

# UIVNAV: Underwater Information-driven Vision-based Navigation via Imitation Learning

Xiaomin Lin<sup>1\*</sup>, Nare Karapetyan<sup>2\*</sup>, Kaustubh Joshi<sup>1</sup>, Tianchen Liu<sup>1</sup>, Nikhil Chopra<sup>1</sup>,  
Miao Yu<sup>1</sup>, Pratap Tokekar<sup>1</sup>, Yiannis Aloimonos<sup>1</sup>

**Abstract**—Autonomous navigation in the underwater environment is challenging due to limited visibility, dynamic changes, and the lack of a cost-efficient, accurate localization system. We introduce UIVNAV, a novel end-to-end underwater navigation solution designed to navigate robots over Objects of Interest (OOI) while avoiding obstacles, all without relying on localization. UIVNAV utilizes imitation learning and draws inspiration from the navigation strategies employed by human divers, who do not rely on localization. UIVNAV consists of the following phases: (1) generating an intermediate representation (IR) and (2) training the navigation policy based on human-labeled IR. By training the navigation policy on IR instead of raw data, the second phase is *domain-invariant* — the navigation policy does not need to be retrained if the domain or the OOI changes. We demonstrate this within simulation by deploying the same navigation policy to survey two distinct Objects of Interest (OOIs): oyster and rock reefs. We compared our method with complete coverage and random walk methods, showing that our approach is more efficient in gathering information for OOIs while avoiding obstacles. The results show that UIVNAV chooses to visit the areas with larger area sizes of oysters or rocks with no prior information about the environment or localization. Moreover, a robot using UIVNAV compared to complete coverage method surveys on average 36% more oysters when traveling the same distances. We also demonstrate the feasibility of real-time deployment of UIVNAV in pool experiments with BlueROV underwater robot for surveying a bed of oyster shells.

## I. INTRODUCTION

Underwater navigation is a challenging task that has been the subject of numerous research studies [1]. In particular, interest in marine exploration and scientific sampling applications [2]–[4] has significantly increased the demand for efficient underwater navigation methods for exploration and data gathering. Many classical navigation methods developed for ground and aerial vehicles have been adapted to underwater environments [5], [6]. Navigation in an underwater environment presents unique difficulties, such as limited visibility, dynamic environments, and a lack of cost-efficient, accurate localization systems. Underwater vehicles are commonly used for data collection to monitor environmental phenomena. To improve the efficiency of data collection, several complete coverage path planning methods

This work was supported by USDA NIFA sustainable agriculture system program under award number 20206801231805.

\* Equal Contributors and Corresponding Authors.

<sup>1</sup>Maryland Robotics Center (MRC), University of Maryland, College Park, MD 20742, USA. Emails: {xlin01, kjoshi, tianchen, nchopra, mmyu, tokekar, jyaloimo}@umd.edu.

<sup>2</sup>Woods Hole Oceanographic Institution (WHOI), Woods Hole, MA 02543, USA. Emails: nare@whoi.edu

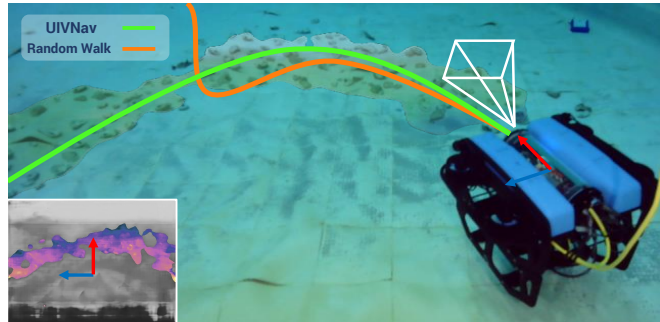


Fig. 1. Example of two trajectories of BlueROV surveying oyster-reef in a pool. On the bottom left is the current frame in the intermediate representation that the robot observes.

have been developed [7], [8] and adapted to underwater environments [9]–[11]. However, these approaches typically rely on partial or complete knowledge about the environment or assume the robot can localize itself.

The absence of reliable localization [12] requires new approaches for tackling underwater navigation or exploration problems. Towards this, vision-only based methods have been proposed for surveying coral reefs [13] and shipwrecks [14]. Nevertheless, these approaches are domain-dependent as they have been trained over specific domains and objects and thus require retraining navigation policy each time a new Object of Interest (OOI) is being surveyed.

We propose a new Underwater Information-driven Vision-based Navigation via Imitation Learning (UIVNAV) method to address some of these challenges. Our method focuses on information gathering as the main objective — we want to visit regions with OOI while avoiding obstacles. UIVNAV is a type of imitation learning that uses only RGB camera information and generates an Intermediate Representation (IR) of each frame, which humans then use to label corresponding actions. IR allows us to train domain-invariant navigation policies to work across different OOI. Moreover, this method does not rely on localization but only on IR. We have compared our method with baseline navigation approaches – random walk and complete coverage. The results indicate that UIVNAV outperforms random walk and even surpasses localization-based complete coverage approach by surveying on average 36% more oysters when traveling the same distances.

The main contributions of this work are:

- 1) The development of a domain-invariant navigation approach. This means that the navigation can gather

information regardless of the type of domain being presented, reducing the need for retraining the navigation policy for new environments. We show this by deploying UIVNAV on two different OOI oysters and large rock reefs, and in different domains the simulation and the pool. This provides a more generalized solution that can be applied in a wide range of underwater environments.

- 2) A robot learns how to navigate successfully while covering (“seeing”) as much of the OOI as possible — without relying on localization. This is an example of a new local navigation technique that would aid a class of global exploration and coverage algorithms. As navigation computational infrastructures (e.g., SLAM and related techniques) are reaching a high level of sophistication, the need arises to perform navigation and solve various tasks while doing so.
- 3) Deployment of the proposed method in real-time in a large pool for surveying an oyster shell bed — demonstrating the feasibility of real-world application of UIVNAV (Fig. 1).

The rest of this paper is organized in the following manner. First, in Section II, we discuss the recent advances in the field. The problem formulation and proposed approach are presented in Section III. Experimental validation in simulation and lab environment are discussed in Section IV. Finally, a conclusion with a discussion on possible future work is presented in Section V.

## II. RELATED WORK

The problem of underwater navigation has been extensively studied in the field [1] and is a fundamental aspect of a variety of applications including underwater exploration, scientific sampling, or underwater photogrammetry. Navigation is typically defined as an optimization problem where the goal is to find the optimal path that minimizes some cost function — obstacle avoidance [15], energy cost [16], etc. Some classical planning methods have been adapted to underwater domains as well. [5] presents an augmentation of a classical optimization package for manipulators, to perform 3D trajectory planning of an underwater vehicle that uses either a known map or an online constructed local map. AquaVis [11], an active perception system, was proposed to enhance this work by incorporating visual inspection. It identifies visual goals in real-time, moves towards the target location, and enhances the view of nearby visual goals from the preferred distance. These methods have been shown to have promising results, but they assume the existence of precise localization.

Pérez-Alcocer et al. [17] developed a robust vision-based navigation system in an underwater environment with limited visibility. Manzanilla et al. [18] employed a vision-based localization approach using an Extended Kalman Filter (EKF), and designed a proportional integral derivative (PID) controller for the autonomous navigation of an underwater vehicle. Bobkov et al. [19] proposed an adaptive method using stereo camera data to improve navigation performance.

These methods attempted to incorporate planning with state estimation but they do not consider efficient data collection as an objective.

Very few attempts have been made in the literature to address underwater coverage and data acquisition problems when operating in 3D space. Extending on classical coverage methods, [20] suggests segmenting the environment based on similar depths and in later work [10] takes into account state estimation uncertainties. [21] proposes a next-best view approach but it is significantly constrained by the certainty of the state estimation. To overcome the complexity of 3D exploration, a simplified formulation of the problem is considered, such as 2D mapping of underwater structure [9]. These methods are attempts towards 3D exploration but they are also limited by the prior information about the environments given in a map form or by the state estimation accuracy.

To overcome limitations of reliable state estimation while collecting underwater data [13] and [14] have proposed an end-to-end vision-based approach that surveys highly unstructured environments such as coral reefs and shipwrecks. In these works, the navigation policy is learned through supervised imitation learning. The human diver is asked to label image sequences with yaw and pitch changes such that the resulting action will drive the robot towards coral/shipwreck and simultaneously avoid obstacles. [22] extended these works to incorporate user-defined goal conditions, while [23] adapted it to navigation in confined areas. Each of these methods is specifically trained for exploring a particular ROI - coral reef or shipwreck and will require new labeling and retraining for navigation over new ROI.

In summary, various methods have been proposed for path planning in underwater environments, with varying degrees of dependence on prior information and the complexity of the environment. The challenge remains to develop a general solution that can navigate effectively in unknown environments while avoiding obstacles and performing efficient data collection. Our work is motivated by human diver-inspired coverage and navigation methods [14], [24], [25]. We introduce a novel domain-invariant navigation policy strategy that using IR reduces policy retraining and labeling overhead. Our approach provides a more generalized solution that is less sensitive to changes in the environment and can be used when the robot is unable to localize, whether in combination with global planning methods or standalone.

## III. APPROACH

Our method, UIVNAV, is designed to maximize data collection for OOI and avoid obstacles without relying on localization. Our proposed method is based on imitation learning and is influenced by intuition used by human divers who do not rely on localization when collecting data. The backbone of the UIVNAV is the intuition that when using only vision-based navigation to collect more data and avoid obstacles, two important cues are intrinsically used by humans: (1) recognition of the OOI — segmentation of the OOI, and (2) spatial understanding of proximity of the

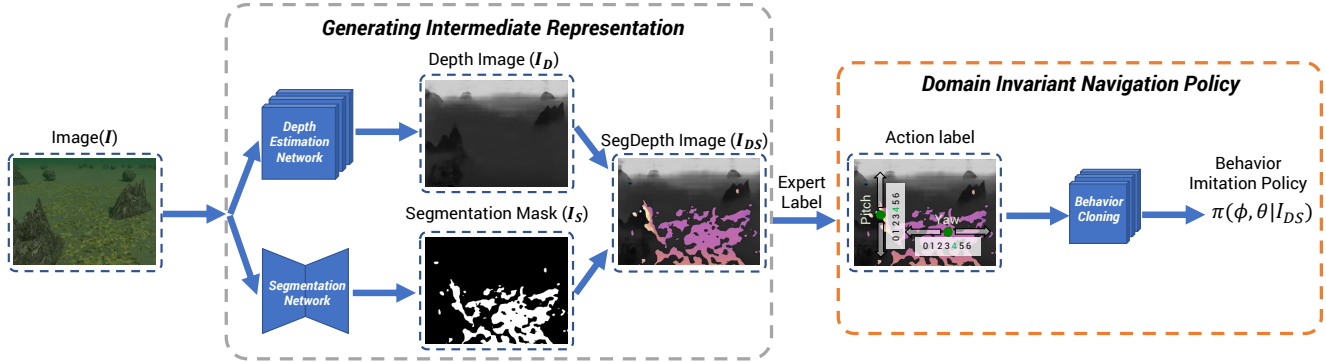


Fig. 2. An overview of our approach. UIVNAV consists of two phases: Generating Intermediate Representation (gray box) and Data Invariant Navigation Policy (orange box). Image ( $I$ ) is fed into the depth estimation network and the segmentation network. The output from the depth estimation network ( $I_D$ ) and the output from the segmentation network ( $I_S$ ) are used to generate the intermediate representation ( $I_{DS}$ ). This is fed into an imitation learning policy that predicts yaw and pitch changes. Due to training on ( $I_{DS}$ ) the navigation policy is domain invariant and will not require restraining nor labeling with new OOI.

obstacles or OOI — depth estimation. We use this intuition to design a framework that achieves this objective.

An overview of the proposed approach is illustrated in Fig. 2. Our two-stage system consists of *Generating Intermediate Representation* and *Domain Invariant Navigation Policy*. In the first stage, the robot’s images ( $I$ ) are fed into the depth estimation network and the segmentation network simultaneously. The outputs from the depth estimation network ( $I_D$ ) and the outputs from the segmentation network ( $I_S$ ) are used to generate the intermediate representation ( $I_{DS}$ ). Creating the intermediate presentation ( $I_{DS}$ ) aims to eliminate domain-specific information and allow the system to gather information for different OOI while avoiding obstacles. We assume that  $I_S$  and  $I_D$  are available or can be generated using existing methods [26]–[29].

In the second stage, the Domain Invariant Control Policy phase, we label the intermediate presentation frames ( $I_{DS}$  from stage one) with actions (pitch and yaw changes) for obstacle avoidance and information gathering. With the labeled  $I_{DS}$ , we train a behavior cloning network that controls the robot to maximize the information gain for the OOI while avoiding obstacles. We present each stage in the following subsections.

### A. Intermediate Representation

The networks that generate the intermediate representation must be altered with other state-of-the-art networks for various applications. For different domain work (air, ground and oceanic), and tasks (coral monitoring, aerial forester monitoring), the dataset used to train those networks shall also be different. We will briefly go through the networks and datasets we used and then define our intermediate representation in this subsection.

1) *Depth Estimation Network*: We need to acquire a corresponding dataset to train a depth estimation network in the underwater oyster reef environments. We utilized a convenient oyster simulation environment, OysterSim [30], to create the dataset consisting of 2000 image pairs of RGB images and the corresponding depth images(ground truth from the simulation). We adopted the monocular depth estimation network from Keras [31], with Unet [28] backbone.

We used Inverse Huber Loss [32] for the depth estimation network. Sample output from the depth estimation network ( $I_D$ ) can be seen in Fig. 2.

2) *Segmentation Network*: Similarly, we utilized OysterSim [30] to create the dataset of 2000 image pairs of RGB images of oyster reefs and the corresponding segmentation masks(ground truth from the simulation). We used the state-of-the-art oyster detection network, Oysternet [27], for segmenting the OOI. Sample output from the segmentation network ( $I_D$ ) can also be seen in Fig. 2.

We want to generate SegDepth Image ( $I_{DS}$ ) for all images ( $I$ ) in the Dataset ( $\mathcal{D}$ ). Depth image ( $I_D$ ) is a grayscale image with only 1 channel. We first want to save only the depth information for the OOI ( $I_{D_{OOI}}$ ) and only the depth information for non-OOI ( $I'_{D_{OOI}}$ ). We then used the color mapping function ( $M$ ) from matplotlib [33] that maps the 1-channel  $I_{D_{OOI}}$  to 3-channel (RGB)  $I_{DS_{OOI}}$ . The depth information is still preserved in the 3-channel image and its color (3-channel information) represents the region of interest. We stack (repeat) the  $I'_{D_{OOI}}$  three times to get the 3-channel depth information ( $I'_{DS_{OOI}}$ ) for non-OOI. Finally, we combine these two 3-channel information to get our SegDepth Image ( $I_{DS}$ ).

---

### Algorithm 1 Generating SegDepth Image

---

```

1: for  $I \in \mathcal{D}$  do
2:    $I_{D_{OOI}} = I_D \ \& \ I_S$ 
3:    $I'_{D_{OOI}} = I_D \ \& \ \neg I_S$ 
4:    $I_{DS_{OOI}} = M(I_{D_{OOI}})$  ▷ Mapping function
5:    $I'_{DS_{OOI}} = \text{stack}(I'_{D_{OOI}})$  ▷ stack 1-channel 3 times
6:    $I_{DS} = I_{DS_{OOI}} + (I'_{DS_{OOI}})$ 
7: end for
8: return  $I_{DS}$ 

```

---

### B. Domain Invariant Navigation Policy

To learn our Domain invariant navigation policy, THE UIVNAV approach follows a similar pipeline as the behavioral cloning method presented in the literature [13], [14], [23], [25]. The major distinctions are the navigation policy

input of the network and the labeling objectives. We use the Intermediate Representation ( $I_{DS}$ ) instead of the RGB image as our input data for imitation learning. We assume the robot is positioned at least 5m above the OOI, has a front-facing camera and the environment contains obstacles that can exceed 5m height. This is a typical case for top-view data collection common in oyster, coral, or forestry monitoring applications. We generate ( $I_{DS}$ ) on the raw images collected from such an environment to perform the navigation policy labeling. A human diver performs the labeling through a Graphical User Interface (GUI) that shows the current frame and possible classes for yaw and pitch changes (see Action Label in Fig. 2).

The labeling objectives of the human diver that collects data over OOI are as follows: (1) prefer to move towards OOI; (2) move towards larger areas of the OOI that are further away; (3) avoid obstacles. Yaw and pitch values are each encoded as a discrete set of seven class labels similar to the related works suggested in the literature [13], [14]. A Convolutional Neural Network (CNN) with a ResNet-50 backbone is trained on 15000 labeled input images ( $I_{DS}$ ) to predict these relative heading changes in yaw and pitch.

We denote the  $f_{\theta}(I_{DS})$  and  $f_{\phi}(I_{DS})$  for the categorical distribution prediction (vector of size 1 by 7 for the probability of each class) for yaw and pitch respectively with the input of  $I_{DS}$  from the behavior imitation policy. There are seven class prediction labels with a mean of 3 for the yaw and pitch which are described as  $C_{\theta}$  and  $C_{\phi} \in \{0, 1, 2, 3, 4, 5, 6\}$ .  $C_{\theta}$  corresponds to desired yaw changes of  $\{3\delta_{\theta}, 2\delta_{\theta}, \delta_{\theta}, 0, -\delta_{\theta}, -2\delta_{\theta}, -3\delta_{\theta}\}$  with respect to the current image frame. Here the positive change represents a clockwise heading change and the negative change represents a counterclockwise heading change with the amount of  $\{\delta_{\theta}, 2\delta_{\theta}, 3\delta_{\theta}\}$ . Similarly  $C_{\phi}$  to the desired pitch changes with respect to the current image frame. Here the positive change represents an upward-heading change.

When applying the behavior imitation policy, we choose  $\delta_{\theta} = 5^{\circ}, \delta_{\phi} = 5^{\circ}$  as sharp changes are undesirable. As suggested by [25], we also performed label smoothing on the ground truth label. We treat  $f_{\theta}(I_{DS})$  and  $f_{\phi}(I_{DS})$  separately as we are training the policy while using the same loss function below. As we believe the wrong prediction for one axis shall not penalize the right prediction for the other axis. The following loss is used to train the policy  $\pi(\phi, \theta|I_{DS})$ :

$$\mathcal{L}_{loss}(\mathcal{D}, w) = \lambda * \mathcal{L}_{CCE}(\mathcal{D}, w) + (1 - \lambda) * \mathcal{L}_{KL}(\mathcal{D}, w) \quad (1)$$

The  $\lambda$  here is a weight factor for the loss function. We choose  $\lambda = 0.1$ .  $\mathcal{L}_{CCE}$  is categorical cross-entropy loss and  $\mathcal{L}_{KL}(\mathcal{D}, w)$  is KL divergence from [34].

The resulting predicted 7-class discrete network values for yaw and pitch are transformed into low-level actuator commands that the robot's motion controllers carry out during inference.

#### IV. EXPERIMENTAL EVALUATION

We use the oyster reef monitoring application as a case study for demonstrating the capabilities of UIVNAV. We

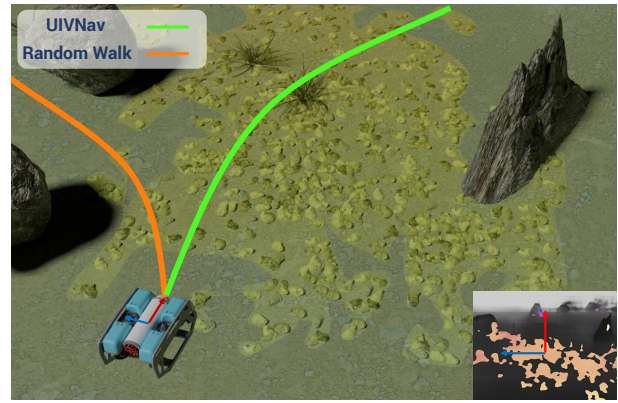


Fig. 3. Example of BlueROV surveying reef of oysters in simulation (translucent mask is applied on the oyster reef for better visualization.)

performed extensive quantitative and qualitative comparisons of our method with two baseline methods — (1) Brownian Bridge (BB) [35] random walk and (2) an offline complete coverage method based on Boustrophedon Cell Decomposition (BCD) [7]. Brownian Bridge-based motions have been used to study animal trails [36] and can have the potential to replicate the distribution of natural phenomena such as oyster or coral reef growth patterns. It has also been widely used in exploration tasks [37], [38]. As our goal is to maximize the collected data of OOI, we compare our method to the complete coverage path planning method, which is the state-of-the-art approach for completely covering the area when the prior map is given. In practice, farmers or scientists will perform a lawnmower pattern — naive version of BCD pattern to navigate their vehicles and collect data or dredge oysters. For the BCD algorithm, we only assume the boundaries of the region and obstacle locations are given but not the distribution of the OOI. Note that we do not use any prior information about the environment when running UIVNAV.

In this section, we will first conduct an evaluation in a simulated environment with oysters. Additionally, to demonstrate the generalizability of our method to navigate over different OOI, we test in a rock reef environment. Finally, we show results from a deployment of UIVNAV on a BlueROV underwater robot in a pool to assess the real-world viability of our approach in real-time.

##### A. Simulation

To generate underwater oyster and rock reef environments we used an open-source oyster simulation framework based on Blender [39] game engine, OysterSim [30]. We use a model of the BlueROV robot in the simulation as shown in Fig 3. We have generated different possible distributions of oyster reefs and one sample of rock reef. When designing the scenarios we took into account that underwater natural reefs – oyster or coral — usually do not grow separately but they tend to be in large groups. Nevertheless, they can have different discontinuous patches. As such, our test scenarios consist of the following instances:

- 1) Grid World - a reef with connected continuous patches of oysters with multiple branches (Fig. 4a).

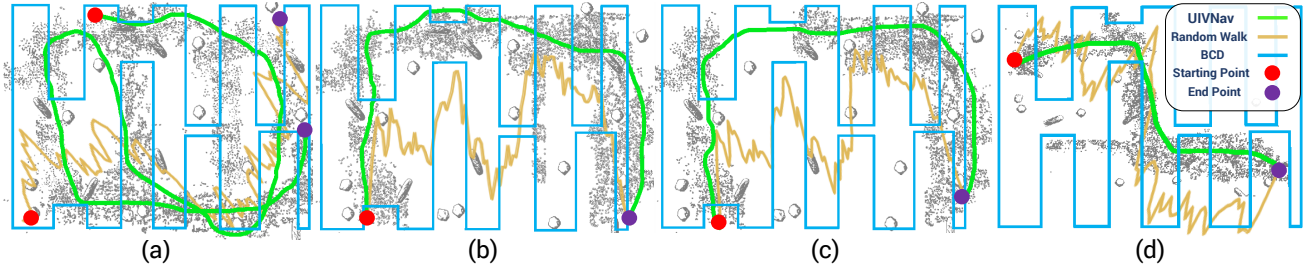


Fig. 4. Comparing various exploration strategies across four oyster reef patterns– (a) Grid world, (b) E-shape, (c) Disconnected Paths, (d) Branching Corridor.

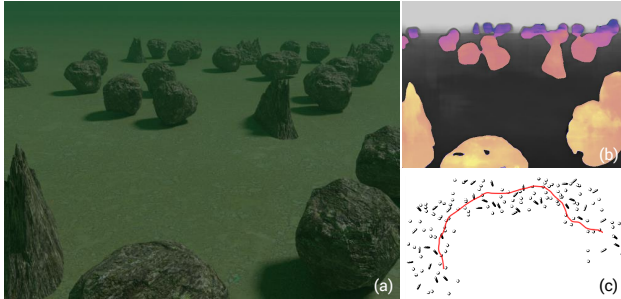


Fig. 5. UIVNAV surveying reef of rocks OOI with no additional human-expert labeling or training. (a) The current frame of the robot field of view in OysterSim simulation. (b) Intermediate representation of the current frame that the robot uses to make navigation decisions. (c) The complete trajectory over a rock reef.

- 2) E-shape - C-shaped wide oyster reef patch with a narrow oyster patch in the middle (Fig. 4b).
- 3) Disconnected Paths - a reef consisting of two large oyster groups that are separated by a large sandy area (Fig. 4c).
- 4) Branching Corridor - a reef with wide and narrow oyster patches branching from the opposite sides of the reef (Fig. 4d).
- 5) Rock Reef - a reef of large rocks with continuous C-shape (Fig. 5c).

## B. Results

Since our objective is to maximize the data collected we have measured *Distance Traveled* with *Distance Traveled over Oysters* in meters and calculated the *Percentage of Oysters* contained within each generated path assuming the robot travels at a consistent speed. As depicted in Fig. 6, our results prove the hypothesis that our method UIVNAV is better than the random walk. Compared to the complete coverage method, the results indicate that a robot using UIVNAV collects on average 29% less oyster data while traveling on average 70% less distance. This shows that UIVNAV trajectories survey on average 36% more oysters than the complete coverage method when traveling the same distance.

In each of the four test environments, we show that UIVNAV makes qualitatively more efficient decisions (Fig. 4). In *Grid World*, UIVNAV almost completely covers the oyster reef without navigating over sand patches and avoids all of the obstacles (Fig. 4a). In *E-shape*, we can observe that UIVNAV chooses to turn on a wider region towards the endpoint rather than traveling through the middle patch — demonstrating the ability to infer “future reward” (Fig. 4b).

*Disconnected Paths* scenario, UIVNAV is able to navigate through the gap and locate a new OOI in sight, enabling it to effectively survey over the remaining portion of the oyster reef (Fig. 4c). The last scenario with oyster reef — *Branching Corridor*, demonstrates UIVNAV’s ability to select wider regions with oyster surface to continue exploration (Fig. 4d).

We also qualitatively evaluate UIVNAV on the *Rock Reef* scenario to show that the navigation policy is domain invariant. To obtain the intermediate representation for different OOI, we re-train the segmentation network for detecting the rocks. On a newly generated IR, we directly apply a trained navigation policy. Robot successfully explores new OOI without the need for additional human-expert labeling or re-training of the policy (Fig. 5).

## C. Lab Deployment

1) *Hardware Setup*: We conducted pool experiments with the BlueROV2 robot in Heavy Configuration [40] with 8 thrusters for motion in all 6 degrees of freedom. The BlueROV2 is equipped with the following onboard sensors: TDK Invensense ICM-20602 Inertial Measurement Unit (IMU), Waterlinked DVL-A50 Doppler Velocity Logger (DVL), and a Bosch BMP280 pressure sensor, an onboard Low Light HD USB Camera with 1080p resolution. The robot works on the *Navigator* flight controller with ArduSub as the software suite. The BlueROV2 is controlled by giving Pulse-Width Modulation (PWM) to the thrusters through I2C interface PCA9685 — a 16-Channel PWM Module.

The BlueROV2 is connected through the tether to the Ground Central Station (GCS) - Dell Precision 7560. The GCS is equipped with the NVIDIA RTX A2000 GPU.

2) *Software Setup*: All drivers for processing the data and autonomous control of the robot are ROS-based software written in Python. The UIVNAV was deployed on GCS, executing navigation commands in real-time at a frequency of 0.5Hz. The commands generated from the UIVNAV network are mapped to the PWM inputs for thrusters to change the BlueROV2’s pitch, yaw, surge, and heave. The network gives us a set of commands ( $c$ ) as single digit numbers, which determine the desired velocity ( $\nu$ ) — ( $\nu = c * \alpha$ ), where  $\alpha$  is a constant determining factor by which the magnitude of speed changes. The input value for the PWM input is calculated as follows:

$$\text{PWM}_{\eta} = 1500\mu\text{s} + k_{\eta}\nu \quad (2)$$

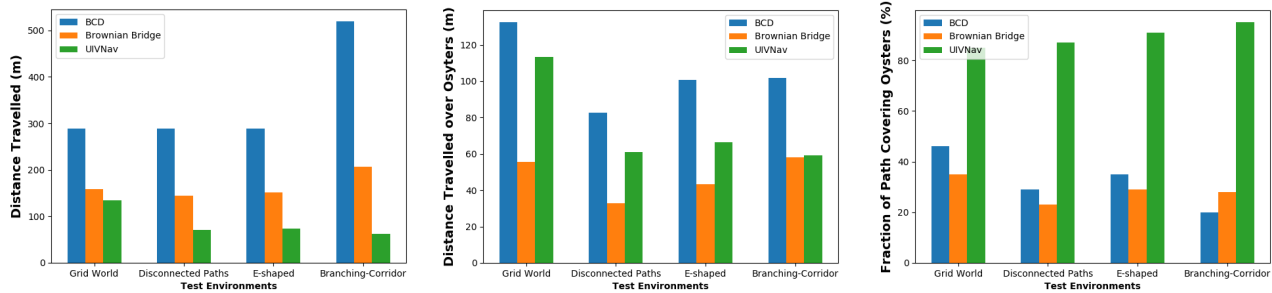


Fig. 6. Quantitative Evaluation of Different Strategies on Different Environments. BB method travels longer distances moving on fewer areas with oysters compared to UIVNAV. Even though compared to BCD, the UIVNAV method collects on average 29% less oyster data it travels 70% less distance. This shows that UIVNAV trajectories survey on average 36% more oysters than the BCD method when traveling the same distances.

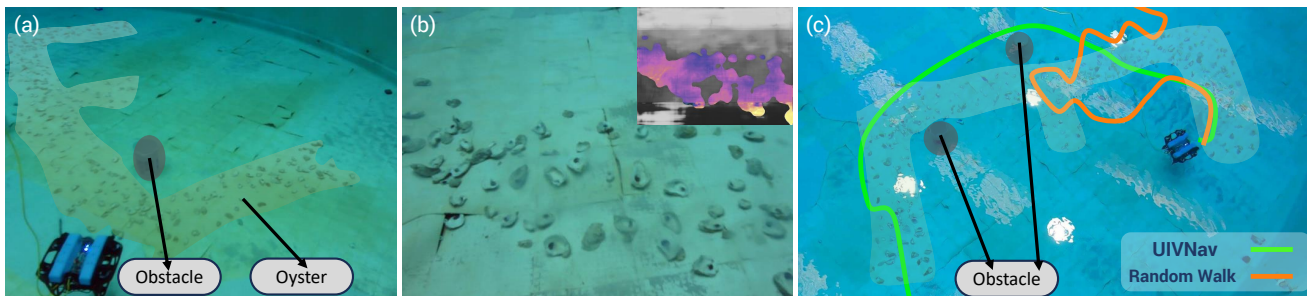


Fig. 7. (a) BlueROV2 Deployed in a Neutral Buoyancy Pool with Yaw and Pitch control with E-shape oyster shells distributed at the bottom. (b) BlueROV's field of view. (c) Experimental result of two trajectories of BlueROV surveying oyster shells in a pool.

where  $1500\mu s$  keeps the thrusters at neutral position,  $\eta$  denotes the concerned degree of freedom and  $k$  denotes the gain for each direction which is calculated experimentally.

3) *Experiment setup*: We distributed up to 300 oyster shells at the bottom of the neutral buoyancy tank. The neutral buoyancy tank is 50 feet across and 25 feet deep. The E-shape pattern of the oyster shell patch is bounded by (Fig. 7a) a 40x20 feet area — which occupies half of the tank. The breadth of the oyster patch averages 1 to 2 feet, which fits nearly 3-5 shells.

The real-world deployment of the robot posed several challenges that significantly increased the task's difficulty. Namely, the robot's motion was influenced by the altitude at which front camera images were captured, drift from turning and surging, and external forces that affected the robot's turning radius and positioning. Nevertheless, the robot successfully surveyed the oyster shells using the navigation policy derived from our method (see Fig. 7c).

In this experiment, without prior knowledge of the environment and localization, the UIVNAV moves the robot towards oyster reefs (Fig. 7b)— maximizing the information gain over OOI while avoiding two obstacles (two water buckets). By using random walk, the robot also traverses through a quarter of the OOI; however, the trajectory went outside of OOI.

## V. CONCLUSION AND FUTURE WORK

We presented UIVNAV, an underwater navigation method that maximizes collected data of OOI while avoiding obstacles. UIVNAV comprises the following steps: (1) creating an intermediate representation (IR) and (2) training the

navigation policy using human-labeled IR. The navigation policy remains effective without the need for retraining when there are changes in the domain or OOI. We have demonstrated this by navigating over both oyster and rock reefs. Furthermore, our method has been deployed in real-time on BlueROV2 robot in the pool to navigate over the patch of oyster shells. The proposed UIVNAV system provides a promising step towards developing generalized and efficient underwater navigation solutions for exploration and monitoring applications.

**Limitations**: Our goal within this work was to explore limitations of navigation without any localization; therefore, by design UIVNAV does not rely on state estimation. As a result, there is no mechanism at the moment to detect the previously visited regions. This can potentially cause robots to visit the same area multiple times without covering the entire region. Nevertheless, our method can be utilized as a local planner for better information gathering within any higher-level global plans when global localization is available.

**Future Work**: To demonstrate scalability and applicability in the real world, we plan to conduct extensive open-water surveying operations on real oyster and coral reefs. Additionally, we intend to incorporate the information from the SONAR sensor, specifically range information, for better depth estimation and obstacle detection. The presented work can be also combined with tracking strategies to monitor dynamic interest regions such as school of swimming fish. Finally, we aim to evaluate the versatility of our algorithm by applying it to diverse domains, including both aerial and terrestrial settings.

## REFERENCES

- [1] L. Christensen, J. de Gea Fernández, M. Hildebrandt, C. E. S. Koch, and B. Webb, "Recent advances in ai for navigation and control of underwater robots," *Current Robotics Reports*, pp. 1–11, 2022.
- [2] J. Hansen, S. Manjanna, A. Q. Li, I. Rekleitis, and G. Dudek, "Autonomous marine sampling enhanced by strategically deployed drifters in marine flow fields," in *OCEANS 2018 MTS/IEEE Charleston*. IEEE, 2018, pp. 1–7.
- [3] N. Karapetyan, J. Moulton, and I. Rekleitis, "Meander-based river coverage by an autonomous surface vehicle," in *Field and Service Robotics*. Springer, 2021, pp. 353–364.
- [4] S. Manjanna, N. Kakodkar, M. Meghjani, and G. Dudek, "Efficient terrain driven coral coverage using gaussian processes for mosaic synthesis," in *2016 13th Conference on Computer and Robot Vision (CRV)*. IEEE, 2016, pp. 448–455.
- [5] M. Xanthidis, N. Karapetyan, H. Damron, S. Rahman, J. Johnson, A. O'Connell, J. M. O'Kane, and I. Rekleitis, "Navigation in the presence of obstacles for an agile autonomous underwater vehicle," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 892–899.
- [6] F. Jalal and F. Nasir, "Underwater navigation, localization and path planning for autonomous vehicles: A review," in *2021 International Bhurban Conference on Applied Sciences and Technologies (IBCAST)*. IEEE, 2021, pp. 817–828.
- [7] A. Xu, C. Viriyasuthee, and I. Rekleitis, "Efficient complete coverage of a known arbitrary environment with applications to aerial operations," *Autonomous Robots*, vol. 36, pp. 365–381, 2014.
- [8] N. Karapetyan, J. Moulton, J. S. Lewis, A. Quattrini Li, J. M. O'Kane, and I. M. Rekleitis, "Multi-robot dubins coverage with autonomous surface vehicles," in *icra*, 2018.
- [9] E. Vidal, N. Palomeras, K. Istenič, J. D. Hernández, and M. Carreras, "Two-dimensional frontier-based viewpoint generation for exploring and mapping underwater environments," *Sensors*, vol. 19, no. 6, p. 1460, 2019.
- [10] E. Galceran, R. Campos, N. Palomeras, D. Ribas, M. Carreras, and P. Ridao, "Coverage path planning with real-time replanning and surface reconstruction for inspection of three-dimensional underwater structures using autonomous underwater vehicles," *Journal of Field Robotics*, vol. 32, no. 7, pp. 952–983, 2015.
- [11] M. Xanthidis, M. Kalaitzakis, N. Karapetyan, J. Johnson, N. Vitzilaios, J. M. O'Kane, and I. Rekleitis, "Aquavis: A perception-aware autonomous navigation framework for underwater vehicles," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2021, pp. 5410–5417.
- [12] B. Joshi, S. Rahman, M. Kalaitzakis, B. Cain, J. Johnson, M. Xanthidis, N. Karapetyan, A. Hernandez, A. Q. Li, N. Vitzilaios, et al., "Experimental comparison of open source visual-inertial-based state estimation algorithms in the underwater domain," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2019, pp. 7227–7233.
- [13] T. Manderson, J. C. G. Higuera, R. Cheng, and G. Dudek, "Vision-based autonomous underwater swimming in dense coral for combined collision avoidance and target selection," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 1885–1891.
- [14] N. Karapetyan, J. V. Johnson, and I. Rekleitis, "Human diver-inspired visual navigation: Towards coverage path planning of shipwrecks," *Marine Technology Society Journal*, vol. 55, no. 4, pp. 24–32, 2021.
- [15] B. Braginsky and H. Guterman, "Obstacle avoidance approaches for autonomous underwater vehicle: Simulation and experimental results," *IEEE Journal of oceanic engineering*, vol. 41, no. 4, pp. 882–892, 2016.
- [16] N. Yang, D. Chang, M. Johnson-Roberson, and J. Sun, "Energy-optimal path planning with active flow perception for autonomous underwater vehicles," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 9928–9934.
- [17] R. Pérez-Alcocer, L. A. Torres-Méndez, E. Olgún-Díaz, and A. A. Maldonado-Ramírez, "Vision-based autonomous underwater vehicle navigation in poor visibility conditions using a model-free robust control," *Journal of Sensors*, vol. 2016, 2016.
- [18] A. Manzanilla, S. Reyes, M. Garcia, D. Mercado, and R. Lozano, "Autonomous navigation for unmanned underwater vehicles: Real-time experiments using computer vision," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 1351–1356, 2019.
- [19] V. A. Bobkov, V. Y. Mashentsev, A. Y. Tolstonogov, and A. P. Scherbatyuk, "Adaptive method for auv navigation using stereo vision," in *The 26th International Ocean and Polar Engineering Conference*. OnePetro, 2016.
- [20] E. Galceran and M. Carreras, "Efficient seabed coverage path planning for asvs and auvs," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2012, pp. 88–93.
- [21] E. Vidal, J. D. Hernández, K. Istenič, and M. Carreras, "Online view planning for inspecting unexplored underwater structures," *IEEE Robotics and Automation Letters*, vol. 2, no. 3, pp. 1436–1443, 2017.
- [22] T. Manderson, J. C. Gamboa Higuera, S. Wapnick, J.-F. Tremblay, F. Shkurti, D. Meger, and G. Dudek, "Vision-based goal-conditioned policies for underwater navigation in the presence of obstacles," *Robotics: Science and Systems XVI*, Jul 2020.
- [23] T. Liu, M. Yu, and N. Chopra, "Learning-based autonomous underwater vehicle navigation following human actions in confined environment," in *OCEANS 2022, Hampton Roads*. IEEE, 2022, pp. 1–8.
- [24] T. Manderson, J. C. G. Higuera, S. Wapnick, J.-F. Tremblay, F. Shkurti, D. Meger, and G. Dudek, "Vision-based goal-conditioned policies for underwater navigation in the presence of obstacles," *arXiv preprint arXiv:2006.16235*, 2020.
- [25] N. Smolyanskiy, A. Kamenev, J. Smith, and S. Birchfield, "Toward low-flying autonomous mav trail navigation using deep neural networks for environmental awareness," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2017, pp. 4241–4247.
- [26] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo, et al., "Segment anything," *arXiv preprint arXiv:2304.02643*, 2023.
- [27] X. Lin, N. J. Sanket, N. Karapetyan, and Y. Aloimonos, "Oysternet: Enhanced oyster detection using simulation," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 5170–5176.
- [28] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Medical Image Computing and Computer-Assisted Intervention—MICCAI 2015: 18th International Conference, Munich, Germany, October 5–9, 2015, Proceedings, Part III 18*. Springer, 2015, pp. 234–241.
- [29] B. Yu, J. Wu, and M. J. Islam, "Udepth: Fast monocular depth estimation for visually-guided underwater robots," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 3116–3123.
- [30] X. Lin, N. Jha, M. Joshi, N. Karapetyan, Y. Aloimonos, and M. Yu, "Oystersim: Underwater simulation for enhancing oyster reef monitoring," in *OCEANS 2022, Hampton Roads*. IEEE, 2022, pp. 1–6.
- [31] F. Chollet et al. (2015) Keras. [Online]. Available: <https://github.com/fchollet/keras>
- [32] I. Laina, C. Rupprecht, V. Belagiannis, F. Tombari, and N. Navab, "Deeper depth prediction with fully convolutional residual networks," in *2016 Fourth international conference on 3D vision (3DV)*. IEEE, 2016, pp. 239–248.
- [33] J. D. Hunter, "Matplotlib: A 2d graphics environment," *Computing in Science & Engineering*, vol. 9, no. 3, pp. 90–95, 2007.
- [34] J. M. Joyce, "Kullback-leibler divergence," in *International encyclopedia of statistical science*. Springer, 2011, pp. 720–722.
- [35] W. C. Chow, "Brownian bridge," *Wiley interdisciplinary reviews: computational statistics*, vol. 1, no. 3, pp. 325–332, 2009.
- [36] J. S. Horne, E. O. Garton, S. M. Krone, and J. S. Lewis, "Analyzing animal movements using brownian bridges," *Ecology*, vol. 88, no. 9, pp. 2354–2363, 2007.
- [37] B. Pang, Y. Song, C. Zhang, H. Wang, and R. Yang, "A swarm robotic exploration strategy based on an improved random walk method," *Journal of Robotics*, vol. 2019, pp. 1–9, 2019.
- [38] I. A. Wagner, M. Lindenbaum, and A. M. Bruckstein, "Robotic exploration, brownian motion and electrical resistance," in *Randomization and Approximation Techniques in Computer Science: Second International Workshop, RANDOM'98 Barcelona, Spain, October 8–10, 1998 Proceedings 2*. Springer, 1998, pp. 116–130.
- [39] B. O. Community, *Blender - a 3D modelling and rendering package*, Blender Foundation, Stichting Blender Foundation, Amsterdam, 2018. [Online]. Available: <http://www.blender.org>
- [40] B. Robotics, "Bluerov2: The world's most affordable high-performance rov," *BlueROV2 Datasheet; Blue Robotics: Torrance, CA, USA*, 2016.