Object Detection on Three Dimensional Reconstructions of Coral Reefs

by

Sushanth Kathirvelu

(Under the Direction of Suchendra M. Bhandarkar)

Abstract

Coral Reefs are structurally complex ecosystems that have been severely affected by natural and anthropogenic stressors. Consequently, there is a need for rapid and accurate ecological assessment of coral reefs, but current approaches entail time-consuming manual data acquisition and analysis. This thesis proposes an algorithmic approach to identify coral entities in 3D ecosystem maps as distinct 3D objects. Given 2D region proposals in an RGB image and an annotated 3D reconstructed surface mesh, our method generates a 3D region proposal for every 2D region proposal using the intrinsic and extrinsic camera parameters associated with the RGB images. The performance of the proposed method on coral reef survey images is compared against the performance of a state of the art end-to-end learning-based approach called the *Frustum PointNet* and the results are tabulated and compared. The average precision values obtained using the proposed method and *Frustum PointNet* are observed to be dependent on the underlying coral class.

Index words: 3D object detection, region proposal networks, 2D object detection, multiview segmentation, 3D point clouds, 3D reconstruction.

Object Detection on Three Dimensional Reconstructions of Coral Reefs

by

Sushanth Kathirvelu

B.E., Sri Shakthi Institute of Engineering and Technology, India, 2015

A Thesis Submitted to the Graduate Faculty of the

University of Georgia in Partial Fulfillment of the Requirements for the Degree.

Master of Science

Athens, Georgia

2021

Object Detection on Three Dimensional Reconstructions of Coral Reefs

by

Sushanth Kathirvelu

Major Professor: Suchendra M. Bhandarkar

Committee: Brian M. Hopkinson

Sheng Li

Electronic Version Approved:

Ron Walcott

Dean of the Graduate School

The University of Georgia

August 2021

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Coral reefs line many tropical coastlines, offering valuable ecosystem services including coastal protection and fisheries while also serving as hotspots of marine biodiversity. However, coral reefs are being increasingly threatened by both natural and anthropogenic stressors across the globe. The combination of natural and anthropogenic stressors including climate change, ocean acidification, sea-level rise, pollutant runoff, and overfishing (Anthony, 2016; Hoegh-Guldberg et al., 2007), have caused rapid deterioration of coral reefs worldwide over the past three decades (Beijbom et al., 2012) making it imperative to map and monitor the health of coral reef ecosystems on a global scale.

While coral reefs are of great cultural and scientific importance, their structural complexity and biodiversity makes them technically challenging to monitor and study. Most current approaches to monitor and study coral reef ecosystems involve time-consuming manual analysis either during a dive survey or on data collected during the survey by domain experts. In particular, accurate and repeatable mapping of coral reef ecosystems is especially challenging. Current mapping approaches include manual mapping performed by human divers which is time-consuming, and satellite and aerial imaging which limits the monitoring to shallow portions of coral reefs since seawater absorbs light strongly (Hedley et al., 2016).

More recently, ecologists and computer scientists have begun to employ techniques from computer vision and robotics such as *Structure-from-Motion* (SfM) and *Simultaneous Localization and Mapping* (SLAM) to produce accurate, high-resolution maps of coral reefs from images (Leon et al., 2015; Storlazzi

et al., 2016). Commercial and open-source SfM/SLAM tools are capable, in most cases, of producing highly accurate 3D reconstructions that represent the underlying coral reef structure in the form of triangulated surface meshes or point clouds. These 3D representations can be analyzed to assess structural complexity which is a key feature of coral reefs, contributing to biodiversity and coastal protection. Additionally, the RGB color information can be overlaid upon these 3D reconstructions to produce a textured 3D reconstruction of the underlying coral reef. Ecologists have manually analyzed these textured 3D reconstructions to assess coral disease prevalence (Burns et al., 2016) and coral and algal abundance (Couch et al., 2021), but these manual analyses are time consuming.

The advent of *Deep Neural Networks* (DNNs) and *Convolutional Neural Networks* (CNNs) has greatly improved the accuracy and ease of use of computer vision tools for automated image analysis (LeCun et al., 2015). These tools have also shown promise in various ecological applications (Brodrick et al., 2019), including automated analysis of corals, algae, and substrates on coral reefs (Williams et al., 2019). Previous notable works on automated mapping and monitoring of coral reefs have focused on integrating advances in CNNs and 3D mapping to automatically generate semantically segmented 3D maps of the coral reef ecosystem (Hopkinson et al., 2020; King et al., 2019). These semantically segmented 3D maps generated using SfM-based reconstruction and CNN-based image classification provide ecologically important metrics including the total surface area occupied by each coral species and their spatial distribution within the reef ecosystem. However, these semantically segmented 3D maps do not explicitly delineate the *individual* members or entities belonging to a coral species.

In chapter 3, we propose a method to identify individual entities within the coral reef ecosystem, providing data necessary to address critical questions in population and community ecology where individuals are the fundamental unit. The identification, enumeration and localization of these individual coral entities provide critical insights into the underlying population dynamics including issues such as symbiosis, predation and competition.

The proposed method serves as one component of a processing pipeline in which images of a coral reef are converted into a multilayered 3D map of the underlying ecosystem. First, an SfM- or SLAM-based

approach is used to create a 3D structural representation of the underlying coral reef ecosystem and to determine the geometric relationships between the inferred 3D structure and source images. Next, the semantic information, in the form of species or substrate classes, is layered upon the structural representation using multiview CNN-based classification (Hopkinson et al., 2020). Finally, individuals are delineated within the semantically segmented 3D map using the method described in the chapter. To identify discrete individuals within a 3D surface mesh, we propose to apply 2D object detection algorithms to the input RGB images of the coral reef ecosystem. The 2D bounding boxes of the detected objects are then back-projected onto the 3D mesh reconstruction to identify individuals in the 3D annotated reconstruction. The proposed method is designed to leverage the recent advances in 2D object detectors designed for conventional RGB images and semantically segmented 3D reconstructions generated using state-of-the-art SfM/SLAM algorithms and multiview CNN-based classifiers to detect and delineate individual coral entities as distinct objects in 3D.

In chapter 4 we compare the results of our proposed algorithmic method that detects 3D objects from RGB images to an existing state of the art end-to-end learning-based approach for 3D object detection from point clouds termed as the *Frustum PointNet* (Qi et al., 2018; Wang & Jia, 2019). The average precision values are observed to be dependent on the underlying coral class. In chapter 5 we conclude the thesis while outlining directions for future work.

Chapter 2

Materials and Methods

2.1 Pinhole Camera

Understanding the geometrical model of the camera projection serves as the core idea for the thesis. In our work, we use the pinhole camera model (Zhang, 2000) with a distortion factor. A Pinhole Camera model is defined using a mathematical relationship $(u, v) = f(X, Y, Z)$ that explains how a point in a 3D space is projected onto the image plane. We can invert this model to also back-project an image pixel to the 3D world space.

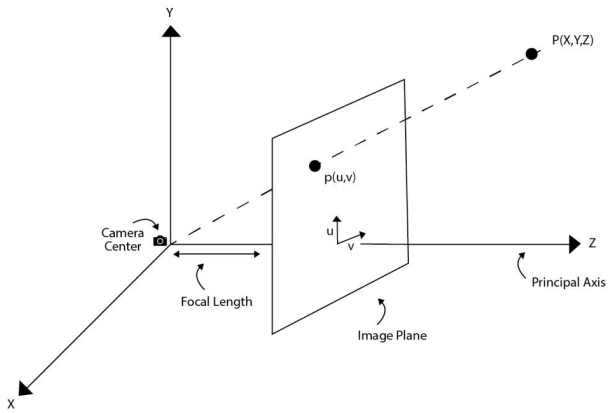

Figure 2.1: Pinhole Camera Model

In a pinhole camera model the camera coordinate reference frame and world coordinate reference frame are related by rotation and translation. The mathematical function that describes projection of 3D world points to 2D image plane is given by

$$\mathbf{p} = \mathbf{K} \times [\mathbf{R}|\mathbf{t}] \times \mathbf{P} \tag{2.1}$$

Where **p** is the pixel point (u, v) in the image plane, **K** is the camera intrinsic matrix that represents the camera calibration parameters such as its focal length and optical center coordinates and $[\mathbf{R}|\mathbf{t}]$ is the camera extrinsic matrix representing the location and orientation parameters of the camera in the 3D scene. The parameters**R** and **t** in the extrinsic camera matrix represent a 3×3 rotation matrix and a 3×1 translation matrix (vector) respectively. **P** represents the 3D point (X, Y, Z) expressed in the world coordinate reference frame.

Eq. (2.1) can be expanded as

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & u_0 \\ 0 & f_y & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \tag{2.2}$$

Where $(u_0, v_0)$is the principal point (i.e., the optical center) of the camera lens, f_x and f_y represents the focal length in pixels which is a measure of the distance between the optical center of the camera lens and the image plane.

2.2 Camera Distortion

Equations (2.1) and (2.2) do not account for lens distortion while computing the pixel coordinates in the image plane. In practice, a camera lens introduces some distortion in the resulting images. To accurately

represent a camera model, the radial and tangential lens distortion terms are incorporated within the pinhole camera model when estimating the pixel positions in the camera coordinate reference frame.

Radial Distortion Radial distortion occurs when the light rays bend unevenly when passing through the lens, causing straight lines to appear curved in the image. In a typical lens, the radial distortion is usually 0 at its optical center and increases as we move radially away from its optical center. Since the light rays farther from the optical center of the lens are curved more than the ones that are closer; radial distortion is most evident near the edge of the images.

Given the pixel coordinates (u, v) obtained via the simple pinhole camera model, radial distortion can be accounted for as follows:

$$\begin{aligned} u_{d_1} &= u(1 + k_1 r^2 + k_2 r^4 + k_3 r^6) \\ v_{d_1} &= v(1 + k_1 r^2 + k_2 r^4 + k_3 r^6) \\ &\text{where } r = \sqrt{u^2 + v^2} \end{aligned} \tag{2.3}$$

Here, (u_{d_1}, v_{d_1}) are the pixel coordinates obtained after taking radial distortion into account and k_1, k_2 and k_3 are radial distortion coefficients of the lens.

Tangential Distortion Tangential distortion typically arises due to a manufacturing defect in the camera causing the camera lens to be not aligned parallel to the image plane, i.e., causing the optical axis of the camera lens to be not perfectly perpendicular to the image plane. This type of distortion causes the image to look tilted, making some objects in the image seem farther than they actually are.

Given the pixel coordinates (u, v) obtained via the simple pinhole camera model, tangential distortion can be accounted for as follows:

$$\begin{aligned} u_{d_2} &= u + [2p_1 uv + p_2(r^2 + 2u^2)] \\ v_{d_2} &= v + [p_1(r^2 + 2v^2) + 2p_2 uv] \\ &\text{where } r = \sqrt{u^2 + v^2} \end{aligned} \tag{2.4}$$

Here, (u_{d_2}, v_{d_2}) are the pixel coordinates obtained after taking tangential distortion into account and p_1 and p_2 are the tangential distortion coefficients of the lens.

We can compute the corrected pixel coordinates (u_d, v_d) in the camera coordinate reference frame after accounting for both the radial and tangential distortion using equations (2.3) and (2.4) as follows:

$$\begin{aligned} u_d &= u(1 + k_1 r^2 + k_2 r^4 + k_3 r^6) + [2p_1 uv + p_2(r^2 + 2u^2)] \\ v_d &= v(1 + k_1 r^2 + k_2 r^4 + k_3 r^6) + [p_1(r^2 + 2v^2) + 2p_2 uv] \\ &\text{where } r = \sqrt{u^2 + v^2} \end{aligned} \tag{2.5}$$

2.3 Ray Tracing

A ray-tracing approach is used, in our proposed 3D object detection algorithm, to determine for a given point or pixel in an image its corresponding location on a 3D surface mesh. Given the origin of the camera coordinate reference frame and a pixel location in the image, both expressed in the world coordinate reference frame, we can construct a ray emanating from the origin of the camera coordinate reference frame and passing through the image pixel. Using the Möller-Trumbore ray tracing algorithm (Möller & Trumbore, 1997), one can determine whether this ray intersects a triangular facet within the 3D mesh reconstruction of the coral reef and, if so, compute the coordinates of the point of intersection within the interior of the triangular facet. The 3D coordinates of the point of intersection specify the point on the 3D coral reef surface (approximated by the 3D triangulated surface mesh) that the image pixel represents.

Any point $\mathbf{p}$ on the ray described above can be represented using the origin $\mathbf{o}$ of the ray, the direction vector $\mathbf{d}$ and the distance t between the origin and the point $\mathbf{p}$ as:

$$\mathbf{p} = \mathbf{o} + t\mathbf{d} \tag{2.6}$$

Likewise, any point $\mathbf{p}$ within a triangle can be defined in terms of its vertices $\mathbf{p}_0$, $\mathbf{p}_1$, $\mathbf{p}_2$ and barycentric coordinates (u, v)

$$\mathbf{p} = (1 - u - v)\mathbf{p}_0 + u\mathbf{p}_1 + v\mathbf{p}_2 \tag{2.7}$$

where (u, v) satisfy the conditions $u \geq 0$, $v \geq 0$ and $u + v \leq 1$ (Möller & Trumbore, 1997). From equations (2.6) and (2.7), for the ray to intersect the triangle, the following must hold:

$$\mathbf{o} + t\,\mathbf{d} = (1 - u - v)\mathbf{p}_0 + u\mathbf{p}_1 + v\mathbf{p}_2 \tag{2.8}$$

Equation (2.8) can be rearranged to solve for t and (u, v) to yield the point of intersection between the ray and the triangular facet and also verify that the point of intersection does lie within the interior of the triangular facet as follows:

$$\begin{bmatrix} -\mathbf{d} & (\mathbf{p}_1 - \mathbf{p}_0) & (\mathbf{p}_2 - \mathbf{p}_0) \end{bmatrix} \times \begin{bmatrix} t \\ u \\ v \end{bmatrix} = \mathbf{o} - \mathbf{p}_0 \tag{2.9}$$

Given the image pixel and the camera origin in the camera coordinate frame, the ray-triangle intersection procedure (Möller & Trumbore, 1997) and enables the back projection of the image pixels onto the 3D triangular mesh.

Chapter 3

Object Detection in 3D Coral Ecosystem Maps from Multiple Image Sequences[1]

[1]Sushanth Kathirvelu*, Suchendra Bhandarkar, Brian Hopkinson. Submitted to the Workshop - Computer Vision in the Ocean (Workshop @ International Conference on Computer Vision, 2021).

3.1 Abstract

Coral reefs are biologically diverse and structurally complex ecosystems that have been severely affected by natural and anthropogenic stressors. Consequently, there is a need for rapid and accurate ecological assessment of coral reefs, but current approaches entail time-consuming manual data acquisition and analysis. In this chapter, we propose a method to identify and localize individual entities within the coral reef ecosystem as distinct 3D objects and assess its performance. Given 2D region proposals in an RGB image, our method generates, for each 2D region proposal, a 3D region proposal based on an existing annotated 3D reef reconstruction and the intrinsic and extrinsic camera parameters associated with the RGB image. The annotated 3D reef reconstruction is generated using a commercial Structure-from-Motion (SfM) software and a previously designed multiview convolutional neural network (CNN) for 3D semantic segmentation. As individual coral reef entities are often viewed in multiple images, a 3D bounding-box merging strategy coupled with a predefined overlap criterion are used to combine multiple 3D region proposals into a single 3D object prediction for the purpose of classification and localization. Experimental results on coral reef survey images show the efficacy of the proposed method.

Keywords: 3D object detection, 2D region proposal networks, 2D object detection, multiview segmentation

3.2 Introduction

Coral reefs line many tropical coastlines, offering valuable ecosystem services including coastal protection and fisheries while also serving as hotspots of marine biodiversity. However, coral reefs are being increasingly threatened by both natural and anthropogenic stressors across the globe. The combination of natural and anthropogenic stressors including climate change, ocean acidification, sea-level rise, pollutant runoff, and overfishing (Anthony, 2016; Hoegh-Guldberg et al., 2007), have caused rapid deterioration of coral reefs worldwide over the past three decades (Beijbom et al., 2012) making it imperative to map and monitor the health of coral reef ecosystems on a global scale.

While coral reefs are of great cultural and scientific importance, their structural complexity and biodiversity makes them technically challenging to monitor and study. Most current approaches to monitor and study coral reef ecosystems involve time-consuming manual analysis either during a dive survey or on data collected during the survey by domain experts. In particular, accurate and repeatable mapping of coral reef ecosystems is especially challenging. Current mapping approaches include manual mapping performed by human divers which is time-consuming, and satellite and aerial imaging which limits the monitoring to shallow portions of coral reefs since seawater absorbs light strongly (Hedley et al., 2016).

More recently, ecologists and computer scientists have begun to employ techniques from computer vision and robotics such as *Structure-from-Motion* (SfM) and *Simultaneous Localization and Mapping* (SLAM) to produce accurate, high-resolution maps of coral reefs from images (Leon et al., 2015; Storlazzi et al., 2016). Commercial and open-source SfM/SLAM tools are capable, in most cases, of producing highly accurate 3D reconstructions that represent the underlying coral reef structure in the form of triangulated surface meshes or point clouds. These 3D representations can be analyzed to assess structural complexity which is a key feature of coral reefs, contributing to biodiversity and coastal protection. Additionally, the RGB color information can be overlaid upon these 3D reconstructions to produce a textured 3D reconstruction of the underlying coral reef. Ecologists have manually analyzed these textured 3D reconstructions to assess coral disease prevalence (Burns et al., 2016) and coral and algal abundance (Couch et al., 2021), but these manual analyses are time consuming.

The advent of *Deep Neural Networks* (DNNs) and *Convolutional Neural Networks* (CNNs) has greatly improved the accuracy and ease of use of computer vision tools for automated image analysis (LeCun et al., 2015). These tools have also shown promise in various ecological applications (Brodrick et al., 2019), including automated analysis of corals, algae, and substrates on coral reefs (Williams et al., 2019). Previous notable works on automated mapping and monitoring of coral reefs have focused on integrating advances in CNNs and 3D mapping to automatically generate semantically segmented 3D maps of the coral reef ecosystem (Hopkinson et al., 2020; King et al., 2019). These semantically segmented 3D maps generated using SfM-based reconstruction and CNN-based image classification provide ecologically im-

portant metrics including the total surface area occupied by each coral species and their spatial distribution within the reef ecosystem. However, these semantically segmented 3D maps do not explicitly delineate the *individual* members or entities belonging to a coral species.

In this chapter, we propose a method to identify individual entities within the coral reef ecosystem, providing data necessary to address critical questions in population and community ecology where individuals are the fundamental unit. The identification, enumeration and localization of these individual coral entities provide critical insights into the underlying population dynamics including issues such as symbiosis, predation and competition.

The proposed method serves as one component of a processing pipeline in which images of a coral reef are converted into a multilayered 3D map of the underlying ecosystem. First, an SfM- or SLAM-based approach is used to create a 3D structural representation of the underlying coral reef ecosystem and to determine the geometric relationships between the inferred 3D structure and source images. Next, the semantic information, in the form of species or substrate classes, is layered upon the structural representation using multiview CNN-based classification (Hopkinson et al., 2020). Finally, individuals are delineated within the semantically segmented 3D map using the method described in this chapter. To identify discrete individuals within a 3D surface mesh, we propose to apply 2D object detection algorithms to the input RGB images of the coral reef ecosystem. The 2D bounding boxes of the detected objects are then back-projected onto the 3D mesh reconstruction to identify individuals in the 3D annotated reconstruction. The proposed method is designed to leverage the recent advances in 2D object detectors designed for conventional RGB images and semantically segmented 3D reconstructions generated using state-of-the-art SfM/SLAM algorithms and multiview CNN-based classifiers to detect and delineate individual coral entities as distinct objects in 3D.

3.3 Related Work

In this chapter, we cast the problem of detection of individual coral entities in a 3D coral reef reconstruction as one of 3D object detection. 2D object detectors for conventional RGB images have attained a

state of maturity and have been shown to achieve high classification and localization accuracy in a variety of applications. In contrast, 3D object detectors are still evolving and although rapid progress is being made, the performance of 3D object detectors is still observed to lag that of its 2D counterparts in terms of accuracy and robustness.

3.3.1 2D Object Detectors

Mature 2D object detectors employ CNNs either in a single-stage or two-stage design. In the single-stage design, object detection is cast as a regression problem that takes a 2D RGB image as input and learns the class probabilities and 2D bounding box parameters of the objects in the image. The single-stage object detector predicts both, a class label and image location for each object in the image after a single pass through the CNN. In contrast, in the two-stage design, the input image is first fed to a *Region Proposal Network* (RPN) that generates the regions of interest (ROIs) within the image. The ROIs are then fed to the second stage comprising of a CNN that performs object classification and bounding box regression. *You Only Look Once* (YOLO) (Redmon et al., 2016), *Single Shot Multibox Detector* (SSD) (Liu et al., 2016) and *RetinaNet* (T.-Y. Lin et al., 2017) are examples of popular single-stage object detectors whereas *Faster-RCNN* (Ren et al., 2015) and *Mask-RCNN* (He et al., 2017) are examples of widely used two-stage designs for object detection and pixel-level segmentation respectively.

In general, two-stage detectors, such as those from the RCNN family, are observed to achieve higher accuracy than the single-stage detectors, such as those from the YOLO family that are faster, computationally efficient, architecturally simpler and better suited for implementation on mobile devices. The more recent *RetinaNet* (T.-Y. Lin et al., 2017), though a single-stage detector, has been observed to yield accuracy comparable to that of two-stage alternatives, an advance enabled by modifying the loss function to down-weight the numerous easily-classified background examples. Both, the single-stage and two-stage object detectors, make flexible use of standard CNN architectures, such as *ResNet* (He et al., 2016), *Inception* (Szegedy et al., 2015) and *VGG* (Simonyan & Zisserman, 2015), as backbones for feature extraction where the CNN backbone can be swapped out to address various speed vs. accuracy tradeoffs.

3.3.2 3D Object Detectors

3D object detection has been receiving increased attention in the computer vision and machine learning communities and methods are evolving rapidly. We focus on 3D object detection from RGB images, which is most closely aligned with our work, as opposed to 3D object detection exclusively from 3D point cloud data obtained via LiDAR or structured light sensors. One family of 3D object detectors uses shape priors to aid the inference of 3D objects from 2D projections (K. Li et al., 2018; K. Li et al., 2020; C.-H. Lin et al., 2019). For example, the *FroDO* scheme (K. Li et al., 2020) merges multiple 2D object predictions from video streams to generate a 3D bounding box and a multi-modal latent object representation (i.e., point cloud, signed-distance function etc.) for each detected object. Neural networks that translate the compressed, latent object representation into 3D objects are trained for each object type using ground truth examples. The latent representation for each object is continuously refined through repeated observations, using geometric, photometric, and silhoutette losses.

Another family of detectors employs a combination of RGB images and point cloud data, using 2D object detectors to identify projections of 3D objects and back-projecting the 2D bounding box to determine portions of the point cloud belonging to each object. Both, the 2D RGB image data and point cloud structural features are used to predict the object class and the full extent of the object as objects may be partially occluded (Qi et al., 2018; Wang & Jia, 2019). Our approach shares similarities with this genre of 3D object detectors in back-projecting 2D object projections into 3D, but our approach differs from this genre in that it assumes an existing semantically segmented 3D mesh, which can be leveraged to improve 3D object predictions. Furthermore, the objects in our domain differ substantially from those encountered in urban/suburban environments, such as vehicles, people, buildings etc., that most existing approaches deal with (Qi et al., 2018; Wang & Jia, 2019). In our case, organisms on a coral reef are typically clonal with extended and irregular shapes and often only a small fraction of an individual object (e.g. a large coral) is visible in a single 2D image. The differences in object structure and size make the use of 3D shape priors, in our case, very challenging.

3.3.3 Ecological Applications of Object Detection

In ecological applications, 2D object detectors have begun to be employed to count plants and animals in images and track animals across video frames. Weinstein et al., 2019 train a *RetinaNet* model to detect tree crowns in aerial imagery and achieve good overall performance, although small trees are observed to be difficult to distinguish from large bushes. Jalal et al., 2020 detect and classify fish in individual video frames by merging information from a YOLO object detector and a custom object detector that proposes ROIs based on optical flow (under the assumption that the fish are moving) and classifies the ROIs using a *ResNet* architecture. 2D object detectors have also been employed to aid fish tracking as demonstrated in a trial experiment in an Australian estuary (Lopez-Marcano et al., 2021). In this case, a *Mask-RCNN* is employed to first detect fish in video streams and once detected, an object tracker is initialized to follow their movements. Our work contributes to the continuing application of object detection to ecological settings, expanding the scope of object detection from 2D to 3D and employing existing annotated 3D reconstructions to overcome the challenges arising from the size and irregularity of individuals (i.e., objects) in coral reefs.

3.4 Materials and Methods

3.4.1 Underwater Image Data Acquisition

The underwater coral images used in this work were manually collected by a team of divers from coral reefs off the Florida Keys, USA using a stereoscopic video camera. An underwater stereoscopic camera rig, comprising of a *GoPro Dual Hero* camera system, was used to capture the underwater video data while swimming over sections of the coral reef in a serpentine (i.e., lawn mower) pattern, thereby enabling the capture of a complete section of the seafloor.

A subset of the acquired images from the above image data set was annotated to provide ground truth bounding boxes for individual entities on the coral reef. During the annotation process, an individual coral

entity in a 2D image was delineated as an independent object and assigned to one of the following seven classes: (1) *Acropora palmata*, (2) *Orbicella*, (3) *Siderastrea*, (4) *Porites astreoides*, (5) *Gorgonia ventalina*, (6) *Sea Rods* and (7) *Antillogorgia*. The first four classes, i.e., *A. palmata*, *Orbicella*, *Siderastrea*, and *P. astreoides*, represent the different coral species commonly found on reefs in the Florida Keys. The remaining single-species class, i.e., *G. ventalina*, represents the common sea fan. The remainder of the classes, i.e., *Sea Rods* and *Antillogorgia* are multi-species classes or general classes.

3.4.2 3D Coral Reef Reconstruction

A 3D reconstruction of the imaged coral reef was generated using an off-the-shelf commercial SfM software (*Agisoft Photoscan 1.4.3* now *Metashape*) from the images acquired manually (Hopkinson et al., 2020). The 3D reconstruction of each imaged coral reef was performed using $\approx$ 100 - 5000 images producing a dense triangular mesh comprising of $\approx$ 200K-300K triangular facets, which can be regarded as a discretized approximate representation of the 3D surface of the imaged coral reef. A texture map derived from the 2D RGB images is overlaid on the triangular mesh for the purpose of visualization as depicted in the first stage of Figure 3.1. The 3D camera transformation matrices and calibration parameters are obtained from the SfM software as part of the 3D reconstruction procedure.

3.4.3 Pixel Back Projection

The geometrical model of camera projection is central to the 3D object detection procedure. The pinhole camera model (Zhang, 2000) provides a simple mathematical relationship $(u, v) = f(x, y, z)$ between a point (x, y, z) in the 3D world and its projection (u, v) onto the image plane. Likewise, the inverse of the camera projection model can be used to establish the relationship between a pixel (u, v) in the image plane and its three-dimensional world coordinates (x, y, z). The mathematical function f that describes the projection of the 3D world coordinates (x, y, z) to image plane coordinates (u, v) can be expressed as:

$$\mathbf{p} = \mathbf{K} \times [\mathbf{R}|\mathbf{t}] \times \mathbf{P} \tag{3.1}$$

where $\mathbf{p} = (u, v)$ is the pixel in the image plane, $\mathbf{K}$ is the camera intrinsic matrix that encodes the calibration parameters of the camera, such as its focal length and the optical center, and $[\mathbf{R}|\mathbf{t}]$ is the camera extrinsic matrix that encodes the location and orientation parameters of the camera (i.e., camera viewpoint parameters) in the 3D world coordinate frame of reference. The submatrices $\mathbf{R}$ and $\mathbf{t}$ in the extrinsic matrix represent a 3×3 rotation matrix and a 3×1 translation matrix (vector) respectively. $\mathbf{P} = (x, y, z)$ represents the 3D point in the world coordinate reference frame. The standard camera intrinsic matrix $\mathbf{K}$ does not account for radial and tangential distortion introduced by the camera lens. The observed pixel coordinates $\mathbf{p}_d = (u_d, v_d)$ that take into account the camera lens distortion are given as:

$$\mathbf{p}_d = \mathbf{R}_d \times \mathbf{p} + \mathbf{t}_d \tag{3.2}$$

where $\mathbf{R}_d$ is radial distortion matrix and $\mathbf{t}_d$ is the tangential distortion vector associated with the camera lens (Zhang, 2000). Equation (3.2) can be used to recover the undistorted pixel coordinates $\mathbf{p} = (u, v)$ from the observed (distorted) pixel coordinates $\mathbf{p}_d = (u_d, v_d)$ (Zhang, 2000).

3.4.4 Ray-Triangle Intersection Determination

A ray tracing approach is used in our 3D object detection algorithm to determine the location on a 3D mesh corresponding to a point in an image. Given the origin of the camera coordinate reference frame and a pixel location in the image, both expressed in the world coordinate reference frame, one can construct a ray emanating from the origin of the camera coordinate reference frame and passing through the image pixel. Using the Möller-Trumbore ray tracing algorithm (Möller & Trumbore, 1997), one can determine whether this ray intersects a triangular facet within the 3D mesh reconstruction of the coral reef and if so, compute the coordinates of the point of intersection within the interior of the triangular facet. The 3D coordinates of the point of intersection specify the point on the 3D coral reef surface (approximated by the 3D triangulated surface mesh) that the image pixel represents.

3.5 Description of the Proposed Method

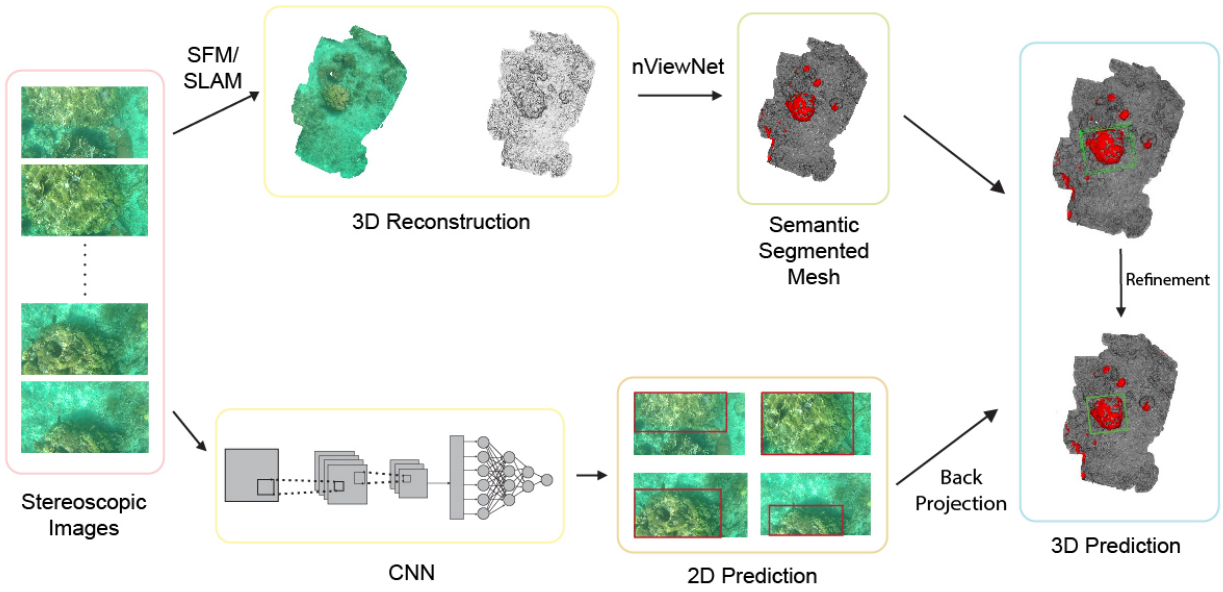

Figure 3.1: **Overview of Proposed Approach:** A 3D surface mesh is generated from RGB images using SfM/SLAM and semantically segmented using *nViewNet*. 2D object detectors are employed on the RGB images and the predicted 2D bounding boxes are back-projected on the 3D surface mesh. The resulting 3D bounding boxes are refined to improve 3D object detection accuracy.

The stereoscopic coral image data are acquired as described in Section 3.4.1 and the 3D triangulated mesh are generated using commercial SfM software as described in Section 3.4.2. The *annotated* 3D reef reconstruction is generated using a multiview convolutional neural network (CNN) for 3D semantic segmentation termed as *nViewNet* (King et al., 2019). The *nViewNet* takes as input the various 2D RGB images of the coral reef acquired from multiple viewpoints and the SfM-generated mesh reconstruction. *nViewNet* classifies each triangular facet in the mesh based on the consensus class label of all patches from all of the 2D RGB images that are back-projected onto that mesh facet resulting in a 3D semantic segmentation of the underlying coral reef (King et al., 2019).

We leverage existing 2D object detectors to identify and localize the individual coral entities on the reef surface in the 2D RGB images. The output of the object detector is a prediction of the class (category) and location of an individual coral entity (i.e., object). The object location is predicted as a bounding box

that is represented by the coordinates of its corner points. We choose two well known object detectors, *RetinaNet* (T.-Y. Lin et al., 2017) which is a single-stage detector and *Faster-RCNN* (Ren et al., 2015) which is a two-stage detector from the RCNN family that is optimized for speed and architectural simplicity. It should be noted that both detectors are considered among the best performing in their respective categories.

The basic principle underlying the proposed method is to leverage knowledge of the intrinsic and extrinsic camera matrices acquired during the construction of the 3D triangulated surface mesh and the class annotations of the mesh facets obtained via 3D semantic segmentation of the surface mesh to predict the class labels and 3D locations of individual coral entities. The 2D bounding boxes predicted by the 2D object detectors are back-projected onto the 3D surface mesh as described in Section 3.4.3. The pixels representing each corner of the predicted 2D bounding box are back projected into the 3D camera coordinate reference frame using the camera intrinsic matrix $\mathbf{K}$ after accounting for both radial and tangential non-linear distortions (Equation (3.2)). The camera extrinsic matrix $[\mathbf{R}|\mathbf{t}]$ is then used to transform the coordinate values from the camera coordinate frame to the world coordinate frame. The 3D mesh element that the pixel represents is determined by constructing a ray emanating from the optical center of the camera and passing through the pixel coordinates (both expressed in terms of their world coordinates) and computing the point of intersection of the ray with the surface mesh as described in Section 3.4.4. The minimum and maximum x, y and z values of the points of intersections (corresponding to the corner pixels of the 2D bounding box) are used to characterize the 3D bounding box on the surface mesh.

Since the individual coral entities on the 3D surface mesh are typically viewed in multiple images from varying viewpoints, multiple 3D bounding boxes are created for the same object when the 2D bounding boxes detected in the RGB images are back-projected on the 3D surface mesh. If the intersection volume of two bounding boxes (characterized by their minimum and maximum x, y and z in world coordinates) that represent the same object class in the reconstructed 3D surface mesh is over a predefined threshold τ_{merge}, we merge them into a single bounding box by updating the minimum and maximum x, y and z values. We term this scheme as the base method for 3D bounding box prediction/estimation. It should be

noted that the base method estimates a 3D bounding box that is significantly larger than the underlying object of interest. Consequently, we refine the merged 3D bounding box to yield a more compact 3D bounding box using the following 2-step procedure:

Step 1: Initial Refinement. We refine our results by exploiting the 3D semantic segmentation of the surface mesh obtained by employing *nViewNet* (King et al., 2019). Note that *nViewNet* provides a class label for each mesh facet via consensus of class labels of all pixels from multiple semantically segmented 2D images that are back-projected onto that facet. In this step we exploit information from multiple views of the same object that each mesh facet represents. For each 3D bounding box predicted by the base method we identify all the mesh elements with the label corresponding to predicted object. These mesh elements effectively comprise the object of interest. The minimum and maximum x, y, and z coordinates of the object mesh elements are used as the refined 3D bounding box.

Step 2: Final Refinement. Given the class labels for each mesh facet and the 3D bounding boxes obtained via back projection, we determine all the mesh facets within the 3D bounding box that match the object class label. Since each triangular mesh facet shares its edges with other triangular facets, we can construct a connectivity graph G where the nodes of the graph represent mesh facets with the given object class label within the 3D bounding box and there exists an edge between two nodes if the corresponding facets share an edge in the mesh. We compute the connected components of G and obtain the refined 3D bounding box parameters by determining the minimum and maximum x, y, and z coordinates from the largest connected component of G (i.e., one with the maximum number of facets).

3.6 Performance Evaluation Metrics

For the purpose of evaluation, we project the predicted 3D bounding boxes back to the 2D images in the test set for comparison with manually annotated 2D bounding boxes. We take this approach as we lack 3D ground truth bounding boxes, which are difficult to generate.

We use the *Intersection-over-Union* (IoU) measure to determine the classification accuracy and localization accuracy of the proposed 3D object detection scheme. The IoU measure computes the degree

of overlap between the predicted and the ground truth 2D bounding boxes for all predicted bounding boxes. If $IoU \geq \tau_{IoU}$, we regard the detected object as a *true positive* (TP) otherwise we regard it as a *false positive* (FP) where τ_{IoU} is a predefined threshold. We regard a bounding box present in the ground truth but not detected by the proposed scheme as a *false negative* (FN). The number of true positives N_{TP}, false positives N_{TP} and false negatives N_{FN} are computed for all 3D object instances in the surface mesh belonging to a given class and across all the 2D images in which its projections appear. We compute the precision (PR) and recall (RC) values as,

$$\begin{aligned} PR &= \frac{N_{TP}}{N_{TP} + N_{FP}} \\ RC &= \frac{N_{TP}}{N_{TP} + N_{FN}} \end{aligned} \tag{3.3}$$

For each object class, we compute the PR and RC values for different values of the IoU threshold τ_{IoU} generating a precision-recall (PR-RC) curve. The average precision (AP) for each object class is computed by determining the area under the PR-RC curve. In our case, we compute the PR values for a set of 11 discrete RC values $\{0.0, 0.1, 0.2, \ldots, 0.9, 1.0\}$ and compute the AP value by averaging the corresponding PR values as follows:

$$AP = \frac{1}{11} \sum_{i=1}^{11} PR(RC_i) \tag{3.4}$$

The *mean average precision* (mAP) measure is computed as the mean of the AP values over all classes. The mAP measure is indicative of the overall performance (i.e., accuracy) of the proposed scheme.

3.7 Experiments and Results

We assessed the performance of the various stages in the proposed scheme for detection of individual 3D coral entities in the reconstructed 3D surface mesh of the underlying reef using the acquired image data described in Section 3.4.1.

The coral data set used in this chapter comprises of three 3D reconstructed surface meshes. Each mesh is divided into two portions; the RGB images corresponding to one portion of the mesh are used to train the 2D object detectors, whereas the images that correspond to the other portion are used to test the 3D object predictions. After the split, the data set contains 532 training and 113 test RGB images for the 7 object categories (Section 3.4.1).

We first trained two 2D object detectors, the single-stage *RetinaNet* and the two-stage *Faster-RCNN*, both with a *ResNet-50* backbone, for coral object detection in the input RGB images. Table 3.1 summarizes the performance of the *RetinaNet* and *Faster-RCNN* 2D object detectors in terms of the mAP measured over a range of IoU threshold (i.e., τ_{IoU}) values when trained and tested on our coral image data set. Both detectors perform reasonably well considering the small training data set size and the challenging characteristics of coral objects, arising primarily from their clonal nature resulting in individuals of varying shapes and sizes, and in some cases, ambiguous boundaries between individuals. The 2D object detection performance was relatively consistent for $\tau_{IoU} \leq 50$, but performance declined rapidly at more demanding IoU threshold values ($\tau_{IoU} > 50$).

Table 3.1: **2D Object Detection Results:** mAP of the 2D object detector computed over a range of IoU threshold (τ_{IoU}) values where mAP_i denotes the mAP for an IoU threshold value of $\tau_{IoU} = i\%$.

Model	Batch Size	mAP_{20}	mAP_{30}	mAP_{40}	mAP_{50}	mAP_{60}	mAP_{70}	mAP_{80}
Faster-RCNN	32	59.91	59.10	57.42	54.23	50.34	41.29	23.39
RetinaNet	32	66.69	65.54	63.82	60.44	53.34	43.19	24.57

RetinaNet, was observed to consistently outperform *Faster-RCNN* in the 2D object detection task by a moderate margin in terms of both, mAP (Table 3.1) and AP for most classes (Table 3.3). The detectors performed extremely well on some classes (e.g. *A. palmata*, *Antillogorgia*) but poorly on others (e.g. *Orbicella*, Sea Rods), reflecting both, the size of the training data sets as well as the homogeneity (or lack thereof) of individuals within each class. For example, *A. palmata* had a relatively large training data set and individuals are relatively homogeneous in appearance contributing to high performance of the 2D

object detector, whereas Sea Rods had a smaller training data set and represent a mixed assemblage of species resulting in lower 2D object detector performance.

Table 3.2: **3D Object Detection Results:** The mean Average Precision (mAP) results for different 3D object detection models based on *Faster-RCNN* and *RetinaNet* 2D object detectors with IoU threshold $\tau_{IoU} = 50\%$ for 2D detectors and $\tau_{IoU} = 25\%$ for 3D detectors and varying values of τ_{merge} for the merging of overlapping 3D bounding boxes into a single 3D bounding box. mAP_{2D} is the mAP of the 2D detector whereas mAP_{3D0}, mAP_{3D1}, mAP_{3D2} denote the mAP values for the base 3D object detection method and with refinement steps 1 and 2 respectively.

Model	τ_{merge}	mAP_{2D}	mAP_{3D0}	mAP_{3D1}	mAP_{3D2}
Faster-RCNN	0.4	54.23	15.12	27.98	25.87
	0.6	54.23	15.65	29.16	27.03
	0.8	54.23	20.33	35.78	29.64
RetinaNet	0.4	60.44	7.61	18.01	22.67
	0.6	60.44	9.60	22.75	24.74
	0.8	60.44	14.76	27.91	24.23

The trained 2D object detectors were then used to make 3D object predictions on a test data set using the method described in Section 3.5. Three variants of the 3D object detection procedure were assessed at an IoU threshold value $\tau_{IoU} = 25\%$: the base method (3D0), the base method with the initial refinement (3D1), and the base method with the initial and final refinements (3D2). Three different values for the threshold, i.e., $\tau_{merge} = 0.4, 0.6$, and 0.8, were tested for the merging of overlapping 3D bounding boxes into a single 3D bounding box. The results in Table 3.2 show that the overall performance of the 3D object detector was substantially lower than that of the corresponding 2D detector in the case of both *Faster-RCNN* and *RetinaNet*. *Faster-RCNN* consistently outperformed *RetinaNet* in the 3D object detection task, in contrast to the 2D case. Analysis of the initial results with the base method revealed that the decreased performance of the 3D detector relative to its 2D counterpart was primarily due to the growth of the 3D bounding box size as multiple 3D bounding boxes from individual images were merged and secondarily due to inappropriate merging of distinct objects based on 3D bounding box overlap. Inappropriate merging was reduced by increasing the overlap threshold τ_{merge} for merging 3D bounding boxes (Table 3.2). Figure 3.2 illustrates the problem of bounding box enlargement for several target classes.

Table 3.3: **Per Class 3D Object Detection Results:** Average Precision (AP) results for different coral classes for IoU threshold $\tau_{IoU} = 25\%$ and varying values of τ_{merge} for the merging of overlapping 3D bounding boxes into a single 3D bounding box.

Models	τ_{merge}	Metric	*Acropora palmata*	*Orbicella*	*Siderastrea*	*Porites astreoides*	*Gorgonia ventalina*	*Sea Rods*	*Antillogorgia*
Faster-RCNN$_{2D}$	N/A	AP_{2D}	79.24	23.70	72.81	53.50	55.27	27.20	67.87
Faster-RCNN$_{3D}$	0.4	AP_{3D0}	11.18	4.69	69.33	0.70	0.76	0.43	18.76
		AP_{3D1}	11.24	8.33	85.00	48.65	15.24	6.88	20.49
		AP_{3D2}	16.74	8.33	67.9	48.65	5.38	3.36	30.70
Faster-RCNN$_{3D}$	0.6	AP_{3D0}	13.13	4.69	69.33	0.70	1.36	0.61	19.71
		AP_{3D1}	15.23	8.33	85.00	48.65	19.89	6.88	20.13
		AP_{3D2}	18.24	8.33	67.94	48.65	11.46	3.36	31.21
Faster-RCNN$_{3D}$	0.8	AP_{3D0}	28.88	15.00	69.33	1.16	3.52	0.93	23.48
		AP_{3D0}	32.14	26.67	85.00	48.65	24.01	6.44	27.55
		AP_{3D0}	30.14	10.42	67.94	48.65	13.09	3.38	33.86
RetinaNet$_{2D}$	N/A	AP_{2D}	87.43	46.16	63.51	62.95	61.59	36.00	65.45
RetinaNet$_{3D}$	0.4	AP_{3D0}	7.63	1.02	27.03	0.88	0.61	0.64	15.48
		AP_{3D1}	10.24	2.22	36.56	45.62	6.92	4.21	26.64
		AP_{3D2}	12.32	2.22	59.16	45.12	9.05	4.21	26.64
RetinaNet$_{3D}$	0.6	AP_{3D0}	15.77	1.02	27.73	1.72	2.48	1.55	16.93
		AP_{3D1}	20.89	2.70	37.26	51.03	21.11	7.43	18.86
		AP_{3D2}	22.26	4.22	60.40	42.14	10.75	4.05	29.36
RetinaNet$_{3D}$	0.8	AP_{3D0}	26.00	6.56	36.07	3.51	25.65	4.65	21.72
		AP_{3D1}	30.55	12.35	36.07	46.12	25.65	8.02	25.09
		AP_{3D2}	25.96	12.35	51.66	32.4	12.14	3.69	31.41

Two refinements to the original base method were incorporated to address the growth of the 3D bounding box making use of information in the semantically segmented 3D mesh. The initial refinement collapses the original 3D bounding box so that it encompasses all the mesh elements corresponding to the predicted class label, but no more. This initial refinement (3D1) greatly improves the performance of the 3D object detection method, roughly doubling the mAP (Table 3.2) and the effect is notable in the visualized results (Figure 3.2). Examining the effect on individual classes shows that this refinement is absolutely critical for many classes (e.g. *P. astreoides*, *G. ventalina*), most of which are relatively small corals and hence subject to massive relative increase in bounding box size during 3D box merging, but less critical for others (e.g. *A. palamata*, *Siderastrea*), most of which are large and less subject to bounding box inflation (Table 3.3). After this initial refinement we noted that some bounding boxes were still inflated because a small number of mesh elements of the predicted class from a neighboring entity would be enclosed in the original 3D bounding box. In an attempt to exclude such spurious portions of neighboring entities we added a second refinement making use of connected components to identify the primary object in the 3D

bounding box. This final refinement (3D2) reduced overall performance (in terms of mAP) since some true entities were composed of multiple connected components leading to bounding boxes that were too small. However, the final refinement did improve the detector's performance on *Antillogorgia*, which often occurs in clumps and so is especially susceptible to overlap between neighboring objects (Table 3.3).

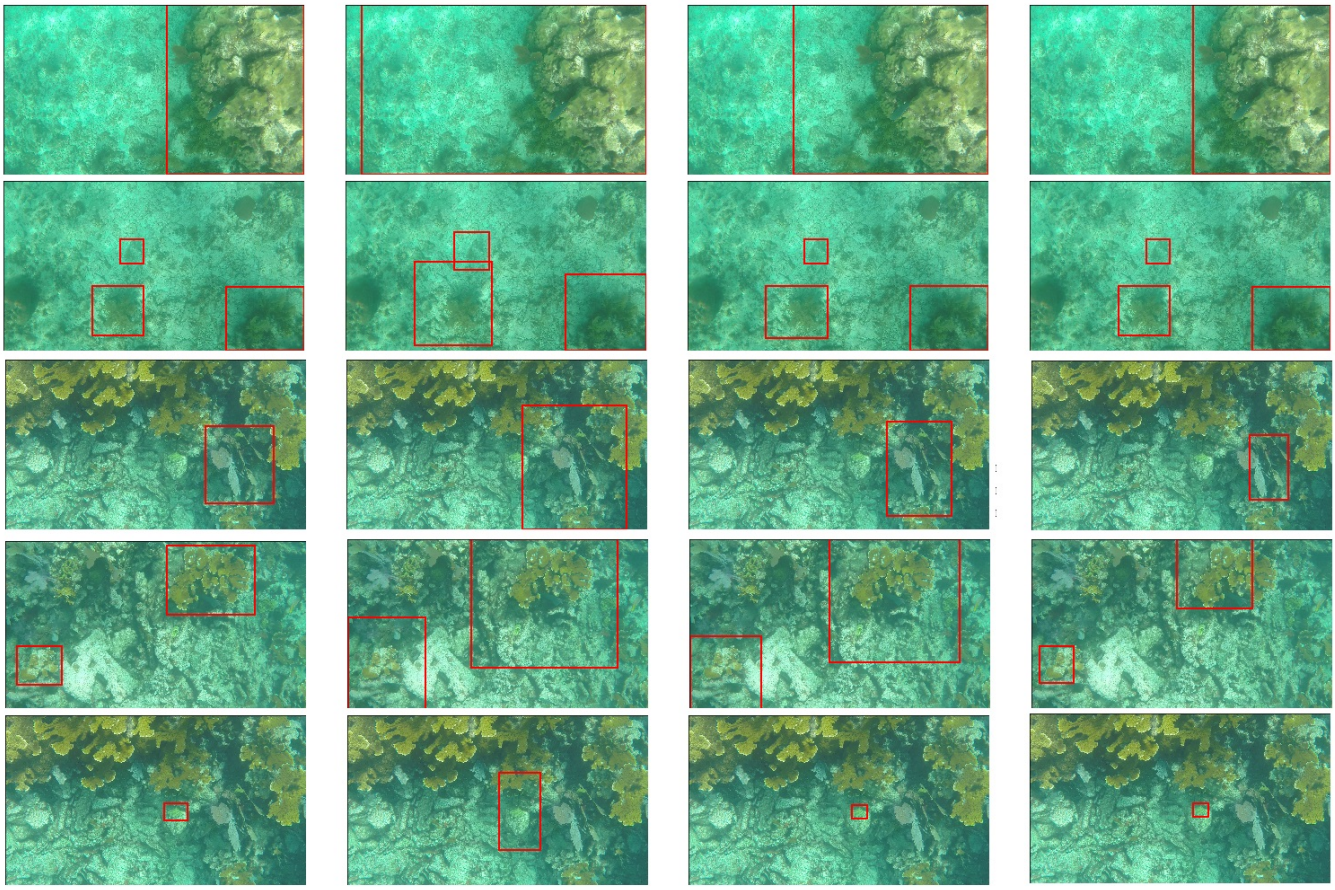

Figure 3.2: **Visual Object Detection Results:** Top row to bottom row: classes *Orbicella*, *Antillogorgia*, *G. ventalina*, *A. palmata*, *P. astreoides*. Left column to right column: 2D object detection with *RetinaNet*, projected 3D bounding boxes using the base method, projected 3D bounding boxes using the base method and refinement step 1 using *nViewNet*, projected 3D bounding boxes using the base method and refinement steps 1 and 2 using *nViewNet*.

3.8 Conclusion and Future Work

The proposed 3D object detection approach employs mature 2D object detectors in conjunction with an existing semantically segmented mesh to identify individual entities in 3D reconstructions of coral reefs. Unlike other 3D object detection methods, our approach is simpler to implement because it does not require manually annotated 3D bounding boxes and is partly algorithmic rather than entirely learned. However, it requires an existing semantically segmented 3D surface mesh, an approach we took because the 3D object detector is envisioned as part of a reef mapping pipeline and because the clonal nature of coral reef organisms complicates object detection. Our method performs fairly well for four of the seven the classes (*A. palamata*, *Siderastrea*, *P. asteroides*, *Antillogorgia*) but has poor performance for the remaining three. The classes on which the method performed poorly also challenged the 2D object detectors due to within-class heterogeneity and limited training data.

While our refinements to the base method generally improved the performance and mitigated bounding box enlargement, important performance issues remain that would be addressed in future work. Currently we do not account for occlusion of portions of a 3D object by reef structure, but since a 3D reef reconstruction exists, this could be employed to ensure only the visible portion of the 3D object is considered when assessing the performance relative to 2D manually annotated boxes. A more fine-grained 3D reef representation could help improve the performance on smaller and thinner classes such as *Sea Rods* and *P. astreoides*. Finally, while the proposed approach is partly algorithmic, in part due to the small size of the training data set, an end-to-end learning-based approach for 3D object detection from point clouds (Qi et al., 2018; Wang & Jia, 2019) could potentially yield better performance and will be a topic for future research.

Chapter 4

Assessing the performance of Frustum Pointnet for 3D Object Detection on Coral Point Clouds[1]

[1]Sushanth Kathirvelu*, Suchendra Bhandarkar, Brian Hopkinson. To be submitted to International Conference on Computer Vision, 2022).

4.1 3D Object Detection

Detection of objects in 3D data is gaining significant importance in many applications like augmented reality, autonomous driving, etc. RGB-D and LiDAR point clouds are the most common data types used for the 3D object detection tasks. This chapter focuses on 3D object detection from point clouds, which is most closely aligned to our work.

The earlier works in the 3D object detection domain converted point clouds to images before feature learning. B. Li et al., 2016 employ a CNN on RGB-D depth data represented as 2D maps to detect and localize objects in 2D images. The *MV3D* architecture (Chen et al., 2017) trains a region proposal network on LiDAR point cloud data the results of which are then projected onto a birds eye view image for generating 3D bounding box proposals.

The *PointNet* architecture (Qi, Su, et al., 2017) performs object segmentation and classification directly on point cloud data. With significant advances in sensors capable of generating 3D point clouds (Qi, Su, et al., 2017; Qi, Yi, et al., 2017) recent research in 3D object detection has focused on learning 3D features directly from 3D point clouds (Qi et al., 2018; Wang & Jia, 2019) .

The *Frustum PointNet* (Qi et al., 2018) architecture projects all the points in the point cloud back into the image plane and finds the points that lies within the 2D region proposal. It then uses the *PointNet* (Qi, Su, et al., 2017) architecture to segment the foreground points from the list of points determined in the previous step. A 3D bounding box is then estimated from the segmented points. The *Frustum ConvNet* architecture (Wang & Jia, 2019) uses a *Fully Connected Network (FCN)* to estimate the 3D bounding boxes using the feature maps generated from the point cloud data.

4.2 Data Set

The coral dataset (chapter 3[subsection 3.4.1]) used in this chapter for 3D object detection using *Frustum PointNets* (Qi et al., 2018) contains three 3D reconstruction meshes generated from ≈ 650 images. The

image data is split into 532 training and 113 test RGB images and are manually annotated for the 7 object categories.

The 3D point clouds are then generated from the reconstructed mesh as described in Section 4.2.1.

4.2.1 3D Point cloud Generation

A 3D mesh reconstruction of the coral reef ecosystems was generated using commercial Structure from Motion (SfM) software (*Agisoft Photoscan 1.4.3* now *Metashape*) from the images (Hopkinson et al., 2020) (Refer chapter 3[subsection 3.4.2]). The world coordinates of all the vertices of the 3D reconstructed mesh are used to build a sparse point cloud representing points on the coral reef ecosystem. The sparse point cloud is augmented to build a dense point cloud by estimating 8-10 random points for each triangular mesh faces, such that the estimated points lie inside the respective faces.

4.2.2 Camera Calibration

The 3D camera intrinsic and extrinsic matrices are obtained from the SfM software as part of the 3D reconstruction procedure. The camera intrinsic matrix maps the pixel points in the image plane to the camera coordinate reference frame. The camera coordinates are transformed to the world coordinates using the camera extrinsic matrix.

Given the extrinsic and intrinsic parameters of a camera, 2D pixel points in an image can be back projected into world coordinates using the equation:

$$\mathbf{p} = \mathbf{K} \times [\mathbf{R}|\mathbf{t}] \times \mathbf{P} \tag{4.1}$$

Where $\mathbf{p}$ is the pixel (u, v) in the image plane, $\mathbf{K}$ is the camera intrinsic matrix that represents the camera calibration parameters such as its focal length and the optical center and $[\mathbf{R}|\mathbf{t}]$ is the camera extrinsic matrix representing the location and orientation parameters of the camera in the 3D scene. The parameters $\mathbf{R}$ and $\mathbf{t}$ in the camera extrinsic matrix represent a 3×3 rotation matrix and a 3×1 translation matrix

(vector) respectively and **P** represents the 3D point (X, Y, Z) expressed in the world coordinate reference frame.

4.3 Experimental Results

Given 2D region proposals in RGB images, the *Frustum PointNet* architecture first projects all the points in the point cloud back to the image plane and finds the list of points that that corresponds to pixels inside a 2D region proposal. It then uses pointnet (Qi, Su, et al., 2017) to segment the foreground points from the list of points determined in the previous step, and from the segmented points a 3D bounding box is estimated.

We use the 2D object detectors as follows: 2D Region proposals are obtained by training state of the art 2D object detectors, *Faster-RCNN* (Ren et al., 2015) and *RetinaNet* (T.-Y. Lin et al., 2017), both with a *ResNet-50* backbone. The performance of the two object detectors are recorded in Table 4.1. Since *RetinaNet* outperforms *Faster-RCNN* for our coral dataset, *RetinaNet* is used as the baseline for estimating 3D region proposals using *Frustum PointNet*.

Table 4.1: 2D Object Detection Results

Model	Batch Size	Backbone	mAP_{50}
Faster-RCNN	32	ResNet 50	54.23
RetinaNet	32	ResNet 50	60.44

To determine all the point cloud points that corresponds to the image and the 2D bounding box proposals, the pinhole camera model and camera distortion algorithms (chapter 2[section 2.1; section 2.2]) are used to project the points from world coordinates to the image plane. *PointNet* uses these determined points for segmentation and 3D bounding box estimation. The resulting bounding box estimations are in the camera coordinates, to view the results in the 3D reconstructed mesh, camera extrinsic parameters and ray tracing algorithms (chapter 2[section 2.3]) are used to project the results from camera coordinate to the world coordinate.

In the *Frustum PointNet* architecture, a single point cloud is obtained for each image in the data set. In contrast, since our 3D point clouds are constructed from multiple images using SfM, the same object is viewed in multiple images from varying viewpoints or parts of a single large objects are viewed in multiple images. This results in multiple bounding boxes being estimated for the same object. We use a merging technique to eliminate repeating bounding boxes. If the intersection volume of two bounding boxes (characterized by their minimum and maximum x, y and z in world coordinates) that represent the same object class in the reconstructed 3D surface mesh is over a predefined threshold of 50%, we merge them into a single bounding box by updating the minimum and maximum x, y and z values as seen in Figure 4.1.

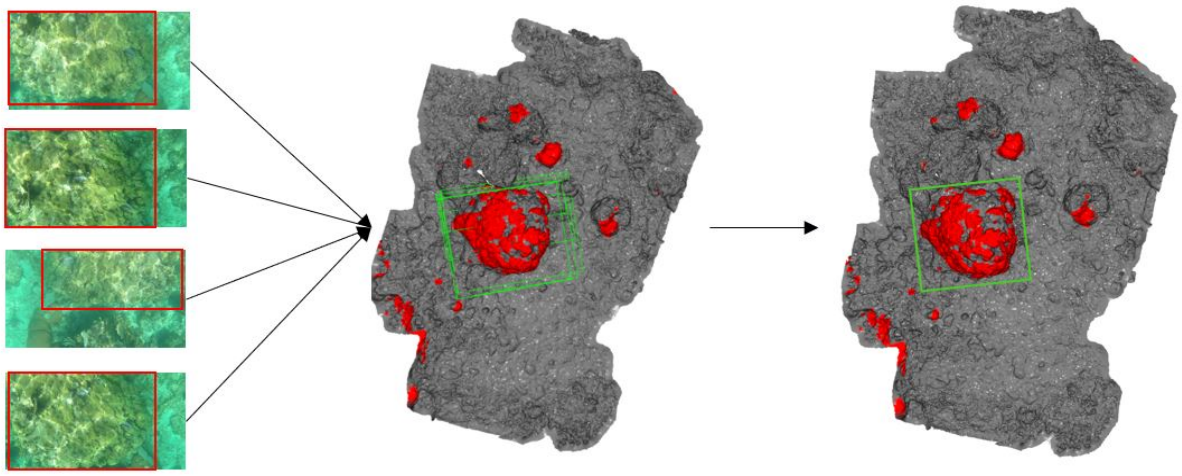

Figure 4.1: **Merging overlapping 3D bounding boxes:** From left to right, 2D object is viewed in multiple images; 3D bounding boxes are predicted; merging of multiple back-projected bounding boxes of the same object into a single bounding box

Although the post processing merge technique that we employ eliminates the presence of duplicate bounding boxes in the 3D ecosystem maps, the resulting 3D bounding boxes estimated after the merge for each object of interest is larger than the object it encloses. This is reflected in the poor *average precision* (AP) values obtained from the *Frustum PointNet*, where the AP values are computed based on the *Intersection over Union* (IOU) measure calculated for the ground truth and the estimated bounding boxes. Table 4.2

shows the performance of Frustum Pointnet on our coral dataset on individual classes. The performance is evaluated in the same manner as described in chapter 3 (section 3.6).

Table 4.2: **Frustum Pointnet results on coral Data set:** Average Precision (AP) results for different coral classes

Class	Average Precision(AP)
Acropora palmata	8.92
Orbicella	6.30
Porites astreoides	2.29
Gorgonia ventalina	0.68
Sea Rods	0.87
Antillogorgia	1.36

The *Frustum PointNet* architecture fails in the accurate detection of objects that are close to each other in 3D. When multiple instances from the same object category are viewed in a frustum (e.g. two *Antillogorgia* objects located close to each other), the multiple instances are regarded as a single object during foreground segmentation process. This results a larger bounding box estimation than the actual bounding box of the object as seen in Figure 4.2.

In addition, the final estimation of 3D bounding boxes in *Frustum Pointnet* relies on the segmentation of foreground points. Since our point clouds are acquired from 3D mesh reconstructions, there are no distinctive difference between the foreground and background points for smaller classes i.e,. *P. astreoides*, *G. ventalina*, which affect the results for these classes. Furthermore, in contrast we notice significantly better results for larger classes(*A. palamata*, *Orbicella*) which have comparatively distinctive foreground and background points after reconstruction due to their size.

The figure 4.2 visualizes the 2D region proposals and their corresponding 3D predictions generated using frustum pointnet. For visualization of results in the 3D reconstructions we back project the result from the 3D camera coordinates to the 3D World coordinates by using the camera extrinsic parameters(4.2.2).

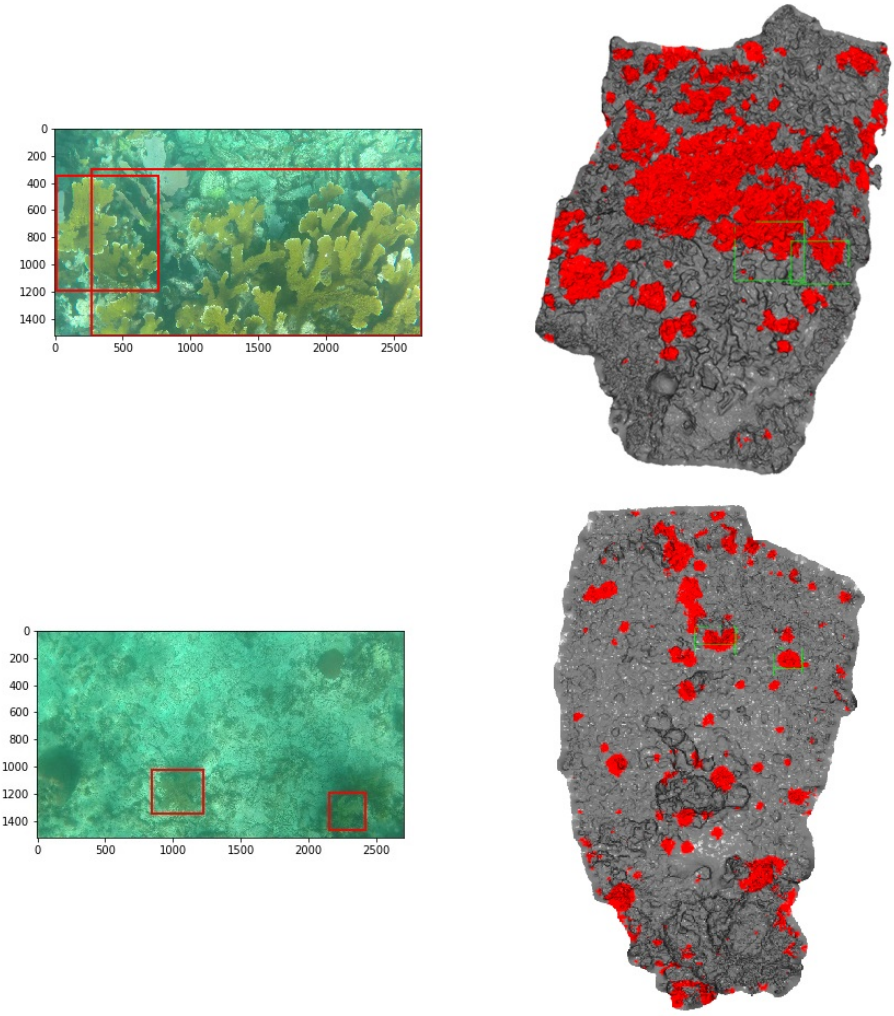

Figure 4.2: **Visualization of Frustum Pointnet results:** Visualization of 2D region proposals versus 3D predictions for the classes *A. palamata* (Top) and *Antillogorgia* (bottom)

4.4 Conclusion and Future work

The *Frustum PointNet* performs comparatively well for larger classes i.e,. *A. palamata*, *Orbicella*, but has poor performance in other smaller classes. Given the colonial nature of the coral reefs, detecting individual coral objects using the *Frustum PointNet* is challenging, since the *Frustum PointNet* architecture assumes one object per frustum when training and testing. Future work should focus on evaluating the performance of other end-to-end learning approaches, such as the *Frustum ConvNet* (Wang & Jia, 2019) and *FroDo* (K. Li et al., 2020) on our coral data set and comparing their performance to that of the *Frustum PointNet*.

Finally, since the image data was collected with stereo cameras, future work could look at incorporating disparity information and generating an individual stereoscopic depth point cloud for each RGB image and assess the performance of *Frustum PointNet* on the RGB image and point cloud pairs.

Chapter 5

Conclusion and Future Work

We presented a partly algorithmic approach for detecting coral reefs in 3D reconstructed ecosystem maps. To identify discrete individual coral entities within a 3D surface mesh surface, we proposed to apply object detection algorithms on the input 2D images of the ecosystem and back-project the resulting predictions onto the 3D surface mesh reconstruction. We also explored and assessed the performance of the *Frustum PointNet* architecture (Qi et al., 2018) on the coral reef ecosystem data set.

By comparing the performance of our proposed method (chapter 3) and *Frustum PointNet* (Qi et al., 2018) for 3D object detection, we conclude our proposed method outperforms the *Frustum PointNet* model. Investigating the results further, we found both the approaches performed comparatively well at detecting the larger classes, i.e., *Acropora palmata* and *Orbicella*. The performance was observed to decline for smaller and less abundant classes, i.e., *Antillogorgia*, *Sea Rods*.

Both our proposed approach and the *Frustum PointNet* architecture rely on the results from the 2D object detectors. If an object is not detected in 2D, the object is not detected in 3D, impacting the performance of the models. We note that in both methods, the classes that performed poorly also challenged the 2D object detectors, further demonstrating the importance of 2D region proposals for the 3D object detection tasks. A larger 2D train dataset could potentially yield better 2D region proposals and subsequently estimate more accurately the underlying 3D objects.

Finally, the coral point clouds generated from the 3D mesh reconstructions do not have distinctive foreground and background points for the smaller objects, which limits the performance of *Frustum PointNet*. A more refined 3D reef representation could help improve the performance of *Frustum PointNet* on smaller and thinner classes such as *Sea Rods* and *P. astreoides*.

Bibliography

Anthony, K. R. (2016). Coral reefs under climate change and ocean acidification: Challenges and opportunities for management and policy. *Annual Review of Environment and Resources*, *41*(1), 59–81. https://doi.org/10.1146/annurev-environ-110615-085610

Beijbom, O., Edmunds, P. J., Kline, D. I., Mitchell, B. G., & Kriegman, D. (2012). Automated annotation of coral reef survey images. *2012 IEEE Conference on Computer Vision and Pattern Recognition*, 1170–1177. https://doi.org/10.1109/CVPR.2012.6247798

Brodrick, P., AB, D., & GP, A. (2019). Uncovering ecological patterns with convolutional neural networks. *Trends in Ecology and Evolution*, *34*(8), 734–745. https://www.sciencedirect.com/science/article/pii/S0169534719300862

Burns, J. H. R., Alexandrov, T., Gates, R. D., & Takabayashi, M. (2016). Investigating the spatial distribution of growth anomalies affecting montipora capitata corals in a 3-dimensional framework. *Journal of invertebrate pathology*, *140*, 51–57. https://doi.org/10.1016/j.jip.2016.08.007

Chen, X., Ma, H., Wan, J., Li, B., & Xia, T. (2017). Multi-view 3d object detection network for autonomous driving.

Couch, C. S., Oliver, T. A., Suka, R., Lamirand, M., Asbury, M., Amir, C., Vargas-Ángel, B., Winston, M., Huntington, B., Lichowski, F., Halperin, A., Gray, A., Garriques, J., & Samson, J. (2021). Comparing coral colony surveys from in-water observations and structure-from-motion imagery shows low methodological bias. *Frontiers in Marine Science*, *8*, 622. https://doi.org/10.3389/fmars.2021.647943

He, K., Gkioxari, G., Dollár, P., & Girshick, R. (2017). Mask r-cnn. *2017 IEEE International Conference on Computer Vision (ICCV)*, 2980–2988. https://doi.org/10.1109/ICCV.2017.322

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 770–778.

Hedley, J. D., Roelfsema, C. M., Chollett, I., Harborne, A. R., Heron, S. F., Weeks, S., Skirving, W. J., Strong, A. E., Eakin, C. M., Christensen, T. R. L., Ticzon, V., Bejarano, S., & Mumby, P. J. (2016). Remote sensing of coral reefs for monitoring and management: A review. *Remote Sensing*, *8*(2). https://doi.org/10.3390/rs8020118

Hoegh-Guldberg, O., Mumby, P., Hooten, A., Steneck, R., Greenfield, P., Gomez, E., Harvell, C., Sale, P., Edwards, A., Caldeira, K., Knowlton, N., Eakin, C., Iglesias-Prieto, R., Muthiga, N., Bradbury, R., Dubi, A., & Hatziolos, M. (2007). Coral reefs under rapid climate change and ocean acidification. *Science*, *318*(5857), 1737–1742.

Hopkinson, B. M., King, A. C., Owen, D. P., Johnson-Roberson, M., Long, M. H., & Bhandarkar, S. M. (2020). Automated classification of three-dimensional reconstructions of coral reefs using convolutional neural networks. *PLoS ONE*, *15*.

Jalal, A., Salman, A., Mian, A., Shortis, M., & Shafait, F. (2020). Fish detection and species classification in underwater environments using deep learning with temporal information. *Ecological Informatics*, *57*, 101088. https://doi.org/https://doi.org/10.1016/j.ecoinf.2020.101088

King, A., M.Bhandarkar, S., & Hopkinson, B. M. (2019). Deep learning for semantic segmentation of coral reef images using multi-view information. *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*.

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, *521*(7553), 436–444. https://doi.org/10.1038/nature14539

Leon, J. X., Roelfsema, C. M., Saunders, M. I., & Phinn, S. R. (2015). Measuring coral reef terrain roughness using 'Structure-from-Motion' close-range photogrammetry. *Geomorphology*, *242*, 21–28. https://doi.org/10.1016/j.geomorph.2015.01.030

Li, B., Zhang, T., & Xia, T. (2016). Vehicle detection from 3d lidar using fully convolutional network.

Li, K., Garg, R., Cai, M., & Reid, I. (2018). Single-view Object Shape Reconstruction Using Deep Shape Prior and Silhouette. *arXiv e-prints*.

Li, K., Rünz, M., Tang, M., Ma, L., Kong, C., Schmidt, T., Reid, I. D., Agapito, L., Straub, J., Lovegrove, S., & Newcombe, R. A. (2020). Frodo: From detections to 3d objects. *CoRR*, *abs/2005.05125*. https://arxiv.org/abs/2005.05125

Lin, C.-H., Wang, O., Russell, B. C., Shechtman, E., Kim, V. G., Fisher, M., & Lucey, S. (2019). Photometric mesh optimization for video-aligned 3d object reconstruction. *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 969–978. https://doi.org/10.1109/CVPR.2019.00106

Lin, T.-Y., Goyal, P., Girshick, R., He, K., & Dollár, P. (2017). Focal loss for dense object detection. *2017 IEEE International Conference on Computer Vision (ICCV)*, 2999–3007. https://doi.org/10.1109/ICCV.2017.324

Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., & Berg, A. C. (2016). Ssd: Single shot multibox detector. In B. Leibe, J. Matas, N. Sebe, & M. Welling (Eds.), *Computer vision – eccv 2016* (pp. 21–37). Springer International Publishing.

Lopez-Marcano, S., L. Jinks, E., Buelow, C. A., Brown, C. J., Wang, D., Kusy, B., M. Ditria, E., & Connolly, R. M. (2021). Automatic detection of fish and tracking of movement for ecology. *Ecology and Evolution*. https://doi.org/https://doi.org/10.1002/ece3.7656

Möller, T., & Trumbore, B. (1997). Fast, minimum storage ray-triangle intersection. *Journal of Graphics Tools*, *2*(1), 21–28. https://doi.org/10.1080/10867651.1997.10487468

Qi, C. R., Liu, W., Wu, C., Su, H., & Guibas, L. J. (2018). Frustum pointnets for 3d object detection from rgb-d data. *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 918–927. https://doi.org/10.1109/CVPR.2018.00102

Qi, C. R., Su, H., Mo, K., & Guibas, L. J. (2017). Pointnet: Deep learning on point sets for 3d classification and segmentation.

Qi, C. R., Yi, L., Su, H., & Guibas, L. J. (2017). Pointnet++: Deep hierarchical feature learning on point sets in a metric space.

Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Ren, S., He, K., Girshick, R. B., & Sun, J. (2015). Faster r-cnn: Towards real-time object detection with region proposal networks. *NIPS*, 91–99. http://dblp.uni-trier.de/db/conf/nips/nips2015.html#RenHGS15

Simonyan, K., & Zisserman, A. (2015). Very deep convolutional networks for large-scale image recognition. *Proc. Intl.Conf. Learning Representations (ICLR)*, 1–14.

Storlazzi, C. D., Dartnell, P., Hatcher, G., & Gibbs, A. E. (2016). End of the chain? rugosity and fine-scale bathymetry from existing underwater digital imagery using structure-from-motion (sfm) technology. *Coral Reefs*, *35*, 889–894. https://doi.org/10.1007/s00338-016-1462-8

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S. E., Anguelov, D., Erhan, D., Vanhoucke, V., & Rabinovich, A. (2015). Going deeper with convolutions. *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 1–9.

Wang, Z., & Jia, K. (2019). Frustum convnet: Sliding frustums to aggregate local point-wise features for amodal 3d object detection. *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 1742–1749.

Weinstein, B., Marconi, S., Bohlman, S., Zare, A., & White, E. (2019). Individual tree-crown detection in rgb imagery using semi-supervised deep learning neural networks. *Remote Sensing*, *11*, 1309. https://doi.org/10.3390/rs11111309

Williams, I. D., Couch, C. S., Beijbom, O., Oliver, T. A., Vargas-Angel, B., Schumacher, B. D., & Brainard, R. E. (2019). Leveraging automated image analysis tools to transform our capacity to assess status and trends of coral reefs. *Frontiers in Marine Science*, *6*, 222. https://doi.org/10.3389/fmars.2019.00222

Zhang, Z. (2000). A flexible new technique for camera calibration. *IEEE Transactions on pattern analysis and machine intelligence*, *22*(11), 1330–1334.