Master's thesis

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

Md Raqibur Rahman

Self Supervised Scale Consistent Depth and Ego-motion Learning From Monocular Video For Underwater Robots

Master's thesis in Marine and Maritime Intelligent Robotics (MSMIR)
Supervisor: Prof. Martin Ludvigsen
Co-supervisor: Prof. Ricard Marxer, Maxime Ferrera

June 2024

NTNU
Norwegian University of
Science and Technology

MIR
MARINE &
MARITIME
INTELLIGENT
ROBOTICS

Md Raqibur Rahman

Self Supervised Scale Consistent Depth and Ego-motion
Learning From Monocular Video For Underwater Robots

Master’s thesis in Marine and Maritime Intelligent Robotics (MSMIR)
Supervisor: Prof. Martin Ludvigsen
Co-supervisor: Prof. Ricard Marxer, Maxime Ferrera
June 2024

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

NTNU
Norwegian University of
Science and Technology

Self Supervised Scale Consistent Depth and Ego-motion Learning from Monocular Video for underwater robots

Md Raqibur Rahman

June 10, 2024

Abstract

Being cost effective, safe and portable, Unmanned Underwater Vehicle (UUV)s are becoming popular for underwater exploration. Navigation in underwater environments is challenging due to the fact that electromagnetic waves do not transmit far underwater. Vision based navigation and mapping can be very useful in this environment for being cheap and easily accessible. In this project, the feasibility of using a self-supervised based method for scene depth and robot pose learning from underwater videos has been studied. Three different depth prediction models Dispnet, Udepth and Dispnet Mvit have been trained with two different input spaces (Red-Green-Blue (RGB) and Red-Max-Intensity (RMI)) along with a pose prediction model. The models have been trained on 3 different years (2016, 2018 and 2020) data from the underwater Eiffel tower dataset and have been tested with data from the year 2015 of that same dataset. For testing the model's generalizability the trained model is also tested with the Varos dataset. The predicted depths has been used to enhance images using the SeaThru pipeline. The udepth model with RMI input space has achieved the best depth prediction result on the eiffel tower dataset with an Root Mean Squared Error (RMSE) of 2.5583m when the maximum depth has been capped at 40m. While on varos dataset the dispnet mvit model with RGB input space performed the best with RMSE of 8.1343m when the maximum depth has been capped at 60m depth. For underwater image enhancement using SeaThru pipeline the dispnet with RMI input space achieved the best performance in terms of Underwater Image Quality Measure (UIQM) of 1.66 which is an 26.68% increase compared to the original images in the eiffel tower dataset. However, in the varos dataset the dispnet model with RGB input space achieved an UIQM of 0.50, a 305.03% of increase from the original images. The Varos dataset being a simulated dataset retains pixel level information even in the shadows of the images, which aids in enhancing the images using the SeaThru pipeline. This makes the enhanced images gain a 305.03% increase in UIQM compared to the original images. To get a better understanding of the image enhancement performance, a new dataset SeaThru-Nerf, consisting of real underwater images, has been used. On the SeaThru-Nerf dataset the dispnet model with RGB input space performed best for image enhancement in all scenes in terms of UIQM. It achieved UIQM of 2.06 a16.09% increase in Curasao scene, 2.01 an increase of 22.27% in Panama scene, 1.21 a 36.32% increase in IUI3-RedSea scene and 1.67 a increase of 29.43% in JapaneseGradens-RedSea scene. The predicted depths from the Udepth (RMI) model on the Eiffel-Tower dataset are used in RGB-D SLAM pipeline in the ORB-SLAM3 framework as a depth sensor. The estimated trajectory from the SLAM yielded an RMSE of 9.79m Absolute Trajectory Error (ATE) when averaged over 3 different runs, which is 2.14% of the total trajectory length. For the pose net when trained with Dispnet along with RMI input space achieved an Absolute Trajectory Error (ATE) of 1532.216m which is 0.311m per frame in the Eiffel tower dataset when a 5-frame snippet has been used to align and scale the trajectory. In Varos dataset the best performing pose net was trained with Dispnet mVit with RMI input space. It achieved a total ATE of 25.84997m and mean ATE of 0.0042m per frame. From the experiments, it can be said the self-supervised learning based scene depth and ego motion learning from videos, which was originally proposed for airborne vision, can be applied in underwater environments where the visual appearance of the scene is significantly different and this also changes with the viewpoint of the camera. Moreover, it can be also concluded that the predicted depths can be used in several application where the depth is required and not readily available from the sensor, like a pseudo depth sensor in RGB-D SLAM and underwater image enhancement techniques like SeaThru.

Preface

This thesis marks the completion of my Erasmus Mundus Joint Masters degree in Marine and Maritime Intelligent Robotics at the Norwegian University of Science and Technology (NTNU) in Norway and Universite de Toulon (UTLN) in France. The thesis was conducted during the spring of 2023.

I would like to thank my supervisors, Martin Ludvigsen from NTNU in Norway, Ricard Marxer from UTLN in France and Maxime Ferrera from IFREMER in France for their guidance and advises throughout the research. I would like to thank them for answering all my questions, providing necessary insights and knowledge.

Moreover, I would like to express my gratitude to the European Commission for awarding me the Erasmus Mundus scholarship, which has enabled me to pursue a master's degree in Marine and Maritime Intelligent Robotics (MSMIR). This thesis project has been an integral part of my studies within the MSMIR program, and I am thankful for the opportunity to delve into this research area through the support of the scholarship.

Finally, I would like to thank my family and friends for their continuous and endless support.

Contents

Figures

Tables

Acronyms

ATE Absolute Trajectory Error. iv, 20, 30, 32, 34, 35

AUV Autonomous Underwater Vehicle. 1

BoW Bag of Words. 6

CNN Convolutional Neural Network. 6–8, 12

DOF Degrees of Freedom. 12, 15

EIF Extended Information Filter. 5

EKF Extended Kalman Filter. 5

GAN Generative Adversarial Network. 7

GNSS Global Navigation Satellite System. 1

GPS Global Positioning System. 1, 2

IMU Inertial Measurement Unit. 11, 46

LiDAR Light Detection and Ranging. 2, 6

MVS Multi View Stereo. 7

PF Particle Filter. 5

RGB Red-Green-Blue. iii, 7, 15–17, 23, 26, 28, 29, 32, 37, 38, 40–42, 45

RMI Red-Max-Intensity. iii, iv, 7, 15–17, 23, 26, 27, 29, 32, 37, 38, 40, 42, 45

RMSE Root Mean Squared Error. iii, 24–27

ROV Remotely Operated Vehicles. 1, 2

SBL Short Baseline. 1

SFM Structure From Motion. ix, 6, 7, 10, 12, 16–19, 24

SLAM Simultaneous Localization and Mapping. 1, 2, 5, 18, 45, 46

SSIM Structural Similarity Index Measure. 14

UCIQE Underwater Colour Image Quality Evaluation. 21, 37, 38, 40–43

UICM Underwater Image Colorfulness Measure. 20

UIConM Underwater Image Contrast Measure. 20

UIQM Underwater Image Quality Measure. iii, 20, 37, 38, 40–43

UISM Underwater Image Sharpness Measure. 20

USBL Ultrashort Baseline. 1

UUV Unmanned Underwater Vehicle. iii, 1, 2

VSLAM Visual Simultaneous Localization and Mapping. vii, 2, 3, 5

Chapter 1

Introduction

1.1 Background

In the realm of autonomous robotics, Simultaneous Localization and Mapping (SLAM) has become a pivotal focus in the 21st century. The ever growing importance of applied research on autonomous positioning for mobile robots has driven the exploration of innovative solutions, especially in environments where traditional methods like the Global Navigation Satellite System (GNSS) may fall short. SLAM is a method that empowers robots to navigate and map unknown environments concurrently, offering a cost-effective and versatile alternative in various applications[1, 2].

Unmanned Underwater Vehicle (UUV)s have gained widespread popularity for underwater exploration, owing to their safety, portability, and cost-effectiveness. These vehicles fall into two main categories: Remotely Operated Vehicles (ROV)s and Autonomous Underwater Vehicle (AUV). UUVs play a crucial role in marine resource investigation, undersea biology research, underwater structure detection, and marine data collection. Knowing the position of the vessel accurately is of utmost importance in these applications. However, achieving precise positioning and navigation for underwater vehicles is challenging due to the rapid attenuation of electromagnetic signals, such as Global Positioning System (GPS), by the underwater environment. Additionally, the inertial navigation approach is susceptible to accumulating errors overtime rendering it useless on it's own[3]. Traditional underwater acoustic positioning methods like Short Baseline (SBL) and Ultrashort Baseline (USBL) involve installing a base array or periodic AUV position corrections. While effective, these methods are expensive, have limited exploration range due to beacons, and often require frequent surfacing, leading to increased exertion for the vehicles[1, 4]. To tackle these issues, researchers are exploring the application of SLAM techniques in the underwater domain, opening up new possibilities for autonomous positioning and navigation of underwater vehicles.[5]

Underwater SLAM can be broadly categorized into Light Detection and Ranging (LiDAR) SLAM, sonar SLAM, and Visual Simultaneous Localization and Mapping (VSLAM) based on sensors used. LiDARs and sonars are expensive, limiting their use for civil robots. Moreover, LiDARs underwater range is restricted due to laser absorption and scattering by particles in the water, resulting in maps lacking semantic information[6]. Although, sonar can be a suitable choice for underwater SLAM, yet it is influenced by water flow, seismic activity, and other factors, posing challenges. Specially in enclosed environments where sonar signals gets reflected and interfere with each other[7]. In contrast, recently the low-cost and portable vision based systems have gained prominence. Although it faces issues with suspended particles in water and scene illumination, it can be mitigated by various underwater image enhancement algorithms [6, 8, 9]. Enhancing the underwater images is an important step of using underwater images in different computer vision application such as SLAM. Despite the advantages of VSLAM, adaptation in general is low in the context of underwater scenario. A reason for this low adaptation is the lack of suitable annotated dataset with ground truth pose of the camera in the GPS denied environment [10]. Another reason being the lack of ambient light in the deep sea and the UUVs need to bring their own light source for illuminating the scene. As a result, when the vessel is further away from the object of interest the illumination changes and that often causes the traditional vision based of the shelf systems like feature matching and estimating fundamental matrix between images to fail. Recent developments in the field of deep learning based computer vision has addressed the problem of lack of annotated data by using self supervised learning. [11] first introduced this method for learning scene depth and camera pose simultaneously from unannotated videos. Later on research like [12–15] improved upon the results.

1.2 Objective

In this project, the objective is to investigate the validity of adapting the self supervised method of learning scene depth and robot pose in the world for leveraging the power of Deep Learning in the deep sea underwater videos captured by ROVs. In this regard the method proposed by [12] will be used for learning pose and scale consistent depth. To improve upon the results, various depth prediction models will be tested within the learning pipeline. The models will be evaluated on a different dataset to check their generalizability to different scenes. Finally the estimated depths will be used in two different applications: improving the underwater image quality and in traditional VSLAM algorithms.

1.3 Scope

In the context of the objective described in the previous section the scopes of the project are:

- Review of relevant theory and existing work on underwater image formation, photogrammetric camera modelling as well as self supervised learning of scene depth and camera pose.
- Collect suitable underwater datasets to train and test the model on.
- Experiment with suitable network architectures for underwater applications.
- Applying the prediction of the depth network in the state of the art VSLAM system ORB-SLAM3 and compare the performances.
- Using the predicted depths in the image enhancement pipelines for improving the image quality.

1.4 Outline

The rest of the thesis is structured as: Chapter 2 contains relevant literature review. Chapter 3 contains a detailed discussion on the materials and methods used in the research. Chapter 4 presents and discusses the result of the research. Chapter 5 summarizes the result and concludes the project objective.

Parts of the chapter 1, chapter 2 and chapter 3.1.2 has been taken from the project report of the course TMR4510 Marine Control Systems, Specialization Project.

Chapter 2

Related Works

In this chapter, a comprehensive review of relevant works is presented that contributed to our methods.

2.1 Traditional VSLAM

The Visual SLAM can be widely categorized in direct and indirect methods. The direct methods use the images directly for estimating map points and poses of the robot where indirect methods include an extra step of computing features from the images before going to the estimation step. Again based on the sensors used in the Visual SLAM it can be categorized in Monocular, Stereo, RGB-D, Monocular-Inertial SLAM [1].

[6] suggested that, in underwater environments, optical flow based direct methods are impractical for visual SLAM due to disturbance and unstable light sources. Feature point methods, specifically those employing key points, are commonly used for front-end calculations in underwater visual SLAM. Notable feature detection and matching methods include SIFT, SURF, and ORB, each designed to identify key points with robustness to image scaling and rotation. ORB is recommended for high real-time requirements, while SIFT and SURF are suitable for high-performance scenarios[16]. Selecting appropriate features is crucial for SLAM problem-solving, with local image feature detection and matching methods enhancing efficiency[17]. Segmenting the target image into background and Region of Interest (RoI) and performing detection and matching solely in the RoI region is an effective approach for local image feature enhancement[18].

Early visual SLAM algorithms utilized filtering methods like Extended Kalman Filter (EKF), Particle Filter (PF), and Extended Information Filter (EIF)[19–21]. The optimization visual SLAM algorithms, considering historical pose and landmark data, excels in large-scale and prolonged scenes[22]. Methods like graph optimization, pose graph, factor graphs, tightly coupled nonlinear optimization, are utilized by ORB-SLAM [23] and its upgrades. Factor graphs model the SLAM problem and are optimized through nonlinear least squares. Approaches such as [24] and [25] integrate data from various sensors in the cost function, optimizing

the system with a tightly coupled nonlinear method. ORB-SLAM2 introduces pose constraints in the estimation process [26], while based on ORB-SLAM3, [27] incorporates visual-acoustic joint optimization in both tracking and local mapping threads, replacing the original vision-only bundle adjustment(BA).
[28] demonstrated that ORB-SLAM could be used effectively under the conditions of a sufficient illumination, low flicker, and rich scene features.[29] used OpenVSLAM algorithm, an algorithm inspired by indirect, sparse graph-based V-SLAM algorithms ORB-SLAM, ORB-SLAM2, ProSLAM, and UcoSLAM, and concluded that Bag of Words Bag of Words (BoW) based methods are not suitable for loop closing underwater due to susceptibility to lighting condition.

2.2 Deep Learning Based Camera Pose Estimation

[30] introduced a Deep Convolutional Neural Network (CNN) model called posenet for predicting a camera pose from a single image. They showed the network learns to compute features which can be easily mapped to pose and generalize to unseen new scenes with a few additional training samples. [31] proposed a geometrically formed loss function in order to improve the performance of the posenet. [32] proposed a method for combining different loss functions to simultaneously learn multiple objectives. [33] concluded that despite being less accurate, these methods are far more robust to noise and easy to use. [34–37] combine the best of both purely geometry based methods and deep learning based methods for the camera pose estimation problem. [38] presented a method to predict robot to robot relative pose for underwater robots. [39] introduced a homography based loss function to properly weight the translation and rotation components in the final error in the pose prediction problem. They compute the error in $SE(3)$ which does not require prior initialization and depends on intuitive parameters.

2.3 Monocular Depth Estimation

[40] introduced an approach for estimating depth from single monocular images using sensor captured depths like Light Detection and Ranging (LiDAR) or RGB-D camera. First they estimated the global structure of the scene and then refined it using local information. [41] introduced new loss function incorporating prior knowledge of the ambiguity in the relation of dual pixel images with scene depth. [42, 43] introduced a new network architecture for improving depth prediction performance. While these methods achieve great performance, it is expensive to capture ground truth data in real world scenario specially in underwater environment. The sensors required for collecting the data are expensive and it is also very expensive to conduct such missions of capturing large scale datasets.

[44] proposed to generate scene depth using Structure From Motion (SFM) on internet videos. [45] addressed the problem of recovering dense geometry in a dynamic scene and proposed a method to predict accurate and dense depth from

videos where both camera and objects in the scene are naturally moving. Later they used the generated data to train a network. [46] used SFM and Multi View Stereo (MVS) to automatically generate depth on internet data and also introduced a new loss function for training depth networks to compensate for the fact that MVS does not tend to reconstruct dynamic objects in the scene. [47] proposed a network architecture to leverage stereo videos for training and predicting depths from monocular sequential frames with non rigid objects during inference time. They also introduced a new loss function to train with unknown camera parameters which outperforms previously used ordinal losses. Although using SFM and MVS can help obtaining cheap ground truth data, there usually exists domain gap between the collected data and the desired scene [12]. It is also difficult to generalize the scale information on different scenes. As a result the models predict relative depth. This causes inconsistency in the depths predicted on a video.

2.4 Depth Estimation on Underwater Images

Due to the number of constraints normally being less than the number of unknown variables, underwater image restoration and depth prediction are an ambiguous problem[48]. [49] uses a CNN to estimate depth map which in turn is used to perform image dehazing based on an atmospheric scattering model.

[50] proposed a new joint learning framework for underwater depth estimation and color correction in order to exploit the correlations between them and possibly benefit both tasks at the same time. [51] proposed a Generative Adversarial Network (GAN) based model to predict depth and use that to enhance underwater images. [52] showed that instead of using the Red-Green-Blue (RGB) images, if the input space is changed to Red-Max-Intensity (RMI) it assists in the task of predicting depth in the underwater images. The authors argue that, the difference of the intensities between the red channel and the maximum of blue and green channels encodes the scene depth information and can be exploited in the depth prediction for getting better depth estimation from the network.

2.5 Self-Supervised Depth and Pose Estimation

[11] first introduced a method to learn scene depth and pose simultaneously on unlabeled videos. [13] proposed a new model called GLNet to solve the interrelated tasks of monocular depth prediction, optical flow, camera pose and intrinsic estimation. [14] introduced a new appearance matching loss for addressing the problem of occluded pixels, an auto masking approach for ignoring pixels where no relative camera motion is observed and a multi scale appearance matching loss to perform all image sampling at the input resolution for reducing depth artifacts. [15] showed it is possible to train deep networks to predict camera intrinsic parameters including lens distortion in an unsupervised manner from videos. They also addressed the occlusion problem directly in a geometric way

from the predicted depth. [53] introduced a new CNN model for high resolution self-supervised depth estimation with new packing and unpacking blocks that jointly leverage 3D convolutions to learn representations that maximally propagate dense appearance and geometric information while being able to run real-time. They also introduced a new loss function to leverage the camera's velocity when available to solve the inherent scale ambiguity in monocular vision. [54] addressed the problem of dynamic objects using semantic guidance. [55] proposed a method to remove the pose net architecture by proposing a novel system that solves the fundamental matrix directly from dense optical flow correspondence and makes use of a two view triangulation module for recovering up to scale 3D structure. [12] introduced a geometric consistency loss in order to encourage the networks to predict scale consistent depths with higher accuracy. They penalize the pixel wise inconsistency in the predicted depths between adjacent frames during training. They also introduced a self-discovered mask for handling dynamic objects during training to be consistent with the assumption of static scene.

In this work, the method proposed by [12] is used to learn scale consistent scene depth and camera pose in an underwater environment.

Chapter 3

Materials & Methods

3.1 Datasets

3.1.1 Eiffel Tower

Eiffel Tower dataset was created to aid the long term visual localization task in a deep-sea environment. This dataset includes images of the same hydrothermal vent edifice, Eiffel Tower, located at 1700 m beneath the surface, taken during four visits over five years (2015, 2016, 2018 and 2020).[56]

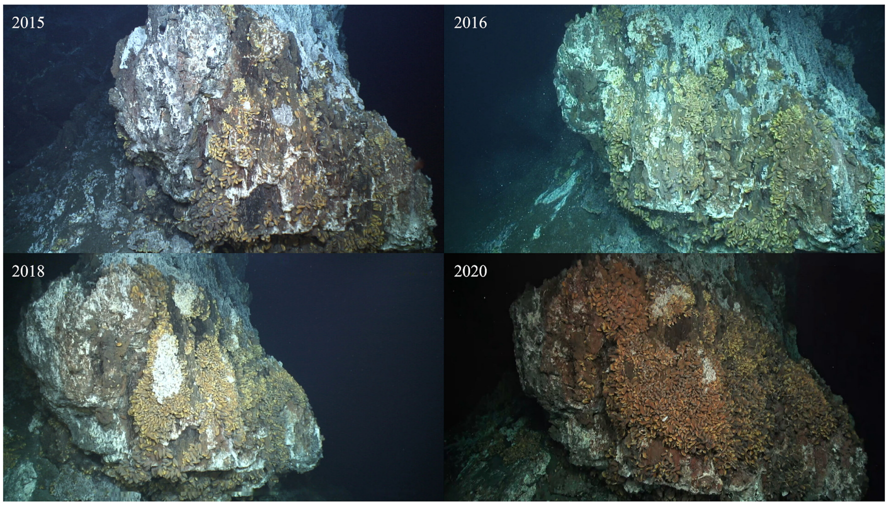


Figure 3.1: Image of the same place (south-east facade of the vent) in different years[56]

The Dataset includes the following:

1. Images taken of the vent during each four visits.
2. Position (latitude, longitude and altitude) of the camera during each image captured

3. 3D models of the vent site recreated for each year of visits using Structure From Motion (SFM).
4. A single, combined 3D model showing the entire vent site with all the images positioned in a consistent reference frame.

The dataset presents changes over time related to all the challenges in underwater imaging (light and color absorption, turbidity and back-scattering, strong differences in illumination depending on the distance of the robot and the scene due to the onboard artificial lighting system).

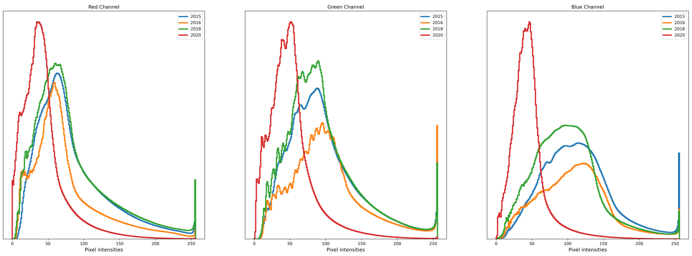


Figure 3.2: Pixel intensities histogram over different year for red, blue and green channel

The figure 3.1 shows the evolution of South-East facade of the hydrothermal vent. The significant change of the scene that happened over the years can be noticed in the figure. Figure 3.2 shows the differences of pixel intensities distribution of red, green, blue channels over the years. These changes overtime makes it difficult for traditional SLAM based algorithms to detect and match features among cross year images.

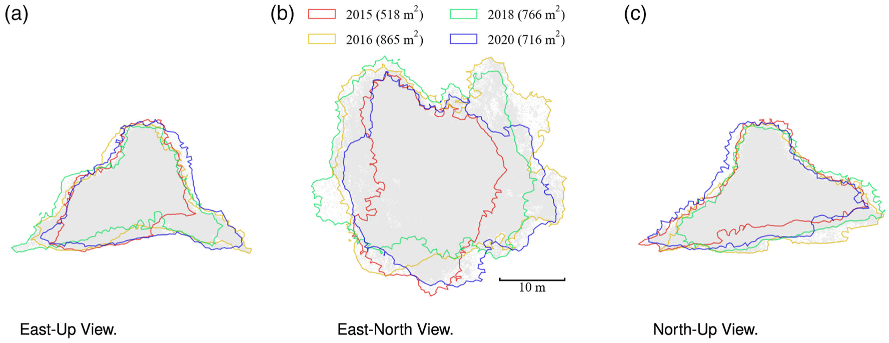


Figure 3.3: Area covered by the dataset images around the hydrothermal vent in different years[56]

Figure 3.3 shows the area covered by the image sequences in the dataset around the hydrothermal vent in different years. The author concluded from the figure that the total area covered in 2016, 2018 and 2020 images, contains almost all the area covered in 2015[56]. Hence, In this study the data from 2016, 2018 and 2020 are used to train the model and the data from 2015 are used to test the model.

3.1.2 Varos

To evaluate the model's performance in an unseen dataset we used the simulated underwater dataset Varos[57]. The dataset consists of 4 type of images, Underwater monocular RGB images, Uniformly illuminated monocular RGB images, Surface normal images, Depth images, along with Inertial Measurement Unit (IMU) data and Depth gauge data and ground truth trajectory. The image sequences contains 4714 images. Figure 3.4 shows different types of images of the same scene of the dataset.

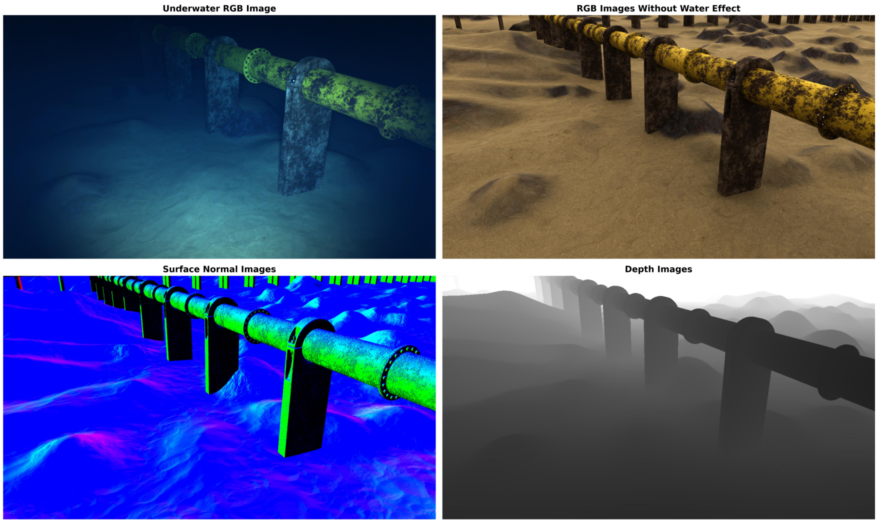


Figure 3.4: Different images of same scene in the Varos Dataset

The camera model and lens parameters used to export the images are given in table 3.1.

Parameters	Value
Render resolution (pixel)	1280 X 720
Sensor width [mm]	4.416
Sensor height [mm]	2.484
Shutter type	Global shutter
Focal length [mm]	3.4
Aperture	1.7

Table 3.1: Camera and lens parameters. [57]

From the given the camera parameters the intrinsic camera matrix calculated is in shown in equation 3.1

$$K = \begin{bmatrix} 985.5072 & 0 & 640 \\ 0 & 985.5072 & 360 \\ 0 & 0 & 1 \end{bmatrix} \tag{3.1}$$

3.1.3 SeaThru-Nerf

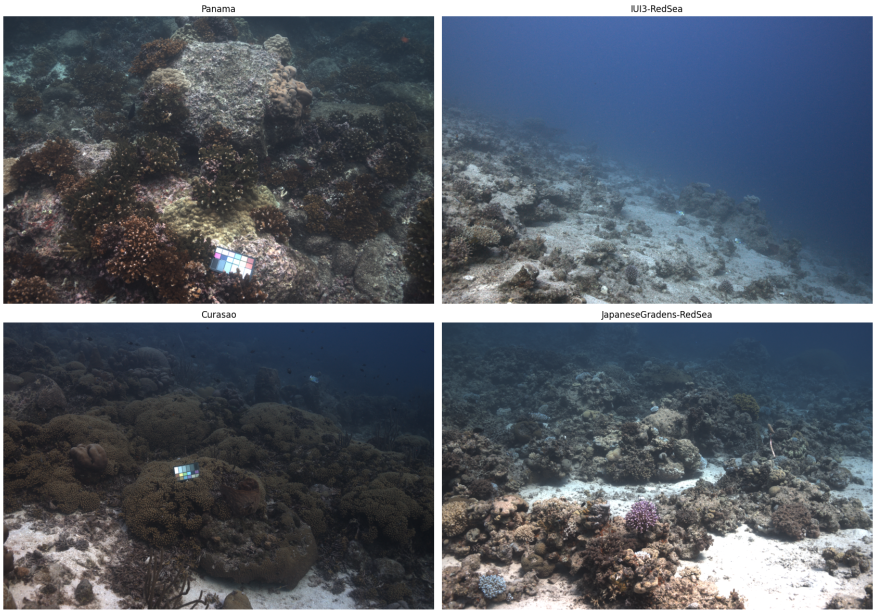


Figure 3.5: Example images of different scenes from SeaThru-Nerf Dataset

In order to evaluate model performance in a real world dataset, SeaThru-Nerf[58] dataset has been used. The dataset contains real world images of 4 scenes namely Panama, Curacao, IUI3-RedSea and JapaneseGradens-RedSea along with the SFM Model and respective image poses. Figure 3.11 shows example images from different scene of the dataset. The total number of images in each scene are 18, 21, 29 and 20 respectively.

3.2 SC-SFMLearner

3.2.1 Framework Overview

The aim of SFMLearner is to train depth and pose Convolutional Neural Network (CNN) using unlabeled image sequences. Figure 3.6 illustrates the SFMLearner pipeline. The process involves sampling two adjacent frames (I_a, I_b) from a video and estimating their depth maps (D_a, D_b) and relative 6Degrees of Freedom (DOF) camera pose P_{ab} using dedicated CNNs. With the predicted depth and pose, a

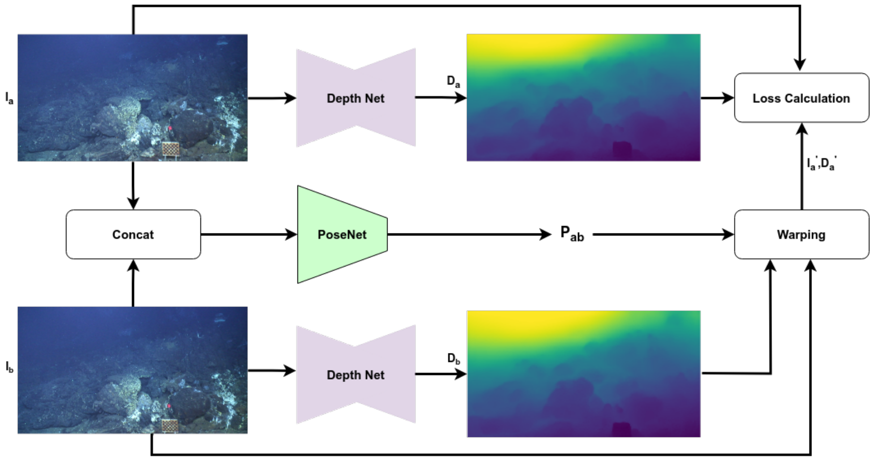


Figure 3.6: Illustration of the SFMLearner pipeline

reference image $I_a^{'}$ is synthesized from the source image I_b through differentiable warping. The network is then supervised by comparing the real image I_a with the synthesized image $I_a^{'}$ using a photometric loss L_p. To ensure scale-consistent depth predictions between adjacent frames,a geometry consistency loss L_G is introduced. For smoothness in the predicted depth, a depth smoothness loss L_s is also used. Additionally, to handle cases like static frames and dynamic objects, two masks: a self-discovered mask M_s to assess depth consistency and an auto-mask (M_a) to exclude stationary points in image pairs with no camera movement.[12] The auto mask is calculated as,

$$M_a(p) = \begin{cases} 1: \quad ||I_a(p) - I_a^{'}(p)||_1 < ||I_a(p) - I_b(p)||_1 \\ 0 \qquad\qquad\qquad\qquad\qquad\qquad\qquad : otherwise \end{cases} \tag{3.2}$$

The masks are applied to the photometric and geometric consistency loss functions as,

$$L = \frac{1}{|V|} \sum_{p \epsilon V} (M_s(p).M_a(p)L(p)) \tag{3.3}$$

Where V is the set of all valid points and L represents photometric and geometric consistency loss function.

$$L = \alpha L_p + \beta L_s + \gamma L_G \tag{3.4}$$

The loss function L combines photometric loss, smoothness loss, and geometric consistency loss, with weighting terms α, β, and γ respectively. The loss is averaged over valid points determined by the auto-mask. The The next section delves into the photometric loss and smoothness loss first, followed by an explanation of the geometric consistency loss in subsequent sections.

3.2.2 Photometric and Smoothness Loss

Classical dense correspondence algorithms often incorporate brightness constancy and spatial smoothness priors. Recent methods, such as SFMLearner[11], SC SFM-Learner[12], utilize photometric loss for unsupervised network training, comparing a warped frame with a reference frame. Through the synthesis of image $I_a^{'}$ via warping I_b based on predicted depth D_a and pose P_{ab}, an objective function L_p is formulated. L_p combines L_1 loss and Structural Similarity Index Measure (SSIM) to effectively handle illumination changes. The loss function is formulated as,

$$L_p = \frac{1}{|V|}\sum_{p \epsilon V}(\lambda||I_a(p) - I_a^{'}(p)||_1) + (1-\lambda)(\frac{1 - SSIM_{aa'}(p)}{2})) \tag{3.5}$$

where V is the set of valid points successfully projected from I_a to the image plane of I_b, p stands for a generic point in V. Here the L_1 norm is chosen due to the robustness property against outliers. The $SSIM_{aa'}$ is the structural similarity index between I_a and $I_a^{'}$. where $SSIM$ is calculated as,

$$SSIM(x,y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)} \tag{3.6}$$

where x, y stands for two 3 × 3 patches around the central pixel. C_1 and C_2 are constants. μ and σ are local statistics of the image color. As [12] the value of $C_1 = 0.0001$ and $C_2 = 0.0009$ are used and $\lambda = 0.15$. Due to the photometric loss not being informative in low texture regions, an edge aware smoothness loss is used,

$$L_s = \sum_p(\exp^{-\nabla I_a(p)} . \nabla D_a(p))^2 \tag{3.7}$$

3.2.3 Geometry Consistency Loss

In [12] the author proposed a differentiable depth inconsistency for computing pixel wise inconsistency between two depth maps. The inconsistency $D_{diff}(p)$ for each pixel $p\epsilon V$ is calculated as,

$$D_{diff}(p) = \frac{|D_b^a(p) - D_b^{'}(p)|}{D_b^a(p) + D_b^{'}(p)} \tag{3.8}$$

Where, D_b^a is the synthesized depth for I_b, which is generated by D_a and pose P_{ab} with the underlying rigid transformation. $D_b^{'}$ is an interpolation of D_b for aligning and comparing with D_b^a. The geometry consistency loss is then,

$$L_G = \frac{1}{|V|}\sum_{p\epsilon V} D_{diff}(p) \tag{3.9}$$

The self discovered mask M_s is calculated from D_{diff} as,

$$M_s = 1 - D_{diff} \tag{3.10}$$

where the M_s is in $[0, 1]$ and it attentively assign low weights for geometrically inconsistent pixels and high weights for consistent pixels.

3.2.4 Model architecture

For the pose network the model proposed in [12] is used. It takes 2 frames and outputs the 6 Degrees of Freedom (DOF) pose between the frames. It is based on Resnet18[59] with modified first layer to accept 2 frames.

For depth network three different models and 2 different input space, RGB and RMI have been used and compared to find the best performing model. The model proposed in the [11] is called the dispnet, a specially designed depth network for underwater images udepth[52] and a hybrid of these two models called dispnet mvit have been used.

Posenet

The pose network used here is the one proposed by [12]. It is a Resnet18 based encoder to extract features from images. The features are then fed to a number of convolutional regression layers to recover relative 6-DOF pose between frames. The first layer of the Resnet18 encoder is modified to be able to take 2 images as input.

Dispnet

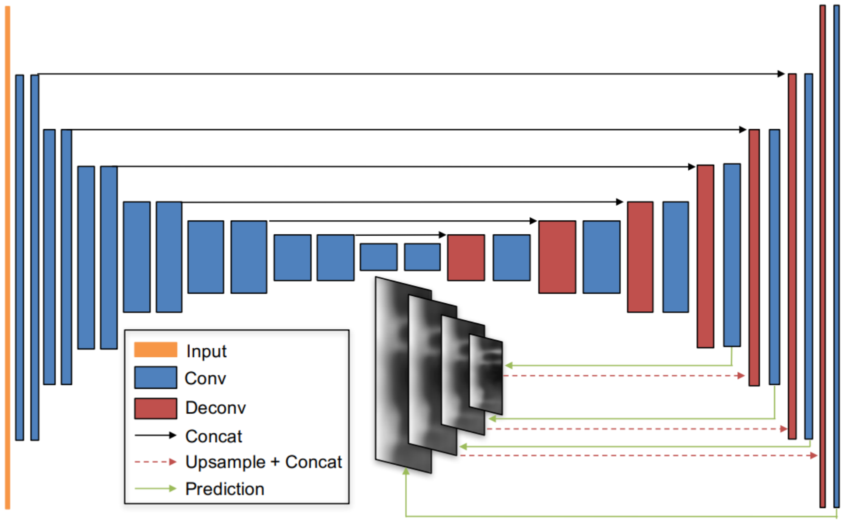


Figure 3.7: Illustration of the dispnet architecture[11]

In the figure 3.7 the network architecture of the dispnet used in [11] is illustrated. The model is an encoder-decoder based model. It takes an image as input and outputs the range map corresponding to the image. The network outputs depth in 4 different layers to be able to be used in multi scale supervision. However, in this study only the last output layer of the network has been used as this provides the best result compared to the multi scale supervision[11].

Udepth

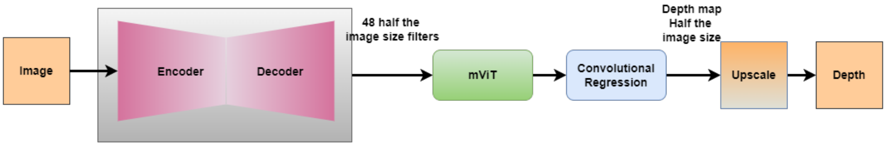


Figure 3.8: Illustration of the udepth architecture

Figure 3.8 illustrates the network architecture of the udepth model. The model is an encoder decoder network with a MobileNetV2 backbone. The output of the decoder is fed to the mVit module which is a vision transformer. The output attention maps from the mVit module are then fed to the convolutional regression layer to get the output depth map. The output of the original udepth network is half the size of the original image. To accomodate in the SFMLearner pipeline, after the network an upscale layer has been added to make the depth outputs same shape as the input images.

Dispnet mVit

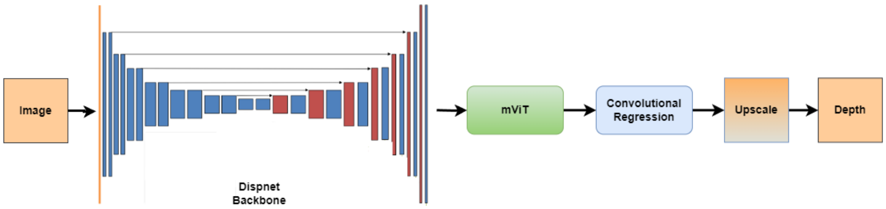


Figure 3.9: Illustration of the Dispnet mVit architecture

Figure 3.9 illustrates the network architecture of the Dispnet mVit model. The main backbone of the model is from Dispnet but the last convolutional layer is replaced by the mVit module followed by the convolutional regression module from Udepth.

3.2.5 RMI input space

The author of udepth[52] suggested that instead of using RGB underwater images directly, using RMI input space gives better result in underwater monocular depth estimation. Although the light attenuates rapidly underwater not all the wavelengths of light attenuate at the same rate. Although the red colour attenuates more aggressively, depending on the physical and chemical property of water green and blue colours attenuate at different rates. As a result [52] showed the relative difference between the intensity values of the R channel and the maximum of the {G,B} channel encodes information about the scene depth.

The RMI input space is derived from the RGB images as, the Red channel staying the same, for the M, a pixel wise maximum is calculated between B and G channels and the I is the grayscale intensities of the image.

In the SFMLearner pipeline, the RMI input space is only used in the depth network. The pose network receives the regular RGB images.

3.3 ORB-SLAM3

ORB-SLAM3 is an indirect visual SLAM system which relies on ORB features extracted from the image frames. ORB-SLAM3 fully relies on Maximum-a-Posteriori (MAP) estimation [60]. It is a multi map system. Whenever the system is lost during tracking it starts a new map which will be merged with the previous map if any part of the existing map is visited again. For underwater visual SLAM systems this property is very useful as it is quite easy for the SLAM system to lose the tracking due the quality of the underwater images. Figure 3.10 shows the overall system overview of ORB-SLAM3. There are 3 threads running simultaneously. The

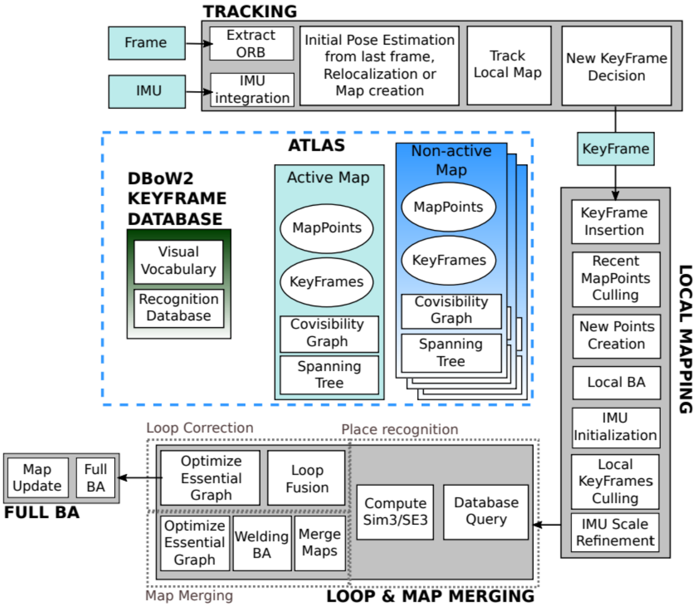


Figure 3.10: System Overview of ORB-SLAM3 [60]

tracking thread continuously processes sensor information to determine the pose of the current frame relative to an active map in real-time. This is achieved by minimizing the reprojection error of matched map features. Additionally, the system determines whether the current frame should be designated as a keyframe. In cases where tracking is lost, the tracking thread attempts to relocalize the current frame within all available maps. If successful, tracking is resumed, potentially switching the active map. If relocalization fails within a specified timeframe, the current active map is stored as non-active, and a new active map is initialized from scratch [60].

The local mapping thread plays a crucial role in updating the active map by integrating new keyframes and points, while also removing redundant elements. This process of map refinement employs visual or visual-inertial bundle adjustment within a localized window of keyframes near the current frame. [60]. Loop and map merging thread detects common regions between the active map and the whole Atlas at keyframe rate. If the common area belongs to the active map, it performs loop correction; if it belongs to a different map, both maps are seamlessly merged into a single one, that becomes the active map. After a loop correction, a full BA is launched in an independent thread to further refine the map without affecting real-time performance [60].

In this application, the RGB-D SLAM mode is used to evaluate and compare the model predicted depth and the ground truth depth obtained using SFM in Eiffel-Tower data set.

3.4 Sea-Thru

The Sea-Thru[61] method is designed to efficiently remove the effects of water in underwater imaging, facilitating analysis of large datasets. It operates by estimating backscatter in RGBD images. Additionally, it leverages the known range map to estimate the range dependent attenuation coefficient. This estimation is facilitated by an optimization framework, utilizing an illumination map generated from local space average color as input.[61]

In our application,the RGB image with the corresponding predicted depth from the network is used in the Sea-Thru pipeline. The depth is normalized as,

$$D_{normalized} = \frac{D - min(D)}{max(D) - min(D)} \tag{3.11}$$

Where, D is the predicted depth, $D_{normalized}$ is the normalized depth, $min(D)$ is the minimum of the predicted depth and $max(D)$ is the maximum of the predicted depth. The predicted normalized depth is first masked at the 40 percent of the maximum predicted depth in order to avoid calculation in the deep dark region where the depth is too large. Then this masked depth is fed to the Sea-Thru pipeline along with the histogram equalized RGB image to recover the image. Figure 3.11 shows the pipeline of Sea-Thru in this work. In this work the implementation found in [62] repository.

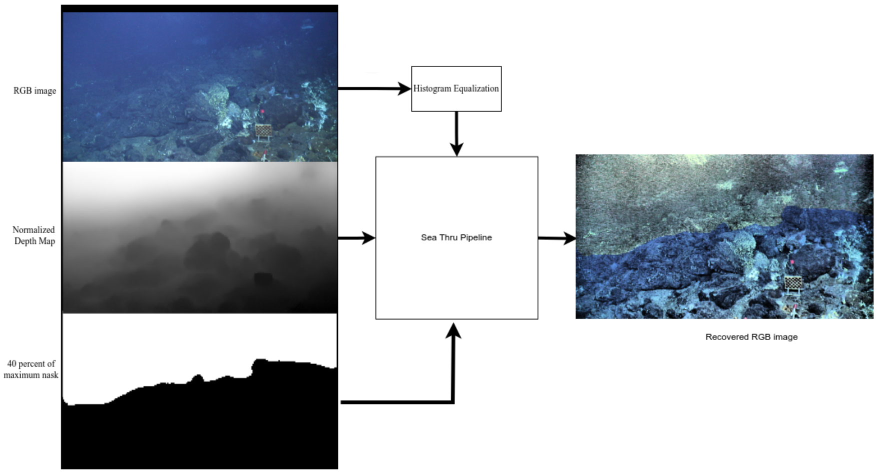


Figure 3.11: Sea-Thru Pipeline

3.5 Evaluation Metrics

3.5.1 Depth Prediction Evaluation

To evaluate the depth estimation of the network, the predicted depth has been compared with the ground truth depth of the dataset. In case of the Varos dataset the ground truth depth provided is generated from the scene geometry of the simulated environment and for the Eiffel-Tower dataset the ground truth depth has been generated from the Structure From Motion (SFM) model provided with the dataset. The predicted depth by the model has been scaled using the median value to match the scale of the ground truth, which can be written as,

$$D_{scaled} = D_{pred} * \frac{median(D_{gt})}{median(D_{pred})} \tag{3.12}$$

where D_{gt} is the ground truth depth, D_{pred} is the predicted depth by the network and D_{scaled} is the scaled depth. The following metrics have been used to evaluate the depth prediction network,

$$Abs\ Rel = \frac{1}{N}\sum_{d \epsilon D}\frac{|d^* - d|}{d^*} \tag{3.13}$$

$$Sq\ Rel = \frac{1}{N}\sum_{d \epsilon D}\frac{|d^* - d|^2}{d^*} \tag{3.14}$$

$$RMSE = \sqrt{\frac{1}{N}\sum_{d \epsilon D}|d* - d|^2} \tag{3.15}$$

$$RMSE\ log = \sqrt{\frac{1}{N}\sum_{d \epsilon D}|log\ d* - log\ d|^2} \tag{3.16}$$

$$\delta = \frac{1}{N} max(\frac{d^*}{d}, \frac{d}{d^*}) < threshold \tag{3.17}$$

Where d, d^* and N denote the scaled depth value, the corresponding ground truth depth value and the number of pixels. Here $Abs\ Rel, Sq\ Rel, RMSE, RMSElog$ are error metrics and $ai(\delta < 1.25^i, i = 1, 2, 3)$ is accuracy metrics.

3.5.2 Pose Prediction Evaluation

For evaluating the pose estimation network, the pose net was applied on the test data to predict poses and then these poses were compared to the ground truth poses from the dataset to calculate the error. In order to overcome the scale ambiguity during evaluation, we first optimize the scaling factor and align the predicted pose with the ground truth pose using least squares estimation of transformation parameters [63] using the code from [64], and then measure the Absolute Trajectory Error (ATE) as the metric. The alignment is done on every 5-frame snippets and then ATE is computed and averaged over the full sequence. The ATE is calculated as,

$$ATE(\hat{X}, X) = \frac{1}{n} \sum_{i=1}^{n} || trans(\hat{X}_i) - trans(X_i) || \tag{3.18}$$

where, $\hat{X} = \{\hat{x}_1, \hat{x}_2, ..., \hat{x}_n\}$ and $X = \{x_1, x_2, ..., x_n\}$ and trans(X) represents the translation part of X.

3.5.3 Image Enhancement evaluation

The following metrics have been used to evaluate the image enhancement result quantitatively.

Underwater Image Quality Measure (UIQM)[65]

The Underwater Image Quality Measure (UIQM) comprises of three attribute measures, namely, Underwater Image Colorfulness Measure (UICM), Underwater Image Sharpness Measure (UISM) and Underwater Image Contrast Measure (UIConM). The overall underwater image quality measure is given by,

$$UIQM = c_1 \times UICM + c_2 \times UISM + c_3 \times UIConM \tag{3.19}$$

The default value used for the parameters c_1, c_2 and c_3 in the paper are $c_1 = 0.0282$, $c_2 = 0.2953$ and $c_3 = 3.5753$. Higher UIQM value is obtained from an image with a better quality, and the authors suggest that a 10% increase in terms of the UIQM measure value leads to a visually distinguishable improvement.

Underwater Colour Image Quality Evaluation (UCIQE)[66]

The Underwater Colour Image Quality Evaluation (UCIQE) is computed in the CIElab color space. UCIQE is calculated as,

$$UCIQE = c_1 \times \sigma_c + c_2 \times con_1 + c_3 \times \mu_s \tag{3.20}$$

where, σ_c is the standard deviation of chroma, con_1 is the contrast of luminance and μ_2 is the average of saturation, and c_1, c_2 and c_3 are weighted coefficients. The values used here are $c1 = 0.4680$, $c2 = 0.2745$ and $c3 = 0.2576$.

Chapter 4

Results and Discussions

In this chapter the validity of adapting self supervised method of learning scale consistent scene depth and robot pose has been examined along with the usability of the predicted depth in applications like image enhancement and RGB-D SLAM has been studied on Eiffel-Tower, Varos and SeaThru Nerf datasets.

4.1 Experimental details

All the depth models have been separately trained with both RGB and RMI input space images. The pose network is kept the same with all the depth networks. All the models have been trained with same hyper parameters listed in table 4.1

Name	**Value**
Optimizer	Adam
Learning Rate	$1e^{-4}$
Momentum	0.9
Photometric Loss weight	1
Geometric Consistency Loss weight	0.5
Smoothness loss weight	0.1
Sequence Length	3

Table 4.1: Training hyper parameters

In all the training early stopping callback has been used. Table 4.2 shows the total parameters in the models used and the inference time of the models on image size (256 × 464). The inference time has been calculated as the average time taken over 100 inferences with batch size 1. The NVIDIA RTX 3090 GPU is used to calculate inference time. The Dispnet achieves an FPS of 26.328 on CPU and 180.031 FPS on GPU. The Udepth achieves an FPS of 11.53 on CPU and 34.583 on GPU. The proposed Dispnet mVit model achieves FPS of 19.8967 on CPU and 23.299 on GPU. The pose net achieves an FPS of 31.942 on CPU and 192.863 on GPU

Model	Total Parameters	Inference Time(s) (CPU)	FPS (CPU)	Inference Time(s) (GPU)	FPS (GPU)
Dispnet	31.59M	0.0379	26.328	0.0055	180.031
Dispnet mVit	32.57M	0.0502	19.896	0.0429	23.299
Udepth	15.59M	0.0867	11.530	0.0289	34.583
PoseNet	13.01M	0.0313	31.942	0.0051	192.863

Table 4.2: Model parameters and inference time of the models on CPU and GPU.

4.2 Depth Evaluation

In this section the performance of the depth models have been evaluated.

4.2.1 Eiffel-Tower

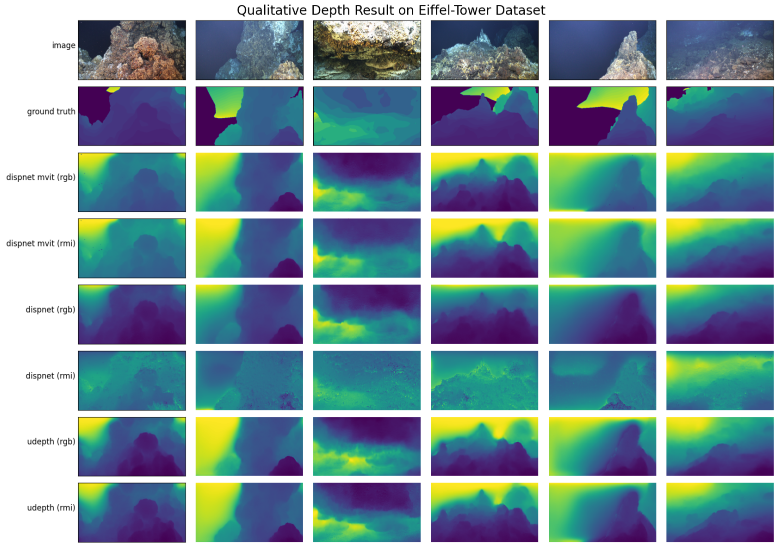


Figure 4.1: Qualitative Depth Evaluation Result on Eiffel-Tower dataset. The lighter the colour represents points further from the camera and the darker colours represent the points closer the camera. The black regions in the ground truth does not have information in the SFM model used to generate the depths.

In this section the depth models have been used to predict depth on the Eiffel Tower dataset. The predicted depth is then evaluated against the ground truth depth generated from SFM model. Figure 4.1 shows the qualitative evaluation of the predicted depths from the model on 6 sample images of the Eiffel-Tower dataset along with the ground truth depth generated from the SFM model.

Figure 4.2 shows the RMSE error changing with respect to the clipping of the maximum predicted depth. The predicted depth is masked at different depths

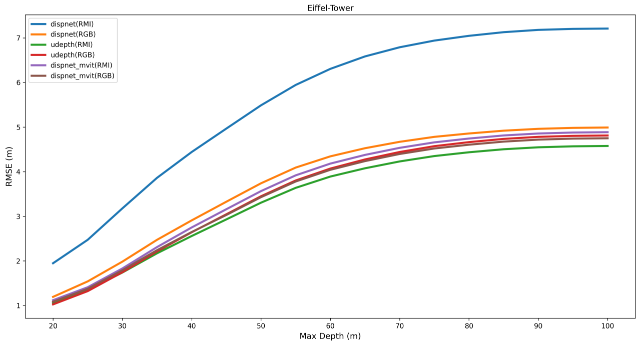


Figure 4.2: Eiffel tower depth result

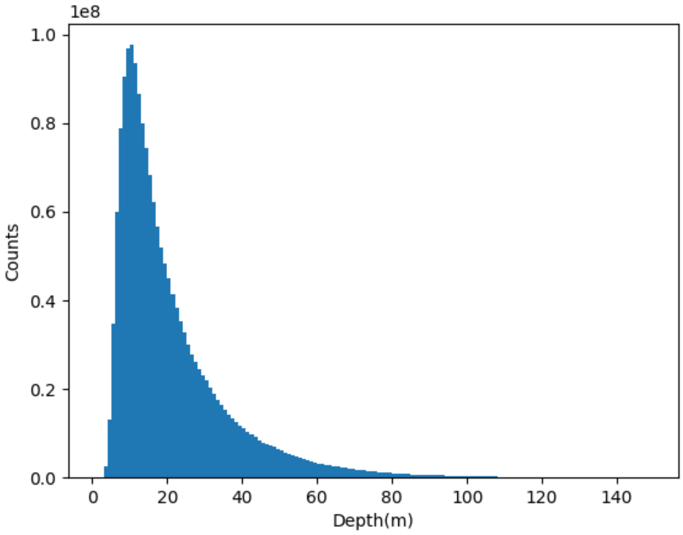


Figure 4.3: Eiffel tower depth distribution

after scaling it with the ground truth depth using the median value of the predicted and the ground truth depth and the RMSE is calculated between them. It can be seen that with the increase of the maximum depth the error increases which is expected as the light attenuates and the image quality degrades with the increase of scene depth. In figure 4.3 the distribution of the ground truth depth maps have been shown. It can be noticed that most of the depths are less than 40m. Due to the nature of light attenuation with distance in underwater environment, the images retain very limited information in the pixels where the scene depth is far. So the problem becomes ill posed in these areas. In order to evaluate and compare the models the maximum predicted depth is clipped at 40 meters and the pixels that has depth more than 40m are excluded in the calculation of the evaluation

model	$abs\ rel \downarrow$	$sq\ rel \downarrow$	$rmse \downarrow$	$rmse\ log \downarrow$	$a1 \uparrow$	$a2 \uparrow$	$a3 \uparrow$
dispnet pretrained[[11]]	0.3676	3.5174	7.4628	0.4402	0.4159	0.7031	0.8599
udepth(rmi) pretrained[52]	0.4558	6.2990	9.8530	0.6866	0.3415	0.5911	0.7403
udepth(rgb) pretrained[52]	0.3981	5.1354	8.7217	0.5774	0.3994	0.6603	0.8008
dispnet (rgb)	0.0970	0.5132	2.9087	0.1416	0.9043	0.9702	0.9876
dispnet (rmi)	0.1667	1.2071	4.4417	0.2502	0.7645	0.9066	0.9501
dispnet mvit (rgb)	0.0872	0.4076	2.6474	0.1300	0.9195	0.9763	0.9906
dispnet mvit (rmi)	0.0928	0.4286	2.7503	0.1367	0.9087	0.9745	0.9902
udepth (rgb)	<u>0.0852</u>	<u>0.3975</u>	<u>2.6424</u>	<u>0.1285</u>	<u>0.9218</u>	<u>0.9765</u>	<u>0.9907</u>
udepth (rmi)	**0.0847**	**0.3886**	**2.5583**	**0.1251**	**0.9234**	**0.9785**	**0.9920**

Table 4.3: Depth evaluation result on Eiffel-Tower (2015) Dataset. The $abs\ rel$, $sq\ rel$, $rmse$, $rmse\ log$ are error metrics while $a1$, $a2$ and $a3$ are accuracy metrics. The ↑ meaning the higher value corresponds to better results while the ↓ meaning the lower value corresponds to better results. In each metric the best performing model result is highlighted with bold font face and the second best model is highlighted with underline.

metrics. The result of the evaluation metrics are tabulated in Table 4.3. From this we can see that the udepth model with RMI input space performs best in all the metrics achieving a RMSE of 2.5583m while the second best performing model is the Udepth with RGB input space. This is an significant improvement compared to the pretrained Dispnet provided by [11] and the pretrained Udepth models provided by the [52]. The pretrained Dispnet is trained on airborne dataset in a self supervised manner while the pretrained Udepth is trained with underwater images in supervised manner.

4.2.2 Varos

model	*abs rel* ↓	*sq rel* ↓	*rmse* ↓	*rmse log* ↓	*a*1 ↑	*a*2 ↑	*a*3 ↑
dispnet pretrained[[11]]	0.2054	2.9053	10.1250	0.2623	0.6408	0.8945	0.9783
udepth(rmi) pretrained[52]	0.3773	8.6647	17.5038	0.4951	0.3897	0.6628	0.8205
udepth(rgb) pretrained[52]	0.3469	7.5390	15.8752	0.4288	0.4297	0.7142	0.8686
dispnet (rgb)	0.1736	1.8128	8.2968	0.2055	0.7179	**0.9668**	**0.9967**
dispnet (rmi)	0.2040	2.6860	9.9274	0.2456	0.6618	0.9153	0.9845
dispnet mvit (rgb)	**0.1682**	**1.7901**	**8.1343**	**0.2009**	**0.7563**	0.9602	0.9950
dispnet mvit (rmi)	0.1835	2.1831	9.0834	0.2139	0.7260	0.9529	0.9942
udepth (rgb)	0.2211	3.6371	10.9784	0.3374	0.6048	0.8038	0.9142
udepth (rmi)	0.2180	3.5606	11.7003	0.2539	0.6659	0.9112	0.9825

Table 4.4: Depth evaluation result on Varos Dataset. The *abs rel*, *sq rel*, *rmse*, *rmse log* are error metrics while *a*1, *a*2 and *a*3 are accuracy metrics. The ↑ meaning the higher value corresponds to better results while the ↓ meaning the lower value corresponds to better results. In each metric the best performing model result is highlighted with bold font face and the second best model is highlighted with underline.

Figure 4.5 shows the qualitative evaluation of the predicted depths from the models on some sample images of the Varos dataset along with the ground truth depths given in the dataset.

Figure 4.6 shows the RMSE error changing with respect to the clipping of the maximum predicted depth. From the figure it can be noticed that, in the short range the dispnet model with RMI input space works better altough the performance degrades rapidly with the increase of the maximum depth allowed in the calculation.

In figure 4.4 the depth distribution of the varos dataset can be seen. Compared to the distribution of the Eiffel-Tower dataset, most of the points are not below 40m level. So it will not justify to clip the depth at 40m in Varos dataset as well. At the same time allowing too much depth also does not give a clear picture of the result as it is too far and the points are too dark in the images for the model to predict any usable good depths. Hence, 60m is selected empirically in Varos dataset to mask the depth error calculation after scaling the depth using median

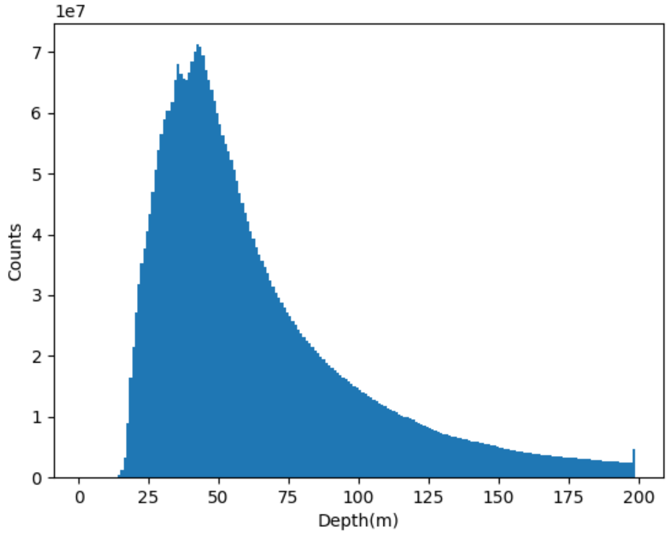


Figure 4.4: Varos depth distribution

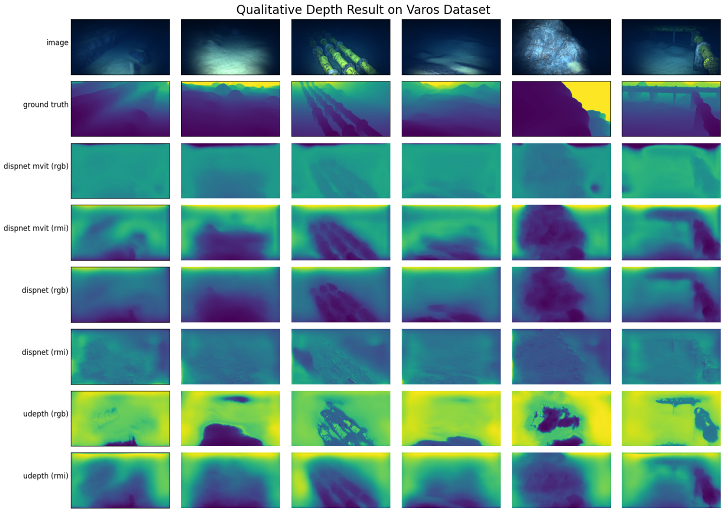


Figure 4.5: Qualitative Depth Evaluation Result on Varos dataset. The lighter the colour represents points further from the camera and the darker colours represent the points closer the camera.

values of the predicted and the groundtruth depth. Table 4.4 shows the depth evaluation result on Varos dataset where the error and accuracy calculation has been masked and clipped at 60m. It can be observed that, for all the error metrics as well as the $a1$ metrics the dispnet mvit model with RGB input space gives the best performance while for $a2$ and $a3$ the dispnet with RGB input space gives the best result. The second best performing model in terms of error metrics is the

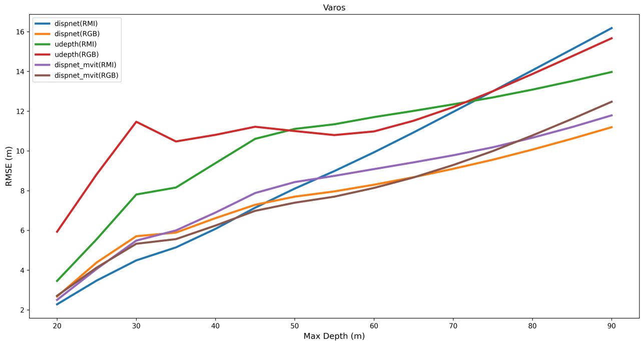


Figure 4.6: Varos depth result

dispnet with RGB input space while for $a1$ the dispnet mvit model with RMI input space and for $a2$ and $a3$ the dispnet mvit with RGB input space is the second best performing model. In Varos dataset as well, this is an significant improvement over the pretrained Dispnet and Udepth model provided by [11] and [52].

4.3 Pose Evaluation

In this section the pose prediction result of the pose net trained with different depth models are evaluated. The predicted pose from the model are aligned and scaled with the ground truth using 5 frame snippets as described in section 3.5.2.

Depth Model Name	Total ATE (m) ↓	Mean ATE (m) ↓
udepth (rmi)	1614.966	0.328
udepth (rgb)	1598.573	0.325
dispnet mvit (rmi)	1594.794	0.324
dispnet mvit (rgb)	1588.771	0.323
dispnet (rmi)	**1532.216**	**0.311**
dispnet (rgb)	1624.323	0.330

Table 4.5: Pose evaluation result on Eiffel Tower (2015) dataset. The ↓ meaning the lower value corresponds to better results. The best result is highlighted with bold font face and the second best result is highlighted with underline

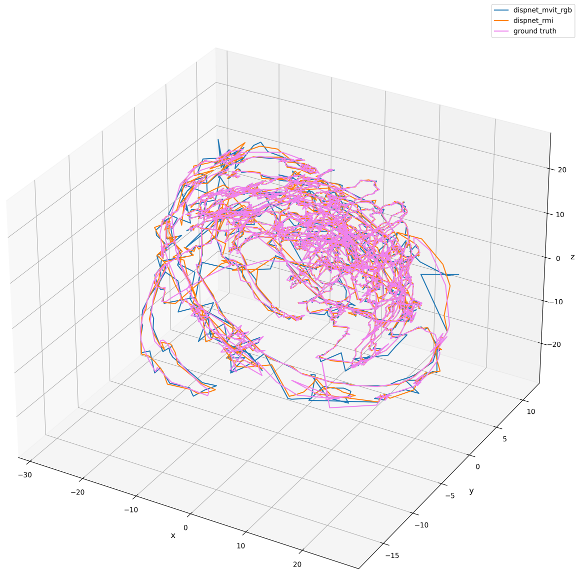
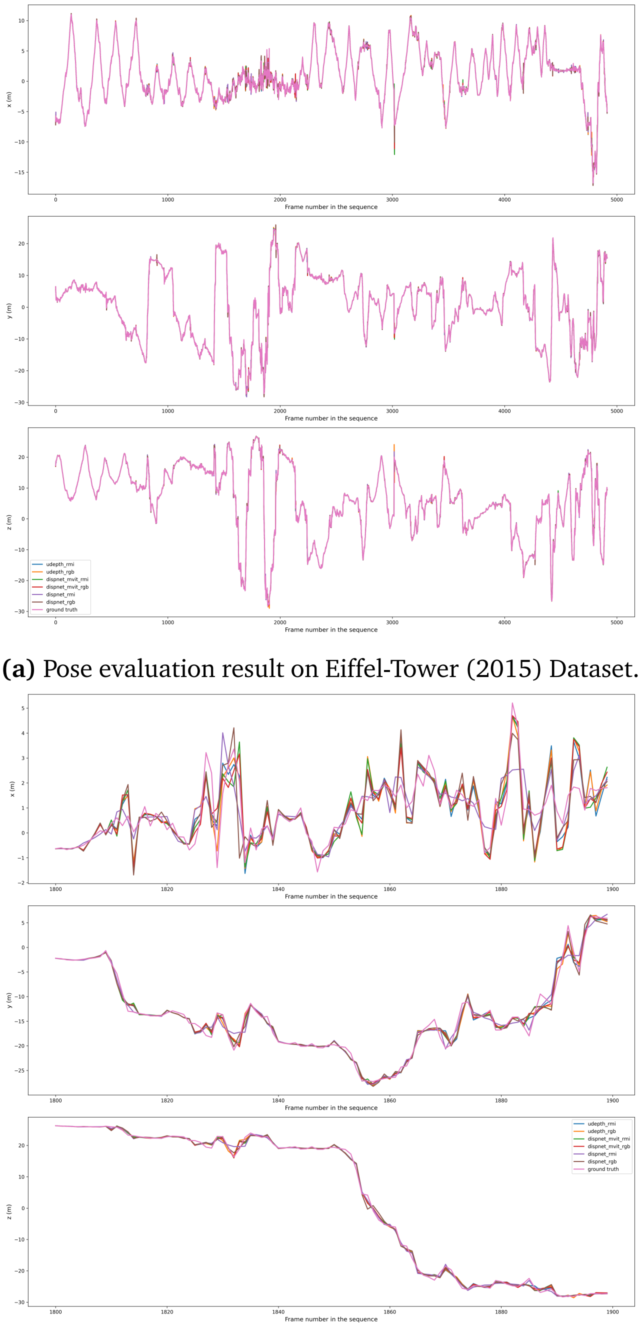


Figure 4.7: 3D plot of the best two performing pose network on Eiffel-Tower (2015) Dataset.

Table 4.5 shows and compares the total ATE and mean ATE achieved by the



(a) Pose evaluation result on Eiffel-Tower (2015) Dataset.

(b) Pose evaluation result on Eiffel-Tower (2015) Dataset focused on frame 1800-1900.

Figure 4.8: Pose evaluation results on the Eiffel-Tower (2015) Dataset. (a) Full sequence. (b) Focused on frame 1800-1900.

pose net trained with different depth models on the eiffel tower dataset. The pose net trained with dispnet along with RMI input space performed the best with total ATE of 1532.216m and mean ATE of 0.311m. The second best performing pose net is the one trained with the dispnet mvit depth model along with RGB input space. It achieved a total ATE of 1588.717m and mean ATE of 0.323m.

Figure 4.8 shows the pose net evaluation result on the Eiffel-Tower dataset in the 2015 sequence. The x, y and z components of the predicted trajectories has been plotted in the figure with the corresponding ground truths. Figure 4.8a shows the result on the whole sequence. For getting a more detailed view on the the differences among different models the Figure 4.8b focuses on the frames from 1800-1900 in the sequence. It can be seen that all the models have been able to track the ground truth trajectory in a fairly good accuracy as suggested from table 4.5. Figure 4.7 shows the 3D trajectory of the predicted pose from the two best performing models and the ground truth.

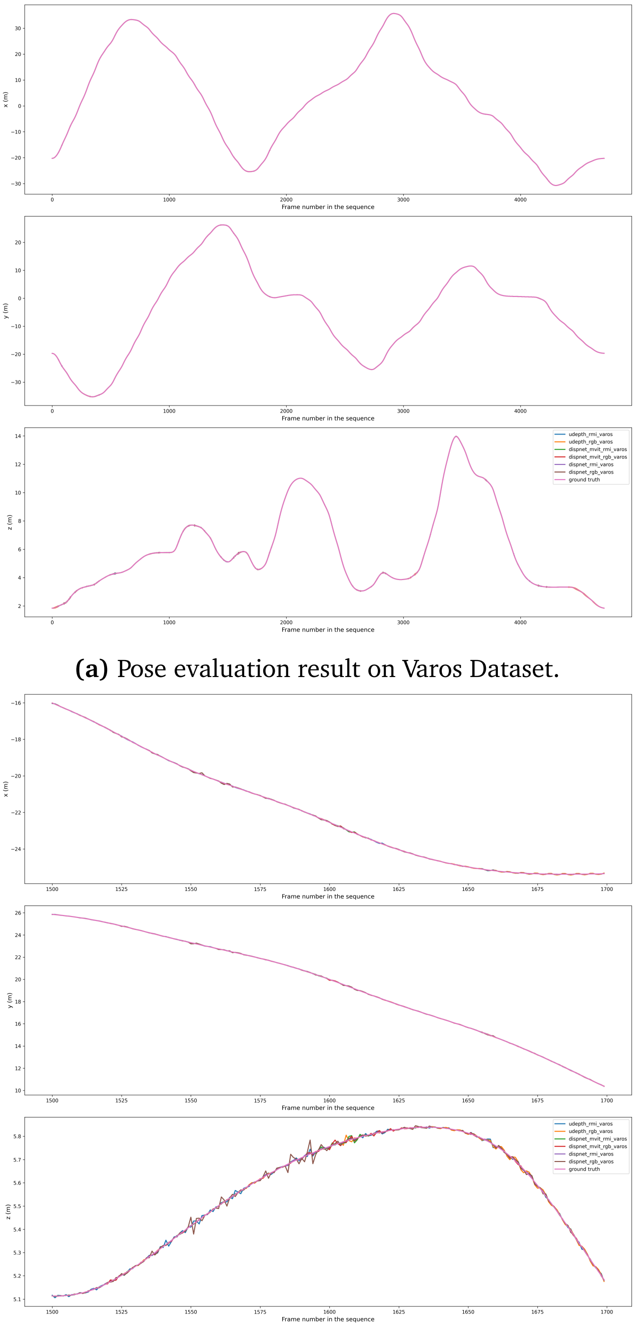
Depth Model Name	Total ATE (m) ↓	Mean ATE (m) ↓
udepth (rmi)	30.56375	0.0064
udepth (rgb)	45.51164	0.0096
dispnet mvit (rmi)	**20.21493**	**0.0042**
dispnet mvit (rgb)	<u>25.84997</u>	<u>0.0054</u>
dispnet (rmi)	28.69131	0.0060
dispnet (rgb)	50.04694	0.0106

Table 4.6: Pose evaluation result on Varos dataset. The ↓ meaning the lower value corresponds to better results. The best result is highlighted with bold font face and the second best result is highlighted with underline

Table 4.6 shows and compares the total ATE and mean ATE achieved by the pose net trained with different depth models on the Varos dataset. The pose net trained with dispnet mvit along with RMI input space performed the best with total ATE of 20.21493m and mean ATE of 0.0042m. The second best performing pose net is the one trained with the dispnet mvit depth model along with RGB input space. It achieved a total ATE of 25.84997m and mean ATE of 0.0054m.

Figure 4.9 shows the pose net evaluation result on the Varos dataset. The x, y and z components of the predicted trajectories has been plotted in the figure with the corresponding ground truths. Figure 4.9a shows the result on the whole sequence. For getting a more detailed view on the the differences among different models the Figure 4.9b focuses on the frames from 1500-1700 in the sequence. It can be seen that all the models has been able to track the ground truth trajectory in a fairly good accuracy as suggested from table 4.6. Figure 4.10 shows the 3D trajectory of the predicted pose from the two best performing models from table 4.6 and the ground truth.

From the pose net result both on Eiffel-Tower dataset and Varos dataset it can be noticed that the model performed much better on the Varos dataset with the



(a) Pose evaluation result on Varos Dataset.

(b) Pose evaluation result on Varos Dataset focused on frame 1500-1700.

Figure 4.9: Pose evaluation results on the Varos Dataset. (a) Full sequence. (b) Focused on frame 1500-1700.

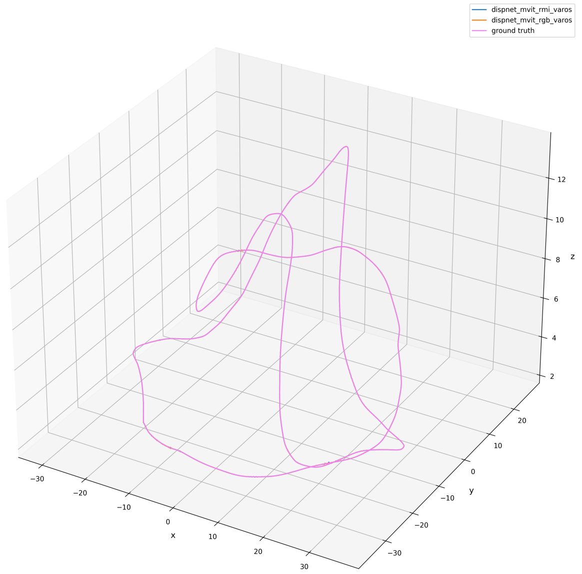


Figure 4.10: 3D plot of the best two performing pose network on Varos Dataset.

dispnet mvit (rmi) model achieving mean ATE of 0.0042m in Varos dataset compared to the mean ATE of 0.311m in the dispnet (rmi) model in the Eiffel-Tower dataset even though the model was not trained on Varos.

If we look into the trajectory of the both dataset, it can be noticed that the Eiffel-tower trajectory has more abrupt changes in the robots position while the Varos dataset being a simulated dataset has a smoother trajectory. This may have contributed to the results of the pose evaluation being better in the Varos dataset. The pose net also needs to be validated on a real underwater image dataset to get a better understanding on the models generalizability.

4.4 ORB-SLAM3 Result

In this section the predicted depth from the best performing depth model Udepth with RMI input space and the ground truth depth from the SFM model have been used and compared as a depth sensor in the RGB-D SLAM pipeline for the Eiffel-Tower Dataset. For comparing and calculating the result only the first 1000 frames from the year 2015 have been used. The total length of the ground truth trajectory of the first 1000 frame is 455.476m. To minimize the influence of coincidental outcomes, ORB-SLAM3 RGB-D SLAM was run three times with each of the depth (SFM,Udepth(RMI)), ensuring a comprehensive assessment and enhancing the robustness of the findings.

Depth	RMSE ATE (m) ↓	Median Error (m) ↓	Max Error (m) ↓
SFM	9.58	8.27	18.86
	9.82	8.54	19.07
	9.78	7.96	19.80
Avg	**9.73**	**8.26**	**19.24**
Udepth (RMI)	9.25	7.64	21.36
	10.07	9.36	19.52
	10.05	8.74	20.82
Avg	9.79	8.58	20.57

Table 4.7: RGB-D SLAM result on Eiffel-Tower Dataset. The ↓ represents that lower value corresponds to better result. The best performing Depth has been highlighted with bold font face. The best performing run on each depth type has been highlighted with underline.

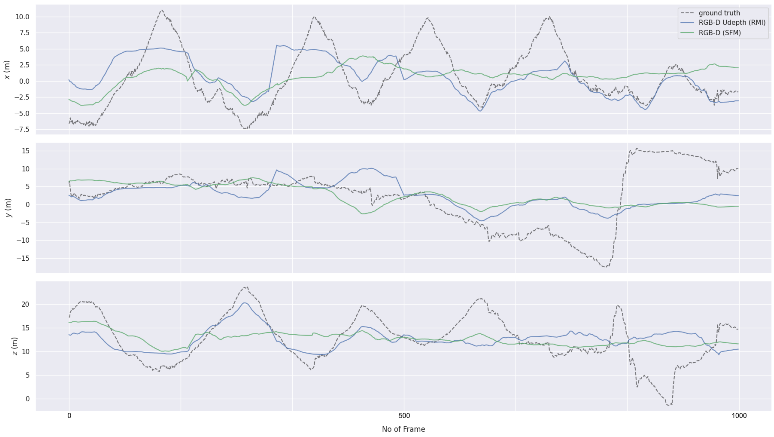


Figure 4.11: RGB-D SLAM result on the Eiffel-Tower Dataset.

Figure 4.11 shows the trajectory of the best of the 3 runs in terms of RMSE error on both the depths. It plots the x, y and z component of the predicted trajectory from the RGB-D SLAM with the ground truth trajectory. Table 4.7 shows the absolute trajectory error of the RGB-D SLAM. The errors are calculated using the predicted trajectory and the ground truth trajectory. The predicted trajectories are scaled and aligned with the ground truth trajectory before calculating the errors. From the table it can be noticed that the ground truth depth from the SFM model gives overall better performance when averaged over three different runs with an average RMSE ATE of 9.73 m compared to the 9.79m of the Udepth (RMI) model predicted depths. The model predicted depth achieved an average of 2.14% RMSE compared to the total length of the ground truth trajectory.

4.5 Image Enhancement Result

In this section, the image enhancement performance using the predicted depth from the depth network using the SeaThru pipeline has been evaluated. All the predicted depths have been normalized using equation 3.11 and masked at the 40% of the maximum predicted depth before being used in the SeaThru pipeline.

4.5.1 Eiffel-Tower

Figure 4.12 shows the qualitative result of the image enhancement procedure on Eiffel-Tower (2015) dataset. The first row shows the input images from the dataset, while the subsequent rows are the predicted depths, the 40% of the maximum depth mask and the enhanced image. The different columns represent the result from the different models being used.

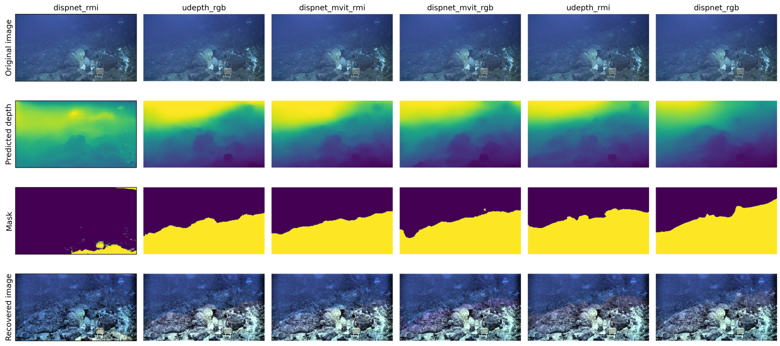


Figure 4.12: Qualitative image enhancement result on Eiffel (2015) Dataset. The light colors in the depth maps represents the areas that are closer and the dark colors represent the areas that are far from the camera. In the mask the yellow colored points are valid points based on the mask.

Table 4.8 shows the quantitative results of the image enhancement method on the Eiffel-Tower dataset. The mean UIQM and UCIQE on the original dataset before image enhancement are 1.31 and 29.89 respectively. From table 4.8 it can be seen that even though all the models led to an increase of the image quality based on UIQM and UCIQE on the Eiffel Tower dataset, the Dispnet with RMI input space performed best in term of UIQM metric. The model achieved an UIQM of 1.66 which is an 26.68% increase from the raw dataset images. In terms of UCIQE the dispnet with RGB input space performed best with UCIQE of 31.44, an increase of 4.94% from the raw dataset. We can notice that the predicted depth resulted in better image enhancement than the SFM model predicted ground truth depth. This may have been a result of not using the full range of depth and using only upto 40% of the maximum depth.

model	uiqm ↑	uiqm increase (%) ↑	uciqe ↑	uciqe increase (%) ↑
Original image	1.32	0	28.89	0
Ground Truth Depth	1.48	12.24	30.98	3.53
dispnet mvit (rgb)	1.509	15.79	31.31	4.62
dispnet mvit (rmi)	1.508	14.02	31.19	4.20
dispnet (rgb)	1.63	23.89	**31.44**	**4.94**
dispnet (rmi)	**1.66**	**26.68**	31.41	4.84
udepth (rgb)	1.53	16.27	31.34	4.63
udepth (rmi)	1.55	17.66	31.35	4.68

Table 4.8: Quantitative image enhancement result on Eiffel-Tower dataset. The ↑ means the higher the value of the metric is the better the model's performance. The best performing models are highlighted with bold face fonts.

4.5.2 Varos

Figure 4.13 shows the qualitative result of the image enhancement procedure on the Varos dataset. The first row shows the input images from the dataset, while the subsequent rows are the predicted depths, the 40% of the maximum depth mask and the enhanced image. The different columns represent the result from the different models being used. From the enhanced images in figure 4.13 we can notice that the method recovers parts of the images where it is too dark to identify anything even with human eye. One reason of this can be, the Varos dataset being a simulated dataset, retains some information even after adding the effects of water while exporting the images. As a result the SeaThru can exploit this information in combination with the depth map to bring out details in the shadows.

Table 4.9 shows the quantitative results on the image enhancement on the Varos dataset. The mean UIQM and UCIQE on the original dataset before image enhancement are 0.12 and 29.52 respectively. Table 4.9 shows significant better image enhancement on the Varos dataset compared to the Eiffel-Tower dataset for all the models even though the model was not trained on the Varos dataset. Table 4.9 shows that the Dispnet with RGB performed the best in terms of UIQM with an value of 0.50 which is 305.03% increase from the raw dataset. In terms of UCIQE the Dispnet with RMI input space performed best among the models with an UCIQE of 31.75 and increase of 6.94% from raw images. It can be noticed that the predicted depth resulted in better image enhancement than the ground truth depth. This may have been a result of not using the full range of depth and using

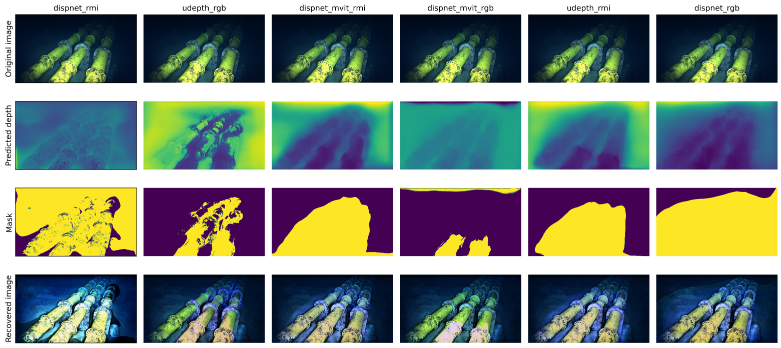


Figure 4.13: Qualitative image enhancement result on Varos Dataset. The light colors in the depth maps represents the areas that are closer and the dark colors represent the areas that are far from the camera. In the mask the yellow colored points are valid points based on the mask.

only upto 40% of the maximum depth.

4.5.3 SeaThru-Nerf

As the image enhancement result on the Varos is much better than the image enhancement on the Eiffel-Tower Dataset, it is important to evaluate the model with a real world dataset for getting a better understanding of the model's ability to generalize. For this purpose the method is evaluated on the SeaThru-Nerf dataset with 4 different scenes namely Curasao, Panama, IUI3 - RedSea and JapaneseGradens - RedSea.

Curasao

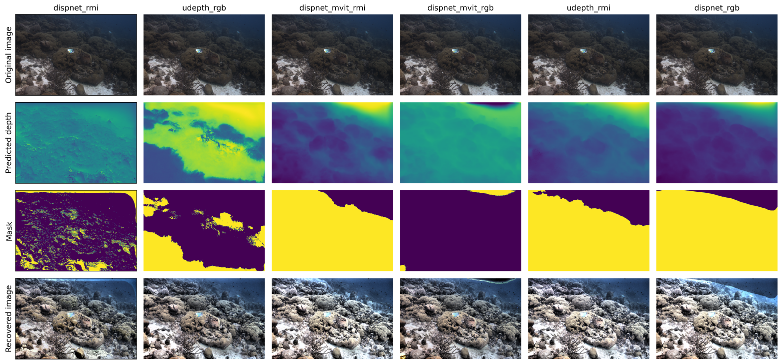


Figure 4.14: Qualitative image enhancement result on SeaThru-Nerf (Curasao) Dataset. The light colors in the depth maps represents the areas that are closer and the dark colors represent the areas that are far from the camera. In the mask the yellow colored points are valid points based on the mask.

model	uiqm ↑	uiqm increase (%) ↑	uciqe ↑	uciqe increase (%) ↑
Original Image	0.12	0	29.52	0
Ground Truth Depth	0.4234	252.83	30.5205	3.3892
dispnet mvit (rgb)	0.17	38.99	29.95	1.43
dispnet mvit (rmi)	0.35	183.83	30.81	4.21
dispnet (rgb)	**0.50**	**305.03**	30.39	2.87
dispnet (rmi)	0.27	113.49	**31.75**	**6.94**
udepth (rgb)	0.18	46.75	30.32	2.61
udepth (rmi)	0.32	156.64	30.71	3.88

Table 4.9: Image enhancement result on Varos. The ↑ means the higher the value of the metric is the better the model's performance. The best performing models are highlighted with bold face fonts

Figure 4.14 shows the qualitative result of the image enhancement procedure on the Curasao scene of the dataset. The first row shows the input images from the dataset, while the subsequent rows are the predicted depths, the 40% of the maximum depth mask and the enhanced image. The different columns represent the result from the different models being used.

The mean UIQM and UCIQE on the original dataset before image enhancement are 1.78 and 28.99 respectively. Table 4.10 shows the Dispnet with RGB input space performs best in terms of UIQM. The model achieves an UIQM of 2.06 with 16.09% increase. While the model Dispnet with RMI input space performs best in terms of UCIQE. It achieves an UCIQE of 32.62 with increase of 7.44%.

Panama

Figure 4.15 shows qualitative result of the image enhancement procedure on the Panama scene of the dataset.

The mean UIQM and UCIQE on the original dataset before image enhancement are 1.64 and 28.63 respectively. Table 4.11 shows that the Dispnet model with RGB input space performed best on the scene based on both UIQM and UCIQE metric. The model achieved UIQM of 2.01 with a 22.27% increase and UCIQE of 30.39 with a 5.80% increase from the raw images.

model	uiqm ↑	uiqm increase (%) ↑	uciqe ↑	uciqe increase (%) ↑
Original Image	1.78	0	28.99	0
dispnet mvit (rgb)	1.80	1.25	29.59	2.02
dispnet mvit (rmi)	2.02	13.61	30.72	5.62
dispnet (rgb)	**2.06**	**16.09**	31.30	7.37
dispnet (rmi)	2.04	14.99	**31.32**	**7.44**
udepth (rgb)	1.94	9.07	30.67	5.45
udepth (rmi)	1.98	11.25	30.59	5.22

Table 4.10: Image enhancement result on Curasao. The ↑ means the higher the value of the metric is the better the model's performance. The best performing models are highlighted with bold face fonts

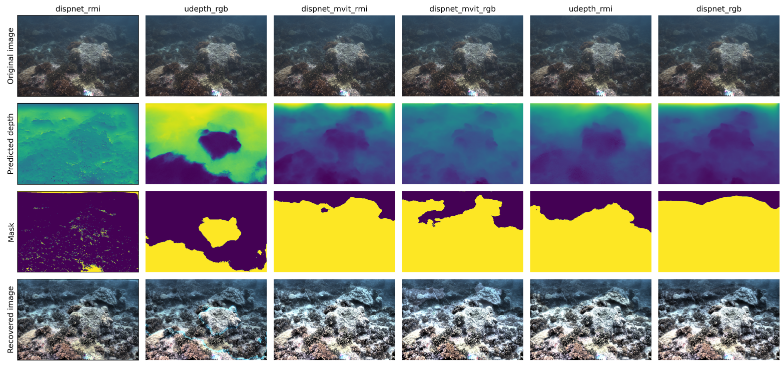


Figure 4.15: Qualitative image enhancement result on SeaThru-Nerf (Panama) Dataset. The light colors in the depth maps represents the areas that are closer and the dark colors represent the areas that are far from the camera. In the mask the yellow colored points are valid points based on the mask.

IUI3-RedSea

Figure 4.16 shows qualitative result of the image enhancement procedure on the IUI3-RedSea scene of the dataset.

The mean UIQM and UCIQE on the original dataset before image enhancement are 0.89 and 30.96 respectively. Table 4.12 shows the Dispnet with RGB input space performs best in terms of both UIQM and UCIQE achieving 1.21 and 32.62 respectively. This is a 36.32% and 5.07% percent increase from the raw dataset.

model	uiqm ↑	uiqm increase (%) ↑	uciqe ↑	uciqe increase (%) ↑
Original Image	1.64	0	28.63	0
dispnet mvit (rgb)	1.82	10.66	29.87	4.15
dispnet mvit (rmi)	1.97	19.97	30.30	5.50
dispnet (rgb)	**2.01**	**22.27**	**30.39**	**5.80**
dispnet (rmi)	1.86	13.10	30.28	5.45
udepth (rgb)	1.73	5.02	29.76	3.78
udepth (rmi)	1.86	13.32	30.27	5.42

Table 4.11: Image enhancement result on Panama. The ↑ means the higher the value of the metric is the better the model's performance. The best performing models are highlighted with bold face fonts.

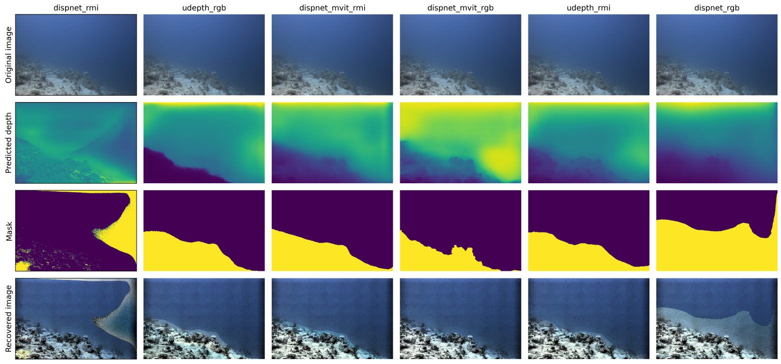


Figure 4.16: Qualitative image enhancement result on SeaThru-Nerf (IUI3-RedSea) Dataset. The light colors in the depth maps represents the areas that are closer and the dark colors represent the areas that are far from the camera. In the mask the yellow colored points are valid points based on the mask.

JapaneseGradens-RedSea

Figure 4.17 shows the qualitative result of the image enhancement on the JapaneseGradens - RedSea scene of the dataset.

The mean UIQM and UCIQE on the original dataset before image enhancement are 1.29 and 29.94 respectively. Table 4.13 shows that the Dispnet with RGB input space works best in terms of UIQM with a value of 1.21, an increase of 36.32%. However, the dispnet mvit model with RMI input space performs best in terms of UCIQE metric, achieving 31.22 and increase of 4.10%.

From the results of the image enhancement it can be concluded that the im-

model	uiqm ↑	uiqm increase (%) ↑	uciqe ↑	uciqe increase (%) ↑
Original Image	0.88	0	30.96	0
dispnet mvit (rgb)	0.96	8.55	32.18	3.79
dispnet mvit (rmi)	1.08	21.98	32.52	4.79
dispnet (rgb)	**1.21**	**36.32**	**32.62**	**5.07**
dispnet (rmi)	1.19	33.80	31.88	2.88
udepth (rgb)	1.00	13.19	32.32	4.22
udepth (rmi)	1.06	18.97	32.52	4.81

Table 4.12: Image enhancement result on IUI3-RedSea. The ↑ means the higher the value of the metric is the better the model's performance. The best performing models are highlighted with bold face fonts.

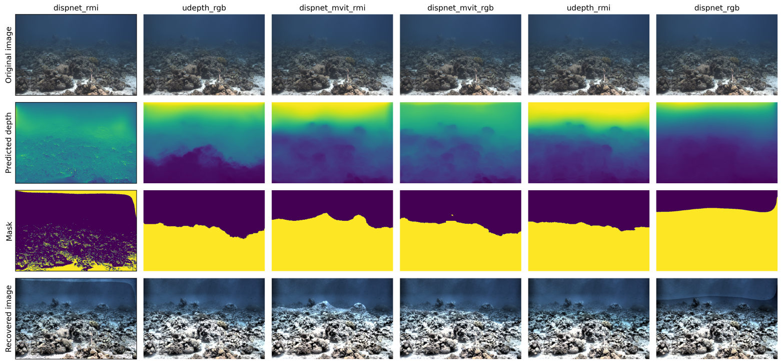


Figure 4.17: Qualitative image enhancement result on SeaThru-Nerf (JapaneseGradens - RedSea) Dataset. The light colors in the depth maps represents the areas that are closer and the dark colors represent the areas that are far from the camera. In the mask the yellow colored points are valid points based on the mask.

age enhancement of the SeaThru-Nerf dataset produced similar results to that of Eiffel-Tower dataset in terms of UIQM and UCIQE values. But in case of the Varos dataset, it achieved far more superior result due to it's being a simulated dataset and containing enough information in the shadows for the SeaThru to exploit.

model	uiqm ↑	uiqm increase (%) ↑	uciqe ↑	uciqe increase (%) ↑
Original Image	1.29	0	29.94	0
dispnet mvit (rgb)	1.42	10.01	30.85	2.92
dispnet mvit (rmi)	1.65	27.91	**31.22**	**4.10**
dispnet (rgb)	**1.67**	**29.43**	31.21	4.06
dispnet (rmi)	1.44	11.32	30.97	3.32
udepth (rgb)	1.48	14.53	30.85	2.94
udepth (rmi)	1.56	20.41	31.14	3.83

Table 4.13: Image enhancement result on JapaneseGradens-RedSea. The ↑ means the higher the value of the metric is the better the model's performance. The best performing models are highlighted with bold face fonts.

Chapter 5

Conclusion

In this thesis, the feasibility of using a self supervised learning based method for predicting scene depth and robot pose from underwater videos has been studied. In this regard, different depth prediction models along with two different input space for the images RMI and RGB have been used and compared. Additionally, the predicted depths has been used to enhance the underwater image quality.

From the results it can be concluded that, the self supervised learning based ego motion learning along with depth learning can be adapted for the underwater imagery, which can potentially solve the problems faced in underwater navigation. The predicted depths can also be used to enhance underwater images using methods such as SeaThru[61] or Sucre[67]. The models are learned with unlabeled videos which solves the problem of scarcity of dataset labeled with pose and depths.

The models are trained with the 3 scenes of Eiffel-Tower dataset and have been evaluated with a different scene from the Eiffel-Tower dataset along with a simulated dataset Varos and a real world dataset SeaThru-Nerf to find out the generalization capability of the models. While there is room for improvement, it is seen that the models could generalize the performance on unseen different datasets satisfactorily. It is seen that in the simulated dataset Varos the image enhancement using the predicted depths worked much better than the Eiffel-Tower dataset, which may have been the result of it being a simulated dataset and it retains some pixel level information even in the shadows of the images. However, in the SeaThru-Nerf dataset we can see that the image enhancement performance was similar to that on the Eiffel-Tower dataset.

SLAM in underwater environments is a challenging task (due to the drastic attenuation of electro magnetic signals underwater and the physical properties of the water) which has not yet been solved. Although the method used in this thesis performs quite well in predicting the scene depth and robot pose, the problem of underwater scene reconstruction is not fully solved yet. The current method does not predict scale consistent pose of the robot because during training small snippets consisting of 3 sequential images from the image sequence has been used in every batch. This results in the model not being able to predict consistent pose for frames that are far away from each other and creates ambiguity. As a result the poses predicted by the network can not be directly used in the SLAM problem. While the predicted depth is scale consistent throughout the scene it is not scaled to the ground truth depths. To solve these problems, one method can be to use the pre-integrated Inertial Measurement Unit (IMU) data in the model for the model to learn the pose and depth that can be directly used in the SLAM problem. Another method of getting consistent pose can be to fine tune the pose network with longer snippets of training data after the initial training has been done.

Bibliography

[1] X. Wang, X. Fan, P. Shi, J. Ni and Z. Zhou, 'An overview of key slam technologies for underwater scenes,' *Remote Sensing*, vol. 15, no. 10, 2023, ISSN: 2072-4292. DOI: 10.3390/rs15102496. [Online]. Available: https://www.mdpi.com/2072-4292/15/10/2496.

[2] C. Cadena, L. Carlone, H. Carrillo, Y. Latif, D. Scaramuzza, J. Neira, I. Reid and J. J. Leonard, 'Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age,' *IEEE Transactions on Robotics*, vol. 32, no. 6, pp. 1309–1332, Dec. 2016, ISSN: 1941-0468. DOI: 10.1109/tro.2016.2624754. [Online]. Available: http://dx.doi.org/10.1109/TRO.2016.2624754.

[3] W. Zhao, T. He, A. Y. M. Sani and T. Yao, 'Review of slam techniques for autonomous underwater vehicles,' in *Proceedings of the 2019 International Conference on Robotics, Intelligent Control and Artificial Intelligence*, ser. RICAI '19, Shanghai, China: Association for Computing Machinery, 2019, pp. 384–389, ISBN: 9781450372985. DOI: 10.1145/3366194.3366262. [Online]. Available: https://doi.org/10.1145/3366194.3366262.

[4] L. Paull, S. Saeedi, M. Seto and H. Li, 'Auv navigation and localization: A review,' *IEEE Journal of oceanic engineering*, vol. 39, no. 1, pp. 131–149, 2013.

[5] A. Burguera, F. Bonin-Font, E. G. Font and A. M. Torres, 'Combining deep learning and robust estimation for outlier-resilient underwater visual graph slam,' *Journal of Marine Science and Engineering*, vol. 10, no. 4, p. 511, 2022.

[6] 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, 2022, ISSN: 1574-0137. DOI: https://doi.org/10.1016/j.cosrev.2022.100510. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1574013722000442.

[7] L. Silveira, F. Guth, P. Drews-Jr, P. Ballester, M. Machado, F. Codevilla, N. Duarte-Filho and S. Botelho, 'An open-source bio-inspired solution to underwater slam,' *IFAC-PapersOnLine*, vol. 48, no. 2, pp. 212–217, 2015, 4th IFAC Workshop on Navigation, Guidance and Controlof Underwater Vehicles NGCUV 2015, ISSN: 2405-8963. DOI: https://doi.org/10.1016/j.

ifacol.2015.06.035. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2405896315002748.

[8] M. J. Islam, Y. Xia and J. Sattar, 'Fast underwater image enhancement for improved visual perception,' *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 3227–3234, 2020. DOI: 10.1109/LRA.2020.2974710.

[9] M. A. Azad, A. Mohammed, M. Waszak, B. Elvesæter and M. Ludvigsen, 'Multi-label video classification for underwater ship inspection,' in *OCEANS 2023-Limerick*, IEEE, 2023, pp. 1–10.

[10] M. Ferrera, V. Creuze, J. Moras and P. Trouvé-Peloux, 'Aqualoc: An underwater dataset for visual–inertial–pressure localization,' *The International Journal of Robotics Research*, vol. 38, no. 14, pp. 1549–1559, 2019. DOI: 10.1177/0278364919883346. [Online]. Available: https://doi.org/10.1177/0278364919883346.

[11] T. Zhou, M. Brown, N. Snavely and D. G. Lowe, 'Unsupervised learning of depth and ego-motion from video,' in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jul. 2017.

[12] J. Bian, Z. Li, N. Wang, H. Zhan, C. Shen, M.-M. Cheng and I. Reid, 'Unsupervised scale-consistent depth and ego-motion learning from monocular video,' in *Advances in Neural Information Processing Systems*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox and R. Garnett, Eds., vol. 32, Curran Associates, Inc., 2019.

[13] Y. Chen, C. Schmid and C. Sminchisescu, 'Self supervised learning with geometric constraints in monocular video: Connecting flow, depth, and camera,' in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, Oct. 2019.

[14] O. Mac Aodha, M. Firman, G. J. Brostow *et al.*, 'Digging into self-supervised monocular depth estimation,' in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)(2019)*, 2019, pp. 3827–3837.

[15] A. Gordon, H. Li, R. Jonschkowski and A. Angelova, 'Depth from videos in the wild: Unsupervised monocular depth learning from unknown cameras,' in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, Oct. 2019.

[16] S. A. K. Tareen and Z. Saleem, 'A comparative analysis of sift, surf, kaze, akaze, orb, and brisk,' in *2018 International conference on computing, mathematics and engineering technologies (iCoMET)*, IEEE, 2018, pp. 1–10.

[17] A. Iqbal and N. R. Gans, 'Data association and localization of classified objects in visual slam,' *Journal of Intelligent & Robotic Systems*, vol. 100, no. 1, pp. 113–130, 2020.

[18] J. Aulinas, M. Carreras, X. Llado, J. Salvi, R. Garcia, R. Prados and Y. R. Petillot, 'Feature extraction for underwater visual slam,' in *OCEANS 2011 IEEE-Spain*, IEEE, 2011, pp. 1–7.

[19] F. Ferreira, G. Veruggio, M. Caccia and G. Bruzzone, 'Real-time optical slam-based mosaicking for unmanned underwater vehicles,' *Intelligent Service Robotics*, vol. 5, no. 1, pp. 55–71, 2012.

[20] T. Maki, H. Kondo, T. Ura and T. Sakamaki, 'Photo mosaicing of tagiri shallow vent area by the auv" tri-dog 1" using a slam based navigation scheme,' in *OCEANS 2006*, IEEE, 2006, pp. 1–6.

[21] I. Mahon, S. B. Williams, O. Pizarro and M. Johnson-Roberson, 'Efficient view-based slam using visual loop closures,' *IEEE Transactions on Robotics*, vol. 24, no. 5, pp. 1002–1014, 2008.

[22] G. Dubbelman and B. Browning, 'Cop-slam: Closed-form online pose-chain optimization for visual slam,' *IEEE Transactions on Robotics*, vol. 31, no. 5, pp. 1194–1213, 2015.

[23] R. Mur-Artal, J. M. M. Montiel and J. D. Tardos, 'Orb-slam: A versatile and accurate monocular slam system,' *IEEE transactions on robotics*, vol. 31, no. 5, pp. 1147–1163, 2015.

[24] S. Rahman, A. Q. Li and I. Rekleitis, 'Svin2: An underwater slam system using sonar, visual, inertial, and depth sensor,' in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2019, pp. 1861–1868.

[25] S. Rahman, A. Q. Li and I. Rekleitis, 'Sonar visual inertial slam of underwater structures,' in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2018, pp. 5190–5196.

[26] R. Mur-Artal and J. D. Tardós, 'Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras,' *IEEE transactions on robotics*, vol. 33, no. 5, pp. 1255–1262, 2017.

[27] S. Xu, T. Luczynski, J. S. Willners, Z. Hong, K. Zhang, Y. R. Petillot and S. Wang, 'Underwater visual acoustic slam with extrinsic calibration,' in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2021, pp. 7647–7652.

[28] F. Hidalgo, C. Kahlefendt and T. Bräunl, 'Monocular orb-slam application in underwater scenarios,' in *2018 OCEANS - MTS/IEEE Kobe Techno-Oceans (OTO)*, 2018, pp. 1–4. DOI: `10.1109/OCEANSKOBE.2018.8559435`.

[29] M. K. Larsen, *Terrain-based navigation for unmanned underwater vehicles using visual simultaneous localization and mapping*, eng, 2021. [Online]. Available: `https://hdl.handle.net/11250/2824614`.

[30] A. Kendall, M. Grimes and R. Cipolla, 'Posenet: A convolutional network for real-time 6-dof camera relocalization,' in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Dec. 2015.

[31] A. Kendall and R. Cipolla, 'Geometric loss functions for camera pose regression with deep learning,' in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jul. 2017.

[32] A. Kendall, Y. Gal and R. Cipolla, 'Multi-task learning using uncertainty to weigh losses for scene geometry and semantics,' in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2018.

[33] N. Piasco, D. Sidibé, C. Demonceaux and V. Gouet-Brunet, 'A survey on visual-based localization: On the benefit of heterogeneous data,' *Pattern Recognition*, vol. 74, pp. 90–109, 2018, ISSN: 0031-3203. DOI: `https://doi.org/10.1016/j.patcog.2017.09.013`. [Online]. Available: `https://www.sciencedirect.com/science/article/pii/S0031320317303448`.

[34] E. Brachmann, A. Krull, S. Nowozin, J. Shotton, F. Michel, S. Gumhold and C. Rother, 'Dsac - differentiable ransac for camera localization,' in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jul. 2017.

[35] E. Brachmann and C. Rother, 'Learning less is more - 6d camera localization via 3d surface regression,' in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2018.

[36] P.-E. Sarlin, C. Cadena, R. Siegwart and M. Dymczyk, 'From coarse to fine: Robust hierarchical localization at large scale,' in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2019.

[37] P.-E. Sarlin, A. Unagar, M. Larsson, H. Germain, C. Toft, V. Larsson, M. Pollefeys, V. Lepetit, L. Hammarstrand, F. Kahl and T. Sattler, 'Back to the feature: Learning robust camera localization from pixels to pose,' in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2021, pp. 3247–3257.

[38] M. J. Islam, J. Mo and J. Sattar, 'Robot-to-robot relative pose estimation using humans as markers,' en, *Auton. Robots*, vol. 45, no. 4, pp. 579–593, May 2021.

[39] C. Boittiaux, R. Marxer, C. Dune, A. Arnaubec and V. Hugel, 'Homography-based loss function for camera pose regression,' *IEEE Robotics and Automation Letters*, vol. 7, no. 3, pp. 6242–6249, 2022. DOI: `10.1109/LRA.2022.3168329`.

[40] D. Eigen, C. Puhrsch and R. Fergus, 'Depth map prediction from a single image using a multi-scale deep network,' *Advances in neural information processing systems*, vol. 27, 2014.

[41] R. Garg, N. Wadhwa, S. Ansari and J. T. Barron, 'Learning single camera depth estimation using dual-pixels,' in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 7628–7637.

[42] H. Fu, M. Gong, C. Wang, K. Batmanghelich and D. Tao, 'Deep ordinal regression network for monocular depth estimation,' in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 2002–2011.

[43] L. Huynh, P. Nguyen-Ha, J. Matas, E. Rahtu and J. Heikkilä, 'Guiding monocular depth estimation using depth-attention volume,' in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXVI 16*, Springer, 2020, pp. 581–597.

[44] W. Chen, S. Qian and J. Deng, 'Learning single-image depth from videos using quality assessment networks,' in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2019.

[45] Z. Li, T. Dekel, F. Cole, R. Tucker, N. Snavely, C. Liu and W. T. Freeman, 'Learning the depths of moving people by watching frozen people,' in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2019.

[46] Z. Li and N. Snavely, 'Megadepth: Learning single-view depth prediction from internet photos,' in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2018.

[47] C. Wang, S. Lucey, F. Perazzi and O. Wang, 'Web stereo video supervision for depth prediction from dynamic scenes,' in *2019 International Conference on 3D Vision (3DV)*, 2019, pp. 348–357. DOI: 10.1109/3DV.2019.00046.

[48] P. L. Drews, E. R. Nascimento, S. S. Botelho and M. F. Montenegro Campos, 'Underwater depth estimation and image restoration based on single images,' *IEEE Computer Graphics and Applications*, vol. 36, no. 2, pp. 24–35, 2016. DOI: 10.1109/MCG.2016.26.

[49] X. Ding, Y. Wang, J. Zhang and X. Fu, 'Underwater image dehaze using scene depth estimation with adaptive color correction,' in *OCEANS 2017 - Aberdeen*, 2017, pp. 1–5. DOI: 10.1109/OCEANSE.2017.8084665.

[50] X. Ye, Z. Li, B. Sun, Z. Wang, R. Xu, H. Li and X. Fan, 'Deep joint depth estimation and color correction from monocular underwater images based on unsupervised adaptation networks,' *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 30, no. 11, pp. 3995–4008, 2020. DOI: 10.1109/TCSVT.2019.2958950.

[51] P. Hambarde, S. Murala and A. Dhall, 'Uw-gan: Single-image depth estimation and image enhancement for underwater images,' *IEEE Transactions on Instrumentation and Measurement*, vol. 70, pp. 1–12, 2021. DOI: 10.1109/TIM.2021.3120130.

[52] B. Yu, J. Wu and M. J. Islam, 'Udepth: Fast monocular depth estimation for visually-guided underwater robots,' in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, 2023, pp. 3116–3123. DOI: 10.1109/ICRA48891.2023.10161471.

[53] V. Guizilini, R. Ambrus, S. Pillai, A. Raventos and A. Gaidon, '3d packing for self-supervised monocular depth estimation,' in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2020.

[54] M. Klingner, J.-A. Termöhlen, J. Mikolajczyk and T. Fingscheidt, 'Self supervised monocular depth estimation: Solving the dynamic object problem by semantic guidance,' in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XX 16*, Springer, 2020, pp. 582–600.

[55] W. Zhao, S. Liu, Y. Shu and Y.-J. Liu, 'Towards better generalization: Joint depth-pose learning without posenet,' in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2020.

[56] C. Boittiaux, C. Dune, M. Ferrera, A. Arnaubec, R. Marxer, M. Matabos, L. Van Audenhaege and V. Hugel, 'Eiffel tower: A deep-sea underwater dataset for long-term visual localization,' *The International Journal of Robotics Research*, vol. 42, no. 9, pp. 689–699, 2023.

[57] P. G. O. Zwilgmeyer, M. Yip, A. L. Teigen, R. Mester and A. Stahl, 'The varos synthetic underwater data set: Towards realistic multi-sensor underwater data with ground truth,' in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops*, Oct. 2021, pp. 3722–3730.

[58] D. Levy, A. Peleg, N. Pearl, D. Rosenbaum, D. Akkaynak, S. Korman and T. Treibitz, 'Seathru-nerf: Neural radiance fields in scattering media,' in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 56–65.

[59] 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. DOI: 10.1109/CVPR.2016.90.

[60] C. Campos, R. Elvira, J. J. G. Rodríguez, J. M. M. Montiel and J. D. Tardós, 'Orb-slam3: An accurate open-source library for visual, visual–inertial, and multimap slam,' *IEEE Transactions on Robotics*, vol. 37, no. 6, pp. 1874–1890, 2021. DOI: 10.1109/TRO.2021.3075644.

[61] D. Akkaynak and T. Treibitz, 'Sea-thru: A method for removing water from underwater images,' in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 1682–1691. DOI: 10.1109/CVPR.2019.00178.

[62] J. Gibson, *Improving sea-thru with monocular depth estimation methods*, https://github.com/hainh/sea-thru, 2020.

[63] S. Umeyama, 'Least squares estimation of transformation parameters between two point patterns,' *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 13, no. 4, pp. 376–380, 1991. DOI: 10.1109/34.88573.

[64] M. Grupp, *Evo: Python package for the evaluation of odometry and slam.* https://github.com/MichaelGrupp/evo, 2017.

[65] K. Panetta, C. Gao and S. Agaian, 'Human-visual-system-inspired underwater image quality measures,' *IEEE Journal of Oceanic Engineering*, vol. 41, no. 3, pp. 541–551, 2016. DOI: 10.1109/JOE.2015.2469915.

[66] M. Yang and A. Sowmya, 'An underwater color image quality evaluation metric,' *IEEE Transactions on Image Processing*, vol. 24, no. 12, pp. 6062–6071, 2015. DOI: `10.1109/TIP.2015.2491020`.

[67] C. Boittiaux, R. Marxer, C. Dune, A. Arnaubec, M. Ferrera and V. Hugel, 'SUCRe: Leveraging scene structure for underwater color restoration,' in *3DV*, 2024.

Appendix A

Additional Material

The code for the project is available in this github repository:
`https://github.com/AyonRRahman/Thesis`

NTNU

Norwegian University of
Science and Technology

MIR

MARINE &
MARITIME
INTELLIGENT
ROBOTICS