Object Detection on 3D maps of Coral ecosystems from images

Sushanth Kathirvelu[1] Suchendra M. Bhandarkar[1,2] Brian M. Hopkinson[3]

[1]Dept. of Computer Science [2]Institute for Artificial Intelligence [3]Dept. of Marine Sciences
University of Georgia, Athens, Georgia 30602, USA
{sushanth.kathirvelu, suchi, bmhopkin}@uga.edu

Abstract

Coral reefs are biologically diverse and structurally complex ecosystems, which have been severally affected by natural and anthropogenic stressors. Hence, there is a need for rapid ecological assessment of coral reefs, current approaches require time-consuming manual analysis. In this work, we propose a method to identify individual species within the ecosystem for further analysis by domain experts. We propose a method to detect individual species in the 3D reconstruction of coral reef ecosystem and assess this method's accuracy. Given 2D region proposals in an RGB image our method generates 3D region proposals for each 2D region proposals. 3D reconstructions were generated using commercial Structure-from-Motion software with images extracted from video surveys. To identify the individual objects in the 3D reconstructed map, camera parameters were used to back project the 2D region proposals into the 3D Reconstruction.

1. Introduction

The marine ecology, more specifically the coral reef ecology, is gaining growing importance recently to study since both natural and anthropogenic stressors threaten coral reefs across the globe. Since these stressors, including climate change, ocean acidification, sea-level rise, pollutant runoff, and overfishing [1, 7], have combined to fast deterioration of coral reefs worldwide over the past three decades [2], it is essential to map and monitor these ecosystems.

While coral reefs are a subject of cultural and scientific importance, their complex ecosystem makes it technically challenging to monitor and study them. The current approach of monitoring the coral reef ecosystem involves time-consuming manual analysis either during a dive survey or on data collected during the survey by domain experts.

The study of the coral reef ecosystem is limited by the difficulty of generating accurate and repeatable maps of the ecosystem. The current mapping approaches that include physical mapping performed by human divers are time-consuming. Though Satellite and Ariel imaging provides decent insights on ecosystems on a large dimensional scale, they are limited because seawater absorbs light strongly, limiting monitoring shallow ecosystems like coral reefs [5]. Acoustic mapping is capable of mapping ecosystems on a large scale; it is inefficient in mapping finer details of the coral reef ecosystem.

Thanks to recent advances in Autonomous Underwater Vehicles (AUV) equipped with high-resolution cameras, the manual mapping techniques used to map coral reef ecology are being replaced by image and video-based robotic surveys. AUVs move systematically over coral reef environments and can continuously acquire high-quality images of small portions of the coral reef ecosystem, which has led to growing interest within the research communities in marine biology and marine ecology to exploit machine learning techniques for automatic mapping and analysis of underwater images []. Using computer vision and machine learning algorithms(SFM and SLAM), the individual images are automatically assembled into a large-scale, 3D reconstruction (or map) of the coral reef ecosystem. Recent advances in machine learning and computer vision, especially deep learning [11] using convolutional neural networks (CNNs / ConvNets), offer the potential to automate the analysis of these ecosystems.

CNN's / ConvNets are multi-layered neural network-based classifiers that apply a convolutional filter to an input image to create a feature map that summarizes the presence of detected features in the images. CNN automatically learns discriminating features from the image data without any human intervention compared to the conventional method of hand-engineered feature extractors that require domain knowledge [11]. CNN provides a framework for many computer vision applications, allowing many basic computer vision tasks to be performed much more

accurately and efficiently than pre-engineered workflows. CNN's excel at image classification, i.e., assigning a single label to an entire image, object detection, i.e., identifying instances of specific objects in an image, and semantic segmentation, i.e., labeling each pixel in an image as belonging to one of several predefined classes.

Our previous works on coral reefs focused on integrating advances in 3D mapping and CNNs to automatically generate semantically segmented 3D maps of the reef ecosystem [9]. The semantically segmented maps generated using SfM based reconstruction and CNN-based image analysis helps in identifying the surface area occupied by a species in the ecosystem.

In our work, we propose a method to identify individuals within the ecosystem, thereby addressing the critical questions in population ecology revolving around individuals, which are the fundamental units of social interaction. Identification of individuals, counting them, and spatial localizing them are essential to understanding the dynamics of the underlying population. Since individuals are also components of ecological communities, this data can be used to identify distinct biological communities. To identify discrete individuals within a 3D mesh surface, we propose to apply object detection algorithms to the input 2D images of the ecosystem. We back-project the bounding boxes of detected objects onto the 3D mesh reconstruction in a manner similar in principle to our 3D semantic segmentation approach[9].

2. Related Work

convolutional Neural Networks (CNN's or ConvNets) and related Deep Neural Networks (DNNs) have revolutionized computer vision, especially regarding image segmentation, feature extraction and classification, and object detection and recognition [10, 11]. The superior performance of CNN- and related DNN-based approaches has led to their rapid adoption in ecological research. Brodrick et al. (2019) argue that CNNs may become essential tools for ecologists due to their power. Williams et al. (2019) have employed CNNs to assess the abundance of major taxa and substrates on coral reefs, achieving classification accuracies similar to human annotators.

Advances in computer vision have also enabled 3D reconstruction and 3D visual mapping from 'local' imagery at a much higher resolution than is possible with remote sensing. These 3D reconstructions and mapping approaches have begun to be used in ecology, and several recent studies illustrate how novel, sophisticated ecological insights can be obtained from the resulting 3D maps. Edwards et al. (2017) mapped coral colonies on Palmyra Atoll and showed that coral spatial patterns were consistent with reproduction models via fragmentation and dispersal.

Previous application of Computer Vision and Machine Learning techniques on the coral reef ecosystem focused on object detection, segmentation, and classification of the images or 3D map generation, treating them as separate problems.

Our previous works [8, 9] merged the state-of-the-art approaches of 3D scene reconstruction (SfM-based reconstruction) and semantic segmentation using CNN's to generate 3D semantically segmented maps of coral reefs. We exploited the fact that we acquire multiple images of a coral reef from varying viewpoints for most coral reef surveys, typically via stereoscopic images or underwater videos. We proposed a patch-based CNN (nViewNet) and an FCNN (TwinNet) architectures to use these stereoscopic and multi-view image information to improve semantic segmentation and classification accuracy.

The TwinNet FCNN architecture processes both the left- and right-perspective stereo images directly to generate a single classification. nViewNet patch-based CNN architecture is capable of processing images from different viewpoints and combining them to yield a single classification via logit pooling [9]. The nViewNet architecture was used to semantically segment 3D reconstructions of coral reefs, identifying corals, algae, and substrates.

Our work uses an object detector that predicts a bounding box outlining each object in an image and identifies its class. The bounding boxes of detected objects will be mapped onto a semantically-segmented 3D reconstruction generated using the nViewNet architecture to identify individuals in the ecosystem.

3. Materials and Methods

3.1. Underwater Image Data Acquisition

The underwater coral images used in the work reported were manually collected by a team of divers from coral reefs off the Florida keys using a stereo-video camera. An underwater stereo camera rig, comprising of a Go Pro Dual Hero camera system, was used to collect the underwater video data while swimming over sections of the coral reef. The stereo camera rig was carried over the reef in a serpentine (i.e., lawn mower) pattern in order to capture a complete section of the seafloor. Stereo images were extracted from the video data at a rate of two frames per second.

A subset of the collected images from our coral reef image data-set was annotated to provide ground truth bounding boxes. During the annotation process, an object in an image is selected and it is assigned to one of the following eight classes: (1) Acropora palmata, (2) Orbicella (3) Siderastrea (4) Porites astreoides, (5) Gorgonia ventalina, (6) Sea Rods and (8) Antillo Gorgia.

3.2. 3D Reconstruction

3D reconstructions was generated from the images collected manually, using an commercial Structure From Motion (SFM) software (Agisoft Photoscan 1.4.3 now Metashape). We use approximately between 100 to 5000 images for each reconstruction.

The SFM software identifies matching features in different images. These features are tracked from image to image and is used to estimate the camera positions and orientations and the coordinates of the features. This produces a sparse point cloud representing the reef. A dense point cloud is constructed by identifying additional common features between the images. A triangular mesh was generated from the dense point cloud, producing roughly 200k-300k triangular faces that represents the surface of the coral reef. A texture map was produced from the images for visualization as shown in figure 1.

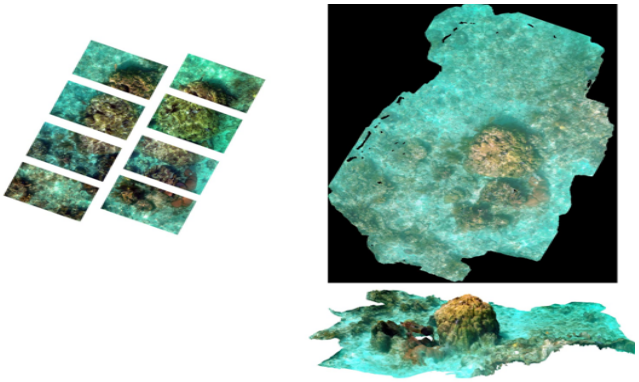

Figure 1. 3D Reconstruction

Camera transformation matrices and camera calibration parameters were obtained from Agisoft Photoscan as part of the 3D reconstruction procedure.

3.3. Object Detection Algorithms

Object detection is one of the important problem in computer vision which predicts what objects are present in an image, with a bounding box enclosing the objects.

Image classification predicts the class of an object in an image. Object localization refers to identifying the location of one or more objects in an image and drawing abounding box around the object. Object detection combines these two tasks and localizes and classifies one or more objects in an image.

There are two types of state of the art object detectors discussed next.

Single Stage Detectors single stage detector algorithms like RetinaNet [12], YOLO [14], SSD etc,. considers object detection as a simple regression problem. It performs both the classification and localization in a single pass. Single stage object detectors are fast but tens to reach lower accuracy.

Two Stage Detectors Two stage detector algorithms like Faster RCNN [15],Mask RCNN [4] etc,. uses a Region Proposal Network (RPN) to narrow down the region of interest in an image filtering out the background data. In the second stage classification is performed these regions. Two stage detectors tend to perform better than a single stage detectors but are comparatively slow.

3.4. Pinhole Camera Model

Understanding the geometrical model of the camera projection serves as the core idea for the paper. In this paper, we use the pinhole camera model [16] with distortion factor [6] [3]. A Pinhole Camera model gives 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 inversely use this model to also back project an image pixel to the 3D world space.

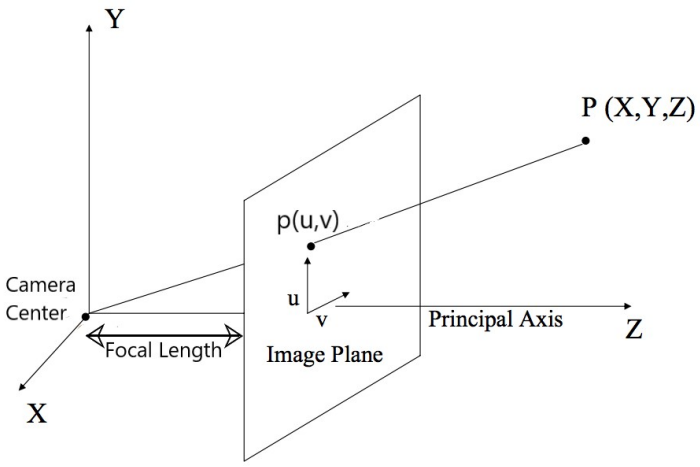

Figure 2. **Pinhole Camera**

In a pinhole camera model the camera coordinate and world coordinate frames are related by rotation and translation. The mathematical function that describes projection of 3D world points to 2D image plane can be written as

$$p = K[R|t] * P \tag{1.1}$$

Where p is the pixel point (u, v) in the image plane, K is the camera intrinsic matrix that represents the camera calibrations like the focal length and the optical center of the camera and $[R|t]$ is the extrinsic parameters representing where the camera was located in the 3D scene. The R and t in the Extrinsic Parameters represents a 3×3 rotation matrix

and a 3×1 translation matrix respectively and P represents the 3D point (X, Y, Z) expressed in the world coordinate.

The Eq. (1.1) can be expanded as

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & u_0 \\ 0 & f_y & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \tag{1.2}$$

Where u_0 and v_0 is the Principal point (or) 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.

3.5. Camera Distortion

The above equations (1.1) and (1.2) does not account for lens distortion while calculating the pixel points in the image plane. In Practice cameras introduces some distortion to its images. To accurately represent a real camera, to the the camera model we include the radial and tangential lens distortion to calculate the pixels in the camera coordinate.

Radial Distortion Radial distortion occurs when the light rays bend when passing through the lens. This type of distortion causes straight lines to appear curved. In a typical lens, the distortion is usually 0 at the center of the image and increases as we moved further outside. In other words, the light rays further from the optical center of the lens are curved more than the ones closer; hence this distortion could be noticed near the edge of the images.

Radial distortion can be corrected for the distorted pixel points (u_d, v_d) using the formula below,

$$\begin{aligned} u_{corrected} &= u_d(1 + k_1 * r^2 + k_2 * r^4 + k_3 * r^6) \\ v_{corrected} &= v_d(1 + k_1 * r^2 + k_2 * r^4 + k_3 * r^6) \\ &where \\ r &= \sqrt{u^2 + v^2} \end{aligned} \tag{2.1}$$

where $(u_{corrected}, v_{corrected})$ are the corrected pixel coordinates of the distorted pixels (u_d, v_d) and k_1, k_2 and k_3 are radial distortion coefficients of the lens

Tangential Distortion Tangential distortion occurs due to some manufacturing mistakes where the lens is not aligned parallel to the image plane. This type of distortion causes the image to look tilted, which in turn makes some objects in the image look further than they really are.

Radial distortion can be corrected for the distorted pixel points (u_d, v_d) using the formula below,

$$\begin{aligned} u_{corrected} &= u_d + [2 * p_1 * u_d * v_d + p_2 * (r^2 + 2 * u_d^2)] \\ v_{corrected} &= v_d + [p_1 * (r^2 + 2 * y^2) + 2 * p_2 * u_d * v_d] \\ &where \\ r &= \sqrt{u^2 + v^2} \end{aligned} \tag{2.2}$$

where $(u_{corrected}, v_{corrected})$ are the corrected pixel coordinates of the distorted pixels (u_d, v_d) and p_1, p_2 are tangential distortion coefficients of the lens.

We can calculate the corrected pixels in the camera coordinate after accounting for both the radial and tangential distortion by adding equations (2.1) and (2.2) to the pixels in the image frame as

$$\begin{aligned} u_{corrected} &= u_d(1 + k_1 * r^2 + k_2 * r^4 + k_3 * r^6) + \\ &\quad [2 * p_1 * u_d * v_d + p_2 * (r^2 + 2 * u_d^2)] \\ v_{corrected} &= v_d(1 + k_1 * r^2 + k_2 * r^4 + k_3 * r^6) + \\ &\quad [p_1 * (r^2 + 2 * y^2) + 2 * p_2 * u_d * v_d] \\ &where \\ r &= \sqrt{u^2 + v^2} \end{aligned} \tag{2.3}$$

3.6. Ray Triangle Intersection

Given the Origin of a Ray and the ray destination , we can draw a line passing through both the points using the Möller-Trumbore ray tracing algorithm [13]. Any point on the line can be calculated by the equation

$$p = O + t * D \tag{3.1}$$

where p is the point in the line, O is the line origin and D is the line direction, t is the distance between the origin and the point p.

Any point in a triangle can be defined as

$$p = (1 - u - v) * p_0 + u * p_1 + v * p_2 \tag{3.2}$$

where p_0, p_1, p_2 are the vertices of the triangle and u, v is a point that lies inside the triangle.

From the equations (3.1) and (3.2), for a ray to intersect the triangle.

$$O + t * D = (1 - u - v) * p_0 + u * p_1 + v * p_2 \tag{3.3}$$

We have 3 unknowns values t,u,v that could be calculated by rearranging the equation (3.3) as

$$\begin{bmatrix} -D & (p_1 - p_0) & (p_2 - p_0) \end{bmatrix} * \begin{bmatrix} t \\ u \\ v \end{bmatrix} = O - p_0 \tag{3.4}$$

3.7. Pixel Back Projection

Given a pixel (u,v) in the image plane, we can calculate the camera coordinates by using the camera's intrinsic parameters.

We account for both tangential and radial lens distortion of the camera to get the undistorted pixel coordinate

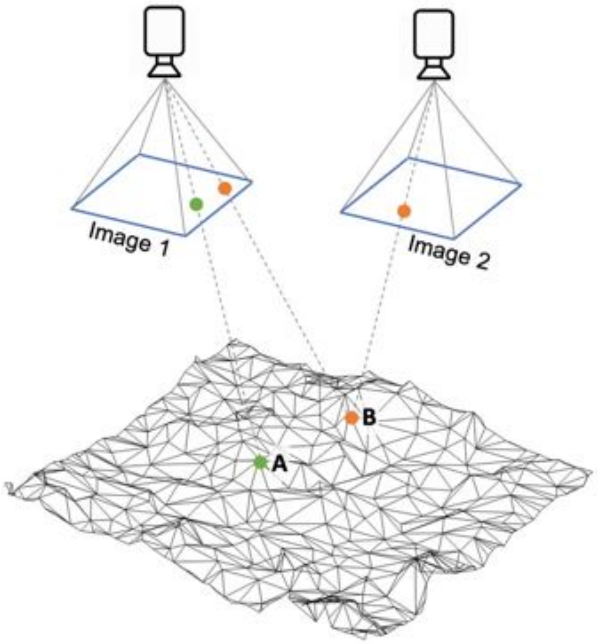

Figure 3. **Relationship between the images and Mesh**

A 3D Point in the camera coordinate can be transformed to the world coordinates by using the cameras extrinsic parameters as

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} R_{3*3} & T_{3*1} \\ 0_{1*3} & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \tag{3.5}$$

Since we can calculate the Ray's origin and the ray's direction using the equation (3.5). We can use the ray triangle intersection algorithm to find which point of the World object the pixel represents.

4. Proposed Method

Our Data Consist of Coral Reef Image that are acquired using a Stereo-Video Camera. We use these acquired stereo images for two purposes, To reconstruct a Triangular Mesh representing the surface of the reefs and to train the object detection models.

3D reconstructions were generated from these images using an commercial Structure From Motion (SFM) software (Agisoft Photoscan 1.4.3 now Metashape). Camera transformation matrices and camera calibration parameters were obtained from Agisoft Photoscan as part of the 3D reconstruction procedure.

Object detector Algorithm were trained to predict the location and the class of different objects present in each image. One Single stage Object detector (RetinaNet [12]) and one dual stage detector (Faster RCNN [15]) were trained and their performances were compared. These Object detectors predicts the location of the objects in the images and their classes in form of a bounding box along the pixel axis as shown in figure 6.

The core idea of our proposed method involves using the Camera calibration matrices obtained during the 3D reconstruction to back project the bounding boxes predicted for an image using the object detection algorithm into the reconstructed 3D mesh. The pixel points representing each corner of the predicted bounding boxes of the objects in the images are projected into the camera plane by applying the camera calibration model accounting for both radial and tangential non-linear distortions. The camera transform matrices were used to transform the pixel points from the camera coordinates to the world coordinates. The 3D mesh element these pixels represent is calculated by projecting a ray from the camera center passing through the calculated world coordinate, checking for the point of intersections with the mesh. Using the information obtained, we predicted the 3D bounding box by finding the minimum and maximum x, y, and z mesh coordinates enclosing all the mesh elements representing the object of interest.

Since objects are typically viewed in multiple images, multiple bounding boxes are created in the 3D space for the same object when we back project the detected 2D bounding boxes, as seen in figure 5. If the intersection volume of any two bounding boxes that represent the same class label in the 3D reconstruction is over a set threshold, we merge them into a single bounding box.

Since we back project the 2D bounding boxes in pixels representing an object into the reconstructed mesh and calculate the minimum and maximum x, y, and z coordinates representing the same object in a 3D mesh, the bounding boxes in the 3D reconstruction tends to be bigger than the object of interest. Hence, we refine our results to get a more compact bounding box enclosing the object in the 3D mesh.

In our initial refinement, we determine all the mesh elements within the mapped bounding box. We then find the minimum and maximum x, y, and z coordinates from all these mesh element centers and predict a new bounding box enclosing the object in the 3D mesh.

To get an even more compact bounding box in the 3D reconstruction we refine our results even further by using the results from the nViewNet [9]. nViewNet provides a segmentation result for each mesh element using information from multiple views of the same object that each mesh

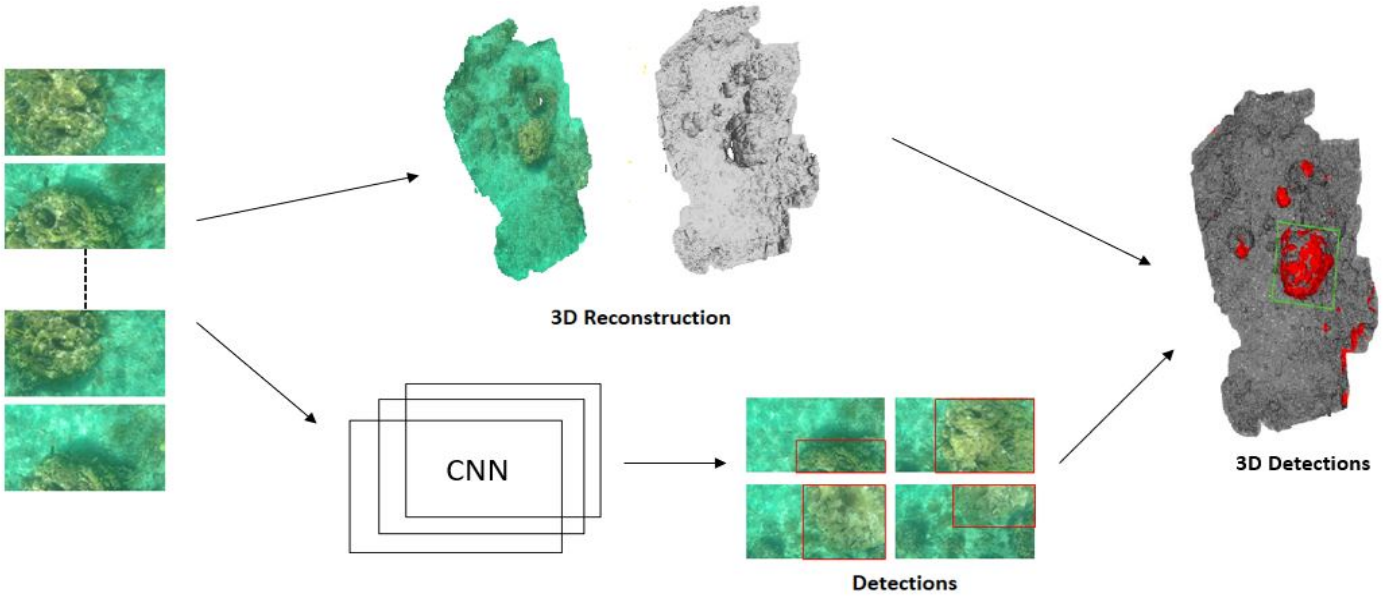

Figure 4. **Proposed Architecture.** We first use a 2D CNN object detector to propose 2D regions and classify their content. We also use the images to reconstruct a 3D structure. Using camera parameters, camera distortion and Ray Tracing techniques we back project the 2D predictions to the 3D structure

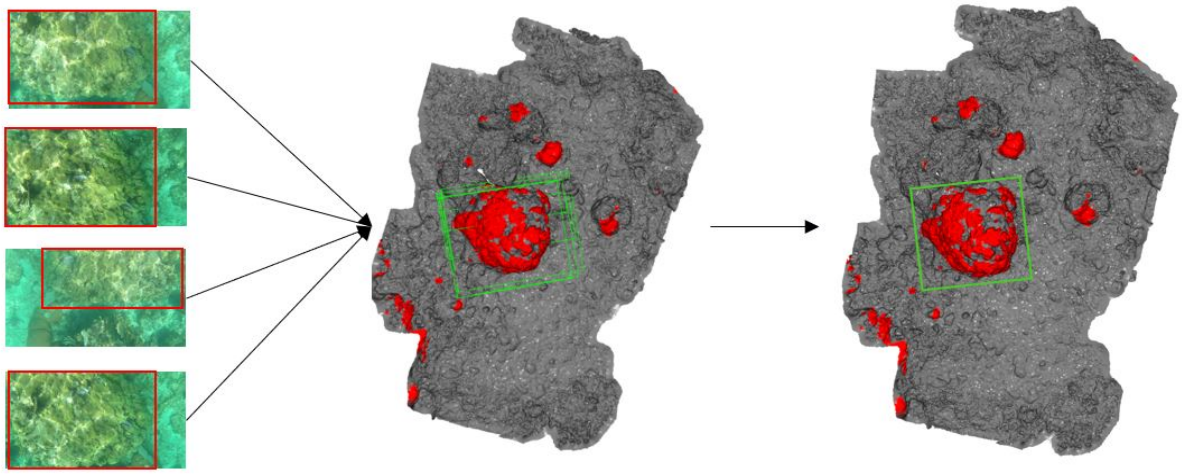

Figure 5. **Back Projecting 2D bounding boxes to 3D** From Left to Right, 1. Results from the 2D object detection algorithm , locating and classifying the objects present in an image, 2. Back Projected results of the images in the 3D reconstruction, 3. Merging multiple back projected bounding boxes of the same object into a single object

element represents.

With the results from the nViewNet specifying the class labels of each mesh element and the back projected bounding boxes predicted by our method, we find all the mesh elements within the mapped bounding box that match the object's class label. Every mesh element share each of its vertex with another element, with this information we can draw a tree structure with all the mesh elements inside the bounding box connected by their vertices. We can find all the connected component representing the faces inside the bounding box that belong to the same object and are connected with each other. We can extract the minimum and maximum coordinates from the connected components and predict a new compact bounding box enclosing only the object in the 3D mesh.

5. Performance Evaluation Metrics

To evaluate our model, we project the 3D Bounding Boxes predicted using our methods back to the 2D images from which the bounding boxes were predicted. The camera transform matrices are inverted and used to transform the mesh elements representing the 3D object bounding box corners from the world coordinate to the camera coordinate. These mesh elements representing the object bounding box corners in the camera coordinates are projected to the image plane by applying the camera calibration model accounting for radial and tangential non-linear distortions. These projected values are used to define a predicted 2D bounding box by finding the minimum and maximum x and y pixel coordinates and compared with manually annotated, ground-truth boxes.

We use the concept of Intersection over Union (IoU) to determine if our predicted bounding boxes are correct. IoU measures the overlap between the predicted and the ground truth bounding boxes for all predicted bounding boxes. If the IoU is over the set threshold, we classify the detected object as a True Positive(TP). On the other hand, if IoU is less than the set threshold, we classify it as False Positive (FP). We classify values present in the ground truth but not detected by our models as False Negative(FN). Using the IoU results we can calculate the precision and recall values as,

$$\begin{aligned} Precision &= \frac{TP}{TP+FP} \\ Recall &= \frac{TP}{TP+FN} \end{aligned} \tag{4.1}$$

With the precision and recall values, we calculate the average precision of all the individual classes representing how accurately our model detected each class. Average Precision(AP) is calculated by finding the area under the precision-recall curve. AP is calculated by segmenting the recall into 11 parts $(0, 0.1, 0.2, .., 0.9, 1)$ and averaging the corresponding precision values like

$$AP = \frac{1}{11} \sum_{Recall_i} Precision(Recall_i) \tag{4.2}$$

The mean Average Precision (mAP) score is calculated by taking the mean AP over all classes, predicting how accurate our model performs.

6. Results

2D Object prediction Training was done with different object detection algorithms . We focus on one single-stage 2D object detector (RetinaNet) and one dual-stage 2D object detector (Faster RCNN) for locating and classifying the objects present in 2D images and compare their performances. The results of the Object Detection models on the two dimensional images are given in Table 1.

Based on our evaluation metric, We project the 3D bounding boxes predicted by our proposed core and refined methods from the 3D reconstruction to the images and calculate the mAP for our model. The mean average precision(mAP) results of our model after projecting the 3D bounding boxes to the image is given in Table 2

Table 2 shows the mAP results produced by projecting the 3D bounding boxes of all the classes to the images. The results generated from the combination of results from the 3D bounding boxes and semantic segmentation from nView net and applying connected component technique to find all the faces inside the bounding boxes that belong to the same class as the detected object and are connected tends to perform better. A breakdown of per class results are shown in table 3 [BH: this paragraph is hard to follow - it is very terse. improve the logical flow and expand on what the results mean and how they relate to the different version of the method]

Table 3 shows that our proposed method tends to work better for stationary objects. The results indicates that we are able to produce almost the same results in 3D as the 2D prediction results from the object detection algorithms for most of the classes [BH: the statement is not supported by the table -there seems to be a substantial drop off in accuracy for most classes]. The results are also influenced by the amount instances of each classes used for training and testing our model.

7. Conclusion and Future Work

The 3D mapping and detection method described in this paper performed well for stationary objects(substrate, trees, corals, etc.). We showed how integrating results from nView net's semantic segmentation information improves the accuracy of detecting individual coral species in the 3D reconstruction of coral reefs.

In the area 2D detection, we Explored two 2D object detection architectures, one which considers the object detection as a simple regression problem, and the other, that uses region proposal network to narrow down the region of interest in the image. In the scope of 3D detection we explored how camera parameters can be used to back project the bounding boxes to the 3D map, we also explored how using the segmentation results of the 3D map can be used to identify the individuals more accurately compared to just back projecting the regions detected by the object detection algorithms.

While our method helps in accurately detecting stationary objects, many ecologically significant organisms moves freely (animals/fishes) or with the water/wind currents (grasses, algae). In our future work we plan to explore

Model	Batch Size	mAP20	mAP30	mAP40	mAP50	mAP60	mAP70	mAP80
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

Table 1. **2D Object Detection Results** Mean Average Precision of the 2D object detector models over a range of IoU's ($20\%, 30\%...80\%$)

Model	Merge Thr.	2D mAP	3D mAP	3D+Faces mAP	3D+Faces+CC mAP
Faster RCNN	0.6	40.42	15.38	25.05	25.41
RetinaNet	0.6	60.44	9.60	22.75	24.74
RetinaNet	0.8	60.44	14.76	27.91	24.23

Table 2. **3D Object Detection Results.** From Left to Right, 2D Object Detection Model, Merging Threshold used for merging multiple bounding boxes in the 3D reconstruction, mAP of 2D object detection model , mAP after back projecting 2D bboxes to 3D reconstruction, mAP after back projecting bboxes to 3D reconstruction and finding the min and max face centers, mAP after back projecting bboxes to 3D reconstruction and using nView Net results

Models	Merge Thr.	Method	Acropora palmata	Orbicella	Siderastrea	Porites astreoides	Gorgonia Ventalina	Sea Rods	Antillo Gorgia
Faster RCNN	0.6	2D mAP	51.02	9.33	67.82	42.75	36.21	24.24	51.58
Faster RCNN	0.6	3D mAP	20.33	6.25	50.42	0.68	7.10	6.26	16.63
Faster RCNN	0.6	3D+Faces mAP	23.67	6.25	50.42	42.61	26.04	2.57	23.77
Faster RCNN	0.6	3D+Faces+CC mAP	26.42	6.25	50.42	39.86	17.69	1.71	35.50
RetinaNet	N/A	2D mAP	87.43	46.16	63.51	62.95	61.59	36.00	65.45
RetinaNet	0.6	3D mAP	15.77	1.02	27.73	1.72	2.48	1.55	16.93
RetinaNet	0.6	3D mAP+Faces	20.89	2.70	37.26	51.03	21.11	7.43	18.86
RetinaNet	0.6	3D mAP+Faces+CC	22.26	4.22	60.40	42.14	10.75	4.05	29.36
RetinaNet	0.8	3D mAP	26.00	6.56	36.07	3.51	25.65	4.65	21.72
RetinaNet	0.8	3D+Faces mAP	30.55	12.35	36.07	46.12	25.65	8.02	25.09
RetinaNet	0.8	3D+Faces+CC mAP	25.96	12.35	51.66	32.4	12.14	3.69	31.41

Table 3. **3D Class Results.** From Left to Right, Object Detection Model, method used and per class results

deep learning approaches to enable better detection of these non stationary objects in the 3D reconstruction of the coral reef.

References

[1] Kenneth R.N. Anthony. 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, 2016.

[2] Oscar Beijbom, Peter J. Edmunds, David I. Kline, B. Greg Mitchell, and David Kriegman. Automated annotation of coral reef survey images. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1170–1177, 2012.

[3] Gary Bradski and Adrian Kaehler. *Learning OpenCV, [Computer Vision with OpenCV Library ; software that sees]*. O'Reilly Media, 1. ed. edition, 2008. Gary Bradski and Adrian Kaehler.

[4] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn, 2018.

[5] John D. Hedley, Chris M. Roelfsema, Iliana Chollett, Alastair R. Harborne, Scott F. Heron, Scarla Weeks, William J. Skirving, Alan E. Strong, C. Mark Eakin, Tyler R. L. Christensen, Victor Ticzon, Sonia Bejarano, and Peter J. Mumby. Remote sensing of coral reefs for monitoring and management: A review. *Remote Sensing*, 8(2), 2016.

[6] J. Heikkila and O. Silven. A four-step camera calibration procedure with implicit image correction. In *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1106–1112, 1997.

[7] O Hoegh-Guldberg, PJ Mumby, AJ Hooten, RS Steneck, P Greenfield, E Gomez, CD Harvell, PF Sale, AJ Edwards, K Caldeira, N Knowlton, CM Eakin, R Iglesias-Prieto, N Muthiga, RH Bradbury, A Dubi, and ME Hatziolos. Coral reefs under rapid climate change and ocean acidification. *Science*, 318(5857):1737– 1742, 2007.

[8] Andrew King, Suchendra M. Bhandarkar, and Brian M. Hopkinson. A comparison of deep learning methods for semantic segmentation of coral reef survey images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2018.

[9] Andrew King, Suchendra M.Bhandarkar, and Brian M. Hopkinson. Deep learning for semantic segmentation of coral reef images using multi-view information. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2019.

[10] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'12, page 1097–1105, Red Hook, NY, USA, 2012. Curran Associates Inc.

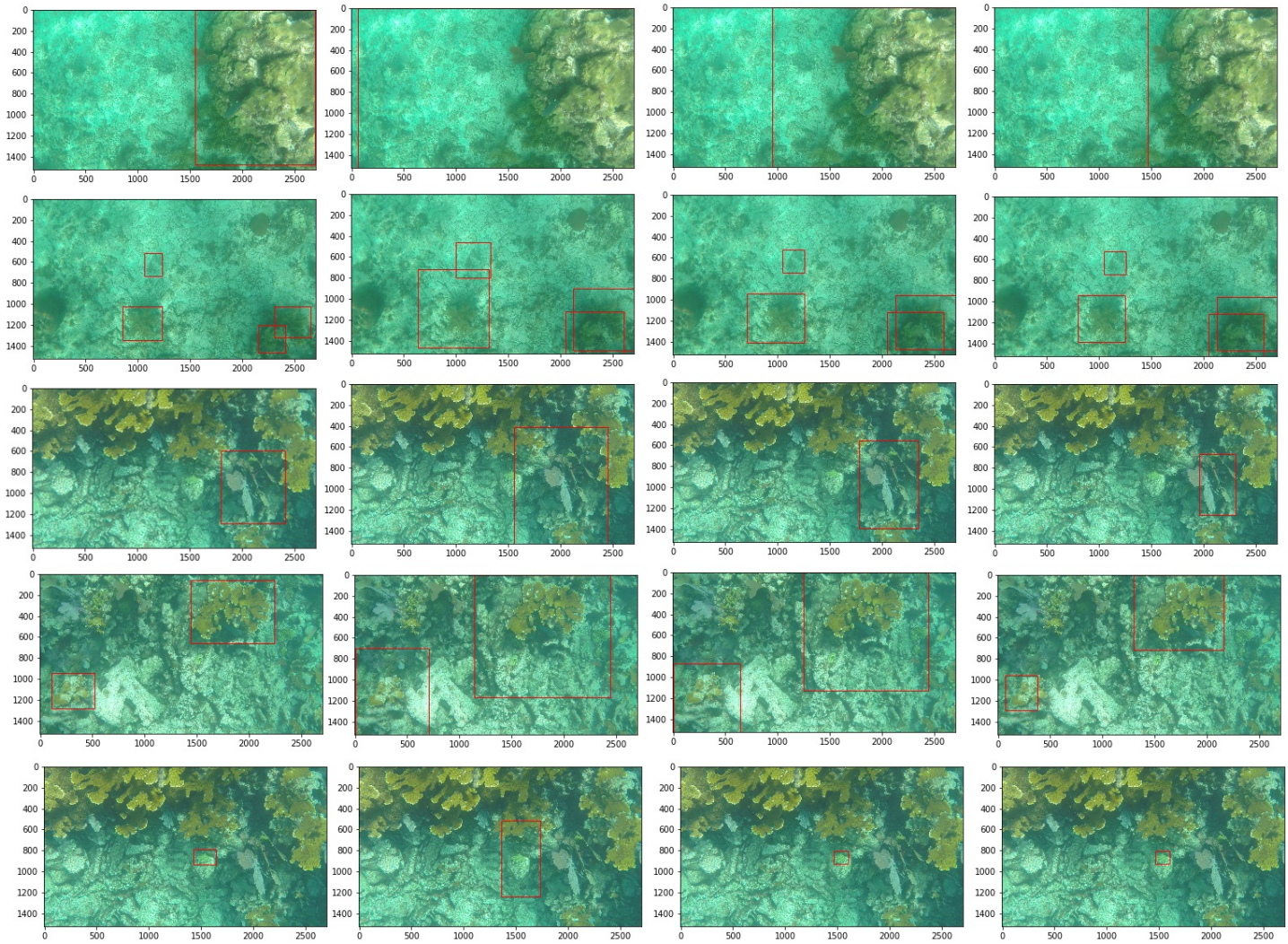

Figure 6. **Resulting Images**from top to bottom, Example Results of classes Orbicella, Antillo Gorgia, Gorgonia Ventalina, Acropora palmata, Porites astreoides respectively. From Left to Right, Result of 2D object detection model, Result after back projecting bboxes to 3D reconstruction, Result after back projecting bboxes to 3D reconstruction and finding the min and max face centers, Result after back projecting bboxes to 3D reconstruction and using nView Net results

[11] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553):436–444, 2015.

[12] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection, 2018.

[13] Tomas Möller and Ben Trumbore. Fast, minimum storage ray-triangle intersection. *Journal of Graphics Tools*, 2(1):21–28, 1997.

[14] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection, 2016.

[15] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks, 2016.

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