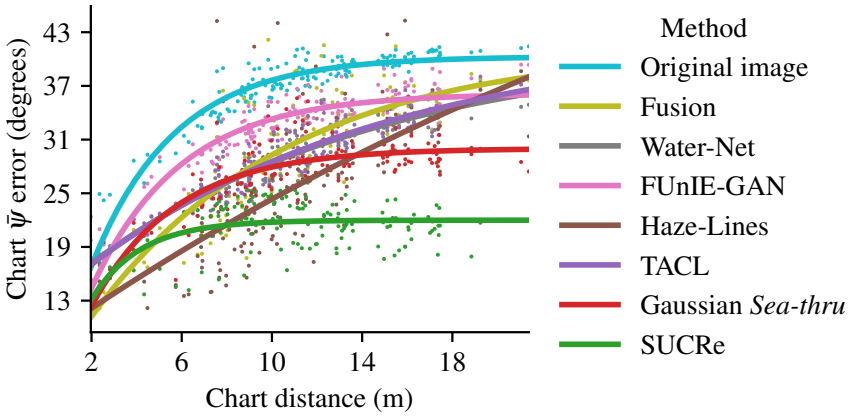


Figure 4.13: **Color chart $\bar{\psi}$ error vs. distance.** Illustrating the relationship between the color charts' distance in the *Sea-thru* D5 dataset and their restoration results using different methods.

Results. As demonstrated in Table 4.1, Gaussian *Sea-thru* displays a mixed performance with low PSNR and high SSIM values on the Varos dataset. This disparity between the two metrics could be attributed to the significant vignetting effect present in Varos dataset images. As discussed in Section 4.2.4, the Gaussian Sea-thru model does not account for this effect, potentially resulting in decreased PSNR along the images' border, while SSIM benefits from the scene structure recovery using depth information. Examining Table 4.2, we observe that Gaussian *Sea-thru* delivers overall good results on the *Sea-thru* D5 dataset. This dataset was captured under natural lighting, which aligns with the settings for which the image formation model described by Eq. (4.1) was designed.

Parameters estimation		PSNR	SSIM
Single-view	Multi-view		
$\boldsymbol{J}_c, \beta_c, B_c, \gamma_c$	—	10.15	0.39
β_c, B_c, γ_c	$\boldsymbol{J}_c$	11.32	0.42
—	$\boldsymbol{J}_c, \beta_c, B_c, \gamma_c$	12.13	0.42

Table 4.3: **Ablation study on Varos.** We show the benefits of using multi-view observations for the estimation of the underwater image formation model parameters and the restored image. In the first row, estimating all parameters in a single view setting is equivalent to Gaussian *Sea-thru*. In the last row, estimating all parameters in a multi-view setting is equivalent to SUCRe.

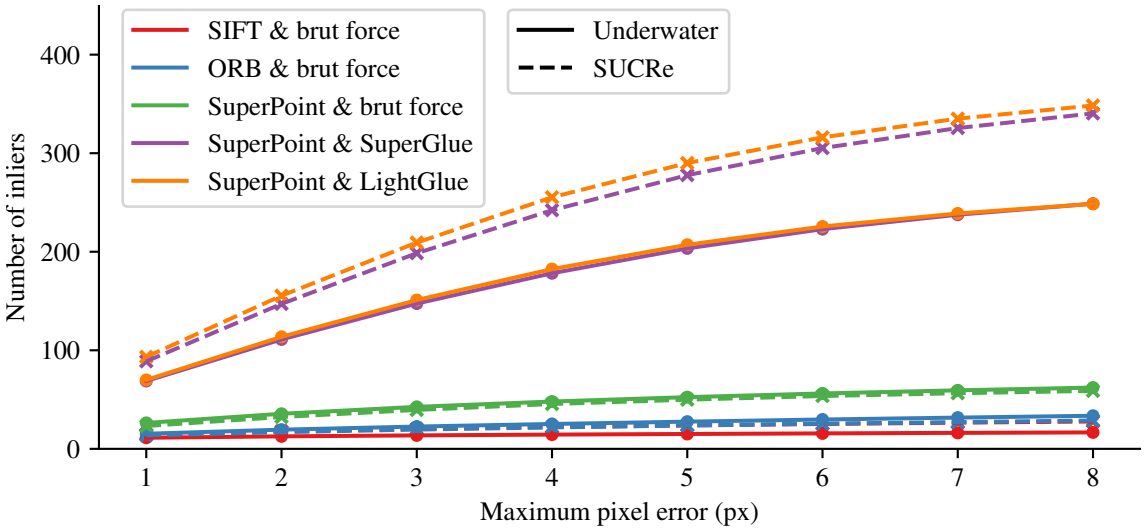


Figure 4.14: **Impact of SUCRe on cross-years feature matching.** We extend Figure 3.7 of Section 3.4.2 by performing feature matching on images restored using SUCRe. Given cross-years image pairs of the Eiffel Tower dataset, we report the number of inliers resulting from five feature matching methods using incremental pixel thresholds.

Both Tables 4.1 and 4.2 indicate that SUCRe outperforms every other method on all full-reference metrics, encouraging the use of multi-view observations for underwater color restoration. Also, SUCRe exhibits consistent restoration outcomes regardless of the observation distance, as highlighted by the low standard deviations in Table 4.2. This stability is further demonstrated in Figure 4.13, showing the connection between color chart distance and $\bar{\psi}$ errors. Notably, SUCRe showcases lower and more uniform $\bar{\psi}$ errors compared to other methods, regardless of the color chart's distance. This effect can be observed to a lesser extent for Gaussian *Sea-thru*. In line with prior studies (Li et al., 2020;

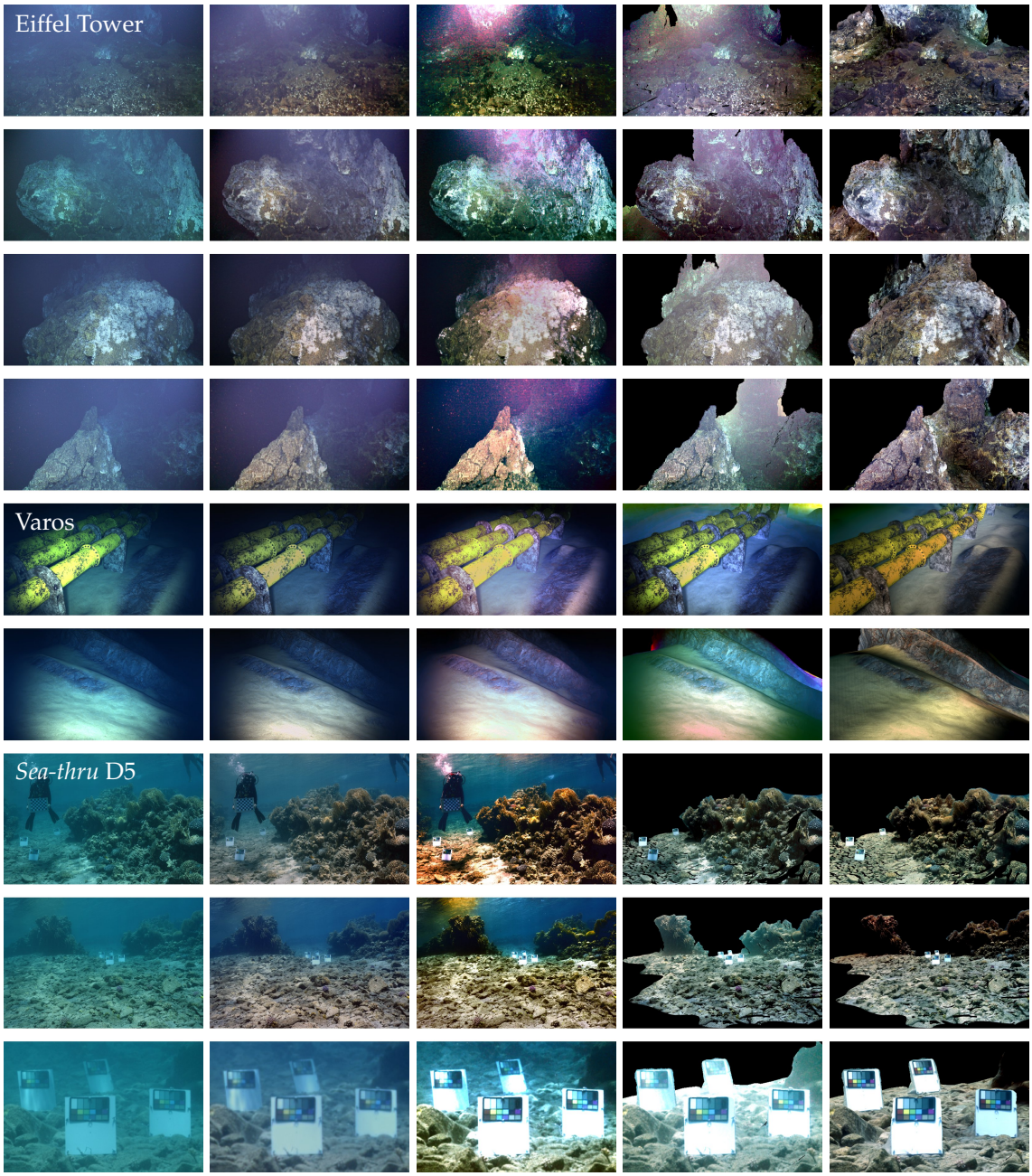
Jiang et al., 2022), we find limited correlation between PSNR/SSIM criteria and UCIQE/UIQM metrics. This is because the latter set of metrics, being no-reference metrics, primarily assess image characteristics like contrast and sharpness, and therefore, they are not capable of comprehensively evaluating the correction of scattering and absorption phenomena.

Gaussian *Sea-thru* serves as a valuable point of comparison with SUCRe, providing insight into the benefits solely attributed to incorporating multi-view observations into the optimization process. This is demonstrated in the brief ablation study in Table 4.3, which investigates the impact of employing multiple views when estimating distinct components of the image formation model. The first row presents Gaussian *Sea-thru* results, where both the image formation model parameters and the restored image are estimated using a single image. The second row fixes the image formation model parameters to those estimated using Gaussian *Sea-thru* and restores the image with multi-view observations, minimizing the same error as SUCRe using Eq. (4.24). The last row displays SUCRe results, where all parameters are estimated within a multi-view context. The results highlight that the SSIM improvement mainly stems from enhanced recovery of low contrast regions when the restored image is estimated using multiple views. The observation of PSNR values suggests that utilizing multi-view observations for estimating both the image formation model parameters and the restored image is essential for achieving the peak performance demonstrated by SUCRe.

Insights for visual localization. Utilizing SUCRe as a preliminary step has the potential to enhance visual localization algorithms. As depicted in Figure 4.14, applying SUCRe to underwater images leads to an increase in the number of inliers resulting from deep-based feature matching methods. In contrast, it maintains a consistent number of inliers when using brute force matching. This observation suggests that, on the one hand, deep-based matchers benefit from mitigating the impact of underwater light propagation, as they were only trained on terrestrial data. On the other hand, brute force matching doesn't yield improvements in cross-years feature matching, primarily because it fails to generalize to topological changes, while deep-based matchers have learned to effectively adapt to these structural changes.

4.4.3 Qualitative evaluation

In Figure 4.15, we present restoration outcomes achieved through various restoration methods. Once again, we observe that Gaussian Sea-thru faces difficulties with the vignetting effect on the Varos dataset, resulting in color distortion along



(a) Underwater image. (b) Water-Net. (c) Haze-Lines. (d) Gaussian *Sea-thru*. (e) SUCRe.

Figure 4.15: **Visual inspection of color restoration results.** As mentioned in Figure 4.9, pixels without depth information are rendered black for Gaussian *Sea-thru* and SUCRe.

the borders. SUCRe effectively recovers the colors of distant elements, which is particularly visible in the final row of images showcasing distant color charts. SUCRe successfully exploits the virtual increase in dynamic range offered by the use of multi-view observations to retrieve the colors of color charts that were considerably attenuated in the original image.

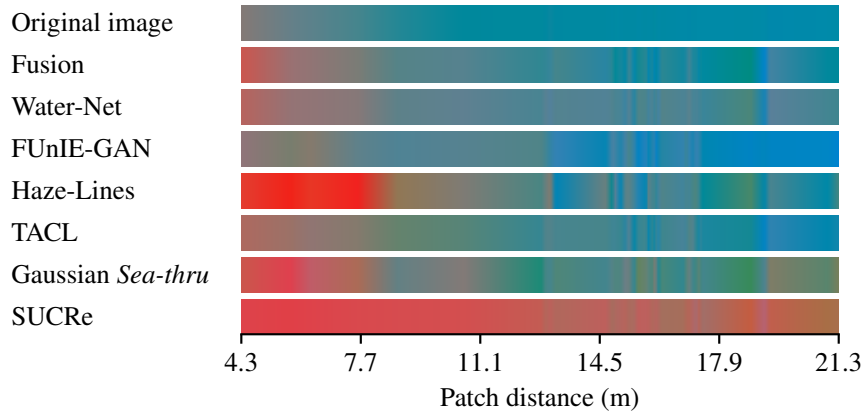


Figure 4.16: **Hue vs. distance.** Tracking the hue value of the red color patch at different distances on the *Sea-thru* D5 dataset using different underwater color restoration methods.

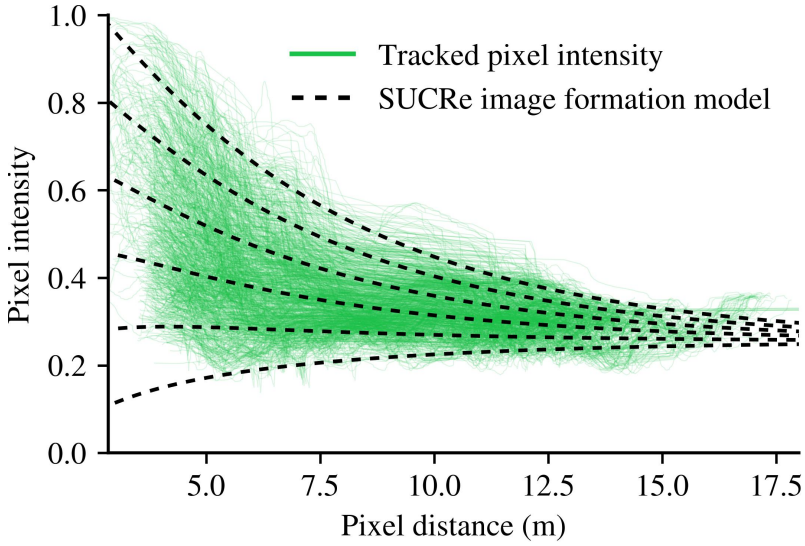


Figure 4.17: **Comparing SUCRe estimated model to deep-sea observations.** Each green curve represents one 3D point observed in multiple images at different distances — curves have been smoothed for visualization purposes. To illustrate how the underwater image formation model used in SUCRe fits these intensities, the black dotted lines show how different initial pixel intensities evolve with distance according to the estimated model.

In Figure 4.16, we track the hue value of a red color patch across different distances for distinct underwater color restoration methods. Notably, physics-based approaches such as Haze-Lines, Gaussian Sea-thru, and SUCRe exhibit better color recovery for distant elements compared to image processing-based methods. This difference may be attributed to the incorporation of distance-dependent underwater light propagation phenomena in physics-based methods. It is also notable that while single-view methods seem constrained by the 8-bits quantization for distant patches, SUCRe effectively overcomes this limitation by

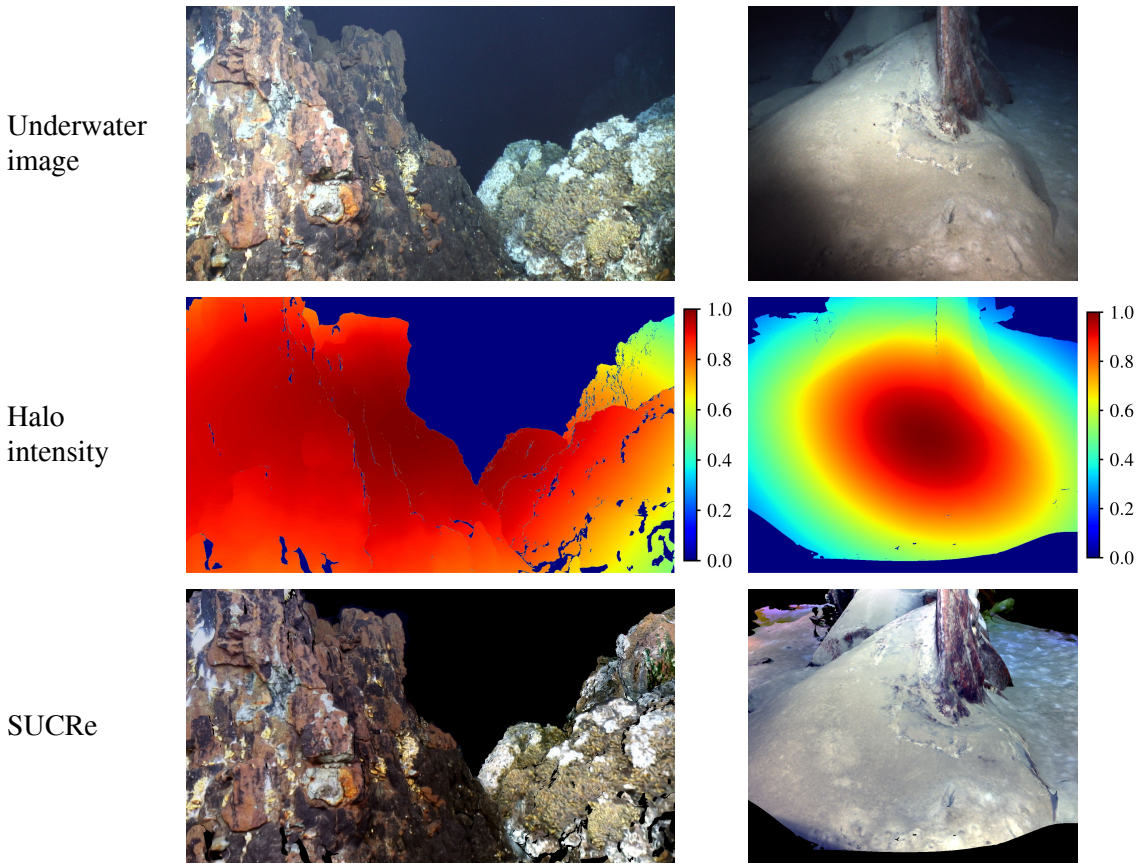


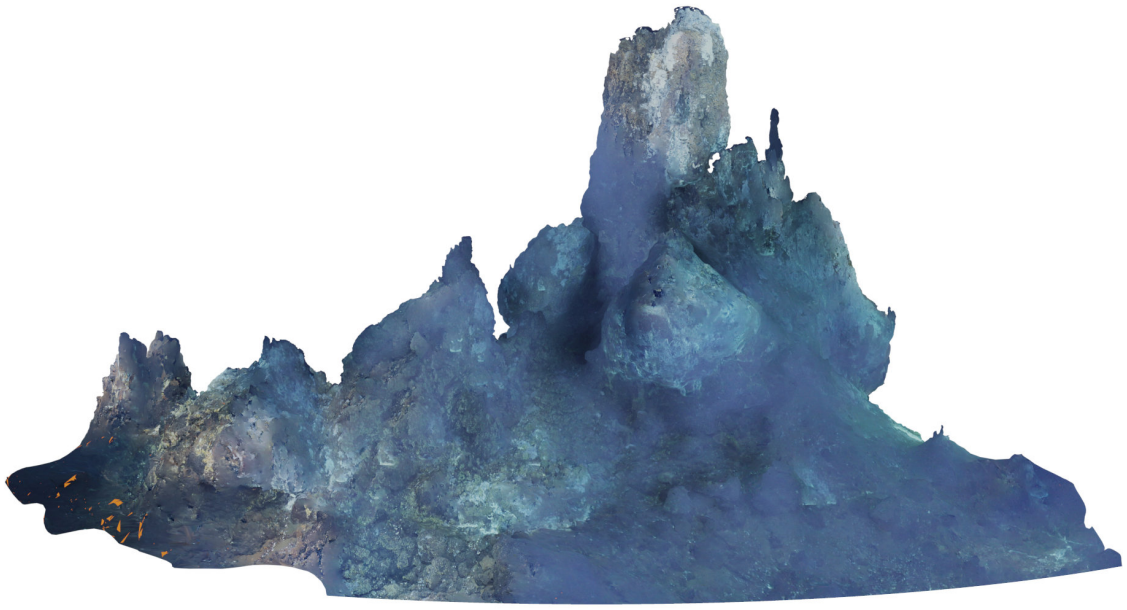
Figure 4.18: **Estimation of the light pattern** projected by the ROV on the scene. Modeling the ROV's artificial lighting system enable the correction of the vignetting effect and unbalanced illumination when using SUCRe.

leveraging closer observations in its optimization process. Moreover, SUCRe delivers consistent restoration results across varying observation ranges.

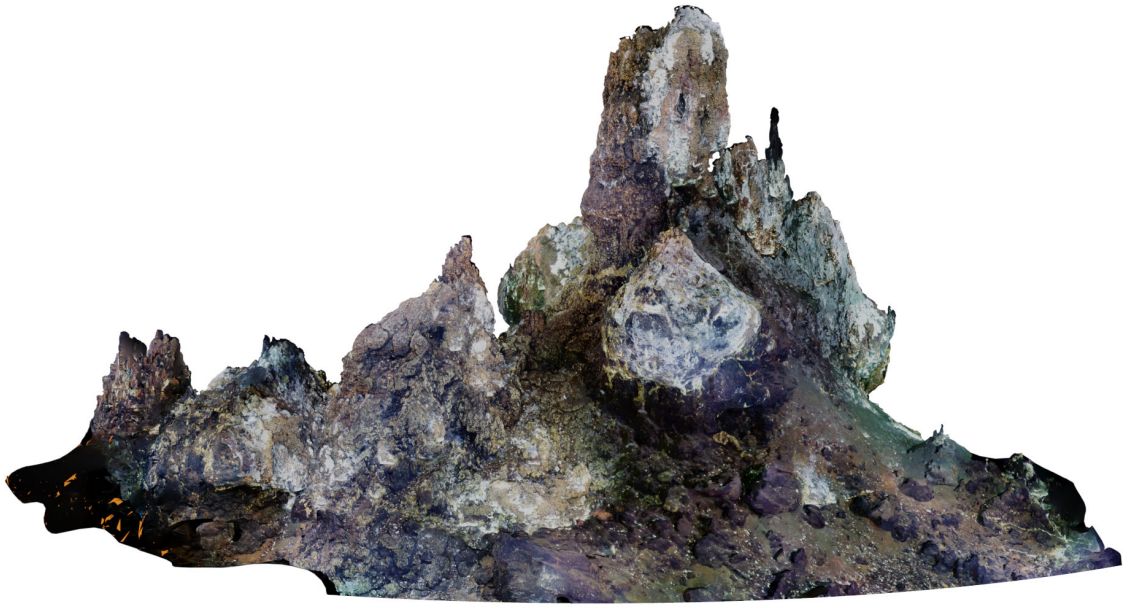
In Figure 4.17, we showcase how the parameters estimated using SUCRe align with pixel intensities observed at varying distances. Despite noisy observations, the model follows the general trend of pixel intensity attenuation.

The light pattern estimated by minimizing Eq. (4.44) is illustrated on two deep-sea images in Figure 4.18. By taking into account the vignetting effect created by the ROV's artificial lighting system, SUCRe effectively rectifies the uneven light distribution in the original image. From the two showcased light patterns, we observe that different artificial lighting systems result in diverse scene illuminations. This is an important factor to consider for the application of long-term visual localization algorithms to deep-sea environments.

Finally, Figure 4.19 illustrates that applying our approach to restore underwater images yields significant enhancements when texturing a 3D mesh, including finer details and coherent colors.



(a) 3D model textured with underwater images.



(b) 3D model textured with SUCRe images.

Figure 4.19: **Texturing the Eiffel Tower hydrothermal vent** 3D model with images restored using SUCRe results in a final model with improved visual quality, including finer details and more accurate colors compared to the original model.

4.5 Conclusion

In this chapter, we have formulated and introduced two novel methods designed to restore the colors of underwater images, alleviating the effects of water on light propagation. Our exploration reveals that underwater color restoration methods have much to benefit from leveraging multiple observations of the scene. This is evidenced by the capacity of SUCRe to uncover colors that might be nearly imper-

ceptible in the original image and to deliver more accurate and consistent color representations across elements at varying distances from the sensor. Furthermore, we demonstrated that restoration methods have the potential to improve visual localization algorithms.

The development and analysis of these methods provided valuable insights into the potential sources of variability that may impair long-term visual localization algorithms in deep-sea environments. For instance, we discovered limitations in working with 8-bit images. We also revealed that artificial lighting systems not only tie the scene's appearance to the vehicle's location but also that different lighting systems produce diverse illumination patterns.

Considering these findings, it is important to acknowledge that the presented approaches are limited by their processing time and their requirement of 3D scene information. These factors render them unsuitable as a preprocessing step for real-time underwater visual localization applications. Nevertheless, SUCRe holds promise for generating extensive datasets with reference images from real-world acquisitions, serving as valuable training data for real-time underwater color restoration neural networks.

As a perspective, it would be interesting to explore the color restoration of image pairs acquired using a calibrated stereo rig. Using stereo cameras could enable the color restoration of non-static elements, as they acquire temporally synchronized multi-view observations of the scene. Disparity map estimation networks, such as HITNet (Tankovich et al., 2021), could then be used to retrieve 3D information of the scene and allow the full use of SUCRe using only two images. Moreover, this approach has the potential of running in real time.

Part of the work presented in this Chapter has been accepted to the *International Conference on 3D Vision* (Boittiaux et al., 2024).

Chapter 5

Pose regression for deep learning

Contents

5.1 Introduction

In Chapters 3 and 4, we explained how computer vision algorithms face significant challenges when confronted with the specificities of underwater images. To address these challenges, Chapter 4 investigated the inversion of underwater image formation models, to compensate for the effect of water on light propagation. While we have shown that these methods have the potential to improve

underwater visual localization, the presented approaches require to have access to 3D information about the scene. Yet, in the context of visual localization, this information is not directly available from a single acquired image.

In pursuit of alternative solutions, we turn to deep learning, that has the potential to be robust to the variability produced by these underwater phenomena. Ideally, we would have access to a large quantity of data, enabling to fine-tune existing neural networks on underwater images. However, since multiannual underwater data is scarce, we took an interest in the neural networks that are trained for each scene individually (Kendall et al., 2015; Kendall and Cipolla, 2017; Brachmann et al., 2017). These neural networks can be seen as a function — they map an image to a 6DoF pose. They are trained on sets of images and their corresponding ground truth poses. A pivotal aspect of training these neural networks lies in their loss functions, that defines the error between the pose estimated by the network and the ground truth pose. These loss functions have the challenging task of embedding a pose error in $SE(3)$ into a single scalar, thus enabling the use of gradient descent.

In this chapter, we set out to address the fundamental problem of embedding an error between two camera poses into a single differentiable scalar. We aim to effectively balance the contributions of rotation and translation components within this final error. To this extent, this chapter makes the following contributions:

- We conduct a comprehensive review of existing loss functions designed for camera pose regression. This investigation not only highlights the strengths of these diverse loss functions but also underscores their limitations. These limitations can manifest as constraints on parameter tuning, lack of physical interpretability, or restrictions in their applicability to deep learning frameworks.

- Building upon this review, we introduce a novel loss function based on the homography principle. This loss function seeks to approximate the traditional reprojection error by representing the observed scene as planes rather than 3D points.

- We assess the performance of the proposed loss function in comparison to other established loss functions using two popular visual localization datasets (Shotton et al., 2013; Kendall et al., 2015). We present and discuss the results obtained from these datasets, which encompass both indoor and outdoor scenarios and employ different methods to generate ground truth data, *i.e.*, SfM and depth-based SLAM.

This chapter is organized as follows. First, Section 5.2 makes a review of existing loss functions and present their characteristics. Then, Section 5.3 presents

the Homography loss, a novel loss function for camera pose regression. Finally, Section 5.4 evaluate its performance compared to other losses.

5.2 Existing functions for camera pose regression

Visual localization aims at finding the 6DoF pose of a camera within a specified reference frame. In this chapter, we use either quaternions or their corresponding rotation matrices to represent the cameras' orientation in 3D space. Thus, we denote camera poses as $[\boldsymbol{q}|\boldsymbol{R},\boldsymbol{t}]$, where $\boldsymbol{q} \in SO(3)$ represents the quaternion vector describing the camera's rotation $\boldsymbol{R} \in SO(3)$ in the reference frame, and $\boldsymbol{t} \in \mathbb{R}^3$ represents the camera's 3D position within the reference frame. In the context of end-to-end deep learning methods, a central element when training neural networks is the design of a loss function. For visual localization applications, this loss needs to embed into a scalar the error between an estimated pose $[\hat{\boldsymbol{q}}|\hat{\boldsymbol{R}},\hat{\boldsymbol{t}}]$ and its corresponding ground truth $[\boldsymbol{q}|\boldsymbol{R},\boldsymbol{t}]$. In this section we make a review of loss functions that have been used to train these end-to-end pose regression models and present their characteristics. Unless specified otherwise, the norm of a vector $\|\cdot\|$ refers to its euclidean norm.

5.2.1 Loss functions

PoseNet. In PoseNet, rotation and translation errors of the estimated pose are weighted using a scale factor (Kendall et al., 2015). This enables to embed both rotation and translation errors into a single differentiable scalar. The rotation error between two camera poses is defined as the euclidean norm of the difference between the quaternions representing the cameras' rotations. The translation error is defined as the euclidean distance between the cameras' positions. This leads to the following loss function:

$$\mathcal{L}_{\mathrm{P}} = \left\|\hat{\boldsymbol{t}} - \boldsymbol{t}\right\| + \lambda \left\|\hat{\boldsymbol{q}} - \frac{\boldsymbol{q}}{\|\boldsymbol{q}\|}\right\|, \tag{5.1}$$

where λ is the positive scale factor that weights the importance of the rotation error over the translation error.

Homoscedastic. Similarly to PoseNet, this loss weights translation and rotation errors (Kendall and Cipolla, 2017). In contrast to PoseNet loss, it tries to reach an optimal balance between rotation and translation errors by optimizing global scalars $\hat{s}_t$ and $\hat{s}_q$ through backpropagation of the following loss function:

$$\mathcal{L}_{\mathrm{HU}} = \left\|\hat{\boldsymbol{t}} - \boldsymbol{t}\right\|_1 e^{-\hat{s}_t} + \hat{s}_t + \left\|\boldsymbol{q} - \frac{\hat{\boldsymbol{q}}}{\|\hat{\boldsymbol{q}}\|}\right\|_1 e^{-\hat{s}_q} + \hat{s}_q, \tag{5.2}$$

where $\hat{s}_t$ and $\hat{s}_q$ respectively represent the natural logarithm of the translational and rotational homoscedastic task noise variance.

Geometric reprojection. This loss function is derived from the classical reprojection error, used for instance in bundle adjustment (Kendall and Cipolla, 2017). In contrast with other losses, it requires known 3D points observed by the camera in addition to the camera's 6DoF pose. This loss computes the distance between the projection of known 3D points in the ground truth image plane and the projection of these 3D points in the estimated image plane. It is defined as follows:

$$\mathcal{L}_{\mathrm{G}} = \frac{1}{|\mathcal{G}|} \sum_{\mathbf{X}_p \in \mathcal{G}} \left\| \pi \left(\boldsymbol{R}\mathbf{X}_p + \boldsymbol{t} \right) - \pi \left(\hat{\boldsymbol{R}}\mathbf{X}_p + \hat{\boldsymbol{t}} \right) \right\|_1 , \tag{5.3}$$

where $\mathcal{G}$ is the subset of 3D points observed by the current view, and $[\boldsymbol{R}, \boldsymbol{t}]$ and $[\hat{\boldsymbol{R}}, \hat{\boldsymbol{t}}]$ are expressed from the reference frame to the camera frame.

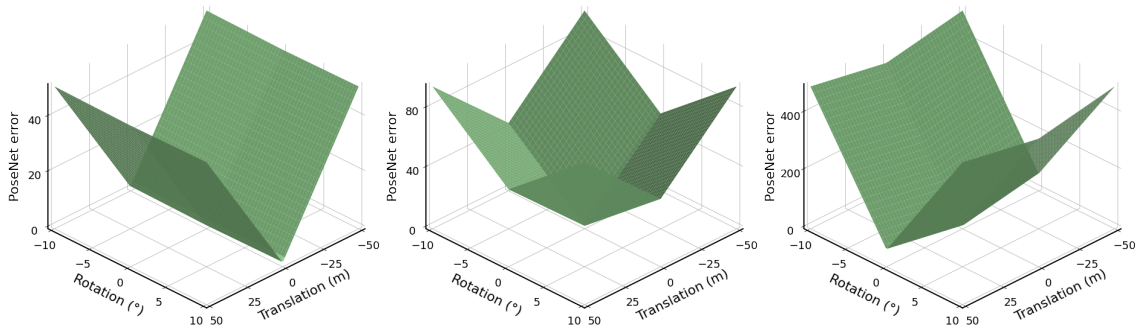
MaxError. DSAC (Brachmann et al., 2017; Brachmann and Rother, 2022) training relies on the following loss function that we will refer to as MaxError:

$$\mathcal{L}_{\mathrm{ME}} = \max \left(\measuredangle(\boldsymbol{q}, \hat{\boldsymbol{q}}),\ \left\| \hat{\boldsymbol{t}} - \boldsymbol{t} \right\| \right) , \tag{5.4}$$

where $\measuredangle(\boldsymbol{q}, \hat{\boldsymbol{q}})$ is the measured angle between rotations in 3D space induced by $\boldsymbol{q}$ and $\hat{\boldsymbol{q}}$. This angle is expressed in degrees, and $\boldsymbol{t}$ and $\hat{\boldsymbol{t}}$ are expressed in centimeters. It is important to note that the DSAC method is a more complex end-to-end visual localization pipeline than a simple CNN inference. In this work, we will only evaluate the performance of the loss within a simpler end-to-end pose regressor.

5.2.2 Losses characteristics

PoseNet. One limitation of the PoseNet loss function lies in the challenge of determining an appropriate value for λ. Specifically, PoseNet rotation error is defined as the norm of the difference between $\hat{\boldsymbol{q}}$ and $\boldsymbol{q}$ unit quaternions, a metric that does not relate to an intuitive geometric phenomenon, making it difficult to compare rotation and translation components. Additionally, all poses are optimized with the same relative weight between translational and rotational errors, no matter what the camera observes. Figure 5.1 shows the influence of λ on PoseNet's loss function. A small value induces strong translational gradients but a flat profile in orientation, whereas high λ values assign importance to rotation over flat translation evolution. Given a scene, a well chosen λ allows optimizing the parameters in all dimensions. PoseNet is a muti-objective loss and presents the common problems encountered in such setting. The λ selection is not obvious and needs to be determined through trial and error. Even with a properly set λ, stochastic optimization may converge to different optima on the Pareto front.



(a) $\lambda = 5$. (b) $\lambda = 500$. (c) $\lambda = 5{,}000$.

Figure 5.1: **Effect of λ on the PoseNet loss** described in Eq. (5.1) for a particular scene. (a) Low λ values lead to high variation of error with respect to translation and little change with respect to rotation. (b) Well-chosen λ leads to a clear local minimum around the optimal parameters. (c) A large λ induces a small variation of translation.

MaxError. The MaxError loss function solves the compromise between rotation and translation errors by heuristically fixing a chosen scale between them, *i.e.*, the rotation error is in degrees and the translation error is in centimeters. While the scale is physically interpretable, it shares the other issues with PoseNet related to the multi-objective optimization. In this case, the problem is tackled by minimizing the highest error at each step. Similarly to PoseNet, it weights rotation and translation error with the same scale factor for every frame.

Homoscedastic uncertainty. The Homoscedastic loss reveals characteristics similar to PoseNet and MaxError losses. However, its parameters are more robust to the initialization, since they are optimized during training.

Geometric reprojection loss. The Geometric reprojection loss mimics the reprojection error, hence implicitly solving the weighting problem associated with rotation and translation errors. The reprojection error consists in measuring the 2D distance between the projection of a set of 3D points into two camera views. If the poses are identical, then the points are superimposed. As we previously discussed, particularly in Section 3.4.3, this error has been widely used to solve 3D computer vision problems, such as bundle adjustment. Its physical meaning is easy to understand because it can be represented graphically in the image plane. Furthermore, optimizing this specific error is highly relevant when we aim to localize camera poses for which the ground truth was generated using SfM, as SfM also minimizes the reprojection error.

However, its use in deep learning models is more cumbersome for multiple reasons. It relies on the choice of the 3D points that are projected on the image

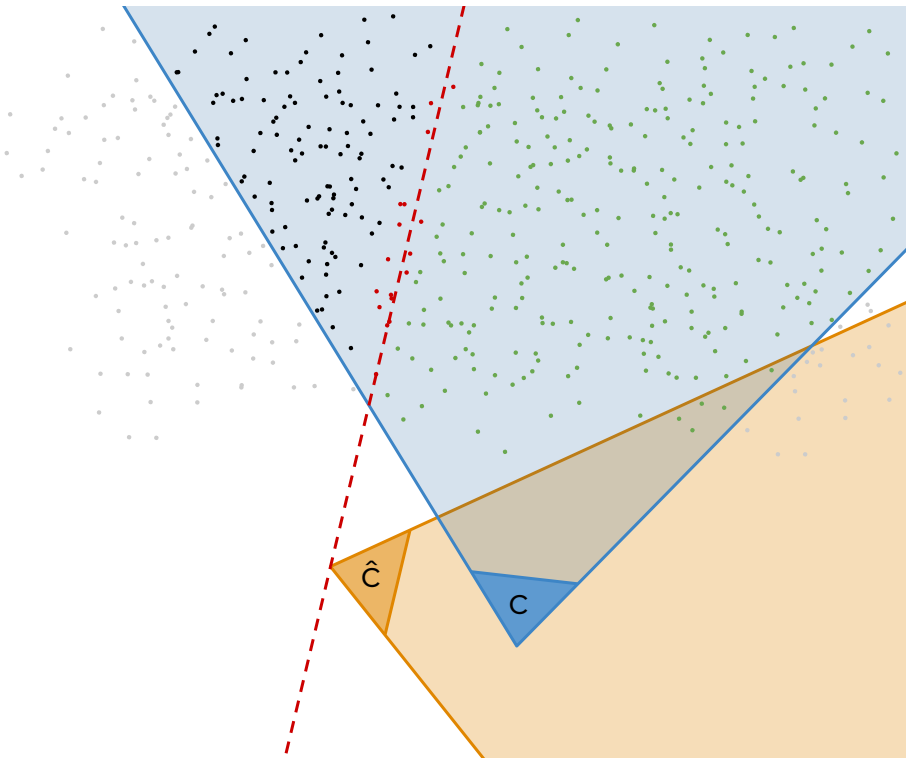


Figure 5.2: **Reprojection error limits.** Top view representation of the ground truth (blue) and estimated (orange) camera views and the scene point cloud. The grey points are outside the field of view of the ground truth. The green points are in front of the image planes of the two cameras. The dashed red line represents the (x, y) plane of the estimated camera frame. The red dots that are close to this plane are projected to infinity through the pinhole projection model. The black dots project backward from the image plane of the estimated camera.

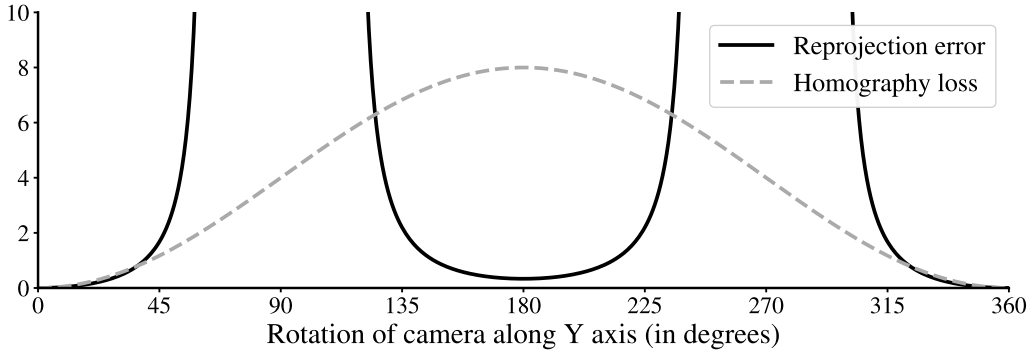


Figure 5.3: **Evolution of the reprojection error and our Homography loss** when the relative pose of the ground truth and estimated cameras varies with a rotational movement along the Y axis.

plane. Depending on the method used to estimate the camera ground truth poses, 3D points of the scene may not be available. This issue can be mitigated by triangulating 3D points from the camera poses and 2D-2D matches if such matches are feasible. A more substantial problem for this error arises during the neural network's initialization. At initialization, the poses predicted by the neural network are usually initialized around an arbitrary value, often far from the ground truth. As depicted in Figure 5.2, this means that some 3D points can be projected to infinity if they lie in the camera's (x, y) plane. Additionally, as shown in Figure 5.3, the reprojection error may also lead to a local minimum when 3D points are projected onto the backside of the image plane. To overcome these problems, the network is usually initialized by first training it with another loss function for a few iterations. In addition, if a point is projected to infinity during the optimization, it results in an infinite loss, causing the model to diverge. In practice, this problem is often addressed by clipping reprojection error distances that exceed a threshold. However, in doing so, all clipped points lay on a flat maximum with a zero-valued gradient, therefore not contributing to the optimization.

Alternatively, the proposed Homography loss function tackles this issue by approximating the observed scene with a set of virtual parallel planes. It offers a competitive accuracy and a high numerical stability making a simple single step learning possible.

5.3 A homography-based loss function for camera pose regression

5.3.1 Motivation

The purpose of this work is to adapt the reprojection error for camera pose regression within the domain of deep learning applications, leveraging its benefits while mitigating its drawbacks.

In this study, we propose that the 3D points used to quantify a pose error, like the reprojection error, do not necessarily need to be real points, but can instead be a set of designated virtual points. To avoid issues associated with infinite errors, one approach could be to regularly sample virtual 3D points positioned in front of both camera image planes. Nevertheless, as the poses become more distant from each other, shared 3D observations become scarce and virtual point sampling becomes increasingly difficult. Building on this line of thinking and eliminating the challenges related to point selection, we divide the scene into planes that encompass an infinite number of these points. These planes induce homographies between the ground truth and estimated camera views. A homography refers to the transformation of a plane in 3D space from one projective view to the other.

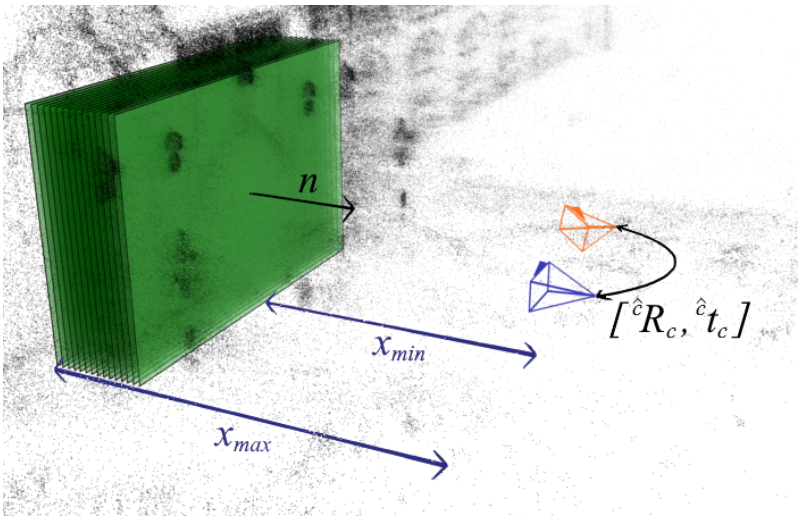


Figure 5.4: **Illustration of the proposed Homography loss function.** We replace the 3D points (black) observed by the ground truth camera (blue) by a set of parallel virtual planes (green). The planes' normal $\boldsymbol{n}$ and the ground truth camera's optical axis are co-linear. For a given plane, we express our error directly in the homography induced by this plane between the ground truth and the estimated (orange) camera poses. We then integrate this error between $x_{\min}$ and $x_{\max}$ distances. Planes are infinite, but for the sake of visualization they are represented as rectangles.

In other terms, given a 3D plane observed in two different camera views, a homography maps any 2D observation of that plane from one view to the other. For a given plane, we can compute our error directly in terms of the homography induced by this plane between the ground truth and estimated camera views. As illustrated in Figure 5.4, we then integrate this error for all possible planes between two given boundaries, $x_{\min}$ and $x_{\max}$, representing the distance of the observed scene.

5.3.2 Method

In this section, we detail how we can approximate the reprojection error using homographies and overcome the issues associated with the Geometric reprojection loss. First, let us establish the computation of a homography matrix from rotation and translation components (Hartley and Zisserman, 2003):

$$ {}^{\hat{c}}\boldsymbol{H}_c = {}^{\hat{c}}\boldsymbol{R}_c - \frac{{}^{\hat{c}}\boldsymbol{t}_c \, {}^{c}\boldsymbol{n}^T}{x}, \tag{5.5} $$

where ${}^{c}\boldsymbol{n}$ is the normal to the considered plane expressed in the ground truth camera frame, x is the distance to that plane and $[{}^{\hat{c}}\boldsymbol{R}_c, {}^{\hat{c}}\boldsymbol{t}_c]$ are the rotation and translation of the ground truth camera expressed in the estimated camera frame. For

the sake of clarity, unless specified otherwise, all future homographies, rotations and translations refer to transformations from the ground truth to the estimated camera frame. More specifically: $\boldsymbol{H} = {}^{\hat{c}}\boldsymbol{H}_c$, $\boldsymbol{R} = {}^{\hat{c}}\boldsymbol{R}_c$ and $\boldsymbol{t} = {}^{\hat{c}}\boldsymbol{t}_c$. Additionally, we note $\boldsymbol{n} = {}^{c}\boldsymbol{n}$. Let $\mathbf{X}$ be a 3D point observed by two cameras. Let $\mathbf{x} = [u, v, 1]^T$ and $\mathbf{x}' = [u', v', 1]^T$ be the 2D homogeneous representations of the projection of $\mathbf{X}$ in the ground truth and estimated camera views, respectively. The reprojection error of $\mathbf{x}$ is defined as:

$$\begin{aligned} \text{repr}(\mathbf{x}) &= (u - u')^2 + (v - v')^2 && (5.6)\\ &= (\mathbf{x} - \mathbf{x}')^T(\mathbf{x} - \mathbf{x}') && (5.7) \end{aligned}$$

We now assume that $\mathbf{X}$ lies in a plane that induces a homography $\boldsymbol{H}$ between the two camera views. We want to retrieve the reprojection error by expressing $\mathbf{x}'$ in terms of $\boldsymbol{H}\mathbf{x}$. Let us explicit $\boldsymbol{H}$ components:

$$\boldsymbol{H} = \begin{pmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & 1 \end{pmatrix} \tag{5.8}$$

We note $\mathbf{x}'$ the 2D homogeneous point resulting from $\boldsymbol{H}\mathbf{x}$:

$$\mathbf{x}' = \boldsymbol{H}\mathbf{x} = \begin{bmatrix} u' \\ v' \\ s \end{bmatrix} \tag{5.9}$$

where $s = h_{31}u + h_{32}v + 1$. By the definition of the homography, $\boldsymbol{H}\mathbf{x} \sim \mathbf{x}'$ in homogeneous coordinates. Thus, in the euclidean space:

$$\mathbf{x}' = \frac{\boldsymbol{H}\mathbf{x}}{s} \tag{5.10}$$

When replacing Eq. (5.10) into Eq. (5.7), we can express the reprojection error in terms of $\boldsymbol{H}$:

$$\begin{aligned} \text{repr}(\mathbf{x}) &= \left(\mathbf{x} - \frac{\boldsymbol{H}\mathbf{x}}{s}\right)^T \left(\mathbf{x} - \frac{\boldsymbol{H}\mathbf{x}}{s}\right) && (5.11)\\ &= \mathbf{x}^T \left(\mathbf{I} - \frac{\boldsymbol{H} + \boldsymbol{H}^T}{s} + \frac{\boldsymbol{H}^T\boldsymbol{H}}{s^2}\right)\mathbf{x} && (5.12) \end{aligned}$$

where $\mathbf{I}$ is the identity matrix. As the estimated pose tends towards the ground truth pose, s tends towards 1. We will use the approximation $s \approx 1$ to simplify Eq. (5.12). This way, our homographic error will tend to the reprojection error when poses are close. Then, Eq. (5.12) becomes:

$$\text{repr}(\mathbf{x}) = \mathbf{x}^T(\mathbf{I} - \boldsymbol{H})^T(\mathbf{I} - \boldsymbol{H})\,\mathbf{x}. \tag{5.13}$$

Since our error is a scalar, it is equal to its trace:

$$\mathrm{repr}(\mathbf{x}) = \mathrm{Tr}\left(\mathbf{x}^T(\mathbf{I}-\boldsymbol{H})^T(\mathbf{I}-\boldsymbol{H})\,\mathbf{x}\right). \tag{5.14}$$

Then, we can use the cyclic property of the trace to isolate $\mathbf{x}$:

$$\mathrm{repr}(\mathbf{x}) = \mathrm{Tr}\left(\mathbf{x}\mathbf{x}^T(\mathbf{I}-\boldsymbol{H})^T(\mathbf{I}-\boldsymbol{H})\right) \tag{5.15}$$

with

$$\mathbf{x}\mathbf{x}^T = \begin{pmatrix} u^2 & uv & u \\ vu & v^2 & v \\ u & v & 1 \end{pmatrix}. \tag{5.16}$$

While we have expressed $\mathrm{repr}(\mathbf{x})$ in terms of $\boldsymbol{H}$, this error still relies on specific 2D points in the camera view. As we do not want our loss to rely on any specific point, we integrate our error on all 2D points of our sensor. Let W and H be the respective width and height of our sensor, this point integration can be computed as follows:

$$\int_{-\mathrm{W}/2}^{\mathrm{W}/2}\int_{-\mathrm{H}/2}^{\mathrm{H}/2} \mathrm{Tr}\left(\mathbf{x}\mathbf{x}^T(\mathbf{I}-\boldsymbol{H})^T(\mathbf{I}-\boldsymbol{H})\right)\mathrm{d}u\,\mathrm{d}v \tag{5.17}$$

$$= \mathrm{Tr}\left(\begin{pmatrix} \frac{\mathrm{HW}^3}{12} & 0 & 0 \\ 0 & \frac{\mathrm{WH}^3}{12} & 0 \\ 0 & 0 & \mathrm{WH} \end{pmatrix}(\mathbf{I}-\boldsymbol{H})^T(\mathbf{I}-\boldsymbol{H})\right). \tag{5.18}$$

This results in a diagonal matrix simply weighting the dimensions of the reprojection according to the size of the sensor. As we want our loss to be generic to the size of the sensor, we will simply drop this matrix. We finally have our homographic error which, by definition, because $(\mathbf{I}-\mathbf{H})$ is real, is equivalent to a Frobenius norm:

$$\mathrm{Tr}\left((\mathbf{I}-\boldsymbol{H})^T(\mathbf{I}-\boldsymbol{H})\right) = \|\mathbf{I}-\boldsymbol{H}\|_F^2. \tag{5.19}$$

We further extend the definition of our single plane homographic error to a full region between two parallel planes. We integrate Eq. (5.19) over the planes within a given range of distances and along a particular direction. Let $x_{\min}$ and $x_{\max}$ be the minimum and maximum distances of the planes containing our observations. We introduce the analytic form of our Homography loss function:

$$\mathcal{L}_{\mathrm{H}} = \frac{1}{x_{\max}-x_{\min}}\int_{x_{\min}}^{x_{\max}} \|\mathbf{I}-\boldsymbol{H}\|_F^2\,\mathrm{d}x. \tag{5.20}$$

Note that we normalize the loss by the region of the considered scene dimension $(x_{\max}-x_{\min})$. This is because every frame has its own distance range of observations. By normalizing, we ensure that each frame cost is on the same scale. We can then solve the integral by substitution of Eq. (5.5) in Eq. (5.20) resulting in our final loss function:

$$\mathcal{L}_{\mathrm{H}} = \mathrm{Tr}\left(\mathbf{A}+\mathbf{B}\,\frac{\log\left(x_{\max}/x_{\min}\right)}{x_{\max}-x_{\min}} + \frac{\mathbf{C}}{x_{\min}\cdot x_{\max}}\right), \tag{5.21}$$

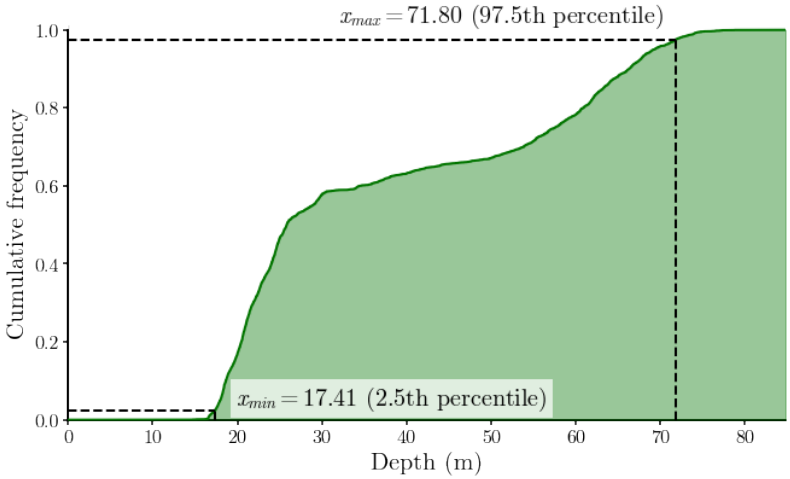


Figure 5.5: **Cumulative histogram of scene points' depths.** $x_{\min}$ and $x_{\max}$ depth values can be selected as the $2.5th$ and $97.5th$ percentile.

with

$$\mathbf{A} = (\bar{\mathbf{I}} - \boldsymbol{R})(\bar{\mathbf{I}} - \boldsymbol{R})^T, \tag{5.22}$$

$$\mathbf{B} = (\mathbf{I} - \boldsymbol{R})\boldsymbol{n}\boldsymbol{t}^T + \left((\mathbf{I} - \boldsymbol{R})\boldsymbol{n}\boldsymbol{t}^T\right)^T, \tag{5.23}$$

$$\mathbf{C} = \boldsymbol{t}\boldsymbol{n}^T \left(\boldsymbol{t}\boldsymbol{n}^T\right)^T. \tag{5.24}$$

The details of this integration are available in Appendix D.

In conclusion, Eqs. (5.21) to (5.24) present the closed form solution of our proposed Homography loss function.

5.3.3 Implementation

When examining Eq. (5.21), we can identify the specific parameters upon which our loss function depends. $\boldsymbol{R}$ and $\boldsymbol{t}$ are directly computed from ground truth and estimated poses. We set $\boldsymbol{n} = [0, 0, -1]^T$, so that all homographies are induced by planes parallel to the ground truth sensor, as if they faced the camera. The remaining two parameters, $x_{\min}$ and $x_{\max}$, represent the minimum and maximum distances of these planes to the ground truth sensor. In this work, we introduce two distinct approaches for configuring these parameters, inspired by different uses of the loss, and leading to different implementations.

Local Homography loss. The first approach best approximates the reprojection error, but necessitates known 3D scene points. Given this information, the two parameters can be computed for each frame individually. For every frame, we compute a depth histogram of its 3D observations. We then set its $x_{\min}$ and $x_{\max}$ parameters as a given percentile of the depth distribution. This is illustrated by Figure 5.5. We refer to this approach as the Local Homography loss function.

Global Homography loss. The other way of setting these parameters requires less information about the 3D scene. For the Global Homography loss, we set $x_{\min}$ and $x_{\max}$ parameters globally for all images. In this case, rotation and translation errors have the same weight for all the images in the training set. However, in contrast with PoseNet and Homoscedastic losses, the parameters have an intuitive physical meaning. Note that if 3D data is available, it is possible to set global $x_{\min}$ and $x_{\max}$ from a global depth distribution histogram.

5.3.4 Homography loss properties

An essential consideration when designing a loss function is to ensure that this loss does not have undesired global minima. As such, this section aims to demonstrate that the proposed loss function only reaches its minimum when poses are superimposed, *i.e.*, $\boldsymbol{R} = \mathbf{I}$ and $\boldsymbol{t} = \mathbf{0}_3$. It is important to note that our loss considers planes parallel to the sensor's image plane. Consequently, this proof only holds for $\boldsymbol{n} = [0, 0, -1]^T$.

As described by Eq. (5.20), our loss function $\mathcal{L}_{\mathrm{H}}$ relies on the squared Frobenius norm between a homography and the identity matrix. Given any matrix $\boldsymbol{M} \in \mathbb{R}^{3\times3}$, its squared Frobenius norm is defined as the sum of the square of all its elements:

$$\|\boldsymbol{M}\|_F^2 = \sum_{a=1}^{3}\sum_{b=1}^{3} \boldsymbol{M}_{ab}^2. \tag{5.25}$$

Considering this, it is evident that $\|\mathbf{I} - \boldsymbol{H}\|_F^2$ can only be positive. It is also clear that its integral over a positive interval is necessarily positive. Consequently, we can infer from Eq. (5.20) that $\mathcal{L}_{\mathrm{H}} \geq 0$. Moreover, we also know from Eq. (5.5) that the homography between two superimposed views is the identity matrix, leading our loss function to be equal to zero:

$$[\boldsymbol{R}, \boldsymbol{t}] = [\mathbf{I}, \mathbf{0}_3] \Rightarrow \boldsymbol{H} = \mathbf{I} \Rightarrow \mathcal{L}_{\mathrm{H}} = 0. \tag{5.26}$$

As we know that $\mathcal{L}_{\mathrm{H}} \geq 0$, Eq. (5.26) implies that the minimum value of our loss is zero. Still from the definition of the Frobenius norm in Eq. (5.25), we can deduce that our loss reaches its minimum only when the homography is the identity matrix:

$$\mathcal{L}_{\mathrm{H}} = 0 \Leftrightarrow \boldsymbol{H} = \mathbf{I}. \tag{5.27}$$

To complete our proof, we only need to show that a homography is equal to the identity matrix only if poses are superimposed, *i.e.*, we need to prove that: $\boldsymbol{H} = \mathbf{I} \Rightarrow [\boldsymbol{R}, \boldsymbol{t}] = [\mathbf{I}, \mathbf{0}_3]$. To achieve this, we will need to leverage the properties of $SO(3)$. Using the definition of the homography described in Eq. (5.5), we can express $\boldsymbol{R}$ when $\boldsymbol{H} = \mathbf{I}$:

$$\boldsymbol{R} = \mathbf{I} + \frac{\boldsymbol{t}\boldsymbol{n}^T}{x}. \tag{5.28}$$

Because $\boldsymbol{R} \in SO(3)$, it has the following property:

$$\boldsymbol{R}\boldsymbol{R}^T = \mathbb{I}. \tag{5.29}$$

We can use this property to constrain $\boldsymbol{t}$:

$$\left(\mathbf{I} + \frac{\boldsymbol{t}\boldsymbol{n}^T}{x}\right)\left(\mathbf{I} + \frac{\boldsymbol{t}\boldsymbol{n}^T}{x}\right)^T = \mathbf{I} \Leftrightarrow \boldsymbol{t}\boldsymbol{n}^T + \boldsymbol{n}\boldsymbol{t}^T + \frac{\boldsymbol{t}\boldsymbol{n}^T\boldsymbol{n}\boldsymbol{t}^T}{x} = \mathbf{0}_{3\times 3} \tag{5.30}$$

We fix $\boldsymbol{n} = [0, 0, -1]^T$ and note $\boldsymbol{t} = [t_x, t_y, t_z]^T$. We can decompose Eq. (5.30):

$$\boldsymbol{t}\boldsymbol{n}^T = \begin{bmatrix} 0 & 0 & -t_x \\ 0 & 0 & -t_y \\ 0 & 0 & -t_z \end{bmatrix} \tag{5.31}$$

$$\boldsymbol{n}\boldsymbol{t}^T = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ -t_x & -t_y & -t_z \end{bmatrix} \tag{5.32}$$

$$\boldsymbol{t}\boldsymbol{n}^T\boldsymbol{n}\boldsymbol{t}^T = \begin{bmatrix} t_x^2 & t_x t_y & t_x t_z \\ t_y t_x & t_y^2 & t_y t_z \\ t_z t_x & t_z t_y & t_z^2 \end{bmatrix}. \tag{5.33}$$

From Eqs. (5.30) to (5.33) we can deduce that:

$$t_x = 0, \tag{5.34}$$

$$t_y = 0. \tag{5.35}$$

And t_z has two possible values:

$$\begin{cases} t_z = 0 \\ t_z = 2x \end{cases}. \tag{5.36}$$

We can further constrain $\boldsymbol{t}$ by using another property of $SO(3)$, that is, $\det(\boldsymbol{R}) = 1$. From Eq. (5.28), we can retrieve $\det(\boldsymbol{R})$ with $t_x = 0$, $t_y = 0$ and $x \neq 0$:

$$\det(\boldsymbol{R}) = \det\left(\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 - t_z/x \end{bmatrix}\right) = 1 - \frac{t_z}{x}. \tag{5.37}$$

By enforcing the aforementioned $SO(3)$ property, we can isolate a single solution for t_z:

$$\det(\boldsymbol{R}) = 1 \Rightarrow 1 - \frac{t_z}{x} = 1 \Rightarrow t_z = 0. \tag{5.38}$$

With Eqs. (5.34), (5.35) and (5.38), we have shown that:

$$\boldsymbol{H} = \mathbf{I} \Rightarrow \boldsymbol{t} = \mathbf{0}_3. \tag{5.39}$$

Finally, we can once again use the definition of the homography along with Eq. (5.39) to retrieve $\boldsymbol{R}$ when $\boldsymbol{H} = \mathbf{I}$:

$$\boldsymbol{R} = \mathbf{I} + \frac{\mathbf{0}_3 \boldsymbol{n}^T}{x} = \mathbf{I}. \tag{5.40}$$

From Eqs. (5.39) and (5.40), we can express that a homography is equal to the identity matrix only if poses are superimposed:

$$\boldsymbol{H} = \mathbf{I} \Rightarrow [\boldsymbol{R}, \boldsymbol{t}] = [\mathbf{I}, \mathbf{0}_3]. \tag{5.41}$$

By putting together Eqs. (5.26), (5.27) and (5.41), we can complete our proof:

$$\mathcal{L}_{\mathrm{H}} = 0 \Leftrightarrow \boldsymbol{H} = \mathbf{I} \Leftrightarrow [\boldsymbol{R}, \boldsymbol{t}] = [\mathbf{I}, \mathbf{0}_3]. \tag{5.42}$$

Our loss minimum is only reached when poses are superimposed.

5.3.5 Additional insights

In Section 5.3.2, we have approximated the reprojection error by replacing observed 3D points by planes. Yet, instead of considering one plane for every 3D point, our error integrates all planes between a minimum and a maximum distance of observation, reducing the observed scene to a slab, *i.e.*, a region between two parallel planes. In this integration, we implicitly consider that every plane has the same weight in the final error, independently of its distance of observation. However, in practice, the depth of observed 3D points is not distributed in a uniform manner. To take into account this non-uniform distribution, we can add a function $F(x)$ to the integral, that weights the importance of each plane according to the depth distribution of the scene:

$$\mathcal{L}'_{\mathrm{H}} = \int_{x_{\min}}^{x_{\max}} F(x) \left\| \mathbf{I} - \boldsymbol{H} \right\|_F^2 \mathrm{d}x. \tag{5.43}$$

We can then choose a probability distribution to approximate the depth distribution of the scene. For instance, we could approximate the scene's depth using a log-normal distribution. This log-normal's parameters, μ_x and σ_x, may be fitted on the observed scene's depth distribution. Then, by injecting the log-normal weighting into the integral, we obtain:

$$\mathcal{L}'_{\mathrm{H}} = \int_{0^+}^{+\infty} \mathrm{Lognormal}(\mu_x, \sigma_x)(x) \left\| \mathbf{I} - \boldsymbol{H} \right\|_F^2 \mathrm{d}x \tag{5.44}$$

$$= \mathrm{Tr}\left(\mathbf{A} + \mathbf{B} \exp\left(\frac{\sigma_x^2}{2} - \mu_x \right) + \mathbf{C} \exp(2\sigma_x^2 - 2\mu_x) \right). \tag{5.45}$$

In this scenario, we eliminate the need to normalize the final loss function, as the integral of a probability distribution is equal to one.

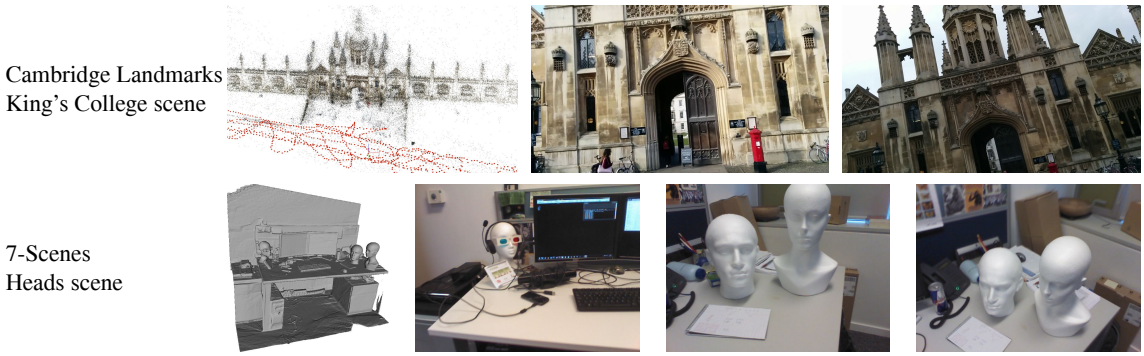


Figure 5.6: **Cambridge Landmarks and 7-Scenes datasets.** Cambridge Landmarks ground truth is obtained using SfM. 7-Scenes ground truth is obtained using RGB-D SLAM.

5.4 Experiments

To evaluate the performance of the proposed Homography loss in comparison to existing alternatives, we conduct a benchmarking study. We re-implement PoseNet (Kendall et al., 2015), Homoscedastic (Kendall and Cipolla, 2017), Geometric (Kendall and Cipolla, 2017) and MaxError (Brachmann et al., 2017; Brachmann and Rother, 2022) losses. Our experiments are conducted on the Cambridge Landmarks (Kendall et al., 2015) and 7-Scenes datasets (Shotton et al., 2013). Losses are evaluated using a pose regressor similar to PoseNet, *i.e.*, with a neural network directly inferring a 6DoF pose from an image.

5.4.1 Benchmark datasets

As discussed in Section 2.2.3, Brachmann et al. (2021) highlighted the importance of considering the method employed to construct ground truth data when evaluating visual localization algorithms. Some approaches or loss functions may be advantaged depending on the method employed to build the ground truth. For instance, as the Geometric reprojection loss minimizes the same error as SfM, it should benefit from SfM ground truths. To investigate this impact when comparing various loss functions, we conducted evaluations on two distinct datasets illustrated in Figure 5.6, each characterized by ground truth poses determined through different approaches. The Cambridge Landmarks dataset (Kendall et al., 2015) was created using SfM and comprises six outdoor scenes within the city of Cambridge. The 7-Scenes dataset (Shotton et al., 2013) was established using depth-based SLAM and encompasses seven diverse indoor scenes. All scenes in both datasets are visited several times, and train and test sequences consist of different visits.

5.4.2 Experimental setup

Pose regression model. Kendall et al. (2015) employed GoogLeNet (Szegedy et al., 2015) as the backbone for their pose regression model. They made modifications by replacing the network's classification head with two dense layers. The first dense layer has a feature size of 2048, while the second dense layer has a feature size of seven — three features for the translation component and four features for representing the quaternion that denotes the rotation component.

In our research, we use a MobileNetV2 (Sandler et al., 2018) backbone provided by PyTorch and proceed to the same replacement. This backbone was selected for its versatility. We load MobileNetV2 weights pretrained on ImageNet, which are readily available from the PyTorch Hub, and follow PyTorch's recommendations for normalizing input images. However, we deviated from Kendall et al. (2015) in one aspect. While in PoseNet the network is typically trained on random crops of resized images, we found during our experiments that this data augmentation technique had a detrimental effect on the results. We suggest that applying a random crop to the image artificially shifts the optical center of the camera, impairing the ability of the network to predict accurate pose estimates. Consequently, we do not use data augmentation when training the model.

These experiments aim to evaluate the performance of aforementioned loss functions using a straightforward end-to-end network, rather than on a more complete pipeline like DSAC (Brachmann et al., 2017). Our goal is to provide an alternative to existing pose regression loss functions, not to develop an entire visual localization pipeline. Consequently, this study focuses on comparing these loss functions using a single regression model to facilitate the comparison and reproducibility of the results. The application of this loss within other visual localization algorithms remains in the scope of future work.

Losses specifications. *PoseNet:* To compare our results with previous work, we fix λ = 500 (Kendall and Cipolla, 2017). *Homoscedastic:* We initialize the parameters as suggested by Kendall and Cipolla (2017), that is, $\hat{s}_t$ = 0.0 and $\hat{s}_q$ = -3.0. *Geometric reprojection:* As discussed in Section 5.2.2, we clip the reprojection distance at 100 to prevent the loss from diverging. *MaxError:* In the process of implementing DSAC loss function, we faced an issue where the estimated quaternion consistently converged towards the null vector. To adress this problem, we introduce a regularization term within the loss function. This additional term enforces the norm of the estimated quaternion to be equal to one: MSE $(\|\boldsymbol{q}\|, 1)$. *Homography:* For both the Local and Global Homography losses, we select $x_{\min}$ and $x_{\max}$ to be respectively the 2.5th and 97.5th percentiles of the observed points' depth distribution.

Training procedure. All models are trained using an Adam optimizer (Kingma and Ba, 2015) with a learning rate of 10^{-4}. The training process consists of 5,000 epochs with a batch size of 64. For each epoch, we drop the last batch if it is smaller. In our experiments, we observed that using an epsilon value of 10^{-14} for Adam, instead of the default 10^{-8} produces better results for Homography losses. This adjustment proves beneficial because, towards the end of optimization, our losses tend to converge to very low values $\sim 10^{-4}$. As for the Geometric reprojection loss, we first initialize the network by initially training it for 500 epochs using the Homoscedastic loss, as suggested by Kendall and Cipolla (2017).

Scene	PoseNet	Homoscedastic	DSAC	Reprojection	Global Homography	Local Homography
	Mean reprojection distance in pixels ↓ Percentage of images localized within (2m, 2°) / (3m, 5°) ↑					
Great Court	**118** **13** / **36.4**	148 7.6 / 26.6	624 0.4 / 2.8	183 1.1 / 8.4	235 0.7 / 6.2	261 0.4 / 1.1
King's College	33.9 64.4 / 92.7	24.7 60.1 / 92.1	204 6.1 / 26.8	**16.2** **71.7** / **94.2**	23.2 61.2 / 92.7	23.1 61.5 / 91.8
Old Hospital	97.5 23.6 / 56	80.6 18.7 / 56.6	177 9.3 / 34.6	**63.8** **28.6** / **71.4**	100 15.9 / 51.6	92.8 23.1 / 61
Shop Façade	135 15.5 / **68**	125 14.6 / 49.5	219 3.9 / 31.1	**117** **18.4** / 56.3	149 11.7 / 47.6	131 12.6 / 58.3
St Mary's Church	162 13.4 / 50.8	125 16.8 / 56.6	260 2.6 / 20.6	**105** 13.6 / 52.6	115 17 / 56.2	108 **18.1** / **57.5**
Street	790 0.4 / 2.1	758 0.2 / 1.8	768 0.1 / 0.4	**505** 0 / 1.1	734 0.5 / 2.4	683 **0.6** / **3.5**

Table 5.1: **Evaluation of pose regression loss functions on the Cambridge Landmarks dataset.**

5.4.3 Evaluation

Metrics. To compare the effectiveness of the presented loss functions, we employ two distinct sets of metrics. First, consistently with previous research (Sarlin et al., 2021; Brachmann et al., 2021; Panek et al., 2022; Pietrantoni et al., 2023), we report the percentage of images that have been localized within specified thresholds, measured in meters and degrees. We use different threshold values for Cambridge and 7-Scenes datasets because the ratio between average translation and rotation errors is significantly different for outdoor and indoor scenes. Secondly, we compute the mean reprojection distance in pixels. For a given image, and for a

Scene	PoseNet	Homoscedastic	DSAC	Reprojection	Global Homography	Local Homography
	Mean reprojection distance in pixels ↓ Percentage of images localized within (0.25m, 10°) / (0.5m, 15°) ↑					
Chess	40.5 78.4 / 91.5	30.7 80.4 / 94	34.8 80.8 / 96.4	**26.5** 80.9 / 95.1	30.3 82 / **96.5**	28.9 **82.3** / 96.2
Fire	80 34.8 / 61.3	88.3 32.5 / 66	86.9 **35.4** / **68.8**	89.2 30.5 / 65.1	83.9 32.3 / 66.3	**79** 31.8 / 64.3
Heads	96 32.1 / 57	90.1 30.8 / 53	71.8 31.6 / **62.4**	75.1 31.7 / 55	76.2 33.3 / 59.2	**69.8** **33.9** / 58.3
Office	55 60 / 90.1	59.2 54.7 / 86.6	59.7 **62.3** / 87.6	50.3 59.5 / 90.3	55.2 57.3 / **90.7**	**46** **62.3** / 86.1
Pumpkin	121 **59** / 74.4	80.3 51.7 / 73.2	89.6 51.9 / **77.2**	87.5 50.9 / 71.9	97.3 50.1 / 71.5	**69.1** 53.5 / 73.9
Redkitchen	70.7 45.1 / 74.6	89.2 45.8 / 73.8	79.4 54.9 / 79.8	83.7 48.4 / 77	78.7 50.5 / 75.4	**66.3** **57.1** / **81.7**
Stairs	123 12.7 / 36.6	127 13.5 / 58.6	133 **22.2** / **59**	154 4 / 28.1	145 18.4 / 57.9	**121** 17 / 55.8

Table 5.2: **Evaluation of pose regression loss functions on the 7-Scenes dataset.**

given observed 3D point, we compute the euclidean norm between the projection of this point into the ground truth and the estimated camera views. This distance is then clipped at 1,000 pixels for each point to reduce the impact of outliers on the metric. We report the mean of all these distances.

Results. Tables 5.1 and 5.2 report the performance of the presented loss functions on the aforementioned metrics for the Cambridge Landmarks and 7-Scenes test sets. Overall, we observe that the Geometric reprojection loss performs better on the Cambridge dataset, while our Homography loss yields the best results on the 7-Scenes dataset. It is worth noting that on the 7-Scenes dataset, the Homography loss achieves an overall lower reprojection error compared to the Geometric reprojection loss, without requiring prior network initialization to prevent divergence.

These findings align with the observation made by Brachmann et al. (2021), claiming that different ground truth generation methods may favor different visual localization approaches. In the case of the Cambridge dataset, where ground truth poses were estimated using SfM, the Geometric reprojection loss minimizes the same quantity leveraging the same data that was used to estimate the ground truth. Conversely, the 7-Scenes dataset relies on poses obtained through depth-based SLAM. Although our loss does not minimize the exact same quantities as RGB-D SLAM, it could benefit substantially from access to dense depth maps, as

its parameters can be directly derived from them.

To complete the observations made by Brachmann et al. (2021), we also suggest that different metrics may favor different loss functions. For example, metrics like the reprojection distance may favor the Geometric reprojection and Homography losses, while metrics measuring the percentage of images localized within certain distance and orientation thresholds may be more favorable for PoseNet, Homoscedastic, and DSAC loss functions, given that they optimize a similar cost.

We provide videos[1] illustrating camera pose regression for a given image in diverse initial conditions using PoseNet, Geometric reprojection and Homography loss functions. When dealing with an initial pose that is oriented in the opposite direction of the scene, we observe that the proposed Homography loss successfully converges toward the correct ground truth pose, whereas the Geometric reprojection loss gets stuck in a local minimum.

5.5 Conclusion

In this chapter, we have introduced a novel loss function designed for camera pose regression in the context of deep learning applications. To gain a comprehensive understanding of the intricacies in designing pose regression loss functions, we conducted a survey of existing losses, revealing certain limitations for their application within deep learning frameworks. Notably, we found that while the Geometric reprojection loss has its advantages, especially when working with SfM-based ground truth, it faces differentiability issues.

Leveraging our discoveries, we introduced the Homography loss, that approximates the traditional reprojection error by representing the observed scene as planes. In comparison to other losses, our approach relies on two physically interpretable parameters, which can either be manually adjusted or computed from 3D data. Additionally, in contrast with the Geometric reprojection loss, it requires no preliminary initialization to converge.

Through experiments conducted on two visual localization datasets, we have shown that depending on the application, our loss offers a compelling alternative to existing pose regression losses. Furthermore, our loss may serve as a viable alternative to the Geometric reprojection loss when 3D data is inaccessible or when the target application necessitates pose regression without relying on specific 3D points.

As a perspective, this loss could potentially be integrated into more complete visual localization pipelines that rely on end-to-end pose regression, such as DSAC (Brachmann et al., 2017) or PixLoc (Sarlin et al., 2021). Also, recent

[1] youtube.com/playlist?list=PLe92vnufKoYIIHrW5I268RYdX6aV4gTa6

research has investigated deep regression in $SO(3)$ using different rotation representations (Brégier, 2021) and developing manifold-aware gradient for backpropagation (Chen et al., 2022b). Similar studies could be conducted for $SE(3)$, benchmarking different representations and exponential map computation techniques, including both closed form (Teed and Deng, 2021) and power series. This exploration can be applied to diverse applications, encompassing various visual localization methods and loss functions.

The work presented in this chapter led to a publication in *IEEE Robotics and Automation Letters* (Boittiaux et al., 2022) and selected for oral presentation at *ICRA 2023*.

Chapter 6

Underwater visual localization

Contents

6.1 Introduction

In Chapter 3, we constructed a dataset for long-term deep-sea visual localization, covering the same area across four visits over a five-year period. Throughout the dataset creation process, we encountered challenges due to the underwater domain shift, impacting various stages of the SfM process. In this chapter, we leverage the knowledge acquired during the creation of this dataset, along with insights from Chapter 5 regarding end-to-end pose regression, to conduct a comprehensive benchmark of diverse visual localization algorithms on the Eiffel Tower dataset.

To this extent, this chapter first introduces the different visual localization methods that will undergo evaluation in this benchmark. We present the characteristics of these algorithms, and detail their benefits in the context of reference camera poses derived from SfM. Subsequently, we present the localization results

achieved by these methods on the Eiffel Tower dataset. While these outcomes align with our initial observations, they also reveal that even the best-performing method still struggles with certain limitations.

In particular, our study highlights the challenges faced by the NetVLAD image retrieval network in this environment. Consequently, we conduct a brief study to assess the influence of image retrieval on the final localization outcomes. Our findings reveal a significant gap between the results achieved with NetVLAD trained on terrestrial images and an ideal image retrieval approach. Furthermore, we demonstrate that fine-tuning NetVLAD using a dataset of a few thousand underwater images substantially reduces this performance gap, leading to significantly improved visual localization results.

To this extent, this chapter makes the following contributions:

- We conduct a benchmark of several visual localization algorithms on the Eiffel Tower dataset. Through an analysis of the localization results, we identify image retrieval as a localization step that is particularly affected by the underwater environment.

- We analyze the limitations of the NetVLAD image retrieval network, and illustrate that there is a large margin for improvement. We then demonstrate that simply fine-tuning this network on underwater images results in a substantial enhancement of visual localization performance.

This chapter is organized as follows. First Section 6.2 presents the visual localization benchmark. Then Section 6.3 investigates the impact of image retrieval on visual localization.

6.2 Benchmarking visual localization algorithms

The Eiffel Tower dataset introduced in Chapter 3 establishes its ground truth through the use of SfM. However, as outlined in Section 2.2.3, it is essential to consider the algorithm used to build the ground truth when benchmarking visual localization algorithms (Brachmann et al., 2021). Indeed, some visual localization methods might minimize the same metric as the one that was used to build the ground truth, leading to a performance bias in favor of these methods. Consequently, this section benchmarks several visual localization algorithms of different nature. More specifically, we benchmark an absolute pose regressor like PoseNet (Kendall et al., 2015) using two different loss functions, as well as hLoc (Sarlin et al., 2019) and PixLoc (Sarlin et al., 2021) approaches. This section details the specificities of each method, presents the experimental setup, and then discusses the visual localization performance achieved by each method.

6.2.1 Evaluated methods

PoseNet. As described in Section 2.2.4 and Section 5.2.1, PoseNet (Kendall et al., 2015) is an absolute pose regressor. More specifically, it is a CNN that predicts a 6DoF pose from a single image. In this benchmark, we use the same backbone and experimental setup as described in Section 5.4.2. During the evaluation, PoseNet method refers to training the absolute pose regressor using the PoseNet loss function with the parameter $\lambda = 500$. It is important to note that this approach does not minimize the same error as SfM.

Homography loss. Similarly to PoseNet, we train the same absolute pose regressor using the Local Homography loss function presented in Chapter 5. We configure the parameters of the loss function following the procedure outlined in Section 5.4.2, which involves choosing $x_{\min}$ and $x_{\max}$ as the depth distribution's 2.5th and 97.5th percentiles, respectively. Like PoseNet, this approach does not minimize the same quantity as the SfM ground truth.

hLoc. hLoc is a hierarchical visual localization toolbox (Sarlin et al., 2019). It operates similarly as the retrieval-based visual localization methods presented in Section 2.2.4. More specifically, it performs the following steps. Given a query image and a set of database images, it starts by retrieving images in the database that are similar to the query image. This is achieved by leveraging global descriptors. By default, hLoc relies on the NetVLAD global descriptor (Arandjelovic et al., 2016). Then, it extracts and matches features between the query image and the retrieved images. In hLoc, the default feature matching technique is to extract SuperPoint features (DeTone et al., 2018) and then match these features using the SuperGlue (Sarlin et al., 2020) matching network. Finally, the query image's 6DoF pose is retrieved through a PnP/RANSAC scheme. It is important to note that all these steps are very similar to those used to create the underwater dataset in Chapter 3. Because of this, hLoc might be strongly advantaged compared to other methods when evaluated on the Eiffel Tower dataset.

PixLoc. PixLoc is a photometric visual localization method (Sarlin et al., 2021). It executes the following steps. Given a query image and a SfM database, it performs the same initial step as hLoc, that is, it retrieves images in the database that are similar to the query image. It subsequently extracts dense local features for the query and the retrieved images using a CNN. Then, PixLoc aims at finding the 6DoF pose that minimizes the difference in appearance between the query image and each reference image. This difference in appearance is quantified directly inside the local feature space. Let us consider a 3D point $\mathbf{X}_p$ observed in the query image q and a reference database image r. The query image has an

estimated pose $[\hat{\boldsymbol{R}}_q, \hat{\boldsymbol{t}}_q]$ and the reference image has a pose $[\boldsymbol{R}_r, \boldsymbol{t}_r]$. Both the query and the reference image have a dense local descriptors map $\boldsymbol{F} \in \mathbb{R}^{W \times H \times D}$, where D is the dimension of each descriptor. The local descriptor at a given pixel coordinates can be obtained by $\boldsymbol{F}[\cdot]$. To estimate their pose directly from the local features space, PixLoc define their residual as:

$$\boldsymbol{r}_{p,q,r} = \left\| \boldsymbol{F}_q\left[\pi(\hat{\boldsymbol{R}}_q \mathbf{X}_p + \hat{\boldsymbol{t}}_q)\right] - \boldsymbol{F}_r\left[\pi(\boldsymbol{R}_r \mathbf{X}_p + \boldsymbol{t}_r)\right] \right\|. \tag{6.1}$$

Consequently, the error that is minimized to estimate the query image's 6DoF pose is different from the one minimized by the SfM ground truth of the Eiffel Tower dataset.

hLoc & PixLoc. hLoc and PixLoc can be used conjointly to refine localization outcomes. In this scenario, PixLoc is initialized using the 6DoF pose estimated by hLoc. The photometric optimization is then applied to the 3D points identified as inliers by hLoc's PnP/RANSAC step.

hLoc w/ SUCRe. To evaluate the visual localization performance degradation caused by the underwater domain shift, we also run the hLoc pipeline using images enhanced with the SUCRe method described in Section 4.3.

6.2.2 Experimental setup

Dataset split. To evaluate the performance of visual localization algorithms, we must split our dataset in two parts: a reference database set and a query set. The performance of a given visual localization method will evaluate its capacity to effectively localize query images with respect to database images. Consequently, a good practice is to make sure that all query images share 3D observations with at least one image in the database. In Section 3.5, Figure 3.12 illustrates that the 2015 dive on the Eiffel Tower vent covers the least amount of ground compared to other expeditions. Moreover, all the explored area during the 2015 dive has also been explored during other dives. To minimize the errors due to previously non-visited areas, we choose the 2015 visit as the query set, and select the 2016, 2018 and 2020 dives as the database set.

Images undistortion. Both hLoc and PixLoc benefit from information about the intrinsics of the query camera. However, absolute pose regressors do not include this information in their regression pipeline. To alleviate this issue, we undistort input images and center their principal point during the preprocessing step of PoseNet and Homography loss. This preprocessing step is also common in scene coordinate regressors (Brachmann et al., 2017; Brachmann and Rother, 2022; Brachmann et al., 2023).

Method	Median errors ↓	Percentage of images localized within ↑				
		1cm, 1°	2cm, 2°	5cm, 5°	50cm, 5°	500cm, 10°
PoseNet	1.98m, 10.73°	0.00	0.00	0.02	3.58	45.75
Homography loss	1.23m, 8.30°	0.00	0.00	0.02	8.47	57.83
hLoc	**0.09m**, 1.14°	**15.61**	**27.96**	43.59	**57.79**	**59.95**
PixLoc	6.55m, 41.09°	0.37	1.61	6.51	15.10	18.36
hLoc & PixLoc	**0.09m**, **1.12°**	13.68	27.72	**43.96**	**57.79**	**59.95**
hLoc w/ SUCRe	**0.05m**, **0.56°**	**14.35**	**30.10**	**51.30**	**73.42**	**75.89**

Table 6.1: **Performance of visual localization methods on the Eiffel Tower dataset.** We show **best** and **second best** performance on median errors and the percentage of images localized within a given threshold.

6.2.3 Results and discussion

In Table 6.1, we report the performance achieved by aforementioned localization methods on the query set of the Eiffel Tower dataset. The evaluation includes two key metrics: the median translation and rotation localization errors in meters and in degrees, as well as the percentage of images that were successfully localized within specified translation and rotation thresholds in centimeters and in degrees.

Overall, the best performance is achieved by the hLoc pipeline. This is not surprising since the method is very similar to the one that was used to create the ground truth reference camera poses. With the exception of the image retrieval step achieved using NetVLAD, hLoc follows the same pipeline as the one described in Section 3.4. Since PixLoc optimization procedure is performed on the set of retrieved images, the method heavily relies on the image retrieval step, and achieves poor performance on all metrics. Moreover, refining poses estimated with hLoc by using PixLoc photometric alignment does not appear to greatly improve localization accuracy.

The criterion assessing the percentage of poses localized within 500 centimeters and ten degrees is more akin to evaluate image retrieval rather than precise localization performance. Interestingly, despite their poor performance in accurate pose regression, absolute pose regressors like PoseNet and Homography loss show good performance for this coarse metric. This observation aligns with the findings of Sattler et al. (2019) that absolute pose regressors are closer to image retrieval methods than precise pose estimators. This might provide motivation for employing absolute pose regressors to address the image retrieval step.

Finally, restoring the images' colors using the SUCRe method shows to im-

prove significantly localization results of the hLoc pipeline. We suggest that this improvement can be attributed to the fact that the NetVLAD image retrieval network, along with the SuperPoint and SuperGlue feature matching networks, were originally trained exclusively on terrestrial images. Consequently, these networks may not adapt effectively to underwater conditions and generalize to underwater phenomena, including scattering, absorption, and the high illumination variability. However, it is essential to acknowledge that each SUCRe images was restored using 3D information from its individual model, including images from the 2015 test year, introducing a potential bias into the evaluation process.

Bringing together the insights gained during the dataset construction in Chapter 3 and the visual localization outcomes presented here, it appears that the primary challenge in underwater visual localization is related to image retrieval. This issue is highlighted in Table 6.1, where PixLoc performance underscores this problem. Since PixLoc heavily depends on the image retrieval step, it struggles to localize images when it is provided incorrectly retrieved candidates. In contrast, when presented with a coherent set of 3D points, such as those provided by hLoc, it maintains pose estimation accuracy. To comprehensively explore the influence of image retrieval on visual localization results, the following section is devoted to a thorough evaluation of this impact.

6.3 Image retrieval impact

To evaluate the impact of image retrieval on visual localization performance, we conduct a short study on the hLoc pipeline. Our objective is to substitute the image retrieval component of hLoc and observe the resulting changes in visual localization outcomes. To this extent, this section explores the impact of fine-tuning the NetVLAD image retrieval network on localization results. In addition, we also report the localization performance obtained when replacing the image retrieval step by a covisibility oracle, *i.e.*, a method that has privileged information about the ground truth and can serve as a top line reference of performance. This section provides a comprehensive overview of these various approaches and subsequently presents and discusses the localization results.

6.3.1 Image retrieval approaches

Before delving into the specificities of fine-tuning NetVLAD, let us provide a concise overview of the NetVLAD architecture. As outlined in Section 2.2.4, NetVLAD is designed to learn simultaneously both dense local features and the assignment of these features to visual words. This learning process can be broken down into two primary components. First, feature learning is accomplished

using a CNN backbone. Second, the assignment of visual words is performed through what is called a VLAD layer. While the dense features are learned by optimizing the CNN weights, the VLAD layer specifically focuses on learning the positions, or centers, of visual words within the feature space.

Fine-tuning NetVLAD centers. In this scenario, we only refine the positions, or centers, of visual words for their application to underwater data. By doing this, we do not alter the representation of dense local features, we simply backpropagate the error on visual words to find better discriminative ones.

Fine-tuning NetVLAD network. In this case, we fine-tune the entire NetVLAD network, which includes adjusting both the weights of the CNN and the positions of visual words. By comparing this approach to only adjusting the visual words, we gain valuable insights into the limitations of dense local features pretrained in terrestrial environments when applied to deep-sea settings.

Covisibility oracle. To assess the efficiency of NetVLAD compared to optimal image retrieval results, we also replace the image retrieval step with a covisibility oracle: for each query image, we use the SfM ground truth to select the top ten database images that share the most 3D point observations.

6.3.2 Experimental setup

Dataset split. For fine-tuning NetVLAD, we rely exclusively on the reference database set, as the query set should only be used for the visual localization benchmark. Consequently, we only train NetVLAD on the 2016, 2018 and 2020 dives. From these three visit years, we split the train and validation sets randomly, keeping 90% of the images for training and 10% for validation.

Training. In the original paper (Arandjelovic et al., 2016), NetVLAD is trained using a triplet loss. Each query descriptor $\boldsymbol{d}^q$ is associated with a set of potential positive descriptors $\{\boldsymbol{d}_a^p\}$ and a set of negative descriptors $\{\boldsymbol{d}_b^n\}$. Using the ground truth poses, positive descriptors are extracted from images that are spatially close to the query image, and negative descriptors are extracted from spatially far images. The network is then trained by minimizing the following loss function:

$$\mathcal{L}_{\text{NetVLAD}} = \sum_b \max\left(\min_a \left(\|\boldsymbol{d}^q - \boldsymbol{d}_a^p\|^2\right) + m - \|\boldsymbol{d}^q - \boldsymbol{d}_b^n\|^2, 0\right), \tag{6.2}$$

where m is a margin parameter. The idea behind this loss can be expressed as follows: the distance between the query descriptor and the closest positive descriptor should be at a minimum distance of m compared to negative descriptors.

In other words, there is at least a real positive match in the positive set, and its corresponding descriptor should be closer by a margin to the query descriptor than the descriptors of negative matches.

Throughout the training process, we define positive images as the top fifteen images positioned within a three-meter radius of the query image. We designate the negative images as a sample of thirty images located more than ten meters away from the query image. Differing from the original implementation, we introduce an additional constraint to encourage the network to learn cross-year discriminative features and visual words. Specifically, we ensure that each image in the positive set is extracted from images captured during a different visit year than the query image: $\text{year}(\boldsymbol{d}^q) \neq \text{year}(\boldsymbol{d}_a^p)$. This measure prevents overly simple matches between the positive set and the query image.

6.3.3 Results and discussion

Method	Median errors ↓	Percentage of images localized within ↑				
		1cm, 1°	2cm, 2°	5cm, 5°	50cm, 5°	500cm, 10°
NetVLAD	0.089m, 1.144°	15.61	27.96	43.59	57.79	59.95
NetVLAD w/ fine-tuned centers	**0.027m, 0.347°**	**21.82**	**40.80**	**61.82**	**75.78**	**77.19**
NetVLAD w/ full fine-tuned network	**0.020m, 0.242°**	**29.98**	**50.63**	**74.30**	**86.14**	**87.42**
Covisibility oracle	**0.014m, 0.189°**	**36.63**	**64.33**	**87.77**	**96.83**	**97.50**

Table 6.2: **Performance of hLoc on the Eiffel Tower dataset** using different image retrieval approaches. We show how the image retrieval step of hLoc influences localization results. We highlight **best** and **second best** performance.

In Table 6.2, we present the localization results achieved using different image retrieval methods within the hLoc pipeline. The covisibility oracle offers insight into what could be achieved with an optimal image retrieval method. Notably, we observe that employing NetVLAD, originally trained on terrestrial images, falls significantly short of optimal performance. In contrast, fine-tuning only the visual words centers already yields a substantial enhancement in visual localization performance, indicating a need to address domain-specific differences between terrestrial and underwater environments. Moreover, fine-tuning the entire network results in a significant performance boost compared to fine-tuning only the visual words. This implies that the dense local features extracted by the CNN

from terrestrial images may lack the necessary discriminative power to accurately represent the diverse environmental conditions encountered in the deep sea, such as those discussed in Section 3.5.

It is worth mentioning that fine-tuning only the visual word centers theoretically requires fewer data samples than fine-tuning the entire network. Visual words essentially constitute a dictionary of distinctive terms for describing an image, and fine-tuning these already substantially improves visual localization results. This observation holds promise for refining networks initially trained on terrestrial images, by using only a small sample of underwater data. This is particularly valuable considering the scarcity and expense associated with acquiring deep-sea data.

6.4 Conclusion

In this chapter, we conducted an evaluation of various visual localization algorithms of diverse nature using the Eiffel Tower dataset introduced in Chapter 3 to assess their adaptability to the underwater environment. While presenting the benchmarked algorithms, we emphasized that certain methods might be in advantage when applied to a SfM reference ground truth, similar to the one provided by the Eiffel Tower dataset. The experimental results seem to point in this direction.

Among the evaluated algorithms, the hierarchical localization pipeline hLoc, which shares many similarities with SfM, demonstrated superior performance compared to other methods. As a side study, we also demonstrated that the underwater color restoration method proposed in Section 4.3 significantly improves visual localization results when used in conjunction with hLoc. However, this hierarchical approach does have its limitations. We have identified that the primary challenge in hLoc lies within the image retrieval step. We have shown that this step can be substantially improved through fine-tuning an image retrieval network on a few thousands underwater images.

Looking ahead, the key challenge in the underwater environment remains the scarcity of data. While future projects may seek to further explore the deep sea and generate data suitable for training these algorithms, it is prudent to focus on methods that demand only a limited dataset for fine-tuning existing algorithms, such as the approach we explored by fine-tuning only NetVLAD visual words. This successful approach was suggested by Torsten Sattler and Assia Benbihi.

Part of this work was published in the *ORASIS 2023* national conference for young researchers (Boittiaux et al., 2023a).

Chapter 7

Conclusion

Contents

This thesis has explored the subject of visual localization within the context of long-term deep-sea monitoring. Existing acoustic positioning systems have limitations regarding their ability to precisely estimate the pose of autonomous underwater vehicles and the cost of their deployment in some scenarios. This limits their capacity to carry out specific tasks that necessitate a high location accuracy, such as mapping a precise area of a site of interest. In pursuit of a more accurate means of estimating the position of underwater vehicles, this thesis has investigated the use of the vehicle's visual observations, thus addressing the challenge of visual localization. However, the majority of visual localization research is primarily focused on terrestrial applications, such as self-driving cars and augmented reality systems. Underwater imaging introduces new sources of variability and scene complexity, setting it apart from the more conventional realm of terrestrial imagery. In light of these challenges, this thesis has been dedicated to identifying the domain-specific sources of variability that affect underwater images. Furthermore, it sought to elucidate the impact of these distinct factors on visual localization algorithms initially designed for terrestrial scenarios. In response, we have proposed solutions, drawing from both physics-based models and deep learning methods, to adapt and optimize these algorithms to deep-sea environments. In this conclusion, we first provide an overview of the material covered throughout this thesis and outline our scientific contributions. We subsequently propose several avenues for improvement and offer insights into potential directions for future research.

7.1 Summary and contributions

Problem presentation. Chapter 2 was dedicated to the presentation of the underwater visual localization challenge. In this regard, we provided an exploration of the unique aspects of underwater imaging and presented the visual localization problem. In the underwater imaging domain, we delved into the specific phenomena encountered in underwater environments, notably the refraction effects caused by the air-glass-water mediums, as well as the distinctive characteristics of light propagation under water, which involve absorption and scattering phenomena. We also offered an overview of established models employed to model these phenomena and explored existing methods designed to restore underwater images, mitigating the impact of water-induced light propagation effects. Shifting our focus to visual localization, we have started by defining the problem. Given that it entails determining the viewpoint of images with respect to a given reference frame, we described various approaches designed to establish the ground truth of such viewpoints, *e.g.*, using depth-based SLAM or SfM. As highlighted by recent research (Brachmann et al., 2021), we emphasized the importance of considering the method used for generating this reference when benchmarking visual localization methods. Finally, we provided an overview of diverse visual localization techniques, each tailored to specific applications and relying on different types of input data.

Deep-sea dataset for visual localization. In Chapter 3, we present a dataset explicitly designed for the evaluation of long-term visual localization algorithms within the deep-sea environment. This dataset comprises images acquired during four visits to a hydrothermal vent, spanning over a five-year period. To establish unified reference camera poses for all these visits, we used SfM. In the process of constructing this unified SfM model, we investigated challenges encountered along every step of the pipeline. We first ran into the problems associated with image retrieval methods, particularly when pairing underwater images of different visits. To overcome this problem, we introduced a pipeline based on the alignment of point clouds derived from individual SfM models. We then conducted a comprehensive benchmark of feature matching algorithms on deep-sea images from different visit years capturing identical scene elements. The goal was to identify the most robust matching methods capable of effectively managing environmental variations. This formulation not only facilitates the retrieval of the model scale and orientation but also effectively guides the optimization of the registered viewpoints during the bundle adjustment process. All of these findings allowed for the creation of a unified model that incorporates images from all visits. Given this unified model, we then conducted a comprehensive survey

of the changes in appearance that have occurred over the years to gain valuable insights regarding the different sources of variability encountered in this environment.

Structure-based underwater color restoration. The contributions of Chapter 4 are twofold: firstly, we formulated and introduced two methods tailored to restore the colors of underwater images, and secondly, we have acquired insights regarding underwater imaging phenomena capable of affecting visual localization algorithms. The first underwater color restoration method relies on a single assumption, which serves as a constraint for estimating the parameters of an underwater image formation model. It only requires an underwater image and its corresponding distance map. In contrast, the second method bypasses this assumption by articulating the underwater image formation model within a multi-view setting. With a collection of underwater images, coupled with their associated camera poses, intrinsic parameters, and depth maps, this approach simultaneously estimates the image formation model parameters and the restored image in a single optimization procedure. Additionally, we extended this underwater image formation model to account for the vignetting effect typically generated by artificial lighting systems of deep-sea underwater vehicles. We demonstrated that the multi-view approach effectively overcomes most of the difficulties encountered in single-view underwater color restoration techniques. Finally, we showed that restoring the colors of underwater images using this multi-view method has the potential to enhance the performance of visual localization algorithms in underwater environments.

Loss function for deep learning based camera pose regression. Chapter 5 was dedicated to the exploration of loss functions tailored for camera pose regression within deep learning applications. In this context, we conducted a comprehensive survey of existing loss functions, highlighting their theoretical and practical limitations. Building upon these insights, we introduced a novel loss function designed to approximate the traditional reprojection error by representing the observed scene as planes. This approach allows our loss function to overcome the differentiability issues associated with the reprojection error. In comparison with other losses, our method depends on two physically interpretable parameters and does not necessitate any initialization for convergence. Depending on the specific application, we demonstrated that the presented loss function is a compelling alternative to existing pose regression losses. Furthermore, it can serve as a suitable loss function for weighting rotation and translation errors when the application necessitates pose regression without relying on specific 3D points.

Benchmark of underwater visual localization. In Chapter 6, we evaluated various visual localization algorithms on the Eiffel Tower dataset introduced in Chapter 3. Our goal was to assess their adaptability to underwater environments. When introducing the benchmarked algorithms, we pointed out their potential advantages when working with a SfM reference ground truth, similar to the one used in the Eiffel Tower dataset. Our evaluation of these algorithms revealed that the hierarchical localization pipeline, which shares many similarities with the SfM pipeline, achieved the best performance on the Eiffel Tower dataset. From our analysis of the visual localization results, we gained two significant insights: firstly, the application of the underwater color restoration method detailed in Section 4.3 significantly improves localization outcomes, and secondly, the primary bottleneck in the performance of compared methods lies in the image retrieval step. Considering this limitation, we explored the impact of fine-tuning an image retrieval network using a few thousand underwater images. By conducting this fine-tuning on different parts of the network, we also showed that there is hope for effective domain transfer between terrestrial and underwater images by fine-tuning specific parts of the neural network using a limited amount of data.

7.2 Lessons learned

Camera poses are estimated in relation to an established reference frame, which is itself determined using methods that introduce their own biases (Brachmann et al., 2021). Additionally, metrics employed to assess localization performance tend to favor particular ground truth generation techniques or visual localization algorithms over others. This does not imply that evaluating visual localization methods is useless — rather, it underscores that this evaluation should always be conducted with consideration for the chosen ground truth generation method and the specific metrics employed.

Absolute pose regressors such as PoseNet face limitations when it comes to accurately estimating camera poses Sattler et al. (2019). Achieving high localization accuracy results while representing the observed scene within a neural network's weights requires to explicitly incorporate geometric phenomena within the network's inference (Brachmann et al., 2017, 2023). Nevertheless, these straightforward absolute pose regressors offer a simplified pipeline for addressing broader theoretical challenges associated with camera pose regression within the context of deep learning frameworks, such as designing loss functions (Kendall and Cipolla, 2017; Boittiaux et al., 2022), or studying different pose representations (Brégier, 2021).

7.3 Perspectives

Make better use of 3D information. While we mainly focused on the degrading effects of the physical phenomena that impact underwater images, it is worth noting that these phenomena can also provide supplementary information about the scene's 3D structure, as they are closely related with the distance between the observed scene and the camera. Consequently, neural networks designed for depth estimation (Ranftl et al., 2022) could benefit from this implicit source of information when estimating depth maps from underwater images. This could serve as a practical means of obtaining real-time 3D scene information. Prior research has already focused on estimating depth maps from a single image and refining initial coarse pose estimates, which were obtained using image retrieval (Piasco et al., 2019a). This refinement is achieved through point cloud alignment, aligning the local point cloud obtained by unprojecting the estimated depth map of the query image with the global point cloud of the scene. In our specific application context, we could replace the image retrieval step of hierarchical localization pipelines by robust point cloud alignment. This would involve aligning the local point cloud derived from the query image estimated depth map with the reference SfM point cloud.

Towards a practical application within a SLAM framework. In practice, Ifremer's underwater vehicles will soon be equipped with an underwater SLAM that will operate in real time during dives (Ferrera, 2019). This means that during each dive, the vehicle will build a local map of the specific area it is exploring. Consequently, instead of attempting to estimate the pose of each acquired image in relation to a given reference model, a pragmatic approach would involve determining only the $Sim(3)$ or $SE(3)$ transformation between this local map and the reference model. This could be achieved, for instance, through a robust point cloud alignment (Yang et al., 2021) between the SLAM point cloud and the reference SfM point cloud. This approach is similar to the one we employed in Chapter 3 during the dataset creation to overcome the challenges encountered with image retrieval. It is expected to offer significant practical benefits since it utilizes a more comprehensive representation of the scene and should become increasingly robust as more of the scene is mapped during the SLAM process, thereby providing an increasing volume of 3D information. This additional data helps in disambiguating and improving the estimation of point cloud alignment.

Stereo underwater color restoration. As discussed in Chapter 4, the task of underwater image color restoration from a single image is underdetermined. At least two views of the same scene are necessary to constrain the estimation of

both the parameters of an underwater image formation model and the intensity values of the restored images. In this context, using a calibrated stereo rig could enable the use of disparity map estimation using neural networks (Tankovich et al., 2021). This estimated disparity map would, in turn, allow to recover the scene's 3D structure alongside 2D-2D correspondences between the two images. Consequently, it would enable the use of the multi-view method described in Section 4.3 using only two images. This approach would run in real time and also support the color restoration of dynamic scene elements, as both views would be captured simultaneously.

End-to-end underwater color restoration. Using the SUCRe method developed in Section 4.3, we can generate a database of underwater images along with their restored counterparts. As the main limitations of SUCRe involve its computational time and its dependence on the 3D structure of the scene, an alternative strategy could involve the training of a U-Net like neural network (Ronneberger et al., 2015) to learn a mapping between underwater images and their restored version. This approach would have the potential benefit of running in real time, without the need for prior information about scene's 3D structure.

Adapting to limited underwater data. In the near future, there might not be a sufficient volume of underwater data to fully train neural networks in the same manner as their terrestrial counterparts. To tackle this challenge, it becomes imperative to develop techniques that can make the most out of the limited underwater data available for fine-tuning neural networks. A successful demonstration of this approach can be found in Chapter 6, where it was effectively applied to the NetVLAD image retrieval network. Extending this concept, identifying specific segments of neural networks that can be fine-tuned with a limited underwater dataset should be explored for other components of the visual localization process, such as feature matching networks.

Appendix A

Résumé étendu

Cette thèse aborde la problématique de la localisation de véhicules sous-marins autonomes en exploitant leurs données visuelles. Plus précisément, nous nous intéressons au problème de la localisation visuelle dans le cadre de revisites à long terme des grands fonds océaniques. Ce domaine de recherche est particulièrement délicat en raison de la disponibilité limitée de données sous-marines. La plupart des algorithmes existants sont conçus pour des applications terrestres et ne se généralisent pas nécessairement bien aux environnements sous-marins, compte tenu des changements spécifiques tels que la variation de turbidité entre deux visites ou la sédimentation des structures observées. Ce manque de données revêt une importance particulière dans un contexte où les approches de pointe sont largement dominées par l'apprentissage profond, exigeant une grande quantité de données d'entraînement pour généraliser à de nouveaux phénomènes. Les contributions majeures de cette thèse comprennent :

- La création d'un nouveau jeu de données sous-marin pour évaluer les performances des algorithmes de localisation visuelle à long terme dans les grands fonds.

- Deux nouvelles méthodes de restauration des couleurs des images sous-marines.

- La proposition d'une nouvelle fonction de coût pour la régression de la pose dans un contexte d'application au deep learning.

- Une évaluation comparative des méthodes de localisation visuelle sur le nouveau jeu de données, en utilisant les méthodes de restauration des couleurs et la nouvelle fonction de coût.

A.1 Jeu de données sous-marin pour la localisation visuelle à long terme

En raison des variations à long terme des grands fonds marins, distinctes des variations rencontrées dans les jeux de données terrestres, il est impératif de créer un jeu de données spécifiquement dédié à la localisation visuelle dans ce contexte particulier. Ce jeu de données doit être constitué de visites récurrentes d'un même site en grands fonds sur plusieurs années afin de documenter les changements environnementaux à long terme. De plus, il doit être constitué en utilisant divers systèmes d'acquisition tels que des caméras différentes, afin de prendre en compte les changements d'équipement pouvant influencer les algorithmes de localisation visuelle. L'Ifremer détient des données pertinentes, notamment des observations de la cheminée hydrothermale appelée Tour Eiffel en 2015, 2016,

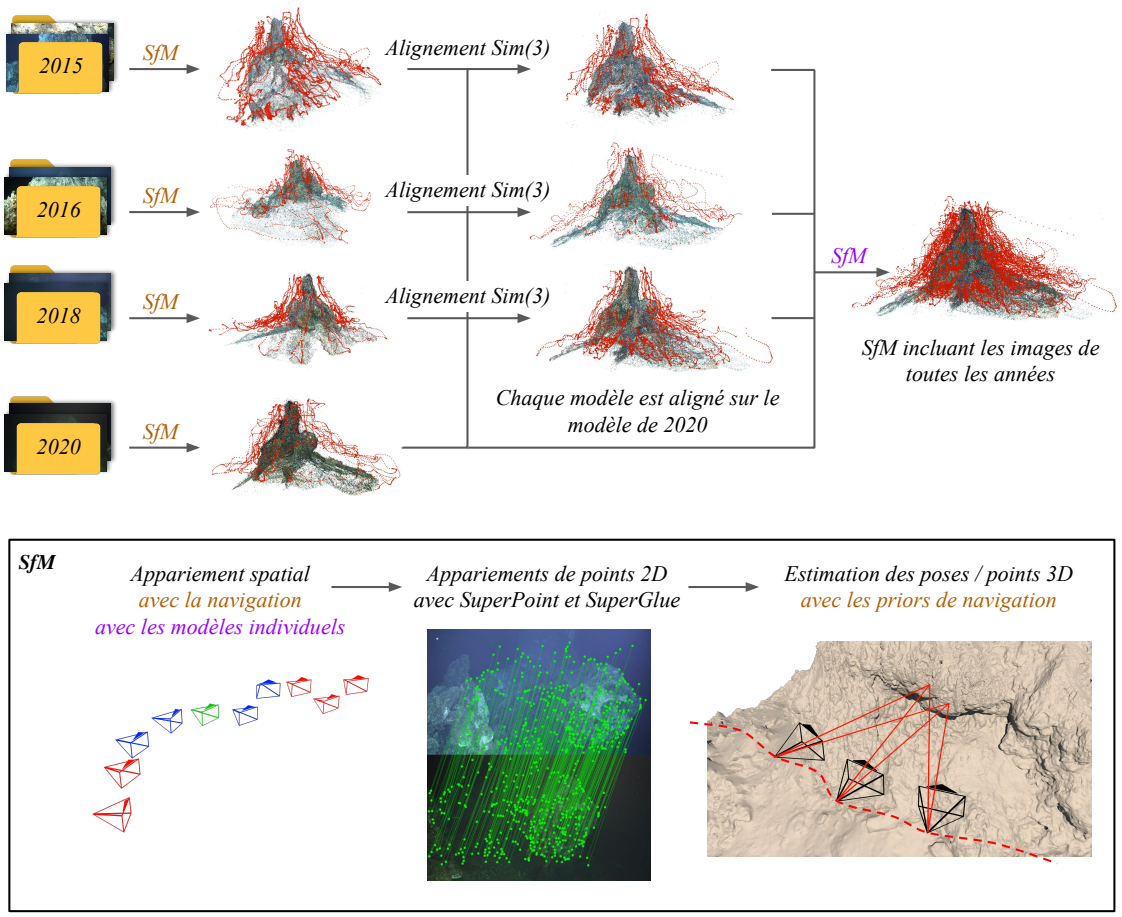


Figure A.1: **Chaîne opératoire de Structure-from-Motion.** Initialement, des modèles indépendants sont construits pour chaque année. Ces modèles sont ensuite alignés dans un référentiel commun. Enfin, un modèle global englobant des images de toutes les années est généré en utilisant une correspondance spatiale basée sur les poses des caméras des modèles individuels.

2018 et 2020, réalisées dans le cadre de missions dédiées à la reconstruction 3D. Pour établir une vérité terrain de notre jeu de données, nous devons déterminer avec précision les poses des caméras pour chaque image, et ce, dans un repère commun aux quatre années d'observation. Pour cela nous avons utilisé une solution de Structure-from-Motion. Cependant, en raison des différences visuelles importantes entre les images des différentes années, nous avons dû revoir les étapes d'appariement d'images, d'appariement de points 2D et de bundle adjustment du Structure-from-Motion. En utilisant une approche combinant réseaux de neurones et méthodes classiques, nous avons développé la chaîne opératoire illustrée dans la Figure A.1, permettant d'obtenir les poses des caméras pour toutes les années dans un repère partagé. Enfin, nous avons montré des changements significatifs dans l'environnement et la structure de la scène au fil des années.

A.2 Restauration de couleurs d'images sous-marines

L'un des principaux facteurs influençant l'apparence des images sous-marines réside dans la manière dont la lumière se propage dans l'eau, engendrant un faible contraste et une déformation des couleurs. Ces altérations résultent principalement de deux phénomènes majeurs : la conversion de la lumière en d'autres

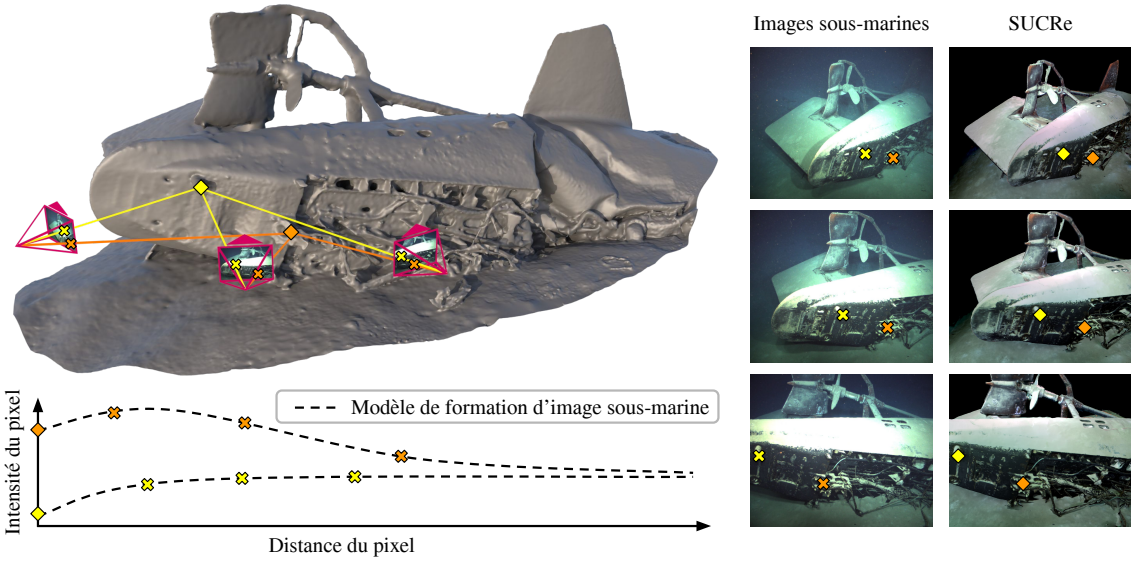


Figure A.2: **Chaîne opératoire de SUCRe.** Nous utilisons les poses de chaque image, leur paramètres intrinsèques et leur cartes de profondeur résultant d'un Structure-from-Motion pour appairer géométriquement les pixels entre différentes vues. Nous projetons les pixels d'une à l'autre, nous permettant d'appairer des points dans des zones de faible contraste. Enfin, nous estimons simultanément les paramètres d'un modèle de formation d'image sous-marine ainsi que l'image restaurée.

formes d'énergie, telles que la chaleur, provoquant une absorption des couleurs, et la réflexion de la lumière par des particules en suspension dans l'eau, entraînant de la rétrodiffusion. Pour atténuer ces effets indésirables sur les images sous-marines, de nombreuses méthodes ont cherché à modéliser et à inverser ces phénomènes. En nous appuyant sur un modèle existant, nous avons proposé deux approches distinctes pour inverser ce modèle et atténuer les effets d'absorption et de rétrodiffusion dans les images sous-marines. L'une de ces méthodes prend en compte une image sous-marine avec sa carte de profondeur associée, en se fondant sur l'hypothèse simple selon laquelle l'intensité des pixels est distribuée de manière gaussienne dans l'image restaurée. L'autre méthode, nommée SUCRe et illustrée dans la Figure A.2, utilise plusieurs images d'une même scène, suivant l'intensité des pixels à travers différentes prises de vue, afin d'estimer simultanément les paramètres du modèle et l'intensité des pixels de l'image restaurée.

A.3 Fonction de coût pour la régression de pose

Les méthodes état de l'art pour la localisation visuelle reposent principalement sur des méthodes de d'apprentissage profond. Parmi celles-ci, de nombreux réseaux de neurones apprennent la pose d'une image en se basant uniquement

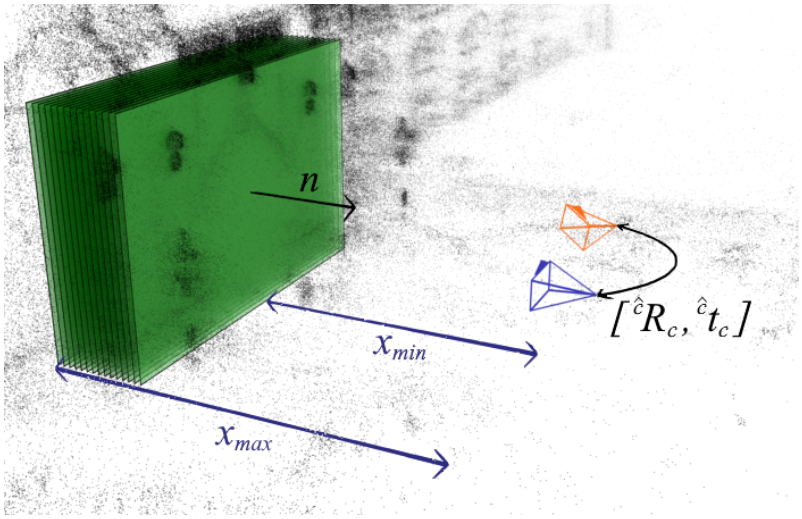


Figure A.3: **Illustration de la fonction de coût homographique.** Nous remplaçons les points 3D (noirs) observés par la pose réelle (bleue) par un ensemble de plans virtuels parallèles (verts). La normale $\boldsymbol{n}$ des plans et l'axe optique de la caméra de vérité terrain sont colinéaires. Pour un plan donné, nous exprimons notre erreur directement dans l'homographie induite par ce plan entre la pose réelle et la pose estimée (orange). Nous intégrons ensuite cette erreur entre les distances $x_{\min}$ et $x_{\max}$.

sur cette dernière. Ces réseaux sont entraînés en supervisant la pose estimée, et en la comparant à la véritable pose de l'image. Pour propager l'erreur à travers le réseau, une fonction de coût est utilisée pour quantifier l'écart entre la pose estimée et la pose réelle. Pour pouvoir propager l'erreur au travers du réseau, ils reposent donc sur une fonction de coût qui quantifie l'erreur entre la pose estimée et la pose vérité terrain. Cependant, alors que l'erreur entre deux pose est en six degrés de liberté, cette fonction de coût doit générer un scalaire, afin que l'erreur puisse être propagée par dérivation chaînée à travers le réseau. Cette fonction de coût doit résoudre le défi de l'équilibre entre les erreurs de rotation et de translation entre deux poses. Une approche qui traite implicitement ce problème est l'erreur de reprojection, car elle exprime l'erreur directement dans le plan du capteur plutôt que sur les poses. Cependant, l'application de cette fonction de coût aux méthodes d'apprentissage profond présente des limites en termes de dérivabilité. Pour surmonter ces limitations, nous proposons une fonction de coût qui approxime l'erreur de reprojection tout en évitant les problèmes de dérivabilité. Notre fonction de coût, illustrée dans la Figure A.3, modélise la scène par des plans plutôt que des points, exprimant ainsi l'erreur directement dans les homographies induites par ces plans entre la pose estimée et la pose réelle.

A.4 Evaluation de la localisation visuelle en milieu sous-marin

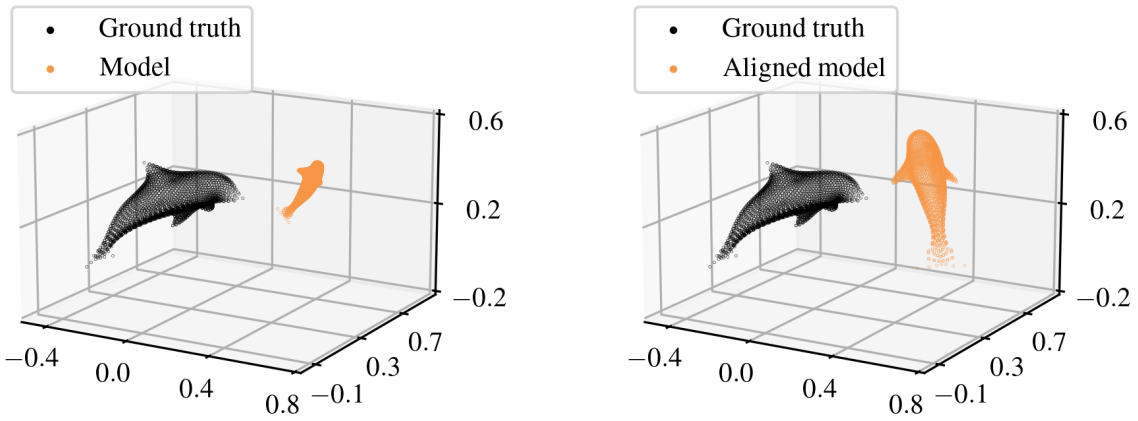
Finalement, nous effectuons une analyse comparative de diverses méthodes de localisation visuelle sur le jeu de données sous-marin établi précédemment. Les résultats révèlent que la méthode de localisation hiérarchique obtient les meilleures performances. Cependant, malgré ses résultats prometteurs, elle ne parvient pas à égaler les performances anticipées sur des jeux de données terrestres. Afin d'améliorer ses performances, nous entreprenons le raffinement des réseaux de neurones utilisés dans la méthode sur des données sous-marines, dans l'espoir de constater une amélioration des résultats. En raison de la disponibilité limitée de données, nous développons deux méthodes de raffinement distinctes, l'une nécessitant théoriquement moins de données que l'autre. Les résultats indiquent que la méthode demandant moins de données améliore déjà de manière significative les performances de localisation visuelle. Les performances sont encore nettement améliorées en utilisant la méthode de raffinement qui nécessite plus de données. Par ailleurs, nous avons testé la méthode de localisation hiérarchique sur des images dont les couleurs ont été restaurées grâce à la méthode multi-vues développée précédemment, montrant que la restauration des couleurs contribue à améliorer les performances de localisation visuelle.

A.5 Conclusion

Cette thèse se concentre sur le défi de la localisation visuelle dans le contexte d'explorations récurrentes en grands fonds marins. Nous avons introduit un nouveau jeu de données permettant d'évaluer les méthodes de localisation visuelle dans ce milieu particulier. À travers ce jeu de données, nous avons mis en évidence diverses sources de variabilité qui émergent au fil des années et qui ont le potentiel d'influencer les résultats des méthodes de localisation visuelle. Afin d'atténuer la variabilité induite par la propagation de la lumière dans l'eau, nous avons présenté deux nouvelles méthodes de correction des couleurs pour les images sous-marines. Dans le contexte de la localisation visuelle, nous avons proposé une nouvelle fonction de coût pour la régression de pose appliquée aux méthodes d'apprentissage profond. Enfin, nous avons évalué plusieurs méthodes de localisation visuelle sur le jeu de données établi. Notre analyse a identifié les étapes de la localisation visuelle les plus impactées par le milieu sous-marin et a suggéré des améliorations avec une faible quantité de données. De plus, nous avons montré que notre algorithme de correction des couleurs multi-vues améliore significativement les résultats de localisation visuelle.

Appendix B

Depth-based points alignment



(a) Point clouds displaced using a $Sim(3)$ transformation.

(b) Point clouds aligned in $Sim(3)$ using only vertical priors.

Figure B.1: **Aligning two point clouds only based on vertical priors.** The horizontal shift and the yaw angle cannot be determined, yet point clouds are both vertically aligned and oriented parallel to the ground.

In Section 3.4.1, the translation, orientation and scale of the model were retrieved using Umeyama's algorithm (Umeyama, 1991) on the positions of the cameras. This algorithm requires priors knowledge about the camera positions in 3D space along all three axes, *i.e.*, X, Y and Z axes. However, such information may not always be available. In particular, in underwater applications, estimating the position of the vehicle in 3D space requires acoustic positioning systems that can be expensive and hard to set up. Yet, measuring the depth of the acquisition system is much more common and accessible as it requires only a pressure sensor. As illustrated by Figure B.1, given a set of depth measurements, it is possible to retrieve the vertical alignment along the Z axis, the partial rotation for the pitch and roll angles, as well as the scale of the model. Since we lack position priors for the X and Y axes, we cannot retrieve the alignment along these axes,

and neither can we retrieve the yaw orientation, hence loosing three degrees of liberty. In this section, we detail an optimization procedure to retrieve the vertical alignment, the pitch and roll angles, as well as the scale of a set of 3D points solely based on depth measurements.

Let $\overline{z}_i$ be the depth measurement of a camera with 3D position $\mathbf{X}_i$. We want to estimate $\boldsymbol{T} \in Sim(3)$ the transformation that aligns these camera centers to their measured depths. We first express z_i, the vertical component of $\mathbf{X}_i$ after it has been transformed by $\boldsymbol{T}$:

$$z_i = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} \boldsymbol{T} \odot \mathbf{X}_i. \tag{B.1}$$

Our objective is to minimize the distance between the depth measurements $\overline{z}_i$ and the aligned camera vertical components z_i. Hence, we define our residuals as follows:

$$r_i = \overline{z}_i - z_i. \tag{B.2}$$

We then estimate $\boldsymbol{T}$ in a least squares manner:

$$\arg\min_{\boldsymbol{T}} \sum_i r_i^2. \tag{B.3}$$

We choose to estimate the transformation matrix $\boldsymbol{T}$ in a Gauss-Newton scheme. To achieve this, we need to start from an initial transformation: we choose $\boldsymbol{T} = \mathbf{I}$, where $\mathbf{I}$ is the identity matrix. The optimization procedure then updates iteratively the parameters of $\boldsymbol{T}$. However, since $\boldsymbol{T} \in Sim(3)$, it is not possible to perform the optimization directly on the values of the matrix, as some step of the optimization might not land in the $Sim(3)$ manifold. To cope with this problem, we express our transformation as:

$$\boldsymbol{T} = \boldsymbol{S} \cdot \boldsymbol{\Delta S}, \tag{B.4}$$

with

$$\boldsymbol{S} = \begin{bmatrix} s\boldsymbol{R} & \boldsymbol{t} \\ \mathbf{0}_{1\times 3} & 1 \end{bmatrix}, \tag{B.5}$$

and

$$\boldsymbol{\Delta S} = \begin{bmatrix} e^{\sigma} \exp([\boldsymbol{\omega}]_{\times}) & \boldsymbol{W}(\sigma, \boldsymbol{\omega})\boldsymbol{\rho} \\ \mathbf{0}_{1\times 3} & 1 \end{bmatrix}. \tag{B.6}$$

In this representation, $\boldsymbol{S} \in Sim(3)$ is the state of $\boldsymbol{T}$ at a given optimization step, with s, $\boldsymbol{R}$ and $\boldsymbol{t}$ the respective scale, rotation and translation components of $\boldsymbol{S}$. And $\boldsymbol{\Delta S} \in Sim(3)$ is the closed-form expression of any $Sim(3)$ transformation, with $\sigma, \boldsymbol{\omega}$ and $\boldsymbol{\rho}$ the corresponding $\mathfrak{sim}(3)$ respective scale, rotation and translation elements of $\boldsymbol{\Delta S}$ (Strasdat, 2012). The function $\boldsymbol{W}(\sigma, \boldsymbol{\omega})$ influences the translation component based on the scale and the rotation — its only important property for the rest of this demonstration is $\boldsymbol{W}(0, \mathbf{0}_3) = \mathbf{I}$.

At each optimization step t, we obtain the Jacobian matrix by computing the partial derivative of r_i with respect to σ, $\boldsymbol{\omega}$ and $\boldsymbol{\rho}$ at the point $\boldsymbol{\Delta S} = \mathbf{I}$, meaning $\sigma = 0$, $\boldsymbol{\omega} = \mathbf{0}_3$ and $\boldsymbol{\rho} = \mathbf{0}_3$, and proceed to the update $\boldsymbol{S}_{t+1} = \boldsymbol{S}_t \cdot \boldsymbol{\Delta S}$. The optimization is stopped when the absolute value of the cost difference between two successive optimization steps is lower than a given threshold, typically 10^{-12}. To compute the Jacobian matrix, we need to express r_i in terms of σ, $\boldsymbol{\omega}$ and $\boldsymbol{\rho}$. First, we compute the transformation of $\mathbf{X}_i$ by $\boldsymbol{T}$:

$$(\boldsymbol{S} \cdot \boldsymbol{\Delta S}) \odot \mathbf{X}_i = s\boldsymbol{R}\left(e^{\sigma} \exp\left([\boldsymbol{\omega}]_{\times}\right)\mathbf{X}_i + \boldsymbol{W}\left(\sigma, \boldsymbol{\omega}\right)\boldsymbol{\rho}\right) + \boldsymbol{t}. \tag{B.7}$$

And we decompose $\boldsymbol{R}$ and $\boldsymbol{t}$:

$$\boldsymbol{R} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{33} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \quad \text{and} \quad \boldsymbol{t} = \begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix}. \tag{B.8}$$

From Eqs. (B.7) and (B.8), we can express r_i in terms of $\sigma, \boldsymbol{\omega}$ and $\boldsymbol{\rho}$:

$$r_i = s \begin{bmatrix} r_{31} & r_{32} & r_{33} \end{bmatrix} \left(e^{\sigma} \exp\left([\boldsymbol{\omega}]_{\times}\right)\mathbf{X}_i + \boldsymbol{W}\left(\sigma, \boldsymbol{\omega}\right)\boldsymbol{\rho}\right) + t_z \tag{B.9}$$

Finally, we can compute the partial derivative of r_i with respect to $\sigma, \boldsymbol{\omega}$ and $\boldsymbol{\rho}$:

$$\left.\frac{\partial r_i}{\partial \sigma}\right|_{\boldsymbol{\omega}=\mathbf{0}_3,\, \boldsymbol{\rho}=\mathbf{0}_3,\, \sigma=0} = -s \begin{bmatrix} r_{31} & r_{32} & r_{33} \end{bmatrix} \cdot \mathbf{X}_i, \tag{B.10}$$

$$\left.\frac{\partial r_i}{\partial \boldsymbol{\omega}}\right|_{\boldsymbol{\omega}=\mathbf{0}_3,\, \boldsymbol{\rho}=\mathbf{0}_3,\, \sigma=0} = s \begin{bmatrix} r_{31} & r_{32} & r_{33} \end{bmatrix} \times \mathbf{X}_i, \tag{B.11}$$

$$\left.\frac{\partial r_i}{\partial \boldsymbol{\rho}}\right|_{\boldsymbol{\omega}=\mathbf{0}_3,\, \boldsymbol{\rho}=\mathbf{0}_3,\, \sigma=0} = -s \begin{bmatrix} r_{31} & r_{32} & r_{33} \end{bmatrix}. \tag{B.12}$$

The Jacobian matrix can then be computed from these partial derivatives.

Appendix C

Shifting a normal distribution

This appendix outlines the process of deriving the parameters of the normal distribution for variable $\boldsymbol{I}_{c,p}$ from the parameters of the normal distribution for variable $\boldsymbol{J}_{c,p}$. In Section 4.2, we made the assumption:

$$\boldsymbol{J}_{c,p} \sim \mathcal{N}(\mu_c, \sigma_c^2). \tag{C.1}$$

Due to the property that the normal distribution family is preserved under linear transformations, this implies that if a variable X follows a normal distribution with a mean of μ and a variance of σ^2, then a variable of the form $aX + b$, where a and b are any real numbers, also follows a normal distribution. In this case, it has a mean of $a\mu + b$ and a variance of $a^2\sigma^2$. Using this property and following the underwater image formation model described by Eq. (4.1), we can deduce:

$$\boldsymbol{J}_{c,p}e^{-\beta_c \boldsymbol{z}_p} + B_c(1 - e^{-\gamma_c \boldsymbol{z}_p}) \sim \mathcal{N}\left(\mu_c e^{-\beta_c \boldsymbol{z}_p} + B_c(1 - e^{-\gamma_c \boldsymbol{z}_p}), \left(e^{-\beta_c \boldsymbol{z}_p}\sigma_c\right)^2\right). \tag{C.2}$$

Then, by using the notations

$$\boldsymbol{m}_{c,p} = \mu_c e^{-\beta_c \boldsymbol{z}_p} + B_c(1 - e^{-\gamma_c \boldsymbol{z}_p}) \tag{C.3}$$

and

$$\boldsymbol{s}_{c,p} = \sigma_c e^{-\beta_c \boldsymbol{z}_p}, \tag{C.4}$$

we retrieve the expression of Eq. (4.3):

$$\boldsymbol{I}_{c,p} = \boldsymbol{J}_{c,p}e^{-\beta_c \boldsymbol{z}_p} + B_c(1 - e^{-\gamma_c \boldsymbol{z}_p}) \sim \mathcal{N}(\boldsymbol{m}_{c,p}, \boldsymbol{s}_{c,p}^2). \tag{C.5}$$

Appendix D

Closed-form integral of the Homography loss

This appendix provides a step-by-step explanation of the mathematical derivation for the integral form of the Homography loss described in Eq. (5.20). More specifically we aim to solve the following integral:

$$\mathcal{L}_{\mathrm{H}} = \frac{1}{x_{\max} - x_{\min}} \int_{x_{\min}}^{x_{\max}} \|\mathbf{I} - \boldsymbol{H}\|_F^2 \, \mathrm{d}x. \tag{D.1}$$

For the sake of clarity, we note:

$$\boldsymbol{M} = \mathbf{I} - \boldsymbol{H}. \tag{D.2}$$

We know that the squared Frobenius norm of $\boldsymbol{M} \in \mathbb{R}^{3\times3}$ is equal to the trace of its multiplication with its transpose:

$$\|\boldsymbol{M}\|_F^2 = \mathrm{Tr}\left(\boldsymbol{M}\boldsymbol{M}^T\right). \tag{D.3}$$

We first develop the integrated part to express it in terms of x:

$$\begin{aligned}
\boldsymbol{M}\boldsymbol{M}^T &= \mathbf{I} - \left(\boldsymbol{R} - \frac{\boldsymbol{t}\boldsymbol{n}^T}{x}\right)^T - \left(\boldsymbol{R} - \frac{\boldsymbol{t}\boldsymbol{n}^T}{x}\right) + \left(\boldsymbol{R} - \frac{\boldsymbol{t}\boldsymbol{n}^T}{x}\right)\left(\boldsymbol{R} - \frac{\boldsymbol{t}\boldsymbol{n}^T}{x}\right)^T && \text{(D.4)}\\
&= \mathbf{I} - \boldsymbol{R}^T + \frac{\boldsymbol{n}\boldsymbol{t}^T}{x} - \boldsymbol{R} + \frac{\boldsymbol{t}\boldsymbol{n}^T}{x} + \boldsymbol{R}\boldsymbol{R}^T - \boldsymbol{R}\frac{\boldsymbol{n}\boldsymbol{t}^T}{x} - \frac{\boldsymbol{t}\boldsymbol{n}^T}{x}\boldsymbol{R}^T + \frac{\boldsymbol{t}\boldsymbol{n}^T\boldsymbol{n}\boldsymbol{t}^T}{x^2} && \text{(D.5)}\\
&= \mathbf{A} + \frac{1}{x}\mathbf{B} + \frac{1}{x^2}\mathbf{C} && \text{(D.6)}
\end{aligned}$$

with

$$\mathbf{A} = (\mathbf{I} - \boldsymbol{R})(\mathbf{I} - \boldsymbol{R})^T, \tag{D.7}$$

$$\mathbf{B} = (\mathbf{I} - \boldsymbol{R})\boldsymbol{n}\boldsymbol{t}^T + \left((\mathbf{I} - \boldsymbol{R})\boldsymbol{n}\boldsymbol{t}^T\right)^T, \tag{D.8}$$

$$\mathbf{C} = \boldsymbol{t}\boldsymbol{n}^T\left(\boldsymbol{t}\boldsymbol{n}^T\right)^T. \tag{D.9}$$

We then solve the integral using Eq. (D.6):

$$\int_{x_{\min}}^{x_{\max}} \|\boldsymbol{M}\|_F^2 \, \mathrm{d}x = \int_{x_{\min}}^{x_{\max}} \mathrm{Tr}\left(\mathbf{A} + \frac{1}{x}\mathbf{B} + \frac{1}{x^2}\mathbf{C}\right) \tag{D.10}$$

$$= \mathrm{Tr}\left(\mathbf{A}(x_{\max} - x_{\min}) + \mathbf{B}\left(\log(x_{\max}) - \log(x_{\min})\right) + \mathbf{C}\left(\frac{1}{x_{\min}} - \frac{1}{x_{\max}}\right)\right) \tag{D.11}$$

After the normalization on the integration range, we obtain the closed form solution of our loss:

$$\mathcal{L}_{\mathrm{H}} = \mathrm{Tr}\left(\mathbf{A} + \mathbf{B}\,\frac{\log\left(x_{\max}/x_{\min}\right)}{x_{\max} - x_{\min}} + \frac{\mathbf{C}}{x_{\min} \cdot x_{\max}}\right). \tag{D.12}$$

Bibliography

Akkaynak, D. and Treibitz, T. (2018). A Revised Underwater Image Formation Model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6723–6732.

Akkaynak, D. and Treibitz, T. (2019). Sea-Thru: A Method for Removing Water From Underwater Images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1682–1691.

Akkaynak, D., Treibitz, T., Shlesinger, T., Loya, Y., Tamir, R., and Iluz, D. (2017). What Is the Space of Attenuation Coefficients in Underwater Computer Vision? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4931–4940.

Ancuti, C., Ancuti, C. O., Haber, T., and Bekaert, P. (2012). Enhancing underwater images and videos by fusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 81–88.

Ancuti, C. O., Ancuti, C., De Vleeschouwer, C., and Garcia, R. (2017). Locally Adaptive Color Correction for Underwater Image Dehazing and Matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 997–1005.

Arandjelovic, R., Gronat, P., Torii, A., Pajdla, T., and Sivic, J. (2016). NetVLAD: CNN Architecture for Weakly Supervised Place Recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5297–5307.

Arnaubec, A., Opderbecke, J., Allais, A.-G., and Brignone, L. (2015). Optical mapping with the ARIANE HROV at IFREMER: The MATISSE processing tool. In *OCEANS*, pages 1–6.

Arnold, E., Wynn, J., Vicente, S., Garcia-Hernando, G., Monszpart, Á., Prisacariu, V., Turmukhambetov, D., and Brachmann, E. (2022). Map-Free Visual Relocalization: Metric Pose Relative to a Single Image. In *European Conference on Computer Vision*, pages 690–708.

Audenhaege, L. V., Matabos, M., Brind'Amour, A., Drugmand, J., Laës-Huon, A., Sarradin, P.-M., and Sarrazin, J. (2022). Long-term monitoring reveals unprecedented stability of a vent mussel assemblage on the Mid-Atlantic Ridge. *Progress in Oceanography*, 204:102791.

Balntas, V., Lenc, K., Vedaldi, A., and Mikolajczyk, K. (2017). HPatches: A Benchmark and Evaluation of Handcrafted and Learned Local Descriptors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5173–5182.

Bay, H., Tuytelaars, T., and Van Gool, L. (2006). SURF: Speeded Up Robust Features. In *European Conference on Computer Vision*, pages 404–417.

Benbihi, A. (2020). *Robust Visual Features for Long-Term Monitoring*. PhD thesis, CentraleSupélec.

Benbihi, A., Arravechia, S., Geist, M., and Pradalier, C. (2020). Image-Based Place Recognition on Bucolic Environment Across Seasons From Semantic Edge Description. In *IEEE International Conference on Robotics and Automation*, pages 3032–3038.

Berman, D., Levy, D., Avidan, S., and Treibitz, T. (2021). Underwater Single Image Color Restoration Using Haze-Lines and a New Quantitative Dataset. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(8):2822–2837.

Berman, D., treibitz, T., and Avidan, S. (2016). Non-Local Image Dehazing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Berman, D., Treibitz, T., and Avidan, S. (2017). Color Restoration of Underwater Images. In *British Machine Vision Conference*, pages 44.1–44.12. BMVA Press.

Boittiaux, C., Dune, C., Arnaubec, A., Marxer, R., Ferrera, M., and Hugel, V. (2023a). Long-term visual localization in deep-sea underwater environment. In *ORASIS*, Carqueiranne.

Boittiaux, C., Dune, C., Ferrera, M., Arnaubec, A., Marxer, R., Matabos, M., Audenhaege, L. V., and Hugel, V. (2023b). Eiffel Tower: A deep-sea underwater dataset for long-term visual localization. *The International Journal of Robotics Research*, 42(9):689–699.

Boittiaux, C., Marxer, R., Dune, C., Arnaubec, A., Ferrera, M., and Hugel, V. (2024). SUCRe: Leveraging Scene Structure for Underwater Color Restoration. In *International Conference on 3D Vision*.

Boittiaux, C., Marxer, R., Dune, C., Arnaubec, A., and Hugel, V. (2022). Homography-Based Loss Function for Camera Pose Regression. *IEEE Robotics and Automation Letters*, 7(3):6242–6249.

Brachmann, E., Cavallari, T., and Prisacariu, V. A. (2023). Accelerated Coordinate Encoding: Learning to Relocalize in Minutes Using RGB and Poses. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5044–5053.

Brachmann, E., Humenberger, M., Rother, C., and Sattler, T. (2021). On the Limits of Pseudo Ground Truth in Visual Camera Re-Localisation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6218–6228.

Brachmann, E., Krull, A., Nowozin, S., Shotton, J., Michel, F., Gumhold, S., and Rother, C. (2017). DSAC - Differentiable RANSAC for Camera Localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6684–6692.

Brachmann, E. and Rother, C. (2018). Learning Less Is More - 6D Camera Localization via 3D Surface Regression. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Brachmann, E. and Rother, C. (2022). Visual Camera Re-Localization From RGB and RGB-D Images Using DSAC. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(9):5847–5865.

Bryson, M., Johnson-Roberson, M., Pizarro, O., and Williams, S. B. (2015). True Color Correction of Autonomous Underwater Vehicle Imagery. *Journal of Field Robotics*, 33(6):853–874.

Brégier, R. (2021). Deep Regression on Manifolds: A 3D Rotation Case Study. In *International Conference on 3D Vision*, pages 166–174.

Calonder, M., Lepetit, V., Strecha, C., and Fua, P. (2010). BRIEF: Binary Robust Independent Elementary Features. In *European Conference on Computer Vision*, pages 778–792.

Cannat, M. and Sarradin, P.-M. (2010). MOMARSAT : MONITORING THE MID ATLANTIC RIDGE.

Cernea, D. (2020). OpenMVS: Multi-View Stereo Reconstruction Library. *arXiv preprint*.

Chen, H., Luo, Z., Zhou, L., Tian, Y., Zhen, M., Fang, T., McKinnon, D., Tsin, Y., and Quan, L. (2022a). ASpanFormer: Detector-Free Image Matching

with Adaptive Span Transformer. In *European Conference on Computer Vision*, pages 20–36.

Chen, J., Yin, Y., Birdal, T., Chen, B., Guibas, L. J., and Wang, H. (2022b). Projective Manifold Gradient Layer for Deep Rotation Regression. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6646–6655.

Chiang, J. Y. and Chen, Y.-C. (2012). Underwater Image Enhancement by Wavelength Compensation and Dehazing. *IEEE Transactions on Image Processing*, 21(4):1756–1769.

DeTone, D., Malisiewicz, T., and Rabinovich, A. (2018). SuperPoint: Self-Supervised Interest Point Detection and Description. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 224–236.

Duntley, S. Q. (1963). Light in the Sea. *Journal of the Optical Society of America*, 53(2):214–233.

Dusmanu, M., Rocco, I., Pajdla, T., Pollefeys, M., Sivic, J., Torii, A., and Sattler, T. (2019). D2-Net: A Trainable CNN for Joint Description and Detection of Local Features. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8092–8101.

Ebner, M. and Hansen, J. (2013). Depth map color constancy. *Bio-Algorithms and Med-Systems*, 9(4):167–177.

Edstedt, J., Bökman, G., Wadenbäck, M., and Felsberg, M. (2023a). DeDoDe: Detect, Don't Describe – Describe, Don't Detect for Local Feature Matching. *arXiv preprint*.

Edstedt, J., Sun, Q., Bökman, G., Wadenbäck, M., and Felsberg, M. (2023b). RoMa: Revisiting Robust Losses for Dense Feature Matching. *arXiv preprint*.

Ferrera, M. (2019). *Monocular Visual-Inertial-Pressure Fusion for Underwater Localization and 3D Mapping.* PhD thesis, Université de Montpellier.

Ferrera, M., Creuze, V., Moras, J., and Trouvé-Peloux, P. (2019). AQUALOC: An underwater dataset for visual–inertial–pressure localization. *The International Journal of Robotics Research*, 38(14):1549–1559.

Ferrera, M., Eudes, A., Moras, J., Sanfourche, M., and Le Besnerais, G. (2021). OV^{2}SLAM: A Fully Online and Versatile Visual SLAM for Real-Time Applications. *IEEE Robotics and Automation Letters*, 6(2):1399–1406.

Fischler, M. A. and Bolles, R. C. (1981). Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. *Communications of the ACM*, 24(6):381–395.

Gao, X.-S., Hou, X.-R., Tang, J., and Cheng, H.-F. (2003). Complete solution classification for the perspective-three-point problem. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(8):930–943.

Garcia-Fidalgo, E. and Ortiz, A. (2015). Vision-based topological mapping and localization methods: A survey. *Robotics and Autonomous Systems*, 64:1–20.

Gilmer, J., Schoenholz, S. S., Riley, P. F., Vinyals, O., and Dahl, G. E. (2017). Neural Message Passing for Quantum Chemistry. In *International Conference on Machine Learning*, volume 70, pages 1263–1272.

Girard, F., Sarrazin, J., Arnaubec, A., Cannat, M., Sarradin, P.-M., Wheeler, B., and Matabos, M. (2020). Currents and topography drive assemblage distribution on an active hydrothermal edifice. *Progress in Oceanography*, 187:102397.

Glocker, B., Izadi, S., Shotton, J., and Criminisi, A. (2013). Real-time RGB-D camera relocalization. In *International Symposium on Mixed and Augmented Reality*, pages 173–179.

Gordo, A., Almazán, J., Revaud, J., and Larlus, D. (2016). Deep Image Retrieval: Learning Global Representations for Image Search. In *European Conference on Computer Vision*, pages 241–257.

Griffith, S., Chahine, G., and Pradalier, C. (2017). Symphony Lake Dataset. *The International Journal of Robotics Research*, 36(11):1151–1158.

Guerrero-Font, E., Massot-Campos, M., Negre, P. L., Bonin-Font, F., and Codina, G. O. (2016). An USBL-aided multisensor navigation system for field AUVs. In *IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems*, pages 430–435.

Hartley, R. and Zisserman, A. (2003). *Multiple View Geometry in Computer Vision*. Cambridge University Press.

He, K., Sun, J., and Tang, X. (2010). Single image haze removal using dark channel prior. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(12):2341–2353.

Humenberger, M., Cabon, Y., Guerin, N., Morat, J., Leroy, V., Revaud, J., Rerole, P., Pion, N., de Souza, C., and Csurka, G. (2022). Robust Image Retrieval-based Visual Localization using Kapture. *arXiv preprint*.

Irschara, A., Zach, C., Frahm, J.-M., and Bischof, H. (2009). From structure-from-motion point clouds to fast location recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2599–2606.

Islam, M. J., Xia, Y., and Sattar, J. (2020). Fast Underwater Image Enhancement for Improved Visual Perception. *IEEE Robotics and Automation Letters*, 5(2):3227–3234.

Jaffe, J. (1990). Computer modeling and the design of optimal underwater imaging systems. *IEEE Journal of Oceanic Engineering*, 15(2):101–111.

Jiang, Q., Gu, Y., Li, C., Cong, R., and Shao, F. (2022). Underwater Image Enhancement Quality Evaluation: Benchmark Dataset and Objective Metric. *IEEE Transactions on Circuits and Systems for Video Technology*, 32(9):5959–5974.

Jégou, H., Douze, M., Schmid, C., and Pérez, P. (2010). Aggregating local descriptors into a compact image representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3304–3311.

Kendall, A. and Cipolla, R. (2017). Geometric Loss Functions for Camera Pose Regression With Deep Learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5974–5983.

Kendall, A., Grimes, M., and Cipolla, R. (2015). PoseNet: A Convolutional Network for Real-Time 6-DOF Camera Relocalization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*.

Kingma, D. P. and Ba, J. (2015). Adam: A Method for Stochastic Optimization. In *The International Conference on Learning Representations*.

Kneip, L., Li, H., and Seo, Y. (2014). UPnP: An Optimal O(n) Solution to the Absolute Pose Problem with Universal Applicability. In *European Conference on Computer Vision*, pages 127–142.

Langmuir, C., Charlou, J.-L., Colodner, D., Costa, I., Desbruyeres, D., Desonie, D., Emerson, T., Fornari, D., Fouquet, Y., Humphris, S., Fiala-Medioni, A., Saldanha, L., Sours-Page, R., Thatcher, M., Tivey, M., Dover, C., Damm, K., Wiese, K., and Wilson, C. (1993). Lucky Strike - A newly discovered hydrothermal site on the Azores platform. *Ridge Events*, 4(2):3–5.

Lepetit, V., Moreno-Noguer, F., and Fua, P. (2009). EPnP: An Accurate O(n) Solution to the PnP Problem. *International Journal of Computer Vision*, 81(2):155–166.

Levy, D., Peleg, A., Pearl, N., Rosenbaum, D., Akkaynak, D., Korman, S., and Treibitz, T. (2023). SeaThru-NeRF: Neural Radiance Fields in Scattering Media. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 56–65.

Li, C., Anwar, S., Hou, J., Cong, R., Guo, C., and Ren, W. (2021). Underwater Image Enhancement via Medium Transmission-Guided Multi-Color Space Embedding. *IEEE Transactions on Image Processing*, 30:4985–5000.

Li, C., Guo, C., Ren, W., Cong, R., Hou, J., Kwong, S., and Tao, D. (2020). An Underwater Image Enhancement Benchmark Dataset and Beyond. *IEEE Transactions on Image Processing*, 29:4376–4389.

Li, Y., Snavely, N., Huttenlocher, D., and Fua, P. (2012). Worldwide Pose Estimation Using 3D Point Clouds. In *European Conference on Computer Vision*, pages 15–29.

Lindenberger, P., Sarlin, P.-E., and Pollefeys, M. (2023). LightGlue: Local Feature Matching at Light Speed. *Proceedings of the IEEE/CVF International Conference on Computer Vision*.

Liu, C., Kim, K., Gu, J., Furukawa, Y., and Kautz, J. (2019). PlaneRCNN: 3D Plane Detection and Reconstruction From a Single Image. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4450–4459.

Liu, R., Jiang, Z., Yang, S., and Fan, X. (2022). Twin Adversarial Contrastive Learning for Underwater Image Enhancement and Beyond. *IEEE Transactions on Image Processing*, 31:4922–4936.

Lowe, D. (1999). Object recognition from local scale-invariant features. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, volume 2, pages 1150–1157.

Lowe, D. G. (2004). Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision*, 60(2):91–110.

Maddern, W., Pascoe, G., Linegar, C., and Newman, P. (2017). 1 Year, 1000km: The Oxford RobotCar Dataset. *The International Journal of Robotics Research*, 36(1):3–15.

Mallios, A., Vidal, E., Campos, R., and Carreras, M. (2017). Underwater caves sonar data set. *The International Journal of Robotics Research*, 36(12):1247–1251.

Martin-Brualla, R., Radwan, N., Sajjadi, M. S. M., Barron, J. T., Dosovitskiy, A., and Duckworth, D. (2021). NeRF in the Wild: Neural Radiance Fields for Unconstrained Photo Collections. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7210–7219.

Matabos, M., Barreyre, T., Juniper, S. K., Cannat, M., Kelley, D., Alfaro-Lucas, J. M., Chavagnac, V., Colaço, A., Escartin, J., Escobar, E., Fornari, D., Hasenclever, J., Huber, J. A., Laës-Huon, A., Lantéri, N., Levin, L. A., Mihaly, S., Mittelstaedt, E., Pradillon, F., Sarradin, P.-M., Sarrazin, J., Tomasi, B., Venkatesan, R., and Vic, C. (2022). Integrating Multidisciplinary Observations in Vent Environments (IMOVE): Decadal Progress in Deep-Sea Observatories at Hydrothermal Vents. *Frontiers in Marine Science*, 9.

McGlamery, B. L. (1980). A Computer Model For Underwater Camera Systems. In Duntley, S. Q., editor, *Ocean Optics VI*, volume 208, pages 221–231. International Society for Optics and Photonics, SPIE.

Menna, F., Nocerino, E., Ural, S., and Gruen, A. (2020). MITIGATING IMAGE RESIDUALS SYSTEMATIC PATTERNS IN UNDERWATER PHOTOGRAMMETRY. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XLIII-B2-2020:977–984.

Mildenhall, B., Srinivasan, P. P., Tancik, M., Barron, J. T., Ramamoorthi, R., and Ng, R. (2020). NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis. In *European Conference on Computer Vision*.

Moreau, A., Piasco, N., Tsishkou, D., Stanciulescu, B., and Fortelle, A. d. L. (2022). LENS: Localization enhanced by NeRF synthesis. In *Proceedings of the 5th Conference on Robot Learning*, volume 164, pages 1347–1356.

Mur-Artal, R., Montiel, J. M. M., and Tardós, J. D. (2015). ORB-SLAM: A Versatile and Accurate Monocular SLAM System. *IEEE Transactions on Robotics*, 31(5):1147–1163.

Nakath, D., She, M., Song, Y., and Köser, K. (2021). In-Situ Joint Light and Medium Estimation for Underwater Color Restoration. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, pages 3724–3733.

Nelder, J. A. and Mead, R. (1965). A Simplex Method for Function Minimization. *The Computer Journal*, 7(4):308–313.

Newcombe, R. A., Izadi, S., Hilliges, O., Molyneaux, D., Kim, D., Davison, A. J., Kohi, P., Shotton, J., Hodges, S., and Fitzgibbon, A. (2011). KinectFusion: Real-

time dense surface mapping and tracking. In *International Symposium on Mixed and Augmented Reality*, pages 127–136.

Nielsen, M. C., Leonhardsen, M. H., and Schjølberg, I. (2019). Evaluation of PoseNet for 6-DOF Underwater Pose Estimation. In *OCEANS*, pages 1–6.

Oliva, A. and Torralba, A. (2001). Modeling the Shape of the Scene: A Holistic Representation of the Spatial Envelope. *International Journal of Computer Vision*, 42(3):145–175.

Ono, Y., Trulls, E., Fua, P., and Yi, K. M. (2018). LF-Net: Learning Local Features from Images. In *Advances in Neural Information Processing Systems*, volume 31.

Panek, V., Kukelova, Z., and Sattler, T. (2022). MeshLoc: Mesh-Based Visual Localization. In *European Conference on Computer Vision*, pages 589–609.

Panetta, K., Gao, C., and Agaian, S. (2016). Human-Visual-System-Inspired Underwater Image Quality Measures. *IEEE Journal of Oceanic Engineering*, 41(3):541–551.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., and Chintala, S. (2019). PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *Advances in Neural Information Processing Systems*, volume 32.

Philbin, J., Chum, O., Isard, M., Sivic, J., and Zisserman, A. (2007). Object retrieval with large vocabularies and fast spatial matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1–8.

Piasco, N., Sidibé, D., Demonceaux, C., and Gouet-Brunet, V. (2019a). Geometric Camera Pose Refinement with Learned Depth Maps. In *IEEE International Conference on Image Processing*, pages 2561–2565.

Piasco, N., Sidibé, D., Demonceaux, C., and Gouet-Brunet, V. (2019b). Perspective-n-Learned-Point: Pose Estimation from Relative Depth. In *British Machine Vision Conference*, pages 192.1–192.15.

Pietrantoni, M., Humenberger, M., Sattler, T., and Csurka, G. (2023). SegLoc: Learning Segmentation-Based Representations for Privacy-Preserving Visual Localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15380–15391.

Radenović, F., Tolias, G., and Chum, O. (2016). CNN Image Retrieval Learns from BoW: Unsupervised Fine-Tuning with Hard Examples. In *European Conference on Computer Vision*, pages 3–20.

Ranftl, R., Bochkovskiy, A., and Koltun, V. (2021). Vision Transformers for Dense Prediction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12179–12188.

Ranftl, R., Lasinger, K., Hafner, D., Schindler, K., and Koltun, V. (2022). Towards Robust Monocular Depth Estimation: Mixing Datasets for Zero-Shot Cross-Dataset Transfer. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(3):1623–1637.

Revaud, J., Weinzaepfel, P., de Souza, C. R., and Humenberger, M. (2019). R2D2: Repeatable and Reliable Detector and Descriptor. In *Advances in Neural Information Processing Systems*.

Riba, E., Mishkin, D., Ponsa, D., Rublee, E., and Bradski, G. (2020). Kornia: an Open Source Differentiable Computer Vision Library for PyTorch. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*.

Ronneberger, O., Fischer, P., and Brox, T. (2015). U-Net: Convolutional Networks for Biomedical Image Segmentation. In *Medical Image Computing and Computer-Assisted Intervention*, pages 234–241.

Rublee, E., Rabaud, V., Konolige, K., and Bradski, G. (2011). ORB: An efficient alternative to SIFT or SURF. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2564–2571.

Ruhl, H. A., André, M., Beranzoli, L., Çağatay, M. N., Colaço, A., Cannat, M., Dañobeitia, J. J., Favali, P., Géli, L., Gillooly, M., Greinert, J., Hall, P. O., Huber, R., Karstensen, J., Lampitt, R. S., Larkin, K. E., Lykousis, V., Mienert, J., Miguel Miranda, J., Person, R., Priede, I. G., Puillat, I., Thomsen, L., and Waldmann, C. (2011). Societal need for improved understanding of climate change, anthropogenic impacts, and geo-hazard warning drive development of ocean observatories in European Seas. *Progress in Oceanography*, 91(1):1–33.

Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., and Chen, L.-C. (2018). MobileNetV2: Inverted Residuals and Linear Bottlenecks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4510–4520.

Sarlin, P.-E., Cadena, C., Siegwart, R., and Dymczyk, M. (2019). From Coarse to Fine: Robust Hierarchical Localization at Large Scale. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12716–12725.

Sarlin, P.-E., DeTone, D., Malisiewicz, T., and Rabinovich, A. (2020). SuperGlue: Learning Feature Matching with Graph Neural Networks. In *Proceedings of the*

IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 4938–4947.

Sarlin, P.-E., Dusmanu, M., Schönberger, J. L., Speciale, P., Gruber, L., Larsson, V., Miksik, O., and Pollefeys, M. (2022). LaMAR: Benchmarking Localization and Mapping for Augmented Reality. In *European Conference on Computer Vision*, pages 686–704.

Sarlin, P.-E., Unagar, A., Larsson, M., Germain, H., Toft, C., Larsson, V., Pollefeys, M., Lepetit, V., Hammarstrand, L., Kahl, F., and Sattler, T. (2021). Back to the Feature: Learning Robust Camera Localization From Pixels To Pose. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3247–3257.

Sattler, T. (2013). *Efficient & Effective Image-Based Localization*. PhD thesis, RWTH Aachen University.

Sattler, T., Leibe, B., and Kobbelt, L. (2012a). Improving Image-Based Localization by Active Correspondence Search. In *European Conference on Computer Vision*, pages 752–765.

Sattler, T., Leibe, B., and Kobbelt, L. (2017). Efficient & Effective Prioritized Matching for Large-Scale Image-Based Localization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(9):1744–1756.

Sattler, T., Maddern, W., Toft, C., Torii, A., Hammarstrand, L., Stenborg, E., Safari, D., Okutomi, M., Pollefeys, M., Sivic, J., Kahl, F., and Pajdla, T. (2018). Benchmarking 6DOF Outdoor Visual Localization in Changing Conditions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8601–8610.

Sattler, T., Weyand, T., Leibe, B., and Kobbelt, L. P. (2012b). Image Retrieval for Image-Based Localization Revisited. In *British Machine Vision Conference*.

Sattler, T., Zhou, Q., Pollefeys, M., and Leal-Taixe, L. (2019). Understanding the Limitations of CNN-Based Absolute Camera Pose Regression. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3302–3312.

Schechner, Y. and Karpel, N. (2005). Recovery of underwater visibility and structure by polarization analysis. *IEEE Journal of Oceanic Engineering*, 30(3):570–587.

Schönberger, J. L. and Frahm, J.-M. (2016). Structure-from-Motion Revisited. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4104–4113.

Schonberger, J. L., Hardmeier, H., Sattler, T., and Pollefeys, M. (2017). Comparative Evaluation of Hand-Crafted and Learned Local Features. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1482–1491.

Schönberger, J. L., Price, T., Sattler, T., Frahm, J.-M., and Pollefeys, M. (2017). A Vote-and-Verify Strategy for Fast Spatial Verification in Image Retrieval. In *Proceedings of the Asian Conference on Computer Vision*, pages 321–337.

Sethuraman, A. V., Ramanagopal, M. S., and Skinner, K. A. (2022). WaterNeRF: Neural Radiance Fields for Underwater Scenes. *arXiv preprint*.

Sharma, G., Wu, W., and Dalal, E. N. (2005). The CIEDE2000 color-difference formula: Implementation notes, supplementary test data, and mathematical observations. *Color Research & Application*, 30(1):21–30.

Shavit, Y., Ferens, R., and Keller, Y. (2021). Learning Multi-Scene Absolute Pose Regression With Transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2733–2742.

She, M., Nakath, D., Song, Y., and Köser, K. (2022). Refractive geometry for underwater domes. *ISPRS Journal of Photogrammetry and Remote Sensing*, 183:525–540.

She, M., Song, Y., Mohrmann, J., and Köser, K. (2019). Adjustment and Calibration of Dome Port Camera Systems for Underwater Vision. In *Pattern Recognition*, pages 79–92.

Shotton, J., Glocker, B., Zach, C., Izadi, S., Criminisi, A., and Fitzgibbon, A. (2013). Scene Coordinate Regression Forests for Camera Relocalization in RGB-D Images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2930–2937.

Sivic, J. and Zisserman, A. (2003). Video Google: a text retrieval approach to object matching in videos. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, volume 2, pages 1470–1477.

Strasdat, H. (2012). *Local accuracy and global consistency for efficient visual SLAM*. PhD thesis, Imperial College London.

Sun, J., Shen, Z., Wang, Y., Bao, H., and Zhou, X. (2021). LoFTR: Detector-Free Local Feature Matching With Transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8922–8931.

Sun, X., Xie, Y., Luo, P., and Wang, L. (2017). A Dataset for Benchmarking Image-Based Localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7436–7444.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., and Rabinovich, A. (2015). Going Deeper With Convolutions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1–9.

Taira, H., Okutomi, M., Sattler, T., Cimpoi, M., Pollefeys, M., Sivic, J., Pajdla, T., and Torii, A. (2018). InLoc: Indoor Visual Localization With Dense Matching and View Synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Tankovich, V., Hane, C., Zhang, Y., Kowdle, A., Fanello, S., and Bouaziz, S. (2021). HITNet: Hierarchical Iterative Tile Refinement Network for Real-time Stereo Matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14362–14372.

Teed, Z. and Deng, J. (2021). Tangent Space Backpropagation for 3D Transformation Groups. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10338–10347.

Tola, E., Lepetit, V., and Fua, P. (2010). DAISY: An Efficient Dense Descriptor Applied to Wide-Baseline Stereo. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(5):815–830.

Tyszkiewicz, M., Fua, P., and Trulls, E. (2020). DISK: Learning local features with policy gradient. In *Advances in Neural Information Processing Systems*, volume 33, pages 14254–14265.

Umeyama, S. (1991). Least-squares estimation of transformation parameters between two point patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(4):376–380.

Valentin, J., Dai, A., Niessner, M., Kohli, P., Torr, P., Izadi, S., and Keskin, C. (2016). Learning to Navigate the Energy Landscape. In *International Conference on 3D Vision*, pages 323–332.

Van Audenhaege, L., Sarrazin, J., Legendre, P., Perrois, G., Cannat, M., and Matabos, M. (2023). Monitoring ecological dynamics on complex hydrothermal structures: a novel photogrammetry approach reveals fine scales of faunal assemblage variability. Under review.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L. u., and Polosukhin, I. (2017). Attention is All you Need. In *Advances in Neural Information Processing Systems*, volume 30.

Yang, H., Shi, J., and Carlone, L. (2021). TEASER: Fast and Certifiable Point Cloud Registration. *IEEE Transactions on Robotics*, 37(2):314–333.

Yang, M. and Sowmya, A. (2015). An Underwater Color Image Quality Evaluation Metric. *IEEE Transactions on Image Processing*, 24(12):6062–6071.

Zhou, Q.-Y., Park, J., and Koltun, V. (2018). Open3D: A Modern Library for 3D Data Processing. *arXiv preprint*.

Zwilgmeyer, P. G. O., Yip, M., Teigen, A. L., Mester, R., and Stahl, A. (2021). The VAROS Synthetic Underwater Data Set: Towards Realistic Multi-Sensor Underwater Data With Ground Truth. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, pages 3722–3730.