TETHER-LESS CONTROL OF REMOTELY OPERATED VEHICLES

by

PENG LUYUAN

(B.Eng., Nanyang Technological University, Singapore)

A THESIS SUBMITTED FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY

DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

NATIONAL UNIVERSITY OF SINGAPORE

2025

Supervisors:

Associate Prof Mandar Chitre, Main Supervisor
Dr. Hari Vishnu, Co-Supervisor

Examiners:

Assistant Professor Mike Shou Zheng
Associate Professor Mehul Motani

Declaration

I hereby declare that this thesis is my original work and it has been written by me in its entirety. I have duly acknowledged all the sources of information which have been used in the thesis.

This thesis has also not been submitted for any degree in any university previously.

PENG Luyuan

November 2025

Acknowledgments

Although a PhD is probably the hardest-earned degree, my PhD years have been the most rewarding and fulfilling period of my life so far. It is difficult to fully capture my gratitude to the many people who supported and guided me throughout this incredible journey, but I will do my best here.

First and foremost, I would like to thank my thesis supervisors, A/Prof Mandar Chitre and Dr. Hari Vishnu. Mandar and Hari have gone far beyond the formal responsibilities of supervision. In the early stages, they gave me patience and space to develop my capabilities. As my research progressed, they offered trust and respect, empowering me to lead my own work. Whenever I had questions across diverse topics, they took the time to explain concepts and broaden my knowledge—even when my interests extended beyond the scope of my thesis. During times of self-doubt, they listened, guided, and supported me. They have not only taught me how to be a good researcher, but also how to be a better person. They are my role models.

I would also like to thank my friends and colleagues at ARL[1]. Special thanks go to Dr. Bharath Kalyan, Dr. Too Yuen Min, Dr. Rajat Mishra, and Ms. Soo Pieng Tan for their invaluable contributions to discussions, technical development, experimental deployments and data collection. Without their

[1]Acoustic Research Laboratory (ARL), Tropical Marine Science Institute (TMSI), National University of Singapore (NUS) - http://www.arl.nus.edu.sg/

support, this thesis would not have achieved its present quality. I am equally grateful to Dr. Li Kexin and Dr. Wu Shuangshuang, who have supported me through my ups and downs and generously shared their knowledge across different topics since I first joined.

I would also like to thank Dr. Basman Elhadidi, my final-year project supervisor at NTU. He helped me discover the joy of research and inspired my passion for field robotics. His encouragement fueled the beginning of this journey. My sincere thanks also go to A/Prof Mehul Motani and A/Prof Mike Shou Zheng, who provided valuable feedback during my qualifying exams.

Last but not least, I would like to thank my partner, Xuming, my family, and my dearest friends. They stood by me through difficult times, shared in my joys, and encouraged me to keep running along this path.

To all who have journeyed with me through these years, thank you.

Contents

Abstract

Tether-less Control of Remotely Operated Vehicles

by

PENG Luyuan

Doctor of Philosophy in Department of Electrical and Computer Engineering

National University of Singapore

Underwater inspection and intervention are essential for the maintenance of subsea infrastructure, maritime operations, and marine research. These missions are typically conducted using remotely operated vehicles (ROVs), which rely on real-time image transmission to support human-in-the-loop control. Most ROVs remain tethered to surface vessels, primarily to meet their power and communication needs. However, tethers introduce significant logistical challenges and operational costs, especially in confined or cluttered environments. Enabling tetherless ROV operation is therefore highly desirable for improving the flexibility and scalability of underwater missions. Achieving this, however, remains difficult due to the limitations of underwater wireless communication. The most commonly used method, acoustic communication, while effective over long ranges, does not provide sufficient bandwidth for transmitting the visual data required for real-time human supervision.

To address the challenge of real-time image transmission over acoustic links, this thesis proposes a novel image compression framework that achieves

high compression efficiency by leveraging prior knowledge of the inspection environment. Since underwater inspection missions are often repeated at the same sites, we exploit this availability of prior information by constructing a photorealistic 3D model of the scene. In subsequent reinspection missions, instead of transmitting full camera images, the system transmits compact latent representations—such as the estimated camera pose within the 3D model—along with a compressed difference image between the rendered view and the actual camera image.

To support this compression pipeline, we evaluate the performance of various novel view synthesis (NVS) techniques for reconstructing underwater structures, focusing on their ability to produce high-fidelity reconstructions efficiently under underwater conditions. We design a deep learning-based visual relocalization method that integrates navigation sensor inputs with augmented training data generated from the NVS model. This approach improves localization accuracy and stability, enabling reliable extraction of the latent representation. To further reduce transmission size and enhance reconstruction fidelity, we develop a latent optimization technique that refines the estimated latent by minimizing the discrepancy between the rendered and observed images through gradient descent on the NVS model.

We validate the proposed method in both controlled indoor test facilities and real seawater deployments. Experiments demonstrate that it significantly outperforms traditional image compression methods in terms of both compression efficiency and visual quality. The system also proves robust to

common underwater challenges, including turbidity, lighting variation, and scene changes. By leveraging prior knowledge for efficient image representation, this work provides a viable solution for enabling tether-less ROV operation in underwater inspection scenarios.

List of Figures

List of Tables

List of Abbreviations

ROV	Remotely Operated Vehicle
NVS	Novel View Synthesis
kbps	kilobits per second
3D	3-dimensional
NeRF	Neural Radiance Fields
EKF	Extended Kalman Filter
GAN	Generative Adversarial Networks
MVS	MultiView Stereo
SfM	Structure-from-Motion
3D-GS	3D Gaussian Splatting
MLP	Multilayer Perceptrons
INS	Inertial Navigation System
DVL	Doppler Velocity Logger
SLAM	Simultaneous Localization and Mapping
VO	Visual Odometry
RANSAC	Random Sampling Consensus
6-DOF	6-degree-of-freedom
CNN	Convolutional Neural Networks

DOB	Deep Ocean Basin
TCOMS	Technology Centre for Offshore and Marine, Singapore
TBW	Torpedo Boat Wreck
PSNR	Peak Signal to Noise Ratio
MS-SSIM	Multi-Scale Structural Similarity Index Measure
LPIPS	Learned Perceptual Image Patch Similarity
SJI	St. Johns Island
FC	Fully Connected
LSTM	Long Short-Term Memory
DCNN	Deep Convolutional Neural Networks
CDF	Cumulative Distribution Function
NFC	Navigation Flight Controller
USBL	Ultra-Short Baseline
ROS	Robot Operating System
MSE	Mean Squared Error
GPS	Global Positioning System
MAE	Mean Absolute Error
DSSIM	Structural Dissimilarity Index Measure

List of Symbols and Notations

$\mathcal{L}$	total loss function
$\mathcal{L}_{\mathbf{p}}$	position loss
$\mathcal{L}_{\theta}$	orientation loss (with respect to angles)
$\mathcal{L}_{\mathbf{q}}$	orientation loss (with respect to quaternion)
d	average distance from the camera to the object of interest
β	trade-off parameter between position and orientation loss
$\mathbf{p}$	true position
$\hat{\mathbf{p}}$	predicted position
$\mathbf{q}$	true orientation (quaternion)
$\hat{\mathbf{q}}$	predicted orientation (quaternion)
$\|\cdot\|_2$	Euclidean (L2) norm
$\mathbf{C}(\mathbf{r})$	observed pixel color along ray $\mathbf{r}$
$\hat{\mathbf{C}}(\mathbf{r})$	predicted pixel color along ray $\mathbf{r}$
$\mathbf{r}$	3D viewing ray
$\mathcal{L}^{\text{te}}$	transient embedding loss
$\beta(\mathbf{r})$	accumulated uncertainty along ray $\mathbf{r}$
$g(\mathbf{r})$	mean transient density along ray $\mathbf{r}$
λ_u	sparsity weight for transient density

$\mathcal{L}^{\text{rl}}$	robust loss
$w(\mathbf{r})$	binary weight for robust loss (inlier/outlier indicator)
L_{mse}	mean squared error between rendered and camera images
N	total number of pixels
$I_{\text{camera}}(i)$	RGB vector of the i^{th} pixel in the camera image
$I_{\text{rendered}}(i)$	RGB vector of the i^{th} pixel in the rendered image
L_{match}	keypoint matching loss
$\mathbf{k}_{\text{camera}}(i)$	location of i^{th} keypoint in the camera image
$\mathbf{k}_{\text{rendered}}(i)$	location of i^{th} keypoint in the rendered image
M	total number of keypoints
$\log_{10}$	base-10 logarithm

Chapter 1

Introduction

1.1 Background

1.1.1 Underwater inspection and intervention

Underwater inspection and intervention refer to monitoring, assessing, and maintaining structures and equipments located beneath the water's surface. They are a critical aspect of oceanic engineering as they optimize the lifespan of critical infrastructure.

Their applications span a diverse range of industries [1]. In the energy sector, these activities are critical for the management of infrastructure such as offshore oil platforms, underwater pipelines, and renewable energy systems like offshore wind farms [1]. In maritime construction, they ensure the stability and integrity of essential structures, including bridges, dams, and port facilities [2]. Furthermore, underwater inspection and intervention play a crucial role in defense operations, where they may be used to detect submerged threats, such as underwater mines, and to inspect naval vessels [3,

4].

Underwater inspection missions are both routine and technically complex. They are often repeated at the same site to monitor long-term changes such as structural degradation or biofouling accumulation [1, 5]. These routine assessments are essential for enabling timely maintenance, preventing costly failures, and extending the operational lifespan of underwater infrastructure. At the same time, each mission can involve intricate tasks—such as navigating around irregular structures, inspecting confined or obstructed areas, and executing precise manipulations [3, 6]. This combination of routine execution and technical difficulty places high demands on the reliability, precision, and adaptability of underwater inspection systems.

To meet these demands, current inspection and intervention tasks rely on either human divers or ROVs[1]. ROVs are robotic systems operated by humans from control stations, often equipped with high-definition cameras, sensors, and manipulators. Their use enables human-in-the-loop operations, which remain essential for ensuring accuracy, safety, and mission reliability—particularly in hazardous or deep-sea environments where direct human intervention is impractical[7, 8, 9].

1.1.2 Limitations of tethered ROVs

Traditional ROVs rely on a tether to stay connected to the control station, which supplies power, transmits control commands, and facilitates the exchange of sensor data [7]. While essential, the tether introduces

siginificant limitations.

Its use necessitates a heavy and bulky tether management system (shown in Fig. 1.1) and a large offshore support vessel (OSV), which restricts operations in high sea states and contributes to high operational expenditure. The daily charter rate for an OSV supporting work-class ROVs can reach hundreds of thousands of dollars, depending on the vessel's size, mission scope, and location. In addition, ROV operations require highly skilled personnel and careful handling of the long, heavy tether. Missions are further constrained by the need for favorable weather conditions due to the reliance on surface support [10].

The tether also reduces maneuverability and increases the risk of entanglement—particularly during tasks that require precise navigation in confined environments, such as pipeline inspections or shipwreck exploration. Entanglement with underwater obstacles like rocks, debris, or structures can delay operations, introduce safety hazards, and demand constant monitoring and planning. These challenges collectively reduce mission efficiency and effectiveness.

1.1.3 Challenges in video transmission underwater

To mitigate these challenges associated with a tether, the development of tetherless ROVs has been a focus in marine robotics [11], with one of the greatest challenges being real-time wireless underwater communication between the ROV and the surface platform.

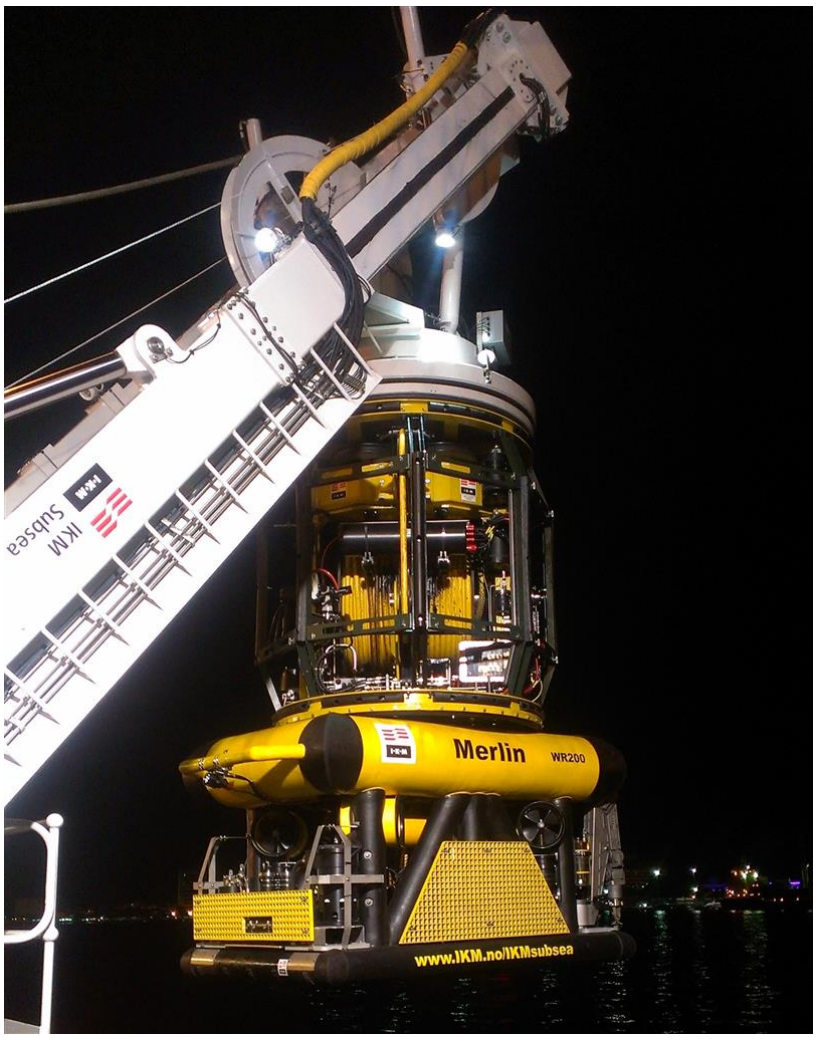


Figure 1.1: Deployment of a work-class ROV with a tether management system on top (Image: IKM Group, CC BY 3.0)

Underwater communication presents unique challenges that differ significantly from terrestrial and aerial systems. As electromagnetic waves do not propagate well in water, traditional wireless technologies like radio and microwave communication become ineffective [12]. Instead, acoustics are typically used for underwater mid-to-long communication [13, 10]. However, acoustic communication is characterized by low data rates. Current state-of-the-art acoustic links provide bit rates in the range of tens of kilobits per

second (kbps) over distances of several hundred meters to a few kilometers, depending on channel conditions [14].

While current acoustic links can handle the transmission of commands and some sensor data, video transmission in real-time or near-real-time, which is crucial for ROV operators to to effectively monitor the environment and control the vehicle, remains a significant challenge.

There are two approaches to achieve real-time image data transmission underwater: increasing the data rate or reducing the size of the data to be transmitted. Much effort has been spent to increase the data rate of underwater communications [14]. In this thesis, we focus on the latter approach, exploring whether efficient data compression can be achieved using the prior information available in inspection and intervention missions. In particular, we focus on image compression, as videos are composed of key frames and motion vectors, with key frames—essentially still images—accounting for the majority of the bandwidth usage.

1.1.4 Motivation

Underwater inspection is critical across many industries, yet tethered ROV operations limit flexibility and drive up costs. Tetherless control is promising, but the low bandwidth of underwater wireless links—especially for real-time video—remains a barrier. At the same time, many inspection missions revisit the same sites, allowing rich priors about those environments to accumulate. These observations motivate us to explore how such prior

information can be leveraged to enable real-time wireless image transmission, potentially overcoming acoustic bandwidth limitations and improving the feasibility of tetherless ROV operations.

1.2 Key Idea

Consider a scenario where you revisit a familiar place—such as a city square or a mueseum—for which a three-dimensional (3D) model already exists. Assuming the scene remains largely static, the view observed from any given pose (i.e., position and orientation) should closely match the one rendered by the model from the same pose. In theory, this means that transmitting only the camera pose may suffice to reconstruct a full image. For example, a 720p RGB image (2.8 megabytes) could be replaced by just 28 bytes of pose data—achieving a theoretical compression ratio of 100,000:1. This highlights the substantial compression potential offered by leveraging prior information in underwater inspection and intervention missions.

Of course, real-world environments are not perfectly static. Dynamic changes—such as the appearance of new objects or variations in lighting—do occur. However, the visual differences induced by these changes are often small and spatially sparse, making them highly compressible.

This idea builds on the concept proposed in CHRIIS [10], which advocates using a photorealistic digital twin of the operational environment—constructed during an initial autonomous inspection stage—to render

Figure 1.2: Rendering outcomes on a real world scene by NeRF. Reproduced from [15].

virtual camera views onshore or onboard. Only the difference between the actual and rendered images is transmitted, enabling high-quality video streaming over bandwidth-limited underwater acoustic links.

Recent advancements in novel view synthesis (NVS), such as Neural Radiance Fields (NeRF) [15], further demonstrate the feasibility and promise of this approach. When trained on images collected during a baseline mapping run, NeRF models can render photorealistic images from arbitrary poses, enabling pose-driven reconstruction of visual data in subsequent inspection and intervention runs, as shown in Fig. 1.2.

1.3 Objectives

The main objectives of this thesis are as follows:

- Investigate suitable models for incorporating prior information about the underwater environment.

- Explore methods to extract efficient, effective, and robust representations (e.g., camera poses) from underwater images.
- Develop techniques to reconstruct images using the extracted representations and the learned models.
- Validate the proposed framework in real-time underwater scenarios.

By addressing these objectives, this research aims to develop a robust prior-based image compression framework for real-time underwater image transmission, ultimately enabling tetherless operation of ROVs.

1.4 Contributions

The followings are the main contributions of this thesis:

1. We present a complete framework for scene-specific image compression that leverages trained NVS models as priors. While CHRIIS previously proposed the high-level idea of using scene priors for compression [10], our work advances this concept by developing the necessary methods, implementing a practical system, and demonstrating its effectiveness in real-world underwater settings. To the best of our knowledge, this is the first work to use trained NVS models for efficient, scene-specific image compression.
2. We evaluate the feasibility of various NVS models for modeling and for real-time photorealistic rendering of underwater structures in both

clear and turbid waters. This study identifies a robust model capable of handling transient changes, such as novel objects and lighting variations, while maintaining high rendering quality and efficiency.

3. We develop an image-based pose estimation framework that includes:

   a) A geometry-aware loss function to enhance interpretability, computational efficiency, and estimation accuracy.

   b) Augmented training data generated using 3D NVS techniques to improve model robustness and generalization across varied underwater conditions.

   c) Sensor fusion using an extended Kalman filter (EKF) to integrate additional data, such as altimeters and compasses, further enhancing localization accuracy and robustness in underwater environments.

4. We introduce a gradient descent-based pose refinement method that leverages NVS-rendered imagery and vehicle motion priors. In clear water, the optimization minimizes the photometric difference between real and rendered views. In turbid water, where photometric cues are less reliable, we propose a feature-based loss to guide refinement. Both approaches reduce the size of the difference image, thereby improving overall compression performance.

5. We validate the proposed framework in field trials conducted in both

clear and turbid waters. Our results show that the prior-based image compression framework outperforms traditional codecs such as WebP and JPEG-XL in compression ratio and image quality and enables real-time image transmission over acoustic links under real-world conditions. These results underscore the framework's practical applicability and effectiveness in bandwidth-constrained underwater environments.

1.5 Thesis organization

Chapter 2 reviews existing image compression techniques, including conventional methods, deep learning-based approaches, and recent advancements in underwater image compression. It also surveys 3D reconstruction techniques with a focus on NVS models, and examines underwater localization solutions, highlighting limitations that motivate this thesis.

Chapter 3 introduces the proposed framework, and presents a feasibility study that assesses the suitability and robustness of various NVS models for underwater scene reconstruction and real-time rendering. To address the challenge of extracting effective scene representations from underwater images for NVS rendering, Chapter 4 presents our proposed image-based estimators and evaluates them across diverse underwater environments. Building on this, Chapter 5 presents our inverse NVS method, which refines the scene representation via gradient-based optimization to minimize view

synthesis discrepancies, thereby enhancing compression performance. We evaluate this method in controlled underwater environments. Chapter 6 demonstrates the enhanced framework's practical viability through real-world field trials, incorporating additional techniques to improve robustness. Finally, Chapter 7 summarizes key findings and outlines directions for future research to address current limitations and extend the framework's applicability.

Chapter 2

Literature Review

We review existing image compression methods, including conventional approaches, deep learning-based techniques, and recent advancements in underwater image compression, to identify the research gap our work addresses. Additionally, we examine recent progress in 3D reconstruction methods, which serve as a key component for incorporating prior information into our framework. Furthermore, we analyze existing underwater localization techniques, highlighting their key limitations and the need for a new method with improved accuracy and robustness.

2.1 Image compression

Traditional image compression methods rely on hand-crafted transformations and statistical encoding techniques to efficiently represent image data. JPEG-XL, a state-of-the-art compression format, combines discrete cosine transform-based block coding with Haar wavelet transforms, enabling high compression efficiency and scalability [16]. WebP, a widely adopted format optimized for web applications, employs block-based predictive coding for

lossy compression and Huffman coding for entropy encoding, making it well-suited for fast decoding and reduced file sizes [17]. These formats balance efficiency, scalability, and practical application across various digital environments.

Recent advances in machine learning have led to the emergence of end-to-end learned image compression methods, which utilize deep neural networks to directly learn efficient, low-dimensional latent representations from large datasets [18, 19, 20, 21, 22, 23, 24, 25, 26, 27]. By leveraging content-adaptive latent spaces, these methods achieve rate-distortion performance comparable to or surpassing classical codecs such as JPEG 2000 and WebP [18, 25, 26, 27]. Among learned compression methods, Ballé et al. [18] and Minnen et al. [25] introduce hierarchical entropy modeling, employing Gaussian scale hyperpriors that significantly improve the compression efficiency by capturing spatial dependencies within latent representations. Subsequent methods, such as Cheng et al.[26] and MLIC[27], further improve compression performance through even more sophisticated context modeling, but this enhancement comes at the expense of substantially higher computational complexity and slower processing speeds. Most recently, MLIC++ proposes a linear-complexity multi-reference entropy model, yielding near-linear computational cost [28]. While MLIC++ reports state-of-the-art results on standard benchmarks, its performance at low resolutions and in underwater imagery has not, to our knowledge, been evaluated.

Given the stringent limitation of data rate in underwater communications, researchers have also explored image compression specific to underwater settings, aiming at higher compression ratio. Early works in this area explore using more efficient transforms to enhance compression performance. Li and Wang propose a Wavelet Tree-based Wavelet Difference Reduction method that removes visual redundancy while preserving spatial information [29]. This approach enhances compression efficiency, especially at low bitrates. Similarly, Zhang et al. utilize adaptive hybrid wavelets and directional filter banks to achieve high compression efficiency while maintaining image quality for low-bandwidth underwater communication [30].

Recent works exploit unique characteristics of underwater imagery with learned compression methods. In the water, light scattering and absorption lead to color loss, contrast degradation, and hazy appearances. These effects lead to less color information and lower-detail backgrounds, making compression strategies that prioritize key regions more effective. Semantic compression techniques exploit this by reducing the bitrate required for background areas while preserving essential details [31]. Deep learning approaches, such as those incorporating physical priors and autoencoder-based techniques, further optimize compression by reducing redundancy and improving perceptual quality [32, 33, 34, 35, 36, 37].

Although existing works have explored the use of priors in image compression, these priors are typically generic across all underwater images rather than tailored to specific scenes. While such generalizations may cap-

ture broad environmental characteristics, the significant variability across underwater scenes often limits their effectiveness. Additionally, these methods generally lack scene understanding—they treat images independently and do not model relationships between them. At the same time, in underwater inspection missions, scene-specific information—such as maps or prior visual data—is often readily available. Motivated by this, our work investigates how leveraging such priors and incorporating scene understanding can improve image compression.

2.2 3D reconstruction

3D reconstruction is a widely used approach to reuse prior scene information and construct a detailed 3D representation of an environment [38]. 3D reconstruction techniques have evolved significantly over the years, beginning with traditional methods that rely on geometric principles and moving toward machine learning-based approaches that enable highly realistic scene reconstruction. Traditional 3D reconstruction methods include Multi-View Stereo (MVS) [39] and Structure from Motion (SfM) [40]. These approaches reconstruct 3D models from multiple two dimensional images by estimating depth and structure from correspondences across different viewpoints. While these methods have been widely used in terrestrial environments, they often struggle with occlusions, texture-less surfaces, and computational inefficiencies, limiting their applicability in complex scenes, such as underwater environments.

NVS emerges as an alternative to traditional 3D reconstruction techniques. NVS is the process of generating images of a scene from viewpoints that were not originally captured. Given a set of input images, NVS predicts the appearance of the scene from new perspectives by inferring the underlying geometry, textures, and lighting conditions. Unlike traditional 3D reconstruction methods, which explicitly build geometric models using SfM and MVS, NVS focuses on synthesizing realistic novel views without necessarily recovering a full 3D structure.

Earlier NVS models primarily leverage deep learning techniques such as conditional GANs and autoencoders to generate new perspectives from limited image inputs [41, 42, 43]. Such approaches typically use an encoder-decoder framework, where the encoder maps input images to a latent space representation, and the decoder, which often employs a conditional GAN-based architecture, reconstructs images from this latent representation while incorporating a desired viewpoint transformation. While GANs and autoencoders enable fast, learned image-based synthesis, they lack 3D fidelity and struggle with multi-view consistency, often introducing artifacts or distortions in the synthesized views.

Recent advances in radiance field-based methods have addressed many of the limitations of earlier approaches. NeRF, for example, represents scenes as continuous volumetric functions and use neural networks to learn the radiance and density at any point in space, enabling highly detailed and photo-realistic view synthesis from a sparse set of input images [15]. Building

on this, 3D Gaussian Splatting (3D-GS) has emerged as a compelling alternative, modeling scenes as a set of Gaussian primitives in 3D space [44]. This probabilistic representation supports efficient rendering and achieves better real-time performance than NeRF, while still maintaining high visual fidelity in novel view synthesis.

The ease of training and real-time, high-quality rendering capabilities of techniques like NeRF and 3D-GS make them especially well-suited for encoding prior knowledge captured during baseline mapping runs in underwater inspection missions.

2.2.1 NeRF

NeRF represents 3D scenes by encoding them into a continuous function using a deep neural network. At the core of NeRF is a fully connected multilayer perceptron (MLP) that maps 3D spatial coordinates and viewing directions to color and volume density values. The model is trained using a sparse set of 2D images with accurate poses, where the goal is to learn a function that can synthesize novel views of the scene [15].

NeRF models a scene as a volumetric field, which is sampled along rays cast from a camera. Each ray is divided into discrete points, and the neural network predicts the RGB color and density at each point. The final pixel color is obtained by integrating these values using a volume rendering equation, where the accumulated density determines the contribution of each sample along the ray. This rendering process allows NeRF to generate

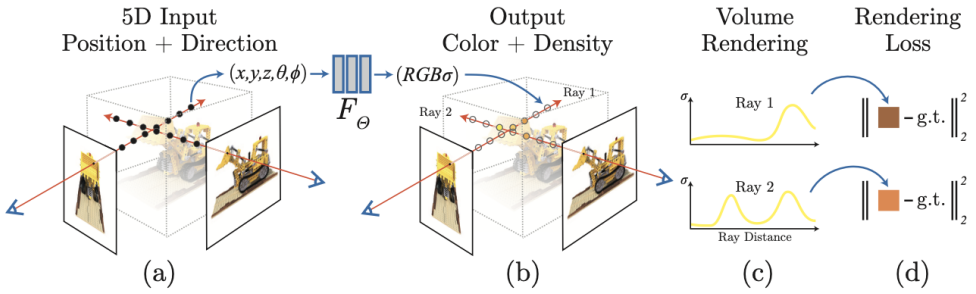


Figure 2.1: An overview of neural radiance field scene representation and differentiable rendering procedure. Reproduced from [15].

highly detailed and view-dependent effects such as specular reflections.

A key challenge in training NeRF is aliasing caused by high-frequency details. To mitigate this, the original NeRF implementation applies positional encoding, where input coordinates are mapped to a higher-dimensional space using a set of sinusoidal functions. This enables the MLP to capture fine details and high-frequency variations, improving reconstruction quality. An overview of NeRF scene representation and differentiable rendering procedure is shown in Fig. 2.1.

Despite its high-quality results, NeRF is computationally expensive. The original model requires querying the MLP thousands of times per ray, making rendering slow. Various optimization strategies have been introduced to address this, such as hierarchical sampling, which allocates more samples to regions of higher importance, and efficient caching techniques like InstantNGP [45], which replaces the MLP with a multi-resolution hash grid representation. These improvements significantly reduce training and rendering times while maintaining visual fidelity.

Extensions of NeRF have tackled its limitations, including generalization

to unseen scenes, adaptation to dynamic environments, and incorporation of real-world priors to handle complex lighting conditions [46, 47, 48]. However, challenges remain in scalability and efficiency, particularly in rendering large-scale or interactive applications.

2.2.2 3D-GS

3D-GS is a novel approach for real-time 3D reconstruction and rendering that has gained significant attention as an alternative to NeRF [44]. Unlike NeRF, which uses an implicit neural network to represent a scene, 3D-GS employs an explicit representation based on 3D point-based primitives with Gaussian attributes.

The core idea of 3D-GS is to model a 3D scene as a collection of anisotropic Gaussians distributed in space. Each Gaussian is defined by its center position, covariance matrix (which determines its shape and orientation), opacity, and color. Rendering is achieved through a differentiable splatting process: the Gaussians are projected onto the image plane and blended using alpha compositing. This eliminates the need for costly volumetric integration required by NeRF and lays the foundation for efficient real-time rendering.

Optimization in 3D-GS typically follows a two-stage process, as illustrated in Fig. 2.2. First, an initial point cloud is generated using a SfM pipeline or depth estimation methods. Then, the Gaussian parameters—positions, densities, colors, and anisotropies—are refined using

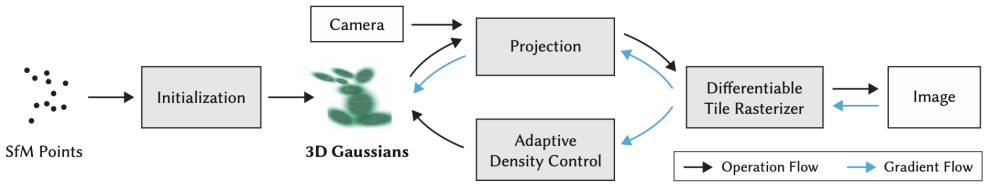


Figure 2.2: Block diagram illustration of the 3D-GS approach. Reproduced from [44]

gradient-based optimization through differentiable rendering. This process improves visual fidelity and enables high-quality view synthesis, even from sparse input views.

One of the key advantages of 3D-GS is its computational efficiency. Unlike NeRF, which requires querying an MLP thousands of times per ray, 3D-GS relies on an explicit representation that can be directly optimized and rasterized. This leads to significantly lower computational costs and enables real-time applications. Kerbl et al. demonstrate interactive frame rates while maintaining photorealistic quality, marking a major breakthrough in neural rendering [44].

Despite its advantages, 3D-GS faces certain limitations. Since it relies on explicit primitives, memory consumption can become a concern when modeling large-scale scenes, as the number of Gaussians increases with scene complexity. Additionally, handling dynamic environments and occlusions remains an open research challenge, as the current approach assumes static scenes.

2.3 Pose estimation

In order to render corresponding views from NVS models, we need accurate camera poses. In underwater environments, the use of global positioning systems is hindered due to the rapid dissipation of electromagnetic waves in water [12]. Traditionally, underwater localization has relied on inertial navigation systems (INS), Doppler velocity loggers (DVL) and acoustic positioning systems. However, these methods face significant challenges in the context of inspection missions. Acoustic navigation is often compromised by shadowing effects and multipath interference near marine structures, which can severely distort signal paths and reduce accuracy. Consequently, achieving precise acoustic navigation requires complex and costly setups [49]. Furthermore, INS and DVL, despite their widespread use, suffer from an accumulation of errors over time [49]. This limits their ability to provide the positioning accuracy required for detailed inspection of underwater structures. Although high-grade INS and DVL may be able to provide sufficient accuracy, they, too, come with high costs.

In recent years, optical sensors, particularly cameras, have become increasingly popular for underwater localization [50]. Cameras are lightweight, low-cost, and often already integrated into inspection vehicles, making them an attractive choice [51]. While some vision-based methods require diver-assisted setups [52, 53], which add operational complexity and cost, visual localization methods in general can estimate vehicle poses directly

from images of the surrounding environment, enabling localization without such additional infrastructure.

Techniques such as simultaneous localization and mapping (SLAM) [49] and visual odometry (VO) [54, 55, 56, 57, 58] have been applied in both terrestrial and underwater settings. These methods estimate camera motion by tracking visual features across frames without requiring prior data. However, SLAM and VO face key limitations in underwater scenarios. Their reliance on continuous feature tracking makes them sensitive to visual degradation from turbidity, lighting variability, and repetitive textures—conditions commonly found in underwater environments. In addition, SLAM systems must build and maintain a consistent map throughout the mission, which increases computational load and can reduce reliability in dynamic or cluttered scenes [59].

In contrast, visual relocalization leverages prior maps or reference images of the scene to directly estimate the camera pose. This approach avoids the need for real-time mapping or drift-sensitive tracking, enabling more robust and efficient localization using only monocular images. Visual relocalization techniques are categorized into feature-based methods such as Active Search [60], and deep-learning methods like PoseNet [61]. Active search achieves image-based localization by systematically identifying and matching 2D features in query images with 3D points in a scene model. Using a 3D model reconstructed from SfM, one can estimate poses by firstly creating 2D-to-3D matches between image features and 3D points in

SfM and then using a n-point solver for pose estimation inside a random sample consensus (RANSAC) loop [62]. Using a visual-vocabulary-based quantization of descriptor space as the prioritization scheme, Active Search speeds up the 2D-to-3D matching process. It then uses 3D-to-2D matching to improve the localization accuracy, achieve the state-of-the-art results [60]. In contrast, PoseNet is a deep learning model that utilizes a pretrained convolutional neural network (CNN) to estimate the 6-degree-of-freedom (6-DOF) poses of a camera directly from images. This approach simplifies the camera relocalization problem by bypassing the traditional feature extraction and matching steps, instead relying on the CNN to learn and estimate the camera's position and orientation within a previously mapped environment directly from the image data. Using transfer learning, PoseNet leverages models pretrained on ImageNet and reduces the dependency on large datasets [61].

While Active Search has demonstrated state-of-the-art results in structured terrestrial environments, its reliance on salient features [61] and high computational cost [63] limits its suitability for underwater scenes, which are often characterized by sparse features and obscured textures due to poor visibility. Learning-based regressors inspired by PoseNet offer an appealing alternative for these environments by enabling fast and robust estimation.

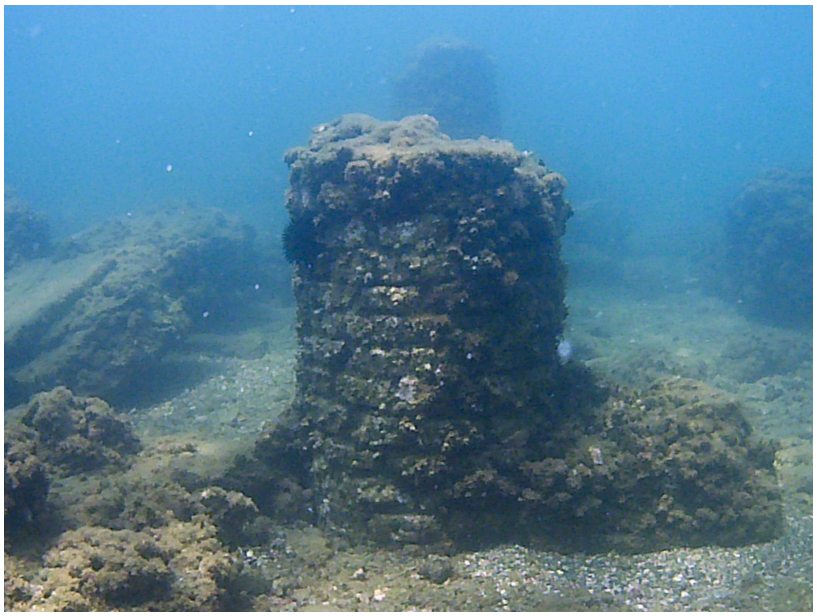
Numerous methods have since been developed by building upon the PoseNet framework. For example, DSAC [64] combines learning-based regression with differentiable RANSAC for robust and accurate localization,

though at significantly higher computational cost. VidLoc [65] extends PoseNet by modeling temporal dependencies across video sequences using recurrent networks. EffLoc [66], in contrast, retains a single-image input at inference but uses a transformer-based encoder to capture long-range spatial dependencies. MapNet [67] introduces additional geometric constraints during training by incorporating sensor-derived relative poses to improve consistency and robustness. While these methods achieve impressive results in terrestrial applications, they all share PoseNet as a foundational architecture.

Previous research has demonstrated PoseNet's efficacy in conducting inspection tasks within tanks with toy structures [68]. However, in real marine environments, underwater images often have much lower visibility and color information compared to terrestrial images and images from clear water tanks, as shown in Fig. 2.3. Moreover, real underwater missions have larger scale and more complicated structures, presenting a more complex problem than the toy setup in [68]. As such, the performance of learning-based pose estimators with realistic structures and in at-sea environments needs to be further investigated.

With the advent of NVS models, recent work has attempted to use NVS models for pose and refinement [69, 70, 71, 72, 73]. Chen et al. propose Direct-PoseNet [74] and DFNet [75] which incorporate NVS as part of the training pipeline of the pose estimator, improving its accuracy and robustness. iNeRF estimates the camera pose by inverting a single

(a) Example image from the Kings College dataset [61].



(b) Example image from an underwater archaeology site (Image: Ruthven, CC0, via Wikimedia Commons).

Figure 2.3: Example images from terrestrial and underwater scenes. As compared with typical outdoor terrestrial scenes (a), underwater scenes in marine environments (b) have lower visibility and less color information.

image using a pretrained NeRF model [69]. It does so by minimizing the pixel-wise difference between the rendered image and the input camera image via gradient descent. iComMa proposes a similar approach using 3D-GS models, minimizing both pixel-wise differences in the image, and

matching loss, which is the mean of Euclidean distances between keypoints in corresponding images [70]. This hybrid loss is proposed as an alternative for enhanced robustness towards poor initialization. Both iNeRF and iComMa use the Adam optimizer [76] for pose optimization. While iComMa has an improved speed over iNeRF due to faster convergence and high rendering speed of 3D-GS models, it still requires about a second for one optimization on a single NVIDIA RTX A6000 GPU, which is prohibitively too slow for the video transmission speed required for tetherless control.

Despite recent advancements in visual localization and the integration of NVS models for camera pose estimation, existing methods have not been tested in real-world underwater environments. Techniques like iNeRF and iComMa, while demonstrating promising results in controlled settings, are computationally expensive and impractical for real-time pose estimation on underwater vehicles. Moreover, their performance in the presence of underwater-specific challenges—such as lighting variations, turbidity, and dynamic marine conditions—remains unexplored.

This thesis addresses a key research gap by investigating the performance and feasibility of pose estimation methods in real underwater environments. In Chapter 4, we introduce an image-based relocalization framework designed specifically for underwater re-inspection missions. Our approach integrates geometry-aware loss functions, NVS-based data augmentation, and sensor fusion, and is evaluated under both controlled and turbid water conditions.

2.4 Summary

Recent literature on image compression techniques highlights the benefits of incorporating prior information. However, the use of scene-specific priors remains largely unexplored. This research gap motivates our investigation into how such priors can be leveraged to improve image compression performance. Our proposed approach to addressing this gap is presented in Chapter 3.

Advancements in NVS models, such as NeRF and 3D-GS, offer valuable insights into the integration of scene-specific knowledge. Their ability to produce photorealistic renderings in real time makes them particularly promising for our image compression framework. Nevertheless, their application to underwater environments has not been thoroughly investigated—an area we explore in Chapter 3.

To effectively utilize NVS models for generating priors, accurate camera pose estimation is essential. Advances in visual-based relocalization methods provide new opportunities for underwater localization. Our contributions in this space are detailed in Chapter 4, Chapter 5, and Chapter 6, where we present techniques for robust pose estimation and refinement in both controlled and real underwater conditions.

Chapter 3

NVS prior-based image compression

The routine nature of underwater inspection and intervention missions provides valuable scene-specific prior information, which can be leveraged to optimize image compression. To understand the benefit of such prior knowledge more formally, we turn to an information-theoretic perspective. Without prior information,the image I visible to the ROV at an underwater site can be modeled as a sample from a distribution $p(I)$. The compressed size achievable with this image would be bounded by its entropy $H(I) = -\mathbb{E}[\log(p(I))]$, where $\mathbb{E}$ denotes the expectation operator. When prior knowledge of the scene is available, it can be encoded as a latent variable z, allowing us to model the image conditionally as $p(I \mid z)$, which yields a lower entropy $H(I \mid z) \leq H(I)$. In general, the more informative the prior, the more predictable the image becomes, and the tighter the lower bound for compression. Thus, exploiting scene-specific prior information has the potential to yield an efficient image compression approach that can enable image transmission in real time via underwater acoustic links.

As reviewed in Chapter 2, previous works in underwater image compression leverage characteristics of underwater images to improve compression performance. However, there have been no attempts to leverage scene-specific prior information for underwater image compression. In this chapter, we propose an NVS prior-based image compression framework, termed *NVSPrior*, which exploits prior scene knowledge to compress images captured by an ROV during underwater inspection missions. We describe its mechanism, its use of prior information, and the motivation for its development. Subsequently, we investigate the feasibility of NVS models for 3D reconstruction of underwater structures in both controlled environments and real seawater. The work presented in this chapter was published in [77], [78], [79] and [80].

3.1 Overview of NVSPrior

Image transmission using classic image compression techniques works as shown in Fig. 3.1. On the ROV side, the camera image is compressed by a classic image compressor. The compressed image is then transmitted to the operator side via communication links. On the operator side, the camera image is reconstructed by decompressing the compressed image.

NVSPrior reduces the size of data to be transmitted between the ROV and the operator by using a shared NVS model trained with prior knowledge of the scene. During the baseline mapping run of the underwater mission, we collect camera images to characterize the scene. Using the collected

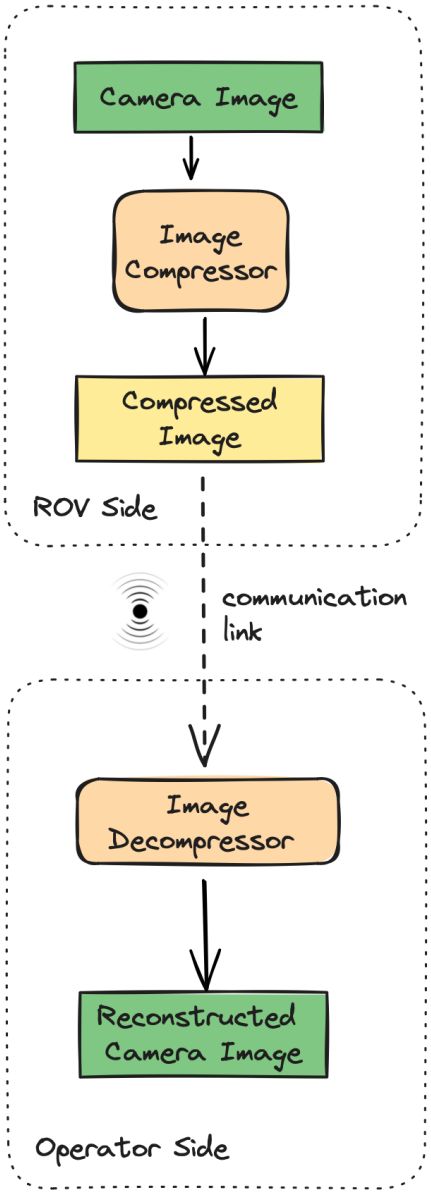


Figure 3.1: Pipeline for image transmission in underwater inspection missions using classical image compression codecs.

information, we can train a NVS model as reviewed in Chapter 2. We then store identical copies of the trained NVS models on both the ROV and the operator's side.

During the following inspection and intervention runs, we first obtain the optimal latent representation of the scene which includes the camera

pose and potentially other elements such as transient embeddings for the NVS model to render an image closest to the camera image. To capture the changes in the environment that the NVS model would not be able to render, such as novel objects that were not present during the mapping run, we compute the "difference image" between the NVS rendered image and the camera image, I_{diff}. We compress I_{diff} using classic compression techniques. We transmit the (1) the optimal latent representation, and (2) the compressed difference image I_{diff} from the ROV to the operator side via communication links. On the operator side, we render the image using the optimal latent representation and decompress I_{diff}. Adding the decompressed I_{diff} onto the rendered image, we reconstruct the estimate of the camera image. An overview of our approach is shown in Fig. 3.2.

The effectiveness of this approach is based on the assumption that the environment remains largely static between visits. This is often true in reality because underwater re-inspection missions occur regularly. In such cases, the NVS-synthesized view during subsequent inspections will closely match the actual camera image—provided the pose is accurate—resulting in a sparse difference image, I_{diff}, which typically is highly compressible. As such, we only need to transmit the camera pose, typically just a few bytes, and a small I_{diff}. This leads to a significant reduction in data transmission compared to traditional compression methods. In exceptional cases where the difference is substantial—i.e., the compressed I_{diff} exceeds the size of the compressed camera image—we instead transmit the compressed camera

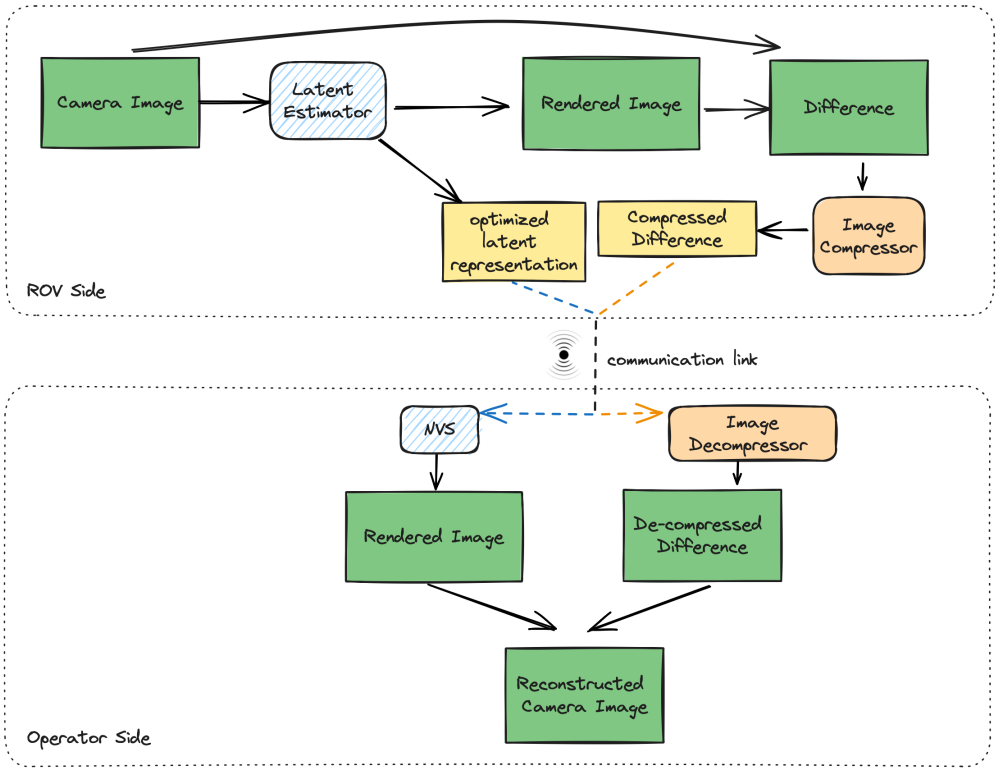


Figure 3.2: Pipeline for image transmission using NVSPrior.

image directly.

As the most integral component of our proposed approach is the NVS, we next conduct a feasibility study on NVS of underwater structures, presented in the next section.

3.2 NVS of underwater structures via NeRF

As reviewed in Chapter 2, NeRF models offer a computationally efficient approach for synthesizing novel views of scenes, and have demonstrated strong performance in photorealistic rendering of static environments. However, their effectiveness degrades in dynamic settings. NeRF models are trained to minimize reconstruction error in the RGB color space, assum-

ing photometric consistency—i.e., that two images taken from the same viewpoint should be nearly identical except for noise [46]. In real-world scenarios, particularly underwater, this assumption often breaks down due to dynamic elements such as sediment, algae, fish, and varying illumination conditions. These factors introduce inconsistencies across input images, leading to rendering inaccuracies and visual artifacts.

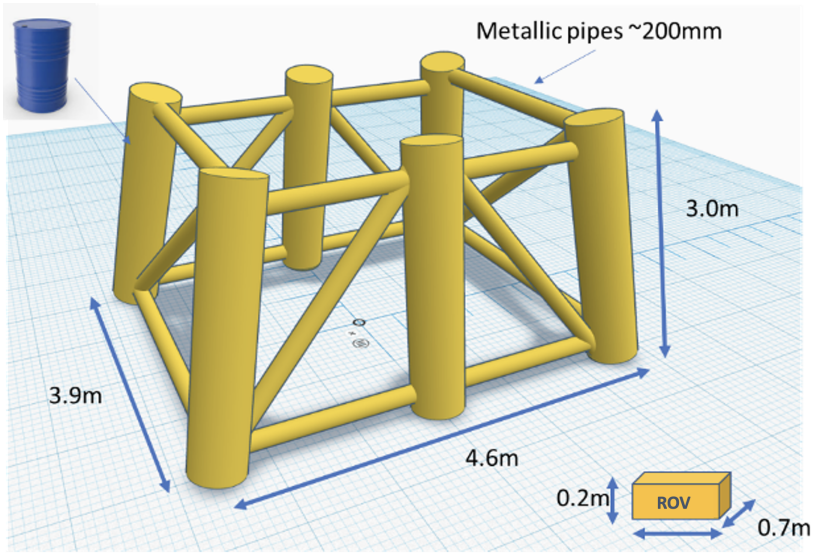
In this section, we investigate the feasibility of NVS for underwater structures using NeRF. We evaluate the performance of (i) a vanilla Nerfacto implementation, (ii) a modified Nerfacto implementation with transient embedding and (iii) a Nerfacto trained with a robust loss function.

3.2.1 Datasets

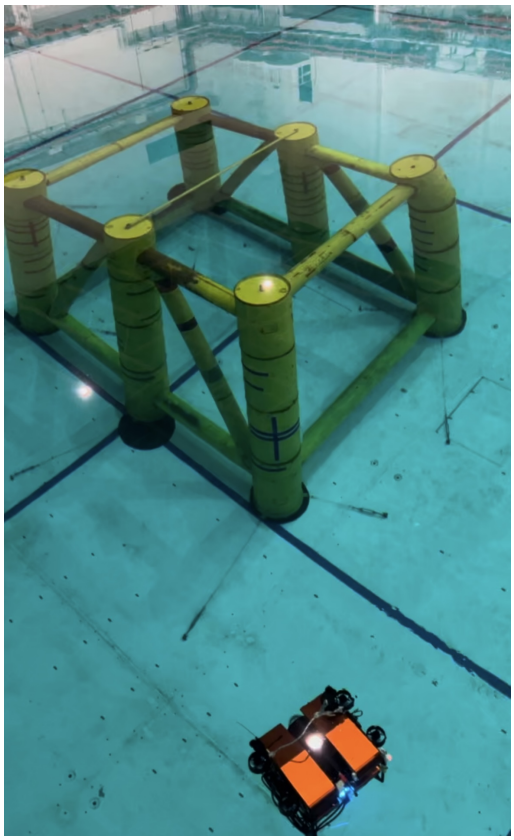
3.2.1.1 Controlled underwater environment dataset

To collect data for the study, we conducted an experiment in a large state-of-the-art Deepwater Ocean Basin (DOB) at the Technology Centre for Ocean and Marine, Singapore (TCOMS) [81].

The DOB at TCOMS is an indoor pool measuring 60 m $\times$ 48 m $\times$ 12 m. As illustrated in Fig. 3.3, a structure was placed in the basin, which consisted of six piles interconnected by metallic pipes, with each pile comprising three metallic oil barrels. The overall dimensions of the structure were approximately 3.9 m $\times$ 4.6 m $\times$ 3.0 m. The whole structure was yellow in color. To better differentiate the barrels, duct tape strips of various colors with different patterns were stuck on each barrel to create



(a) Schematic of the structure.



(b) Topview of ROV surveying the structure.

Figure 3.3: The structure surveyed in the TCOMS facility.

uniquely identifiable features.

Figure 3.4: Customized hybrid ROV: Hydra.

For the purpose of testing our approach, we customized a hybrid ROV named Hydra (shown in Fig. 3.4), equipped with a wide-angle, low-light monocular camera for collecting RGB image data, a compass for collecting orientation information, and an altimeter for collecting altitude information.

The vehicle was deployed within the DOB, and navigated around the structures, with the camera directed towards the structure. It captured a series of images during the run emulating a real-world inspection mission. During the mission, there were occasional transient objects picked up at the scene apart from the structure. For example, there were instances where the tether connecting the vehicle to the top-side controller was incidentally captured on the camera (see Fig. 3.7, row 3, column 1), which is a challenge that may occur in a regular underwater inspection mission. Additionally, to study the robustness of NeRF under variable illumination—common in

real-world scenarios—we varied the lighting in the DOB during some runs and sporadically toggled the vehicle's onboard lights.

3.2.1.2 Open marine shipwreck dataset

To evaluate model performance in real-world underwater conditions, we used the Torpedo Boat Wreck (TBW) dataset [82, 83], a publicly available collection of images captured during a hybrid ROV survey of a shipwreck site in the Mediterranean Sea. The dataset features challenging conditions typical of open marine environments, including murky water, low visibility, and the presence of marine snow. These characteristics make it well-suited for testing the robustness of NeRF models in non-controlled underwater settings.

3.2.2 Methods

3.2.2.1 Original Nerfacto

Let $\mathbf{p} = [x, y, z]$ be a 3D point, $\mathbf{d} = [d_x, d_y, d_z]$ be a unit-norm camera viewing direction, $\mathbf{c} = [r, g, b]$ be color in red green and blue, and σ be a density. NeRF leverages MLPs to map ($\mathbf{p}$, $\mathbf{d}$) to ($\mathbf{c}$, σ). By aggregating colors and densities along a camera ray, denoted by $\mathbf{r}$, through a pixel on the camera plane, the model predicts the color of the pixel, represented by $\hat{\mathbf{C}}(\mathbf{r})$. The model is typically trained by minimizing an L2 reconstruction loss [15]:

$$\mathcal{L} = \|\hat{\mathbf{C}}(\mathbf{r}) - \mathbf{C}(\mathbf{r})\|_2^2 \tag{3.1}$$

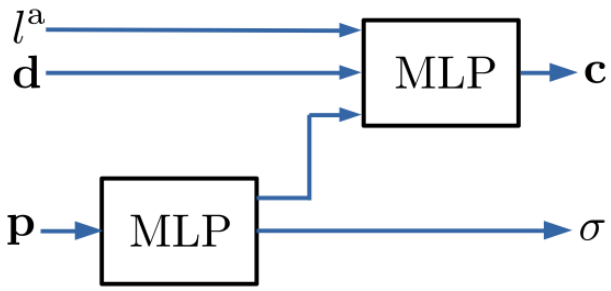


Figure 3.5: Model structure of Nerfacto.

where $\mathbf{C}(\mathbf{r})$ is the observed pixel color of ray $\mathbf{r}$ from an input image. As a baseline model for our comparative study, we employ Nerfacto, which is a simplistic modular NeRF implementation that adopts recent advancements to improve computational efficiency and handle unbounded scenes [84]. Figure 3.5 illustrates the Nerfacto model structure. Nerfacto integrates per-image appearance embeddings to effectively address the impact of diverse lighting conditions. Each appearance embedding, which is denoted by l^{a}, is a trainable real-valued vector of length n^{a}.

3.2.2.2 Nerfacto with transient embedding

Inspired by NeRF variant known as NeRF in the Wild (NeRF-W) [48], we modify the model structure of Nerfacto as shown in Fig. 3.6. Each transient embedding, which is denoted by l^{t} (where $^{\mathrm{t}}$ denotes "transient"), is a trainable real-valued vector of length n^{t}. The transient head emits a field of uncertainty, denoted as β, enabling the model to adaptively adjust its reconstruction loss function by ignoring pixels and 3D points that are likely to involve dynamic changes. The color of the pixel is calculated by aggregating not only the static components ($\mathbf{c}$, σ) but also the

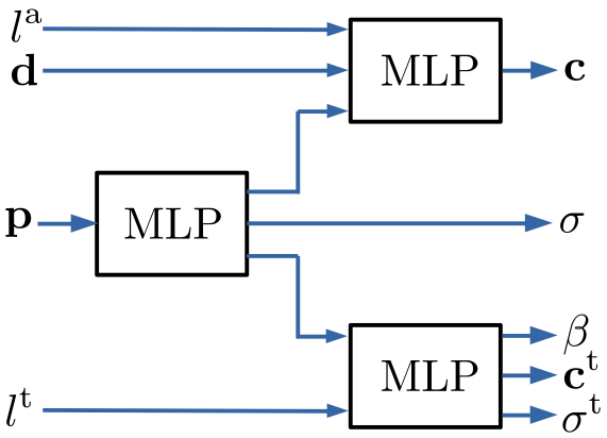


Figure 3.6: Model structure of Nerfacto with transient embeddings.

transient components ($\mathbf{c}^{\mathrm{t}}$, σ^{t}). The loss function of Nerfacto with transient embeddings is written as

$$\mathcal{L}^{\mathrm{te}} = \frac{\|\hat{\mathbf{C}}(\mathbf{r}) - \mathbf{C}(\mathbf{r})\|_2^2}{2\beta(\mathbf{r})^2} + \log(\beta(\mathbf{r})) + \lambda_u g(\mathbf{r}) \tag{3.2}$$

where $\beta(\mathbf{r})$ is obtained by accumulating β and σ^{t} along $\mathbf{r}$, $g(\mathbf{r})$ represents the average of σ^{t} along r, weighted by a non-negative scalar denoted as λ_u. The second term in Equation 3.2 balances the reconstruction loss and the third term with a multiplier λ_u enforces sparsity on the transient density.

3.2.2.3 Nerfacto with robust loss

Inspired by the NeRF variant algorithm called RobustNeRF [46], we replace Nerfacto's original loss function with a robust loss function:

$$\mathcal{L}^{\mathrm{rl}} = w(\mathbf{r})\|\hat{\mathbf{C}}(\mathbf{r}) - \mathbf{C}(\mathbf{r})\|_2^2 \tag{3.3}$$

where $w(\mathbf{r})$ is a binary weight function of $\mathbf{r}$ with intuitive parameters that adapt naturally through model fitting [46]. The robust loss function, characterized as a squared sum of trimmed entries, automatically distinguishes

inconsistent image regions and treats them as optimization outliers during the training process. Specifically, the weight function $w(\mathbf{r})$ dynamically adjusts the classification of inliers and outliers during model training, facilitating the rapid learning of fine-grained image details that are not considered outliers. It categorizes pixels along the rays as either inliers or outliers, guided by an inductive bias towards the smoothness of the outlier process. It is essential to note that this assumption may not hold true for small transient objects, such as marine snow in underwater scenes.

3.2.3 Experiments and results

To evaluate the performance of Nerfacto and its variants, we conducted training and testing on NVS tasks, focusing on the underwater structure during the inspection mission. The models were either built upon or adapted from nerfstudio version 1.0.0 [84]. Nerfstudio is an open-source library that provides a modular and user-friendly framework for training, and evaluating NVS-based 3D scene representations [84]. We used the hyperparameters $n^{\mathrm{a}} = 32$, $n^{\mathrm{t}} = 32$, $\lambda_u = 0.1$ to train the models. The full set of DOB data, consisting of 899 images, underwent a division into training and test sets using a $90-10$ random split. The estimation of camera poses for the images was accomplished using COLMAP, an SfM package [40, 85]. Number of rays per batch used for training iterations was 16384. The training set was used to train the NeRF while the test set was used for performance evaluation.

Fig. 3.7 illustrates the novel view synthesis performance of the mod-

els on the test dataset collected at the DOB. The first column contains ground-truth images captured by the camera, while the subsequent columns display images synthesized by different NeRF models from the same camera viewpoints. We evaluated the models across four scenarios:

- vehicle light on,
- vehicle light off,
- vehicle light off with the tether present,
- vehicle light on with the tether present,

presented respectively in rows 1-4 in the figure. All the models demonstrate robust performance under varying lighting conditions. In the third and last scenario, Nerfacto exhibits artifacts in the image when the vehicle tether is present. These artifacts likely stem from the presence of the tether, as it was incidentally captured in some training images taken by the vehicle's camera in close proximity to the pose corresponding to the capture of the third and last scenario's image. In contrast, the models modified with transient embeddings or trained with a robust loss function outperform Nerfacto by consistently generating photorealistic images without the artifact for all four test scenarios. For quantitative analysis, we compared the camera images with the synthesized images based on peak signal-to-noise ratio (PSNR), multi-scale structural similarity index measure (MS-SSIM) [86], and learned perceptual image patch similarity (LPIPS) [87] as summarized in Table 3.1.

Table 3.1: Quantitative results on the DOB test dataset. The direction of the arrow indicates performance preference: ↑ means higher is better.

	PSNR ↑	MS-SSIM ↑	LPIPS ↓
Nerfacto	29.382	0.915	0.468
Nerfacto + transient embeddings	30.074	0.934	0.446
Nerfacto + robust loss	30.679	0.929	0.458

Given that camera images may include variable lighting and transient objects that might not be present in the synthesized counterparts, we took several preprocessing steps. Firstly, we standardized each synthesized image based on the mean and standard deviation of the corresponding camera image. Subsequently, to remove any transient objects present in the camera image, we applied a manually labeled binary mask on both images. This approach mitigates the potential differences in lighting and transient elements between the two sets of images. The resulting modification to the Nerfacto showcases a slight enhanced image quality compared to the original Nerfacto. The improvement is marginal as inconsistencies in the scene represent only a small fraction of the overall training data.

For further validation in real-world conditions, we evaluated the models on the TBW dataset [82, 83]. The TBW data consists of 442 images, with 90% allocated for training and 10% for testing using a random split. The performance gap between vanilla Nerfacto and the models with improved features becomes even more apparent with this dataset, as shown in Fig. 3.8. Nerfacto with transient embeddings yields sharper results compared to Nerfacto trained with the robust loss. However, the former method tends

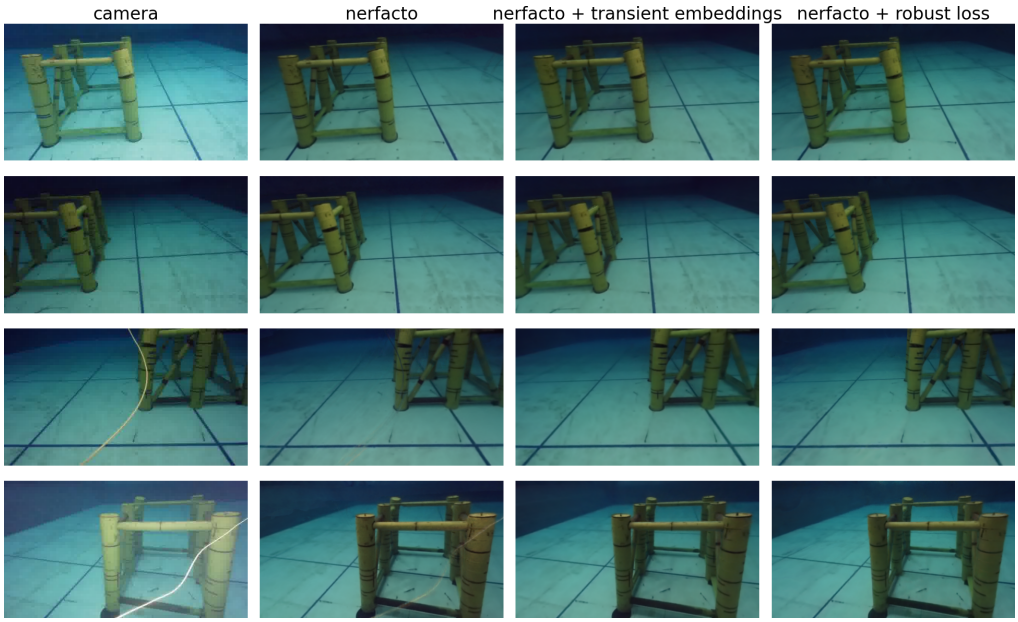


Figure 3.7: Evaluation on the test set of the DOB scene in terms of synthesized novel views from different camera viewpoints, compared to camera ground truth (column 1). The rows indicate the four scenarios, and columns 2-4 show synthesized views from the 3 NeRF approaches. This figure is reproduced from [79].

Table 3.2: Quantitative results on the TBW test dataset. The direction of the arrow indicates performance preference: ↑ means higher is better.

	PSNR ↑	MS-SSIM ↑	LPIPS ↓
Nerfacto	26.269	0.863	0.507
Nerfacto + transient embeddings	30.068	0.930	0.434
Nerfacto + robust loss	29.926	0.896	0.447

to produce images with a darker tone which are not true to the scene. The quantitative results based on the TBW test dataset are shown in Table 3.2. Based on both the qualitative results in Fig. 3.8 and in Table 3.2, both enhanced Nerfacto models demonstrate a superior ability to handle the presence of marine snow in the training images, resulting in the generation of synthesized images with significantly higher quality compared to the original Nerfacto model.

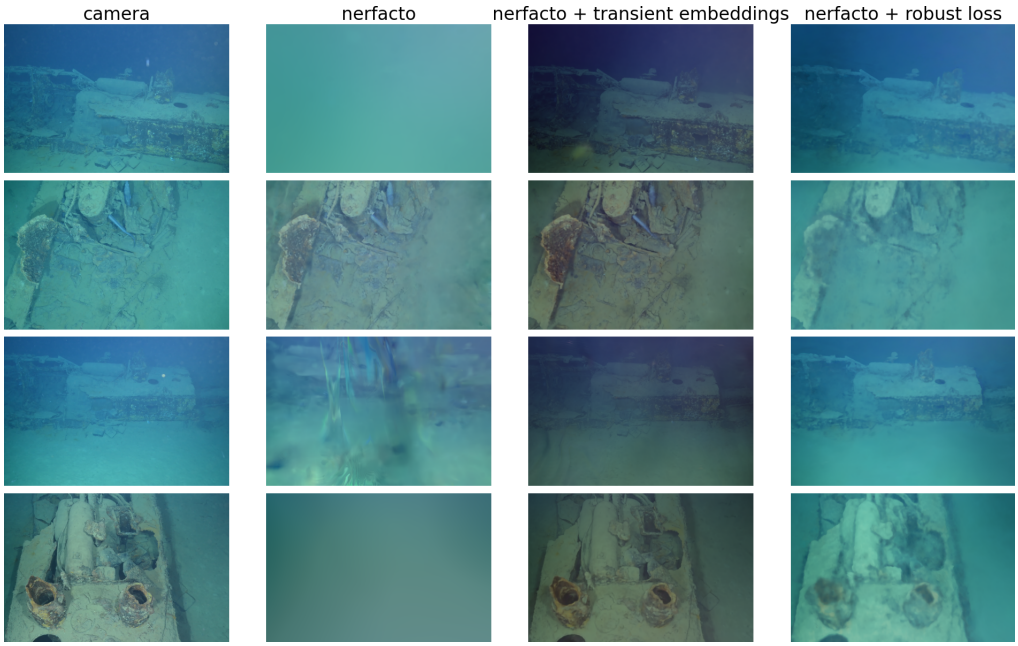


Figure 3.8: Evaluation on the test set of the TBW scene. Each row corresponds to a different camera viewpoint. This figure is reproduced from [79].

3.3 NVS of underwater structures via 3D-GS

In this section, we explore the use of 3D-GS for NVS of underwater structures. Unlike NeRF, which relies on neural representations and ray-marching, 3D-GS employs explicit point-based representations, leveraging Gaussian splats to represent scene geometry and appearance efficiently [44]. This approach allows for real-time rendering and provides a practical solution for our NVS prior-based image compression approach.

3.3.1 Method

We adopt Splatfacto, an optimized 3D-GS implementation provided by nerfstudio [84], to train and render novel views of underwater structures. The

method represents a scene as a collection of anisotropic 3D Gaussians, each defined by a position, anisotropic covariance, opacity (α), and color. The rendering process involves projecting these 3D Gaussians onto a 2D image plane and aggregating their contributions using a differentiable tile-based rasterizer for α-blending [44]. The training procedure optimizes position, anisotropic covariance, opacity and color to minimize the reconstruction error between the rendered and observed images [44].

The key advantage of 3D-GS is its ability to generate high-quality novel views with significantly faster training and inference times compared to NeRF-based approaches, primarily due to efficient GPU rasterization. For instance, rendering an image with Nerfacto typically takes 1–2 seconds, while Splatfacto renders the same image in under 0.03 seconds. To optimize visual quality for underwater scenes, we adjusted two key hyperparameters: the alpha cull threshold and the scale regularizer proposed by PhysGaussian [88]. The alpha cull threshold defines the opacity cutoff for discarding gaussians—lowering it retains more semi-transparent gaussians, which improves completeness in sparse regions at the cost of slower training. We also enabled scale regularization to encourage gaussians to adopt more balanced, compact shapes. While these changes slightly increase training time, they have no impact on rendering speed, which remains a critical priority for our application.

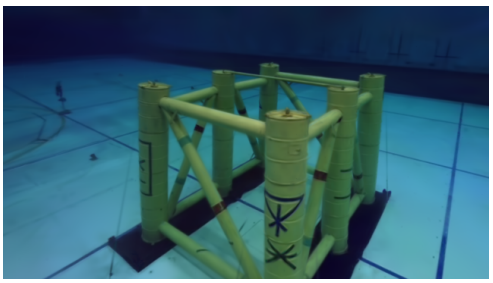
3.3.2 Results in controlled environment

We evaluated the performance of Splatfacto using the data collected from the DOB described in Section. 3.2.1.1. When we trained Splatfacto with the default parameters on the DOB data, we observed long, spiky, black Gaussians on the top of the structures and "cloud"-like artifacts around the structures. This is likely due to lack of sufficient training data with a view of the top of the structure, and lighting differences at different depths. After tuning the hyperparameters of Splatfacto, we managed to remove most of the artifacts. As shown in Fig. 3.9, Splatfacto successfully reconstructed the scene and synthesized novel views with high fidelity. Minor artifacts are observed (as shown in 3.9c and 3.9d), particularly around the boundaries of the collected data. These artifacts can increase the size of the difference image between the rendered and actual images. Compared to NeRF-based approaches, 3D-GS appears to produce more artifacts, potentially leading to larger difference images and, consequently, greater transmission size. Nevertheless, due to the significant advantage of 3D-GS in rendering speed, we select it for our application, which prioritizes real-time performance.

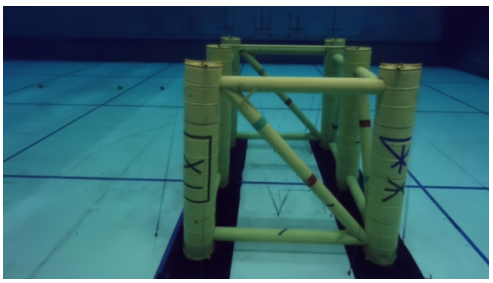
3.3.3 Results in Singapore waters

3.3.3.1 Singapore waters dataset

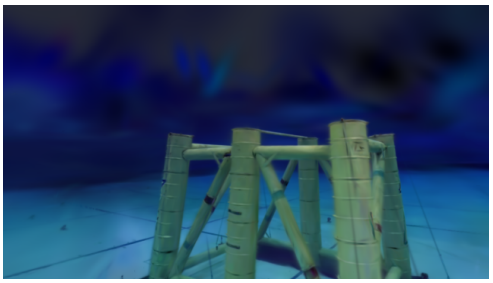
To further evaluate the effectiveness of 3D-GS on real underwater data, we collected a dataset from a bay near St. John's Island, Singapore (SJI). Using Hydra, our customized ROV, we surveyed a submerged pile approxi-



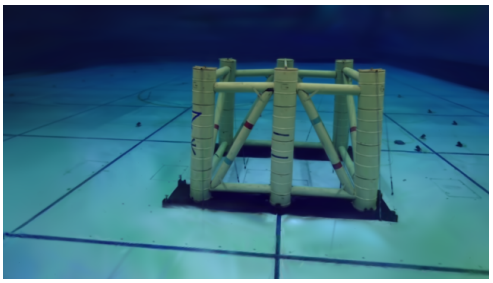
(a) Artifact-free 3D-GS rendering of the DOB structure.



(b) Artifact-free rendering from a different viewpoint.



(c) Artifacts near the top region when viewed from below due to sparse training coverage.



(d) Corner artifacts from distant viewpoints, where less training data was available.

Figure 3.9: Evaluation on the test set of the DOB scene using 3D-GS. Subfigures (a) and (b) demonstrate accurate, artifact-free rendering. Subfigures (c) and (d) highlight cases where insufficient training views lead to noticeable reconstruction artifacts.

mately 5 m tall and 0.5 m in diameter. Although the pile was a plain black structure, barnacles and algae growing on its surface provided useful visual features. During the initial mapping run, The ROV followed a vertical lawnmower path around the pile while recording video with its onboard camera.

High turbidity in the water severely limited visibility (see Fig.3.10a). To capture usable imagery, the ROV was operated at close range, with an average distance of about 1 m from the structure. This proximity restricted the camera's field of view and reduced overlap across viewpoints, making camera pose estimation with COLMAP more difficult. Furthermore,

while the top 1.5 m of the pile exhibited sharp and distinctive features (see Fig.3.10b), deeper regions were heavily biofouled. These areas were covered with soft, repetitive algae structures that moved with the water (see Fig. 3.10c), further complicating reliable pose computation in COLMAP due to the lack of static, distinctive features for robust multi-view matching.

3.3.3.2 Results

As shown in Fig.3.11, Splatfacto successfully reconstructs underwater scenes, particularly the upper portion of the pile, despite the challenging real-world conditions. Minor artifacts remain near the boundaries of the dataset (see Fig.3.12a), likely due to limited training data at those viewpoints. However, these have minimal impact on the overall reliability of the 3D reconstruction. The lower portion of the pile appears noticeably blurrier (see Fig. 3.12b), primarily because of the motion of soft biofouling features. Mitigating such issues may require more advanced modeling techniques capable of handling non-rigid or dynamic scene elements.

3.4 Summary

In this chapter, we introduced NVSPrior, a novel NVS prior-based image compression approach for underwater inspection missions. This approach leverages NVS-rendered priors to enhance compression efficiency by reducing redundancy in camera-captured images from previously visited environments.

To evaluate its feasibility, we analyzed the performance of NeRF and

(a) Low visibility at a distance due to high turbidity.

(b) Top portion of the pile with sharp, distinctive features.

(c) Heavily biofouled lower portion with soft, repetitive algae structures affected by water motion.

Figure 3.10: Representative camera images captured by the ROV during inspection at Singapore waters, illustrating challenges posed by turbidity, limited visibility, and biofouling.

3D-GS models in reconstructing underwater structures across different environments. We first assessed NeRF methods using a dataset collected in the DOB facility with Nerfacto. The results demonstrated that NeRF

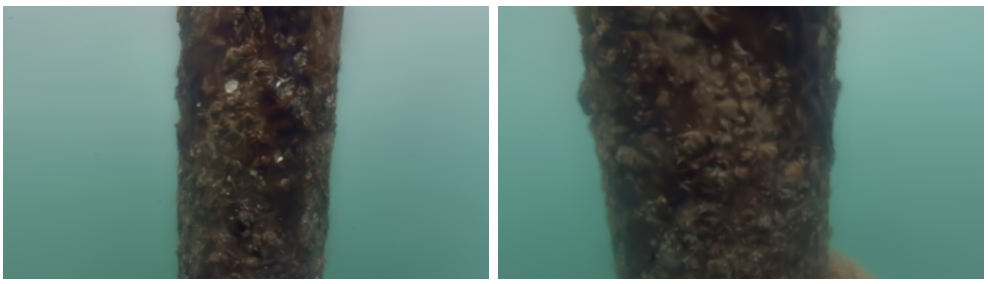


Figure 3.11: High-quality 3D reconstructions produced by 3D Gaussian Splatting, demonstrating the method's effectiveness in recovering static features in challenging underwater environments.

(a) Boundary artifacts likely caused by limited training views at dataset edges.

(b) Blurring in the lower pile due to motion of non-rigid biofouling features.

Figure 3.12: Examples of reconstruction artifacts in 3D-GS results.

models were capable of generating photorealistic underwater reconstructions. However, standard NeRF models exhibited artifacts caused by transient objects, such as moving cables, which degraded reconstruction quality. Our experiments showed that incorporating transient embeddings or using

robust loss functions effectively mitigated these issues and enabled high-fidelity image synthesis. These improvements in rendering fidelity translated directly to smaller difference images—critical for enhancing compression performance within our framework.

We then extended our evaluation to real-world submarine data collected in turbid waters. Under these challenging conditions, standard NeRF models failed to produce high-quality novel views due to limited visibility and increased scattering. In contrast, robustness-enhanced variants successfully reconstructed underwater structures by selectively handling scene inconsistencies, rendering them more suitable for real-world deployment.

Additionally, we explored 3D-GS using Splatfacto and evaluated its performance on datasets from both the DOB and open waters around Singapore. Our results indicated that 3D-GS was a compelling alternative for underwater NVS, offering efficient training and real-time rendering capabilities. Although artifacts were more pronounced compared to robust NeRF models, careful hyperparameter tuning substantially reduced these issues and resulted in visually consistent, reliable reconstructions of complex underwater structures.

Overall, this study highlighted the potential of NVS techniques—particularly 3D-GS—for efficient underwater scene reconstruction and image compression. With a rendering speed exceeding 100 frames per second, 3D-GS enables real-time execution of our NVS prior-based method.

Chapter 4

Pose Estimation from Camera Images

Accurately estimating the pose at which the rendered image best matches the camera image is critical to the effectiveness of NVSPrior. However, as discussed in Chapter 2, reliable pose estimation remains a significant challenge in underwater missions. Visual localization offers a promising solution, with deep learning-based methods like PoseNet [61] capable of estimating camera pose from a single image in real-world scenes.

In this chapter, we evaluate the performance of deep learning-based visual localization methods across a range of underwater environments and propose techniques to improve their robustness and accuracy. Section 4.1 presents a preliminary feasibility study, where we implement PoseNet and its variants and assess their performance on datasets collected in an underwater simulator and a tank. We further examine the models' sensitivity to lighting variations and explore strategies to enhance their robustness and accuracy. Section 4.2 extends this evaluation to realistic scenarios by testing pose estimators in a deep ocean basin with a realistic inspection target, and

introduces a novel loss function that incorporates geometric constraints specific to inspection missions. This new formulation improves interpretability, computational efficiency, and localization performance. In Section 4.3, we address the challenge of limited training data in underwater environments by employing NVS techniques, such as NeRF, to generate photorealistic training samples. Section 4.4 enhances localization accuracy by integrating additional sensor data—such as altimeters and compasses—into an EKF for sensor fusion and tracking, increasing robustness in dynamic conditions. Finally, Section 4.5 presents results from open-sea field trials, demonstrating the practical applicability and effectiveness of our proposed pose estimation framework in real-world settings. The work presented in this chapter has been published in [89], [90] and [91].

4.1 Feasibility study of visual localization methods

To perform a preliminary feasibility study of visual localization methods in underwater inspection missions, we implement PoseNet and its variants and evaluate their performance on datasets collected from an underwater simulator and a tank. We also investigate the robustness of such deep learning-based visual localization models against changes in lighting conditions and the effectiveness of different techniques to improve the model performance and robustness.

4.1.1 Methods

The pose regression problem is to estimate a 6-DOF pose from a single RGB image. The pose consists of the x-y-z position and roll-pitch-yaw angle orientation. We use quaternions to represent orientation to avoid wrap-around problems associated with Euler angles [92]. Given a monocular RGB image, I, pose-estimator model outputs a a 7-dimensional (7D) estimated pose vector $\mathbf{y} = [\hat{\mathbf{p}}, \hat{\mathbf{q}}]$ containing a position vector estimate $\hat{\mathbf{p}}$ and quaternion vector estimate $\hat{\mathbf{q}}$.

The original PoseNet model proposed by Kendall et al.[61] used a deep convolutional neural network (DCNN) based on a modified GoogLeNet architecture[93], pretrained on the ImageNet dataset [94]. The softmax classifiers were replaced with an affine regressor, and an additional fully connected (FC) layer with a feature size of 2048 was inserted before the final regressor. However, directly regressing a 7D pose vector from this high-dimensional representation proved suboptimal [62]. To address this, a subsequent study by Walch et al. restructured the FC layer into a 32×64 matrix and applied four long short-term memory (LSTM) networks to perform structured dimensionality reduction [62]. This approach, referred to as CNN+LSTM, demonstrated improved pose estimation performance over the original PoseNet in terrestrial environments. We implement and evaluate both model architectures – the CNN (shown in Fig. 4.1) and the CNN+LSTM (shown in Fig. 4.2). For both architectures, we use pretrained

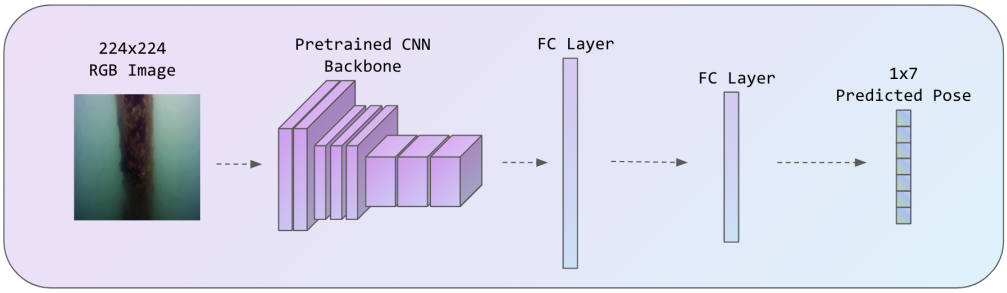


Figure 4.1: Overview of the CNN-based architecture for visual relocalization.

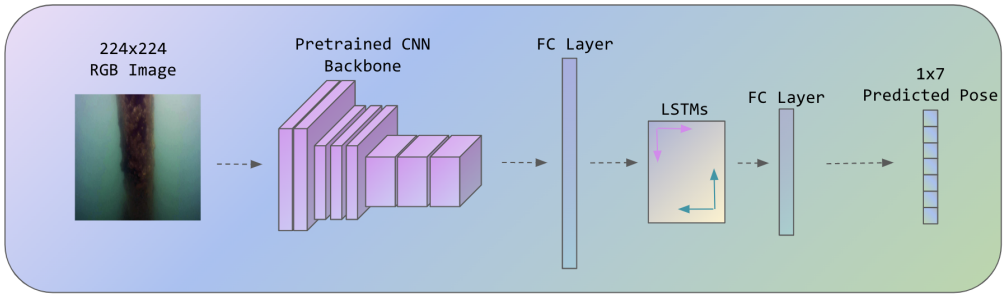


Figure 4.2: Overview of the CNN+LSTM-based architecture for visual relocalization.

DCNNs as the back bone. Pretrained models leverage the benefits of transfer learning as large underwater datasets are not widely available and it is computationally expensive to train the model on underwater datasets from scratch.

For model training, we use a composite loss function that is a weighted sum of the position error and orientation error squared [61]:

$$\mathcal{L} = \mathcal{L}_{\mathbf{p}} + \beta \mathcal{L}_{\mathbf{q}}, \tag{4.1}$$

where $\mathcal{L}_{\mathbf{p}} = ||\mathbf{p} - \hat{\mathbf{p}}||_2$ and $\mathcal{L}_{\mathbf{q}} = \left\| \mathbf{q} - \frac{\hat{\mathbf{q}}}{\|\hat{\mathbf{q}}\|} \right\|_2$, and $\mathbf{p}$ and $\mathbf{q}$ represent the true position and true orientation. β is a free parameter that determines the trade-off between desired accuracy in position and orientation.

The input images used in the training are rescaled to 256×256 pixels

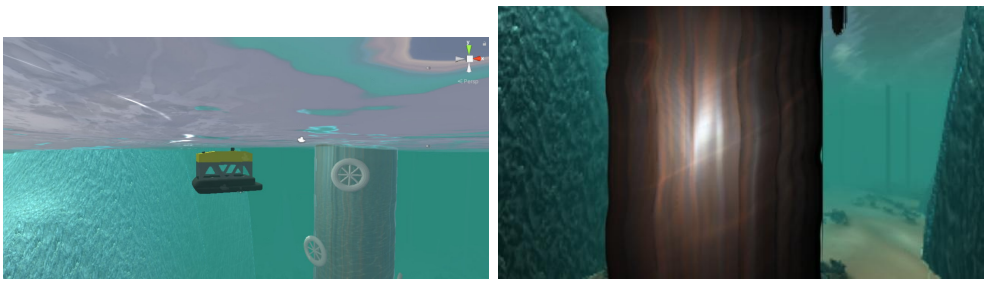


Figure 4.3: Underwater simulator (left) and the image captured by the ROV (right).

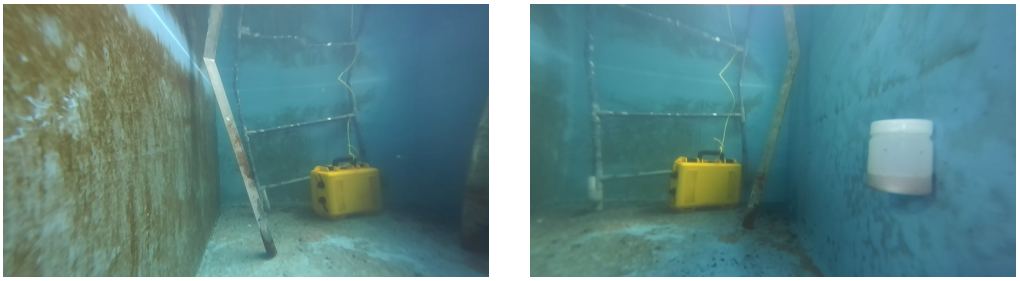


Figure 4.4: Example images from our underwater tank datasets.

before cropping into a 224×224 feature input using center cropping. To speed up training, the images are normalized using their mean and standard deviation. The poses are also normalized to lie within [-1, 1].

4.1.2 Datasets

To train and test our model, we used one dataset collected from an underwater robotics simulator [95] as shown in Fig. 4.3 as well as two datasets collected from a tank as shown in Fig. 4.4.

4.1.2.1 Simulator dataset

In the underwater simulator, we placed a ROV 0.5 m from a vertical pile with 0.7 m diameter. We then operated the ROV to inspect the pile in a downwards spiral motion. The total spatial extent covered by the ROV was about 2 m×4 m×2 m. We recorded the images captured by the front

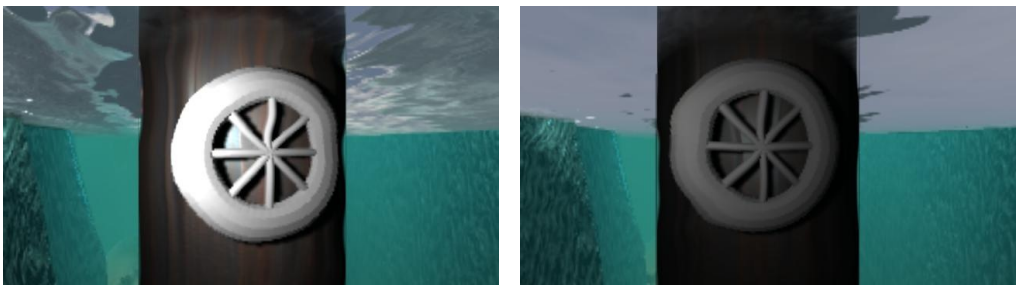


Figure 4.5: Image from original dataset (left) and the image from dimmer dataset (right).

camera of the ROV as well as the poses of the ROV in terms of position vectors and unit quaternions.

We first collected a dataset containing 14,400 images and corresponding poses. We refer to this dataset as Simulator. We randomly selected 70% of the dataset for training and used the remaining 30% for testing. We used random selection to ensure that both train and test sets had information of all scenes covered.

One of the common dynamic changes in underwater inspection missions is change in vehicle lighting. To test the robustness of the pose-estimator model, we dimmed the light on the ROV and collected another dataset; see in Fig. 4.5. The only difference between this dataset and the previously collected is the lighting condition. We refer to this dataset as Simulator-dimmer. We randomly selected 30% of Simulator-dimmer to test the robustness of the model against change in lighting conditions.

4.1.2.2 Tank dataset

We collected two datasets from the tank using a customized ROV. The ROV was equipped with a stereo camera which had an INS integrated. The

INS reported the pose of the left camera. In the first dataset, the ROV followed a lawnmower path with minimal rotations, primarily undergoing translational motion. This dataset contains 3,437 samples and covers a spatial extent of 0.4×0.6×0.2 m. We refer to this dataset as Tank-1. The second dataset was collected while the ROV performed rotational maneuvers at five selected points, resulting in 4,977 samples. We refer to this dataset as Tank-2.

For both datasets Tank-1 and Tank-2, camera poses were obtained from the integrated INS. Given the small coverage area, we assume INS drift is negligible. We only use the images from the left camera.

To study the effect of using a larger dataset, we augmented Tank-1 by adding the right-camera data, and thereby using the geometry of the stereo camera placement to provide more training data. We refer to this dataset as Tank-1-augmented.

4.1.3 Experiments & results

We used a DCNN with GoogLeNet [96] as our baseline model. We trained the model using stochastic gradient descent with a base learning rate of 0.003 and with a momentum of 0.9. Training took about an hour with a batch size of 4 using a RTX2060 Nvidia GPU. We set β as 4 for training the simulator dataset and 30 for training the tank datasets. The base learning rate and batch size were chosen through hyper-parameter tuning using random search while β was chosen using grid search. We used Ray

Tune [97], an open-source package, for implementing the hyperparameter searching. To improve the efficiency of hyper-parameter tuning, we utilized an asynchronous hyperband scheduler [98], which ensured that compute resources were used efficiently by continuously scheduling new trials and stopping underperforming ones as soon as possible.

We evaluate the model performance by comparing the predicted camera poses with the ground-truth poses over all test frames. For each frame, the positional error is defined as the Euclidean distance between the estimated and ground-truth translation vectors, and the angular error $\mathcal{L}_\theta$ quantifies the rotation discrepancy between the predicted and ground-truth orientations. Given their quaternion representations $\mathbf{q}$ and $\hat{\mathbf{q}}$, we compute the relative quaternion $\Delta\mathbf{q} = \mathbf{q} \cdot \hat{\mathbf{q}}^*$, and derive the angular error in degrees as:

$$\mathcal{L}_\theta = 2\cos^{-1}(|\Delta q_w|) \times \frac{180}{\pi}, \tag{4.2}$$

where Δq_w is the scalar component of $\Delta\mathbf{q}$. In this section, we report each model's localization performance in terms of mean positional error and mean angular geodesic error across test dataset; lower values indicate better performance.

The baseline model demonstrates strong localization performance, achieving centimeter-level positional accuracy and angular errors consistently below 3° in both simulated environment and small tank, as shown in Table. 4.1. In the simulator dataset, it reaches a positional error of 0.0891 m and an angular error of 2.91°. In the Tank 1 dataset, the baseline model achieves

Table 4.1: Mean localization error on Simulator and Tank datasets across model variants, reported as positional and angular errors.

Dataset	Model	Position error (m)	Orientation error (°)
Simulator	Baseline	0.0891	2.91
	ResNet-50	0.0657	2.15
	LSTM	0.0624	2.06
Tank-1	Baseline	0.0427	1.29
	ResNet-50	0.0507	1.17
	LSTM	0.0340	0.79
Tank-2	Baseline	0.0464	1.42
	ResNet-50	0.1070	2.18
	LSTM	0.0649	1.45
Tank-1 (aug.)	Baseline	0.0350	0.67
	ResNet-50	0.0234	4.71
	LSTM	0.0406	0.57

0.0427 m positional error and 1.29° angular error. Similarly, in the Tank 2 dataset, the model maintains strong performance, with a positional error of 0.0464 m and an angular error of 1.42°. The errors are minimal and of comparable magnitude to the noise in the pose recorded by the camera sensors. This consistency in performance across different datasets highlights the model's robustness and reliability. The accurate localization of the baseline model is further illustrated in Fig. 4.6, Fig. 4.7 and Fig. 4.8. These plots show that the model effectively estimates both position and orientation with high precision, demonstrating minimal sensitivity to variations in movement patterns.

To evaluate the impact of deeper networks on performance, we utilized ResNet [99] models pretrained on ImageNet. By leveraging residual blocks, ResNets mitigate exploding and vanishing gradient issues, allowing for signif-

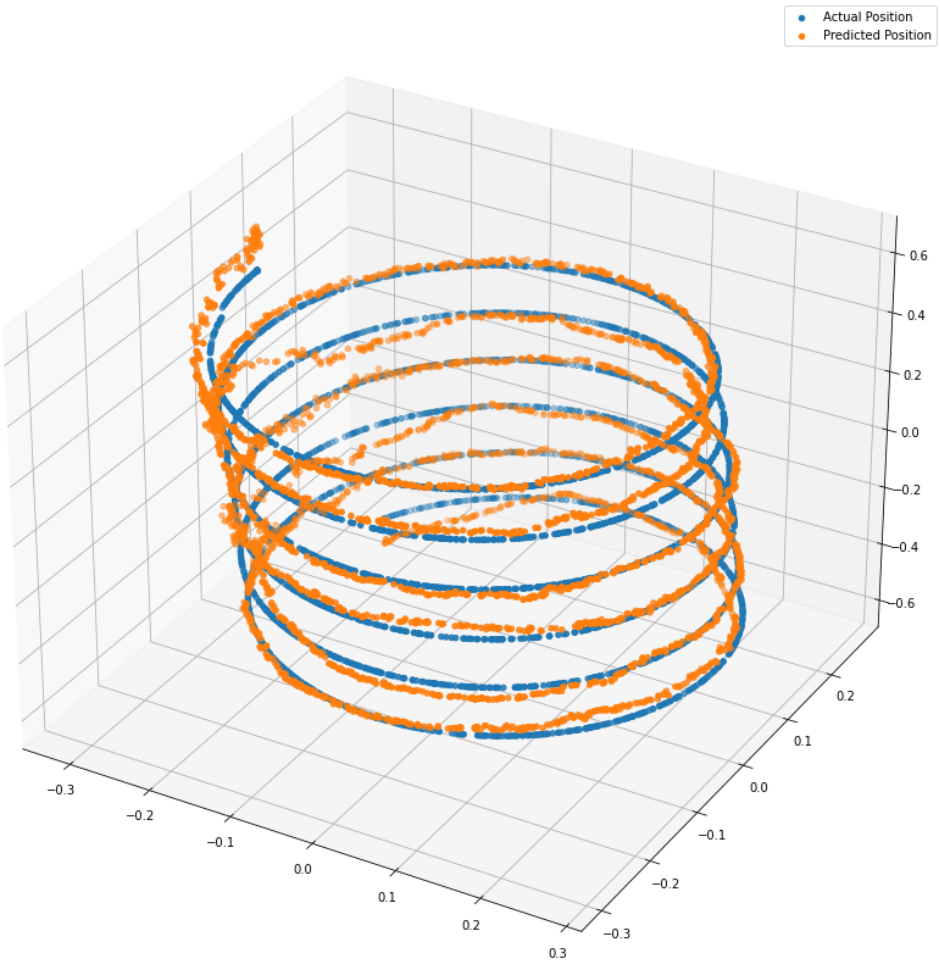


Figure 4.6: Predicted trajectory (orange) vs real trajectory (blue) for simulator dataset.

icantly deeper architectures than GoogLeNet. They have also demonstrated superior performance in various visual recognition tasks, including ImageNet classification [99]. We implemented four ResNet models of varying depths and observed a clear trend: deeper networks consistently achieved higher localization accuracy; see Table. 4.2. However, this improvement comes at the cost of increased model complexity, requiring more storage and longer training times. Striking a balance between computational efficiency and

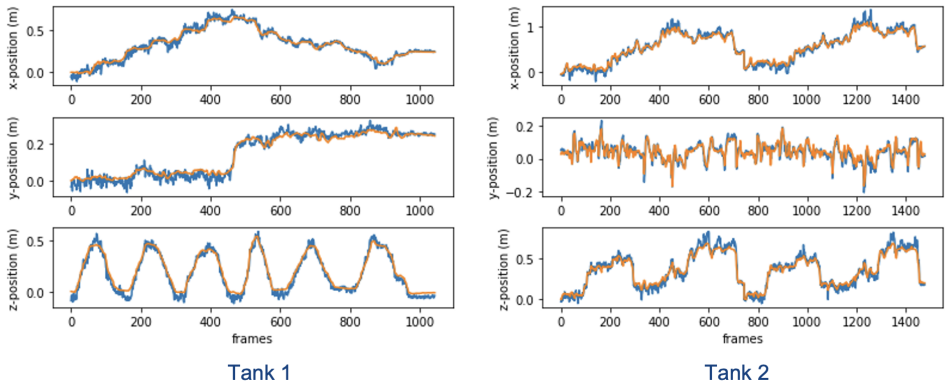


Figure 4.7: **Position estimation for Tank-1 and Tank-2.** Estimated position of the vehicle (orange) is close to the actual position (blue).

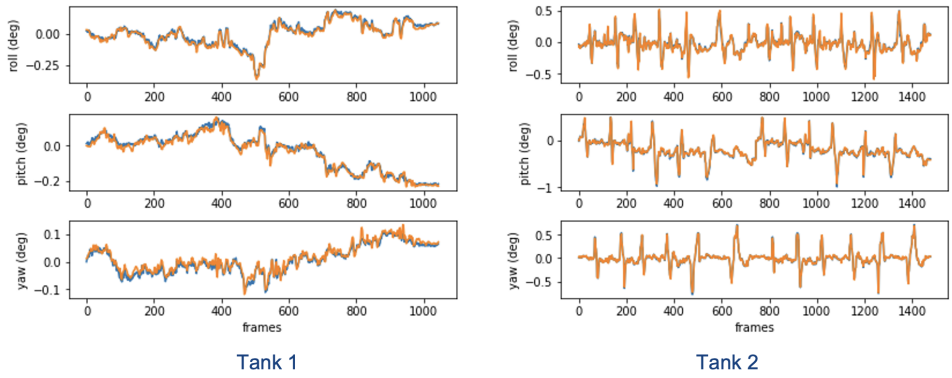


Figure 4.8: **Orientation estimation for Tank-1 and Tank-2.** Estimated orientation of the vehicle (orange) is close to the actual orientation (blue).

Table 4.2: Mean localization error by backbone. Models are trained on **Simulator** and evaluated on **Simulator** and **Simulator-dimmer**.

	Simulator		**Simulator-dimmer**	
Backbone	**Position (m)**	**Orientation (°)**	**Position (m)**	**Orientation (°)**
GoogLeNet	0.125	2.63	0.389	15.3
ResNet-18	0.089	3.22	0.421	17.0
ResNet-34	0.078	2.36	0.330	21.3
ResNet-50	0.066	2.15	0.348	11.1
ResNet-101	0.059	1.70	0.370	6.59

accuracy, we selected ResNet-50 as the backbone for subsequent experiments.

To assess the effect of incorporating LSTMs to exploit spatial correla-

tion of the image features and to achieve more structured dimensionality reduction, we introduced an LSTM layer on top of the DCNN with the ResNet-50 model.

As shown in Table. 4.1, on the simulator dataset, which is free from noise, turbidity, light distortion, and other real-world underwater challenges, the ResNet-50 and LSTM model outperformed the baseline. However, on tank datasets, which include distortions typical of underwater environments, ResNet-50 underperformed compared to the baseline, whereas the LSTM-enhanced model achieved better results on Tank-1 (which primarily featured translation with minimal rotation). This suggests that despite regularization, ResNet-50 may be slightly overfitting, limiting its generalization to the tank dataset.

Data augmentation using images from both cameras in a stereo setup significantly improves model performance. Thus, when available, stereo data should be leveraged to bolster results.

4.1.3.1 Robustness towards change in vehicle lighting

We tested the baseline model, which is trained on the Simulator dataset, on the Simulator—dimmer dataset. Fig. 4.9 presents the cumulative distribution function (CDF) of position and orientation errors in performance when tested on the two datasets. The results indicate a significant increase in localization errors when the baseline model—trained on data collected under brighter lighting conditions—is applied to scenes with dimmer light-

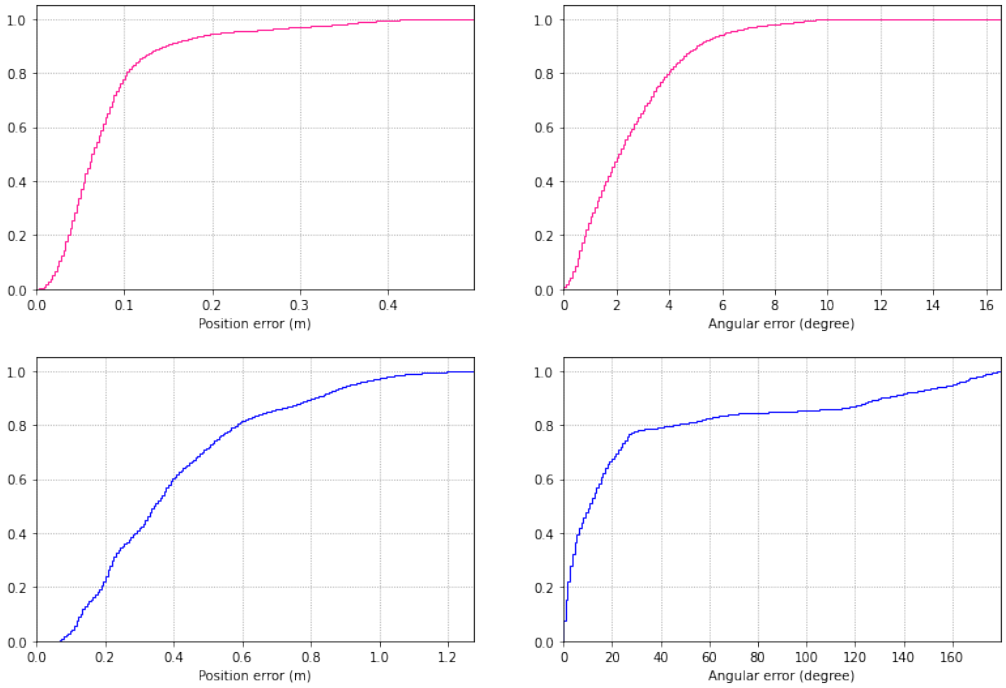


Figure 4.9: CDFs of position and angular estimation errors for Simulator (top) and Simulator-dimmer (bottom). Higher/left-shifted curves indicate lower error. The dimmer set shows larger errors and heavier tails, especially for orientation.

ing. This suggests that the model lacks robustness to lighting variations, which is expected, as differences in illumination can substantially alter the visual appearance of the scene.

To improve the robustness towards lighting changes, we investigated the effectiveness of using data augmentation and deeper networks.

We applied color jittering to the standard dataset during training by randomly changing image brightness, contrast, saturation and hue. As shown in Table 4.3, after applying color jittering, the robustness to changes in lighting condition improved slightly. We also tested the effectiveness of applying contrast limited adaptive histogram equalization, a more advanced color enhancement method, during training. The robustness towards lighting

Table 4.3: Effect of training-time photometric augmentation on localization error under lighting shift. Models are trained on **Simulator** (with/without augmentation) and evaluated on **Simulator** and **Simulator-dimmer**.

	Simulator		**Simulator-dimmer**	
Data Augmentation	**Position (m)**	**Orientation (°)**	**Position (m)**	**Orientation (°)**
Disabled	0.089	2.91	0.468	37.8
Enabled	0.125	2.63	0.389	15.3

Table 4.4: Localization error using a model trained on the combined **Simulator + Simulator-dimmer** dataset.

Test dataset	**Position (m)**	**Orientation (°)**
Simulator	0.21	1.83
Simulator-dimmer	0.27	2.00

changes was further improved.

We also explored the impact of training with mixed lighting conditions. We combined the Simulator and Simulator-dimmer datasets to create a larger dataset, randomly selecting 70% for training. The trained model was then evaluated separately on test sets from both lighting conditions, as shown in Table 4.4. Training with more diverse data significantly improved robustness to lighting changes by making the training set more representative of the test data. However, performance on Simulator dataset declined notably, indicating potential overfitting in our previous results. This underscores the importance of training with a more diverse dataset to enhance generalization.

To assess the impact of deeper networks on robustness, we used ResNet [99] models of different layers pretrained on ImageNet. As shown in Table 4.2, deeper networks offer only a modest improvement in robustness to lighting changes.

4.2 Pose estimation in controlled environment

The previous section demonstrated the effectiveness of PoseNet and its variants in controlled environments, such as tanks with toy structures and simulated underwater settings. However, their performance in real-world scenarios remains untested. In this section, we evaluate the performance of neural network-based pose estimators in confined water environments with realistic structures and explore several extensions to improve their applicability to underwater inspection. These include a geometry-informed loss function tailored for inspection tasks, an investigation into the use of grayscale inputs for computational efficiency, and an assessment of the models' ability to both interpolate within and generalize across different datasets.

4.2.1 Methods

We implement and evaluate both model architectures – the CNN (shown in Fig. 4.1) and the CNN+LSTM (shown in Fig. 4.2), following the methods described in section 4.1.1. Additionally, we assess the performance of these using a pretrained ResNet50 [100] as the backbone.

Kendall et al. [61] used a composite loss function (Equation 4.1), which is a weighted sum of the (1) L2 loss $\mathcal{L}_{\mathbf{p}}$ between the predicted positions and the true positions, and the (2) L2 loss $\mathcal{L}_{\mathbf{q}}$ between the predicted quaternions and the true quaternions. It uses a free parameter, β, to determine the trade-off

between the desired accuracy in translation and orientation. In PoseNet and CNN+LSTM, the value of β is fine-tuned using a grid search to ensure the expected value of position and orientation errors are approximately equal, which the authors suggest lead to overall optimal performance. We refer to this loss function as the β-loss.

We argue that the β-loss is not the optimal approach to our problem, due to three reasons. Firstly, we argue that optimal performance is not necessarily achieved when position and orientation errors are roughly equal. Instead, the performance criteria and loss should incorporate geometry and physics relevant to the inspection task at hand. Secondly, the L2 loss between the predicted and true quaternions does not directly translate to an orientation error interpretable in degrees or radians, and thus, it does not accurately reflect the geometric distance between the predicted and true orientations. Thirdly, searching for the optimal β value often involves extensive computational resources. This search can become a significant bottleneck, especially in scenarios where training needs to be done fast.

To overcome these shortcomings, we propose a new loss function more relevant to our problem, the d-loss, to improve the training effectiveness, interpretability and efficiency. The d-loss is defined as:

$$\mathcal{L} = \mathcal{L}_{\mathbf{p}} + d\mathcal{L}_{\theta}. \tag{4.3}$$

Note that we have replaced the quaternion loss in (Equation 4.1) with a loss based on the Eulerian angular difference, $\mathcal{L}_{\theta}$, which is calculated as follows.

We first determine the rotation between the estimated and ground truth quaternions through quaternion multiplication, $\mathbf{\Delta q} = \mathbf{q}\,(\hat{\mathbf{q}}/\|\hat{\mathbf{q}}\|)^*$, where * denotes the conjugate of the quaternion. $\mathbf{\Delta q}$ is a unit quaternion which can be expressed as $(r, \vec{v})$ where r is the scalar part of the quaternion, and $\vec{v}$ is the vector part. r is related to a spatial rotation around a fixed point of $\mathcal{L}_\theta$ radians about a unit axis by $r = \cos(\mathcal{L}_\theta/2)$ [101], thus $\mathcal{L}_\theta = 2\cos^{-1}(r)$. We approximate $\mathcal{L}_\theta \approx \frac{\pi}{2}(1 - r)$, using a Taylor series approximation, which is valid for small rotation angles where $r \approx 1$. For large angle deviations, this approximation becomes less accurate, and using the exact formula $\mathcal{L}_\theta = 2\cos^{-1}(r)$ is recommended. In our use case, however, the model is trained to minimize pose errors, and angular deviations remain small in practice. The approximation also offers computational efficiency for model training. The Eulerian angular difference loss provides a more intuitive and direct measure of orientation error.

Additionally, we replace the hyperparameter weight factor β in (Equation 4.1) which required tuning, with the average distance d between the camera and the object of interest. In our experiments, d is computed based on prior knowledge of the inspection setup. In more typical deployments, d can be estimated using onboard sensors such as forward-looking sonars, or stereo depth estimation. The intuition here is that this factor translates the rotational error to an equivalent "average" translational error (attributed to the orientation difference). Thus, the overall loss can be interpreted as the "total positional error" in meters, including contributions from translational

and orientation error components. Note that this formulation relies on several assumptions typical of underwater reinspection scenarios. It assumes that the vehicle maintains a relatively constant distance d from the structure during inspection, and that the camera is generally oriented toward the target (i.e., the bearing is aligned). We also assume the orientation errors are small enough for the small-angle approximation to hold. These conditions are commonly met in many kinds of underwater inspection tasks, where deliberate movements are required for safety and image quality.

The translation between rotational error and the "average" translational error is described as follows. As illustrated in the example in Fig. 4.10, if the camera has a pitch orientation error $\mathcal{L}_\theta$ of θ, the point it observes on the structure remains roughly the same as if the camera had an equivalent translational error $\mathcal{L}_\mathbf{p}$ of h (i.e., moves up by h) for small values of h and θ. Based on the geometry, equivalent translational error can be expressed in terms of orientation error $\mathcal{L}_\theta$ and the average horizontal range between the camera and the structure as:

$$\mathcal{L}_\mathbf{p} = d\tan(\mathcal{L}_\theta). \tag{4.4}$$

Assuming the case when the rotational error is small, we approximate $\tan(\mathcal{L}_\theta) \approx \mathcal{L}_\theta$. Thus, we obtain:

$$\mathcal{L}_\mathbf{p} \approx d\mathcal{L}_\theta. \tag{4.5}$$

This modified loss function (Equation 4.3) leverages the inherent geometric relationship between positional and rotational errors in inspection

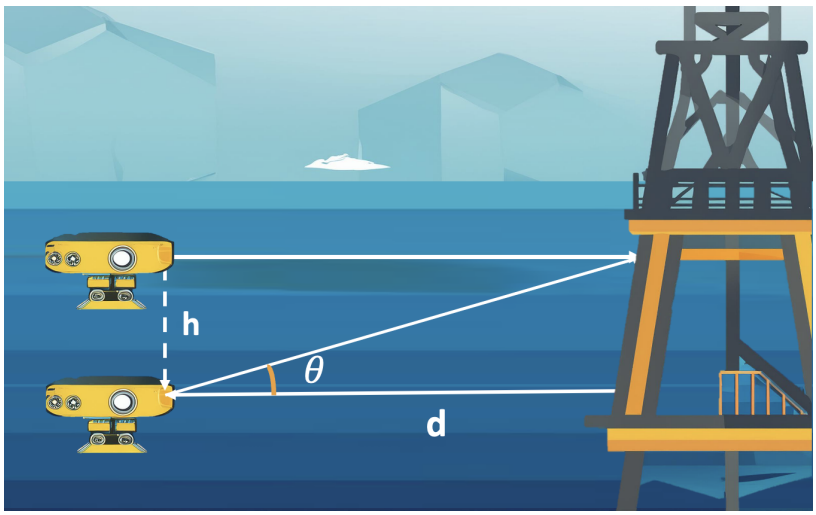


Figure 4.10: Schematic showing the interpretation of the orientation error in terms of equivalent translational error. This assumes slow motion, small angles, and constant range.

missions. By converting orientation error into an equivalent translational error using the physical distance d, both terms are expressed in the same unit (meters). This provides a more interpretable loss function with physical meaning, and avoids the need for manually tuning trade-off weights like β, simplifying the scaling challenge in pose estimation.

To evaluate the effectiveness of deeper backbones, additional LSTM layers, the proposed d-loss, and the color information in images, we tested multiple configurations of the two visual localization network architectures. The details of these configurations are summarized in Table 4.5.

As part of our investigation into color information, we explored the use of grayscale input to reduce input dimensionality and potentially improve computational efficiency, under the assumption that underwater images often contain limited useful color information due to turbidity and poor lighting. To preserve the benefits of transfer learning, we adapted a pretrained

Table 4.5: Description of configurations

ID	Architecture	Backbone	Loss	Color
C1	CNN	GoogLeNet	β-loss	RGB
C2	CNN	GoogLeNet	d-loss	Grayscale
C3	CNN	GoogLeNet	d-loss	RGB
C4	CNN	ResNet50	d-loss	RGB
C5	CNN+LSTM	GoogLeNet	d-loss	RGB
C6	CNN+LSTM	ResNet50	d-loss	RGB

GoogLeNet model, which is originally designed for RGB input, to accept grayscale input. This was done by modifying the first convolutional layer to accept a single-channel input instead of three channels. The weights were initialized by summing across the RGB channels of the pretrained filters, and the modified layer was fine-tuned during training. This modification reduced the number of parameters in the first convolutional layer by a factor of three, since it now operates on a single channel instead of three.

During both training and testing for all configurations, we rescaled input images directly into a 224×224 pixels input, deviating from PoseNet's approach of resizing the images to 256×256 before cropping into 224×224. This adjustment was made to minimize the loss of image information, a concern particularly acute in underwater images where available information is inherently more limited compared to terrestrial settings. To speed up training, we normalized the images against the ImageNet dataset's mean and standard deviation. Additionally, poses are normalized to lie within the range [-1, 1].

We used the PyTorch deep learning framework to implement and train

the models. The experiments were conducted using an RTX 6000 Ada GPU. For training, we used the stochastic gradient descent optimizer for configurations C1, C2, and C3. For the remaining configurations, we used the Adam optimizer. A batch size of 32 was used. Hyperparameters, including the learning rate, weight decay, and β for C1, were tuned using grid search strategy over a predefined set of values. The best set of hyperparameters was selected based on validation performance. Training continued until early stopping was triggered.

4.2.2 Datasets

We collected data from the DOB setup described in Section 3.2.1.1.

We used our ROV, Hydra, for data collection. Hydra was customized based on the BlueROV2 platform. The ROV was equipped with a 1080p monocular camera provided by Blue Robotics for collecting RGB image data. For navigation and sensor integration, the vehicle used the BlueROV2 Navigator Flight Controller (NFC) mounted on a Raspberry Pi 4. The NFC includes an onboard IMU, compass, depth sensor, and Analog-to-Digital Converter sensors. The compass and depth sensor were used to provide orientation and depth information, respectively. To estimate the horizontal (x-y) position of the ROV, we employed a customized ultra-short baseline (USBL) positioning system. This setup was based on a Subnero high-speed acoustic modem (model WNC-S40HSS4+xCh) configured with four receivers and deployed near the operating region, as illustrated in

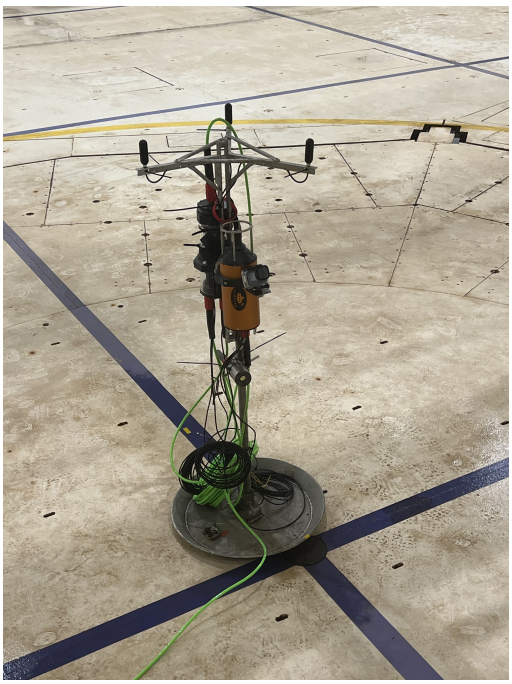


Figure 4.11: The USBL setup at TCOMS to estimate the location of the ROV.

Fig. 4.11. The USBL system enabled accurate localization of the vehicle during the trials in the TCOMS basin.

We executed three trials within the environment at different depths to gather data while the ROV surveyed the structure. Each trial features a roughly similar lawnmower trajectory around the structure, with a total path length of approximately 37 m per trial. The trials were conducted at average depth levels of -1.5 m, -3 m, and -4 m, respectively.

The sensor data from the vehicle was captured using ROS (Robot Operating System) and sampled at a frequency of 5 Hz. We synchronized the sampled data with the USBL position estimates based on timestamps and interpolated where necessary. For ground truth, we used the x and y coordinates from the USBL, and the z coordinate and the orientation data

Table 4.6: Description of datasets

ID	Dataset Name	Dataset Size
D1	Clear Water-Deep	2165
D2	Clear Water-Shallow	2956
D3	Clear Water-Mid	933
D4	Clear Water-NVS	4193
D5	Sea Water-1	2360
D6	Sea Water-2	735
D7	Sea Water-NVS	18918

Table 4.7: Performance of all configurations trained and tested on dataset D1. $\mathcal{L}_{\mathbf{p}}$ and $\mathcal{L}_{\theta}$ tabulated are the median of estimation errors across the test data. $\mathcal{L}$ was calculated using Equation 4.3 with $d = 3$ m. The best performance for each metric is highlighted in bold.

ID	$\mathcal{L}$ (m)	$\mathcal{L}_{\mathbf{p}}$ (m)	$\mathcal{L}_{\theta}$ (°)	Inference time (ms)
C1	2.41	2.36	**0.86**	2.20
C2	0.61	0.53	1.50	1.65
C3	0.41	0.36	0.99	1.62
C4	0.34	0.29	0.88	1.16
C5	0.30	0.22	1.51	0.78
C6	**0.19**	**0.12**	1.34	**0.77**

from the NFC.

From the recorded data collected during these trials, we curated three datasets, referred to as D1, D2, and D3. These datasets vary in depth and size, as summarized in Table 4.6. Notably, D3 was constructed by downsampling the raw data from the third trial to create a more challenging dataset for testing purposes.

4.2.3 Results & discussions

We present the performance of different configurations in Table 4.7. The benchmark for our evaluation is the performance of C1.

We observe the following:

1. Comparing the performance of C3 against C1, our results demonstrate that training with our proposed d-loss significantly enhances model performance, especially in terms of the overall performance metric $\mathcal{L}$. This improvement is attributed to the d-loss's ability to provide a physically interpretable measure of pose error by expressing both translation and orientation errors in the same unit (meters). This eliminates the need for manual tuning of trade-off weights and leads to more stable training.

2. Comparing the performance of C2 against C3, it can be observed that using grayscale images shows significantly worse performance and too little an improvement in inference time, contrary to our initial expectation. The worse performance of grayscale images can be attributed to the fact that since D1 was collected in a non-turbid fresh water environment, the color information in the underwater images is not as limited as one might anticipate in an image taken in a sea environment. As shown in Fig. 4.12, the underwater RGB images in D1 retain valuable color information that may provide distinguishing features in these environments. Thus, the grayscale images have much less information than RGB images and thus lead to poorer performance. The lack of improvement in inference time is due to the fact that we only reduce the number of channels in the first CNN layer of the pretrained model, resulting in a minimal reduction in computational load. To achieve more substantial computational savings, the entire model architecture would need to be better streamlined for grayscale images, not

just the initial layer.

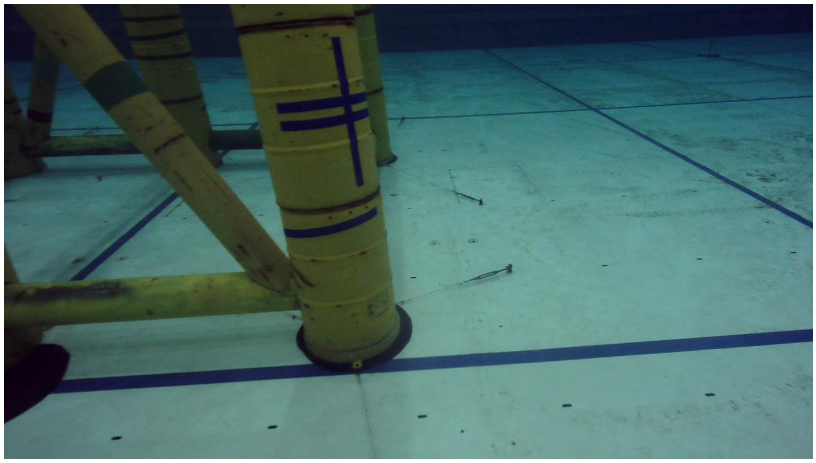


Figure 4.12: Sample camera image captured in TCOMS.

3. Comparing the performance of C6 to C5 and C4 to C3 shows that using ResNet50, a deeper network, as the backbone, improves performance for both CNN and CNN+LSTM. This is likely due to ResNet50's higher representational capacity and its residual connections, which facilitate better feature extraction and gradient flow during training. These benefits are especially useful in underwater scenes where discriminative features may be subtle or degraded. The observed improvements were consistent across several configurations, indicating that the choice of backbone architecture plays a substantial role in pose estimation accuracy.

4. Comparing the performance of C6 to C4 and C5 to C3 shows that the CNN+LSTM architecture consistently outperforms the CNN architecture. This improvement can be attributed to the LSTM layers' ability to perform structured dimensionality reduction, helping the network learn more meaningful and stable pose representations.

Among all the configurations, C6, which uses the CNN+LSTM architecture with the ResNet50 backbone and is trained using the proposed d-loss, performs the best, achieving 0.12 m of positional accuracy and 1.34°of orientation accuracy with an inference time of 0.77 ms.

We test the performance of generalization using the model with the best configuration, C6. We first trained the model on D1 and tested on D3. A significant performance degradation is observed, as shown in the first row of Table 4.8. This is on expected lines because the test data is sampled from a different distribution than the training data with possibly different paths and conditions, and deep-learning models often fail to extrapolate beyond the bounds of the training data.

To address this issue, we evaluate the use of a larger and more diverse training dataset, by expanding the training data to include both D1 and D2. This augmentation introduces a wider distribution of data, notably enhancing the diversity in depth information. This leads to a 49% improvement in model performance in overall loss, as shown in the second row in Table 4.8.

These findings underscore the importance of comprehensive baseline mapping to collect sufficiently diverse training data. This is essential for training models that are robust enough to perform accurate localization during reinspection tasks.

Table 4.8: Performance of configuration C6 on dataset D3. $\mathcal{L}_{\mathbf{p}}$ and $\mathcal{L}_{\theta}$ are median estimation errors across the test data. $\mathcal{L}$ was calculated using Equation 4.3 with the average distance $d = 3$ m. The best performance for each metric is highlighted in bold.

Training Dataset	**EKF**	**Color Jittering**	**Performance Metrics**		
			$\mathcal{L}$ (m)	$\mathcal{L}_{\mathbf{p}}$ (m)	$\mathcal{L}_{\theta}$ (°)
D1			1.45	1.34	2.09
D1+D2			0.75	0.58	3.20
D1+D2	✓		0.47	0.47	0.00
D1+D2+D4			0.52	0.40	2.28
D1+D2+D4		✓	0.20	0.15	0.93
D1+D2+D4	✓	✓	**0.11**	**0.11**	**0.00**

4.3 Augmented Training with Novel View Synthesis

The previous section demonstrated the importance of diverse training data with good coverage of the surveyed location. Although it may sometimes be possible to collect such data by extensively covering areas during the baseline mapping run, the practical constraints of cost and labor often limit this approach or render it infeasible. We explore alternative approaches to improve model performance in such data-limited scenarios. We propose to use NVS techniques to create models of the 3D scene, and then use these to generate more images from new aspects to augment the training data. In this section, we present the methods of augmenting training data using NVS models and the results of this approach.

4.3.1 Methods

We first select 540 images from D1 and D2 to train an NVS model for the TCOMS scene. For this, we employ COLMAP [102, 40], an open-source

SfM computation software, to compute the camera pose associated with each image within an arbitrary reference coordinate.

We employed the nerfacto pipeline from nerfstudio, an open-source library that provides a modular and user-friendly framework for training, and evaluating NVS-based 3D scene representations [84], as our NVS model to render views for training data augmentation. Nerfacto is a simplistic modular NeRF implementation that adopts recent advancements to improve computational efficiency and handle unbounded scenes [84].

To train the model, we used 540 images from the original trials, along with their corresponding poses estimated via COLMAP. Inspired by the RobustNeRF variant [46], we replaced the default nerfacto loss with a robust photometric loss that down-weights inconsistent or noisy regions during training. This improves rendering quality in scenes with transient features or non-uniform illumination. The details of training the model are presented in our previous chapter 3.

To generate novel camera poses for rendering, we applied controlled perturbations to the original COLMAP-estimated poses. For each pose, we randomly sampled a new depth value within the feasible range, defined by the minimum and maximum depths observed in the collected data, and replaced only the z-coordinate to preserve the viewing direction. Additionally, we perturbed the x and y positions by scaling the vector from the pose to the structure using a random factor sampled from the range [0.8, 1.2], effectively varying the lateral distance while maintaining orientation toward

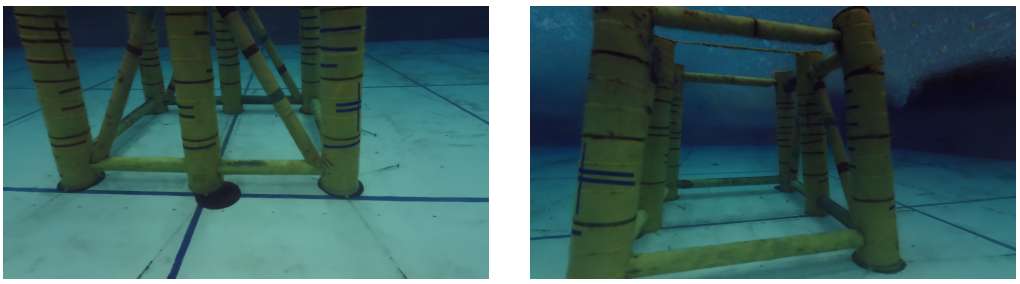


Figure 4.13: NVS rendered images for scenes in TCOMS. The rendered images produce photorealistic views of the structure but exhibit discrepancies in brightness. Some of the rendered views have artifacts in the background as shown in the image on the right.

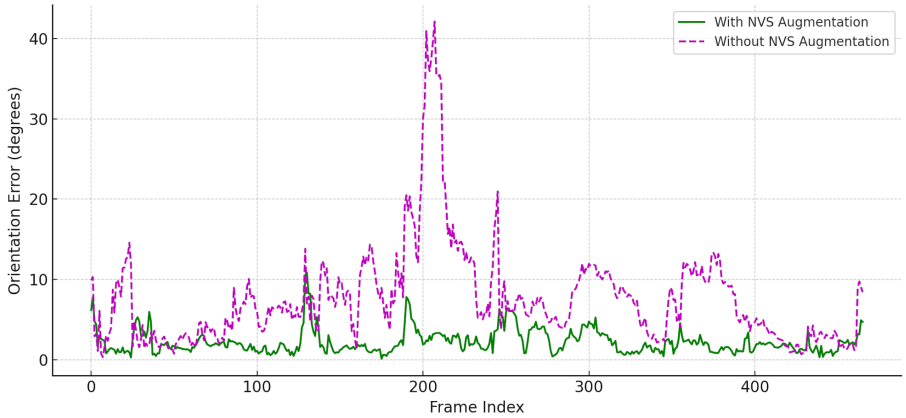
the target. These synthesized poses were kept within the scene bounds to ensure rendering consistency. The trained NVS model was then used to render photorealistic images at these new viewpoints, which were added to the pose estimator's training set to improve generalization. In total, we generate 4193 images, and we refer to this dataset as D4. We then use D1, D2, and D4 for training, and D3 for validation and testing to test the improvement provided by using the NVS-based augmentation.

Additionally, it is noted that the images in D4 exhibited different brightness levels and background noise as compared to the original data, introduced during the NVS model reconstruction. To address the potential degradation due to this, we further augment the data by jittering the color of each image during training, thus making the pose estimator robust to minute color and lighting changes. For evaluation, we use the same GPU, framework, and hyperparameter tuning methods as described in the previous section.

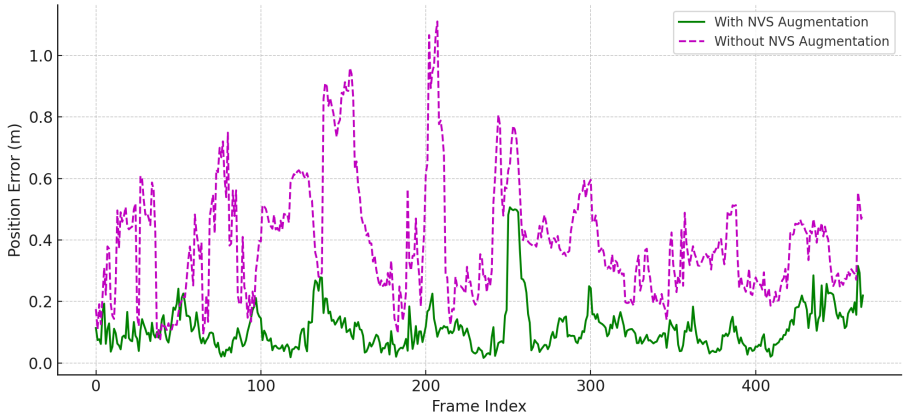
4.3.2 Results & discussion

Our results show that utilizing augmented training data generated by a NVS model leads to a significant enhancement in localization accuracy. Comparing row 2 and row 4 in Table 4.8, we find that by augmenting the training data with D4, the overall localization error can be reduced by 30%.

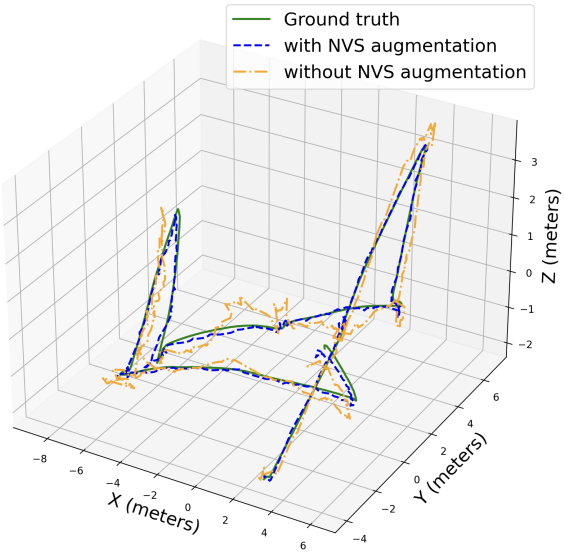
Color jittering augmentation is also highly effective in further improving the model performance, further reducing the error by an additional 61.5%. We compare the performance of the augmented training with color jittering with the performance without augmented training in Fig. 4.14 and Fig. 4.15. These plots show that the proposed augmented training with NVS significantly improves the pose estimator's accuracy and reliability in terms of both position and orientation.



(a) Orientation error over time.



(b) Positional error over time.



(c) 3D trajectory comparison between the ground truth and model predictions.

Figure 4.14: Comparison of pose estimation results with and without NVS-based training augmentation in a controlled environment.

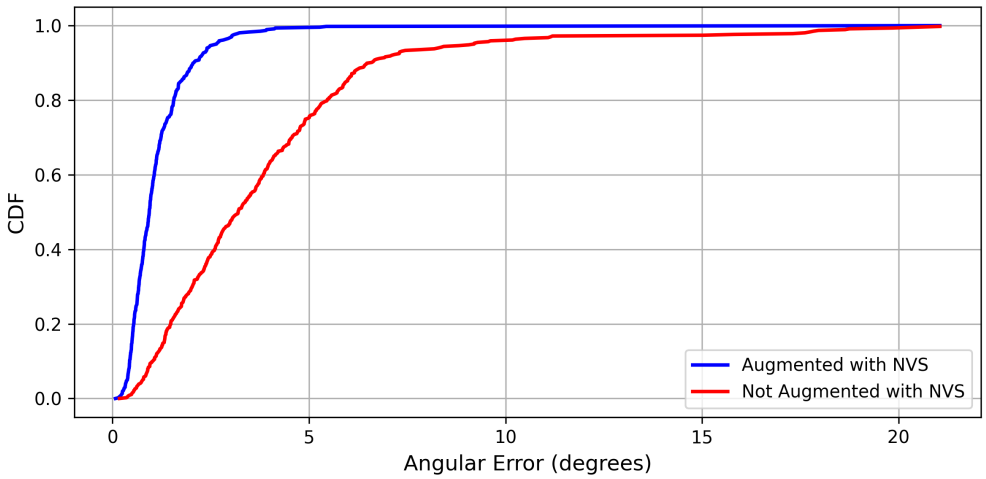
Nonetheless, we observed the presence of outliers. Upon examining the data, we found that these outliers were caused by transient objects, such as the tether shown in Fig. 4.16(b), which were not present in the training data.

4.4 Localization enhancement via sensor data fusion

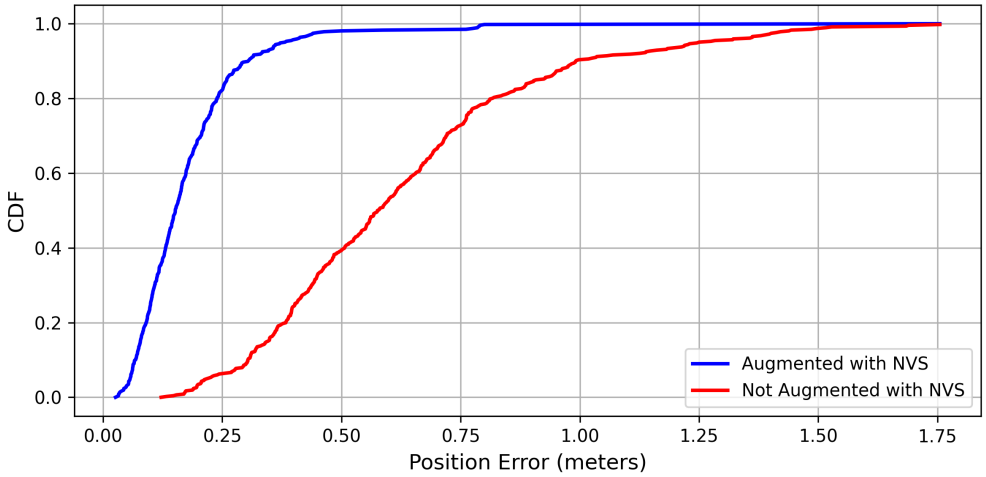
While the trained pose estimators yield small median orientation and position errors, their estimates exhibit some volatility. Our model currently treats each sample independently, ignoring temporal context, and utilizes only the camera inputs during deployment. However, additional information, such as temporal information and other sensor inputs from the ROV, is available. To enhance localization accuracy and achieve a more stable trajectory estimation, we propose sensor fusion using an EKF. This section details the integration of the pose estimator with additional sensor data and presents the results of the sensor fusion.

4.4.1 Methods

Given the sequential nature of data in reinspection missions and the availability of additional sensors, incorporating temporal information and other sensor data presents a viable strategy for improving the model's estimation stability and accuracy. Currently, the visual localization model without sensor fusion occasionally results in estimation of poses that are



(a) CDF of orientation errors.

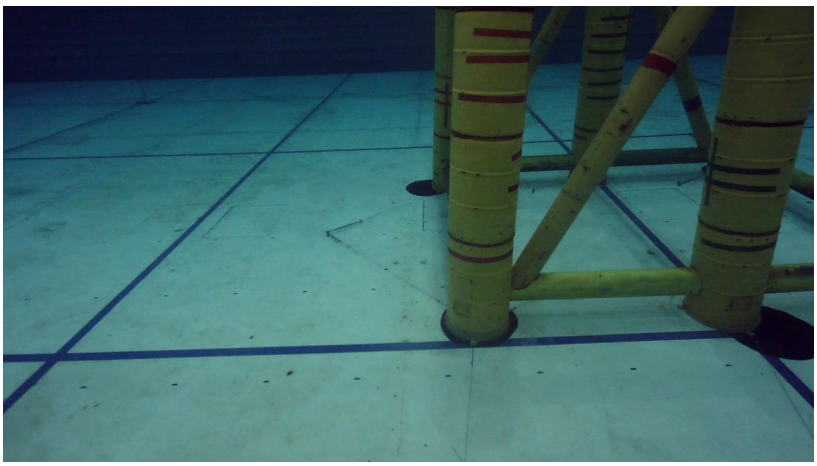


(b) CDF of position errors.

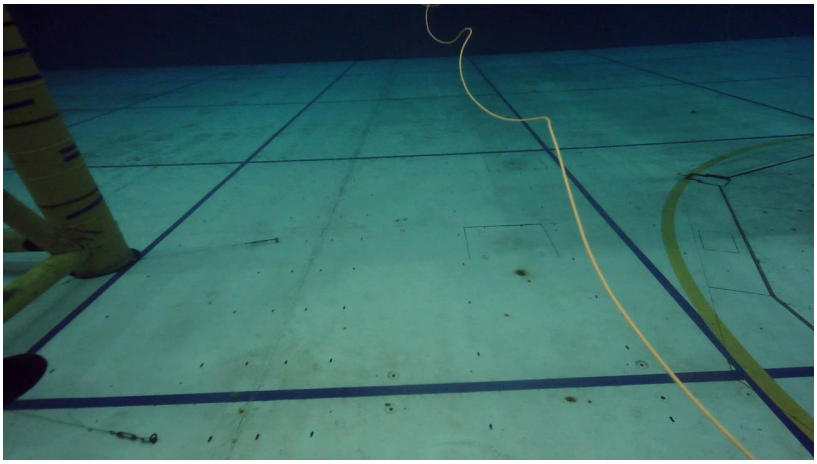
Figure 4.15: CDF of errors for models trained with and without NVS augmentation in controlled environment. The plots show that augmented training with NVS yields significantly lower errors for both orientation and position compared to training without augmentation.

physically implausible or outliers, in context of the dynamics from previous poses. By integrating knowledge of the ROV's physics model and leveraging previous pose estimates, we can enhance pose accuracy and stability.

Furthermore, during reinspection missions, ROVs are commonly equipped with depth sensors and compasses, which have a reasonable accuracy. As



(a) The test image with one of the lowest pose estimation errors.



(b) The test image with one of the highest pose estimation errors.

Figure 4.16: Test images from Clear Water-Mid with amongst the best and worst pose estimation accuracy.

such, we could use these reliable depth and orientation measurements during reinspection to further improve the overall localization accuracy.

We assume that the vehicle moves with a constant translational velocity and constant angular velocity since the vehicle normally moves slowly during inspection missions. Our EKF fuses measurements from three sources: the pose estimator (x, y, z position and orientation in quaternion form), compass (orientation), and depth sensor (z position). The filter maintains

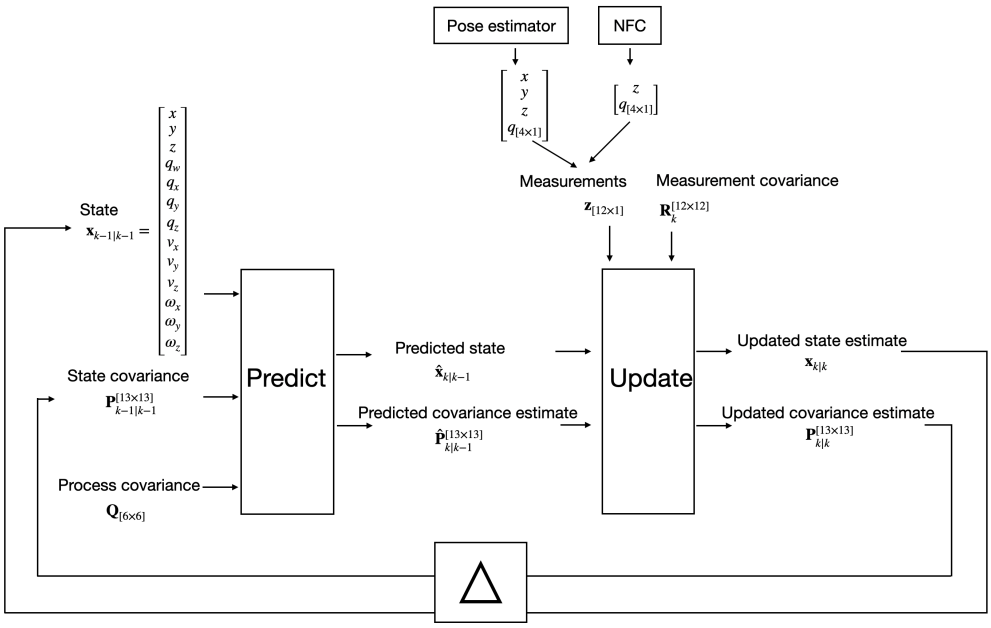


Figure 4.17: EKF schematic for sensor fusion.

a 13-dimensional state vector comprising position, velocity, quaternion orientation, and angular velocity. The structure of the EKF, including its iterative prediction-update loop, is illustrated in Fig. 4.17.

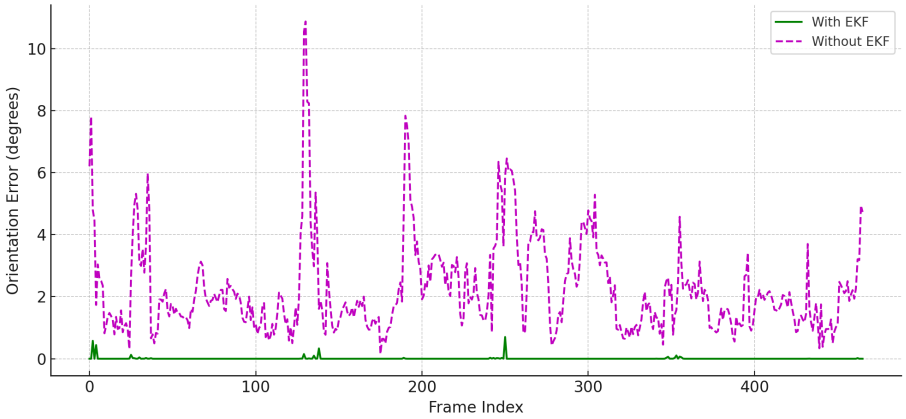
The EKF maintains and updates three covariance matrices: the state covariance $\mathbf{P}$, the process noise covariance $\mathbf{Q}$, and the measurement noise covariance $\mathbf{R}$. The state covariance $\mathbf{P} \in \mathbb{R}^{13\times13}$ reflects the uncertainty in the estimated state and is propagated and corrected at each timestep. The process noise covariance $\mathbf{Q} \in \mathbb{R}^{6\times6}$ is treated as a tunable hyperparameter and models uncertainty in the velocity and angular velocity components. The measurement noise covariance $\mathbf{R} \in \mathbb{R}^{12\times12}$ incorporates nominal noise levels from manufacturer specifications for the compass and depth sensor.

Characterization of the pose estimator's measurement noise requires a more involved process. The noise primarily stem from the fact that

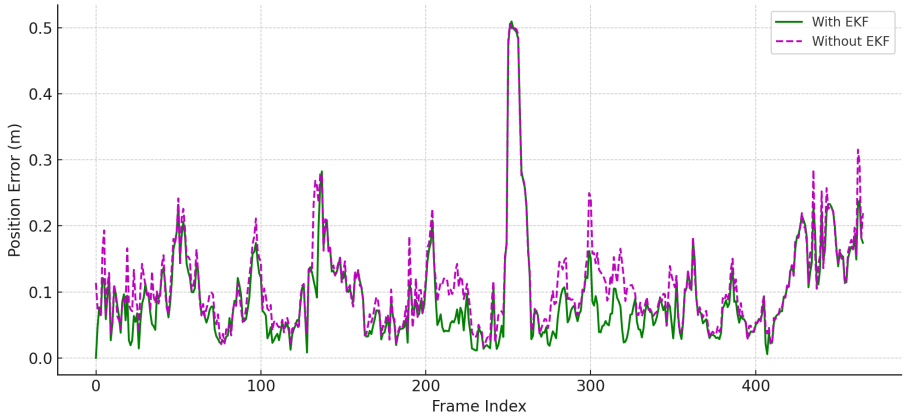
network estimations are inconsistent and can sometimes exhibit substantial errors. As such, setting a static value for the pose estimator's measurement noise, such as the standard deviation of localization error derived from validation performance, is inadequate. To more accurately represent the dynamic noise in the pose estimator, we employ dropout techniques at test time for Monte Carlo sampling from the model output's posterior distribution. Dropout is a technique commonly used as a regularizer in training neural networks to prevent overfitting. Recent works have shown that using dropout during inference can be used to approximate Bayesian inference over the distribution of the network's weights at test time, without requiring any additional model parameters [103]. Here, we apply Monte Carlo dropout at inference - specifically, we enable dropout in the second-to-last fully connected layer of the pose estimator using a dropout rate of 0.1. At test time, we perform 100 forward passes per image and compute the variance across pose predictions. This variance is then used to populate the relevant entries in $\mathbf{R}$, allowing the EKF to down-weight lower-confidence visual estimates and improve robustness in uncertain conditions. We did not observe any consistent bias in the compass, depth sensor measurements or pose estimator outputs during the controlled environment trials and thus did not model a bias term in the EKF formulation. As such, we assume the measurement noise is zero-mean and unbiased.

4.4.2 Results & discussion

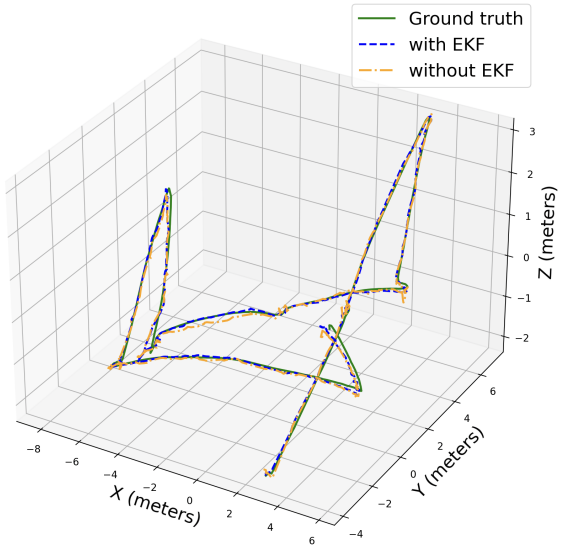
As shown in Table 4.8, sensor fusion with the EKF consistently improves pose estimation accuracy across different training setups. For configuration C6 trained on D1+D2 and tested on D3 (see rows 2 and 3), applying EKF reduces the median position error $\mathcal{L}_{\checkmark}$ from **0.58 m** to **0.47 m**, and the orientation error $\mathcal{L}_\theta$ from **3.20°** to **0.00°**. Similarly, for C6 trained with the NVS-augmented dataset (see rows 5 and 6), EKF reduces the position error from **0.15 m** to **0.11 m**, and orientation error from **0.93°** to **0.00°**. This consistent improvement demonstrates the robustness of the EKF-based fusion method in filtering noisy frame-level predictions and leveraging inertial priors. As also illustrated in Fig. 4.18, the estimated trajectory becomes noticeably smoother and more aligned with ground truth. While the inference time increases by approximately 10 times due to Monte Carlo sampling, this trade-off may be acceptable in scenarios where pose stability and accuracy are critical.



(a) Orientation error over time.



(b) Positional error over time.



(c) 3D trajectory comparison between the ground truth and model predictions.

Figure 4.18: Comparison of pose estimation results with and without EKF in a controlled environment.

4.5 Field trials at sea

To further validate our proposed methods, we conducted field trials in a bay near St. John's Island, Singapore (SJI). In this section, we present the methods, results and challenges encountered in using our proposed methods from the previous section in a real-world setting.

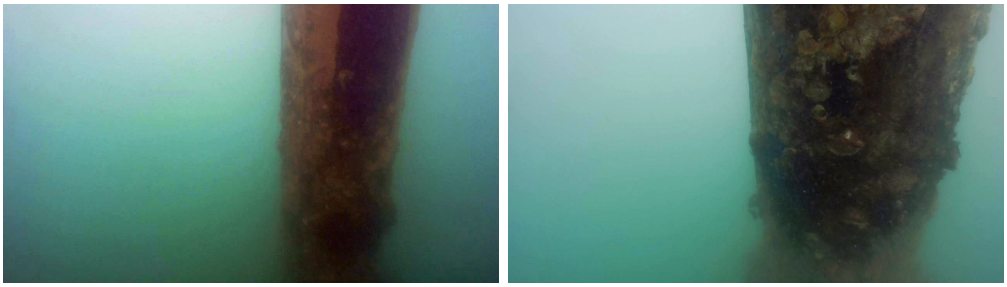
4.5.1 Methods

We used Hydra to collect data in the site described in Section. 3.3.3.1. We drove Hydra following a vertical lawnmower path around the pile, while recording the video from the camera. Due to the high turbidity in the water, we operated the ROV in close proximity to the structure with the average distance being 1 m.

The accuracy of USBL in our at-sea experiments was compromised due to high measurement noise and the absence of detailed information about the deployment geometry. In contrast, COLMAP was able to produce camera pose estimates with centimeter-level accuracy using structure-from-motion on the collected images. We therefore used COLMAP to estimate camera poses using the collected image data. Although these poses are not ground-truth in the absolute sense, they provide a consistent reference trajectory suitable for evaluating relative pose estimation performance in the field setting.

We collected two datasets, named as D5 and D6, on two different days. Although the inspection was carried out on the same structure with similar

(a) Sample images from the Sea Water-1 dataset.



(b) Sample images from the Sea Water-2 dataset.

Figure 4.19: Representative camera images from the two underwater datasets collected. The images from Sea Water-2 dataset show higher turbidity and thus fewer observable features than images from Sea Water-1.

trajectories, there were noticeable differences in environmental conditions between the two runs. D6 was collected under higher turbidity compared to D5, resulting in fewer visual features and noisier images. This variability reflects typical challenges encountered in real-world underwater inspections, where it is difficult to guarantee the same visibility, lighting, or exact path between mapping and reinspection runs. Samples of images collected in these datasets are shown in Fig. 4.19.

We use D5 to train an NVS model following the method described in Section 4.3.1. New camera poses are generated using the same approach. The NVS model is then utilized to create an augmented training dataset, named D7. Samples of images generated at new poses using the NVS model

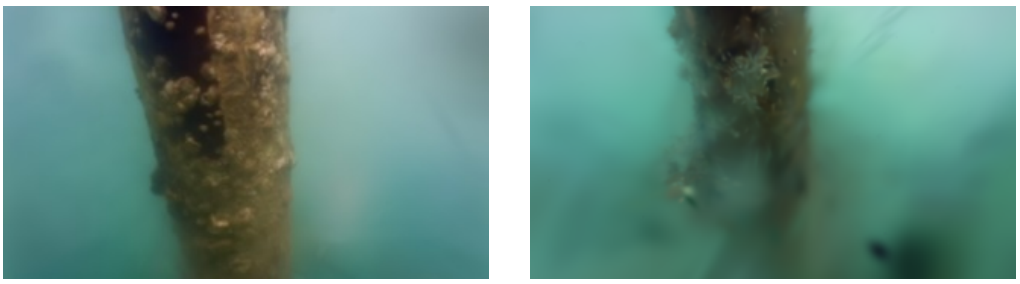


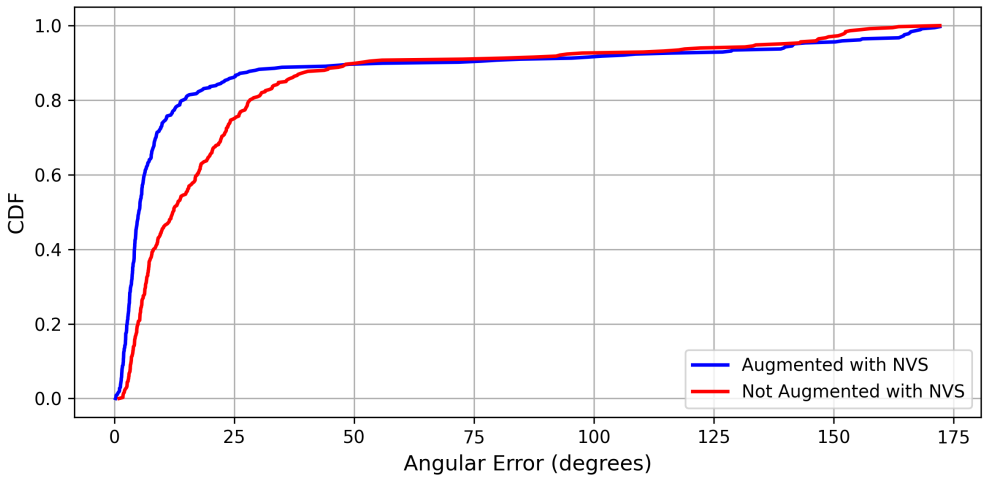
Figure 4.20: NVS rendered images for scenes in the bay near SJI. The rendered images produce photorealistic views of the structure but exhibit some artifacts and noise depending on the camera pose.

are shown in Fig. 4.20.

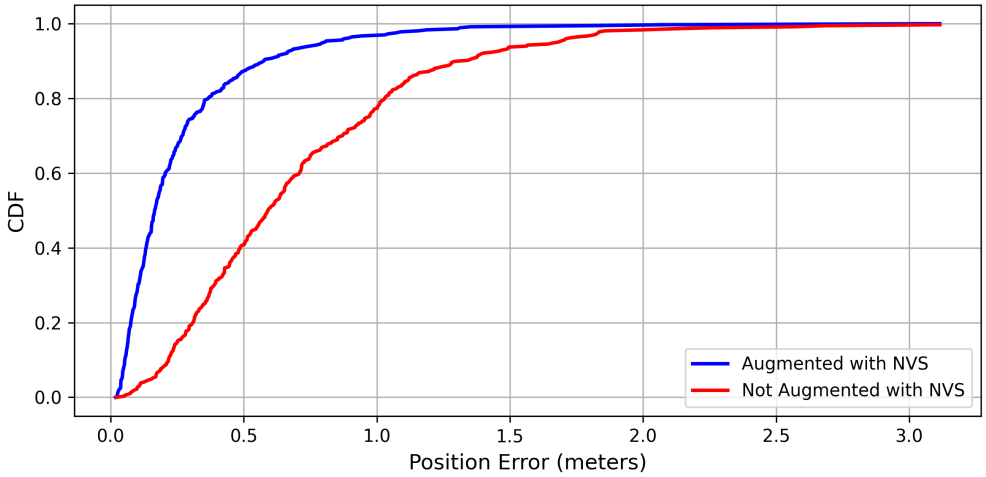
We train the best visual localization architecture configuration, C6, both with augmented training data (datasets D5+D7) and without any augmentation (only D5). Dataset D6 is used for validation and testing. The training methods are similar to those described in Section 4.3.1.

4.5.2 Results & discussion

As shown in Fig. 4.21, augmented training with NVS yields significant improvement in both position and orientation accuracy compared to training without NVS augmentation. The improvements brought by NVS can be attributed not only to an increase in training samples, but more importantly to the expanded coverage of viewpoints, especially those that may be under-represented or missing due to inevitable variations in inspection trajectories. This highlights the strength of NVS augmentation in realistic underwater applications, where achieving complete and repeatable scene coverage is inherently difficult. With configuration C6 and augmented training, we are able to achieve a position accuracy of 0.17 m and orientation accuracy



(a) CDF of orientation errors.



(b) CDF of position errors.

Figure 4.21: CDF of errors for models trained with and without NVS augmentation in at-sea environment. The plots show that augmented training with NVS yields significantly lower errors for both orientation and position compared to training without augmentation.

of 5.09°. We present the performance of C6 on D6 in Table 4.9. While the median accuracy is comparable to the performance in the controlled environment, we note that the standard deviation in the errors are much larger at sea.

Clearly, the real-world setting at sea presents several challenges that

Table 4.9: Performance of configuration C6 on dataset D6. $\mathcal{L}_{\mathbf{p}}$ and $\mathcal{L}_{\theta}$ are median values across the test data. $\mathcal{L}$ was calculated using Equation 4.3 with the average distance $d = 1$ m.

Training Dataset	**Color Jittering**	**Performance Metrics**		
		$\mathcal{L}$ (m)	$\mathcal{L}_{\mathbf{p}}$ (m)	$\mathcal{L}_{\theta}$ (°)
D5		0.80	0.59	12.15
D5+D7	✓	**0.26**	**0.17**	**5.09**

are not present in controlled environments. The biggest challenge is the turbidity of the water, which significantly affects the quality of the images. Moreover, lighting is inconsistent at different camera poses and on different days, causing high variablity in the image quality. This introduced three new challenges. First, the noisy images make it challenging to compute camera poses in COLMAP, resulting in a sparse number of registered images. Consequently, the EKF model could not be used for performance improvement since it would not be feasible to assume constant velocity and angular velocity in the vehicle model. Second, the turbidity and inconsistent lighting in the training data introduced artifacts in the NVS model. Thus, the rendered images are more noisy compared to images in clear waters, as shown in Fig. 4.20. Third, the high variability in image quality can lead to more estimation outliers and large errors during inference. As illustrated in Fig. 4.22, some test images contain rich textures and clear structure boundaries, which are favorable for accurate pose estimation. In contrast, others suffer from turbidity and challenging environment lighting, resulting in severely degraded visual features and consequently poor pose estimates. All of these factors contribute to a decrease in the model's performance.

(a) An image with one of the best pose estimation accuracy.

(b) An image with one of the worst pose estimation accuracy.

Figure 4.22: Test images from Sea Water-2 with amongst the best and worst pose estimation accuracy.

4.6 Summary

In this chapter, we addressed the challenge of localization in underwater inspection missions with a neural-network based pose estimator. We conducted preliminary feasibility studies of using neural-network based pose estimators for underwater localization using data collected from a simulator and a tank. Our results showed that such pose estimators were able to localize the vehicle accurately and their performance could be further improved when trained on a larger and more diverse dataset using deeper neural networks as the backbone.

To further improve the performance of the pose estimators, we proposed a new loss function to train the pose estimator, and demonstrated that training with d-loss significantly improved the model's performance in pose estimation tasks. This improvement was attributed to the incorporation of domain-specific physics, as the d-loss accounts for the relevant geometric considerations in the inspection mission. Furthermore, this loss function

also provides more interpretability. Employing the ResNet50 backbone with a CNN+LSTM architecture enabled us to efficiently use the available visual information to estimate the pose, and yielded improvements in the localization performance as compared to benchmark architectures.

In terms of the generalization, using more diverse data with a wider distribution significantly enhanced the localization performance on test data that lies outside the training distribution. We also investigated the use of NVS techniques to augment training data and showed that this significantly improved the estimator's performance with previously unsurveyed poses. Thus, we provide a cost-effective and information-efficient method to improve the generalization performance without having to undertake expensive field trials to collect additional data. Further integrating the pose estimator with an EKF allowed us to fuse sensor data with the visual-based estimates, and we demonstrated that this further improved the performance and stability. We validated our proposed methods in both controlled environments in a clear water ocean basin facility and in real-world settings at sea.

Overall, our results showed that our proposed methods significantly improved the visual localization performance in both controlled underwater environments and real-world settings and achieve good localization accuracy to within desired limits, providing a cost-effective alternative or complement to existing localization solutions. Real-world challenges such as turbidity and noise limited the performance achievable, but the proposed method still performed reasonably, especially when data augmentation using color-based

augmentation was used to robustify the technique against color distortion.

Chapter 5

Inverse NVS

In Chapter 3, we propose an NVS prior-based image compression approach. Our method leverages an NVS model and uses an optimal latent representation, such as camera pose, as the compressed form of an image. To account for dynamic changes in the scene, we compute the difference between the camera image and the corresponding rendered image and compress it using a classical lossy compression technique.

By transmitting the compressed difference along with the latent representation instead of the original RGB image captured by the ROV camera, we aim to achieve high compression ratios while preserving high image quality. This strategy enables real-time image transmission over bandwidth-limited acoustic links. The effectiveness of this approach, however, depends on having an accurate camera pose. When the pose is correct, the rendered image closely matches the camera view, resulting in a small difference image that primarily reflects the scene changes.

In practice, pose estimates—whether obtained via neural networks or localization sensors—inevitably contain errors. Even slight misalignment

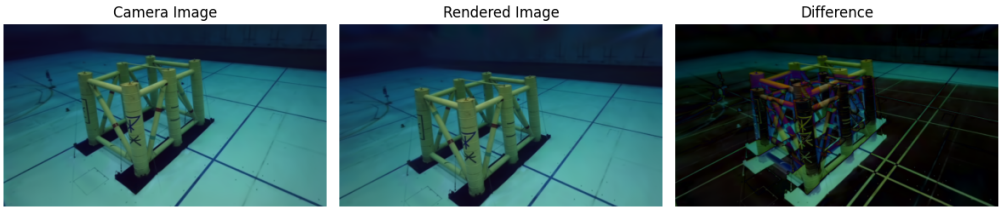


Figure 5.1: Effect of a 5° rotation error on the rendered image. The left image is the camera image, the middle image is the image rendered at the latent representation rotated by 5° about the x-axis, and the right image is the difference between the two images.

of a few pixels between the rendered and actual images can substantially increase the size of the compressed difference image, I_{diff}. While the pose estimators proposed in Chapter 4 provide sufficient accuracy for navigation, they often fall short for image compression. As illustrated in Fig. 5.1, small errors in the latent representation can lead to significant visual discrepancies, especially when the ROV is close to the object of interest.

To address this issue, we propose two techniques to minimize the difference image: (1) an affine transformation to refine image alignment, and (2) an inverse novel view synthesis method, dubbed *iNVS*, which searches for the optimal latent representation that produces the closest match between the model-rendered and actual camera images. The work in this chapter was presented in [77] and [78].

5.1 Minimizing difference image by affine transform

One effective strategy is to represent the minor differences between the rendered and actual image using a simple affine transformation. This

method helps correct small misalignment, thereby reducing the size of I_{diff}.

5.1.1 Method

An affine transformation is a geometric transformation that preserves collinearity and parallelism while allowing for scaling, rotation, shearing, and translation. Given a camera image, I_{camera}, and a NVS-rendered image, I_{NVS}, we define a transformation that maps coordinates (x, y) in I_{NVS} to new coordinates (x', y') in I as:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} e \\ f \end{bmatrix}$$

where (x, y) are pixel coordinates in I_{NVS}, and (x', y') are the corresponding transformed coordinates in I_{camera}. The parameters a, b, c, d define the linear transformation (scaling, rotation, and shear), while e, f define the translation.

To determine the optimal affine transformation parameters, we first extract keypoints from both the rendered and camera images using the Harris corner detector [104]. These keypoints are then described using FREAK descriptors [105] and matched to establish correspondences between the two images. To ensure robustness to outliers, we apply a RANSAC-based algorithm [106] that iteratively samples minimal sets of correspondences, estimates candidate affine transformations, and selects the one with the highest number of inliers. The final affine parameters are refined by minimizing alignment error over the inlier set.

Once the affine transformation is determined, it is applied to the reference image I_{NVS} to obtain a warped image $I_{\text{NVS}}{}'$. The difference image is then computed as:

$$I_{\text{diff}} = I_{\text{camera}} - I_{\text{NVS}}{}', \tag{5.1}$$

where $I_{\text{NVS}}{}'$ is expected to be better aligned with I, leading to a smaller difference and improved compression performance.

In cases where insufficient keypoint matches are found or the estimated transformation is ill-conditioned, the transformation is reverted to the identity matrix, ensuring stability in the compression process. Additionally, if the condition number of the transformation matrix exceeds a predefined threshold, it is considered unstable and discarded to ensure compression reliability.

5.1.2 Experiments & results

We collected data from the TCOMS DOB, conducting two survey runs where the ROV followed a similar lawnmower trajectory in each trial. These trials are referred to as the mapping run (M1) and test run (T1).

As introduced in Chapter 3, we used a *Splatfacto* model [107], a 3D-GS implementation provided by the *nerfstudio* framework [84], for training and image rendering.

We trained the neural network-based latent representation estimator, referred to as CNN+LSTM, using the approach described in Chapter 4. We utilize COLMAP [102] to compute the latent representation associated with

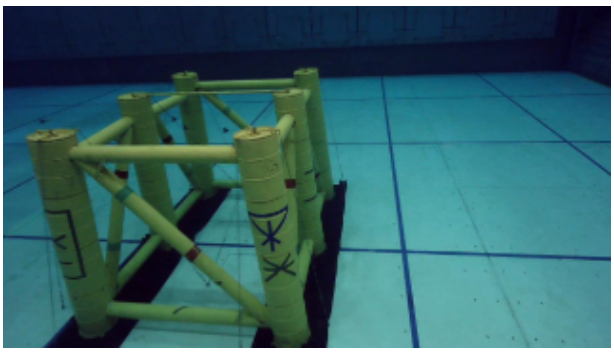


Figure 5.2: Original camera image.

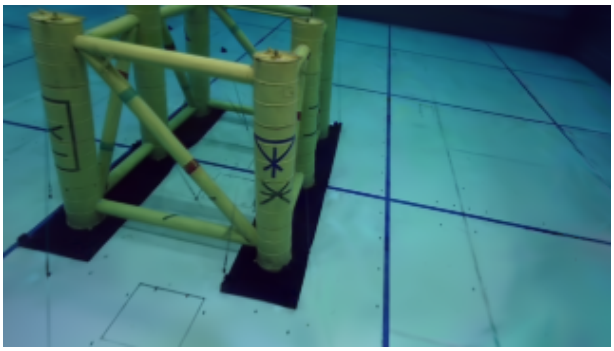


Figure 5.3: Rendered image using the estimated pose.

each image within an arbitrary reference coordinate system. The computed COLMAP representation and corresponding images serve as the training data for CNN+LSTM.

For evaluation, we used the trained CNN+LSTM to estimate the camera poses of the selected images from T1. These estimated poses were then fed into the trained Splatfacto model to generate corresponding rendered images. Fig. 5.2 shows an example of an original camera image from T1, and Fig. 5.3 presents the corresponding rendered image generated using the estimated pose.

To align the rendered image with the original camera image, we computed and applied an affine transformation. The difference between the

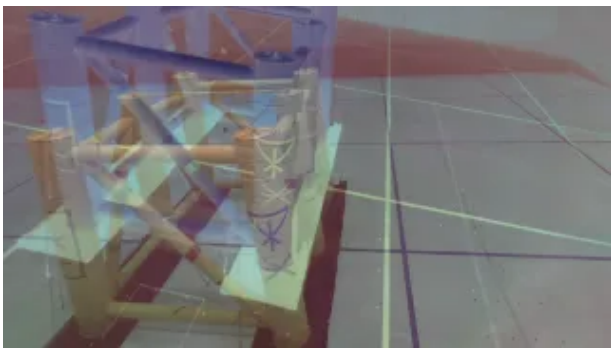


Figure 5.4: Difference image between the original camera image and the rendered image.

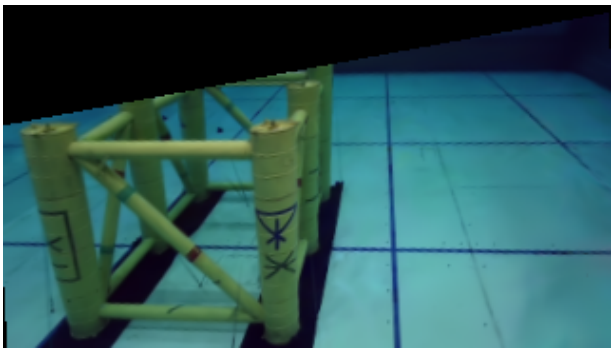


Figure 5.5: Affine transformed image.

original camera image and the rendered image before applying the affine transformation is shown in Fig. 5.4. After applying the transformation, we obtained the adjusted image shown in Fig. 5.5, and the corresponding difference image is depicted in Fig. 5.6. These visual comparisons illustrate the impact of affine transformation in reducing misalignment qualitatively.

To quantitatively evaluate the effectiveness of this alignment, we measured the size of the difference image I_{diff} compressed using the WebP lossy image compression format. We performed this evaluation over 1,000 image pairs. Without affine transformation, the average size was 4905 bytes. With affine transformation, the average size was reduced to 4164 bytes including the bytes needed to represent the pose and the transformation parameters.

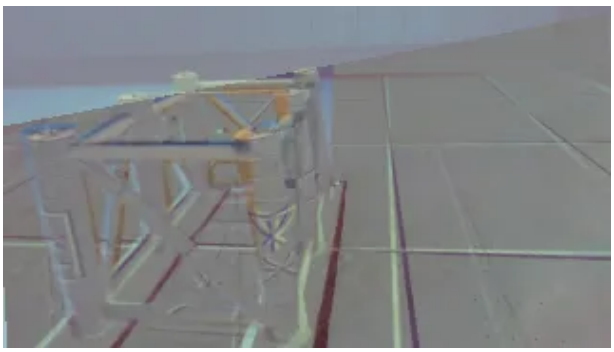


Figure 5.6: Difference image of the original camera image and the affine-transformed image.

These results confirm that applying affine alignment leads to a substantial improvement in compression performance.

5.2 Inverse NVS for optimal latent representation

Although affine transformation of the rendered view effectively reduces the size of the difference image, it is not an ideal solution, as the affine transform implicitly assumes a 2D world. The model mismatch manifests in terms of artifacts in the difference image (see Fig. 5.5), thus increasing the data size to be transmitted. Moreover, with the presence of novel objects that were not present during the mapping run, the affine transform technique may not be able to find sufficient matches between the camera image and rendered image, thus compromising its robustness.

In this section, we propose to optimize the pose being transmitted, via gradient descent through trained NVS models in order to minimize the difference image. We refer to this method as iNVS. We conduct an in-depth examination of various loss functions, optimization algorithms,

and initialization methods.

We rigorously evaluate the performance of our proposed technique in a confined underwater environment. The results demonstrate that NVS-based image compression outperforms existing image compression methods in both compression ratio and image quality. We assess the robustness of our method when encountering novel objects and occlusions within the scene from structures, both large and small. Our findings reveal that our method can effectively handle these cases, maintaining high image quality and reliability in dynamic underwater settings.

5.2.1 Methods

iNVS aims to rapidly estimate the latent representation that minimizes the difference between a real camera image and the image rendered by a trained 3D-GS model. The steps involved in iNVS are detailed in 5.7,and its key components—initialization strategy, optimization algorithm, and objective function—are discussed below.

An effective initialization is crucial for the rapid convergence of optimization algorithms. In inspection missions where the vehicle moves slowly and steadily, there are minimal changes in the latent representation between consecutive frames. Assuming such a scenario, we use the optimized latent representation from the previous frame as an initialization point for estimating the latent representation in the current frame, provided that a "good" previous frame exists. This approach leverages the small inter-frame

variations in the latent representation, enabling faster convergence due to the proximity of the initial estimate to the true latent representation. By utilizing the optimized latent representation from the previous frame whenever possible, we mitigate the issues associated with sensor drift, biases, and estimator noise, providing a more accurate starting point for optimization.

To determine whether a previous frame is "good", we compare the rendered image at that latent representation with the current camera image. If their difference falls below a predefined threshold, the estimate is reused.

When a "good" previous frame is unavailable, such as at the start of a mission, or when the difference exceeds the threshold, alternative initialization sources are employed. These may include measurements from vehicle sensors or estimates from learned latent estimators.

The optimization step in iNVS refines the initial latent representation by minimizing a differentiable image similarity loss between the rendered and observed images. Since the number of parameters to be optimized is small (typically a 6-DoF pose), both deterministic and stochastic optimization methods can be considered. In our implementation, we explore both a quasi-Newton method and a stochastic gradient-based method, as described in Section 5.2.2.

The objective function quantifies the discrepancy between the rendered image and the camera image. A commonly used choice is the mean squared error (MSE), defined as:

$$L_{\text{mse}} = \frac{1}{N}\sum_{i=1}^{N}\|I_{\text{camera}}(i) - I_{\text{rendered}}(i)\|_2^2 \tag{5.2}$$

where $I_{\text{camera}}(i)$ and $I_{\text{rendered}}(i)$ are the RGB vectors at the i^{th} pixel for the camera image and the rendered+affine-transformed image, respectively. $\|\cdot\|_2$ denotes the Euclidean (L2) norm across the color channels, and N is the total number of pixels.

Alternative loss functions, such as the keypoint-based matching loss, may improve robustness under poor initialization [70]. These are evaluated in Section 5.2.3.

We describe the algorithm of iNVS in Algorithm. 1.

Algorithm 1 iNVS: Inverse Novel View Synthesis for Latent Optimization

Require: Camera image I_{cam}, previous latent z_{prev} (optional), external initialization z_{ext}, NVS model M, threshold τ
Ensure: Optimized latent representation z^*
if z_{prev} exists and $\text{MSE}(I_{\text{cam}}, M(z_{\text{prev}})) < \tau$ **then**
    $z_0 \leftarrow z_{\text{prev}}$
else
    $z_0 \leftarrow z_{\text{ext}}$ ▷ Fallback to external source (e.g., sensor or estimator)
end if
Initialize $z \leftarrow z_0$
repeat
    $I_{\text{render}} \leftarrow M(z)$
    $\mathcal{L} \leftarrow \text{MSE}(I_{\text{cam}}, I_{\text{render}})$
    Update z to minimize $\mathcal{L}$ using a gradient-based optimizer
until convergence criteria are met
return $z^* \leftarrow z$

5.2.1.1 Datasets

We used the same dataset as the previous section, M1 and T1. On top of these two datasets, we collected another test run, where the ROV surveyed

the structure using a similar trajectory. We refer to this trial as test run 2 (T2). In T2, a new metallic structure was placed next to the existing one to test the robustness of our technique towards novel objects in the scene, as shown in Fig. 5.8c. We used T2 to investigate the robustness of our proposed methods towards novel objects. We selected and pre-processed the images in the same way as described in the previous section.

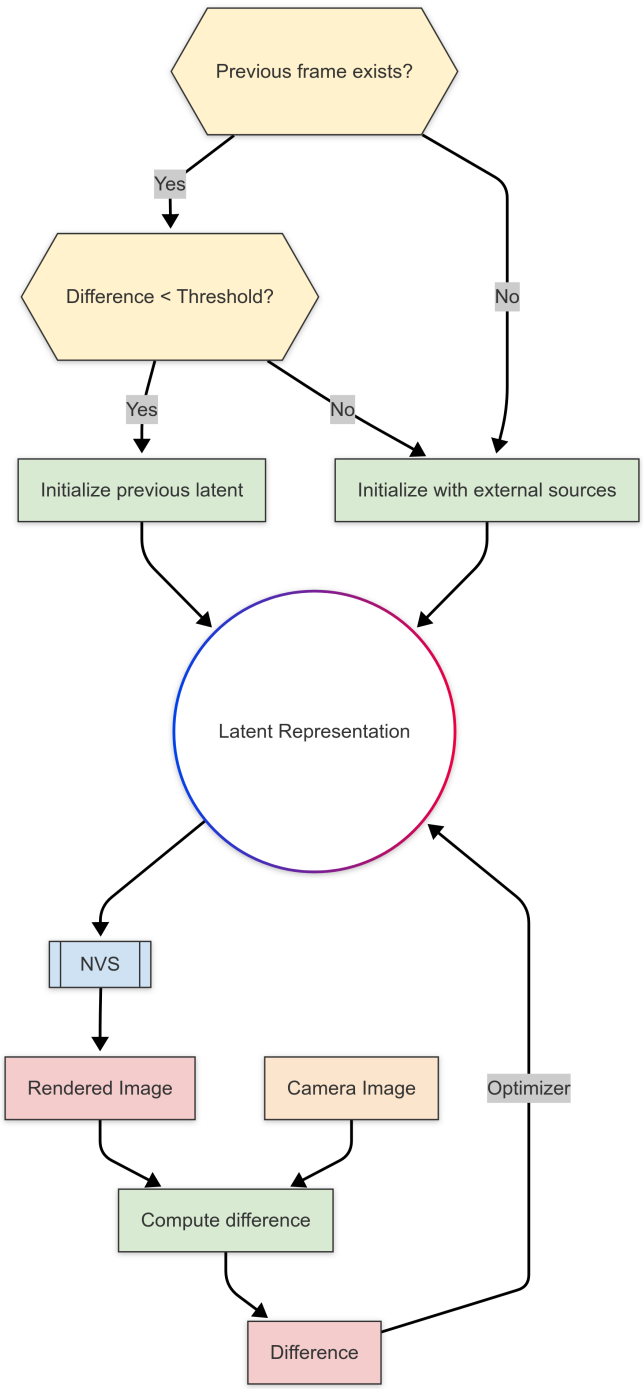
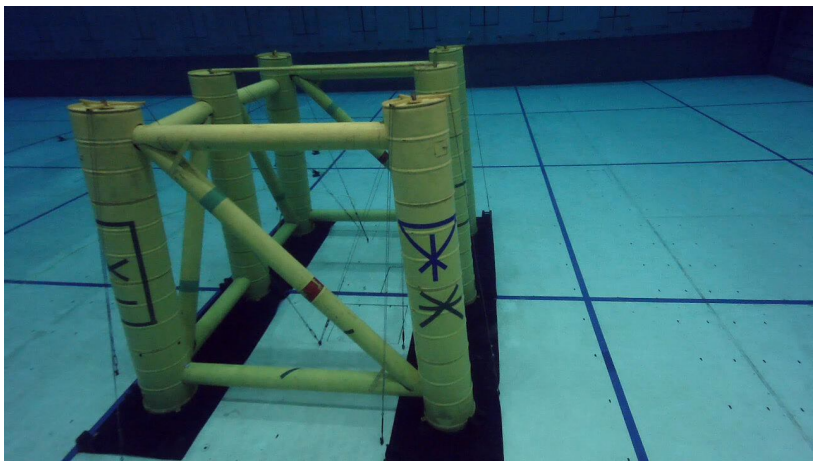
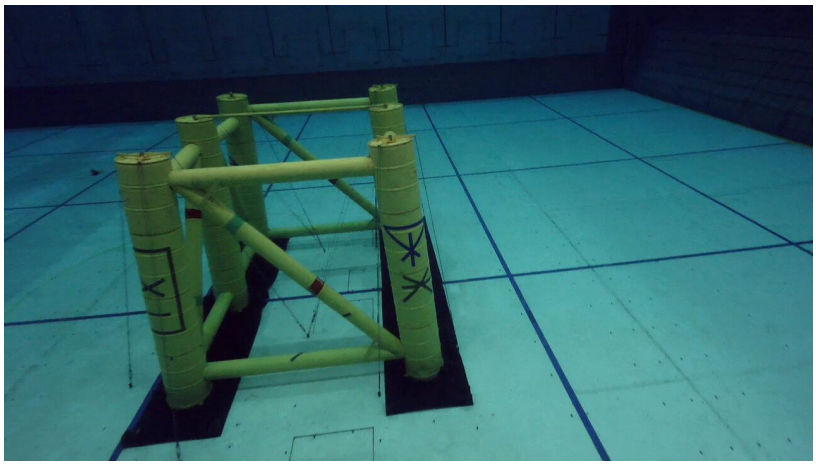


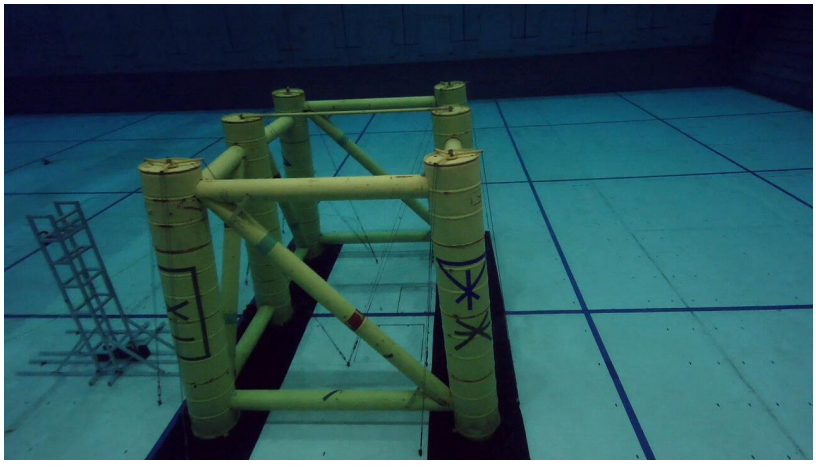
Figure 5.7: Flow diagram of the iNVS optimization process. The system determines whether to initialize the latent representation using the previous frame or external sources based on availability and difference threshold. The initialized representation is iteratively refined by minimizing the difference between the rendered and camera images using a pretrained NVS model. The resulting optimized latent representation is transmitted along with the residual image as a compressed representation of the image, enabling reconstruction at the topside.



(a) An image from the mapping run (M1), showing the ROV surveying the original structure.



(b) An image from test run 1 (T1), where the ROV continues to survey the same structure for performance evaluation.



(c) An image from test run 2 (T2), featuring an additional metallic structure placed next to the original to test the robustness of our technique towards novel objects in the scene

Figure 5.8: Example images from the datasets.

5.2.2 Implementation details

5.2.2.1 iNVS configuration

Scene representation We adopt 3D-GS as the underlying scene representation due to its high rendering speed, which enables real-time optimization. To train the 3D-GS model, we use camera images from the initial survey run (the "mapping run") and employ the *Splatfacto* pipeline [107] provided by the *Nerfstudio* framework [84].

Pose initialization For the controlled dataset, we use PoseLSTM, a neural network estimator trained to estimate 6-DoF camera poses from single RGB images. PoseLSTM is composed of a pretrained ResNet-50 [108] as a feature extractor and a bidirectional LSTM for dimensionality reduction. We generate ground-truth poses for training using *COLMAP* [102]. We match the generated ground-truth poses with corresponding images from the mapping run to form the training dataset.

To determine whether the estimated latent representation is suitable for initializing the next frame, we evaluate the MSE between the normalized camera image and the normalized NVS-rendered image at the optimized latent. We set a threshold of 1×10^{-3} for the controlled dataset.

Objective functions We evaluate two loss functions for pose refinement. The first is the MSE between the rendered and camera images, defined in Equation 5.2, which provides a direct pixel-level comparison

and is computationally efficient. The second is a keypoint-based matching loss Equation 5.3 introduced by iComMa [70], which uses LoFTR [109] to extract and match keypoints across the two images. This approach aims to improve robustness under poor initialization by focusing on larger-scale structurally meaningful features rather than pixel-level alignment. To balance computational cost and robustness, we limit the number of keypoints to 20.

The matching loss is defined as:

$$L_{\text{match}} = \frac{1}{M} \sum_{i=1}^{M} \|\mathbf{k}_{\text{camera}}(i) - \mathbf{k}_{\text{rendered}}(i)\|^2 \tag{5.3}$$

where $\mathbf{k}_{\text{camera}}(i)$ and $\mathbf{k}_{\text{rendered}}(i)$ denote the i^{th} keypoints in the camera image and the rendered image, respectively. Each keypoint $\mathbf{k}(i) \in \mathbb{R}^2$ is a 2D coordinate representing a salient point in the image. The loss computes the mean squared Euclidean distance between corresponding keypoints across the two images, serving as a measure of geometric alignment between the rendered image and the observed camera view.

Optimization methods For refining the latent representation, we evaluate both deterministic and stochastic optimization methods. We select the Broyden–Fletcher–Goldfarb–Shanno [110, 111, 112, 113] (BFGS) algorithm as the deterministic optimizer. BFGS is a quasi-Newton method that approximates the Hessian matrix using gradient evaluations and iteratively updates parameters to converge towards a local minimum. This choice is motivated by the deterministic nature of our problem, which involves a

small number of parameters. Since iNVS starts with a reasonably good initialization, the solution is often near the global minimum, allowing BFGS to converge quickly and reliably. We implement the BFGS algorithm using the *pytorch-minimize* package [114]. We set the gradient tolerance to 1×10^{-5} and the parameter tolerance to 1×10^{-6} to ensure convergence.

We also implement Adam [76] as a stochastic optimizer using *PyTorch*, with an initial learning rate of 1×10^{-3}, halving it if no loss improvement is observed over three epochs. We compare these configurations in Section 5.2.3.

Difference image compression The difference image I_{diff} between the camera and rendered views is compressed using WebP and JPEG-XL. We select these formats due to their support for fast encoding and decoding, and their suitability for low-bitrate transmission scenarios.

5.2.2.2 Runtime and Computational Setup

Training of the NVSPrior model was conducted on a high-performance workstation equipped with an NVIDIA RTX 6000 Ada GPU using PyTorch for model optimization. Each scene required approximately 14 hours of training at full resolution. The experiments presented in this chapter were evaluated on the same workstation at a resolution downsampled by a factor of 4 for faster iteration. To assess deployability under embedded constraints, inference and optimization were also tested on an onboard NVIDIA Jetson Orin module. The onboard runtime was approximately three times longer

per frame compared to the workstation, reflecting the trade-off between computational efficiency and real-time feasibility on power-limited ROV platforms.

5.2.2.3 Benchmarking methods

We compare against two families of baselines: (i) classical codecs—WebP and JPEG-XL, and (ii) learned codecs implemented with *CompressAI* [115], namely, the Mean & Scale Hyperprior method [25] and MLIC++ [27]. For classical codecs, we compress each RGB frame after resizing to 320×180 pixels.

For the learned baselines, we train the model on resized RGB frames with the standard rate–distortion objective

$$\mathcal{L} = \mathcal{R} + \lambda\, 255^2\, \mathcal{D},$$

where $\mathcal{R}$ denotes bitrate and $\mathcal{D}$ is the MSE computed on images normalized to $[0, 1]$; the 255^2 factor follows CompressAI's convention. A larger λ emphasizes minimization of distortion, yielding higher reconstruction quality. We report the highest-quality operating points using the largest λ provided by each implementation.

We also benchmark against NVSPrior+Affine, an adaptation of the approach proposed by Mishra et al. [77], which uses a learned affine transformation to align the rendered image with the camera image. In this method, we first estimate the latent representation using PoseLSTM and render the image using the 3D-GS model. We then compute affine transformation

parameters to align the rendered image with the camera view. The resulting difference image is compressed using either WebP or JPEG-XL. This approach serves as a baseline to assess the benefits of our proposed iNVS method, which directly refines the latent representation to minimize the difference image without relying on affine warping.

5.2.3 Ablation studies

We evaluate different loss functions and optimization methods for iNVS to identify the component algorithms that deliver the best compression performance for our proposed NVSPrior framework.

5.2.3.1 Objective functions

In this subsection, we study the performance of iNVS with different objective functions, and how it varies with the error in the initial latent representation. We introduce perturbations to the latent representation to simulate initialization errors that occur during actual survey runs. For each perturbation, we randomly select either a translation or rotation axis and sample a perturbation value from a uniform distribution within specified ranges. Translation perturbations are sampled uniformly within [-1.58 m,1.58 m], and rotation perturbations are sampled uniformly within [-40°, 40°]. Using the perturbed latent representation as the initialization, we optimize the latent representation using either MSE loss or matching loss until convergence.

After optimization, we compute the PSNR of the rendered images

compared to the camera images in dB, energy in I_{diff} and number of bytes of compressed I_{diff} between the camera image and rendered image at the optimized latent representation. These metrics jointly capture both visual fidelity and communication cost: PSNR reflects reconstruction quality, energy measures the magnitude of residual errors, and compressed size relates directly to the data rate, which is the ultimate focus of this work. We also record the number of iterations required for convergence. This process is repeated on the 1,422 images from M1. The results are shown in Fig. 5.9 and Fig. 5.10. The PSNR is defined as:

$$\text{PSNR} = 10 \cdot \log_{10}\left(\frac{1}{L_{\text{mse}}}\right) \tag{5.4}$$

where L_{mse} is defined in Equation 5.2.

The model trained using MSE loss outperforms that using matching loss across all metrics and perturbation values in terms of median performance, despite exhibiting higher variance. Overall, using MSE loss achieves a rendered image with higher PSNR and thus a more compressed I_{diff} than using matching loss. The larger variability in performance observed when using MSE loss is likely due to its sensitivity to initialization and the presence of more local minima in its loss landscape. Moreover, MSE loss converges faster than matching loss, even though it requires more iterations. As a result, we select MSE loss as the objective function for iNVS.

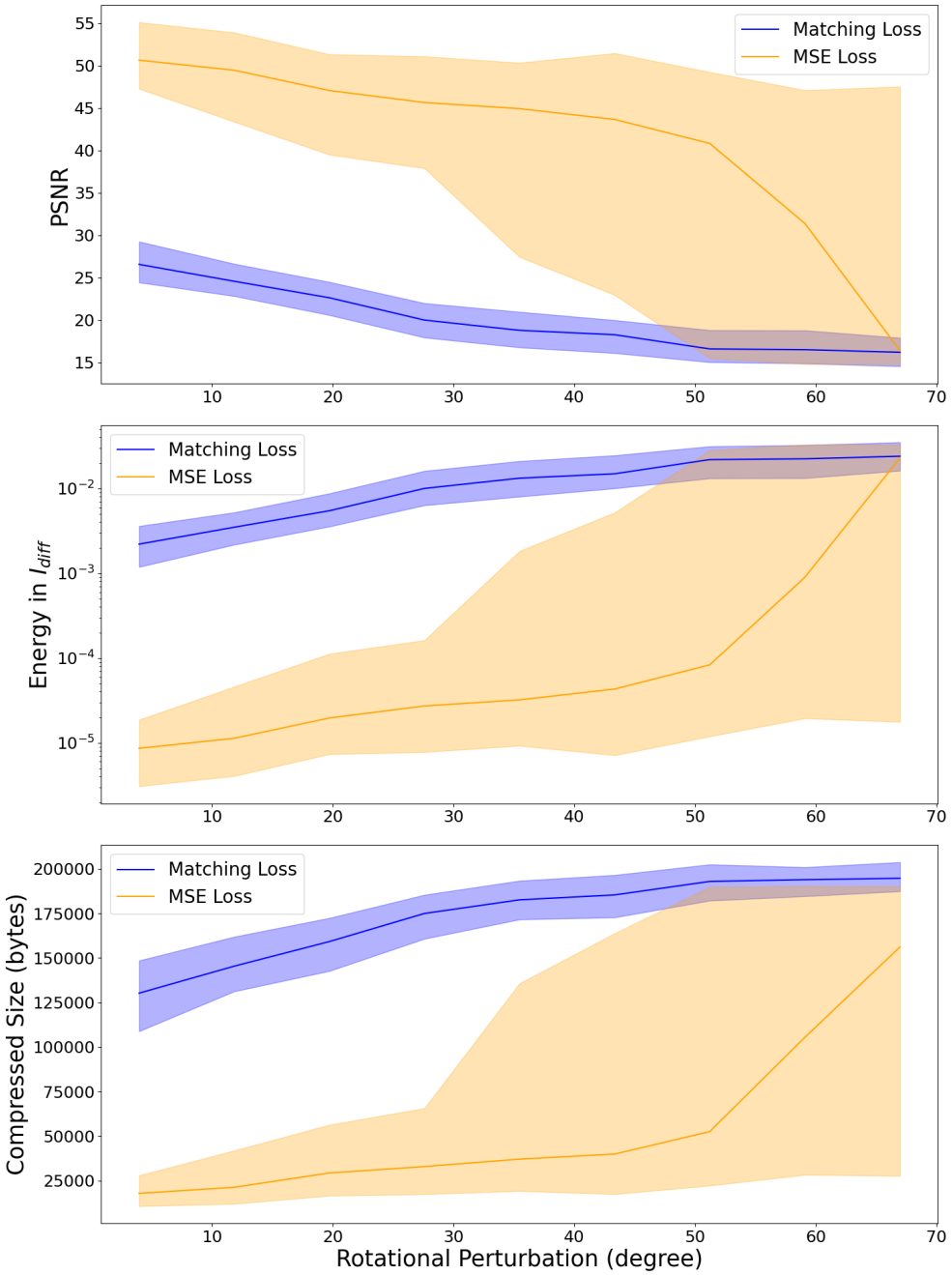


Figure 5.9: Comparison of performance using MSE loss and Matching Loss as objective functions across different levels of rotational initialization perturbation. In each ribbon plot, the solid line indicates the median value, while the shaded region denotes the interquartile range across samples. Metrics include PSNR, energy of the difference image, and compressed size.

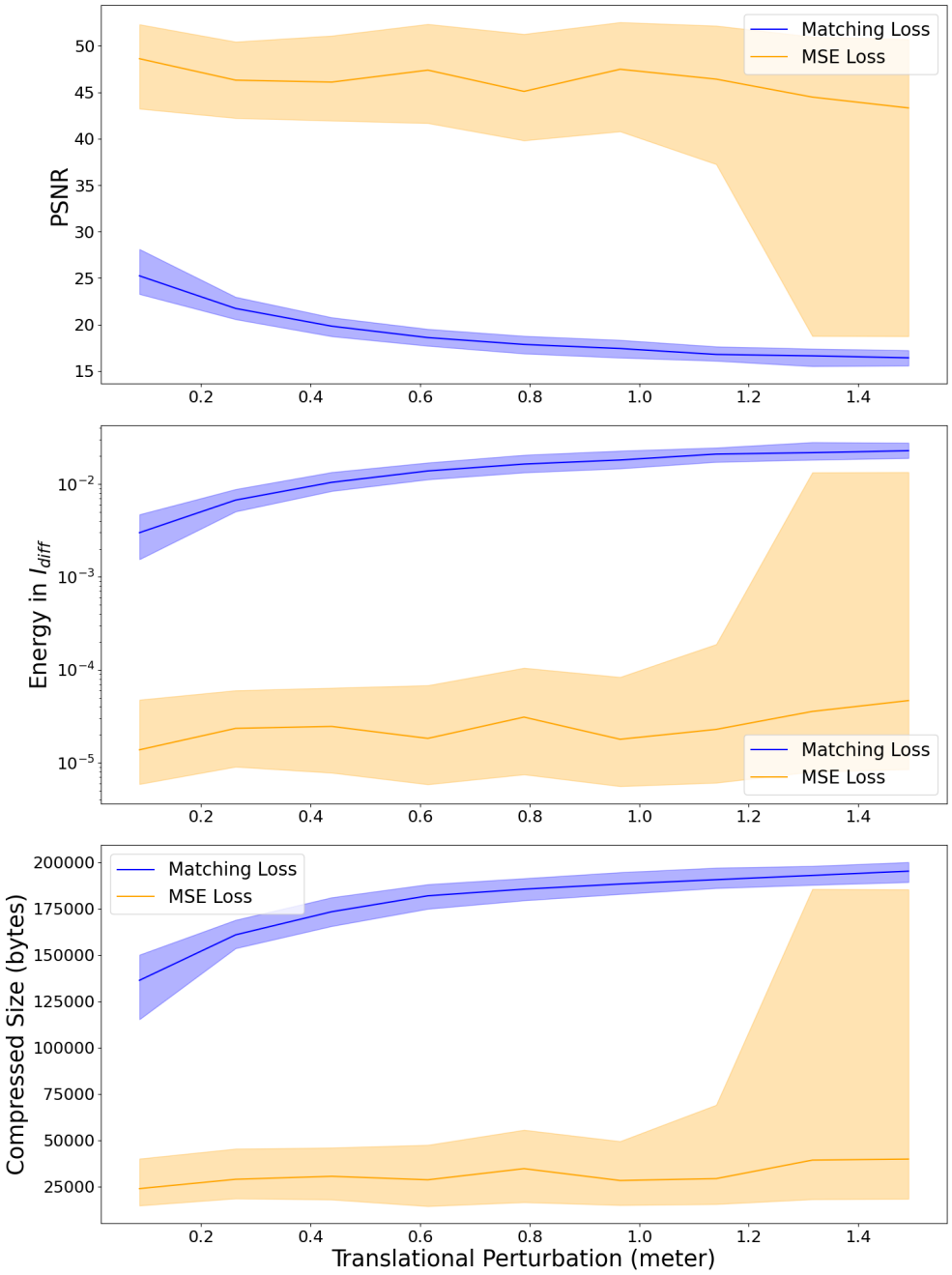


Figure 5.10: Comparison of performance using MSE loss and Matching Loss as objective functions across different levels of translational initialization perturbation.

5.2.3.2 Optimization methods

With the MSE loss as the objective function, we compare the performance of the BFGS and Adam optimization algorithms using a similar approach as described above. Additionally, we investigate the performance of a combined method, denoted as Adam+BFGS, where Adam is used as the optimizer until convergence, followed by BFGS for finetuning. This hybrid approach is a common practice in optimization [116, 117]. We present the results in Fig. 5.11 and Fig. 5.12. We find that both BFGS and Adam+BFGS outperform Adam across all three metrics and perturbation values. Adam+BFGS performs slightly better than BFGS at small translational perturbations; however, overall, BFGS achieves better performance than Adam+BFGS at higher perturbation levels. Adam+BFGS exhibits the smallest variance among all three optimization methods at high perturbations and hence better reliability.

Apart from exhibiting better median performance, BFGS also requires fewer iterations to converge than Adam and Adam+BFGS, and each of its iterations is faster. This is likely because BFGS, a second-order optimizer, leverages curvature information and progresses efficiently even when the initial pose is moderately inaccurate. In contrast, Adam, as a momentum-based first-order method, may converge more slowly or inconsistently due to noisy gradients, particularly in this low-dimensional setting. Although the hybrid Adam+BFGS method is more stable under small perturbations,

it adds computational overhead without consistent benefits at larger errors. Therefore, we choose BFGS as the optimization method for iNVS henceforth.

It is important to note that the performance of iNVS degrades rapidly when the perturbations are large (e.g., greater than 1.3 m in translation or 37° in rotation). Therefore, a good initialization is crucial for the optimal performance of iNVS.

5.2.3.3 Initialization

We compare the convergence performance of iNVS using two initialization methods: (1) the latent representation estimated from the previous frame and (2) the latent representation estimated by PoseLSTM on the current frame. Our results show that initializing with the previous frame's latent representation requires fewer iterations for convergence and is more computationally efficient, as it eliminates the need for neural network inference.

5.2.4 Compression performance on controlled dataset

We evaluate the compression performance of NVSPrior on the controlled dataset T1 using iNVS (shown in Fig. 5.13), configured with BFGS optimization and MSE loss. For both iNVS and the baselines, we test WebP and JPEG-XL as the codecs for compressing the difference image I_{diff}.

We compare against the following families of baselines: (1) WebP and JPEG-XL applied directly to the original RGB image captured by the ROV camera, (2) Mean & Scale Hyperprior and MLIC++ [25, 27], and (3)

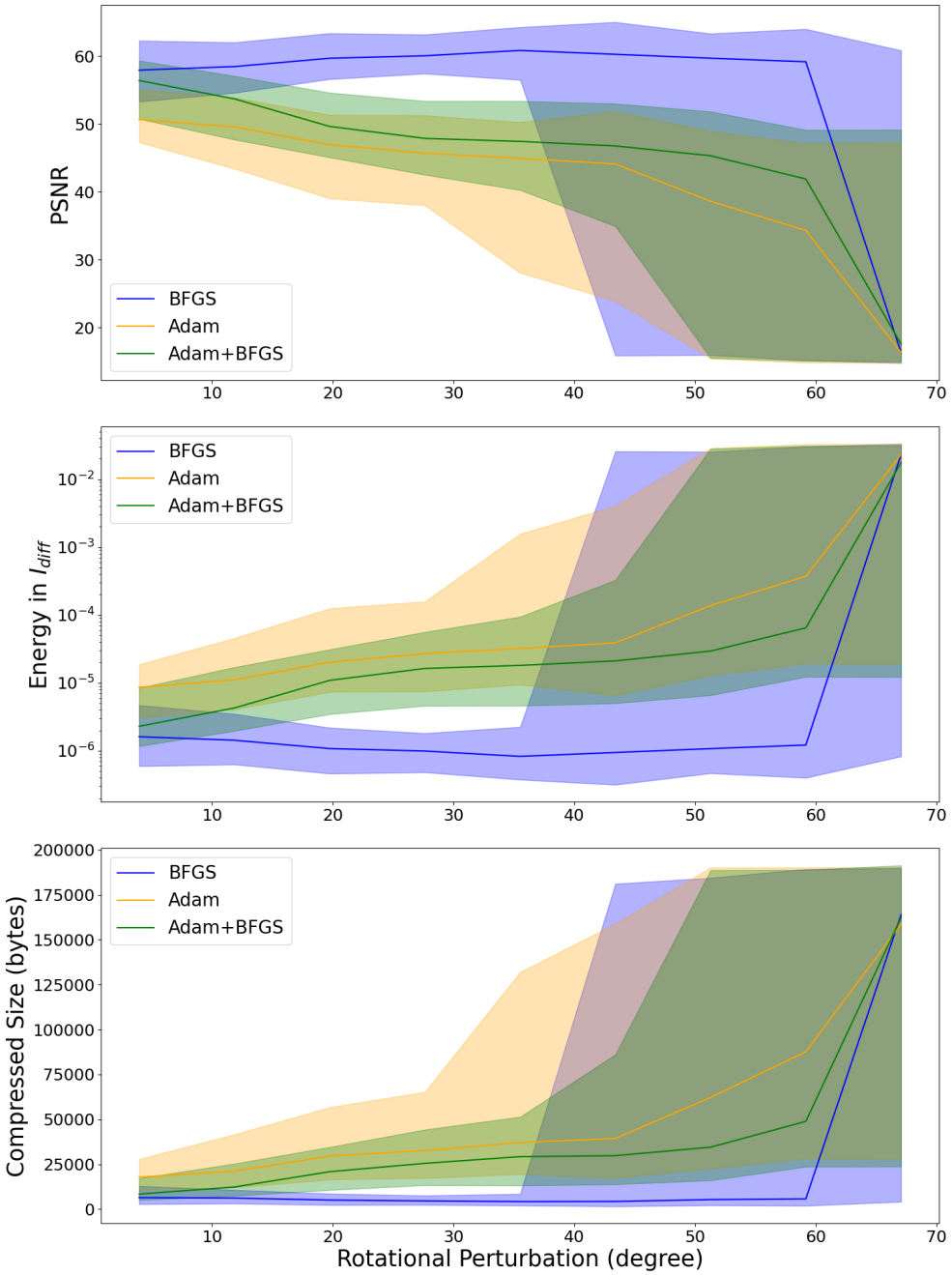


Figure 5.11: Comparison of performance using Adam and BFGS as optimizers across different levels of rotational initialization perturbation.

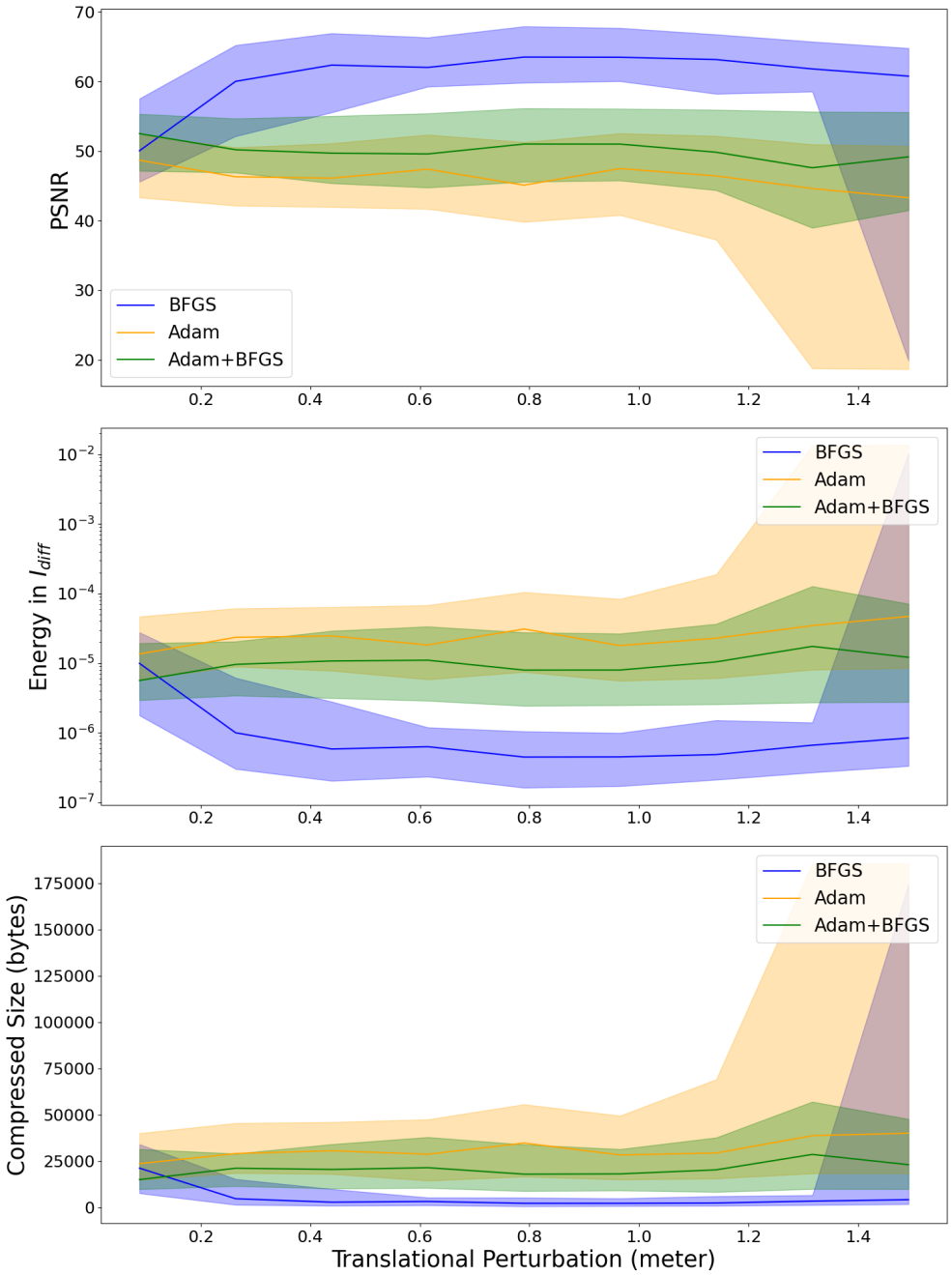


Figure 5.12: Comparison of performance using Adam and BFGS as optimizers across different levels of translational initialization perturbation.

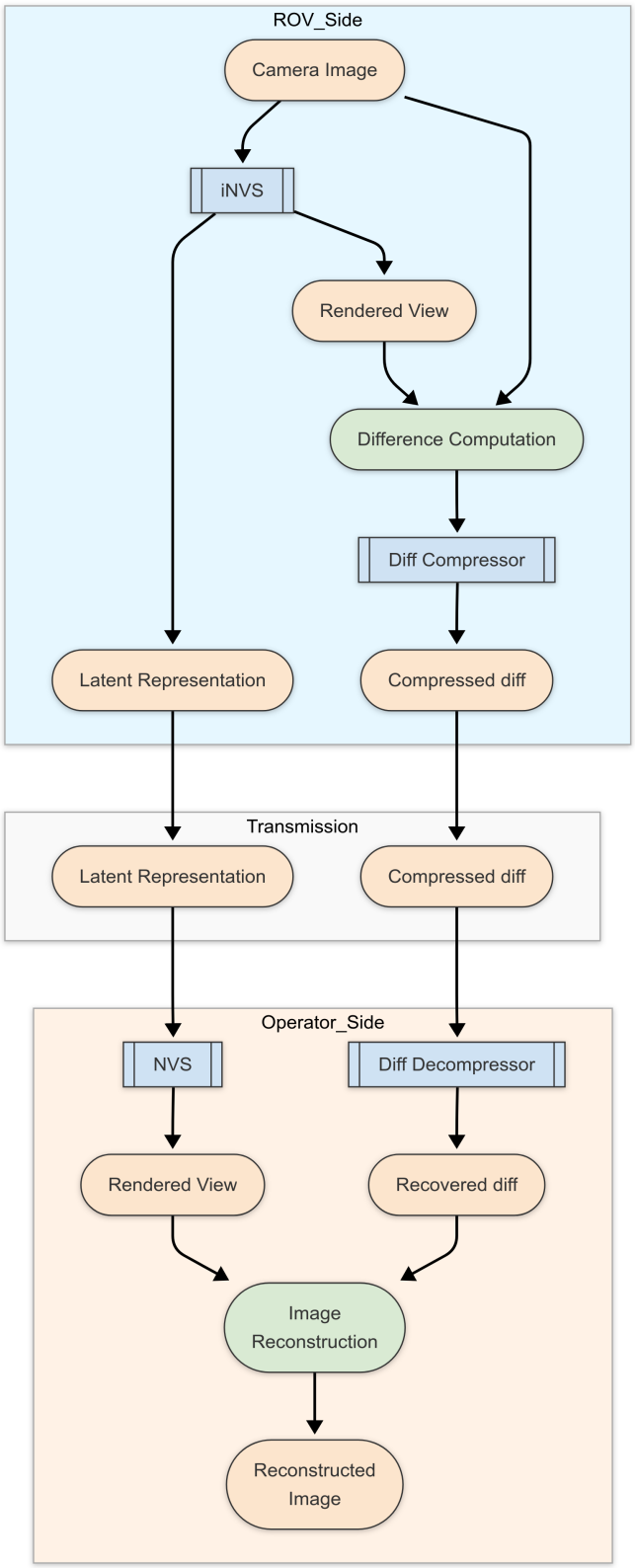


Figure 5.13: NVSPrior pipeline with iNVS for underwater image transmission.

NVSPrior + Affine, which uses affine warping.

Our evaluation metrics are:

1. **Transmitted data size** (in bytes): the total size of data required to reconstruct the image. For our method, this includes both the optimized latent representation and the compressed difference image I_{diff}. The size of the latent representation is fixed at 28 bytes, consisting of 7 float32 values (3 to represent translation of the pose and 4 for quaternion rotation). For the classical codecs, we report the size of the compressed image. For the learned compression method, we report the size of the quantized latent representations after entropy coding [115].

2. **Compression ratio**: the ratio between the size of the original uncompressed RGB image (320 $\times$180 pixels, 3 bytes per pixel) and the size of the transmitted data.

3. **PSNR**: as defined in Equation 5.4, computed between the camera image and reconstructed image, quantifying reconstruction quality.

4. **Processing time per frame**: time required to compress and reconstruct an image, reflecting the method's computational efficiency and real-time feasibility.

We report the results in Table 5.1.

Table 5.1: Quantitative results on the T1 dataset, averaged over 1000 images. Size stands for the size of the transmitted data in bytes. Ratio stands for the compression ratio. Time refers to processing time per frame in milliseconds. Arrows show the increasing/decreasing trend of the metric indicating improvement.

Method	Size ↓	Ratio ↑	PSNR ↑	Time ↓
WebP	3544	48.76	33.30	~ 6
JPEG-XL	5711	30.25	33.57	$\sim \mathbf{1}$
Mean & Scale Hyperprior [25]	10783	16.03	34.81	$\sim$100
MLIC++ [27]	4174	41.40	31.19	~ 129
NVSPrior+Affine+WebP	4164	41.50	31.85	~ 64
NVSPrior+Affine+JPEG-XL	4401	39.26	31.31	~ 59
NVSPrior+iNVS+WebP	**1219**	**141.76**	35.83	~ 62
NVSPrior+iNVS+JPEG-XL	1552	111.34	**36.15**	~ 57

Our results demonstrate that our NVSPrior+iNVS approach achieves the best overall performance among all methods. It achieves a significantly higher compression ratio than WebP and JPEG-XL while maintaining a higher PSNR. iNVS with WebP achieves the highest compression ratio, which is 2.90 times higher than WebP and 4.67 times higher than JPEG-XL.

In Fig. 5.14, we show an example of the iterative optimization process of iNVS. Given a camera image, iNVS rapidly optimizes the camera latent representation to minimize the difference between the camera image and the rendered image. The optimization process converges within a few iterations, demonstrating the efficiency and effectiveness of our technique.

iNVS with JPEG-XL achieves the highest PSNR of 36.15 dB, which is 2.85 dB higher than WebP and 2.58 dB higher than JPEG-XL. In Fig. 5.15a, we observe the reconstructed image from iNVS is clearer and sharper than the image compressed and decompressed by classical methods.

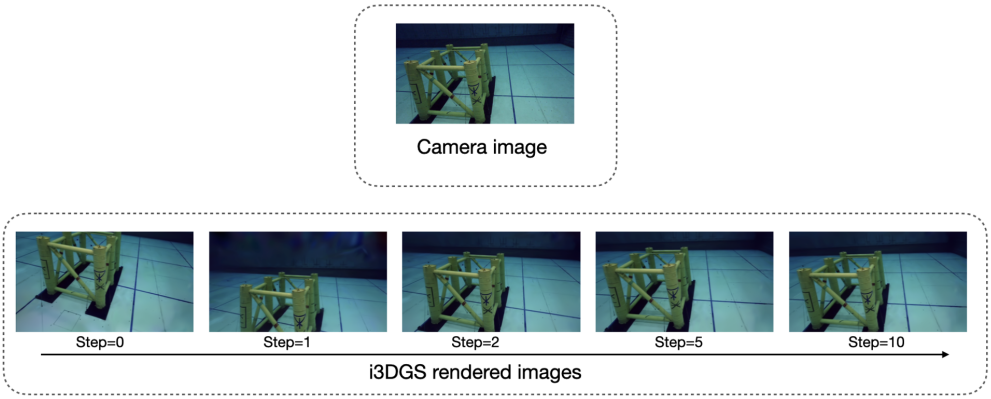
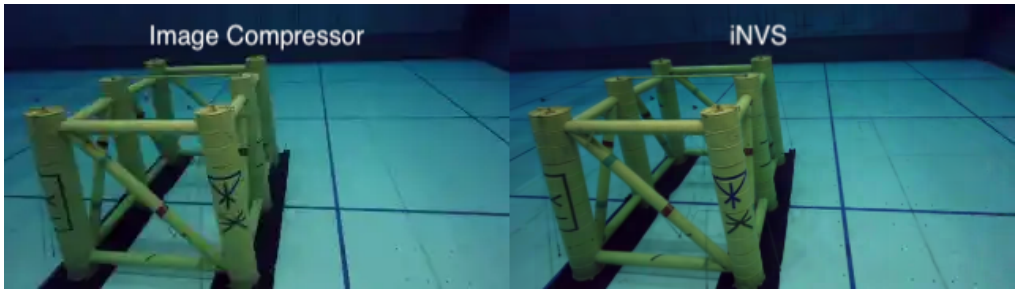
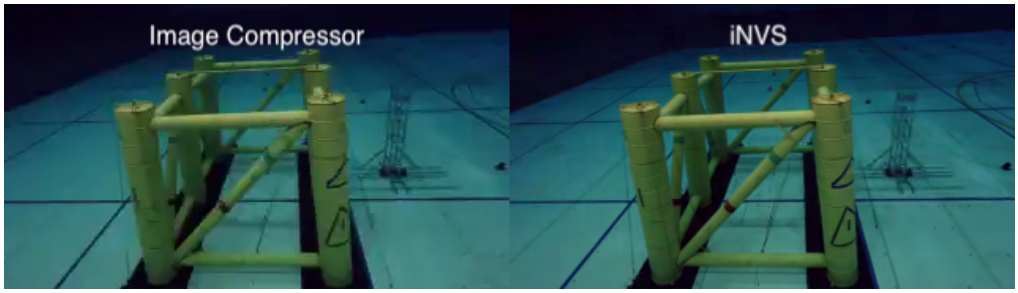


Figure 5.14: iNVS optimization process. Given a camera image, iNVS rapidly and accurately optimizes the latent representation to minimize the difference between the camera image and the rendered image.



(a) Reconstructed images for T1



(b) Reconstructed images for T2

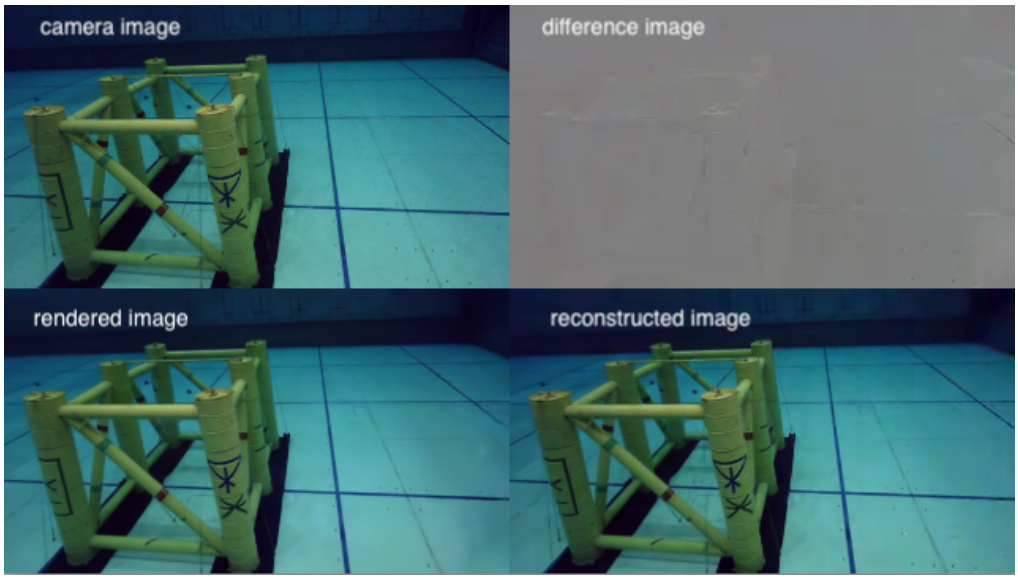
Figure 5.15: Visualization of reconstruction quality for T1 and T2. The left image is the compressed/decompressed image by JPEG-XL, the right image is the image reconstructed using NVSPrior+iNVS+JPEG-XL.

The results demonstrate that our iNVS technique is more effective than classical compression methods for real-time image transmission over limited-bandwidth acoustic links.

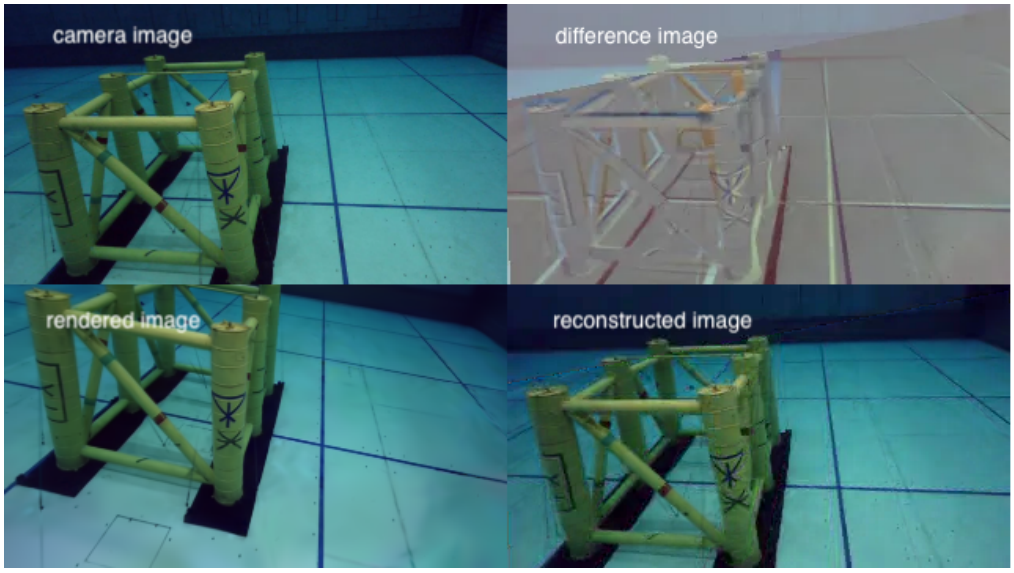
Despite strong results on standard learned image compression bench-

marks [115, 27], Mean & Scale Hyperprior and MLIC++ underperform in our underwater setting. We attribute this to (i) limited domain-specific training data, which constrains learned priors and weakens context modeling, and (ii) a resolution mismatch—MLIC++ is primarily evaluated at higher resolutions, whereas our inputs are small, diminishing the benefit of its multi-reference context modules. In contrast, our NVS-based approach exploits a scene-specific 3D prior learned across viewpoints, assimilating 3D information more efficiently from modest training datasets and generalizing across repeat surveys; under these data-scarce, low-resolution, and underwater-degraded conditions, it achieves better rate–distortion performance.

We also find the overall performance of NVSPrior+Affine is worse than both NVSPrior+iNVS and classical codecs. This is likely due to the affine transformation method assumes a 2D scene geometry and introduces visual artifacts. As illustrated in Fig. 5.16b, the latent representation estimated by PoseLSTM often results in significant misalignment between the rendered and camera images, leading to larger residuals. This misalignment cannot be fully corrected by affine warping, which introduces visual artifacts, thereby increasing the entropy of the difference image. This results in a much larger compressed size using Affine than using iNVS. This underscores the effectiveness of the latent representation optimization by iNVS.



(a) Result of NVSPrior+iNVS.



(b) Result of NVSPrior+Affine.

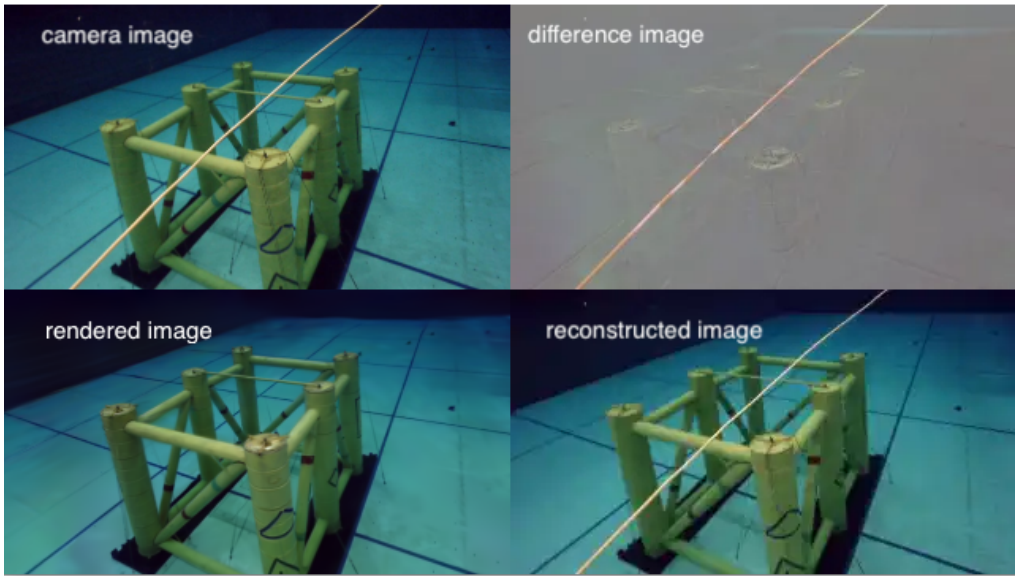
Figure 5.16: Visualization of the compression performance using NVSPrior with either the iNVS (a) or the Affine approach (b). In each of the subfigures, we present the (i) camera image, (ii) rendered image at the estimated latent representation, (iii) the difference between the two images and (iv) the final reconstructed image. The visible artifacts in (b) arise from pose estimation errors and the limitations of affine transformation.

Using our approach, the average transmitted data size is 1.2 kB, allowing approximately 10 frames per second to be sent over a 100 kbps acoustic link. Although our method involves an additional optimization step, iNVS remains computationally efficient, with a runtime of approximately 62 ms per frame. This is largely due to effective initialization from the previous frame's optimized latent representation, which enables rapid convergence of the BFGS optimizer. These properties make NVSPrior+iNVS a practical and scalable solution for real-time image transmission in bandwidth-constrained underwater inspection scenarios.

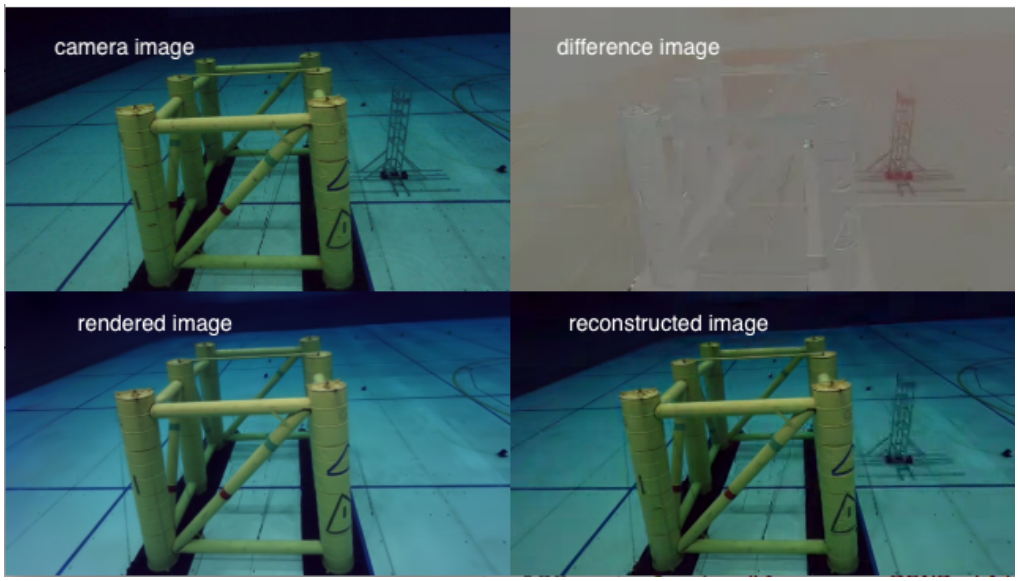
5.2.5 Robustness to novel objects in the scene

In inspection missions, it is common to encounter changes in the scene with time, such as the presence of additional structures or objects (e.g. fish, biological growth, corrosion, etc). We evaluate the robustness of our NVSPrior+iNVS technique to novel objects in the scene using dataset T2, in a similar manner as described above.

We test the performance of iNVS on two examples, representing novel objects commonly encountered in inspection missions. The first is a thin yellow safety line that moves with the vehicle and appears on the camera, as shown in Fig. 5.17a. The second is a stationary metallic object with dimensions approximately 1.0 m $\times$ 0.25 m $\times$ 0.25 m, as shown in Fig. 5.17b. We find that NVSPrior+iNVS handles both types of novel objects well. The average compressed data size with the presence of these objects is 1.65 kB,



(a) Result of the NVSPrior+iNVS with the presence of a safety line.



(b) Results of the NVSPrior+iNVS with the presence of a new structure.

Figure 5.17: Visualization of the compression performance using NVSPrior with iNVS in presence of novel objects. As Fig. 5.16, we present the camera image, rendered image, difference image, and final reconstruction.

allowing us to transmit about 7 frames per second over a 100 kbps acoustic link. The results are summarized in Table 5.2.

We find that even with the presence of novel objects in the scene, our NVS-prior approach remains the most bitrate-efficient and also attains

Table 5.2: Quantitative results on the T2 dataset, averaged over 1000 images. Size stands for size of the transmitted data in bytes. Ratio stands for compression ratio. Time refers to processing time per frame in milliseconds. Arrows indicate the direction of improvement.

Method	Size ↓	Ratio ↑	PSNR ↑	Time ↓
WebP	3655	47.28	33.43	~ 6
JPEG-XL	5827	29.66	33.86	$\sim$ **1**
Mean & Scale Hyperprior [25]	10889	15.87	34.81	$\sim$87
MLIC++ [27]	4260	40.57	30.93	~ 128
NVSPrior+Affine+WebP	4452	38.81	31.83	~ 92
NVSPrior+Affine+JPEG-XL	4629	37.33	31.37	~ 86
NVSPrior+iNVS+WebP	**1651**	**104.66**	35.32	~ 125
NVSPrior+iNVS+JPEG-XL	2073	83.36	**35.55**	~ 119

the highest reconstruction quality. NVSPrior+iNVS+WebP achieves the smallest transmitted size, improving compression by 2.21 times over WebP and 3.53 times over JPEG-XL. NVSPrior+iNVS+JPEG-XL yields the highest PSNR, exceeding WebP by 2.12 dB and JPEG-XL by 1.69 dB, while still reducing size to 2073 bytes.

Learned baselines do not perform well in this setting: the Mean & Scale Hyperprior achieves 35.35 dB PSNR but at a much higher bitrate, and MLIC++ produces lower quality with a larger size than our methods. Relative to the prior Affine variant, iNVS improves both rate and distortion, underscoring the importance of latent refinement rather than 2D warping.

Compared to the results in T1, the performance of iNVS degrades slightly in T2 due to two main reasons. First, the presence of a novel object in the scene increases the difficulty for the trained estimator to provide a good initialization. For the initial frames, we rely on the PoseLSTM estimator,

as the rendered image from the previous frame is determined to be too different from the current frame, as shown by its MSE which is greater than the threshold. Hence, the I_{diff} energies for the first few frames are larger due to less accurate pose initialization. Second, the presence of the novel object increases the entropy of I_{diff}, resulting in a larger compressed size. Nonetheless, the performance of iNVS remains significantly better than that of WebP and JPEG-XL, enabling near-real-time image transmission over limited-bandwidth acoustic links.

5.3 Summary

In this chapter, we proposed affine transformation and iNVS to reduce the compressed size of I_{diff}. While applying affine transformation to match the rendered image to corresponding camera image successfully reduces the compressed size of I_{diff}, its implicit assumption of a 2D world causes artifacts in the difference image, thus increasing the data size to be transmitted. To further reduce the compressed size of I_{diff}, we propose iNVS which optimizes the latent representation through gradient descent by minimizing the MSE between rendered and camera image. We evaluate the performance of different loss functions, optimization methods, and initialization methods for iNVS and demonstrate that MSE loss, BFGS based optimization, and using the pose of the previously acquired frame as the initialization are the most effective options. We evaluate the compression efficiency and reconstruction quality of our NVSPrior approach combined with iNVS in

a confined underwater environment and demonstrate that it outperforms existing image compression techniques, such as WebP, in terms of both compression ratio and image quality. We also examine the robustness of our method towards novel objects in the scene and demonstrate that it can handle occlusion from both small and large structures. Overall, our NVS prior-based technique outperforms both classical codecs and learned compression methods significantly and is a promising solution for real-time image transmission over limited-bandwidth acoustic links in inspection missions.

Chapter 6

NVSPrior in the wild

By now, we have seen that the NVSPrior image compression framework, enhanced by iNVS (as shown in Fig. 6.1), can deliver high compression ratios and superior reconstruction quality, outperforming both classical codecs and learned compression methods. These results, obtained in controlled clear-water environments, demonstrated the power of using NVSPrior for efficient underwater image transmission via bandwidth-limited acoustic links. But the real world, especially the Singapore waters, presents a far more complex challenge.

Here, visibility is often compromised by high turbidity, reducing image

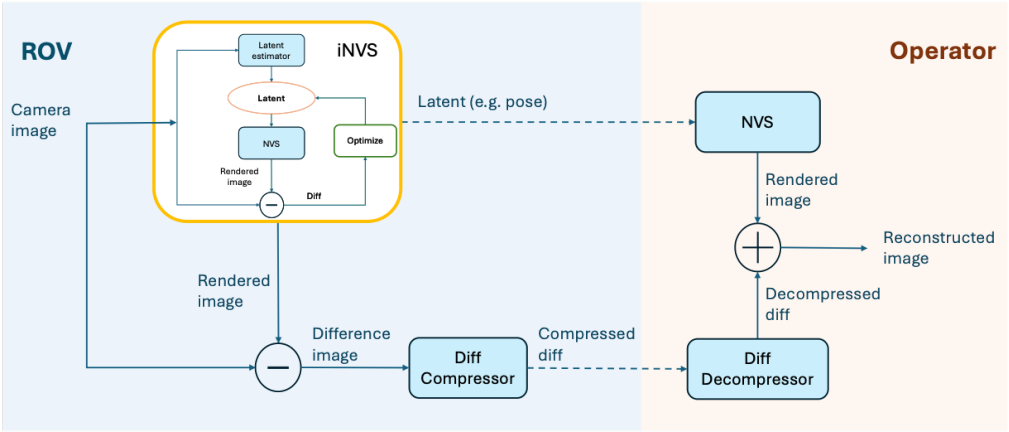


Figure 6.1: NVSPrior enhanced by iNVS.

contrast and introducing haze. Suspended particles and marine snow introduce noise, while sunlight reflections, shadows, and marine growth movement create dynamic visual elements that violate the static-scene assumptions underlying both pose estimation and novel view synthesis. Earlier chapters have already hinted at these challenges: Chapter 3 showed that rendered views from a Splatfacto model trained in Singapore waters suffered from visible artifacts; Chapter 4 revealed that the performance of our pose estimator degraded under the same conditions. These observations raise a critical question: Can our system, successful in the tank, stand up to the noise, uncertainty, and variability of the sea?

In this chapter, we test the complete compression pipeline under real-world conditions. As shown in Figure 6.2, we conduct this evaluation during a live ROV deployment in turbid waters at SJI. We investigate whether the original iNVS configuration remains effective under such conditions, and explore enhancements—such as improved pose initialization and feature-based losses—to increase robustness. This final evaluation brings the full system into an operational setting, assessing the practical effectiveness of NVSPrior for underwater image compression in challenging environments.

6.1 Field evaluation of the original iNVS

We deployed our ROV, Hydra (shown in Fig. 6.3), at the same SJI site described in Section 4.5.1, inspecting a submerged pile in turbid coastal waters.

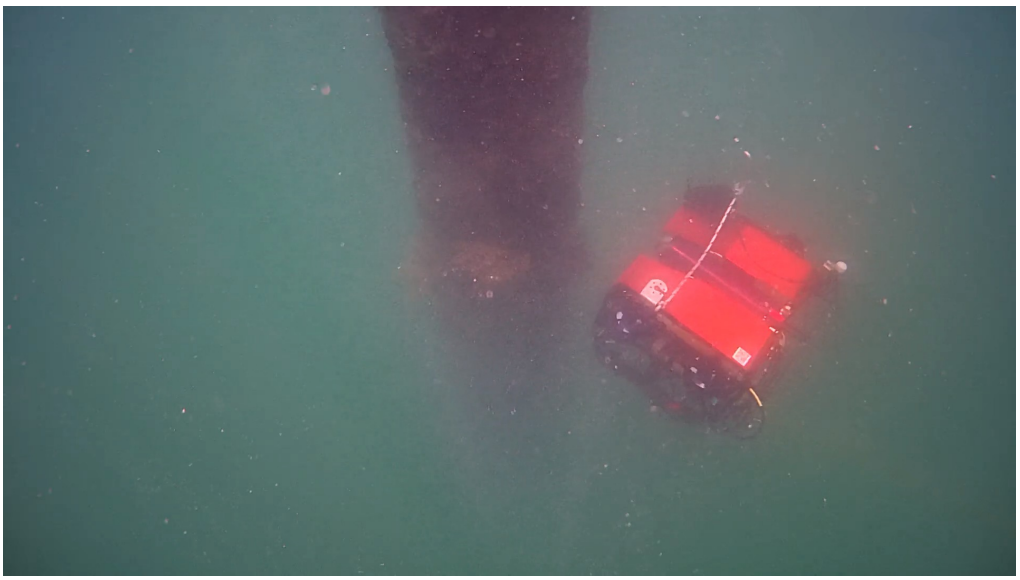


Figure 6.2: ROV deployed during our field demonstration in Singapore waters. The environment features high turbidity.

We conducted two data collection runs on separate days, resulting in two datasets named Sea Water-1 and Sea Water-2. In both runs, Hydra followed a vertical lawnmower trajectory around the pile, capturing imagery at approximately 3 frames per second (fps). Sea Water-1 comprises 1,924 RGB images, while Sea Water-2 contains 399 RGB images. Compared to Sea Water-1, Sea Water-2 was acquired under higher turbidity, yielding noisier imagery and fewer trackable features. This natural variability reflects real-world challenges of underwater inspection, where visibility, lighting, and vehicle trajectory cannot be perfectly controlled.

We use Sea Water-1 for training the NVS model and evaluate on Sea Water-2, following the procedure outlined in Chapter 5. This enables us to test the robustness of our approach under environmental variability.

During this deployment, we tested the original iNVS configuration

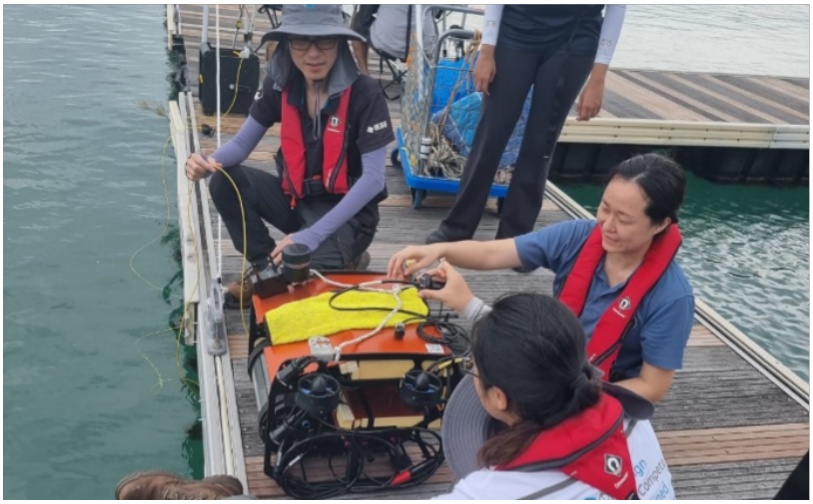


Figure 6.3: Preparation for field deployment at St. John's Island.

proposed in Chapter 5. To recap, the system first attempts to reuse the latent representation from a previous frame by computing the MSE between the normalized camera image and the normalized rendered image generated from that representation. If the MSE falls below a predefined threshold, the frame is considered "good" and its latent code is reused. If no such prior is available, PoseLSTM is used to generate an initial estimate of the latent representation, which is then refined using the BFGS optimization algorithm with the MSE as the objective.

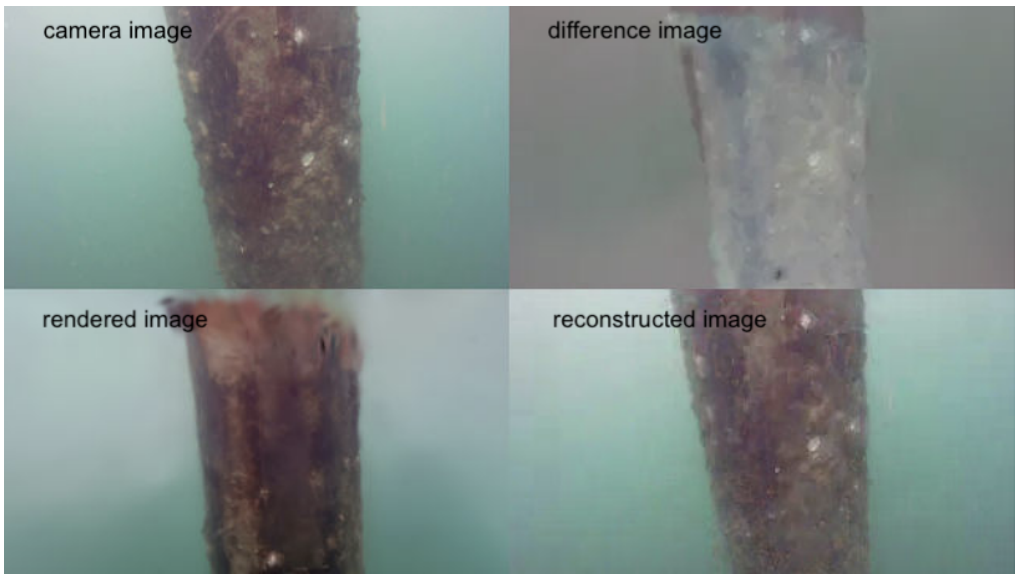
The system achieved a higher compression ratio than WebP and enabled real-time image transmission over the acoustic modem. However, we observed that it operated at only 2–3 frames per second, which is significantly slower than in controlled environments.

Table 6.1 summarizes the performance of different pose initialization and refinement configurations on Sea Water-2, evaluated using PSNR, SSIM, and average compressed size. As shown in Table 6.1, the baseline configu-

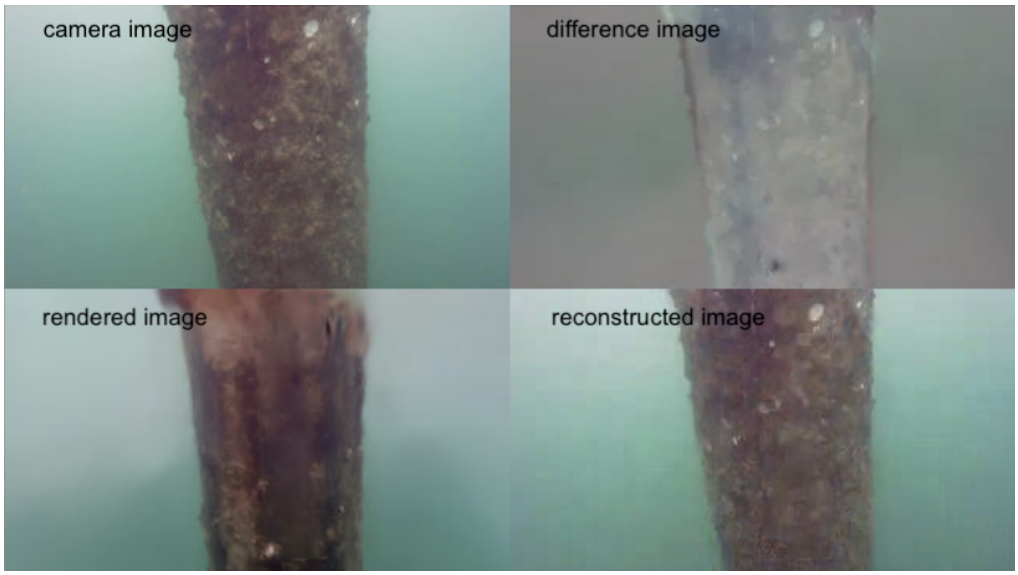
ration of our method (PoseLSTM+refine, row 5) outperformed traditional compression baselines in terms of quantitative metrics. Compared to the case with no prior (row 1), which achieved a compressed size of 6720 bytes, the original iNVS configuration reduced the number of transmitted bytes to 4101.11—a reduction of approximately 39%. These results confirm that, under real-world conditions, the baseline iNVS configuration still achieves a compression performance superior to classical methods.

However, visualization of the rendered views at the optimized poses reveals that iNVS failed to find the correct latent representation. In particular, it often produced a nearly fixed viewpoint of the pile, regardless of the actual camera pose in the input image. As illustrated in Fig.6.4, even as the camera moves downward along the structure—from Fig.6.4a to Fig. 6.4b—the rendered image remains largely unchanged. This indicates that pose optimization fails, yet the MSE remains below the preset threshold, incorrectly identifying the frame as a good prior.

Conversely, in cases where the rendered and camera images are well-aligned, the MSE is sometimes high, leading to the false rejection of valid priors. This typically occurs when there are noticeable lighting differences in the background. We suspect that the observed mismatch between MSE scores and actual alignment quality arises in part from the simplicity and repetitive appearance of the structure. The pile's visual features are largely uniform due to its cylindrical shape and repetitive patterns, resulting in minimal variation between different sections. Moreover, the pile typically



(a)



(b)

Figure 6.4: Visualization of the results in SJI with baseline configuration.

occupies less than half of the image, with the remaining area dominated by background. As a result, background lighting differences contribute more to the pixel-wise MSE than structural differences.

These findings indicate that MSE is not a robust indicator for determining whether a frame is suitable for initialization. Furthermore, they suggest

that MSE may no longer be an appropriate objective function for pose optimization in real-world underwater environments, where background noise and environmental variability can distort pixel-level comparisons.

These observations prompted a re-evaluation of both our reliance on the NVS-rendered image as a prior and the use of MSE as the objective function. In the following sections, we investigate whether simpler priors, such as a static background, can offer similar benefits. We also explore enhancements to pose initialization and alternative loss functions designed to improve robustness under the challenging and variable conditions of real-world underwater environments.

6.2 Revisiting the role of priors

In our baseline, the NVS-rendered image often failed to match the correct section of the structure but still led to improved compression performance over classical methods. This raised a key question: how much of this improvement comes from leveraging prior information about the structure, and how much simply from having a visually stable background?

To explore this, we tested a much simpler prior—a static image with no structural content—to isolate the contribution of background similarity.

In this experiment, we used the same image, captured from the water column with no visible structure, as the prior for every frame. For each frame, we computed the difference between the camera image and the static prior, compressed this difference, and transmitted it. At the receiver's

Table 6.1: Comparison of different pose initialization and refinement methods on dataset 2. Best values for PSNR, SSIM, and average compressed size in bytes are shown in bold.

Method	**PSNR**	**SSIM**	**Compressed size**
No prior	-	-	6720
Background prior	13.44	0.6362	3760
PoseLSTM	18.14	0.7072	4459

end, the reconstructed image was obtained by adding the decompressed difference back to the static prior.

Surprisingly, this naive method (background prior) outperformed our baseline iNVS (PoseLSTM+refine) in terms of compression efficiency. As shown in Table 6.1, despite a noticeable drop in image quality, the method achieved better compression performance, with a 9.5% reduction in compressed size compared to the baseline iNVS configuration.

These results indicate that background information plays a critical role in compression performance. In this case, the static background prior even outperformed the NVS prior probably because it avoided structural misalignment errors, which can lead to larger difference images and reduced compression efficiency.

Although the performance of the simple background prior is primarily due to the nature of the scene being simple and repetitive, and may not generalize to more complex environments, it provides a valuable benchmark. Our method, which incorporates more detailed information about the scene, should be able to outperform this baseline, even under such unfavorable conditions.

The following sections focus on addressing the limitations identified in our baseline configuration. In particular, we explore improvements to pose initialization and the introduction of more robust loss functions, with the goal of enhancing the overall reliability and performance of our approach in real-world underwater environments.

6.3 iNVS-w: In-the-Wild Variant

To address the limitations identified in our baseline configuration, we propose iNVS-w, a robustified variant that integrates a DFNet-inspired pose regressor and a perceptual refinement loss. iNVS-w retains the optimization-based refinement of the latent representation, but improves robustness through two key enhancements: (i) a more accurate initialization strategy based on a DFNet-inspired regressor, and (ii) a perceptual similarity loss based on multi-scale structural similarity (MS-SSIM).

6.3.1 Pose Initialization with DFNet-inspired Regressor

Original iNVS employed a PoseLSTM regressor, an extension of PoseNet [91, 62]. While effective in structured environments, such regressors tend to generalize poorly in the wild, particularly underwater, where turbidity, illumination, and scattering induce strong photometric shifts. As a result, pose predictions can drift significantly when training and deployment domains differ.

To address this, we adopt DFNet [75], which extends PoseNet by explic-

itly incorporating image-level feedback from NVS. Rather than regressing poses solely from ground-truth supervision, DFNet aligns training with the downstream objective of iNVS: rendering consistency. During training, a real image and its ground-truth pose are paired with a synthetic view generated by a pretrained NVS model. Both the real and synthetic images are fed through a shared feature extractor and pose regressor. Supervision comes from two components: (i) a standard pose regression loss comparing predicted poses to ground truth, and (ii) a triplet loss in feature space, which enforces that features from real and rendered views of the same pose are closer than those from different poses—thus encouraging domain-invariant, pose-sensitive representations. This combination enhances robustness to appearance variations common in images collected underwater.

We implement DFNet in PyTorch with an EfficientNet-B3 backbone [75, 118]. We train the pose estimator using Sea Water-1 images and their corresponding COLMAP poses. Images are resized to the network resolution and normalized to ImageNet statistics. To improve generalization in turbid conditions, we generate synthetic data from the trained 3DGS model by jittering each training pose.

We train DFNet following a two-stage procedure. First, we optimize the feature extractor using paired real and rendered images to encourage consistency between feature representations. Next, we train the full network for absolute pose regression, using both photometric and feature-matching losses. Training is performed with the Adam optimizer and early stopping;

standard hyperparameters are used.

6.3.2 Perceptual Loss

Our previous work adopted a pixel-wise MSE loss between the rendered image and the input camera image. However, MSE penalizes all pixel-level differences equally, making it highly sensitive to local lighting variation and turbidity-induced color inconsistencies.

To provide robustness against photometric variability and transient changes, we optimize the MS-SSIM between the rendered and observed images. MS-SSIM extends the structural similarity index by combining luminance, contrast, and structure components across progressively downsampled image scales [86]:

$$\text{MS-SSIM}(x, y) = \Big[l_M(x, y)\Big]^{\alpha_M} \prod_{j=1}^{M} \Big[c_j(x, y)\Big]^{\beta_j} \Big[s_j(x, y)\Big]^{\gamma_j},$$

where l_M, c_j, s_j denote the luminance, contrast, and structure terms at scale j, and $\alpha_M, \beta_j, \gamma_j$ are scale-dependent weights.

The loss is defined as the dissimilarity:

$$L_{\text{DMS-SSIM}}(I_{\text{cam}}, I_{\text{rend}}) = 1 - \text{MS-SSIM}(I_{\text{cam}}, I_{\text{rend}}).$$

Compared to MSE, MS-SSIM is less sensitive to global illumination or contrast shifts, and emphasizes structural alignment, which is particularly effective for handling transient changes in real-world underwater scenes.

6.3.3 Results and Discussion

We comprehensively evaluate NVSPrior with iNVS and iNVS-w against both conventional codecs and representative learned compression models. The evaluation includes (i) quantitative benchmarks of rate–distortion trade-offs, perceptual fidelity, and runtime efficiency, (ii) qualitative comparisons to examine the perceptual quality of reconstructions and their suitability for operator use, and (iii) ablation studies isolating the effect of loss functions, feature-space supervision, and temporal initialization.

To emulate realistic transmission conditions, we evaluate the performance using images of 640 × 360 pixels for high-speed acoustic links (about 100 kbps) and 320 × 180 pixels for standard acoustic links (about 30 kbps), matching the effective bandwidth of commonly used underwater modems. Beyond reporting metrics, we interpret results in the context of underwater deployment, showing how our method achieves modem-feasible bitrates, what runtime limitations remain, and how these insights extend to real-world inspection scenarios.

6.3.3.1 Quantitative Evaluation

We benchmark iNVS and iNVS-w against two categories of baselines: (i) conventional codecs and (ii) learned codecs. For conventional baselines, we select JPEG XL, the most advanced member of the JPEG family, and WebP, one of the most widely deployed image compression standards [16, 17]. For learned baselines, we evaluate Cheng'20 and MLIC++, two learned

Table 6.2: **Quantitative results at the highest-compression (lowest-bitrate) setting.** Comparison of iNVS-w with iNVS, conventional codecs, and learned compression methods. Metrics: PSNR, SSIM, LPIPS, BPP, and runtime per frame. Best results per column in bold.

(a) 640×360

Method	PSNR↑	SSIM↑	LPIPS↓	BPP↓	Time(s)↓
iNVS-w (WebP)	34.57	0.9142	**0.1302**	**0.0353**	0.572
iNVS (WebP)	33.49	0.8893	0.1494	0.0438	0.341
JPEG-XL	**38.49**	**0.9398**	0.1351	0.0651	0.092
WebP	37.15	0.9324	0.1334	0.0681	**0.011**
Cheng'20	24.20	0.8674	0.4108	0.0946	4.328
MLIC++	34.91	0.9263	0.2412	0.0387	0.242

(b) 320×180

Method	PSNR↑	SSIM↑	LPIPS↓	BPP↓	Time(s)↓
iNVS-w (WebP)	33.93	**0.9364**	**0.0867**	**0.0469**	0.627
iNVS (WebP)	32.09	0.8536	0.1133	0.0599	0.458
JPEG-XL	**37.01**	0.9244	0.1367	0.0983	0.043
WebP	35.99	0.9164	0.1240	0.1000	**0.003**
Cheng'20	22.54	0.8157	0.3824	0.2407	1.077
MLIC++	33.60	0.9048	0.1926	0.0697	0.187

codecs are representative of state-of-the-art research methods optimized for rate–distortion performance [26, 115, 27]. Both learned codecs are trained on the SeaWater-1 dataset. Across all methods, we report compression efficiency in bits per pixel (bpp), reconstruction fidelity using peak signal-to-noise ratio (PSNR), SSIM, and learned perceptual image patch similarity (LPIPS) as well as runtime per frame measured on a single NVIDIA RTX-3090 GPU.

Performance at highest compression Table 6.2 summarizes the quantitative performance of the proposed iNVS-w method against iNVS, conventional codecs and learned compression baselines at both image resolutions.

At the higher resolution (640 × 360), our proposed iNVS-w achieves 0.0353 bpp, corresponding to approximately 12 fps over a 100 kbps high-speed acoustic link. At the lower resolution (320 × 180), the operating point of 0.047 bpp translates to approximately 11 fps on a 30 kbps standard link. As shown in Fig. 6.5, iNVS-w is the only method capable of exceeding 10 fps using a standard-bandwidth link, confirming its suitability for real-time transmission under realistic underwater constraints.

Across all metrics, iNVS-w achieves the best balance between compression efficiency and perceptual fidelity. Relative to iNVS, it improves both compression efficiency and reconstruction fidelity, albeit with higher runtime. It also outperforms conventional codecs with markedly lower LPIPS and comparable SSIM at substantially reduced bitrates, and further achieves better efficiency and fidelity than state-of-the-art learned codecs.

Rate–distortion performance To comprehensively assess compression efficiency, we analyze the rate–distortion performance of all methods across multiple compression levels. As shown in Fig. 6.6, iNVS-w consistently dominates the low-bitrate regime, achieving higher perceptual quality at lower bitrates compared with all other methods. We further compute the Bjøntegaard Delta Rate (BD-Rate) [119] using LPIPS as the quality metric and WebP as the anchor codec. The proposed iNVS-w achieves an average 45–60% bitrate reduction at equivalent perceptual quality, demonstrating substantial gains in perceptual rate–distortion efficiency.

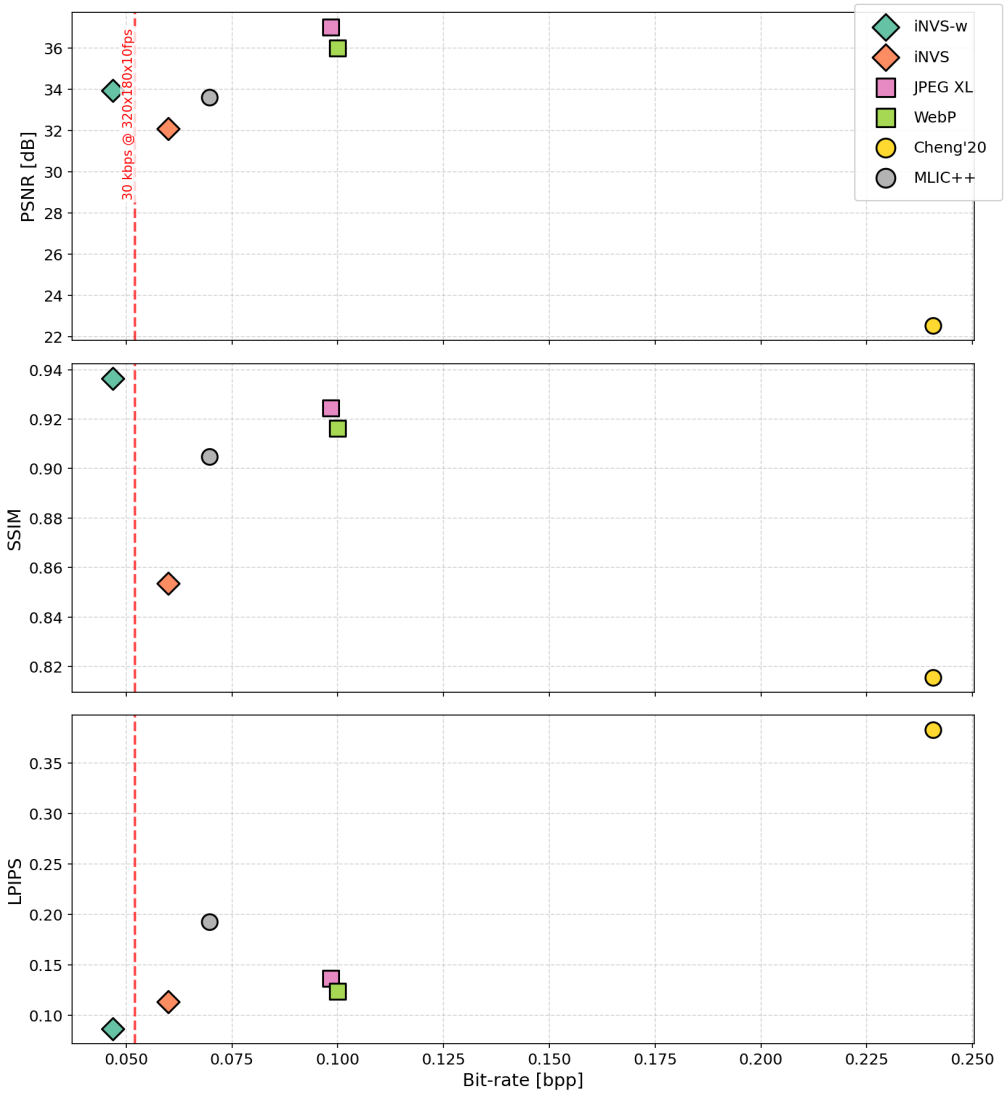


Figure 6.5: Performance of all methods at highest compression (320×180). The red dashed line marks the communication budget of 30 kbps.

Runtime analysis Table 6.2 reports the per-frame latency measured on an NVIDIA RTX 3090 GPU. The proposed iNVS-w achieves 0.57–0.63 s per frame, which is comparable to state-of-the-art learned methods but one to two orders of magnitude slower than classical codecs. In our framework, runtime is governed by the number of optimization iterations performed during compression. More iterations reduce the difference between rendered

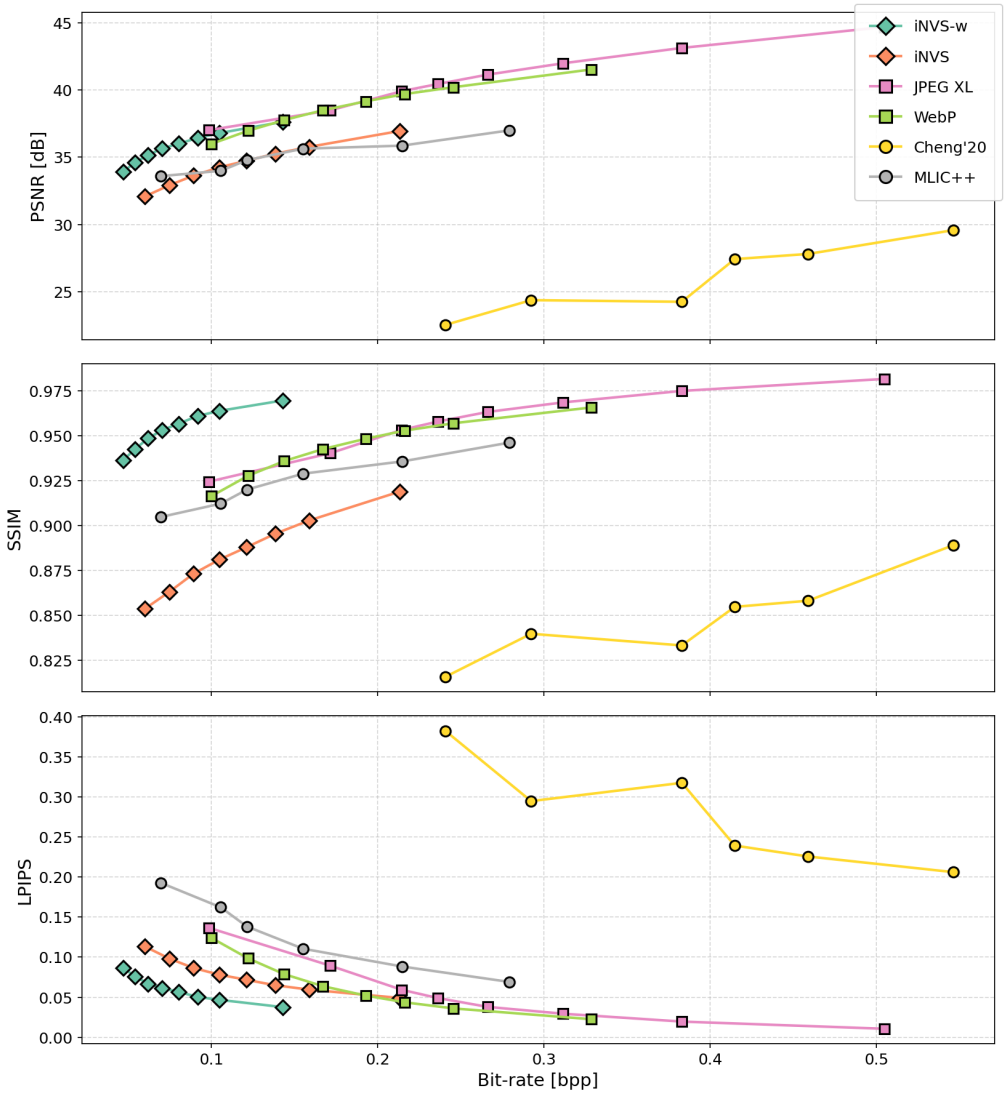


Figure 6.6: Rate–distortion curves across the full operating range (320×180).

and captured images, yielding a lower bit rate at the cost of longer processing time.

Runtime can be tuned by limiting iterations, tightening convergence tolerances, or performing coarse-to-fine refinement at lower resolutions. These adjustments provide significant speedups with modest impact on compression quality, enabling users to balance bandwidth efficiency against computational latency based on mission requirements.

Although iNVS-w is not yet real-time, it offers a practical balance among quality, bitrate, and runtime, with a clear path toward deployability in future underwater inspection systems.

6.3.4 Qualitative Results

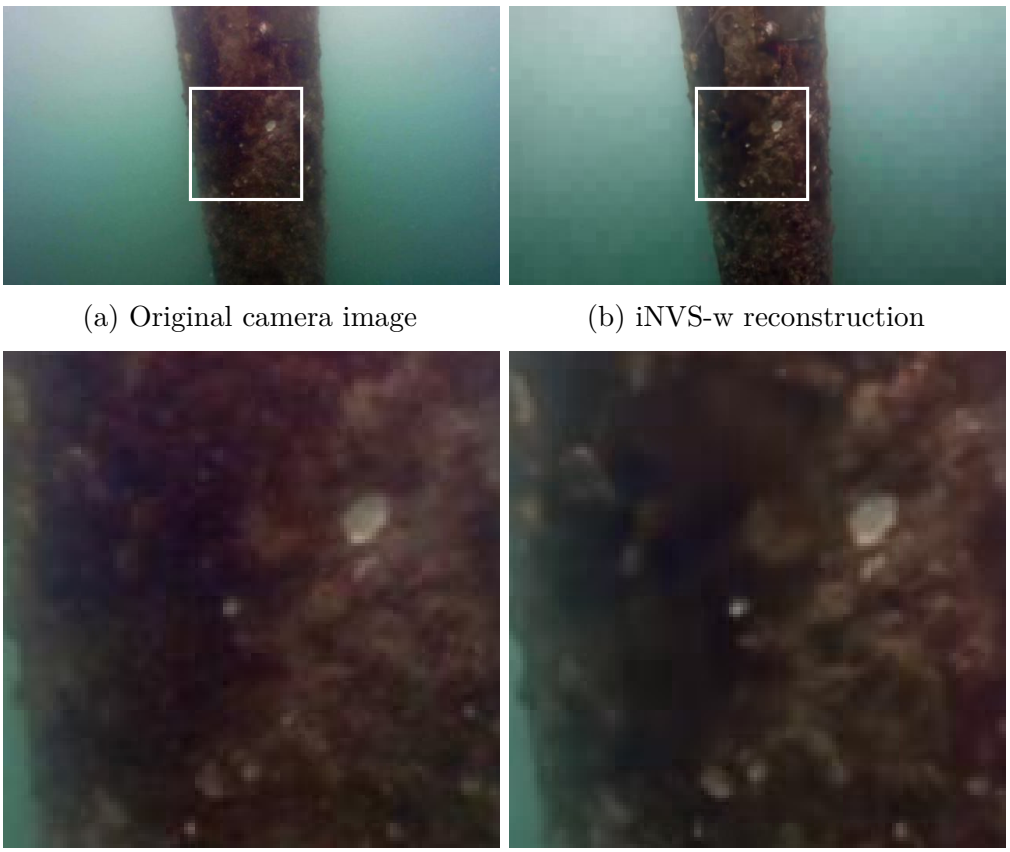
Beyond numerical metrics, we examine reconstruction quality of NVSPrior+iNVS-w directly. Figure 6.7 compares a representative raw camera frame with its reconstruction by NVSPrior+iNVS-w, along with zoomed crops. Despite operating at only 0.0353 bpp, the reconstruction preserves the key geometric structures on the pile surface and retains sufficient clarity for operator situational awareness.

Figure 6.8 shows that our reconstructions consistently preserve high fidelity across the test set. These results highlight that iNVS-w achieves modem-feasible bitrates without compromising the level of visual fidelity required for field deployment.

6.3.5 Ablation Studies

We conduct ablation studies with respect to three design choices in iNVS-w using the DFNet backbone with WebP: the *refinement loss function*, *feature-space supervision*, and *temporal initialization*. Figures 6.9 reports rate–distortion curves; quantitative aggregates appear in Table 6.3.

Choice of loss function We first evaluate the impact of standard pixel- and perceptual-domain losses. As shown in Table 6.3, using MSE or mean



(a) Original camera image (b) iNVS-w reconstruction

(c) Zoom of camera image (d) Zoom of reconstruction

Figure 6.7: Qualitative comparison between a raw camera frame and its reconstruction by **iNVS-w**.

absolute error (MAE) does not improve performance over the DFNet baseline and in fact yields slightly lower PSNR and SSIM. In contrast, adopting MS-SSIM consistently raises PSNR across the bitrate range. This indicates that pixel-domain errors are not robust to photometric variations induced by underwater lighting, turbidity, and other factors, whereas perceptual similarity provides a more reliable optimization target under such conditions.

Loss in alternative spaces We next examine whether computing the loss in alternative domains can improve robustness. The first variant converts

Figure 6.8: Representative qualitative comparisons of raw camera frames (top) and iNVS-w reconstructions (bottom), shown at 0.0353 bpp.

Table 6.3: Ablation study on DFNet with WebP/20, showing the effect of refinement loss, feature-space loss, grayscale MAE, and previous-pose initialization. Metrics are averaged over the test set. Best values are in bold.

Refine Loss	Feature Loss	Prev. Pose	Gray-scale	PSNR ↑	SSIM ↑	BPP ↓	Time (s)↓
No refine	–	–	–	32.18	0.923	0.042	**0.01**
MSE	–	–	–	32.19	0.923	0.043	0.37
MAE	–	–	–	32.15	0.924	0.043	0.38
MAE	–	–	✓	32.12	0.923	0.043	0.41
MS-SSIM	–	–	–	**32.76**	**0.940**	**0.034**	0.58
MS-SSIM	✓	–	–	32.74	0.940	0.034	1.54
MS-SSIM	–	✓	–	32.71	0.939	0.034	0.59

images to grayscale before computing the MAE, with the aim of reducing sensitivity to unstable color channels. However, as shown in Table 6.3, this does not yield any benefit over standard MAE and in fact produces slightly lower PSNR and SSIM. A likely explanation is that color channels may contain useful cues for pose estimation. In underwater imagery, localized beams or flares can cause sharp intensity variations, which in grayscale are integrated into a single channel and therefore penalized more strongly.

The second variant evaluates the effectiveness of computing the loss in feature space, with the goal of extracting features that are invariant to such photometric changes. Here, we implement a differentiable version

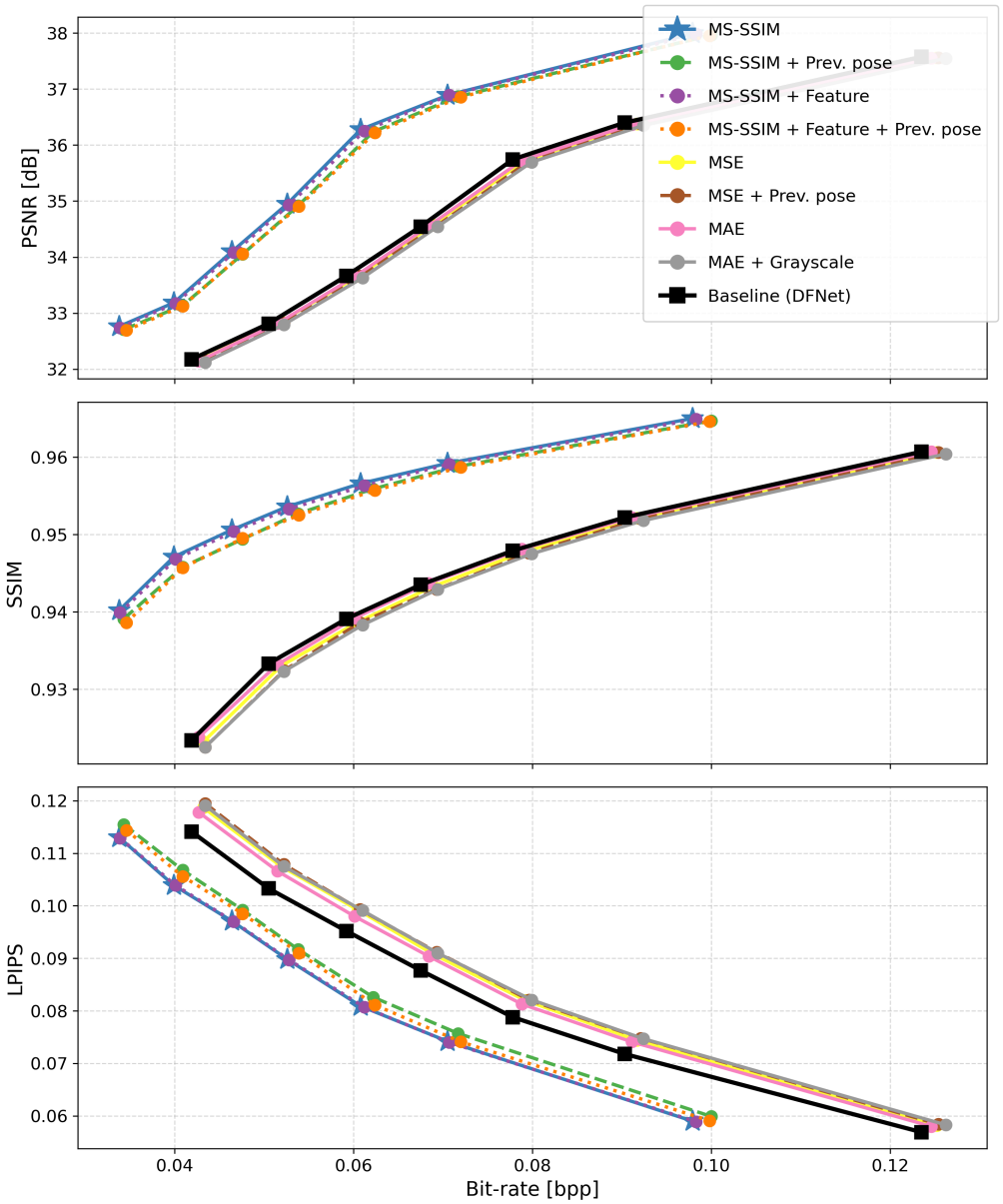


Figure 6.9: Ablation of loss domain, feature-space supervision, and temporal priors with DFNet+WebP. The selected iNVS-w configuration (image-space MS-SSIM, no auxiliary modules) offers the best quality–bitrate trade-off with favorable runtime.

of descriptor fields [120] as a feature extractor, and compute MS-SSIM between rendered and observed images in this feature space. While this formulation improves robustness to illumination and background variation, Table 6.3 shows that its quantitative performance is comparable to image-

space MS-SSIM but incurs a significantly higher computational cost.

Previous-pose initialization Finally, we assess the benefit of initialization using temporally recent priors instead of a dedicated pose regressor. Our earlier work [91] adopted this strategy to exploit continuity in ROV video streams, improving accuracy and reducing runtime. In this study, we evaluate both reusing the previous pose estimate, and using a constant-velocity model based pose initialization instead. As shown in Table 6.3, neither approach provides gains in accuracy, bitrate, or runtime, and both in fact perform slightly worse than directly using the regressor. This is because the DFNet-inspired pose regressor is already highly accurate, producing pose estimates closer to the ground truth than temporal priors. This suggests that in our setting, initialization exploiting temporal correlation is unnecessary, though it may remain useful when weaker regressors are used.

6.3.6 Discussion

The results highlight both the promise and limitations of NVSPrior-based compression in real-world underwater settings. NVSPrior + iNVS-w consistently achieves modem-feasible bitrates while maintaining high reconstruction fidelity, outperforming learned baselines and matching conventional codecs at a fraction of the bitrate. Ablation studies identify perceptual loss as the key enabler of robustness under photometric variability. Together, these findings suggest that iNVS-w is a practical candidate for enabling continuous video feedback in tetherless ROV operations.

Nonetheless, NVSPrior+iNVS-w is still not fully adequate as a real-time solution for tetherless teleoperation: the current latency of 0.58 s/frame falls short of real-time requirements. Further optimizations such as pruning or reducing refinement iterations could narrow this gap, especially as embedded GPUs continue to improve. Additional gains may also be achieved by explicitly modeling transient changes in the scene—such as soft coral movement, suspended particulates, or lighting variation—so that novel-view synthesis produces fewer artifacts, enabling faster and more stable gradient descent during refinement.

In summary, our results show that iNVS-w achieves a favorable balance between bitrate, quality, and runtime for underwater teleoperation, but dynamic scene changes and runtime latency remain key challenges. While perceptual losses mitigate photometric variability, static-scene priors struggle with moving structures, inflating residuals and complicating gradient refinement. Addressing these limitations will be critical for robust real-time deployment in natural underwater environments.

6.4 From tank to field: why compression performance degrades

In the previous section, we showed that our best configuration—iNVS initialized with DFNet and refined using a feature-based loss—achieves higher compression efficiency than classical methods. However, results from the SJI trials remain significantly worse than those obtained in controlled

environments, with compressed difference images approximately three times larger.

A large difference image can arise either from rendering at an incorrect viewpoint or from changes in the scene between the time of reconstruction and the time of inspection. Here, we aim to determine whether the observed performance gap is primarily due to inaccuracies in pose estimation or scene changes.

To investigate this, we evaluate the compression performance of three configurations: (1) our original iNVS configuration, (2) our best configuration (DFNet + refinement with feature-based loss), and (3) ground-truth poses. For ground-truth evaluation, each camera image is rendered using its corresponding Splatfacto pose, and the difference image is compressed using WebP. As it was not straightforward to apply COLMAP on the test dataset to estimate reliable ground-truth poses, we restrict this evaluation to the training dataset.

The results are summarized in Table 6.4. Our best configuration achieves compression performance close to that of ground-truth poses, with a compressed size of 2476 bytes compared to 2456 bytes and nearly identical SSIM. Compared to the original configuration, the gap to ground-truth performance is significantly reduced.

However, even with ground-truth poses, the compressed difference images remain substantially larger than those obtained in controlled environments (about 1269 bytes). This confirms that scene changes in the field are a

Table 6.4: Compression performance on the training set using ground-truth and predicted poses. Ground-truth poses are only available for the training set. Best values for PSNR, SSIM, average compressed size in bytes are shown in bold.

Method	**PSNR**	**SSIM**	**Compressed Size**
No prior	-	-	7012
Groundtruth	**24.01**	**0.9420**	2456
PoseLSTM+refine	24.69	0.8726	3598
DFNet+refine+feature	24.06	**0.9413**	**2476**

significant source of compression inefficiency within our proposed framework. Traditional image compression algorithms such as WebP, which assume global smoothness and low entropy, are less effective on difference images in open water, where noise from turbidity, marine snow, and dynamic lighting is prevalent.

These findings highlight the need to design future approaches that directly address the compressibility of difference images under real-world variability.

6.5 Summary

In this chapter, we evaluated the performance of our iNVS-based visual compression framework in real-world conditions at sea, which present substantially greater visual complexity than confined underwater environments. We revisited the same at-sea site used in earlier pose estimation experiments and established a baseline using the pipeline introduced in the previous chapter.

We showed that while our approach provided clear compression advan-

tages over classical methods such as WebP and JPEG-XL, its performance was initially limited by unreliable pose initialization and sensitivity to appearance changes. To improve robustness, we replaced PoseLSTM with DFNet—a learning-based pose regression model trained using feature-space consistency through novel view synthesis—and introduced a feature-based loss inspired by descriptor fields. These enhancements led to measurable improvements in both reconstruction quality and compression efficiency.

Our best-performing configuration, DFNet initialization combined with feature-based refinement, achieved compression performance close to the upper bound set by ground-truth poses. This demonstrates that, with accurate initialization and a robust objective function, iNVS can operate reliably even under challenging real-world conditions.

Despite these improvements, two major limitations remain. First, the system is computationally intensive: because MSE is no longer a reliable indicator for prior reuse, refinement must be performed on every frame, and the feature-based loss adds additional overhead. Second, even with accurate alignment, difference images in seawater remain difficult to compress using classical compression techniques due to scene changes, turbidity, and dynamic lighting, limiting overall efficiency.

These findings underscore both the potential and the challenges of iNVS-based compression in open-water settings, motivating future research on accelerating the pipeline and designing methods that directly address the compressibility of difference images.

Chapter 7

Conclusion and Future Research

7.1 Conclusion

Underwater inspection missions are essential for ensuring the integrity, safety, and operational continuity of critical subsea assets such as pipelines, offshore platforms, and scientific installations. ROVs have long been the preferred tool for conducting these tasks, offering the capability to perform complex operations in deep and hazardous environments under the control of human operators.

The development of tetherless ROVs has become an important objective, promising greater operational flexibility, reduced deployment costs, and access to environments that are otherwise challenging for tethered systems. However, tetherless operations introduce a new set of challenges, chief among them being the need for real-time visual feeds over severely bandwidth-constrained acoustic links.

To address this challenge, we proposed NVSPrior, a novel framework that leverages prior information collected during earlier mapping runs to enable efficient real-time transmission. Instead of transmitting fully compressed

images, NVSPrior trains a NVS model using images collected during the mapping run. During the inspection run, it renders the operator's view based on an estimated camera pose, compresses the difference between the actual camera view and the synthesized view, and transmits both the estimated pose and the compressed difference image to the operator. Using the received estimated pose and compressed difference image, NVSPrior reconstructs the camera view on the operator side. This strategy significantly reduces the communication load while preserving visual fidelity.

To establish the foundation for this approach, we evaluated the feasibility of novel view synthesis techniques for underwater environments. We analyzed NeRF and 3D Gaussian Splatting models across controlled and real-world datasets, addressing challenges such as transient artifacts, turbidity, and scattering. Our results showed that robustness-enhanced NeRF variants and carefully tuned 3D-GS models can generate high-fidelity reconstructions, laying the groundwork for the NVSPrior framework.

Accurate latent estimation is critical for ensuring that the NVS rendered image aligns closely with the actual camera view, which directly impacts the compression performance of NVSPrior. To address this, we developed a neural network-based pose estimator trained with a domain-informed loss function. We further proposed augmenting the training data with synthesized views generated by NVS techniques, which significantly improves the generalization of the pose estimator to unseen poses. In addition, integrating the pose estimates with sensor measurements through an EKF

improves the smoothness and stability of the estimates. Together, these developments enable accurate localization and robust performance under real-world challenges.

Small pose inaccuracies can lead to large difference images, resulting in inferior image compression performance. To address this, we proposed iNVS, which refines the estimated pose by minimizing the difference between the rendered and camera images. We evaluated NVSPrior in controlled clear underwater environments and found that incorporating iNVS significantly improved compression efficiency and reconstruction quality. Our approach outperformed both traditional codecs and learned compression methods, demonstrated robustness to novel objects and occlusions, and successfully enabled real-time underwater image transmission over acoustic links.

In practice, the efficiency and effectiveness of the proposed approaches may be impacted by highly dynamic and unpredictable elements. Specifically, such elements can increase the size of the difference image due to pose estimation errors and mismatches between camera images and rendered images, thereby affecting both computation efficiency and compressed size of difference image.

To extend the applicability of our approach to real-world conditions, we enhanced iNVS by improving pose initialization with a more robust pose regression model and replacing the objective function with a feature-based loss. Field trials conducted in open seawater environments with varying turbidity and lighting conditions confirmed the effectiveness of NVSPrior combined

with enhanced iNVS. Despite the increased environmental complexity compared to controlled settings, the system achieved outstanding improvements over traditional methods, highlighting its potential for enabling real-time tetherless ROV operations in the field.

7.2 Future directions

While this thesis has demonstrated the feasibility and effectiveness of leveraging NVS prior-based image compression for real-time underwater image transmission and tetherless control of ROV, several promising directions remain for future exploration:

- **Handling dynamic elements in the environment**

  Currently, the NVS models primarily focus on reconstructing static underwater structures. Future work could extend these models to better handle moving features such as algae, enabling more robust novel view synthesis in dynamic underwater scenes.

- **Modeling environmental variability**

  Changing ambient light conditions and turbidity levels can significantly affect underwater imaging. Incorporating models that explicitly account for these environmental factors would improve the realism and reliability of synthesized views, particularly in highly variable natural settings.

- **Optimizing a richer latent representation for novel view**

synthesis

Rather than regressing and optimizing only the camera pose, future work could explore predicting a richer latent representation that captures not only spatial position but also environmental factors such as ambient light conditions, turbidity, and scene variability. This enhanced latent representation could then be fed into the NVS model to generate more realistic and adaptive synthesized views, improving the overall compression performance in dynamic underwater environments.

- **Learning-based compression of difference images**

  Instead of using classical image codecs for compressing difference images, future research could develop learning-based compressors tailored specifically for the statistical properties of the difference images produced by NVSPrior. Such compressors could exploit the structured residual information to achieve higher compression ratios.

- **End-to-end joint optimization**

  A promising direction would be to jointly optimize the latent representation prediction and the difference image compression in an end-to-end manner. By directly optimizing for compression performance as the training objective, the system could learn to produce intermediate representations that are inherently more favorable for novel view synthesis and efficient transmission.

- **Latency management via predictive modeling**

  Ensuring low-latency feedback to remote operators is a key challenge in underwater environments, where acoustic communication introduces significant and unavoidable delays. Future work could incorporate a predictive model that simulates both the vehicle's hydrodynamic response and its surrounding environment in real time. By mirroring operator commands on this local model, the system can generate immediate visual feedback, effectively bypassing communication latency. The predicted outcomes can then be incrementally corrected using delayed data from the actual vehicle, enabling responsive control while preserving consistency with real-world observations.

- **Autonomous path planning and multi-robot collaboration**

  The efficient visual representation developed in this work can support higher-level autonomy tasks such as path planning and cooperative mapping. By enabling robots to perceive and interpret their environment in a compact and spatially consistent form, the framework opens possibilities for extending the current human-in-the-loop operation toward autonomous decision-making and multi-robot collaboration in marine environments.

Together, these directions highlight the rich potential for extending NVSPrior beyond its current scope. By addressing dynamic environments, environmental variability, and richer latent representations, the system

could evolve to support more adaptive and intelligent underwater perception. Moreover, incorporating learning-based compression, joint optimization, and predictive modeling may unlock unprecedented gains in efficiency and responsiveness. Advancing along these fronts will not only strengthen the technical foundations laid in this thesis but also move the field closer to realizing fully tetherless operation of ROVs in complex, real-world settings.

Bibliography

[1] F. Nauert and P. Kampmann, "Inspection and maintenance of industrial infrastructure with autonomous underwater robots", *Frontiers in Robotics and AI*, vol. 10, 2023.

[2] P. Ridao, M. Carreras, D. Ribas, and R. Garcia, "Visual inspection of hydroelectric dams using an autonomous underwater vehicle", *Journal of Field Robotics*, vol. 27, no. 6, pp. 759–778, 2010.

[3] D. Mcleod, J. Jacobson, M. Hardy, and C. Embry, "Autonomous inspection using an underwater 3d lidar", *2013 OCEANS - San Diego*, pp. 1–8, 2013.

[4] L. Zacchini, A. Topini, M. Franchi, N. Secciani, V. Manzari, L. Bazzarello, M. Stifani, and A. Ridolfi, "Autonomous underwater environment perceiving and modeling: An experimental campaign with feelhippo auv for forward looking sonar-based automatic target recognition and data association", *IEEE Journal of Oceanic Engineering*, vol. 48, pp. 277–296, 2023.

[5] U. von Ammon, S. Wood, O. Laroche, A. Zaiko, L. Tait, S. Lavery, G. Inglis, and X. Pochon, "The impact of artificial surfaces on marine bacterial and eukaryotic biofouling assemblages: A high-throughput

sequencing analysis." *Marine environmental research*, vol. 133, pp. 57–66, 2018.

[6] S. Aldhaheri, G. Masi, É. Pairet, and P. Ard'on, "Underwater robot manipulation: Advances, challenges and prospective ventures", *OCEANS 2022 - Chennai*, pp. 1–7, 2022.

[7] K. Shepherd, "Remotely operated vehicles (ROVs)*", in *Encyclopedia of Ocean Sciences (Second Edition)*, J. H. Steele, Ed., Oxford: Academic Press, Jan. 1, 2001, pp. 742–747, ISBN: 978-0-12-374473-9.

[8] A. Trembanis, M. Lundine, and K. McPherran, "Coastal mapping and monitoring", in *Encyclopedia of Geology (Second Edition)*, D. Alderton and S. A. Elias, Eds., Oxford: Academic Press, Jan. 1, 2021, pp. 251–266, ISBN: 978-0-08-102909-1.

[9] P. Aird, "Chapter 5 - deepwater: Essentials and differences", in *Deepwater Drilling*, P. Aird, Ed., Gulf Professional Publishing, Jan. 1, 2019, pp. 165–224, ISBN: 978-0-08-102282-5.

[10] B. Kalyan and M. A. Chitre, "Concept of operations for collaborative human robot inspection and intervention system in challenging underwater environments",, presented at the Offshore Technology Conference, May 1, 2023, D041S055R001.

[11] A. D. Bowen, M. V. Jakuba, N. E. Farr, J. Ware, C. Taylor, D. Gomez-Ibanez, C. R. Machado, and C. Pontbriand, "An un-tethered

rov for routine access and intervention in the deep sea", in *2013 OCEANS - San Diego*, 2013, pp. 1–7.

[12] H.-P. Tan, R. Diamant, W. K. G. Seah, and M. Waldmeyer, "A survey of techniques and challenges in underwater localization", *Ocean Engineering*, vol. 38, no. 14, pp. 1663–1676, Oct. 1, 2011, ISSN: 0029-8018.

[13] M. Stojanovic and J. Preisig, "Underwater acoustic communication channels: Propagation models and statistical characterization", *IEEE Communications Magazine*, vol. 47, no. 1, pp. 84–89, 2009.

[14] M. Chitre, S. Shahabudeen, L. Freitag, and M. Stojanovic, "Recent advances in underwater acoustic communications & networking", in *OCEANS 2008*, vol. 2008-Supplement, 2008, pp. 1–10.

[15] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng, "NeRF: Representing scenes as neural radiance fields for view synthesis", Aug. 3, 2020. arXiv: 2003.08934[cs].

[16] J. Alakuijala, R. Van Asseldonk, S. Boukortt, M. Bruse, I.-M. Comșa, M. Firsching, T. Fischbacher, E. Kliuchnikov, S. Gomez, R. Obryk, *et al.*, "Jpeg xl next-generation image compression architecture and coding tools", in *Applications of digital image processing XLII*, SPIE, vol. 11137, 2019, pp. 112–124.

[17] Google Developers. “An image format for the web | WebP”, (2024), [Online]. Available: https://developers.google.com/speed/webp.

[18] J. Ballé, D. Minnen, S. Singh, S. J. Hwang, and N. Johnston, “Variational image compression with a scale hyperprior”, in *International Conference on Learning Representations*, 2018.

[19] J. Ballé, V. Laparra, and E. P. Simoncelli, “End-to-end optimization of nonlinear transform codes for perceptual quality”, in *2016 Picture Coding Symposium (PCS)*, IEEE, 2016, pp. 1–5.

[20] G. Toderici, D. Vincent, N. Johnston, S. J. Hwang, D. Minnen, J. Shor, and M. Covell, “Full resolution image compression with recurrent neural networks”, in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI: IEEE, Jul. 2017, pp. 5435–5443, ISBN: 978-1-5386-0457-1.

[21] J. Whang, A. Acharya, H. Kim, and A. G. Dimakis, “Neural distributed source coding”, Number: arXiv:2106.02797, May 23, 2022. arXiv: 2106.02797[cs,math].

[22] K. Gregor, F. Besse, D. J. Rezende, I. Danihelka, and D. Wierstra, “Towards conceptual compression”, Apr. 29, 2016. arXiv: 1604.08772[cs,stat].

[23] A. Sento, “Image compression with auto-encoder algorithm using deep neural network (DNN)”, in *2016 Management and Innovation*

Technology International Conference (MITicon), Oct. 2016, MIT–99–MIT–103.

[24] L. Theis, W. Shi, A. Cunningham, and F. Huszár, "Lossy image compression with compressive autoencoders", *arXiv preprint arXiv:1703.00395*, 2017.

[25] D. Minnen, J. Ballé, and G. D. Toderici, "Joint autoregressive and hierarchical priors for learned image compression", in *Advances in Neural Information Processing Systems*, S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, Eds., vol. 31, Curran Associates, Inc., 2018.

[26] Z. Cheng, H. Sun, M. Takeuchi, and J. Katto, "Learned image compression with discretized gaussian mixture likelihoods and attention modules", in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 7936–7945.

[27] W. Jiang, J. Yang, Y. Zhai, F. Gao, and R. Wang, "Mlic++: Linear complexity multi-reference entropy modeling for learned image compression", *ACM Trans. Multimedia Comput. Commun. Appl.*, vol. 21, no. 5, May 2025, ISSN: 1551-6857.

[28] W. Jiang, J. Yang, Y. Zhai, F. Gao, and R. Wang, "Mlic++: Linear complexity multi-reference entropy modeling for learned image compression", *ACM Trans. Multimedia Comput. Commun. Appl.*, vol. 21, no. 5, May 2025, ISSN: 1551-6857.

[29] Q.-Z. Li and W. Wang, "Low-bit-rate coding of underwater color image using improved wavelet difference reduction", *J. Vis. Commun. Image Represent.*, vol. 21, pp. 762–769, 2010.

[30] Y. Zhang, S. Negahdaripour, and Q.-Z. Li, "Low bit-rate compression of underwater imagery based on adaptive hybrid wavelets and directional filter banks", *Signal Process. Image Commun.*, vol. 47, pp. 96–114, 2016.

[31] A. Tolstonogov and A. Shiryaev, "The image semantic compression method for underwater robotic applications", *OCEANS 2021: San Diego – Porto*, pp. 1–9, 2021.

[32] Z. Fang, L. Shen, M. Li, Z. Wang, and Y. Jin, "Priors guided extreme underwater image compression for machine vision and human vision", *IEEE Journal of Oceanic Engineering*, vol. 48, no. 3, pp. 888–902, 2023.

[33] Z. Fang, L. Shen, M. Li, Z. Wang, and Y. Jin, "Prior-guided contrastive image compression for underwater machine vision", *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 33, no. 6, pp. 2950–2961, 2023.

[34] F. Yuan, L. Zhan, P.-w. Pan, and E. Cheng, "Low bit-rate compression of underwater image based on human visual system", *Signal Process. Image Commun.*, vol. 91, p. 116 082, 2021.

[35] M. Li, L. Shen, Y. Lin, K. Wang, and J. Chen, "Extreme underwater image compression using physical priors", *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 33, no. 4, pp. 1937–1951, 2023.

[36] M. Li, L. Shen, P. Ye, G. Feng, and Z. Wang, "Rfd-ecnet: Extreme underwater image compression with reference to feature dictionary", in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, Oct. 2023, pp. 12 980–12 989.

[37] M. Li, L. Shen, X. Hua, and Z. Tian, "Euicn: An efficient underwater image compression network", *IEEE Transactions on Circuits and Systems for Video Technology*, 2024.

[38] Y. Ying, W. Zhu, H. Shang, and L. Song, "Dense scene 3d reconstruction based on semantic information of indoor environment: A review", *2021 International Conference on Networking Systems of AI (INSAI)*, pp. 110–117, 2021.

[39] S. Seitz, B. Curless, J. Diebel, D. Scharstein, and R. Szeliski, "A comparison and evaluation of multi-view stereo reconstruction algorithms", *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, vol. 1, pp. 519–528, 2006.

[40] J. L. Schönberger and J.-M. Frahm, "Structure-from-motion revisited", in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[41] J. Wang, L. E. Hafi, A. Taniguchi, Y. Hagiwara, and T. Taniguchi, "Extending hologan by embedding image content into latent vectors for novel view synthesis", *2022 IEEE/SICE International Symposium on System Integration (SII)*, pp. 383–389, 2022.

[42] K. Liu, Q. Li, and G. Qiu, "Posegan: A pose-to-image translation framework for camera localization", *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 166, pp. 308–315, 2020, ISSN: 0924-2716.

[43] X. Chang, D. Chen, Q. Chen, T. Jia, and H. Wang, "View synthesis by shared conditional adversarial autoencoder", in *Target Recognition and Artificial Intelligence Summit Forum*, 2020.

[44] B. Kerbl, G. Kopanas, T. Leimkuehler, and G. Drettakis, "3d gaussian splatting for real-time radiance field rendering", *ACM Transactions on Graphics*, vol. 42, no. 4, pp. 1–14, Aug. 2023, ISSN: 0730-0301, 1557-7368.

[45] T. Müller, A. Evans, C. Schied, and A. Keller, "Instant neural graphics primitives with a multiresolution hash encoding", *ACM Trans. Graph.*, vol. 41, no. 4, 102:1–102:15, Jul. 2022.

[46] S. Sabour, S. Vora, D. Duckworth, I. Krasin, D. J. Fleet, and A. Tagliasacchi, "Robustnerf: Ignoring distractors with robust losses",

in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 20 626–20 636.

[47] A. Pumarola, E. Corona, G. Pons-Moll, and F. Moreno-Noguer, "D-NeRF: Neural Radiance Fields for Dynamic Scenes", en, in *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Nashville, TN, USA: IEEE, Jun. 2021, pp. 10 313–10 322, ISBN: 978-1-66544-509-2.

[48] R. Martin-Brualla, N. Radwan, M. S. M. Sajjadi, J. T. Barron, A. Dosovitskiy, and D. Duckworth, "NeRF in the Wild: Neural Radiance Fields for Unconstrained Photo Collections", en, in *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Nashville, TN, USA: IEEE, Jun. 2021, pp. 7206–7215, ISBN: 978-1-66544-509-2.

[49] S. Zhang, S. Zhao, D. An, J. Liu, H. Wang, Y. Feng, D. Li, and R. Zhao, "Visual SLAM for underwater vehicles: A survey", *Computer Science Review*, vol. 46, p. 100 510, Nov. 1, 2022, ISSN: 1574-0137.

[50] S. Kuutti, S. Fallah, K. Katsaros, M. Dianati, F. Mccullough, and A. Mouzakitis, "A survey of the state-of-the-art localization techniques and their potentials for autonomous vehicle applications", *IEEE Internet of Things Journal*, vol. 5, no. 2, pp. 829–846, Apr. 2018, ISSN: 2327-4662.

[51] A. D. Buchan, E. Solowjow, D.-A. Duecker, and E. Kreuzer, "Low-cost monocular localization with active markers for micro autonomous underwater vehicles", in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Vancouver, BC: IEEE, Sep. 2017, pp. 4181–4188, ISBN: 978-1-5386-2682-5.

[52] A. G. Chavez, C. A. Mueller, T. Doernbach, D. Chiarella, and A. Birk, "Robust gesture-based communication for underwater human-robot interaction in the context of search and rescue diver missions", *ArXiv*, vol. abs/1810.07122, 2018.

[53] D. Chiarella, M. Bibuli, G. Bruzzone, M. Caccia, A. Ranieri, E. Zereik, L. Marconi, and P. Cutugno, "Gesture-based language for diver-robot underwater interaction", in *OCEANS 2015 - Genova*, May 2015, pp. 1–9.

[54] B. Teixeira, H. Silva, A. Matos, and E. Silva, "Deep learning for underwater visual odometry estimation", *IEEE Access*, vol. 8, pp. 44 687–44 701, 2020.

[55] A. Bucci, L. Zacchini, M. Franchi, A. Ridolfi, and B. Allotta, "Comparison of feature detection and outlier removal strategies in a mono visual odometry algorithm for underwater navigation", *Applied Ocean Research*, 2022.

[56] M. Ferrera, J. Moras, P. Trouvé-Peloux, and V. Creuze, "Real-time monocular visual odometry for turbid and dynamic underwater environments", *Sensors*, vol. 19, no. 3, 2019, ISSN: 1424-8220.

[57] S. Wirth, P. L. N. Carrasco, and G. Codina, "Visual odometry for autonomous underwater vehicles", *2013 MTS/IEEE OCEANS - Bergen*, pp. 1–6, 2013.

[58] A. Burguera, F. Bonin-Font, and G. Oliver, "Trajectory-based visual localization in underwater surveying missions", *Sensors (Basel, Switzerland)*, vol. 15, pp. 1708–1735, 2015.

[59] E. Vargas, R. Scona, J. S. Willners, T. Luczynski, Y. Cao, S. Wang, and Y. R. Petillot, "Robust underwater visual SLAM fusing acoustic sensing", in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, Xi'an, China: IEEE, May 30, 2021, pp. 2140–2146, ISBN: 978-1-72819-077-8.

[60] T. Sattler, B. Leibe, and L. Kobbelt, "Efficient & effective prioritized matching for large-scale image-based localization", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 9, pp. 1744–1756, Sep. 1, 2017, ISSN: 0162-8828, 2160-9292.

[61] A. Kendall, M. Grimes, and R. Cipolla, "PoseNet: A convolutional network for real-time 6-DOF camera relocalization", in *2015 IEEE International Conference on Computer Vision (ICCV)*, ISSN: 2380-7504, Dec. 2015, pp. 2938–2946.

[62] F. Walch, C. Hazirbas, L. Leal-Taixe, T. Sattler, S. Hilsenbeck, and D. Cremers, "Image-based localization using LSTMs for structured feature correlation",, p. 11,

[63] Y. Shavit and R. Ferens, "Introduction to camera pose estimation with deep learning", 2019. arXiv: 1907.05272 [cs.CV].

[64] E. Brachmann, A. Krull, S. Nowozin, J. Shotton, F. Michel, S. Gumhold, and C. Rother, "Dsac — differentiable ransac for camera localization", in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 2492–2500.

[65] R. Clark, S. Wang, A. Markham, N. Trigoni, and H. Wen, "Vidloc: A deep spatio-temporal model for 6-dof video-clip relocalization", in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 2652–2660.

[66] Z. Xiao, C. Chen, S. Yang, and W. Wei, "Effloc: Lightweight vision transformer for efficient 6-dof camera relocalization", in *2024 IEEE International Conference on Robotics and Automation (ICRA)*, 2024, pp. 8529–8536.

[67] S. Brahmbhatt, J. Gu, K. Kim, J. Hays, and J. Kautz, "Geometry-aware learning of maps for camera localization", *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2616–2625, 2017.

[68] M. C. Nielsen, M. H. Leonhardsen, and I. Schjolberg, "Evaluation of PoseNet for 6-DOF underwater pose estimation", in *OCEANS 2019 MTS/IEEE SEATTLE*, Seattle, WA, USA: IEEE, Oct. 2019, pp. 1–6, ISBN: 978-0-578-57618-3.

[69] L. Yen-Chen, P. Florence, J. T. Barron, A. Rodriguez, P. Isola, and T.-Y. Lin, "iNeRF: Inverting neural radiance fields for pose estimation", in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, ISSN: 2153-0866, Sep. 2021, pp. 1323–1330.

[70] Y. Sun, X. Wang, Y. Zhang, J. Zhang, C. Jiang, Y. Guo, and F. Wang, "iComMa: Inverting 3d gaussian splatting for camera pose estimation via comparing and matching", version: 2, Mar. 20, 2024. arXiv: 2312.09031[cs].

[71] C. Liu, S. Chen, Y. Bhalgat, S. Hu, M. Cheng, Z. Wang, V. A. Prisacariu, and T. Braud, "Gs-cpr: Efficient camera pose refinement via 3d gaussian splatting", 2025. arXiv: 2408.11085 [cs.CV].

[72] K. Botashev, V. Pyatov, G. Ferrer, and S. Lefkimmiatis, "Gsloc: Visual localization with 3d gaussian splatting", in *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2024, pp. 5664–5671.

[73] J. Lee and T. Bretl, "Gsfeatloc: Visual localization using feature correspondence on 3d gaussian splatting", 2025. arXiv: 2504.20379 [cs.CV].

[74] S. Chen, Z. Wang, and V. Prisacariu, "Direct-posenet: Absolute pose regression with photometric consistency", *2021 International Conference on 3D Vision (3DV)*, pp. 1175–1185, 2021.

[75] S. Chen, X. Li, Z. Wang, and V. A. Prisacariu, "Dfnet: Enhance absolute pose regression with direct feature matching", in *European Conference on Computer Vision*, 2022.

[76] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization", 2017. arXiv: 1412.6980 [cs.LG].

[77] R. Mishra, M. Chitre, B. Kalyan, Y. M. Too, H. Vishnu, and L. Peng, "An architecture for virtual tethering of rovs", in *2024 OCEANS-MTS/IEEE Singapore*, IEEE, 2024.

[78] L. Peng, M. Chitre, H. Vishnu, Y. M. Too, B. Kalyan, R. Mishra, and S. P. Tan, "Image compression using novel view synthesis priors", *IEEE Journal of Oceanic Engineering*, 2025, Under Review.

[79] Y. M. Too, H. Vishnu, M. Chitre, B. Kalyan, L. Peng, and R. Mishra, "A feasibility study on novel view synthesis of underwater structures using neural radiance fields", in *OCEANS 2024 - Singapore*, 2024, pp. 1–5.

[80] R. Mishra, B. Kalyan, M. Chitre, Y. M. Too, T. Soo Pieng, H. Vishnu, and L. Peng, "Design and demonstration of a wireless hybrid auv/rov for subsea inspections", in *2025 OCEANS-MTS/IEEE Brest*, IEEE, 2025.

[81] "TCOMS Research & Development",

[82] A. Arnaubec and R. Ewen, "Torpedo boat wreck (mediterranean, 43.124n;6.523e): Imagery and 3d model", SEANOE, `https://doi.org/10.17882/79028`, 2021.

[83] A. Arnaubec, M. Ferrera, J. Escartín, M. Matabos, N. Gracias, and J. Opderbecke, "Underwater 3d reconstruction from video or still imagery: Matisse and 3dmetrics processing and exploitation software", *Journal of Marine Science and Engineering*, vol. 11, no. 5, p. 985, 2023.

[84] M. Tancik, E. Weber, E. Ng, R. Li, B. Yi, J. Kerr, T. Wang, A. Kristoffersen, J. Austin, K. Salahi, A. Ahuja, D. McAllister, and A. Kanazawa, "Nerfstudio: A modular framework for neural radiance field development", in *ACM SIGGRAPH 2023 Conference Proceedings*, ser. SIGGRAPH '23, 2023.

[85] J. L. Schönberger, E. Zheng, M. Pollefeys, and J.-M. Frahm, "Pixelwise view selection for unstructured multi-view stereo", in *European Conference on Computer Vision (ECCV)*, 2016.

[86] Z. Wang, E. P. Simoncelli, and A. C. Bovik, "Multiscale structural similarity for image quality assessment", in *The Thrity-Seventh Asilomar Conference on Signals, Systems & Computers, 2003*, Ieee, vol. 2, 2003, pp. 1398–1402.

[87] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, "The unreasonable effectiveness of deep features as a perceptual metric", in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 586–595.

[88] T. Xie, Z. Zong, Y. Qiu, X. Li, Y. Feng, Y. Yang, and C. Jiang, "Physgaussian: Physics-integrated 3d gaussians for generative dynamics", in *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024, pp. 4389–4398.

[89] L. Peng and M. Chitre, "Regressing poses from monocular images in an underwater environment", in *OCEANS 2022 - Chennai*, Feb. 2022, pp. 1–4.

[90] L. Peng, H. Vishnu, M. Chitre, Y. M. Too, B. Kalyan, and R. Mishra, "Improved image-based pose regressor models for underwater environments", in *AUV Symposium 2022 - Singapore*, 2022, pp. 1–3.

[91] L. Peng, H. Vishnu, M. Chitre, Y. M. Too, B. Kalyan, R. Mishra, and S. P. Tan, "Pose estimation from camera images for underwater inspection", *IEEE Journal of Oceanic Engineering*, 2025.

[92] A. Kendall and R. Cipolla, "Geometric loss functions for camera pose regression with deep learning", in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI: IEEE, Jul. 2017, pp. 6555–6564, ISBN: 978-1-5386-0457-1.

[93] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, " Going deeper with convolutions ", in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Los Alamitos, CA, USA: IEEE Computer Society, Jun. 2015, pp. 1–9.

[94] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A Large-Scale Hierarchical Image Database", in *CVPR09*, 2009.

[95] A. Chaudhary, R. Mishra, B. Kalyan, and M. Chitre, "Development of an underwater simulator using unity3d and robot operating system", in *OCEANS 2021: San Diego – Porto*, ISSN: 0197-7385, Sep. 2021, pp. 1–7.

[96] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions", in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, ISSN: 1063-6919, Jun. 2015, pp. 1–9.

[97] R. Liaw, E. Liang, R. Nishihara, P. Moritz, J. E. Gonzalez, and I. Stoica, "Tune: A research platform for distributed model selection and training", *arXiv preprint arXiv:1807.05118*, 2018.

[98] L. Li, K. Jamieson, A. Rostamizadeh, K. Gonina, M. Hardt, B. Recht, and A. Talwalkar, "Massively parallel hyperparameter tuning", 2018.

[99] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition", *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[100] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition", in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.

[101] E. Bernardes and S. Viollet, "Quaternion to Euler angles conversion: A direct, general and computationally efficient method", *PLOS ONE*, vol. 17, no. 11, e0276302, Nov. 10, 2022, ISSN: 1932-6203.

[102] J. L. Schönberger, E. Zheng, J.-M. Frahm, and M. Pollefeys, "Pixel-wise view selection for unstructured multi-view stereo", in *Computer Vision – ECCV 2016*, B. Leibe, J. Matas, N. Sebe, and M. Welling, Eds., Cham: Springer International Publishing, 2016, pp. 501–518, ISBN: 978-3-319-46487-9.

[103] A. Kendall and R. Cipolla, "Modelling uncertainty in deep learning for camera relocalization", in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, 2016, pp. 4762–4769.

[104] C. G. Harris and M. J. Stephens, "A combined corner and edge detector", in *Alvey Vision Conference*, 1988.

[105] A. Alahi, R. Ortiz, and P. Vandergheynst, "Freak: Fast retina keypoint", in *2012 IEEE Conference on Computer Vision and Pattern Recognition*, 2012, pp. 510–517.

[106] M. A. Fischler and R. C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography", *Commun. ACM*, vol. 24, no. 6, pp. 381–395, Jun. 1981, ISSN: 0001-0782.

[107] V. Ye and A. Kanazawa, "Mathematical supplement for the `gsplat` library", 2023. arXiv: `2312.02121 [cs.MS]`.

[108] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition", *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[109] J. Sun, Z. Shen, Y. Wang, H. Bao, and X. Zhou, "Loftr: Detector-free local feature matching with transformers", in *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 8918–8927.

[110] C. G. Broyden, "The Convergence of a Class of Double-rank Minimization Algorithms 1. General Considerations", *IMA Journal of Applied Mathematics*, vol. 6, no. 1, pp. 76–90, Mar. 1970, ISSN:

0272-4960. eprint: https://academic.oup.com/imamat/article-pdf/6/1/76/2233756/6-1-76.pdf.

[111] R. Fletcher, "A new approach to variable metric algorithms", *The Computer Journal*, vol. 13, no. 3, pp. 317–322, Jan. 1970, ISSN: 0010-4620. eprint: https://academic.oup.com/comjnl/article-pdf/13/3/317/988678/130317.pdf.

[112] D. Goldfarb, "A family of variable-metric methods derived by variational means", *Mathematics of Computation*, vol. 24, no. 109, pp. 23–26, 1970, ISSN: 00255718, 10886842.

[113] D. F. Shanno, "Conditioning of quasi-newton methods for function minimization", *Mathematics of Computation*, vol. 24, no. 111, pp. 647–656, 1970, ISSN: 00255718, 10886842.

[114] R. Feinman, "Pytorch-minimize: A library for numerical optimization with autograd", 2021.

[115] J. Bégaint, F. Racapé, S. Feltman, and A. Pushparaja, "Compressai: A pytorch library and evaluation platform for end-to-end compression research", *arXiv preprint arXiv:2011.03029*, 2020.

[116] J. Gamper, H. G. Gallmetzer, A. K. Weiss, and T. S. Hofer, "A general strategy for improving the performance of pinns – analytical gradients and advanced optimizers in the neuralschrödinger framework", *Artificial Intelligence Chemistry*, 2024.

[117] V. Dharanalakota, P. K. J, and P. K. Ghosh, "Loss-based optimizer switching to solve 1-d helmholtz equation using neural networks", *The Journal of the Acoustical Society of America*, 2023.

[118] M. Tan and Q. V. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks", 2020. arXiv: `1905.11946 [cs.LG]`.

[119] G. Bjøntegaard, "Calculation of average psnr differences between rd-curves",, 2001.

[120] A. Crivellaro and V. Lepetit, "Robust 3d tracking with descriptor fields", in *2014 IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 3414–3421.

Publications during PhD Study

[1] L. Peng and M. Chitre, "Regressing poses from monocular images in an underwater environment", in *OCEANS 2022 - Chennai*, Feb. 2022, pp. 1–4.

[2] L. Peng, H. Vishnu, M. Chitre, Y. M. Too, B. Kalyan, and R. Mishra, "Improved image-based pose regressor models for underwater environments", in *AUV Symposium 2022 - Singapore*, 2022, pp. 1–3.

[3] L. Peng, H. Vishnu, M. Chitre, Y. M. Too, B. Kalyan, R. Mishra, and S. P. Tan, "Pose estimation from camera images for underwater inspection", *IEEE Journal of Oceanic Engineering*, 2025.

[4] L. Peng, M. Chitre, H. Vishnu, Y. M. Too, B. Kalyan, R. Mishra, and S. P. Tan, "Image compression using novel view synthesis priors", *IEEE Journal of Oceanic Engineering*, 2025, Under Review.

[5] Y. M. Too, H. Vishnu, M. Chitre, B. Kalyan, L. Peng, and R. Mishra, "A feasibility study on novel view synthesis of underwater structures using neural radiance fields", in *OCEANS 2024 - Singapore*, 2024, pp. 1–5.

[6] R. Mishra, M. Chitre, B. Kalyan, Y. M. Too, H. Vishnu, and L. Peng, "An architecture for virtual tethering of rovs", in *2024 OCEANS-MTS/IEEE Singapore*, IEEE, 2024.

[7] R. Mishra, B. Kalyan, M. Chitre, Y. M. Too, T. Soo Pieng, H. Vishnu, and L. Peng, "Design and demonstration of a wireless hybrid auv/rov for subsea inspections", in *2025 OCEANS-MTS/IEEE Brest*, IEEE, 2025.

Appendix A

Pose Alignment Between Sensor and COLMAP Coordinate Frames

To align the coordinate frames of the onboard navigation system and the Splatfacto reconstruction, we compute a similarity transformation using paired camera poses from each system. Let $\mathbf{P}_{\text{ROV}}^{(i)} \in \mathbb{R}^{4\times4}$ and $\mathbf{P}_{\text{COLMAP}}^{(i)} \in \mathbb{R}^{4\times4}$ denote the i-th pose from the onboard navigation system and COLMAP, respectively. We extract their position components, denoted $\mathbf{p}_{\text{ROV}}^{(i)}, \mathbf{p}_{\text{COLMAP}}^{(i)} \in \mathbb{R}^3$.

The similarity transformation is parameterized by:

$$\mathbf{x} = [s, \alpha, \beta, \gamma, x, y, z]^T$$

where $s \in \mathbb{R}^+$ is a scale factor, α, β, γ are Euler angles, and (x, y, z) is a translation vector.

The corresponding 4×4 transformation matrix is defined as:

$$\mathbf{T}(\mathbf{x}) = \begin{bmatrix} s \cdot \mathbf{R}(\alpha, \beta, \gamma) & \mathbf{t} \\ \mathbf{0}^T & 1 \end{bmatrix}$$

where $\mathbf{R}(\alpha, \beta, \gamma) \in \mathrm{SO}(3)$ is the rotation matrix derived from the Euler angles, and $\mathbf{t} = [x, y, z]^T$.

Using homogeneous coordinates

$$\tilde{\mathbf{p}} = \begin{bmatrix} \mathbf{p} \\ 1 \end{bmatrix},$$

the cost function is defined as:

$$J(\mathbf{x}) = \sum_i \left\| \mathbf{T}(\mathbf{x})\, \tilde{\mathbf{p}}^{(i)}_{\mathrm{ROV}} - \tilde{\mathbf{p}}^{(i)}_{\mathrm{COLMAP}} \right\|^2$$

The optimal parameters $\mathbf{x}^*$ are obtained via nonlinear optimization. Once the transformation is estimated, it is applied to the full ROV pose matrices as:

$$\mathbf{P}^{(i)}_{\mathrm{aligned}} = \mathbf{T}(\mathbf{x}^*) \cdot \mathbf{P}^{(i)}_{\mathrm{ROV}} \cdot \mathbf{C}$$

where $\mathbf{C}$ is a fixed coordinate exchange matrix used to account for camera convention differences between the two systems:

$$\mathbf{C} = \begin{bmatrix} 0 & 0 & -1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$