Proceedings of the 2018 IEEE
International Conference on Robotics and Biomimetics
December 12-15, 2018, Kuala Lumpur, Malaysia

Coral Identification and Counting with an Autonomous Underwater Vehicle*

Md Modasshir[1], Sharmin Rahman[1], Oscar Youngquist[2], and Ioannis Rekleitis[1]

***Abstract*— Monitoring coral reef populations as part of environmental assessment is essential. Recently, many marine science researchers are employing low-cost and power efficient Autonomous Underwater Vehicles (AUV) to survey coral reefs. While the counting problem, in general, has rich literature, little work has focused on estimating the density of coral population using AUVs. This paper proposes a novel approach to identify, count, and estimate coral populations. A Convolutional Neural Network (CNN) is utilized to detect and identify the different corals, and a tracking mechanism provides a total count for each coral species per transect. Experimental results from an Aqua2 underwater robot and a stereo hand-held camera validated the proposed approach for different image qualities.**

I. INTRODUCTION

Coral reefs are an essential part of the marine ecosystem and support a rich diversity of life [1], [2] with significant economic value and social amenity. Projected increases in global temperatures of $2 - 4.5°C$ by 2100 [3] indicate that mass coral bleaching events are likely to become an annual phenomenon by 2050 [4], [5]. The widespread mortality of corals following mass bleaching events reduces the structural complexity of reefs, thus eliminating the 3-D habitat. This habitat loss affects diversity and population of coral reef fish and invertebrates communities adversely. Therefore, the monitoring of health of coral ecosystems by scuba divers and new technologies, such as underwater robots [6], has become increasingly significant. As a result, millions of images are being collected. While image acquisition has evolved, reef monitoring still requires the identification and counting of different coral species, a task primarily performed by human experts.

Underwater and surface autonomous vehicles have been used for a variety of monitoring tasks, mainly focusing coral reef inspections, [7]–[9] even in deep waters [10]. Furthermore, floating cameras have also been employed [11], [12] to collect visual data with reduced cost. In this paper we have utilized an Aqua2 [13] Autonomous Underwater Vehicle (AUV) [14] and a hand-held stereo GoPro camera[1].

Object counting is an active research field, and in particular, the coral counting problem is challenging for many reasons. Firstly, visibility, color suppression and hazing underwater make detection extremely difficult. Objects within the field of view are often so obscure that both deep models and humans cannot identify. Moreover, coral counting encompasses both spatial and temporal domains. Once an object is detected, it is imperative to prevent recount after detection in the subsequent image frames. Finally, performing detection and counting on a low-powered AUV poses significant constraints on the choice of the detection model and the frequency of the detection process.

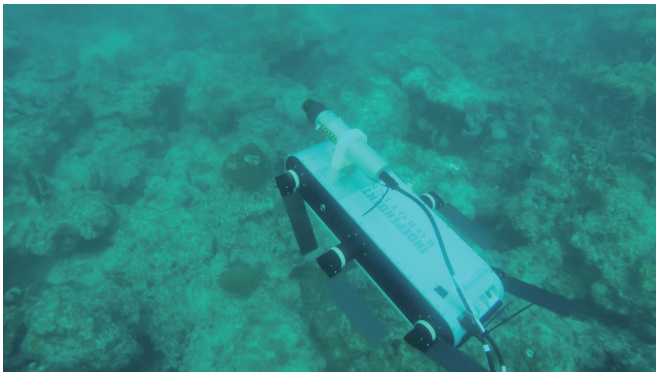

Fig. 1. Aqua2 AUV collecting visual and acoustic data over a coral reef, Barbados.

To analyze coral reef visual data, marine biologists cover certain transects, such as straight lines or rectangles, over the coral reef. Afterwards, domain experts analyze the video to count or annotate coral species to estimate population density. This paper proposes a novel technique to automate this process. In this work we have slightly modified a recent deep learning network, RetinaNet [15], to account for the much-reduced number of training examples, in the presence of high-class imbalance among coral samples in the dataset. The modified network identifies and localizes different coral species in an image. The complete system has a constant stream of images as input, either from an AUV or a hand-held camera; images are fed into the Convolutional Neural Network (CNN), and different coral species are identified and localized in the image by a bounding box. Then each bounding box is tracked in successive images using the KCF tracker [16] from OpenCV[2]. If a new detection occurs in the vicinity of a tracked bounding box, then the bounding box coordinates are updated, but the coral count does not increase.

It is worth noting that the proposed methodology for coral identification and counting, even when performed off-line,

*This work was supported by NSF award 1513203.

[1]M. Modasshir, S. Rahman, and I. Rekleitis are with Computer Science and Engineering Department, University of South Carolina, USA. [modssshm,srahman]@email.sc.edu, yiannisr@cse.sc.edu

[2] O. Youngquist is with the Department of Computer Science and Software Engineering, Rose-Hulman Institute of Technology, USA. youngqom@rose-hulman.edu

[1]http:\\www.gopro.com

[2]http://www.opencv.org

978-1-7281-0377-8/18/$31.00 © 2018 IEEE

automates the process of identifying the biodiversity for a fixed trajectory using data collected by a robot. Different hardware architectures have been evaluated [17] to port the detection and tracking system on-line, a task that is beyond the scope of this paper.

Experimental verification over two datasets collected at the coral reefs of Barbados has demonstrated the accuracy of the proposed system. To our knowledge, there has not been any other work on automated coral counting.

The next section provides an overview of related work. Section III describe the proposed methodology for autonomous coral identification and counting. Experimental results from an AUV and a hand-held stereo camera are discussed in Section IV. We conclude this paper with a discussion of experimental results and directions for future research.

II. Related Work

A large number of algorithms has been proposed to tackle the counting problem in the visual domain. These algorithms mostly fall into either regression-based methods or detection based methods. Regression-based methods [18]–[23] use CNN based models to predict the number of objects in the image without explicitly classifying and localizing the objects. Recent works on regression based methods generate density maps for image patches and later integrate over the map to produce the object count. However, most of these regression based methods aimed at solving counting the problem in a single image, thus have no mechanism for tracking over successive images. On the other hand, in detection based methods [24]–[26], a detector (usually CNN based) is utilized to localize the objects in the image with bounding boxes. These bounding boxes are counted to estimate the number of objects present in the image. In CNN based detectors, there exist two distinct approaches: two-stage detectors (region proposal based) and one-stage detectors. In the two-stage approach, the first stage generates a sparse set of object proposals and the second stage classifies the proposals into foreground classes and background. R-CNN [27] significantly improved two-stage detectors using a CNN to generate proposals. The Faster-RCNN [24] integrated the region proposal generation and the proposal classification into a single CNN, achieving higher speed and accuracy gains in object detection. On the other hand, one stage detectors uses Feature Pyramid Networks [28] to enable object detection at multiple scale. In one-stage detectors, OverFeat [29], SSD [30], [31] and YOLO [25], [32] showed tremendous speed improvements but with accuracy trade-off. Even with large computational resource available, the single stage detectors trail in accuracy behind the two stage detectors.

Recent work by Lin et al. [15] significantly advanced the one-stage detection network matching state-of-the-art results of the two-stage detectors. Lin et al.indicated that the class imbalance during training of one-stage detectors is the main reason for low accuracy performance. This class imbalance is usually handled in two-stage detectors by using different sampling heuristics such as a fixed foreground-to-background ratio or On-Line Hard Example Mining (OHEM). To better train detector models in the presence of class imbalance, Lin et al. [15] proposed a new loss function that is a dynamically scaled cross entropy loss. Our work is inspired by the model proposed by Lin et al. [15] as it closely matches our problem domain.

Visual tracking is an active research area in computer vision. Numerous works utilize correlation filter (CF) for robust visual tracking. The popularity of CF based trackers are due to their rapid speed and efficient learning techniques. With low computational load, CF can learn a large number of samples. The early CF-based trackers used a single channel feature as input with tremendous tracking speed. The MOSSE [33] tracker exploited adaptive correlation filter. Henriques et al. [34] introduced kernel trick in the correlation filter formula. Later, Henriques et al. [16] further improved CF by integrating multi-channel input and introduced KCF. Inspired by the improvement on multi-channel correlation filters, many CF based trackers employing deep learning features [35]–[38] achieved state-of-the-art performance. However, even with GPU acceleration, most of these CNN based trackers cannot track in real-time (15-30 fps) which is a requirement for tracking and counting coral objects in AUVs.

III. Methodology

Automated collection of visual data has an advantage that images are acquired in a sequence, so the location of the camera together with the structure of the scene can be recovered. Therefore, a coral detected in an image can be effectively tracked over successive images, while in the field of view, and not over-counted. Furthermore, tracking the location of the different corals reduces the rate of detection, resulting in a more efficient algorithm.

There are two steps to automate the coral population estimation process: detection and tracking. Detection of coral species allows the localization of coral objects in the image. Later, tracking is essential to prevent the recount in subsequent detection. Our system integrates both of these steps as shown in Fig. 2. When processing a stream of images from either Aqua2 or hand-held camera, the first frame, f_0 is fed to the detection model which produces labels and bounding boxes for coral objects in the image. These bounding boxes are used as the Region Of Interest (ROI) to initialize a tracker that keeps updating the location of those detected coral species in frames f_1 to f_{n-1}. The system keeps count of the ROIs with labels. Then the coral detector model runs again on frame f_n and produces another set of bounding boxes (ROIs) with labels. At this point, the ROIs are compared with the earlier tracked ROIs and only the new ROIs with overlap (intersection over union) less than a threshold are counted as new coral objects and tracked in the subsequent frames along with earlier ROIs. This step ensures that no coral is counted more than once. For the coral detector model, we utilize a deep convolutional neural network inspired by the RetinaNet model [15], and as

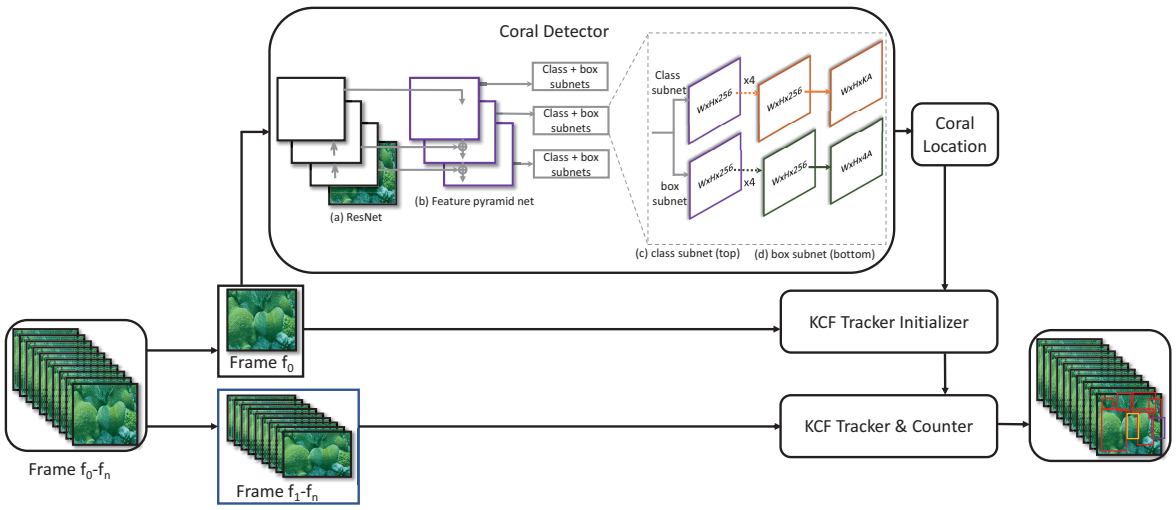

Fig. 2. Proposed system with detection and tracking. For n frames, detection is performed on f_0. Then jointly f_0 and detected object bounding boxes are used to initialize the tracker. From Frames f_1 to f_{n-1} are sent only to the tracker to update object locations.

a tracker, we use the KCF [16]. Next we will explain the coral detector model and the KCF tracker.

A. *RetinaNet*

RetinaNet [15] is a one-stage detector comprising a backbone network and two subnetworks. The two subnetworks are used for classification and bounding box regression. Using the output of the backbone network the first subnet computes class confidence and the second subnet regresses bounding box coordinates. We choose the Feature Pyramid Network (FPN) [28] as the backbone network. The FPN creates multi-scale features from a single resolution image by augmenting a standard CNN with top-down pathway and lateral connections as shown in Fig. 2. This multi-scale feature pyramid allows any level of features to be used to detect objects at a different scale. We choose ResNet50 with 50 layers, a variant of deep Residual Networks [39] as the base network for the FPN. We redesigned the final layers for RetinaNet to reflect the eight classes of interested.

Focal Loss: We choose focal loss [15] to optimize classification subnet. Focal loss is designed to facilitate training under high dataset imbalance. For estimated probability $p_t \in [0, 1]$ for a target class t, the focal loss is defined as

$$loss(p_t) = -\alpha\,(1 - p_t)^{\gamma}\,log(p_t)$$

where $\gamma \geq 0$ is a tunable focusing parameter and $\alpha \in [0, 1]$ is a weighting factor. The focal loss assigns a lower loss to easily classified examples and focuses more on the misclassified data. Therefore, the network is better tuned to recognize difficult samples. Moreover, γ parameter controls the down-weighting of easy examples smoothly. The weighting factor α is chosen as the inverse of samples for classes in our dataset.

Smooth L1 Loss: For bounding box regression, smooth L1 loss is used. For bounding box prediction t and v, the loss is defined as

$$Loss_{bbox} = \sum_{i \in x,y,w,h} smooth_{L_1}(t_i - v_i)$$

where

$$smooth_{L_1}(x) = \begin{cases} 0.5\,x^2, & \text{if } |x| < 1. \\ |x| - 0.5, & \text{otherwise.} \end{cases} \quad (1)$$

Training: We initialize the ResNet50 [39] network with the weights trained on imagenet [40], and all other layers are randomly initialized to zero-mean Gaussian distribution with a standard deviation of 0.01. Stochastic gradient descent was used to optimize the parameters. The network is trained for 150 iterations on our dataset with an initial learning rate of 0.001 and a decay of $1e^{-5}$.

B. *KCF Tracker:*

The KCF [16] tracker improves generic correlation filter by using multiple channel data of color images. Let $y = [y_1, y_2, ..., y_k]^T \in \mathbb{R}$ be the Gaussian shaped response and let $x_d \in \mathbb{R}^{k \times 1}$ be the input vector. The correlation filter learns filter weights w by optimizing:

$$\dot{w} = arg\,\min_{w} \sum_{k=1}^{K} (y_k - \sum_{d=1}^{D} X_{k,d}^T W_d)_2^2 + \lambda \left\| W \right\|_2^2 \quad (2)$$

where $X_{k,d}$ is the k-step circular shift of the input vector X_d, y_k is the k-th element of y, $W = [W_1^T, W_2^T, ..., W_D^T]^T$ where $W_d \in \mathbb{R}^{K*1}$ refers to the filter of the d-th channel [41].

IV. Experimental Results and Discussion

For this paper, we have selected different approximately straight line trajectories over open areas with sparse live coral populations. We have trained our network to detect seven different kinds of corals (Brain, Maze, Mustard, Finger, Fire, Star, and Starlet) and sponges. The training samples are annotated from several other underwater videos. Using the system described above each sequence of the images is fed

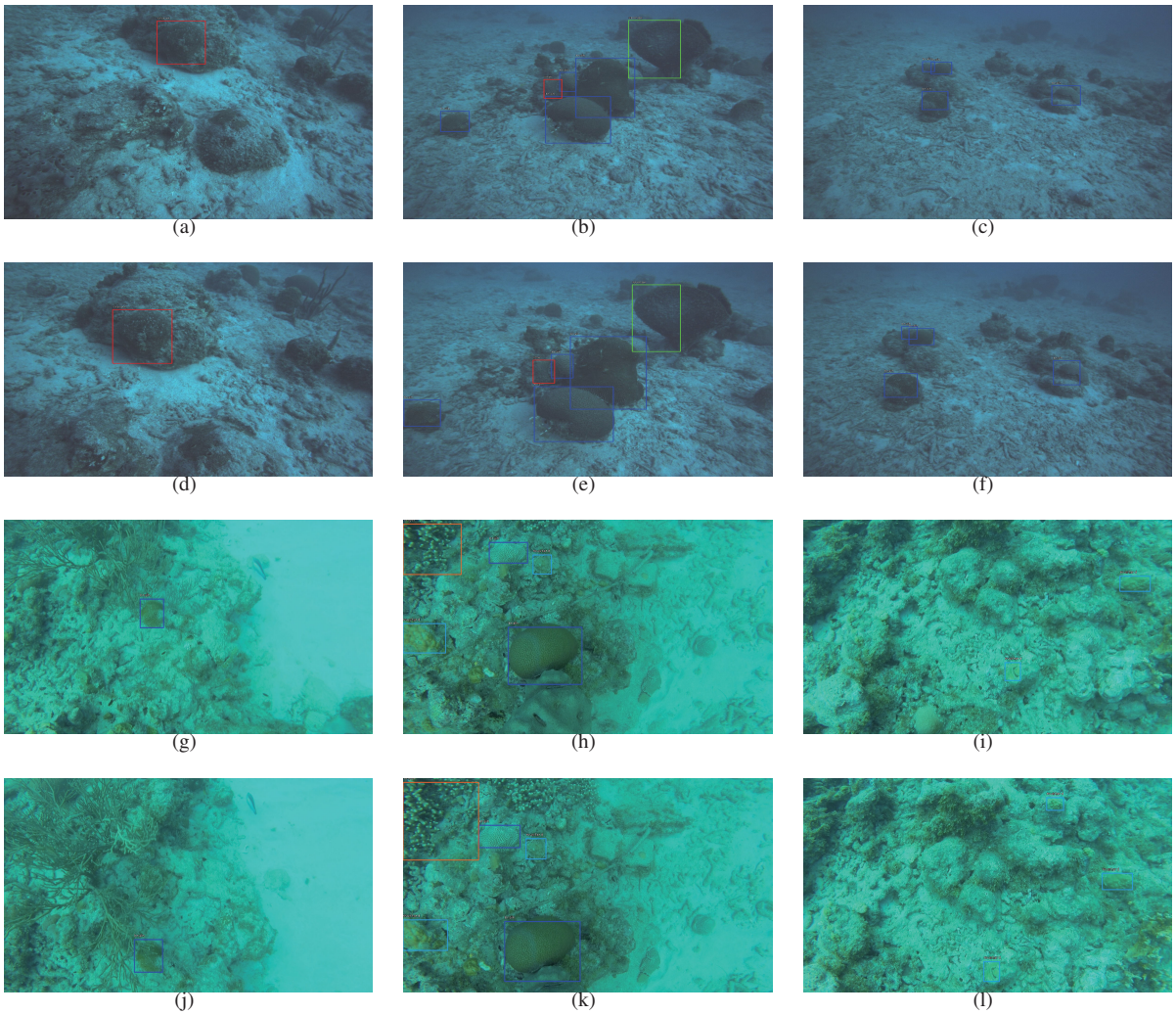

Fig. 3. Examples of different coral detections. First two rows ((a)-(f)) are collected with an Aqua2 AUV; second two rows ((g)-(l)) using a stereo GoPro camera. Each two rows are separated by a couple of seconds: (a),(d) Single brain coral detection. (b),(e) Multiple corals (brains, and star) and a sponge detected. (c),(f) Multiple brain corals detected in a single image. (g),(j) Single brain coral detection. (h),(k) Multiple corals (brain, mustard, and finger) detected. (i),(l) Multiple mustard corals detected.

	Brain	Mustard	Finger	Star	Starlet	Maze	Fire	Sponge
T1:GoPro	173/197	125/147	15/9	1/3	11/18	15/20	3/3	2/3
T1:Aqua2	57/66	5/9	0/0	0/0	2/2	0/0	0/0	12/9

TABLE I. CORAL IDENTIFICATION AND COUNTING FOR DIFFERENT TRAJECTORIES. CNN-PREDICTION/HUMAN-ANNOTATED

into the tracking system, and the total count for each class is reported at the end.

Fig. 3 present representative images with different corals tracked. The first two rows are images collected by the Aqua2 AUV in two different time instances. As can be seen in Fig. 3(a) a single Brain coral is detected at the top of the image; on the second row, Fig. 3(d), the same coral is tracked when the AUV approached closer. The first column presents a single coral tracked over several successive frames (only two displayed). In the second column, several corals from different classes are tracked, while the last column displays tracking of several different corals belonging to the same class. More specifically, Fig. 3(b) and Fig. 3(e) present the detection and tracking of Brain and Star corals and a large sponge (three classes); and Fig. 3(c) and Fig. 3(f) display the tracking of multiple Brain corals. The images collected from the AUV have lower resolution and display a certain amount of blurriness. However, identification and tracking of different coral species were feasible.

The last two rows of Fig. 3 presents underwater images from a GoPro camera over a coral reef. The image resolutions are different, and the image quality is relatively higher. The three columns contain results from single coral, multiple corals from different classes, and multiple corals from the same class, respectively. A single brain coral is detected in Fig. 3(g) and then tracked across the image in Fig. 3(j). Multiple corals belonging to the Brain, Mustard, and Finger coral classes are tracked in Fig. 3(h) and Fig. 3(k). Finally,

multiple corals all belonging to the Mustard coral class are tracked in Fig. 3(i) and Fig. 3(l).

While most corals were localized, corals belonging to the Finger coral presented a challenge as they are usually distributed over a much larger area. If there are patches of dead coral inside a large patch of Finger corals, then the patch is counted as two different occurrences. Future work will specifically target finger corals and how to identify all the corals that belong to the same patch.

Quantitative results are presented in Table I for two sample trajectories. For each of the seven coral species (Brain, Maze, Mustard, Finger, Fire, Star, and Starlet), the number of corals detected by our system against the number of corals annotated by a human is presented, and the same numbers are presented for the sponge class. The estimated number of individual coral population is reasonably close to the human annotated numbers for corals and sponge for hand-held camera. For Aqua2 transact, there is close matches with ground truth. We did not have any finger coral in the Aqua2 transact. However, for finger coral, there are more predictions than actual instances. On the speed performance, the system can run at 15 fps which will allow online detection and tracking in Aqua2.

In our empirical analysis, there were fewer mispredictions and the difference in number of coral species between predicted and ground truth are due to detector's incapability to detect some coral species. Furthermore, although KCF performed reasonably well, there were some instances of track losses, which resulted in over-counting of several coral objects. The detector model could be further be improved by utilizing more data in the training process. Using a better tracker will certainly improve the performance of the system. We observed that for such system of counting, the ability to correctly identify at close range is more important than the ability to detect all coral objects within field of view for a detector model. This is because once detected, a coral object is being tracked and counted, therefore, earlier or subsequent detection failure does not hamper the counting process. The number of frames between subsequent detection plays crucial role here. For hand-held camera which has more clarity and higher resolution, 10 frames between detection provided good result as tracking worked reasonably well. However, for Aqua2 which has lower resolution, less illumination, and less clear images, 5 frames between detection was chosen and more than 5 frames reduced the counting performance because of frequent tracking failure.

One future direction of this work is utilize vision-based state estimation information to prevent recount. State estimation is still a very challenging problem [42] especially underwater. Fig. 4 sketches the main idea of utilizing state estimation in counting corals, in which the AUV moves over the coral reef; inertial, visual, depth, and acoustic data are collected. Depending on the choice of software, all or a subset of the data are used to estimate the pose of the cameras and over time. From the pose of the camera for each image, a simple projection can identify the 3D position of each detected coral. Therefore, detected corals from different images are distinguished if they are projected in 3D locations that are sufficiently apart.

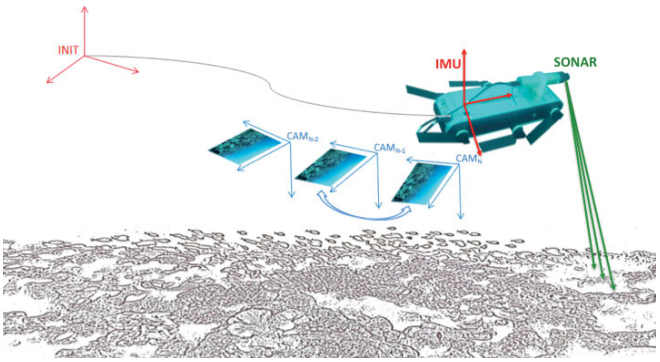

Fig. 4. Sonar, Visual, Inertial Estimation underwater.

V. Conclusions

In this paper, we presented an automated system for identifying and counting the different corals encountered by an AUV. A modification of a popular CNN architecture [15] ensures improved performance with a limited dataset of eight classes. The current performance of the proposed system ensures near real-time performance on state of the art GPU machines, without any optimization. Ongoing work on porting the algorithm on a portable system such as the NVidia TX2 or the Intel Neural Compute Stick will enable the on-line deployment. The system as it is can achieve 12 fps on NVidia TX2.

Tighter integration of the camera position into the tracking algorithm will enable the full 3D localization of the different corals and prevent double counting when a coral exits and then re-enters the camera's field of view. Furthermore, integrating a dense reconstruction of the observed corals will enable the study of the structural complexity of reefs. Modeling the 3D habitat, which is critical to maintaining diversity and population of coral reef fish communities, will allow for better health assessment of the reef ecosystem.

References

[1] C. Rogers, G. Garrison, R. Grober, Z. Hillis, and M. F.ie, "Coral reef monitoring manual for the caribbean and western atlantic," *Virgin Islands National Park, 110 p. Ilus.*, 1994.

[2] "Corals of the world - variation in species," http://coral.aims.gov.au/info/taxonomy-variation.jsp, (Accessed on 11/30/2017).

[3] IPCC, "The physical sciences basis: Contribution of working group i to the fourth assessment report of the intergovernmental panel on climate change," Tech. Rep., 2007.

[4] G. Hodgson and J. Liebeler, *The Global Coral Reef Crisis: Trends and Solutions*. Reef Check Foundation, 2002.

[5] M. Mulhall, "Saving the rainforests of the sea: An analysis of international efforts to conserve coral reefs," *Duke Environmental Law and Policy Forum*, vol. 19, pp. 321–351, 2007.

[6] S. B. Williams and I. Mahon, "Design of an unmanned underwater vehicle for reef surveying," *IFAC Proceedings Volumes*, vol. 37, no. 14, pp. 175–180, 2004.

[7] C. Beall, B. Lawrence, V. Ila, and F. Dellaert, "3d reconstruction of underwater structures," in *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, Oct 2010, pp. 4418–4423. [Online]. Available: https://smartech.gatech.edu/bitstream/handle/1853/38324/Beall10iros.pdf

[8] P. Giguere, G. Dudek, C. Prahacs, N. Plamondon, and K. Turgeon, "Unsupervised learning of terrain appearance for automated coral reef exploration," in *Canadian Conference on Computer and Robot Vision*, Kelowna, British Columbia, May 2009.

[9] H. Singh, R. Armstrong, F. Gilbes, R. Eustice, C. Roman, O. Pizarro, and J. Torres, "Imaging coral i: imaging coral habitats with the seabed auv," *Subsurface Sensing Technologies and Applications*, vol. 5, no. 1, pp. 25–42, 2004.

[10] M. Grasmueck, G. P. Eberli, D. A. Viggiano, T. Correa, G. Rathwell, and J. Luo, "Autonomous underwater vehicle (auv) mapping reveals coral mound distribution, morphology, and oceanography in deep water of the straits of florida," *Geophysical Research Letters*, vol. 33, no. 23, 2006.

[11] R. Hoeke, J. Gove, E. Smith, P. Fisher-Pool, M. Lammers, D. Merritt, O. Vetter, C. Young, K. Wong, and R. Brainard, "Coral reef ecosystem integrated observing system: In-situ oceanographic observations at the us pacific islands and atolls," *Journal of Operational Oceanography*, vol. 2, no. 2, pp. 3–14, 2009.

[12] D. Boydstun, M. Farich, J. M. III, S. Rubinson, Z. Smith, and I. Rekleitis, "Drifter sensor network for environmental monitoring," in *Conference on Computer Robot Vision (CRV)*, Halifax, NS, Canada, Jun. 2015, pp. 16–22.

[13] G. Dudek, M. Jenkin, C. Prahacs, A. Hogue, J. Sattar, P. Giguere, A. German, H. Liu, S. Saunderson, A. Ripsman, S. Simhon, L. A. Torres-Mendez, E. Milios, P. Zhang, and I. Rekleitis, "A visually guided swimming robot," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, Aug. 2-6 2005, pp. 1749–1754.

[14] J. Sattar *et al.*, "Enabling autonomous capabilities in underwater robotics," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2008, pp. 3628–3634.

[15] T.-Y. L. P. G. Ross and G. K. H. P. Dollár, "Focal loss for dense object detection."

[16] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista, "High-speed tracking with kernelized correlation filters," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 3, pp. 583–596, 2015.

[17] M. Modasshir, A. Q. Li, and I. Rekleitis, "Deep neural networks: a comparison on different computing platforms," in *Conference on Computer and Robot Vision*, Toronto, ON, Canada, May 2018, pp. 383–389.

[18] L. Boominathan, S. S. Kruthiventi, and R. V. Babu, "Crowdnet: A deep convolutional network for dense crowd counting," in *Proceedings of the 2016 ACM on Multimedia Conference*. ACM, 2016, pp. 640–644.

[19] V. Lempitsky and A. Zisserman, "Learning to count objects in images," in *Advances in neural information processing systems*, 2010, pp. 1324–1332.

[20] V.-Q. Pham, T. Kozakaya, O. Yamaguchi, and R. Okada, "Count forest: Co-voting uncertain number of targets using random forest for crowd density estimation," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 3253–3261.

[21] B. Xu and G. Qiu, "Crowd density estimation based on rich features and random projection forest," in *Applications of Computer Vision (WACV), 2016 IEEE Winter Conference on*. IEEE, 2016, pp. 1–8.

[22] D. B. Sam, S. Surya, and R. V. Babu, "Switching convolutional neural network for crowd counting," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, vol. 1, no. 3, 2017, p. 6.

[23] Y. Zhang, D. Zhou, S. Chen, S. Gao, and Y. Ma, "Single-image crowd counting via multi-column convolutional neural network," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 589–597.

[24] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Advances in neural information processing systems*, 2015, pp. 91–99.

[25] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 779–788.

[26] C. Gao, P. Li, Y. Zhang, J. Liu, and L. Wang, "People counting based on head detection combining adaboost and cnn in crowded surveillance environment," *Neurocomputing*, vol. 208, pp. 108–116, 2016.

[27] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Computer Vision and Pattern Recognition*, 2014.

[28] T.-Y. Lin, P. Dollár, R. B. Girshick, K. He, B. Hariharan, and S. J. Belongie, "Feature pyramid networks for object detection." in *CVPR*, vol. 1, no. 2, 2017, p. 4.

[29] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun, "Overfeat: Integrated recognition, localization and detection using convolutional networks," *arXiv preprint arXiv:1312.6229*, 2013.

[30] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "Ssd: Single shot multibox detector," in *European conference on computer vision*. Springer, 2016, pp. 21–37.

[31] C.-Y. Fu, W. Liu, A. Ranga, A. Tyagi, and A. C. Berg, "Dssd: Deconvolutional single shot detector," *arXiv preprint arXiv:1701.06659*, 2017.

[32] J. Redmon and A. Farhadi, "Yolo9000: better, faster, stronger," *arXiv preprint*, 2017.

[33] D. S. Bolme, J. R. Beveridge, B. A. Draper, and Y. M. Lui, "Visual object tracking using adaptive correlation filters," in *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*. IEEE, 2010, pp. 2544–2550.

[34] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista, "Exploiting the circulant structure of tracking-by-detection with kernels," in *European conference on computer vision*. Springer, 2012, pp. 702–715.

[35] M. Danelljan, G. Bhat, F. S. Khan, and M. Felsberg, "Eco: Efficient convolution operators for tracking." in *CVPR*, vol. 1, no. 2, 2017, p. 3.

[36] M. Danelljan, G. Hager, F. Shahbaz Khan, and M. Felsberg, "Convolutional features for correlation filter based visual tracking," in *Proceedings of the IEEE International Conference on Computer Vision Workshops*, 2015, pp. 58–66.

[37] M. Danelljan, A. Robinson, F. S. Khan, and M. Felsberg, "Beyond correlation filters: Learning continuous convolution operators for visual tracking," in *European Conference on Computer Vision*. Springer, 2016, pp. 472–488.

[38] D. Onoro-Rubio and R. J. López-Sastre, "Towards perspective-free object counting with deep learning," in *European Conference on Computer Vision*. Springer, 2016, pp. 615–629.

[39] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

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

[41] C. Sun, D. Wang, H. Lu, and M.-H. Yang, "Correlation tracking via joint discrimination and reliability learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 489–497.

[42] A. Q. Li, A. Coskun, S. M. Doherty, S. Ghasemlou, A. S. Jagtap, M. Modasshir, S. Rahman, A. Singh, M. Xanthidis, J. M. O'Kane, and I. Rekleitis, "Experimental comparison of open source vision based state estimation algorithms," in *International Symposium of Experimental Robotics (ISER)*, Tokyo, Japan, Mar. 2016.