Learning Underwater Motion and 3D Reconstruction from Optical and Acoustic Sensors

Shubham M Wagh

Supervised by:

Dr. Sen Wang Prof. Yvan Petillot

Ocean Systems Laboratory
Institute of Sensors, Signals and Systems
Heriot-Watt University



A Thesis Submitted for the Degree of
MSc Erasmus Mundus in Vision and Robotics (VIBOT)

· 2018 ·





Abstract

Autonomous Underwater Vehicles (AUVs) are in increasing demand to perform different underwater tasks such as coral reef monitoring, harbours security, archaeology, oil and gas industry and mine countermeasure missions. However, its autonomy still presents some challenges in terms of perception, navigation and localization. For instance, the perception range of optical sensors has limited visibility where as sonar images are quite noisy due to high acoustic clutter. This implicitly affects the navigation and localization of AUV. Although the task has been addressed and accomplished, but heavily relies on manual troubleshooting to analyze failure cases and refine localization results. One of the key features still missing from the current system is the capability to automatically gain knowledge and improve performance through end-to-end learning.

Deep Convolutional Neural Networks (DCNN) have shown record-shattering performances in a variety of computer vision problems. CNNs have been recently used to solve the problem of localization by estimating depth and pose using images on land. They perform appreciatively compared to established SLAM systems under similar input settings.

Based on the recent works, we propose two unsupervised learning framework for 3D reconstruction and motion estimation in underwater scenarios using raw optical and sonar image frames individually. Unsupervised learning removes the need for separate supervisory signals (depth/elevation or ego-motion ground truth, or multi-view frames). To the best of our knowledge, no previous system exist that learns motion estimates and 3D information (i.e depth map from optical images and elevation map from sonar images) in an unsupervised fashion for underwater environments. We trained the two models using raw sequences of underwater optical and sonar images respectively. The results of the experiments demonstrate logical results and showed a great potential in improving the performance if supplemented with large number of good datasets both for sonar and optical images.

Nothing in the world can take the place of persistence. Talent will not; nothing is more common than unsuccessful men with talent. Genius will not; unrewarded genius is almost a proverb. Education will not; the world is full of educated derelicts. Persistence and determination alone are omnipotent.

- Calvin Coolidge

Contents

Acknowledgments **ix**

1 Introduction **1**

1.1 Motivation 1

1.2 Problem Statement 2

1.3 Objectives 3

1.4 Project Management 4

1.5 Contributions and Outline 5

2 Background **6**

2.1 Optical Imaging Geometry 6

2.2 Forward Looking Sonar Imaging Geometry 8

2.3 Comparing Monocular Camera and FLS Imagery 10

2.3.1 Analogy 10

2.3.2 Key Differences 11

2.4 Deep Learning 11

2.4.1 Convolutional Neural Networks 12

2.4.2 Back Propagation 14

2.4.3 Unsupervised Deep Learning 15

3 Literature Review **18**

3.1 Underwater 3D Reconstruction 18

3.1.1 Optical Sensor based Methods 18

3.1.2 Acoustic Sensor based Methods 20

3.1.3 Opti-Acoustic Sensor based Methods 23

3.2 Underwater Motion Estimation 23

3.2.1 Motion Estimation using Optical Sensor 24

3.2.2 Motion Estimation using Acoustic Sensor 25

3.2.3 Motion Estimation using Opti-Acoustic Sensor 25

3.3 Deep Learning Methods 26

3.3.1 Deep Learning for Depth Estimation 26

3.3.2 Deep Learning for Pose Estimation 28

4 Methodology **32**

4.1 Proposal Description 32

4.2 Depth Map and Ego-Motion from Optical Images 33

4.2.1 Problem Geometry 33

4.2.2 Network Architecture 35

4.2.3 Loss Functions 35

4.2.4 Learning Setup 37

4.3 Elevation Map and Ego-Motion from Sonar Images 38

4.3.1 Problem Geometry 38

4.3.2 Proof of Concept 40

4.3.3 SonarNet Architecture 43

4.3.4 Learning Setup 44

5 Experiments and Results **46**

5.1 Depth Map and Pose for Monocular Camera Images 46

5.1.1 Training Details 46

5.1.2 Single View Depth Estimation 47
5.1.3 Pose Estimation 50
5.2 Elevation Map and Pose for Sonar Images 51
5.2.1 Training Details 52
5.2.2 Elevation Estimation 52
5.2.3 Pose estimation 55

6 Conclusion **58**
6.1 Final Remarks 58
6.2 Challenges 59
6.3 Future Work 61

A Miscellaneous **62**
A.1 ARIS 3000 Sonar System Configuration 62
A.2 Pixel coordinate to Real world coordinate for Sonar Imagery 62
A.3 Real world coordinate to pixel coordinate for Sonar Imagery 64
A.4 Training loss curves - optical images 64
A.5 Training loss curves - sonar images 65

Bibliography **73**

List of Figures

1.1 Motion estimation and 3D reconstruction answers for "*where am I?*" question . . 3

1.2 Master thesis planning. The project is planned per month. 4

2.1 Optical Imaging geometry process under perspective projection of a pinhole camera [71] 7

2.2 Coordinate system and sonar projection 8

2.3 Sonar projection geometry. A 3D point $\mathbf{P}(R, \theta, \phi)$ is mapped into a point p on the image plane along the arc defined by the elevation angle. 9

2.4 Optical and Sonar image of a scene on a water tank bottom [10] 10

2.5 Graphs of Sigmoid and ReLU actiavtion function [65] 13

2.6 Simple CNN architecture 14

2.7 Autoencoder architectures 16

2.8 Convolutional Autoencoder architecture: It maps a wide and thin input space to narrow and thick latent space [20] 17

3.1 3D reconstruction from SV using graph SLAM (25×10 m, Mallorca) [16] 20

3.2 Reconstruction of concave hemi-cylinder - Aykin *et al.* [7] 21

3.3 3D reconstruction pipeline - Guerneve *et al.* [33] 22

3.4 Scheme of the algorithm - Garcia *et al.* [27] 24

3.5 Model Architecture - Eigen *et al.* [22] 27

3.6 CNN-SLAM Overview - Tateno *et al.* [68] 27

3.7 Overview pipeline of deep CNN for single view depth - Garg *et al.* [29] 28

3.8 Sampling strategies for backward mapping. This approach uses the left image to produce disparities for both images, improving quality by enforcing mutual consistency - Godard *et al.* [31]. 28

3.9 Overview pipeline of SfMLearner architecture - Zhou *et al.* [73] 29

3.10 Overview pipeline of PoseNet - Kendall *et al.* [40] 29

3.11 Architecture overview of dual-stream CNN for indoor relocalization - Li *et al.* [45] 30

3.12 Training scheme of UnDeepVO - *Li et al.* [46] 31

4.1 Overview of the supervision pipeline based on view synthesis. The depth network takes only the target view as input, and outputs a per-pixel depth map D_t. The pose network takes both the target view (I_t) and the nearby/source views (e.g., I_{t-1} and I_{t+1}) as input, and outputs the relative camera poses ($T_{t\to t-1}, T_{t\to t+1}$). The outputs of both networks are then used to inverse warp the source views using Q_t to reconstruct the target view, and the image reconstruction loss is used for training the CNNs. By utilizing view synthesis as supervision, we are able to train the entire framework in an unsupervised manner from videos. 34

4.2 Network architecture for our depth/pose prediction modules. The width and height of each rectangular block indicates the output channels and the spatial dimension of the feature map at the corresponding layer respectively, and each reduction/increase in size indicates a change by the factor of 2. For single-view depth network, we adopt the SfMLearner architecture which is in turn based on DispNet [48] architecture with multi-scale side predictions. (b) The pose networks predict 6-DoF. relative pose. 36

4.3 (a) Polar form of sonar image taken in a water tank containing an old Tyre (viewing direction is from the top). (b) Cartesian form of sonar image (viewing direction from the bottom). (c) Synthetically generated elevation map of sonar image 40

4.4 Object and shadow detections based on thresholds T_o and T_s, respectively [66] . 41

4.5 (a) Original sonar image. (b) Synthesized sonar image at a relative distance of 0.5m 42

4.6 Sonar image transformations 42

4.7 Overview of the supervision pipeline based on view synthesis. The elevation network takes only the target view as input, and outputs a per-pixel elevation map E_t. The pose network takes both the target view (S_t) and the nearby/source views (e.g., S_{t-1} and S_{t+1}) as input, and outputs the relative camera poses ($T_{t\to t-1}, T_{t\to t+1}$). The outputs of both networks are then used to inverse warp the source views using Q_t to reconstruct the target view, and the image reconstruction loss is used for training the CNNs. By utilizing view synthesis as supervision, we are able to train the entire framework in an unsupervised manner from raw sonar image dataset. 43

4.8 Network architecture for the SonarNet model. The width and height of each rectangular block indicates the output channels and the spatial dimension of the feature map at the corresponding layer respectively, and each reduction/increase in size indicates a change by the factor of 2. For single-view depth network, we adopt the SfMLearner architecture which is in turn based on DispNet [48] architecture with multi-scale side predictions. (b) The pose networks predict 6-DoF. relative pose. 45

5.1 Our sample depth predictions on the Girona's underwater dataset using the model trained on the same dataset 47

5.2 Example 1 - Reconstructed 3D point of an input image with different views . . . 48

5.3 Example 2 - Reconstructed 3D point of an input image with different views . . . 49

5.4 Example 3 - Reconstructed 3D point of an input image with different views . . . 49

5.5 Depth map predictions affected due to artificial illumination 50

5.6 (a) Original Image, (b) CLAHE processed image and (c) Depth Map prediction. The bright spots become even more effective. Thereby the depth prediction gets affected 51
5.7 2D pose trajectory of camera for Girona's underwater dataset 51
5.8 Our sample elevation predictions on the sonar underwater dataset collected in a water tank using the model trained on the same dataset 52
5.9 Sonar image view synthesis 53
5.10 Case when sonar image is very sparse thereby affecting the training process and predictions 54
5.11 Elevation map predictions affected 54
5.12 Sonar image frames in Cartesian form with ocean color map 55
5.13 Sonar pose trajectory for images shown in fig. 5.12 image frames 56
5.14 Point cloud of a scene using input image shown in fig. 5.9 57

A.1 Raw sonar image specifications. View from the top 63
A.2 Image reconstruction and structured similarity loss curves during training upto 200,000 iterations 64
A.3 Depth smoothness and total loss curves during training upto 200,000 iterations . 65
A.4 Image reconstruction, elevation smoothness and total loss curves during training up to 300,000 iterations 66

List of Tables

3.1 Summary of optical 3D reconstruction solutions 20
3.2 Summary of sonar 3D reconstruction solutions 22
3.3 Summary of opti-acoustic 3D reconstruction solutions 23

Acknowledgments

First of all, I would like to thank my supervisors, Dr. Sen Wang and Dr. Yvan Petillot, for the opportunity they gave me to work under their guidance. This thesis would not have been possible without the continuous support from my advisor Dr. Sen Wang, who always believed in me and gave me so many constructive suggestions. Thank you for your patience, effort and trust.

I would like to acknowledge the contribution of all the teachers and everyone involved in the VIBOT program for providing me with the background in computer vision and robotics to work in this project. I would also like to acknowledge the Girona Underwater Vision and Robotics group and Ocean Systems Lab, Heriot Watt University for providing the dataset.

Finally, the warmest gratitude to my family and friends. Their daily encouragement and moral support have always pushed me to give my best.

Chapter 1

Introduction

1.1 Motivation

Over the past few years, Autonomous Underwater Vehicles (AUVs) have greatly improved as a tool for underwater survey and exploration. They have been increasingly used to perform different underwater missions such as ship hull inspection, pipeline survey, coral reef monitoring, harbour patrolling, mine counter-measures, dam inspection and seabed mapping. Due to the nature of underwater environments, sonar sensors are usually used over optical sensors to achieve full autonomy and accurate localization. However, its autonomy still presents some challenges in terms of perception, navigation and localization. Earlier methods make use of a range of possibilities: inertial and Doppler velocity sensors, acoustics beacons, 3d sonar imaging system, terrain based navigation system and Simultaneous Localization And Mapping (SLAM) based approaches for building a map of the vehicle's environment and using it to obtain estimates of its location. These aforementioned sensors at the same time increase the payload and power consumption of AUVs, hence making it undesirable. Many research centres and commercial companies are striving to achieve robust autonomy in underwater scenarios totally based on vision sensors (optical camera and/or sonar camera) in order to make the operation more efficient.

In recent years, data-driven or deep learning based approach has drawn significant attention due to its potential in learning capability and robustness to challenging environments [46]. These techniques have proved to be quite effective in a variety of computer vision problems and have achieved a breakthrough performance in natural images and video processing. However, it is still not explored how these algorithms will perform scene reconstruction and motion estimation in the underwater environments thereby yielding estimates of the vehicle's location.

Convolutional Neural Networks (CNNs) based deep learning approach are now state-of-the-art for object detection [30], per-pixel image classification [60], human pose estimation [24], depth and normal prediction from single image [44] and many other applications. These models learn directly from the data at different levels of abstractions. Intuitively, the initial layers of these models help in identifying low-level characteristics (e.g. points, edges, borders) while higher layers contain more selective and semantic knowledge of the data (e.g. specific arrangements of the low-level features forming patterns or shapes).

On seeing the recent advancements in CNN based deep learning approach for land images, we believe that these techniques can be employed to solve the challenging problem of localization and navigation in underwater environments too. And in coming years, deep learning approach might be able to help achieve complete autonomy of AUVs in underwater exploration and mapping task.

1.2 Problem Statement

In recent years, there has been an increasing need for autonomous vehicles that operate underwater. AUVs have many applications as they perform dangerous and monotonous tasks like ship hull inspection, coral reef monitoring, surveying etc. AUVs should be able to localize itself accurately in the environment in order to carry out these operations efficiently and autonomously. However, its autonomy presents challenges on multiple levels in terms of localization and navigation. In order for an AUV to localize itself in the underwater environment, it should be able to build a map of its surrounding environment and then use the built map to obtain location estimates. There are existing systems like acoustic beacons, inertial and doppler velocity sensors, GPS, 3D sonar imaging systems etc. which make use of the measurements from these sensors and use SLAM based approaches to build the map thereby localizing the vehicle. However, these sensors are not cost effective, increases the payload of AUV and at the same time increases power consumption, thereby making it undesirable. The idea is to make complete use of vision sensors to perform 3D scene reconstruction and motion estimation thereby giving estimates of the vehicle's location in the environment.

The key elements i.e. 3D reconstruction and motion estimation in AUVs answer the question "*Where am I* ?" in two different ways [6].

- 3D reconstruction gives 3D coordinates of all landmarks sensed by the AUV.
- Motion estimation gives 3D motion vector relative to another position in the AUV path.

There are state-of-the-art geometric feature based methods which make use of vision sensors (optical camera and/or sonar sensor) for 3D scene reconstruction and ego-motion estimation

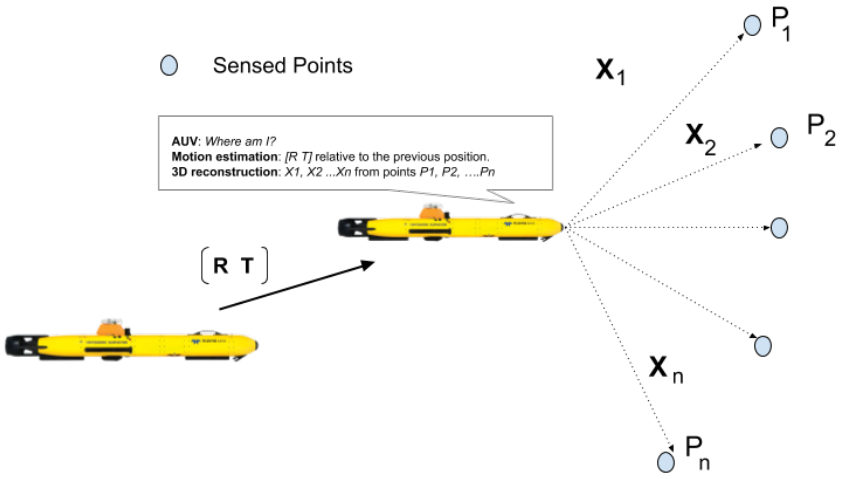


Figure 1.1: Motion estimation and 3D reconstruction answers for "*where am I?*" question

and have demonstrated an incredible performance in terms of accuracy. However, one of the key features still missing from the current system is the capability to automatically gain knowledge and improve performance through end-to-end learning. The current systems heavily rely on manual troubleshooting to analyze failure cases and refine localization results [69].

Deep learning approach have shown great success in a variety of computer vision problems, such as visual object recognition, detection, segmentation etc. CNNs based deep learning are now not only the state-of-the-art but also allowed easy representation of complex functions through increasing layer of numbers in a model. However, it is still not explored how CNN models will perform 3D reconstruction and motion estimation in the underwater scenarios. Also, considering the fact that most of the surveying and mapping tasks in underwater scenarios are carried out in turbid and murky environments where AUVs equipped with optical systems are constrained by their limited visibility range and/or 2D sonar systems constrained by high acoustic clutter, it is not clear as to how CNN models will perform scene reconstruction and pose estimation when given hazy underwater optical image or noisy 2D sonar image (which are effective as compared to optical images in underwater environment) as input.

1.3 Objectives

In this master thesis, we propose an **end-to-end unsupervised deep learning framework for 3D reconstruction and motion estimation** in underwater scenarios using optical images and sonar images individually. To the best of our knowledge, no previous systems exist that

Task	Feb	March	April	May	June
Literature Review - Underwater 3D reconstruction and Motion Estimation					
Deep Learning - state of the art models review and implementation					
Proposed unsupervised learning model design and implementation using optical images					
Unsupervised learning concept development using sonar images and validation					
Proposed unsupervised learning model design and implementation using sonar images					
Dissertation, paper and poster writing					
Defense presentation preparation					

Figure 1.2: Master thesis planning. The project is planned per month.

learns motion estimates and 3D information (i.e. depth map from optical images and elevation map from sonar images) in an unsupervised manner from raw image sequences in underwater environments. As this is the first attempt ever to do so, we believe that the outcome of this project will be used to improve and enhance the autonomy of AUVs in these extreme challenging underwater scenarios.

1.4 Project Management

The different specific tasks are listed as follows and the corresponding planning is presented in fig. 1.2

1. Review state of the art methods on underwater 3D reconstruction and motion estimation. The goal is to understand how scene reconstruction and ego-motions for underwater scenarios has been addressed in the literature, the problem geometry and analyze their advantages and limitations.

2. Review recent state-of-the-art deep learning models for scene reconstruction and motion estimation in general environment. The intention of this task is to understand the network architecture and their behaviour so as to develop a network for the given problem.

3. Propose an unsupervised learning framework using optical images of underwater environment based on state-of-the-art. The goal of this task is to predict depth map and pose estimation using the proposed framework.

4. Develop the concept of unsupervised learning using sonar images for elevation map and pose prediction based on the model proposed using optical images. The goal is to understand and frame a problem geometry for sonar images.

5. Propose an unsupervised learning model using sonar image sequences for elevation and

pose estimation in underwater scenarios and conduct experiments to validate our implementations.

1.5 Contributions and Outline

In this thesis we propose two unsupervised deep learning framework for underwater scenarios :

- Unsupervised learning of *single view depth map and pose* from optical image sequences.
- Unsupervised learning of *single view elevation map and pose* from polar form sonar image sequences, also known as **SonarNet**.

To the best of our knowledge, this is the first ever attempt to make use of CNN based deep learning framework to solve the problem of 3d reconstruction and ego-motion estimation for underwater environments.

The code of the above two proposed framework will be made available to download in an open repository on *Github* [1], along with the guidelines for the installation of the needed modules.

The reminder of this master thesis is organized as follows:

- **Chapter 2** describes the imaging geometry of optical and sonar cameras, establishes an analogy between these sensors and points out key differences in them. It also provides some background on deep learning techniques and models which are used in this thesis.
- **Chapter 3** explains state of the art methods for 3d reconstruction and pose estimation using optical and sonar images.
- **Chapter 4** addresses the proposed deep learning frameworks in detail.
- **Chapter 5** shows the experiments and results of the implementation.
- Lastly, the document is concluded in **Chapter 6** summarizing the main contributions and the future research directions to be explored.
- **Appendix A** shows sonar sensor settings used, some formulae and training loss curves as a miscellaneous part of this project.

[1] Project source code: `https://github.com/shubhamwagh/MScThesis.git`

Chapter 2

Background

In this chapter, we provide technical background on optical and sonar camera imaging geometry, their analogy and key differences. We also discuss basics of deep learning and describe core building blocks of Covolutional Neural Network (CNN), method to train them and a brief overview on CNN based Autoencoders.

2.1 Optical Imaging Geometry

The basic principle of image formation on analog and/or digital cameras has been taken from the way, the human eye works. When the light falls upon a particular object, it is reflected back after striking the object. The image is formed on the retina or 2D image plane when the light rays pass through the lens of eye or camera forming a particular angle.

In this section, we discus the process of camera image formation in terms of geometry.

Consider a 3D point $\mathbf{P} = [x^w, y^w, z^w]$ in the world coordinate system. As seen, the visual pipeline is capturing the 3D point in world coordinate system (W) to camera coordinate system (C), then is mapped to the physical retina i.e physical image plane to get the image coordinates $\mathbf{p} = [u, v]^T$

For ease of computation, assume a normalized image plane located at the focal length $f = 1$. In this normalized image plane, the pinhole (c) is mapped to the origin of the image plane (c'), and $\mathbf{p}^c$ is mapped to $\mathbf{p'} = [u', v']^T$.

$$p' = \begin{bmatrix} u' \\ v' \\ 1 \end{bmatrix} = \frac{1}{z^c} \begin{bmatrix} \mathbf{I} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{p}^c \\ 1 \end{bmatrix} = \frac{1}{z^c} \begin{bmatrix} \mathbf{I} & \mathbf{0} \end{bmatrix} \begin{bmatrix} x^c \\ y^c \\ z^c \\ 1 \end{bmatrix} \tag{2.1}$$

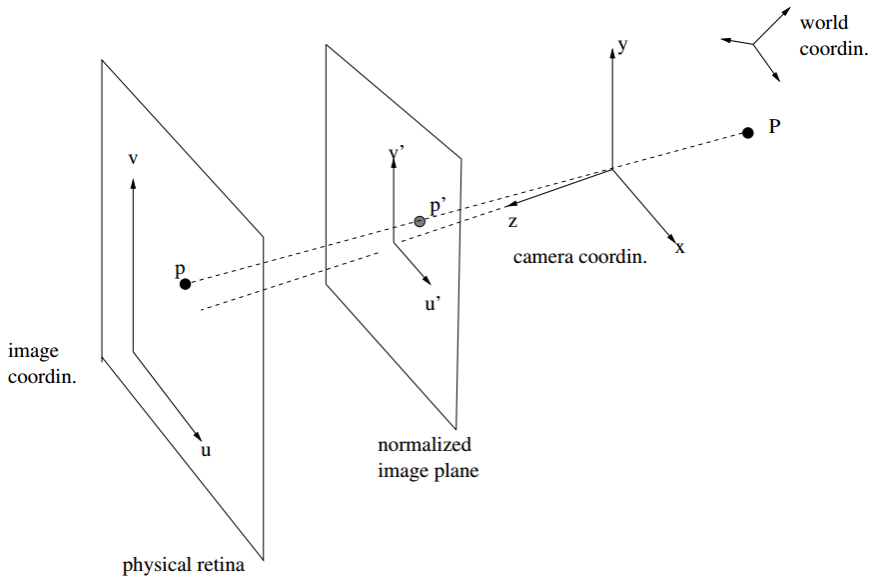


Figure 2.1: Optical Imaging geometry process under perspective projection of a pinhole camera [71]

Also we have

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \frac{1}{z^c} \begin{bmatrix} f_x & 0 & u_0 \\ 0 & f_y & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x^c \\ y^c \\ z^c \end{bmatrix} = \frac{1}{z^c} \begin{bmatrix} f_x & 0 & u_0 \\ 0 & f_y & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \mathbf{I} & \mathbf{0} \end{bmatrix} \begin{bmatrix} x^c \\ y^c \\ z^c \\ 1 \end{bmatrix} \tag{2.2}$$

Here, $f_x = kf$ and $f_y = lf$. Parameters f_x, f_y, u_0 and v_0 are called camera intrinsic parameters. We can write

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \frac{1}{z^c} \begin{bmatrix} f_x & 0 & u_0 & 0 \\ 0 & f_y & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x^c \\ y^c \\ z^c \\ 1 \end{bmatrix} = \frac{1}{z^c} \begin{bmatrix} f_x & 0 & u_0 & 0 \\ 0 & f_y & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0}^T & 1 \end{bmatrix} \begin{bmatrix} x^w \\ y^w \\ z^w \\ 1 \end{bmatrix} \tag{2.3}$$

Here, $\mathbf{R}$ and $\mathbf{t}$ are call extrinsic parameters, which represent the coordinate transformation between the camera coordinate system and the world coordinate system. So, we can write

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \frac{1}{z^c} \mathbf{M}_1 \mathbf{M}_2 \mathbf{P} = \frac{1}{z^c} \mathbf{M} \mathbf{P} \tag{2.4}$$

where, $\mathbf{M}_1$ is an intrinsic camera matrix and $\mathbf{M}_2$ is an extrinsic camera matrix. Hence $\mathbf{M}$ is called projection matrix. This is a very casual representation of the above process happening through the visual pipeline.

2.2 Forward Looking Sonar Imaging Geometry

Forward Looking Sonar (FLS) is an acoustic sensor. The imaging formation principle is quite different as compared to the optical imaging devices. The sonar insonifies the scene with an acoustic wave, spanning its Field of View (FOV) or beam width in azimuth (θ) and elevation (ϕ) directions, and then the intensity of acoustic return is sampled by an array of transducers as a function of range (R) and bearing (θ) [35]. Therefore, the sonar images are arranged and represented in polar coordinates.

In this section, we mainly analyze the imaging geometry of FLS sensor. Consider 3D co-

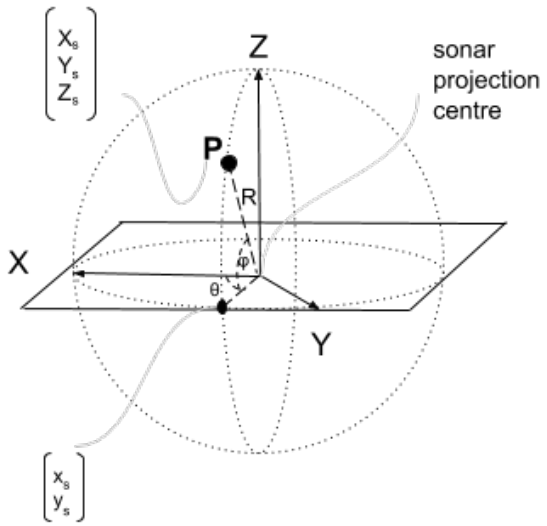


Figure 2.2: Coordinate system and sonar projection

ordinate of a point $\mathbf{P}$ in a sonar coordinate system denoted $\mathbf{P}_s = (X_s, Y_s, Z_s)^T$. This can be represented in spherical coordinates by (R, θ, ϕ), as illustrated in the figure. The conversions

between the Cartesian and the spherical sonar coordinates are given by

$$\mathbf{P}_s = \begin{bmatrix} X_s \\ Y_s \\ Z_s \end{bmatrix} = R \begin{bmatrix} \cos\phi\cos\theta \\ \cos\phi\sin\theta \\ \sin\phi \end{bmatrix} \tag{2.5}$$

and

$$\begin{bmatrix} R \\ \theta \\ \phi \end{bmatrix} = \begin{bmatrix} \sqrt{X_s^2 + Y_s^2 + Z_s^2} \\ \tan^{-1}(Y_s/X_s) \\ \tan^{-1}(Z_s/\sqrt{X_s^2 + Y_s^2}) \end{bmatrix} \tag{2.6}$$

This 3D point $\mathbf{P}_s$ is projected in a point $\mathbf{p} = (x_s, y_s)$ on the image plane $(X_s Y_s)$ following a nonlinear model:

$$\mathbf{p} = \begin{bmatrix} x_s \\ y_s \end{bmatrix} = h(\mathbf{P}_s) = \begin{bmatrix} R\cos\theta \\ R\sin\theta \end{bmatrix} = \frac{1}{\cos\phi} \begin{bmatrix} X_s \\ Y_s \end{bmatrix} = \begin{bmatrix} X_s \\ Y_s \end{bmatrix}_{\phi=0} \tag{2.7}$$

As can be seen in Equation 2.7, the projection is introduced as a function of the elevation

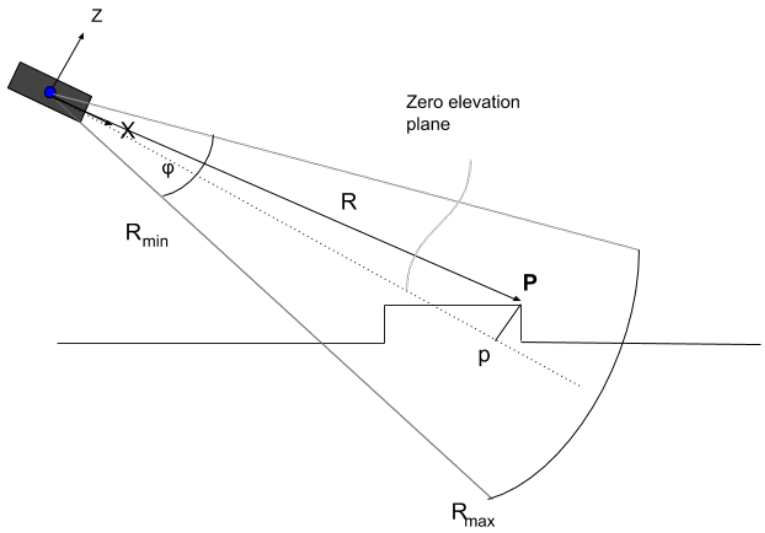


Figure 2.3: Sonar projection geometry. A 3D point $\mathbf{P}(R, \theta, \phi)$ is mapped into a point p on the image plane along the arc defined by the elevation angle.

angle. Hence, each image point can be interpreted as the mapping $h(\mathbf{P}_s)$ of 3D scene points $\mathbf{P}_s$ onto the zero elevation $(X_s Y_s)$ plane, along meridians defined by the range (R) and azimuth angles (θ) of the 3D point (fig. 2.3). Therefore the dimensions of a raw frame correspond to

the number of beams in the angular direction and the number of range samples in the range axes. This representation is then converted to the final 2D image in Cartesian coordinates for an easier interpretation. It is worth noting that this process produces images with non-uniform resolution as one pixel in polar domain is mapped to multiple pixels with the same intensity in the Cartesian coordinates as the range increases [35].

2.3 Comparing Monocular Camera and FLS Imagery

In the previous section, we have seen that the imaging geometry of optical camera and acoustic FLS sensor is completely different. The former makes use of light to form an image while the latter forms an image by insonifying the scene with an acoustic wave.

Although both the sensors have different imaging principles and geometries, but an analogy can be established between the two in terms of the type of information conveyed through the images which is obtained from each of these sensors.

2.3.1 Analogy

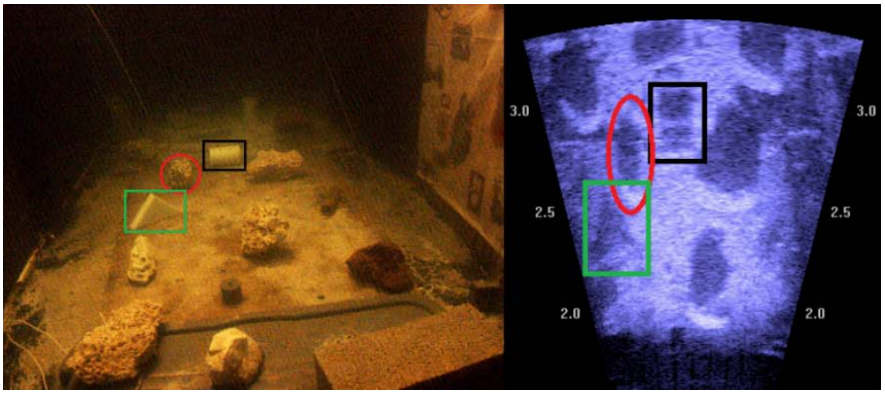


Figure 2.4: Optical and Sonar image of a scene on a water tank bottom [10]

1. **Monocular Camera** :

   - The 3D information is lost during the process of projecting into a 2D image as the reflected light rays could have originated from anywhere along the depth (z) of the scene.
   - The position of the object in scene can be retrieved from the image using the intrinsic camera parameters.

2. **Forward Looking Sonar** :

   - The 3D information is lost during the process of projecting into a 2D image as the reflected acoustic wave could have originated from anywhere along the corresponding elevation (ϕ) arc.
   - The position of the object in the scene can be retrieved from the sonar image i.e by linearly scaling (R, θ) coordinates of the polar image or (x_s, y_s) coordinates of Cartesian image.

From the above mentioned points, we see that in each of the above described sensor's image formation process, one information is lost which is essential enough to understand the real world scene.

2.3.2 Key Differences

1. **Monocular Camera** :

   - Working principle is based on light i.e an optical sensor
   - Each pixel in the image contributes to some information in the real scene.
   - The color, texture etc. of the object is retained.

2. **Forward Looking Sonar** :

   - Working principle is based on acoustic wave i.e an acoustic sensor.
   - Sonar image is usually very noisy. And not every pixel in the image contributes to information in the real scene.
   - The color, texture etc. of the object is lost.

2.4 Deep Learning

Deep learning, also known as Deep Structured Learning, Hierarchical Learning or Deep Machine Learning is a specific subfield of machine learning. It is a new take on learning representations from data that puts an emphasis on learning successive layers of increasingly meaningful representations. In deep learning, these layered representations are learned via models called neural networks, structured in literal layers stacked on top of each other. The structure of neural network is like any another kind of network; there is an interconnected web of nodes called *neurons* and the *edges* that join them together. A neural network's main function is to receive a set of inputs, perform progressively complex calculations and then use the output to solve a

problem. Neural Networks are used in a lot of different applications like object classification, recognition, handwriting transcription, machine translation etc.

In this section we particularly discuss about neural networks for computer vision problems i.e Convolutional Neural Networks (CNNs). Further we also deliberate about a method to train CNNs and unsupervised deep learning model based on CNN.

2.4.1 Convolutional Neural Networks

One of the most widely adopted approaches of deep neural networks is the Convolutional Neural Networks, or CNNs, also known as *convnets*. They are a specialized kind of neural network for processing data that has a known array or grid-like topology. CNNs are simply neural networks that use convolution operation in place of general matrix multiplication in at least one of their layers [32]. In this section, we discuss the building blocks of CNNs and study their characteristic properties.

Convolution Layer

The convolution layer is the core building block of CNN architecture. Each convolution layer contains a set of filters which are learned in the training process. Each one of these kernels is slide over the entire input image to extract patches and applies the same transformation to all of these patches, producing an output *feature map*. These feature maps encode specific aspects of the input data: initial layers extract simple features whereas the ones in deeper layers represent more complex and high-level structures. This key characteristic gives *convnets* two interesting properties [18]:

- The patterns they learn are *translation invariant* i.e after learning a certain pattern in the lower-right corner of an image, a *convnet* can recognize it anywhere.
- They can learn *spatial hierarchies of patterns* i.e first convolution layer will learn small local patterns such as edges, a second convolution layer will learn larger patterns based on features of the previous layer and so on.

Non Linearity Layer

The above convolution layer performs several convolutions in parallel to produce a set of linear activations. Each linear activation is run through a non linear activation function to produce non-linear decision boundary via linear combinations of the weighted inputs. This is to make the network more powerful and add ability to learn complex and complicated data form and represent non-linear complex arbitrary functional mappings between inputs and outputs.

Two of the most common activation functions used in this project are the *Sigmoid activation function* and *Rectifier Linear Units*, referred as ReLU.

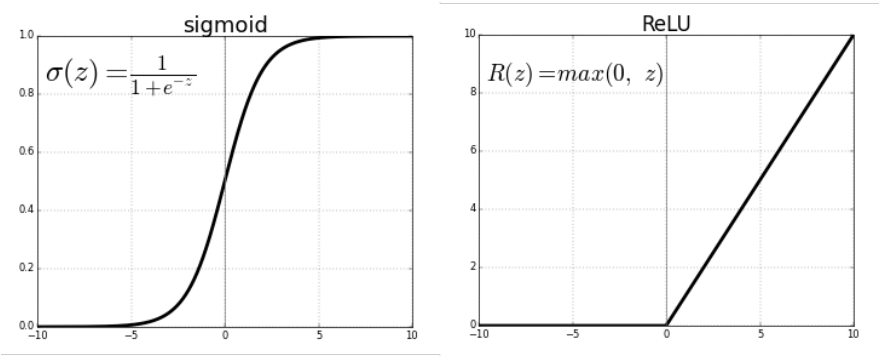


Figure 2.5: Graphs of Sigmoid and ReLU actiavtion function [65]

1. **Sigmoid Activation function** : It is a activation function of form

$$\sigma(z) = \frac{1}{1 + \exp(-z)}$$

   Its range is between 0 and 1. However, it has some drawbacks due to vanishing gradient problem and slow convergence.

2. **ReLU** : It sets the negative values in the feature map to zero.

$$R(z) = \max(0, z)$$

   It avoids and rectifies vanishing gradient problem. However, a problem with ReLU is that some gradients can be fragile during training and can die. It can cause a weight update which will make it never activate on any data point again [39].

Pooling Layer

Pooling layers are periodically inserted in between the convolution layers in *convnet* architectures. Its function is to extract the summary statistics of non-overlapping neighbourhoods thereby progressively reducing the spatial size of the representation to reduce the amount of parameters and computation in the network, and hence to also control overfitting [12]. Among several pooling options, max pooling is the most common approach due to its empirically demonstrated performance.

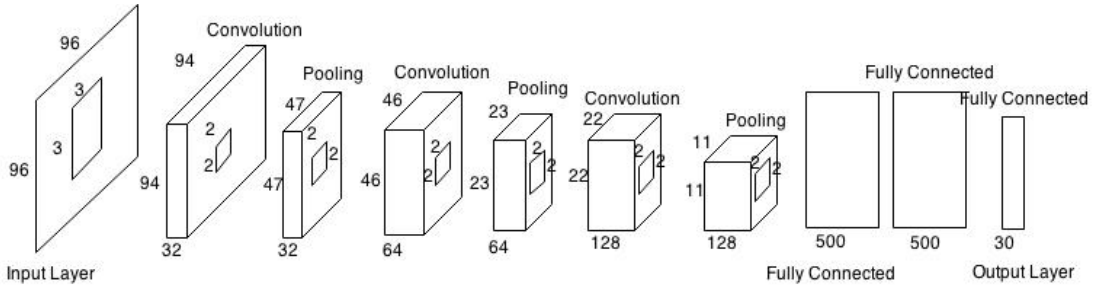


Figure 2.6: Simple CNN architecture

Normalization

Another important CNN building block is channel-wise normalization. This operator normalizes the vector of feature channels at each spatial location in the input map [39].

Fully Connected Layer

After a series of convolution and max pooling layers, the high level reasoning is done by the fully connected layer (FC). Each neuron in a fully connected layer is connected to all the activations in the previous layer and their activations can be computed by a simple matrix multiplication followed by a bias offset [39]. The process usually consists in flattening the feature maps coming from convolutional layers, to achieve a one dimension feature vector representation, and, then, inputting it into the FC layer.

Loss Layer

The loss layer manages the modifications of weights across the network. Before the training of the network begins, the weights in the convolution and fully-connected layers are randomly initialized. Then during training, the loss layer continually checks the fully-connected layer's guesses against the actual values with the goal of minimizing the difference between the guess and the real value as much as possible. The loss layer does this by adjusting the weights in both the convolution and fully-connected layers [5].

2.4.2 Back Propagation

Back propagation is a method of training the neural networks. Firstly the input map is passed through a series of convolution and pooling layers and subsequently to FC layer to get the predicted output. This is called *forward propagation* pass. Then this output is compared against the ground truth value in a loss layer and an error is computed. This error is then back

propagated into the network i.e in other words it is fed back to the network for optimizing the inter neuron weights. These weights are updated using the gradients. This updating occurs first for the out-most weights, and then for lower layers. The intuition is that the errors with respect to the outmost neurons are known first, and then the errors of the inner neurons are derived from them. Each layers errors are calculated from the errors of the neurons they provide input to [37]. This elaborated method to automatically adjust the weight is the called *backpropagation* algorithm.

2.4.3 Unsupervised Deep Learning

As discussed earlier, deep learning has the ability to recognize incredibly complex patterns. Especially neural networks. As patterns get more complex, neural networks outperform all competition. But, the majority of practical deep learning uses *supervised deep learning*. When for a given input variable (x) its corresponding output (y) is known then CNNs are used to learn the mapping function from the input to the output.

$$y = f(x)$$

This is called supervised deep learning. If we have a dataset that is properly labelled be it images, audio or text, we are in luck! Deep Learning works really well for labelled datasets. That's because there is always a function that represents the relationship between the input and output variables. The goal is to approximate the mapping function so well that for a new input data its corresponding output can be predicted robustly. Supervised deep learning have proved to be effective in various learning problems like classification and regression [15].

In the current big data era, very few datasets are properly labelled. Though supervised deep learning gives a structure for solving a problem, but it limits the potential of that algorithm in two ways:

- *It is bound by the biases in which it is being supervised* in i.e. it is prohibited to think of other corner cases that could occur when solving the problem.

- As the learning is supervised - *there is a huge manual effort involved in creating the labels for the algorithm.*

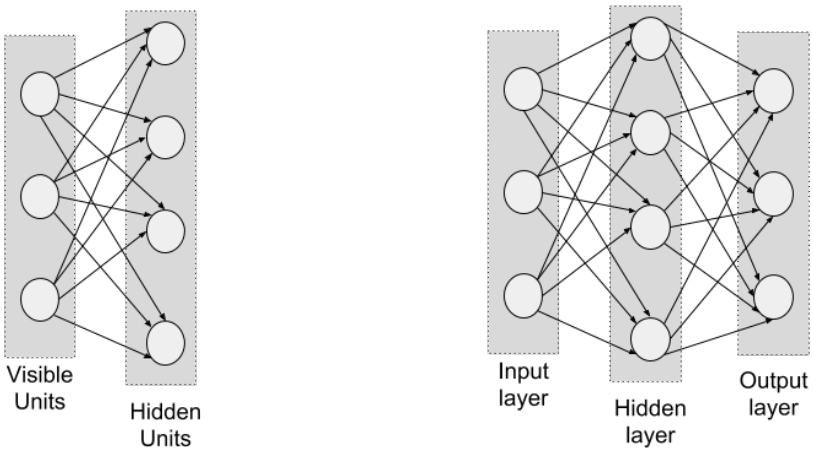
To solve this issue in an intelligent way, one can make use of unsupervised deep learning. Learning to find patterns in these unlabelled datasets is called *unsupervised deep learning*. The goal for unsupervised deep learning is to model the underlying structure or distribution in the data in order to learn more about the data. These are called unsupervised deep learning

because unlike supervised deep learning above there is no ground-truth or output known. These algorithms are left on their own to discover and present the interesting structure in the data.

Unsupervised deep learning models cover Deep Belief Networks (DBNs) and variants of stacked Auto Encoders (AEs) (i.e. stacked sparse AEs (SAEs) stacked denoising AEs (DAEs), and stacked contract AEs (CAEs)). Moreover, the building block of DBNs is a Restricted Boltzmann machine (RBM), and that of stacked AEs is an Autoencoder (AE). Although the first breakthrough result is related to DBNs, similar gains can also be obtained later by Autoencoders [67]. In this section, we discuss in detail about Autoencoders and their structure.

Autoencoders

Autoencoders are a family of neural nets that are well suited for unsupervised learning, a method for detecting inherent patterns in a data set. These nets can also be used to label the resulting patterns.



(a) RBM architecture (b) Simple 2 layer Autoencoder architecture

Figure 2.7: Autoencoder architectures

Essentially, autoencoders reconstruct a data set and, in the process, figure out its inherent structure and extract its important features. They work by compressing the input into a latent-space representation, and then reconstructing the output from this representation. They are typically shallow nets, the most common of which have one input layer, one hidden layer, and one output layer. Some nets, like the RBM, have only two layers instead of three. Input signals are encoded along the path to the hidden layer, and these same signals are decoded along the path to the output layer. Like the RBM, the autoencoder can be thought of as a 2-way

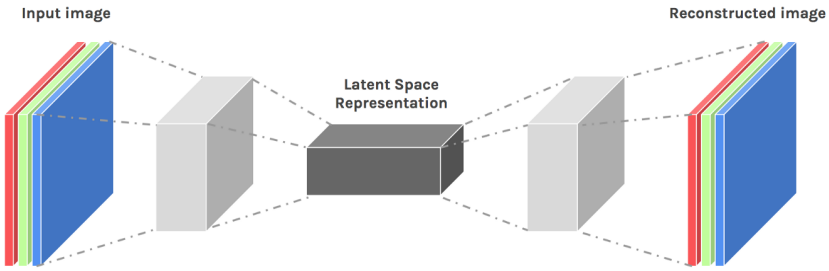


Figure 2.8: Convolutional Autoencoder architecture: It maps a wide and thin input space to narrow and thick latent space [20]

translator. AEs are trained with backpropgation and a concept using loss. Loss measures the amount of information about the input that was lost through the encoding-decoding process. The lower the loss value, the stronger the net.

A really popular use for autoencoders is to apply them to images. The trick is to replace fully connected layers by convolutional layers. These, along with pooling layers, convert the input from wide and thin (*lets say* 256×256 *px with* 3 *channels* - RGB) to narrow and thick. This helps the network extract visual features from the images, and therefore obtain a much more accurate latent space representation. The reconstruction process uses upsampling and convolutions. The resulting network is called a *Convolutional Autoencoder* [20]. These autoencoders have a very deep structure, with an equal number of layers for both encoding and decoding. A key application for deep autoencoders is dimensionality reduction. For example, these nets can transform a 256 × 256 pixel image into a representation with only 30 numbers. The image can then be reconstructed with the appropriate weights and bias; as an addition, some nets also add random noise at this stage in order to enhance the robustness of the discovered patterns. The reconstructed image wouldn't be perfect, but the result would be a decent approximation depending on the strength of the net. The purpose of this compression is to the reduce the input size on a set of data before feeding it to a deep classifier. Smaller inputs lead to large computational speedups, so this preprocessing step is worth the effort.

Convolutional Autoencoders are used in many applications like image reconstruction, image colorization, semantic segmentation, image denoising etc. In this thesis, we particulary use convolutional autoencoders for depth map and elevation map prediction for a given input optical and sonar image respectively.

Chapter 3

Literature Review

In this chapter, we review different methodologies for underwater 3D reconstruction and motion estimation for Autonomous Underwater Vehicles (AUVs). Underwater 3D reconstruction and motion estimation is discussed in the following perspectives: 1) *using 2D optical images*, 2) *using 2D acoustic images* and 3) *using both 2D optical and acoustic images*.

These methodologies are rather based on imaging geometry, traditional feature extraction techniques and correspondence problems as they are the core problems in 3D reconstruction and motion estimation alike. Currently no work has been done on deep learning based 3D reconstruction and ego motion for AUVs in underwater environments. And as this thesis is the first attempt to do so, we deliberate about state of the art deep learning techniques for general 3D reconstruction and motion estimation using optical sensor for air images.

3.1 Underwater 3D Reconstruction

In this section we discuss three categories to perform underwater 3D reconstruction using 2D images obtained from optical and/or acoustic cameras.

3.1.1 Optical Sensor based Methods

Computer vision techniques commonly applied for 3D reconstruction in land have found their way to underwater environments. Many remarkable attempts have been attained to perform 3D reconstruction from optical images acquired from moving AUV or Remotely Operated Vehicle (ROV).

Sedlazeck *et al.* in [64] reconstructs a real 3D scenario from the ROV Kiel 6000 using a HD color camera. Features are selected based on image gradients using a corner detector. Later,

the outliers are filtered using the RANSAC [25] procedure after the features have been matched.

In [61], Pizarro *et al.* perform optical surveys using the SeaBED AUV equipped with a 1280×1024 px CCD camera. Modified Harris corner detector and a generalized color moment are used as its feature detector and descriptor respectively.

Meline *et al.* in [50] compare Harris and SIFT features using a 1280×720 px camera in shallow water. In the paper, the authors reconstruct a statue bust. They infer that SIFT is not robust to speckle noise unlike Harris/correlation method. Moreover, Harris exhibited a better inlier count in the different scenarios.

Andono *et al.* in [4] use a Contrast Limited Adaptive Histogram Equalization(CLAHE) enhancement technique for 3D surface reconstruction of underwater coral reef. The SIFT (Scale Invariant Feature Transform) image matching algorithm is used to automatically extract the corresponding points between the image pairs. They demonstrate that the combination of CLAHE and outlier removal provides a better enhancement technique to improve the quality of the image.

In [49], McKinnon *et al.* use GPU SURF features and a high resolution camera of 2272×1704 px to reconstruct a piece of coral. This setup presents a few difficulties in terms of occlusions of the different views. With their structure from motion (SfM) approach, they achieve 0.7 mm accuracy at 1 to 1.5 m.

Sedlazeck and Koch in [38] develop a novel refractive SfM algorithm that takes into account the refraction of glass ports in water. By considering the refraction coefficient between the air-glass-water interface, their proposed refractive SfM enhances the results of generic SfM.

Nicosevici *et al.* [59] use SIFT features in a robotics approach, with an average error of 11 mm.

Beall *et al.* [11] use a wide baseline stereo rig to extract SURF features from left and right image pairs. After a smoothing and mapping (SAM) step, they recover structure of the environment by tracking these features. Then, the 3D points are triangulated using Delaunay triangulation, and the image texture is mapped to the mesh. This setup is applied to reconstruct coral reefs in the Bahamas.

Negre *et al.* [13, 16] use a SLAM approach in a micro AUV equipped with two stereo rigs to perform 3D reconstruction of underwater environments. In fig. 3.1, a 3D reconstruction of Santa Ponça Bay is displayed, covering an area of 25×10 m.

There are critical issues in performing 3D reconstruction based totally on underwater optical images, especially the tremendous light attenuation and backscatter. As a matter of fact, locating correspondences between the stereo perspectives are required to accomplish 3D reconstruction. Factors like challenging environmental conditions, low visibility and low lighting conditions make the task of finding correspondences between two underwater optical views more challenging. The solutions presented are summarized in Table 3.1

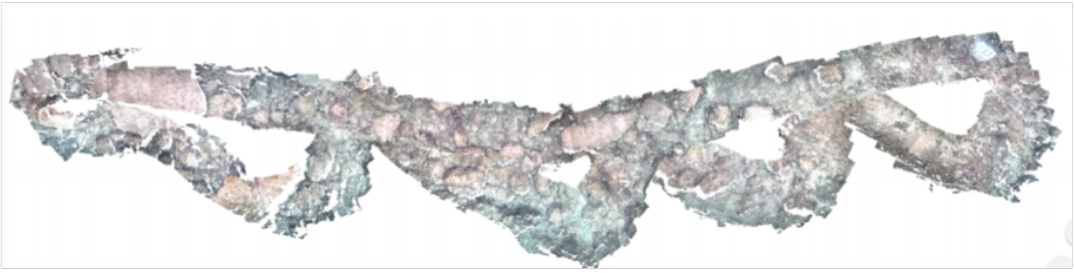


Figure 3.1: 3D reconstruction from SV using graph SLAM (25 × 10 m, Mallorca) [16]

References	Feature	Matching Method	Accuracy
Sedlazeck [64]	Corner	KLT Tracker	-
Pizarro [61]	Harris	Affine invariant region	3.6cm
Meline [50]	Harris	SIFT	-
Andono [4]	SIFT	SIFT	-
McKinnon [49]	SURF	SURF	0.7mm
Jordt-Sedlazeck [38]	-	KLT Tracker	-
Beall [11]	SURF	SURF & SAM, baseline - 60cm	-

Table 3.1: Summary of optical 3D reconstruction solutions

3.1.2 Acoustic Sensor based Methods

In recent times, a significant number of surveying tasks in underwater scenarios are carried out in turbid waters. And knowing the limitations of optical devices, sonar technology have proved to be quite effective for underwater operations. Sonar sensors are quite reliable for obstacle avoidance, navigation, localization and mapping [23, 42, 43, 62, 63].

There are many different types of sonar e.g. 3D sonar systems based on the signal processing methodology like *beamforming systems*, *holographic systems* and *acoustic lens based systems*, T-configured 3D sonars, lens concentrator systems that use mechanical methods to steer the acoustic beam, interferometric sonar, echoscope, eclipse, profiling sonar, multibeam echosounders, scanning imaging sonar and side-scan sonar. Even though 3D sonar system are quite capable of directly measuring three dimensional underwater environment scenes they are not cost effective and may also increase the payload of AUV. Whereas only a snapshot image is produced of a moving contact for other sonar systems due to low frequency rate. 3D reconstruction methods based on the above mentioned sonar systems are out of scope of this thesis.

In this thesis, we mainly focus on the new generation sonar namely the two-dimensional forward-looking sonars (2D FLS) and 3D reconstructions methods based on these sonar systems.

They are quite effective in the environments with reduced visibility and capable enough of producing high quality acoustic images at a near-video frame rate [1, 2].

Negahdaripour *et al.* in [52] utilize radiometric information in a sonar image to detect object and shadow regions, and various geometric cues and constraints to determine sizes, shapes and spatial distributions of 3-D objects resting on the sea floor. Furthermore, in [53], an SfM approach from a set of images taken from an imaging sonar is used to recover 3D data.

In [14], Brahim *et al.* implement the problem of 3D reconstruction from a sequence of sonar images using Covariance Matrix Adaptation Evolution Strategy (CMA - ES) optimization method based on punctual information i.e. 2D pixel coordinates are considered as 3D circle projection from the scene.

Assalih *et al.* in [6], propose the concept of Acoustic Stereo Imaging (ASI) system for underwater 3D reconstruction based on forward looking sonars. The proposed system features cheaper implementation than planar array sonars.

Aykin *et al.* in [7] estimate the lost elevation angles by first determining the 3-D information at object boundaries based on cast shadow cues and then employ the proposed image formation model to obtain the remaining elevation values through each object (fig. 3.2). Its main requirement is that the shadow is distinguishable and that it lays on a known flat surface.

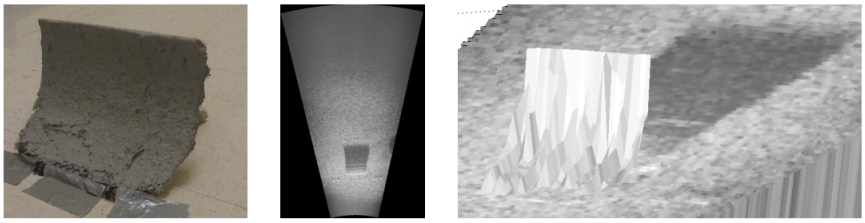


Figure 3.2: Reconstruction of concave hemi-cylinder - Aykin *et al.* [7]

While Guerneve *et al.* in [33] demonstrate near real time accurate 3D mapping using off the shelf BlueView imaging sonar system (fig. 3.3).

In [9], Aykin and Negahdaripour propose a novel method of space carving framework for the 3-D reconstruction of targets from multiple forward scan sonar images captured at known sonar poses. They demonstrate accurate reconstruction from the views obtained by several roll angles, ω_y, at each sonar position, providing way more information than images captured while circumventing the target.

The solutions presented are summarized in Table 3.2 Although the above methods are able to reconstruct 3D objects, there are some limitations in the existing sonar based methods. Firstly, these methods manually select the features and to which matching is done only for those sparse

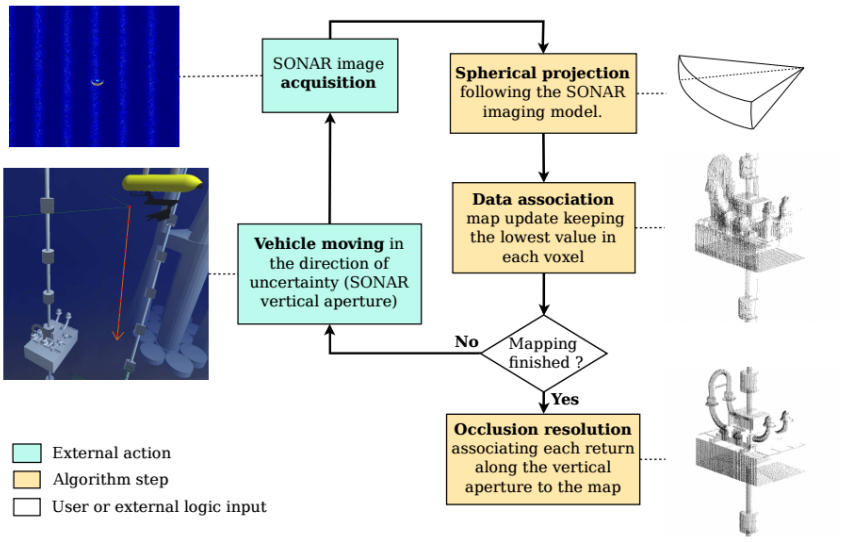


Figure 3.3: 3D reconstruction pipeline - Guerneve *et al.* [33]

References	Sonar Type	Scope	Accuracy
Negahdaripour [52]	FLS	Sparse scene geometry	-
Brahmin [14]	FLS	Sparse scene geometry	0.5m
Assalih [6]	FLS + FLS	Acoustic stereo system	-
Aykin [7]	FLS	Smooth surfaces 3D reconstruction of target object	≈ 15cm
Guerneve [33]	FLS	Sparse scene geometry	-
Aykin [9]	mulitple views of FLS	Sparse reconstruction of target object	-

Table 3.2: Summary of sonar 3D reconstruction solutions

features. And, it makes use of very small number of sparse features for 3D reconstruction i.e no dense reconstruction. Also, the 3D reconstruction is done only for a particular object of interest in the scene and not the complete underwater scenario.

Acoustic camera like forward looking sonar have proved to be quite effective as compared to optical cameras in turbid and murky underwater environments. Although sonar images are noisy but with the potential of solving the problem of 3D reconstruction in the underwater scenario, we can say that 2D forward looking sonar contains more information than the eye can extract.

3.1.3 Opti-Acoustic Sensor based Methods

As discussed in the previous two sections, we see that optical cameras provide high-detailed images of target surfaces under low turbidity water conditions and its visibility range can also be extended in naturally illuminated water thanks to light polarization in scattering media, where as acoustic images obtained from FLS are quite noisy due to high acoustic clutter. The integration of optical and acoustic sensors may enhance the performance of imagery in comparison to the processing of the images from each sensor alone. There are some methods tackling the problem of 3D reconstruction using opti-acoustic cameras to enhance the 3D output and to better correlate the sonar correspondences.

In [54], Negadaripour derive the epipolar geometry for opti-acoustic system. The author computes the missing Z coordinate of a scene in virtual world and ended up with thee solutions the range solution, the azimuth solution and the third one is derived by expressing point coordinates in the sonar reference frame using the rigid transformation between the sonar reference frame and the camera reference frame.

Furthermore, in [57] Negahdaripour *et al.* uses a stereo system formed by a camera and an imaging sonar. Correspondences between the two images are described in terms of conic sections.

Babaee *et al.* in [10] use a FLS and a optical camera, and feature correspondences between the two images are provided manually to perform reconstructions.

References	Sonar Type	Scope	Accuracy
Negahdaripour [54, 57]	FLS + Optical camera	Alternative to stereo system	$\approx$ 5cm
Babaee [10]	FLS + Optical camera	Alternative to stereo system	-

Table 3.3: Summary of opti-acoustic 3D reconstruction solutions

The opt-acoustic configuration is promising though but it still has some cons. Firstly, the concept of epipolar geometry for opti-acoustic system is theoretical in nature meaning it works very well only for perfect conditions in terms of illumination and visibility. Also, there is very little intersection area between field of view (FOV) of optical camera and FOV of acoustic camera hence preventing it from applications like surveying, monitoring behaviour of the fishes etc.

3.2 Underwater Motion Estimation

Motion estimation for AUVs is a wide topic and can be viewed from different perspectives. Earlier methods estimate motion of AUV using inertial sensors (e.g. gyroscope and accelerometer) combined with Doppler Velocity Log (DVL), Global Positioning System (GPS) combined with

Inertial Navigation System (INS), Acoustic Doppler Current Profiler (ADCP) integrated with INS, acoustic beacons (e.g. floating acoustic buoys, Long Base-Line (LBL) system, Short Base-Line (SBL) system, and Ultra Short Base-Line (USBL) system) and geophysical navigation systems like Terrain Based Navigation (TBN) system. Filtering methods like Particle Filter, Kalman Filter and Extended Kalman Filter estimate AUV positions based on measurements from the aforementioned sensors. In this section, we particularly focus on approaches based solely on optical and/or acoustic (obtained from FLS) images for estimating the motion apart from previously discussed SfM approaches.

3.2.1 Motion Estimation using Optical Sensor

Xu *et al.* in [72] propose a direct method for accurate 3-D motion estimation from the sea bed optical images. They make use of this information to perform video mosaicing.

In [27], Garcia *et al.* present a mosaicing algorithm which includes: *feature selection and matching*, *detection of points describing the dominant motion, homography computation* and *mosaic construction*; based on this it is able to provide positional coordinates to an underwater vehicle (fig. 3.4).

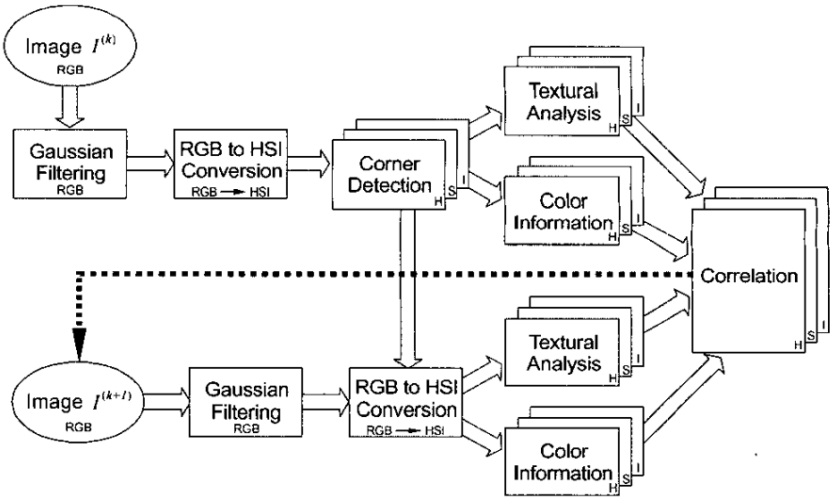


Figure 3.4: Scheme of the algorithm - Garcia *et al.* [27]

Futhermore, in [28] propose a new method to improve image matching in underwater image sequences for estimating the motion of an underwater robot. However, it has serious limitations when all the correspondence matches are coplanar.

Carreras *et al.* in [17] present a vision-based localization system for an underwater robot in

a structured environment. They conclude that it can be used to evaluate the performance of video mosaicing system, designed to work in unstructured environment.

Based on the above methods, it can be seen that image registration is a pre-requisite to accomplish both 3D reconstruction and motion estimation. Motion estimation methods based on the concept of SfM are already discussed in section 3.1.1.

3.2.2 Motion Estimation using Acoustic Sensor

High frquency 2D FLS can provide images with target details whereas optical systems are ineffective due to restricted visibility and high levels of turbidity. As seen in the previous section, to estimate ego-motion of the optical camera or AUV, image registration is an imperative step. Based on similar lines motion estimation of FLS sonar sensor is carried out. In this section, we discuss some of the methodologies to do the same.

Negahdaripour *et al.* in [51] investigate the model of sonar image flow in 2D FLS by tracking the 2-D images of scene features to enable automated dense correspondences and 3D motion estimation. Furthermore in [55], the author present the mathematical models of sonar image flow for 3-D objects and their cast shadows and utilize them in devising various 3-D sonar motion estimation solution.

In [8], Aykin *et al.* present a novel solution for FLS sonar registration by employing a more accurate motion transformation representations. They formulate the problem of image registration as an optimization over six degrees of freedom in 3D sonar motion by incorporating the detected key features and landmarks. This method improved on the state of the art in many important ways.

3.2.3 Motion Estimation using Opti-Acoustic Sensor

Negahdaripour *et al.* in [56] present integration of visual motion cues in the opti-acoustic stereo imaging system to estimate 3D motion using only the correspondence between matches in sonar and optical sequences, individually.

Moreover in [58], Negahdaripour *et al.* investigate the 3D motion estimation by making full use of the visual cues in optical and sonar video for vision based navigation and 3D positioning of platforms. They improve the motion estimation accuracy by tracking the features in the sonar and optical motion sequences independently, hence no multi-modal association required between corresponding optical and sonar features.

3.3 Deep Learning Methods

Deep Learning (DL) has recently been dominating many computer vision tasks such as object detection and recognition, and has been applied to robotic applications, e.g. robotic manipulation and grasping, with promising or even super human performance. With the recent development of deep neural networks for image classification, current state-of-the-art performance is achieved by CNNs. CNNs have proved to yield better results than traditional techniques which use hand-crafted features for detection. They have proved to be more effective in matching images or identifying objects in recent years and may one day make traditional feature detection techniques obsolete. However, no work has been done on deep learning based 3D reconstruction and motion estimation using optical and/or sonar images for underwater scenario yet. Nevertheless, we discuss some state of the art DL techniques for motion estimation and 3D reconstruction using optical images in land.

3.3.1 Deep Learning for Depth Estimation

In this section, we discuss some latest state of the art deep learning methods for depth estimation using optical images in land. We discuss both supervised and unsupervised deep learning techniques for depth map estimation yielding 3D scene reconstruction.

Supervised Methods

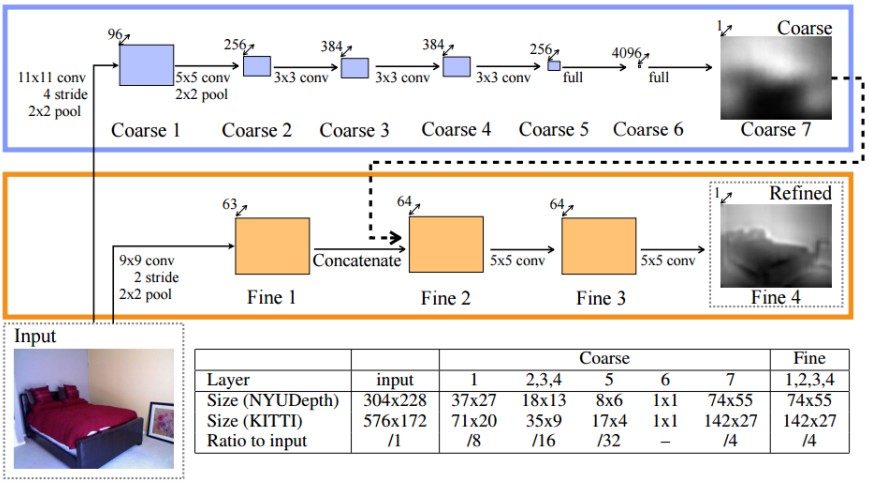
In [22], Eigen *et al.* present depth map estimation framework using two deep networks, one that estimates the global depth structure, and another that refines it locally at finer resolution. It achieved a state-of-the-art on this task for NYU Depth and KITTI dataset (fig. 3.5).

Tateno *et al.* in [68], propose a method where CNN-predicted dense depth maps are naturally fused together with depth measurements obtained from direct monocular SLAM and present a framework to efficiently fuse semantic label yielding coherent scene reconstruction (fig. 3.6).

Unsupervised Methods

Garg *et al.* in [29], present first unsupervised framework to learn a deep CNN for single view depth prediction based on encoder-decoder design, without requiring annotated ground-truth depths. They train the network using raw stereo pair of images and once the model is trained, single view depth map is estimated during test-time inference giving comparable result as the supervised learning approaches (fig. 3.7).

In [31], Gordard *et al.*, propose a novel loss function for single view depth estimation enforcing consistency between the predicted depth maps from each camera view during training and subsequently improving predictions (fig. 3.8).



		Coarse					Fine
Layer	input	1	2,3,4	5	6	7	1,2,3,4
Size (NYUDepth)	304x228	37x27	18x13	8x6	1x1	74x55	74x55
Size (KITTI)	576x172	71x20	35x9	17x4	1x1	142x27	142x27
Ratio to input	/1	/8	/16	/32	–	/4	/4

Figure 3.5: Model Architecture - Eigen *et al.* [22]

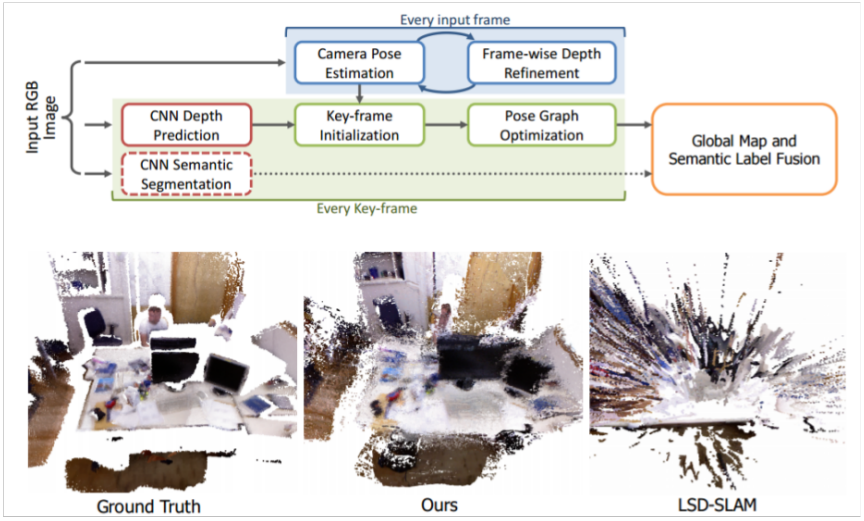


Figure 3.6: CNN-SLAM Overview - Tateno *et al.* [68]

Recently, Zhou *et al.* in [73], propose a novel SfMLearner architecture for the task of monocular depth and camera motion estimation from unstructured video sequences based on the task of view synthesis as supevisory signals. Despite achieving good performance on benchmark dataset, still a number of major challenges like occlusion, scene dynamics to name a few needs to be addressed. This setup is most aligned with our work as we similarly learn depth/elevation

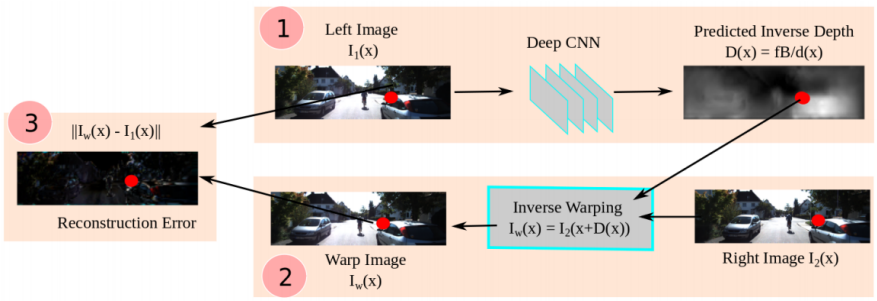


Figure 3.7: Overview pipeline of deep CNN for single view depth - Garg *et al.* [29]

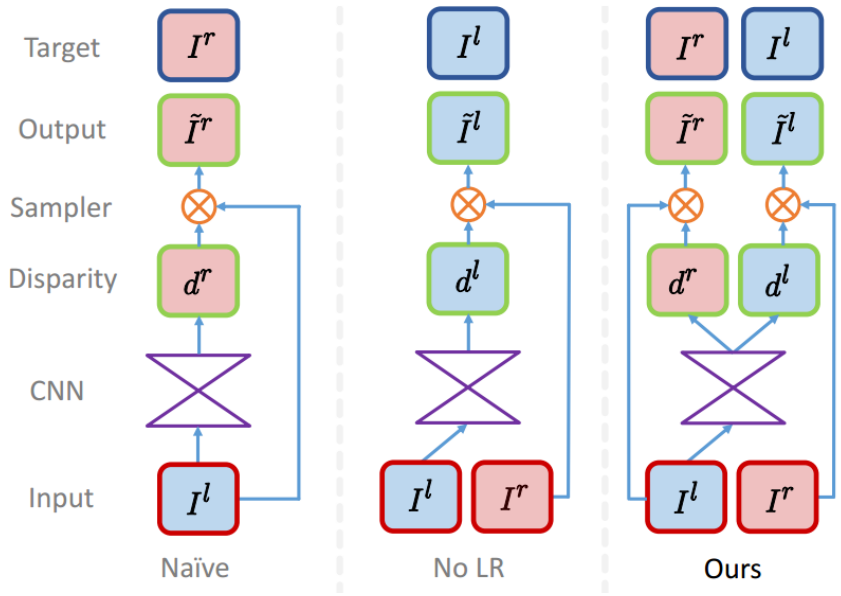


Figure 3.8: Sampling strategies for backward mapping. This approach uses the left image to produce disparities for both images, improving quality by enforcing mutual consistency - Godard *et al.* [31].

and ego-motion from optical/sonar image in an unsupervised setting for underwater scenarios (fig. 3.9).

3.3.2 Deep Learning for Pose Estimation

Here, we discuss state of the art CNN based deep learning techniques used for pose estimation.

Kendall *et al.* in [40], present a 23 layer deep CNN, also known as PoseNet to regress

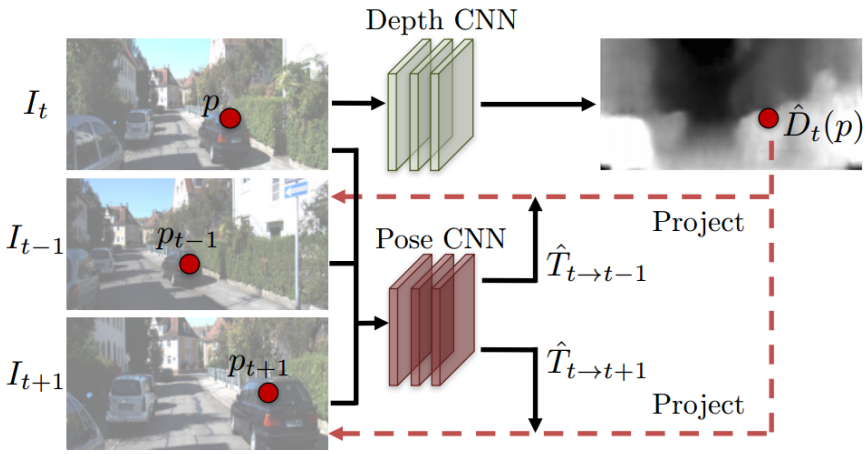


Figure 3.9: Overview pipeline of SfMLearner architecture - Zhou *et al.* [73]

the 6-DOF camera pose from a single RGB image in an end-to-end manner. The architecture proposed by the authors localizes from high level features and is robust to motion blur, difficult lighting conditions etc. where point based SIFT registration fails (fig. 3.10).

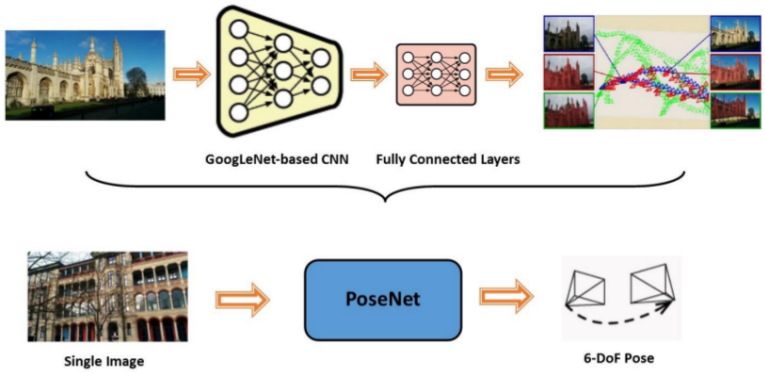


Figure 3.10: Overview pipeline of PoseNet - Kendall *et al.* [40]

In [45], Li *et al.* present an indoor relocalization using color image and depth images as input in a dual stream deep CNN. The relocalization accuracy improved significantly over PoseNet by making use of depth images in a separate stream thus enabling it to learn texture features from color images and structural range features from depth images. However, the localization accuracy produced by the model is not as good as geometric method in normal scenes (fig 3.11).

Recently, Li *et al.* in [46] propose an UnDeepVO architecture which is able to estimate the 6-DoF pose of a monocular camera and the depth of its view by using deep neural networks

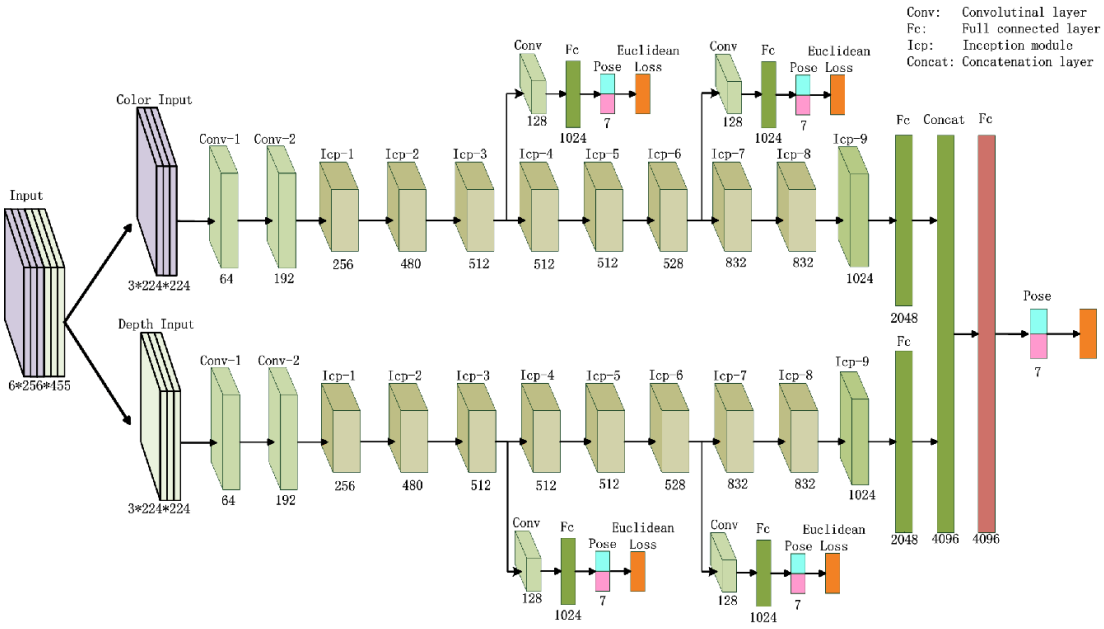


Figure 3.11: Architecture overview of dual-stream CNN for indoor relocalization - Li *et al.* [45]

in an unsupervised fashion. They incorporate spatial and temporal dense information in the loss function for training the network. However, a number of challenges such as robustness to image blurs, camera parameters, or illumination are need to be addressed (fig. 3.12).

There are also methods using Recurrent Neural netowrks (RNN) and Recurrent Convolutional Neural networks (RCNNs) for pose estimation based on spatial temporal information of monocular video stream [19, 21, 69]. However, the scope of this thesis is only limited to CNN based deep learning techniques.

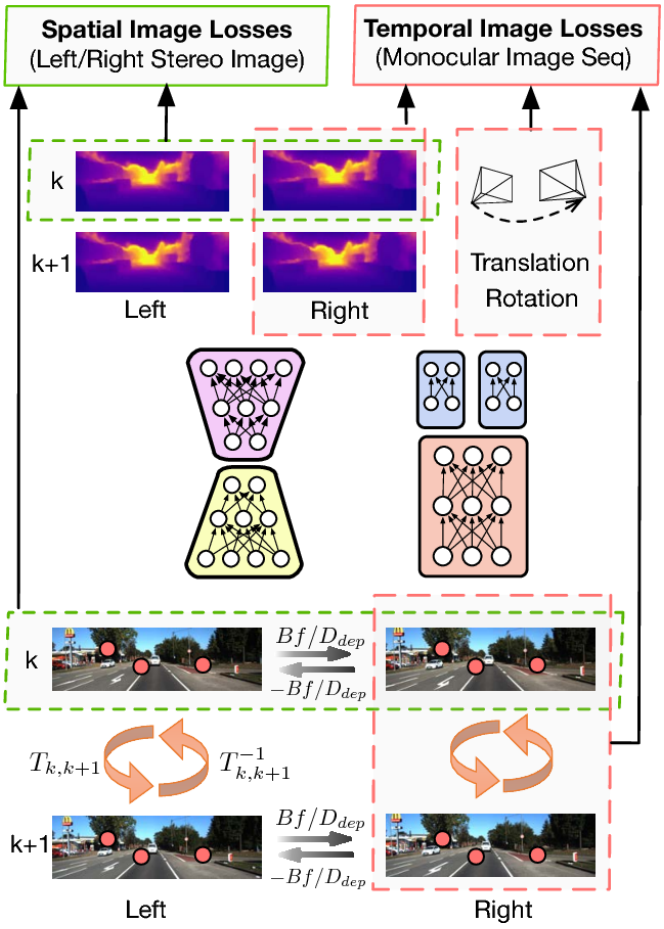


Figure 3.12: Training scheme of UnDeepVO - *Li et al.* [46]

Chapter 4

Methodology

In the previous chapter, we presented state of the art approaches to perform 3D reconstruction and motion estimation. Also, we discussed advantages and limitations of those methods. In this chapter, we present our unsupervised learning framework for the task of monocular depth (for optical camera) / elevation map (for FLS) and camera ego-motion for unstructured video sequences (optical or acoustic images) of underwater scenarios.

4.1 Proposal Description

CNNs are state of the art deep learning techniques and have obtained very good results in different tasks that use optical color images. They provide an ideal theoretical framework to build an end-to-end system. According to the literature review, we see that there is no existing work available on motion estimation and 3D reconstruction for underwater environments as a complete end-to-end system. Also the use of CNNs in analysis of sonar images for elevation map and pose estimation is not yet explored.

In this work, we propose two *unsupervised learning framework* based on CNNs for underwater scenes :

1. depth map and pose prediction network using unstructured sequences of optical images.
2. elevation map and pose prediction network using unstructured sequences of sonar images in polar form.

Since there is no depth map / elevation map or pose ground truth for publicly available datasets, the only option to make this completely end-to-end is by using unsupervised learning. Also, unsupervised learning removes the need of such supervisory signals. In the following sections we discuss each of these frameworks in detail.

4.2 Depth Map and Ego-Motion from Optical Images

Classical solutions to depth map and ego motion computation involve stereo and feature matching techniques, whereas recent methods have shown some success using deep learning. Inspired by the recent deep learning techniques for ego-motion and depth map estimation using monocular optical images in land (KITTI, CityScapes dataset), on similar lines we propose an unsupervised learning framework using optical video sequences of underwater environments for pose and depth estimation. This setup is most aligned with [73] as we similarly learn depth and ego-motion from monocular video in an unsupervised setting. In this section, we discuss the geometry of the problem, network architecture and then describe the individual loss functions used in the framework.

4.2.1 Problem Geometry

Th goal of this unsupervised learning framework is to learn depth and pose from a single monocular video stream for underwater environment. The problem is formalized as follows:

1. Given an input image I_t at time t, we estimate depth D_t and with the help of nearby source views i.e images I_{t-1} and I_{t+1} at time $t-1$ and $t+1$ respectively, ego-motions $T_{t\to t-1}$ and $T_{t\to t+1}$ are estimated. The depth and ego-motion estimations are done using depth and pose network which is described in the next section.

2. Using the pixel coordinates (i, j) of image I_t and estimated depth D_t^{ij}, it is then projected into a structured 3D point cloud Q_t.

$$Q_t^{ij} = D_t^{ij} K^{-1}[i, j, 1]^T \tag{4.1}$$

   where K is intrinsic matrix of the camera (using homogeneous coordinates)

3. Given the estimates for $T_{t\to t-1}$, the camera movement from t to $t-1$ and $T_{t\to t+1}$, the camera movement from t to $t+1$, Q_t can be transformed to get the estimates for $\hat{Q}_{t-1}$ 3D point cloud at $t-1$ and $\hat{Q}_{t+1}$ 3D point cloud at $t+1$.

$$\hat{Q}_{t-1} = T_{t\to t-1} Q_t \tag{4.2}$$

$$\hat{Q}_{t+1} = T_{t\to t+1} Q_t \tag{4.3}$$

4. $\hat{Q}_{t-1}$ and $\hat{Q}_{t+1}$ can then be projected onto the camera frame at $t-1$ and $t+1$ respectively as $K\hat{Q}_t - 1$ and $K\hat{Q}_t + 1$.

5. Combining this transformation and projection, we obtain I_t image's projected pixel coordinates onto the source views $\hat{I}_{t-1}$ and $\hat{I}_{t+1}$ at time $t-1$ and $t+1$ respectively.

$$\hat{p}_{t-1} = KT_{t\to t-1}D_t(p_t)K^{-1}p_t \tag{4.4}$$

$$\hat{p}_{t+1} = KT_{t\to t+1}D_t(p_t)K^{-1}p_t \tag{4.5}$$

where,
p_t is pixel coordinate of input image at time t
$\hat{p}_{t-1}$ is pixel coordinates of $\hat{I}_{t-1}$
$\hat{p}_{t+1}$ is pixel coordinates of $\hat{I}_{t+1}$
$D_t(p_t)$ is depth value at p_t pixel coordinate

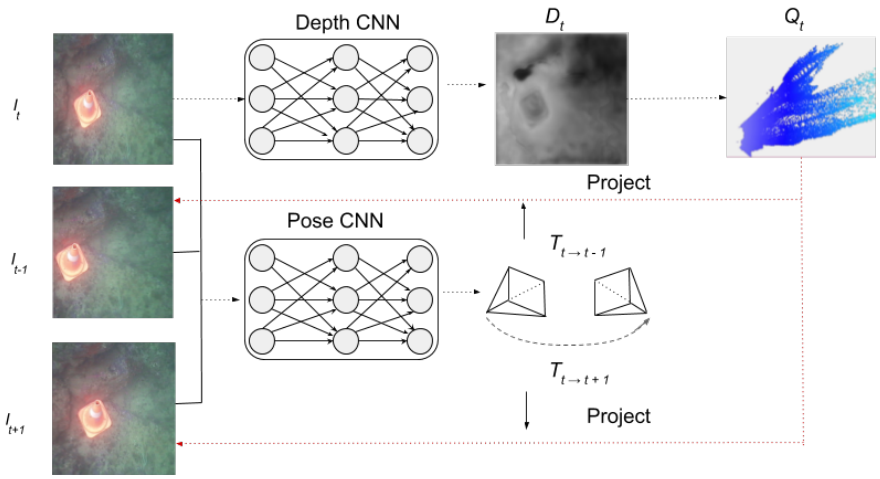


Figure 4.1: Overview of the supervision pipeline based on view synthesis. The depth network takes only the target view as input, and outputs a per-pixel depth map D_t. The pose network takes both the target view (I_t) and the nearby/source views (e.g., I_{t-1} and I_{t+1}) as input, and outputs the relative camera poses ($T_{t\to t-1}, T_{t\to t+1}$). The outputs of both networks are then used to inverse warp the source views using Q_t to reconstruct the target view, and the image reconstruction loss is used for training the CNNs. By utilizing view synthesis as supervision, we are able to train the entire framework in an unsupervised manner from videos.

The above formulation allows us to reconstruct the image frame $\hat{I}_{t-1}$ and $\hat{I}_{t+1}$ by warping input image I_t based on D_t and ego-motions $T_{t\to t-1}$ and $T_{t\to t+1}$. As the projected pixel coordinates are continuous values we follow the procedure given in [73] wherein a soft sampling is performed from the four pixels in I_t whose pixel coordinates overlap with $\hat{p}_{t-1}$ to get $\hat{I}_{t-1}$ or $\hat{p}_{t+1}$ to get $\hat{I}_{t+1}$.

4.2.2 Network Architecture

In this section, we briefly discuss the depth and pose network architecture. Note that the CNN network proposed for underwater optical sequences can be jointly trained. Despite being trained together it can be used independently during test-time inference.

Single View Depth Network

We adopt the SfMLearner architecture [73] which is in turn based on DispNet [48]. It is mainly an encoder-decoder or autoencoders design with skip connections and multiscale predictions. Given a single image as input it produces a dense depth estimate mapping each pixel of the input to a metric depth value. The network is fully convolutional. All *conv* layers are followed by a ReLU actiavtion function except for the prediction layers where sigmoid function is used which is scaled by a factor of 10 and an offset of 0.01 is added for positive depth value constraints.

Pose Network

The input to the pose estimation network is a sequence of three images (e.g. at time $t-1$, t and $t+1$). We concatenate the target image (i.e image I_t) with nearby source images (i.e I_{t-1} and I_{t+1}) along the color channels. The outputs of the pose network are the relative poses between the target image and each of the source views (i.e $T_{t\rightarrow t-1}$ and $T_{t\rightarrow t+1}$). These ego motion predictions are represented by six numbers corresponding to relative 3D rotation (Euler angles) and metric translations between the two frames. The network is again fully convolutional. All *conv* layers are followed by a ReLU activation function except for the last layer where no nonlinear activation is applied.

4.2.3 Loss Functions

In this section, we describe the loss functions to drive our proposed framework for continuous predictions.

Image Reconstruction loss

We use novel view synthesis supervision to form the image reconstruction. We described in section 4.2.1, that given an input image it can be warped to its nearby source views. Let $< I_1, I_2, ...I_N >$ be training image sequence with one of the frames being target image and the rest being the source view $I_s (1 \leq s \leq N, s \neq t)$. The image reconstruction loss is formulated as:

$$L_{rec} = \sum_{ij} ||I_t^{ij} - \hat{I}_s^{ij}|| \tag{4.6}$$

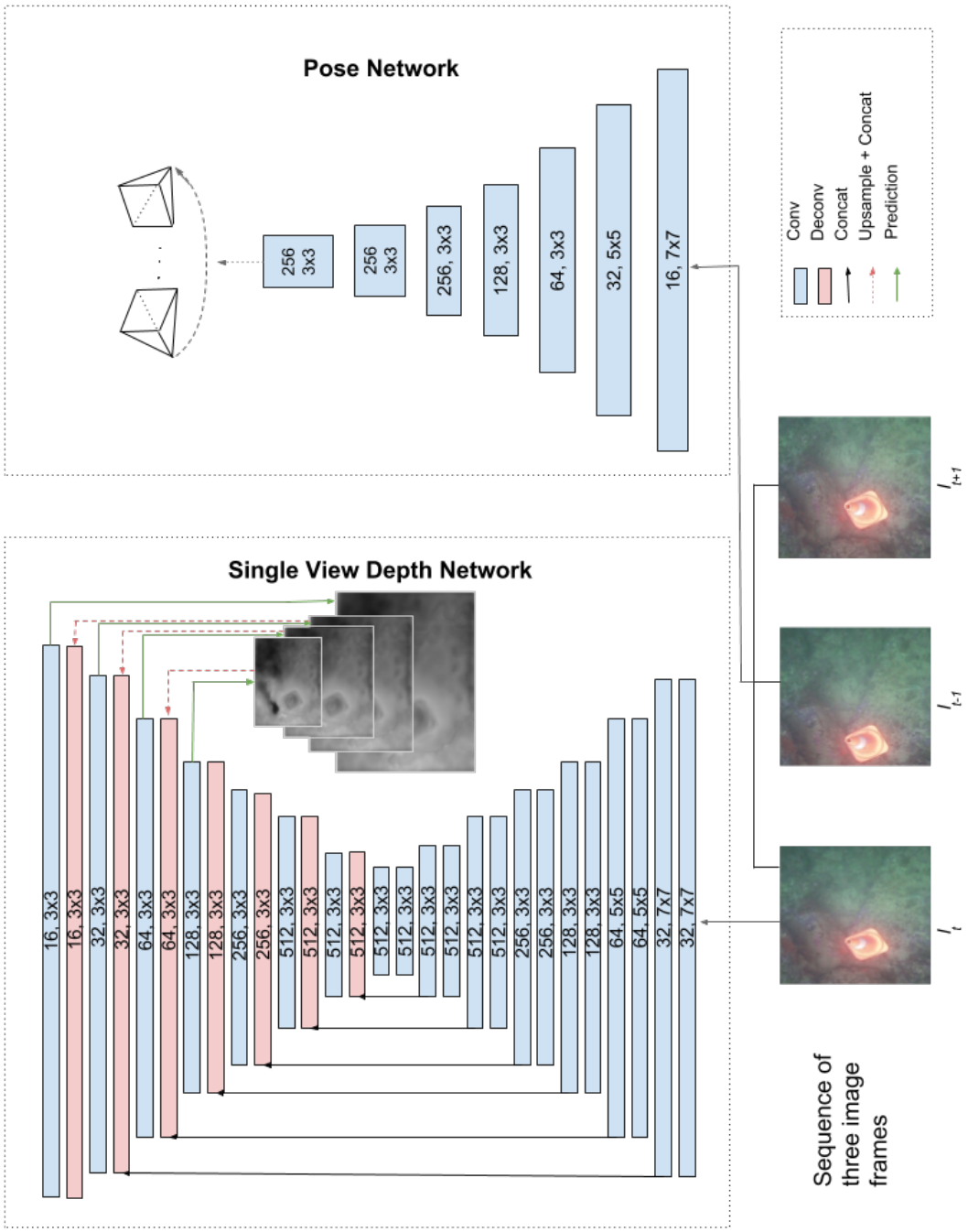


Figure 4.2: Network architecture for our depth/pose prediction modules. The width and height of each rectangular block indicates the output channels and the spatial dimension of the feature map at the corresponding layer respectively, and each reduction/increase in size indicates a change by the factor of 2. For single-view depth network, we adopt the SfMLearner architecture which is in turn based on DispNet [48] architecture with multi-scale side predictions. (b) The pose networks predict 6-DoF. relative pose.

where, $\hat{I}_s$ is the source view I_s warped to the target coordinate frame based on the predicted depth D_t, ego-motion $T_{t\to s}$ and the input source view I_s as discussed in the problem geometry.

The reconstructed image $\hat{I}_s$ is an approximate and a relatively crude image. It does not take into account for effects such as lighting, shadows or reflections. As a result, the loss becomes noisy and results in artifacts. To avoid this problem, we add a strong regularization in order to reduce this artifacts.

Structured Similarity loss

Structured Similarity is a metric to evaluate the quality of image predictions. It measures the similarity between two images patches x and y and is defined as:

$$SSIM(x,y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x + \sigma_y + c_2)} \tag{4.7}$$

where μ_x, σ_x are the local means and variances [70]. μ and σ are computed by simple pooling operation, $c_1 = 0.01^2$ and $c_2 = 0.03^2$. Since SSIM is upper bounded to 1 and needs to be maximized, we instead minimize:

$$L_{SSIM} = \sum_{ij}[1 - SSIM(\hat{I}_s^{ij}, I_t^{ij})] \tag{4.8}$$

Depth Smoothness Loss

A depth smoothness loss is also employed to regularize the depth estimates. Similarly to [31, 34] we use a depth gradient smoothness loss that takes into account the gradients of the corresponding input image. It incorporates sharp changes in depth at pixel coordinates where there are sharp changes in the image. It is a refinement of the depth smoothness loss defined in Zhou *et al.* [73].

$$L_{smooth} = \sum_{ij} ||\partial_x D^{ij}||e^{-||\partial_x I^{ij}||} + ||\partial_y D^{ij}||e^{-||\partial_y I^{ij}||} \tag{4.9}$$

4.2.4 Learning Setup

All the aforementioned loss functions are applied at all the scales s, ranging from the models input resolution, to an image that is $\frac{1}{8}$ in width and height. The total loss is defined as:

$$L_{total} = \sum_s \alpha L_{rec}^s + \beta L_{SSIM}^s + \gamma L_{smooth}^s \tag{4.10}$$

where α, β and γ are hyper-parameters.

4.3 Elevation Map and Ego-Motion from Sonar Images

As discussed in the chapter 2 regarding FLS sonar image formation process, we see that the elevation angle ϕ is lost through projection from the 3D space (R, Θ, Φ) onto the 2D (r, θ) or (x_s, y_s) sonar image domain. For accurate 3D scene reconstruction from 2D FLS sonar imagery, we need to compute the unknown elevation angle from the image data with high precision. However, an approximate estimation is more than enough for 3D sonar ego-motion estimation.

According to the literature review, we see that most of the 3D reconstruction and motion estimation methods revolve around the computation of elevation map for its corresponding sonar image data. But, these elevation maps are computed under some assumptions. In [52], Negahdaripour *et al.* show that an elevation map of the imaged plane can be determined from its surface normal. But planar seabed assumptions are taken to compute the surface normal which might fail in the real world. Also, the sonar settings and configuration relative to the scene, vertical sonar beam width, and measurements from auxiliary sensors (e.g., altimeters) are known for the computation of the normal vector for this background plane. Aykin *et al.* in [8], make use of object-shadow pairs extracted from detected blob regions to estimate the elevation angle of 3D features. But it assumes that the range increases monotonically over 3D objects and elevations values from the leading to trailing edge on each object are filled by linear interpolation.

Inspired by the recent success of CNNs able to estimate depth map and ego-motion using optical images we propose an analogous unsupervised learning framework known as "**Sonar-Net**" to estimate elevation map and ego-motion using sonar video stream. This is the first attempt ever to make use of deep learning for the estimation of elevation map and pose solely based on sonar image sequences in underwater scenarios. This setup is very much similar to the the setup discussed in previous section for optical images with certain changes. In this section, we discuss geometry of the problem, perform some proof of concept to support our hypothesis, brief overview of the network architecture and then describe the learning setup of the architecture.

4.3.1 Problem Geometry

The goal of this proposed framework or "SonarNet" is to learn elevation angles and pose from a single sonar video stream for underwater environment in an unsupervised fashion. The problem is formalized as follows:

1. Given an input sonar image S_t at time t, we estimate elevation map E_t and with the help of nearby source views i.e images S_{t-1} and S_{t+1} at time $t-1$ and $t+1$ respectively, ego-

motions $T_{t\to t-1}$ and $T_{t\to t+1}$ are estimated. The elevation map and ego-motion estimations are done using SonarNet architecture which is described in the next section. Note that the input sonar image is in polar form (r, θ) as that is the raw image obtained from the FLS sonar sensor.

2. Sonar being an acoustic sensor, the concept of intrinsic matrix is not valid. So the pixel coordinates (r, θ) are linearly scaled to actual (R, Θ) measurements depending on the window length and FOV of the sonar sensor, where R is in meters and Θ in degrees. See **appendix A** for details.

3. Elevation map is defined as for every (R, Θ) real world coordinate of an object in the scene, there is a corresponding elevation angle Φ (in degrees) for that object. The elevation angle range for mostly every sonar sensor is within -7 deg to $+7$ deg. The predicted elevation map (E_t) from the network have values in the scale ranging from 0 to 1 (i.e. linear mapping from [-7 deg - $+7$ deg] scale to range [0 - 1]) which is further linearly mapped to gray scale intensity range $0 - 255$ for visualization purpose.

$$E_t(r, \theta) = \phi \tag{4.11}$$

where (r, θ) are pixel coordinates of elevation map and ϕ is the intensity value of elevation map at that coordinate.

4. A structured 3D point cloud in spherical coordinate system is formed by linearly scaling the pixel coordinates (r, θ) and its corresponding intensity (ϕ) of elevation map E_t to get real world coordinates (R, Θ, Φ). The azimuth (Θ) and elevation (Φ) angles are converted to radians for further computations.

5. The structured 3D point cloud is converted to Cartesian coordinate system using equation 2.5 to get (X, Y, Z) real world coordinates, denoted by Q_t.

6. Given the estimates for $T_{t\to t-1}$, the sonar sensor movement from t to $t-1$ and $T_{t\to t+1}$, the sonar sensor movement from t to $t+1$, Q_t can be transformed to get the estimates for $\hat{Q}_{t-1}$ 3D point cloud at $t-1$ and $\hat{Q}_{t+1}$ 3D point cloud at $t+1$.

$$\hat{Q}_{t-1} = T_{t\to t-1} Q_t \tag{4.12}$$

$$\hat{Q}_{t+1} = T_{t\to t+1} Q_t \tag{4.13}$$

7. $\hat{Q}_{t-1}$ and $\hat{Q}_{t+1}$ can then be projected onto the sonar image frame at $t-1$ and $t+1$ respectively using equation 2.6 to get (R, Θ) and then by appropriate linear scaling to get (r, θ) pixel coordinates.

8. Combining this transformation and projection, we obtain S_t image's projected pixel coordinates onto the source views $\hat{S}_{t-1}$ and $\hat{S}_{t+1}$ at time $t-1$ and $t+1$ respectively.

The above formulation allows us to reconstruct the sonar image frame $\hat{S}_{t-1}$ and $\hat{S}_{t+1}$ by warping input sonar image S_t based on E_t and ego-motions $T_{t\to t-1}$ and $T_{t\to t+1}$. As the projected pixel coordinates are continuous values we follow the similar procedure of soft sampling described in the previous section 4.2.1 for optical images.

4.3.2 Proof of Concept

In the geometry of the problem for optical images, we have seen that given an input image of the scene, we can synthesize a new image of the scene at different camera pose based on estimated per-pixel depth map and ego-motion. In analogous to this, we carry out a proof of concept work using sonar image. The hypothesis is that given an input sonar image, we can synthesize a new sonar image of the scene at different FLS sonar pose based on elevation map and ego-motion. In this section, we demonstrate the synthesis of a new sonar image at new location by manually setting the relative pose using an example sonar image and its corresponding generated elevation map.

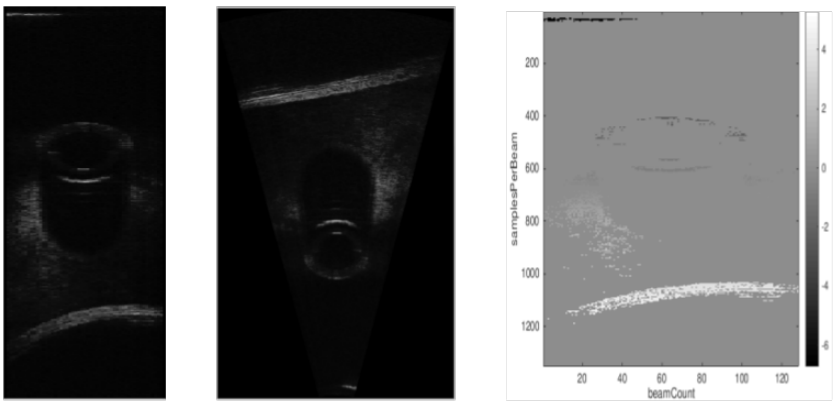


Figure 4.3: (a) Polar form of sonar image taken in a water tank containing an old Tyre (viewing direction is from the top). (b) Cartesian form of sonar image (viewing direction from the bottom). (c) Synthetically generated elevation map of sonar image

Consider the image in the above fig. 4.3 (a). This sonar image is collected using ARIS 3000 sonar system in a water tank containing an old Tyre. Here, we prove our hypothesis that given

elevation map and relative pose between two locations, we can synthesize a new sonar image at new location. For this, we synthetically generate an elevation map of the above sonar image as there is no actual elevation map data available for the given sonar image.

Elevation Map Generation

The elevation map is generated based on sonar image characteristics. We firstly threshold the noise intensities values in spatial domain of sonar image based on object (T_o) and (T_s) shadow threshold values. For this, intensities along a particular beam are analyzed as per the graph mentioned in fig. 4.4. This thresholding is done so that elevation values are not assigned to noisy features in the image. On this thresholded image, we incrementally assign values between -7 deg to $+7$ deg from minimum range of the sonar image to its maximum range i.e. from top to bottom of the image (polar form). The generated elevation map is shown in fig. 4.3 (c)

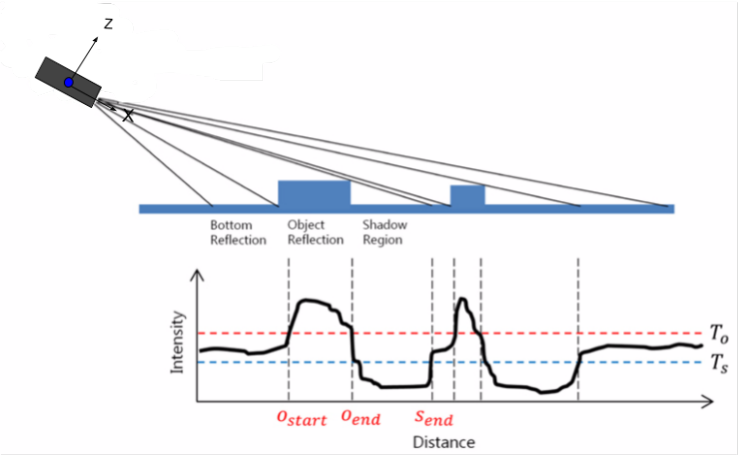


Figure 4.4: Object and shadow detections based on thresholds T_o and T_s, respectively [66]

Once the elevation map is generated, we follow the procedure described in the problem geometry section 4.3.1. Using the sonar image and the elevation map, we first create a 3D point cloud. Apply manual transformation operation on the point clouds to get the transformed point cloud and subsequently projecting these transformed point clouds to polar form of sonar image. Thereby, synthesizing a new sonar image at new location. In the following section we show some synthesized sonar image view examples by manually applying simple transformations. Note in these examples we show result in Cartesian form of sonar image as it is easy to interpret.

Examples

1. **Translation in $+x$ direction by** 0.5 **m** [fig. 4.5 (b)]. Note the grey patch in the synthesized image is a new scene introduced after translating the sonar in $+x$ direction by 0.5 m. As the image is synthesized using original image, hence this new scene is not known to us.

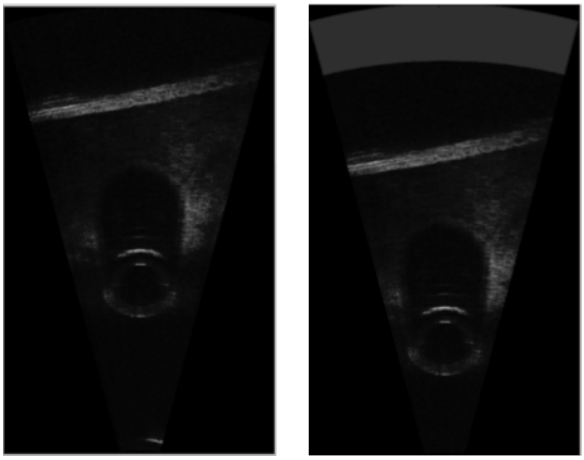


Figure 4.5: (a) Original sonar image. (b) Synthesized sonar image at a relative distance of 0.5m

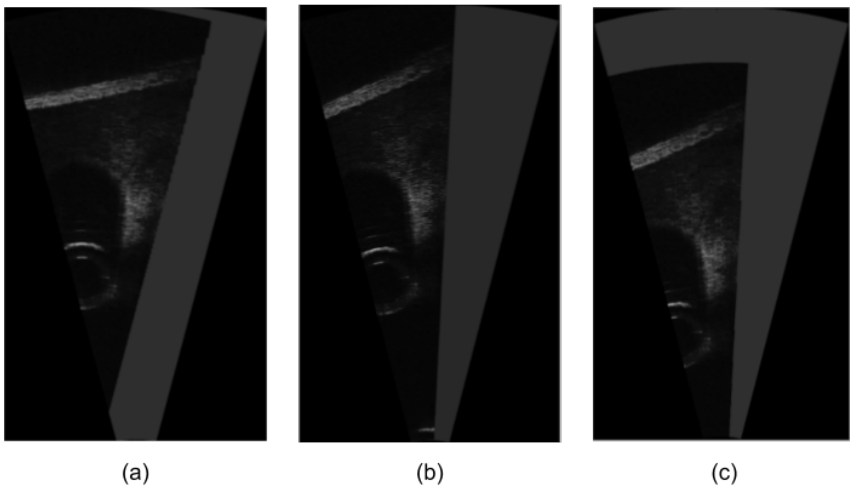


Figure 4.6: Sonar image transformations

2. **Translation in $+y$ direction by** 0.5 **m** [fig. 4.6 (a)].

3. **Rotation about $+z$ axis by** 12 **deg** [fig. 4.6 (b)].

4. **Rotation about $+z$ axis by** 12 **deg, translation in $+x$ and $+y$ direction by** 0.5 **m and** 0.1 **m respectively** [fig. 4.6 (c)].

Based on the above transformation results, we confirm that view synthesis supervision can be used for sonar images as well, given if elevation map and relative pose is known. This concept is used in a loss function to drive the SonarNet model as image reconstruction loss which is analogous to optical image reconstruction loss. Next, we describe in detail the SonarNet architecture.

4.3.3 SonarNet Architecture

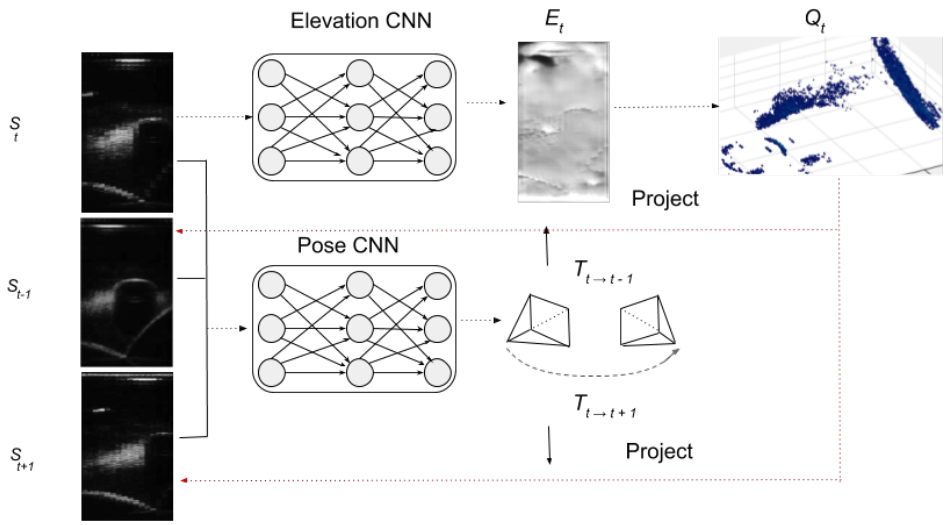


Figure 4.7: Overview of the supervision pipeline based on view synthesis. The elevation network takes only the target view as input, and outputs a per-pixel elevation map E_t. The pose network takes both the target view (S_t) and the nearby/source views (e.g., S_{t-1} and S_{t+1}) as input, and outputs the relative camera poses ($T_{t\to t-1}, T_{t\to t+1}$). The outputs of both networks are then used to inverse warp the source views using Q_t to reconstruct the target view, and the image reconstruction loss is used for training the CNNs. By utilizing view synthesis as supervision, we are able to train the entire framework in an unsupervised manner from raw sonar image dataset.

In this section, we briefly describe the SonarNet model i.e elevation and pose network architecture. Again, SonarNet proposed for underwater sonar sequences can be jointly trained. Despite being trained together it can be used independently during test-time inference.

Elevation Map Network

The elevation map network architecture is very much similar to the single view depth network architecture except here we don't have inverse depth prediction. Given a single sonar image (gray scale image in polar form) as input it produces a elevation estimate mapping each pixel of the input to an elevation angle value within the sonar senor's elevation angle range (i.e -7 deg to $+7$ deg). All *conv* layers are followed by a ReLU activation function except for the prediction layers where sigmoid function is used to get the elevation angle values mapped between 0 and 1.

Pose Network

The input to the pose estimation network is a sequence of three sonar images (e.g. S_{t-1}, S_t and S_{t+1}). We concatenate the target image (i.e. image S_t) with nearby source images (i.e. S_{t-1} and S_{t+1}) along the color channel. The outputs of the pose network are relative poses between the target and each of the source views (i.e. $T_{t\to t-1}$ and $T_{t\to t+1}$) each of which is represented by six numbers corresponding to relative 3D rotations (Euler angles) and metric translations between the two frames. All *conv* layers are followed by ReLU acivation function except for the last layer where no nonlinear activation is applied.

4.3.4 Learning Setup

The learning setup is similar to the the setup used for optical images (section 4.2.4) except here we also try to experiment with the elevation map smoothness loss apart from the smoothness loss used for depth maps. For smoothness, we minimize the $L1$ norm of the second-order gradients for the predicted elevation map.

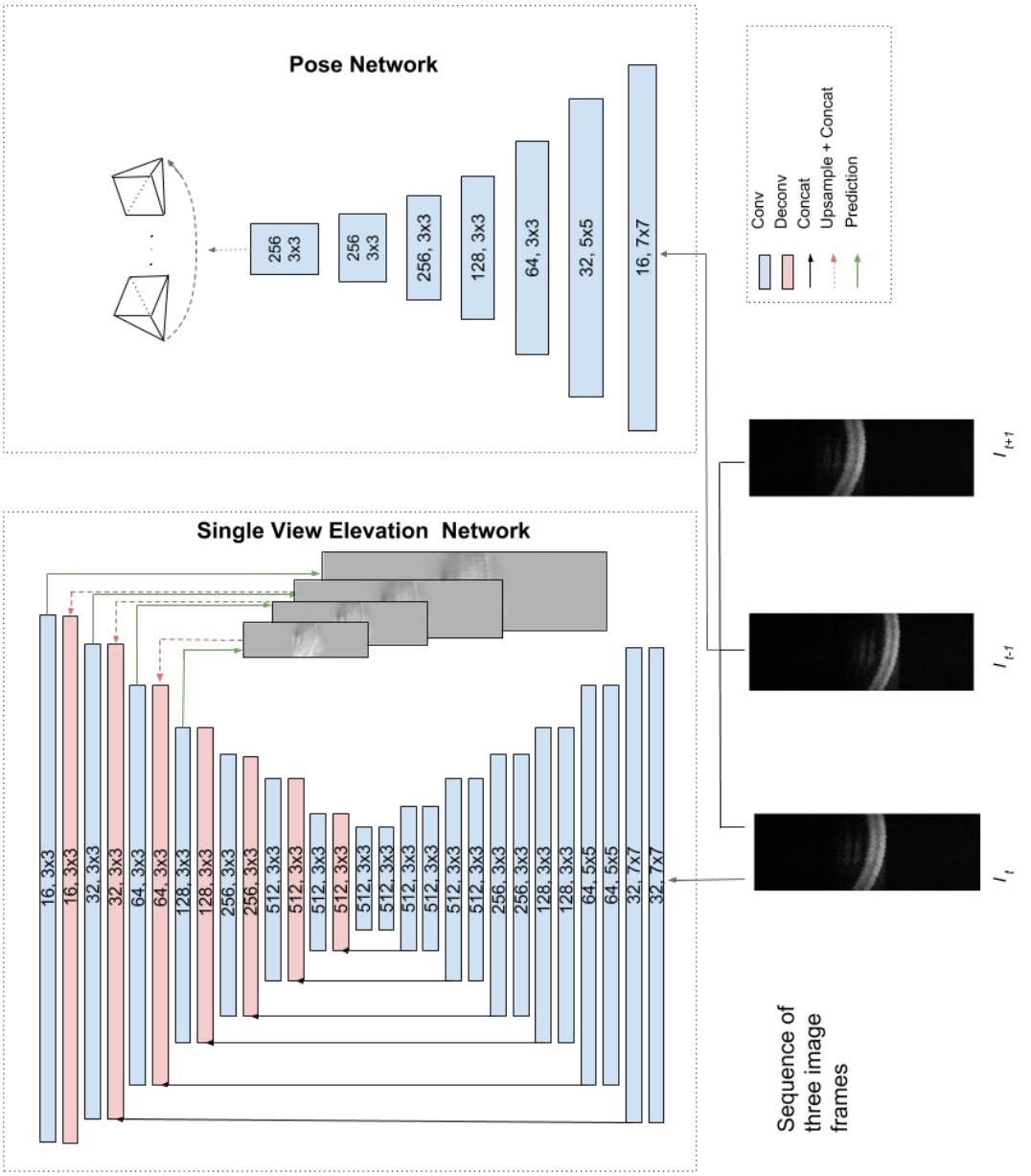


Figure 4.8: Network architecture for the SonarNet model. The width and height of each rectangular block indicates the output channels and the spatial dimension of the feature map at the corresponding layer respectively, and each reduction/increase in size indicates a change by the factor of 2. For single-view depth network, we adopt the SfMLearner architecture which is in turn based on DispNet [48] architecture with multi-scale side predictions. (b) The pose networks predict 6-DoF. relative pose.

Chapter 5

Experiments and Results

In this chapter, we present experimental details and show results of our proposed deep learning framework in two sections. Firstly, we explain model training details and results of depth map and pose estimation from monocular optical images. In the second section, we provide model training details and experimental results of elevation map and pose estimation from sonar image sequences.

5.1 Depth Map and Pose for Monocular Camera Images

In this section, we demonstrate the results of our proposed deep learning framework using monocular underwater image sequences.

5.1.1 Training Details

We implemented the proposed system using the publicly available Tensorflow framework [3]. For all the experiments, as per the learning setup in section 4.2.4, we set the value of $\alpha = 0.15$, $\beta = 0.85$ and $\gamma = 0.1$ for all the scales. During training, we used batch normalization [36] for all the layers except for the output layers, and the Adam [41] optimizer with $\beta_1 = 0.9$, $\beta_2 = 0.999$, learning rate of 0.0001 and mini-batch size of 16. We trained the system for a maximum iteration number of $200,000$.

We train the system using *Universitat de Girona's* publicly available data set collected with an AUV testbed in the unstructured environment of an underwater cave complex [47]. It contains a total of 9774 raw image frames. having a dimension of 288×384. Both the depth and pose networks can be run fully-convolutionally for images of arbitrary size at test time.

5.1.2 Single View Depth Estimation

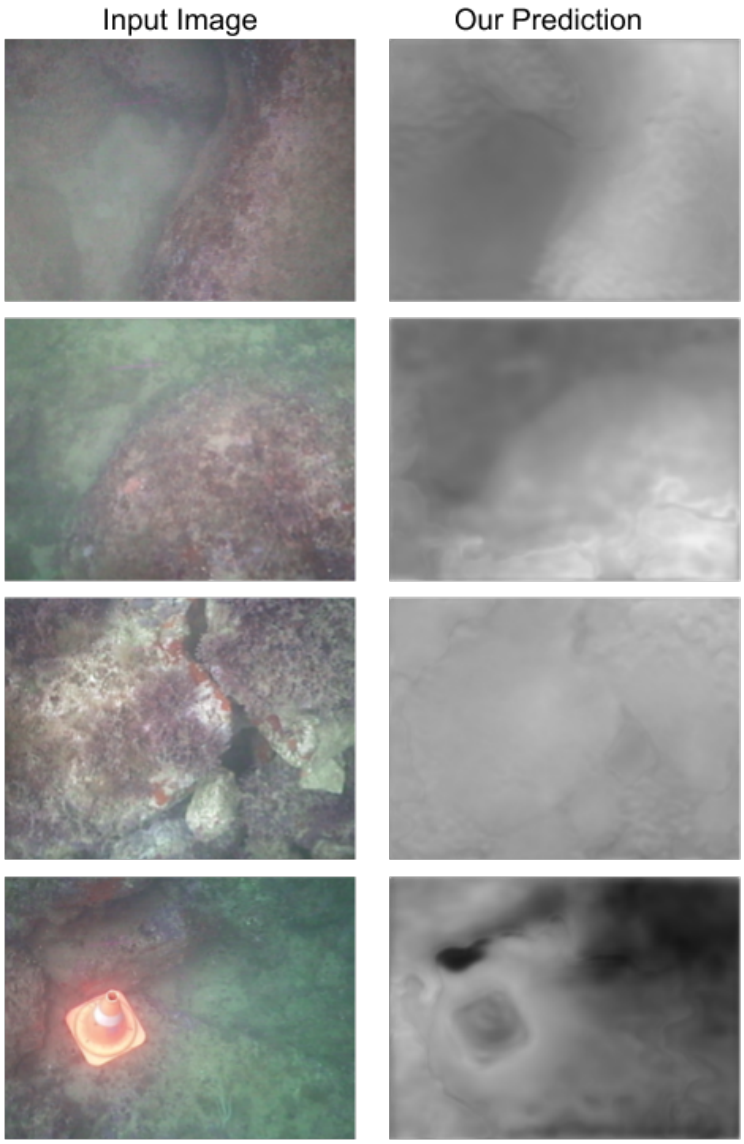


Figure 5.1: Our sample depth predictions on the Girona's underwater dataset using the model trained on the same dataset

We train our depth and pose network simultaneously on modern GPU TITAN Xp by fixing the length of image sequences to be 3 frames and treat the central frame as the target view and

the ±1 frames as the source views. This results in a total of 9772 sequences. To the best of our knowledge, no previous systems exist that learn single-view depth estimation for underwater scenario in an unsupervised manner from monocular videos. Therefore, currently there is no ground truth depth map data available for underwater dataset. We only present the depth map predictions for the model trained on Girona's underwater dataset as shown in fig. 5.1. It is to be noted that the depth predicted by our method is defined only up to a scale factor.

As mentioned earlier that currently we do not have ground truth depth map of the used dataset, the only way to validate our prediction is by reconstructing the point cloud of a scene with the help of predicted depth maps. In the following figures, we show examples of reconstructed dense 3d point clouds using input image and its corresponding predicted depth map. We scale the depth map to a max range of 10 m. Note here the camera coordinate is at 10 m along z axis.

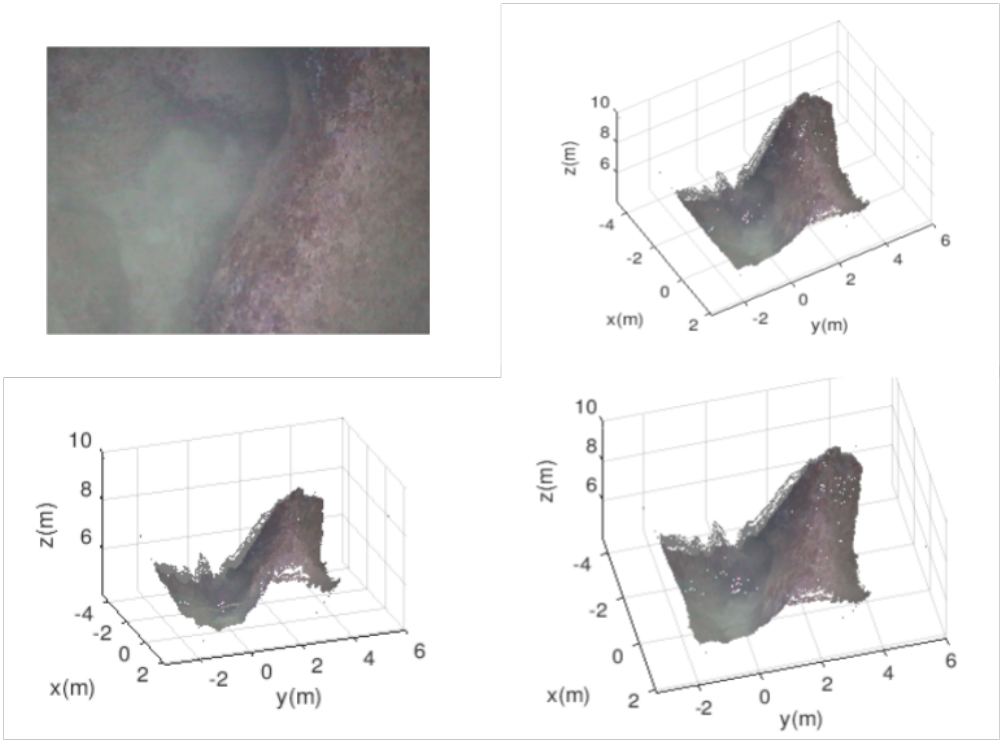


Figure 5.2: Example 1 - Reconstructed 3D point of an input image with different views

From the examples demonstrated, we can say that our depth predictions are quite reasonable and appreciable. This is the first ever approach to estimate 3D scene in underwater scenarios

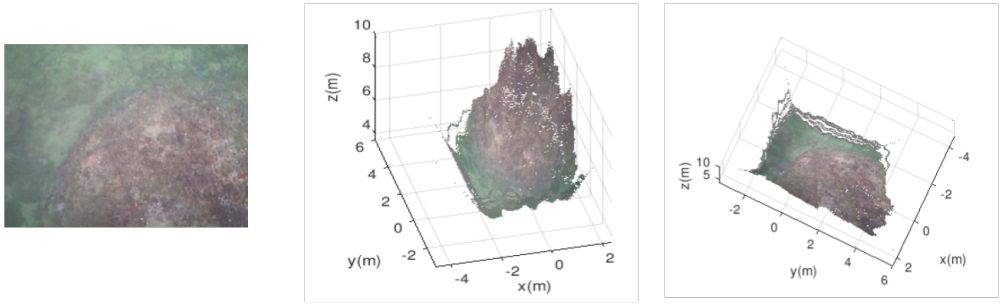


Figure 5.3: Example 2 - Reconstructed 3D point of an input image with different views

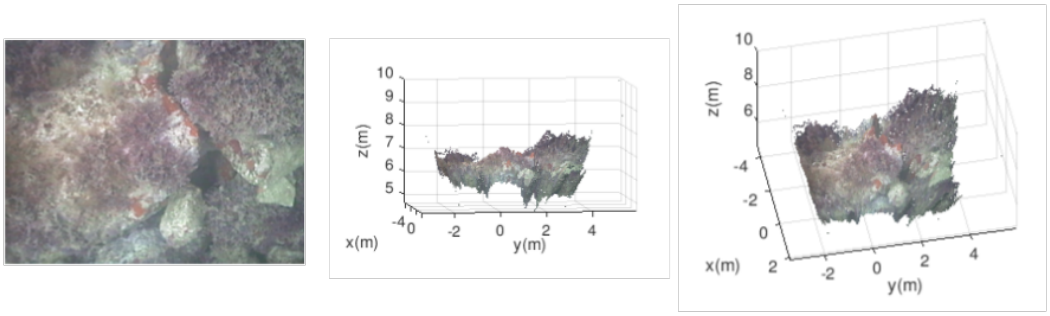


Figure 5.4: Example 3 - Reconstructed 3D point of an input image with different views

just using input image.

Limitations

There are very few publicly available underwater optical image datasets. Currently, we have trained our framework using only 9772 image sequences which is not sufficient enough for unsupervised learning. We also trained the network on Marine Robotics datasets - Tasmania O'Hara provided by Australian Centre for Field Robotics (ACFR) [26]. The number of image frames were approximately equal to Girona's underwater dataset which is again not sufficient for learning. Also, we would like to point out that ACFR's dataset were having significantly huge motion between each frames as comapared to Girona's underwater dataset which produces difficulty in learning these challenging underwater scenarios.

Girona's underwater image dataset is collected with the help of artificial illumination because

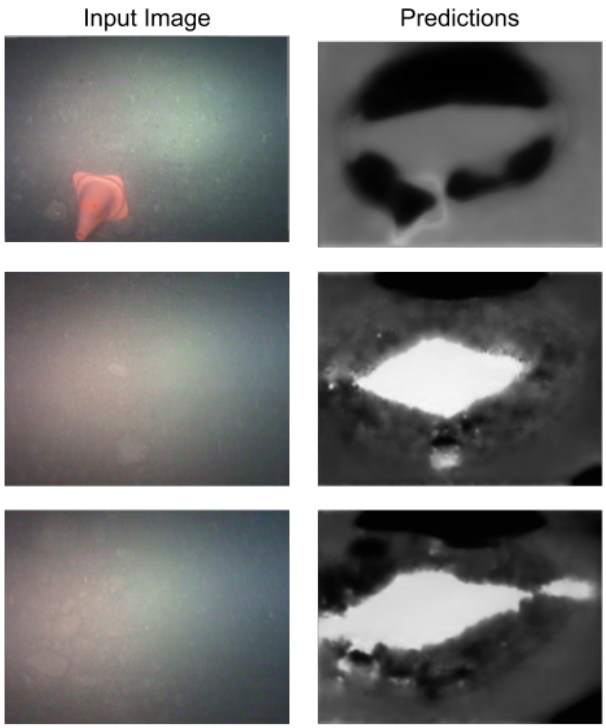


Figure 5.5: Depth map predictions affected due to artificial illumination

of the complex cave environment. This creates noticeable bright and dark spots in the images along with the haze. These bright spots significantly affect the depth map prediction as can be seen in the above fig. 5.5. Also the spots created due to artificial light are very challenging to remove in the pre-processing stage. We performed pre-processing on the dataset to remove haze in the underwater images. We used Contrast Limited Adaptive Histogram Equalization (CLAHE) algorithm [74] on images to remove the haze as some visual features were not visible due to haze, but it worsened the training process, as the bright spots were now even more effective.

5.1.3 Pose Estimation

In the previous section, we pointed out some limitations and discussed some of the reasons of bad depth map predictions. This consequently does affect the pose estimation. As mentioned earlier, we make use of relative poses in view synthesis supervision to compute our image reconstruction loss and this implicitly affects the structured similarity loss thereby increasing

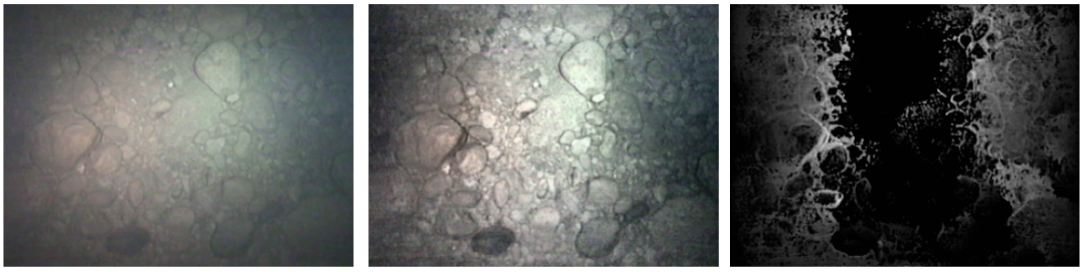


Figure 5.6: (a) Original Image, (b) CLAHE processed image and (c) Depth Map prediction. The bright spots become even more effective. Thereby the depth prediction gets affected

the projection error. Due to the limited dataset, our network has not been able to fully learn the complex underwater environment and we think that our pose estimation may not be correct at this point. Nevertheless, we still plot the entire trajectory of Girona's underwater dataset.

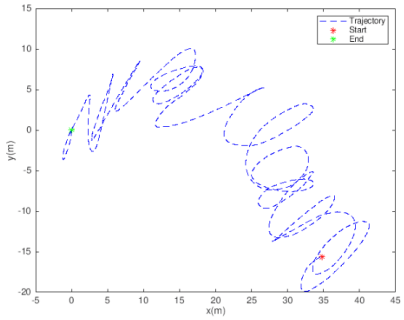


Figure 5.7: 2D pose trajectory of camera for Girona's underwater dataset

The fig. 5.7 shows that there is a lot of drift and the 2D pose trajectory is not correct at this point. This can be significantly improved if more number of underwater optical images is provided.

5.2 Elevation Map and Pose for Sonar Images

In this section, we demonstrate the results of SonarNet model using polar form sonar image sequences.

5.2.1 Training Details

We implemented the SonarNet model again using the Tensorflow framework. For all the experiments, as per the learning setup, we set value of $\alpha = 1$, $\beta = 0$ and $\gamma = 0.25$ for all the scales. During training, we used batch normalization for all the layers except for the output layers, and the Adam optimizer with $\beta_1 = 0.9$, $\beta_2 = 0.999$, learning rate of 0.00001 and mini-batch size of 16. We trained the system for a maximum $300,000$ iterations.

The model is trained using the available underwater sonar dataset collected in a water tank at Ocean Systems Laboratory, Heriot-Watt University. It contains a total of $21,348$ sonar image frames, having a dimension of 1352×128. We resize this image dataset to 467×128 as an appropriate image size for CNN's input.

5.2.2 Elevation Estimation

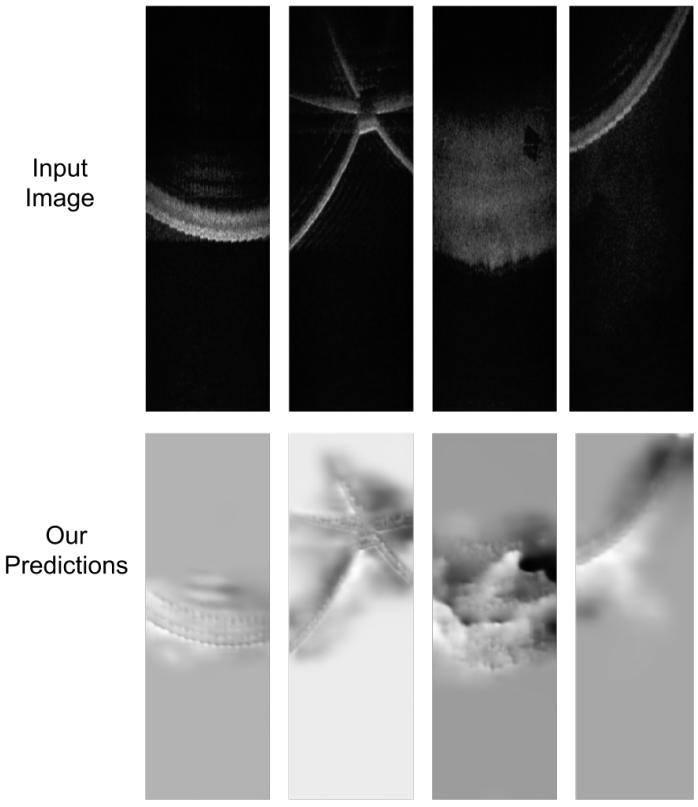


Figure 5.8: Our sample elevation predictions on the sonar underwater dataset collected in a water tank using the model trained on the same dataset

We trained the SonarNet model i.e. both elevation and pose network simultaneously on GPU TITAN Xp. Using the aforementioned sonar image dataset, we drop every five images to form a sequence of three images which amounts to a total of $20,550$ sonar image sequences (polar form) for training. The dataset is downsampled so as to get noticeable motions between each consecutive frames in a sequence. The central frame is treated as the target view and the ± 1 frames as the source views. We believe that this is the first work that learns single view elevation estimation using sonar images of underwater scenarios in an unsupervised manner. Therefore, currently there is no ground truth elevation map data available for evaluation of our model. We present the elevation map predictions for the model trained using the previously mentioned dataset as shown in fig. 5.8.

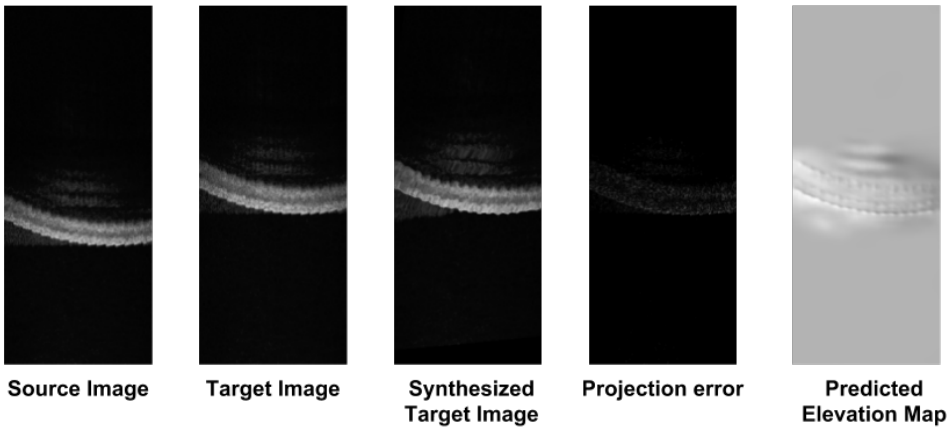


Figure 5.9: Sonar image view synthesis

In the fig. 5.9, we show an example of view synthesis using sonar image of a tank boundary. The concept of view synthesis is used to reconstruct the target image using source view image which is then used in a loss function (image reconstruction loss) along with smoothness loss to drive the SonarNet model.

Limitations

Currently, there are no publicly available underwater sonar image datasets. We have trained the SonarNet model using the existing dataset available at Ocean Systems Lab which was collected in a water tank. We believe that the number of sonar image sequences used for training i.e.

$20,550$ is not sufficient enough for unsupervised learning approach. Secondly, the sonar image dataset which is used for training is very sparse in nature meaning there are no significant features present in the image which is essential for sonar view synthesis supervision (fig. 5.10). Also, it enhances the performance of CNN when images have more features thus enabling them to learn better kernel weights for elevation prediction.

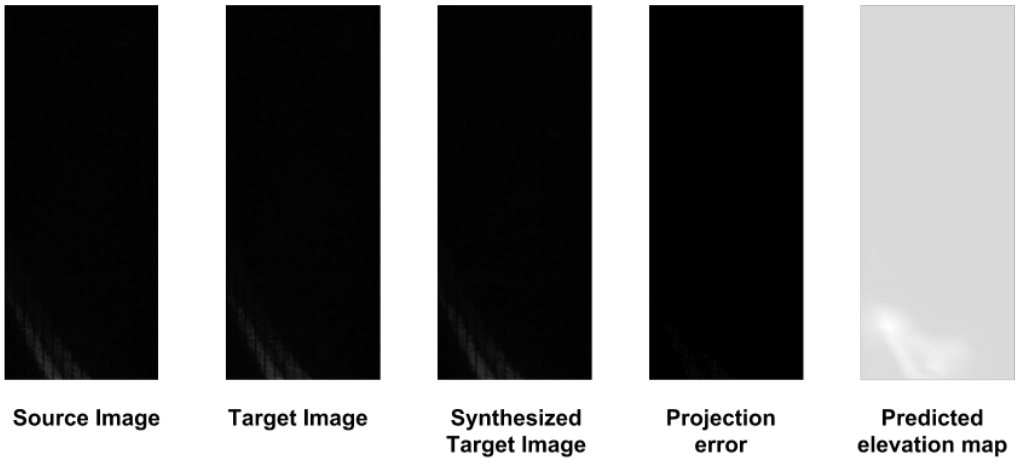


Figure 5.10: Case when sonar image is very sparse thereby affecting the training process and predictions

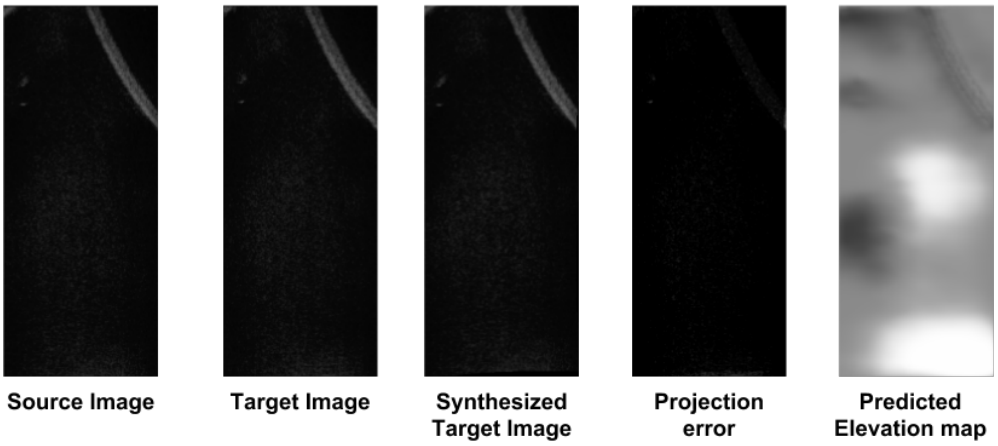


Figure 5.11: Elevation map predictions affected

Sonar image at the acquisition time, are corrupted with noise. Due to sonar image charac-

teristics, the noise in the sonar images do not have a fix pattern. Thereby, most of the time even though the projection error image is visually sparse but still some noise is present in the image. Owing to which the elevation predictions are affected as the model tries to fit this random noise of sonar images (fig. 5.11). We also denoised the sonar image in frequency domain and trained the network to which we found out due to sparse sonar image dataset, the model was not able to learn enough to give reasonable elevation predictions.

5.2.3 Pose estimation

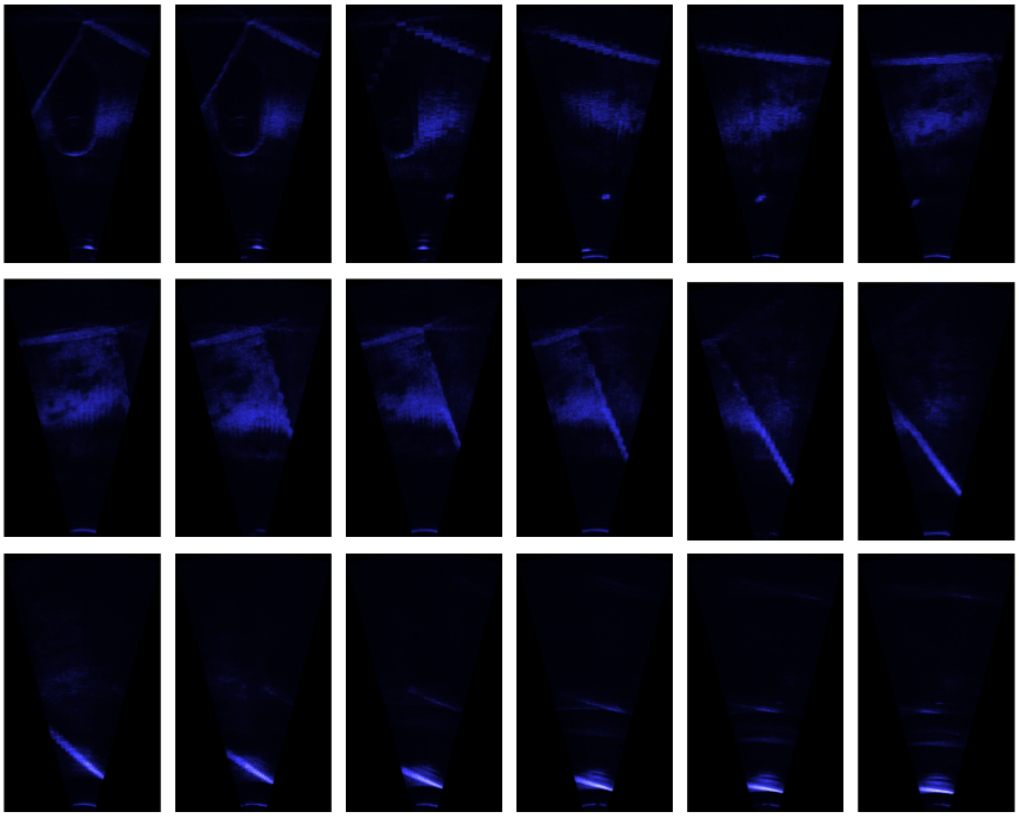


Figure 5.12: Sonar image frames in Cartesian form with ocean color map

As mentioned earlier in the case of optical images, the pose estimation implicitly depends on the quality and accuracy of elevation map prediction. Nevertheless, we plot the pose trajectory of sonar image frames as shown in fig. 5.12. Note the figure just shows the 18 sample frames in cartesian form from start to end for easy interpretation of the motion. But in reality it

consists of 72 sonar image frames. The predicted pose trajectory of the sonar can be seen in

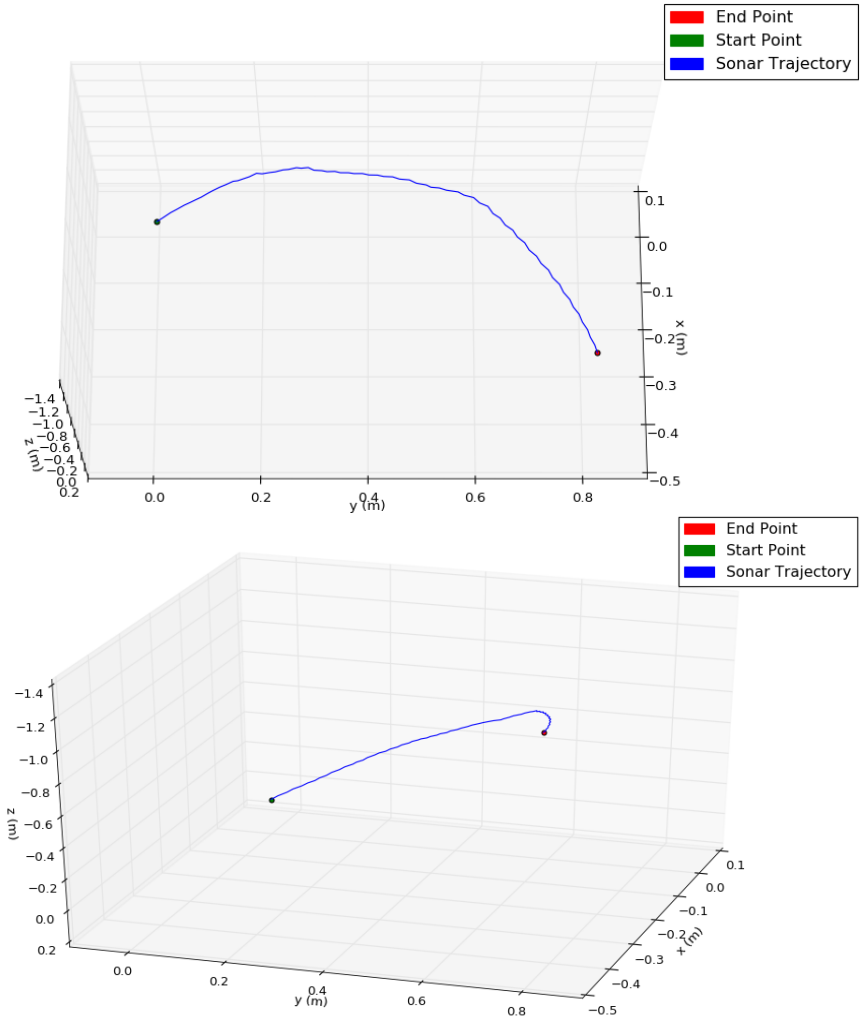


Figure 5.13: Sonar pose trajectory for images shown in fig. 5.12 image frames

fig. 5.13. Here, we only plot relative translation coordinates. To build the full trajectory we combine all these poses groups in the following manner: In 3 sequence case - first, we collect 3-poses sequences for the overlapping frame ids: e.g. first group - frames [1,2,3], second group - frames [3,4,5] etc. Then we translate each sequence so that the coordinate system is set to the first pose in the group. Then we iterate through these groups and use the overlapping pose to translate every poses sequence to global coordinate system.

The predicted pose trajectory seems reasonable as the motion interpreted from the shown

sonar image frames look as if the sonar is entirely rotating about $+z$ axis with some sideways motion along y axis. And this is validated in the pose trajectory as there is very little motion along x axis and more motion along y axis and about $+z$ axis. At this point, we do not have ground truth pose of the sonar through which this can be validated. Also, we do not know the initial configuration of sonar like grazing angle, the depth at which sonar was present etc. as we use the existing sonar image dataset. Overall, it does give us some reasonable pose prediction of the sonar sensor solely based on sonar image frame sequences even though there is a little drift along z axis which cannot be validated as we are not aware of the positioning of the sonar sensor that was set during data collection.

Presently, we do not have any groundtruth neither for elevation nor for pose. Considering this is the first work done based on learning approach for sonar images, we also try to reconstruct a point cloud using a particular sonar image frame (using fig. 5.9) and its predicted elevation map. The reconstructed point cloud of the scene is shown in the below fig. 5.14

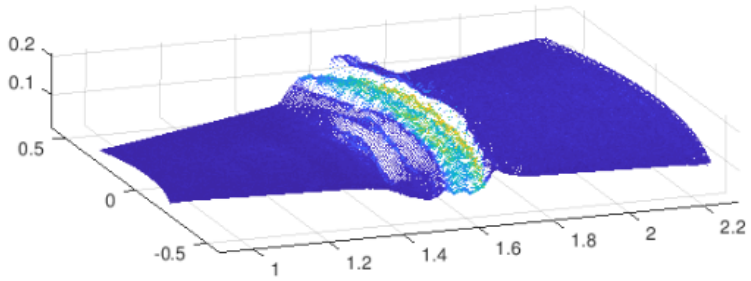


Figure 5.14: Point cloud of a scene using input image shown in fig. 5.9

The scene in fig. 5.9 contains a tank boundary. As can be seen in fig. 5.14, we get a reasonable elevation of the tank boundaries to a moderate extent. Though it is not accurate one as it explicitly depends on quality of elevation map predictions. Currently, the model has not been able to completely learn underwater scenario due to limited sonar dataset. We believe that with more training data, this can be significantly improved.

Chapter 6

Conclusion

6.1 Final Remarks

In this master thesis, we proposed two unsupervised deep learning framework for 3D reconstruction and motion estimation in underwater scenarios using optical and sonar images individually based on the recent work performed in the field of Deep Learning. We carried out this thesis work in three main parts which are detailed as follows:

1. **We reviewed state of the art conventional methods for 3D reconstruction and motion estimation in underwater scenarios**. The analysis was done in terms on their processing pipeline and their advantages and disadvantages were discussed. We also observed that, most of the work for 3D reconstruction and motion estimation is done using optical images in spite of its limited visibility range. There are existing methods that make use of sonar images which have managed to achieve sparse 3D reconstruction and have shown great potential in doing the same.

2. **We also reviewed state of the art CNN based deep learning methods for optical images for 3D reconstruction and motion estimation.** The considered methods were analyzed in terms of their architecture and processing pipelines. After an extensive review of the different techniques, we observed that there was no existing CNN based deep learning method for 3D reconstruction and ego-motion estimation in underwater scenarios. Though there is a rich literature on object detection, recognition etc both for optical and sonar images in underwater scenarios, but learning of 3D scenes and ego-motion for underwater environments are not yet addressed.

3. With this motivation set and based on the state of the art deep learning methods, **we**

proposed two unsupervised deep learning framework for underwater scenarios using optical and sonar images individually that learns the 3D scenes and ego-motions.

(a) **Depth and pose prediction network using unstructured sequences of optical images**. We adopted the SfMLearner architecture which in turn is based on DispNet architecture. The model has single view depth network and a pose network which are jointly trained. Considering the case of underwater scenario, we proposed a new loss function i.e structured similarity metric loss in addition to the image reconstruction loss used in SfMLearner's architecture. We also modify the smoothness loss by taking into consideration the gradients of the input image. We were able to get reasonable single view depth predictions as shown in the results. Due to limited dataset, artificially illuminated images etc. the depth prediction got affected and subsequently the pose prediction of the network was not proper during the training process. Being unsupervised learning framework, the network has not been able to learn enough features of the complex underwater environment due to limited dataset used for training.

(b) **Elevation and pose prediction network using unstructured sequences of sonar images (polar form), also known as *SonarNet***. Based on the above model, we extended this model for sonar images to estimate elevation and pose. The SonarNet architecture is very much similar to the model used for optical images except here we didn't use structured similarity loss as sonar images do not have much features like color, texture etc. SonarNet model is completely based on image reconstruction loss and smoothness loss. The smoothness loss used here also varies from the above described model. Instead here we minimized the $L1$ norm on the second order gradients of the image. We were able to get satisfactory elevation map and pose predcitions as shown in the results. With more amount of dataset the training performance can be significantly improved.

6.2 Challenges

In this master thesis project, we faced many challenges during the process which we would like to mention here in this section. Also, considering the fact that this was the first ever attempt to explore the CNN based deep learning methods for learning 3D scenes and pose estimates in underwater scenarios, the work itself was very challenging at the beginning as most of the existing learning methods are applied to optical images on land. We list down some of the challenges which we faced during the project.

- Apart from Uiveristat de Girona's underwater cave dataset and ACFR's Tasmania O'Hara dataset, there are no publicly available underwater datasets. The underwater optical image dataset in reality is very limited. Though the method is unsupervised, our method can not be applied to any random video sequences without the knowledge of intrinsic camera parameter with which the sequence is captured.

- Collection of optical underwater dataset requires the use of underwater vehicles which was not in operation at Heriot-Watt University due to technical and logistic issues. Hence, we were not able to collect the data by ourselves before the end of this master project in order to improve the model's training performance.

- Unsupervised Learning was proposed simply because currently there is no ground truth depth map available for underwater scenarios to incorporate it as supervisory signal in the network so as to make the model completely supervised.

- Presently, there is no existing dataset available for sonar images. To our best knowledge, there is no existing work, that uses sonar image to learn 3d scenes and pose estimates in underwater environments. This itself makes it challenging enough as there is no ground truth to validate the predictions.

- The sonar dataset used for training SonarNet model is the existing one available at Ocean System's Laboratory, Heriot-Watt Univeristy. This dataset was originally intended for underwater object detection and recognition using sonar images. As the dataset had considerable motion between the frames, we used this dataset for training the SonarNet model. Again, due to limited dataset, the model has not been able to learn good kernel weights to understand the scene features. Also, the images in the dataset were very sparse which implicitly affected the learning process.

- We collected some data in a water tank using sonar sensor, but the sonar sensor was manually moved meaning it was not mounted on any test platform underwater robot. The sonar sensor being heavy, we were not able to collect enough good quality sonar image frames.

- For the existing sonar dataset, there is no ground truth pose trajectory available through which we can validate our predictions and evaluate the model.

These were some of the challenges which we faced during the project term and we would like to incorporate those challenges as a future work for this project.

6.3 Future Work

In this section, we present some ideas to carry out as future work. Some of them were not implemented for this master thesis due to time constraints.

Although a large number of experiments have been carried during this master thesis, but the results presented in this thesis are the ones which gave us reasonable results and show a great potential if supplemented with large number of good datasets both for sonar and optical images.

We would like to continue in this research line as follows:

1. Firstly, we will collect large number of good quality actual underwater data set - both optical and sonar image frames. This data collection will take place during end of June at Fort William, Scotland. During this dataset collection, we will also have GPS coordinates of UAV so that we can validate our pose trajectory predictions and evaluate the model.

2. We will train these datasets using our proposed model to check the model's performance.

3. We also plan to modify the depth network and elevation network depending on the characteristics of the image frames collected. This will be addressed by increasing the number of levels in multi scale depth/elevation predictions or by increasing the layers in encoder-decoder design.

4. Currently, the pose network is completely CNN based. We would like to experiment it by leveraging the deep Recurrent Convolutional Neural Networks (RCNNs) as they have proved to be very promising in learning pose for visual odometry on land.

5. Currently, we trained SonarNet model using polar form of sonar images. We would like to train the network using cartesian form of sonar images to see if the model is able to learn better features than the former.

6. We believe that currently our predictions are reasonable enough but not accurate, hence reconstructed 3D scenes were not perfect. We would like to experiment more on this and construct a 3D underwater environment based on our model's predictions.

7. We would also like to incorporate depth extracted from nearby frames by directly comparing 3D point clouds in a common reference frame i.e. a 3D information based loss function.

In the distant future, we believe that with this very first attempt on learning 3D reconstruction and pose estimates for underwater scenarios, we have obtained background on the field which would be the pillar of the PhD.

Appendix A

Miscellaneous

A.1 ARIS 3000 Sonar System Configuration

Following configurations were set for ARIS 3000 sonar system :

- **windowStart** = 0.9425 m i.e. minimum range
- **windowEnd** = 2.2968 m i.e. maximum range
- **beamCount** = 128
- **Field of view** = 30^o i.e. (-15 deg to +15 deg)
- **Elevation angle** ranges from (-7 deg to +7 deg) i.e. 14 deg

A.2 Pixel coordinate to Real world coordinate for Sonar Imagery

For this project we make use of sonar image in polar form (1352×128) and then resize it to 467×128. Here we show an example using original image for scaling. In the polar image form, the viewing is from the top A.1.

We convert this pixel coordinate (r, θ) to real world coordinate (R, Θ) in meters and degrees respectively by linearly scaling using the following:

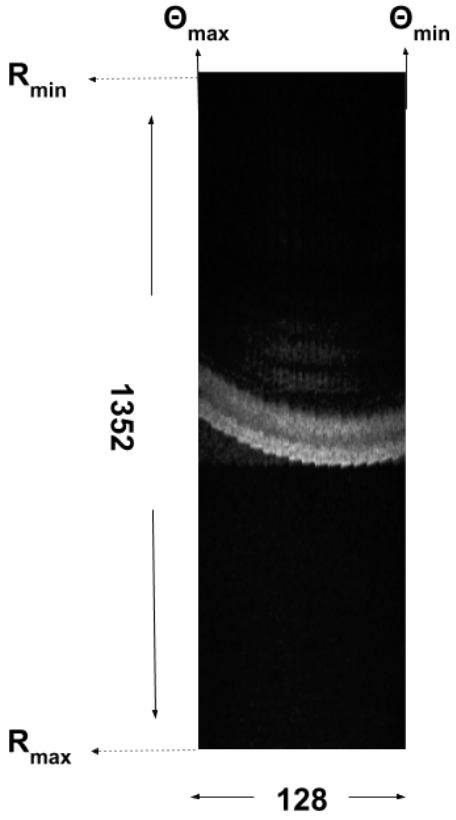


Figure A.1: Raw sonar image specifications. View from the top

$$R = \frac{R_{max} - R_{min}}{1351} r + R_{min} \tag{A.1}$$

$$\Theta = \frac{\Theta_{max} - \Theta_{min}}{-127} (\theta - 127) + \Theta_{min} \tag{A.2}$$

Note in python indexing starts from zero.

A.3 Real world coordinate to pixel coordinate for Sonar Imagery

Similarly, we scale back to get the pixel coordinates using the following:

$$r = 1351 \frac{R - R_{min}}{R_{max} - R_{min}} \tag{A.3}$$

$$\theta = -127 \frac{\Theta - \Theta_{min}}{\Theta_{max} - \Theta_{min}} + 127 \tag{A.4}$$

A.4 Training loss curves - optical images

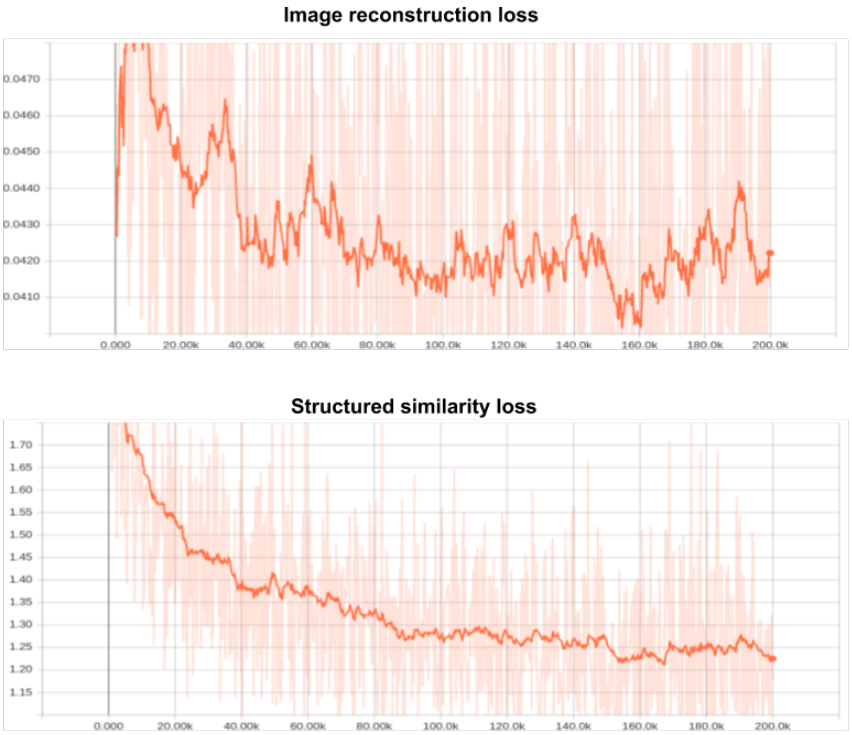


Figure A.2: Image reconstruction and structured similarity loss curves during training upto 200,000 iterations

From, the above figure, we see that the image reconstruction loss and structured similarity

loss is not yet converged. Due to which it is not able to learn complex underwater environment as the dataset was not sufficient enough for this unsupervised learning framework. The above

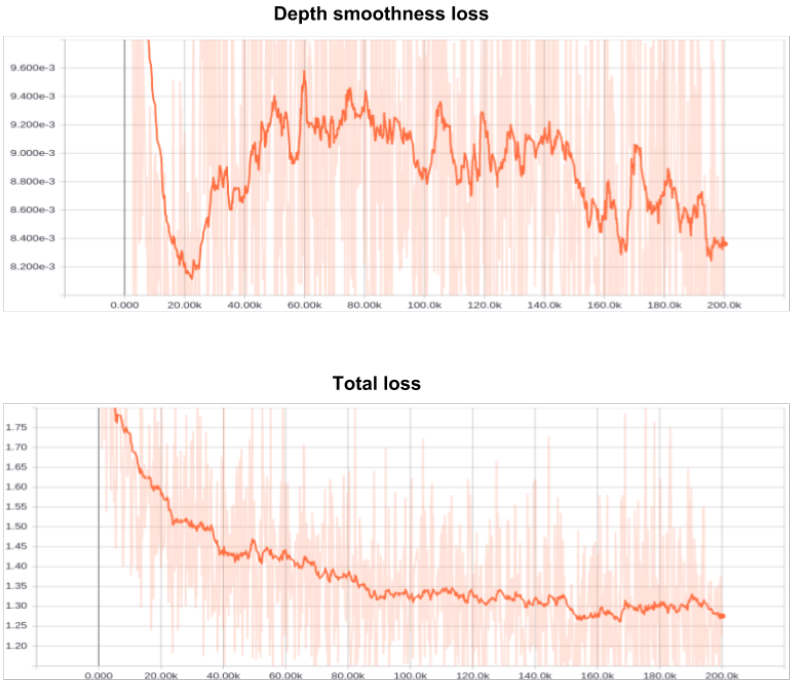


Figure A.3: Depth smoothness and total loss curves during training upto 200,000 iterations

figures show the depth smoothness loss and total loss curves. The smoothness loss has minor perturbations and this is because of the quality of dataset which were having dark and bright spots. This affected our depth map predictions. Hence the total loss didn't converge properly and were not able to learn good kernel weights useful for feature extraction.

The above training loss curves shown are for the ones that gave us reasonable and logical predictions which is presented in the thesis.

A.5 Training loss curves - sonar images

The below figure shows the training loss curves for underwater sonar images. We can clearly see in the figure that the image reconstruction loss for sonar images has not yet converged because of the inconsistencies in the dataset. The dataset contained many sparse sonar images as the

data was originally collected in a water tank. This also affected the elevation smoothness loss and the loss started increasing even though the values are very small. Hence the total loss didn't converge and the network was not able to lean significant features which affected both elevation map and pose estimates.

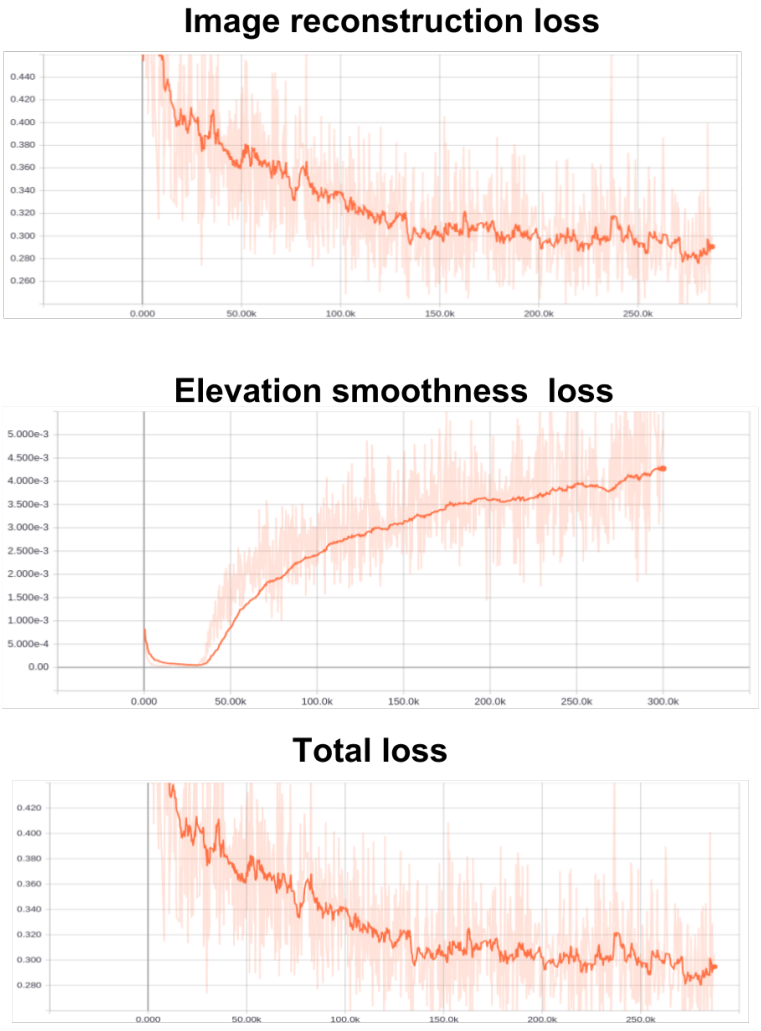


Figure A.4: Image reconstruction, elevation smoothness and total loss curves during training up to 300,000 iterations

Bibliography

[1] Blueview technologies, inc. (2018). `http://www.blueview.com`. Accessed: May 30, 2018.

[2] Soundmetrics corp. (2018). `http://www.soundmetrics.com`. Accessed: May 30, 2018.

[3] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, et al. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *arXiv preprint arXiv:1603.04467*, 2016.

[4] Pulung Nurtantio Andono, Ricardus Anggi Pramunendar, Catur Supriyanto, Guruh Fajar Shidik, I Ketut Eddy Purnama, and Mochamad Hariadi. Enhancement of 3d surface reconstruction of underwater coral reef base on sift image matching using contrast limited adaptive histogram equalization and outlier removal. *Kursor*, 7(1), 2013.

[5] Ifu Aniemeka. A friendly introduction to convolutional neural networks. `https://bit.ly/2JaVD0w`. Accessed: May 30, 2018.

[6] Hassan Assalih et al. *3D reconstruction and motion estimation using forward looking sonar*. PhD thesis, Citeseer, 2013.

[7] Murat D Aykin and Shahriar Negahdaripour. Forward-look 2-d sonar image formation and 3-d reconstruction. In *Oceans-San Diego, 2013*, pages 1–10. IEEE, 2013.

[8] Murat D Aykin and Shahriar Negahdaripour. On feature matching and image registration for two-dimensional forward-scan sonar imaging. *Journal of Field Robotics*, 30(4):602–623, 2013.

[9] Murat D Aykin and Shahriar Negahdaripour. On 3-d target reconstruction from multiple 2-d forward-scan sonar views. In *OCEANS 2015-Genova*, pages 1–10. IEEE, 2015.

[10] M Babaee and S Negahdaripour. 3-d object modeling from occluding contours in opti-acoustic stereo images. In *Oceans-San Diego, 2013*, pages 1–8. IEEE, 2013.

[11] Chris Beall, Brian J Lawrence, Viorela Ila, and Frank Dellaert. 3d reconstruction of underwater structures. In *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, pages 4418–4423. IEEE, 2010.

[12] Ashwin Bhandare, Maithili Bhide, Pranav Gokhale, and Rohan Chandavarkar. Applications of convolutional neural networks. *International Journal of Computer Science and Information Technologies*, pages 2206–2215, 2016.

[13] Francisco Bonin-Font, Aleksandar Cosic, Pep Lluis Negre, Markus Solbach, and Gabriel Oliver. Stereo slam for robust dense 3d reconstruction of underwater environments. In *OCEANS 2015-Genova*, pages 1–6. IEEE, 2015.

[14] Naouraz Brahim, Didier Guériot, Sylvie Daniel, and Basel Solaiman. 3d reconstruction of underwater scenes using didson acoustic sonar image sequences through evolutionary algorithms. In *OCEANS, 2011 IEEE-Spain*, pages 1–6. IEEE, 2011.

[15] Jason Brownlee. Supervised and unsupervised machine learning algorithms. `https://machinelearningmastery.com/supervised-and-unsupervised-machine-learning-algorithms/`. Accessed: May 30, 2018.

[16] Pep Lluis Negre Carrasco, Francisco Bonin-Font, and Gabriel Oliver Codina. Stereo graph-slam for autonomous underwater vehicles. In *Intelligent Autonomous Systems 13*, pages 351–360. Springer, 2016.

[17] Marc Carreras, Pere Ridao, Rafael García, and Tudor Nicosevici. Vision-based localization of an underwater robot in a structured environment. In *Robotics and Automation, 2003. Proceedings. ICRA'03. IEEE International Conference on*, volume 1, pages 971–976. IEEE, 2003.

[18] Franois Chollet. Deep learning with python. `http://www.deeplearningitalia.com/wp-content/uploads/2017/12/Dropbox_Chollet.pdf`. Accessed: May 30, 2018.

[19] Ronald Clark, Sen Wang, Andrew Markham, Niki Trigoni, and Hongkai Wen. Vidloc: A deep spatio-temporal model for 6-dof video-clip relocalization. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, page 3, 2017.

[20] Julien Despois. Autoencoders - deep learning bits. `https://hackernoon.com/autoencoders-deep-learning-bits-1-11731e200694`. Accessed: May 30, 2018.

[21] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Toward geometric deep slam. *arXiv preprint arXiv:1707.07410*, 2017.

[22] David Eigen, Christian Puhrsch, and Rob Fergus. Depth map prediction from a single image using a multi-scale deep network. In *Advances in neural information processing systems*, pages 2366–2374, 2014.

[23] Nathaniel Fairfield, Dominic Jonak, GA Kantor, and David Wettergreen. Field results of the control, navigation, and mapping systems of a hovering auv. In *Proceedings of the Unmanned Untethered Submersible Technology Conference (UUST)*, pages 20–22. Citeseer, 2007.

[24] Xiaochuan Fan, Kang Zheng, Yuewei Lin, and Song Wang. Combining local appearance and holistic view: Dual-source deep neural networks for human pose estimation. *arXiv preprint arXiv:1504.07159*, 2015.

[25] Martin A Fischler and Robert C Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. In *Readings in computer vision*, pages 726–740. Elsevier, 1987.

[26] Australian Centre for Field Robotics. Marine robotics datasets. `http://marine.acfr.usyd.edu.au/datasets/`. Accessed: May 30, 2018.

[27] Rafael Garcia, J Batlle, Xavier Cufi, and Josep Amat. Positioning an underwater vehicle through image mosaicking. In *Robotics and Automation, 2001. Proceedings 2001 ICRA. IEEE International Conference on*, volume 3, pages 2779–2784. IEEE, 2001.

[28] Rafael Garcia, Xevi Cufi, and Marc Carreras. Estimating the motion of an underwater robot from a monocular image sequence. In *Intelligent Robots and Systems, 2001. Proceedings. 2001 IEEE/RSJ International Conference on*, volume 3, pages 1682–1687. IEEE, 2001.

[29] Ravi Garg, Vijay Kumar BG, Gustavo Carneiro, and Ian Reid. Unsupervised cnn for single view depth estimation: Geometry to the rescue. In *European Conference on Computer Vision*, pages 740–756. Springer, 2016.

[30] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587, 2014.

[31] Clément Godard, Oisin Mac Aodha, and Gabriel J Brostow. Unsupervised monocular depth estimation with left-right consistency. In *CVPR*, page 7, 2017.

[32] Ian Goodfellow, Yoshua Bengio, Aaron Courville, and Yoshua Bengio. *Deep learning*, volume 1. MIT press Cambridge, 2016.

[33] Thomas Guerneve and Yvan Petillot. Underwater 3d reconstruction using blueview imaging sonar. In *OCEANS 2015-Genova*, pages 1–7. IEEE, 2015.

[34] Philipp Heise, Sebastian Klose, Brian Jensen, and Alois Knoll. Pm-huber: Patchmatch with huber regularization for stereo matching. In *Computer Vision (ICCV), 2013 IEEE International Conference on*, pages 2360–2367. IEEE, 2013.

[35] Natàlia Hurtós, David Ribas, Xavier Cufí, Yvan Petillot, and Joaquim Salvi. Fourier-based registration for robust forward-looking sonar mosaicing in low-visibility underwater environments. *Journal of Field Robotics*, 32(1):123–151, 2015.

[36] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.

[37] Sachin Joglekar. Backpropagation for dummies. `https://codesachin.wordpress.com/2015/12/06/backpropagation-for-dummies/`. Accessed: May 30, 2018.

[38] Anne Jordt-Sedlazeck and Reinhard Koch. Refractive structure-from-motion on underwater images. In *Computer Vision (ICCV), 2013 IEEE international conference on*, pages 57–64. IEEE, 2013.

[39] Andrej Karpathy. Convolutional neural netowrks for visual recognition. `http://cs231n.github.io/convolutional-networks/`. Accessed: May 30, 2018.

[40] Alex Kendall, Matthew Grimes, and Roberto Cipolla. Posenet: A convolutional network for real-time 6-dof camera relocalization. In *Computer Vision (ICCV), 2015 IEEE International Conference on*, pages 2938–2946. IEEE, 2015.

[41] D Kinga and J Ba Adam. A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*, 2015.

[42] James C Kinsey, Ryan M Eustice, and Louis L Whitcomb. A survey of underwater vehicle navigation: Recent advances and new challenges. In *IFAC Conference of Manoeuvering and Control of Marine Craft*, volume 88, pages 1–12, 2006.

[43] John J. Leonard, Andrew A. Bennett, Christopher M. Smith, Hans Jacob, and S. Feder. Autonomous underwater vehicle navigation. In *MIT Marine Robotics Laboratory Technical Memorandum*, 1998.

[44] Bo Li, Chunhua Shen, Yuchao Dai, Anton van den Hengel, and Mingyi He. Depth and surface normal estimation from monocular images using regression on deep features and hierarchical crfs. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1119–1127, 2015.

[45] Ruihao Li, Qiang Liu, Jianjun Gui, Dongbing Gu, and Huosheng Hu. Indoor relocalization in challenging environments with dual-stream convolutional neural networks. *IEEE Transactions on Automation Science and Engineering*, 2017.

[46] Ruihao Li, Sen Wang, Zhiqiang Long, and Dongbing Gu. Undeepvo: Monocular visual odometry through unsupervised deep learning. *arXiv preprint arXiv:1709.06841*, 2017.

[47] Angelos Mallios, Eduard Vidal, Ricard Campos, and Marc Carreras. Underwater caves sonar data set. *The International Journal of Robotics Research*, 36(12):1247–1251, 2017.

[48] Nikolaus Mayer, Eddy Ilg, Philip Hausser, Philipp Fischer, Daniel Cremers, Alexey Dosovitskiy, and Thomas Brox. A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4040–4048, 2016.

[49] David McKinnon, Hu He, Ben Upcroft, and Ryan N Smith. Towards automated and in-situ, near-real time 3-d reconstruction of coral reef environments. In *OCEANS 2011*, pages 1–10. IEEE, 2011.

[50] Arnaud Meline, Jean Triboulet, and Bruno Jouvencel. Comparative study of two 3d reconstruction methods for underwater archaeology. In *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, pages 740–745. IEEE, 2012.

[51] Shahriar Negaharipour. On 3-d motion estimation from 2-d sonar image flow. In *Oceans, 2012*, pages 1–7. IEEE, 2012.

[52] Shahriar Negaharipour. On 3-d scene interpretation from fs sonar imagery. In *Oceans, 2012*, pages 1–9. IEEE, 2012.

[53] S Negahdaripour. On 3-d reconstruction from stereo fs sonar imaging. In *OCEANS 2010*, pages 1–6. IEEE, 2010.

[54] Shahriar Negahdaripour. Epipolar geometry of opti-acoustic stereo imaging. *IEEE transactions on pattern analysis and machine intelligence*, 29(10):1776–1788, 2007.

[55] Shahriar Negahdaripour. On 3-d motion estimation from feature tracks in 2-d fs sonar video. *IEEE Transactions on Robotics*, 29(4):1016–1030, 2013.

[56] Shahriar Negahdaripour, Hamed Pirsiavash, and Hicham Sekkati. Integration of motion cues in optical and sonar videos for 3-d positioning. In *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, pages 1–8. IEEE, 2007.

[57] Shahriar Negahdaripour, Hicham Sekkati, and Hamed Pirsiavash. Opti-acoustic stereo imaging: On system calibration and 3-d target reconstruction. *IEEE Transactions on image processing*, 18(6):1203–1214, 2009.

[58] Shahriar Negahdaripour and Ali Taatian. 3-d motion estimation by integrating visual cues in 2-d multi-modal opti-acoustic stereo sequences. *Computer Vision and Image Understanding*, 114(8):928–941, 2010.

[59] Tudor Nicosevici, Nuno Gracias, Shahriar Negahdaripour, and Rafael Garcia. Efficient three-dimensional scene modeling and mosaicing. *Journal of Field Robotics*, 26(10):759–788, 2009.

[60] Hyeonwoo Noh, Seunghoon Hong, and Bohyung Han. Learning deconvolution network for semantic segmentation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1520–1528, 2015.

[61] Oscar Pizarro, Ryan Michael Eustice, and Hanumant Singh. Large area 3-d reconstructions from underwater optical surveys. *IEEE Journal of Oceanic Engineering*, 34(2):150–169, 2009.

[62] Chris Roman and Hanumant Singh. Improved vehicle based multibeam bathymetry using sub-maps and slam. In *Intelligent Robots and Systems, 2005.(IROS 2005). 2005 IEEE/RSJ International Conference on*, pages 3662–3669. IEEE, 2005.

[63] I Tena Ruiz, S Reed, Y Petillot, J Bell, DM Lane, et al. Concurrent mapping & localisation using side-scan sonar for autonomous navigation. In *Proc. Int. Symp. on Unmanned Untethered Submersible Technology*, 2003.

[64] Anne Sedlazeck, Kevin Koser, and Reinhard Koch. 3d reconstruction based on underwater video from rov kiel 6000 considering underwater imaging conditions. In *OCEANS 2009-EUROPE*, pages 1–10. IEEE, 2009.

[65] Sagar Sharma. Activation functions: Neural networks. `https://towardsdatascience.com/activation-functions-neural-networks-1cbd9f8d91d6`. Accessed: May 30, 2018.

[66] Young-eun Song and Seung-Joon Choi. Underwater 3d reconstruction for underwater construction robot based on 2d multibeam imaging sonar. *Journal of Ocean Engineering and Technology*, 30(3):227–233, 2016.

[67] Yanan Sun, Gary G Yen, and Zhang Yi. Evolving unsupervised deep neural networks for learning meaningful representations. *arXiv preprint arXiv:1712.05043*, 2017.

[68] Keisuke Tateno, Federico Tombari, Iro Laina, and Nassir Navab. Cnn-slam: Real-time dense monocular slam with learned depth prediction. *arXiv preprint arXiv:1704.03489*, 2017.

[69] Sen Wang, Ronald Clark, Hongkai Wen, and Niki Trigoni. Deepvo: Towards end-to-end visual odometry with deep recurrent convolutional neural networks. In *Robotics and Automation (ICRA), 2017 IEEE International Conference on*, pages 2043–2050. IEEE, 2017.

[70] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004.

[71] Ying Wu. Image formation and camera calibration. `http://users.eecs.northwestern.edu/~yingwu/teaching/EECS432/Notes/camera.pdf`. Accessed: May 30, 2018.

[72] Xun Xu and Shahriar Negahdaripour. Vision-based motion sensing for underwater navigation and mosaicing of ocean floor images. In *OCEANS'97. MTS/IEEE Conference Proceedings*, volume 2, pages 1412–1417. IEEE, 1997.

[73] Tinghui Zhou, Matthew Brown, Noah Snavely, and David G Lowe. Unsupervised learning of depth and ego-motion from video. In *CVPR*, page 7, 2017.

[74] Karel Zuiderveld. Contrast limited adaptive histogram equalization. *Graphics gems*, pages 474–485, 1994.