# Underwater Exploration and Mapping

Bharat Joshi[1], Marios Xanthidis[2], Monika Roznere[3], Nathaniel J. Burgdorfer[4], Philippos Mordohai[4], Alberto Quattrini Li[3], and Ioannis Rekleitis[1]

***Abstract*—This paper analyzes the open challenges of exploring and mapping in the underwater realm with the goal of identifying research opportunities that will enable an Autonomous Underwater Vehicle (AUV) to robustly explore different environments. A taxonomy of environments based on their 3D structure is presented together with an analysis on how that influences the camera placement. The difference between exploration and coverage is presented and how they dictate different motion strategies. Loop closure, while critical for the accuracy of the resulting map, proves to be particularly challenging due to the limited field of view and the sensitivity to viewing direction. Experimental results of enforcing loop closures in underwater caves demonstrate a novel navigation strategy. Dense 3D mapping, both online and offline, as well as other sensor configurations are discussed following the presented taxonomy. Experimental results from field trials illustrate the above analysis.**

## I. Introduction

Exploring the underwater realm is a challenging but fascinating task. Mapping the Titanic shipwreck [1], exploring the deepest Cenotes [2] and largest underwater caves [3], monitoring the world's coral reefs [4] all present unique challenges but push the limits of human knowledge. Marine archaeology, environmental monitoring, infrastructure maintenance, search and rescue are some of the domains where underwater mapping plays a crucial role. Automating mapping with Autonomous Underwater Vehicles (AUVs) will remove humans from hazardous situations, improve the repeatability, and enable longer operations times.

Operations underwater present many challenges. Localization is difficult due to the lack of GPS and multi-path effects for acoustic positioning. Vision based sensing is affected by light and color attenuation [5], [6], blurriness, floating particulates, varying illumination, and lack of features [7]. Furthermore, caustic patterns in shallow waters, and total lack of ambient light inside caves and wrecks produce moving features independent of the camera's motion. AUVs moving cannot instantaneously halt their motion due to inertia, while water movement affects the vehicles in unpredictable ways.

[1] Computer Science and Engineering Department, University of South Carolina, USA bjoshi@email.sc.edu, yiannisr@cse.sc.edu

[2] SINTEF Ocean, Norway, marios.xanthidis@sintef.no

[3] Dartmouth College, USA, {monika.roznere.gr, alberto.quattrini.li}@dartmouth.edu

[4] Stevens Institute of Technology, Hoboken, NJ, USA, 07030, {nburgdor,pmordoha}@stevens.edu

This work was possible through the generous support of the National Science Foundation (NSF 1943205, 1919647, 2024741, 2024541, 2024653, 2144624) and the Research Council of Norway (ResiFarm, NO-327292). The authors would also like to acknowledge the help of the Woodville Karst Plain Project (WKPP) and El Centro Investigador del Sistema Acuífero de Quintana Roo A.C. (CINDAQ) in collecting data, providing access to underwater caves, and mentoring us in underwater cave exploration. Last but not least, we would like to thank Halcyon Dive Systems for their support with equipment.

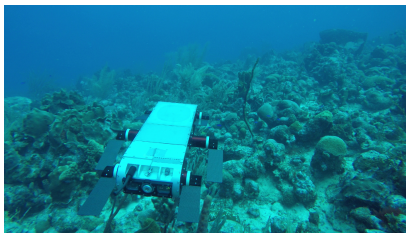


Fig. 1. Aqua2 AUV collecting data over the coral reef, Barbados.

Trajectory planning needs to take into account the uncertainty in motion in order to maintain adequate safety margins.

In addition, every target environment underwater requires different motion strategies depending on the desired outcome. The complexity of the three dimensional structure provides a guideline to analyze different mapping strategies. Coral reefs and archaeological sites present a mostly flat profile, where the AUV traverses over. Wrecks, and infrastructure present a 3D structure, where the AUV has to change depths and navigate around. Finally, overhead environments, such as caves and the interiors of wrecks, require mapping all around the AUV.

The main contribution of this paper is in identifying open challenges and potential approaches that can be pursued to achieve fully autonomous exploration of underwater environments. After an overview of related work on AUV state estimation and navigation, we present a classification of underwater environments that affect the choices on methods for state estimation and navigation. Based on the classification, we then discuss different exploration strategies, environment representations, and sensor configurations. We conclude the paper highlighting future research directions.

## II. Related Work

A review of general approaches for AUV localization and navigation and the commonly used sensors is presented in [8], [9]. Sonar (e.g., multibeam sonar, scanning profiling sonar, and imaging sonar [10]) and/or cameras are used to bound the odometry drift from dead-reckoning systems, i.e., IMU or Doppler Velocity Log (DVL). Acoustic sensors such as Ultra-Short Baseline (USBL) and DVL are most commonly used for navigation. However, acoustic sensors are costly and they only return range to obstacles, missing important semantic information present only in visual data.

State estimation underwater is extremely challenging – see [12], [13] where several popular Visual Odometry (VO) and Visual/Inertial Odometry (VIO) packages are compared. Rahman *et al.* [14] fused visual, inertial, acoustic, and water depth data to accurately estimate the pose of the sensing system in a variety of environments. A multi-state constrained Kalman filter (MSCKF) [15] implementation for an AUV system with a

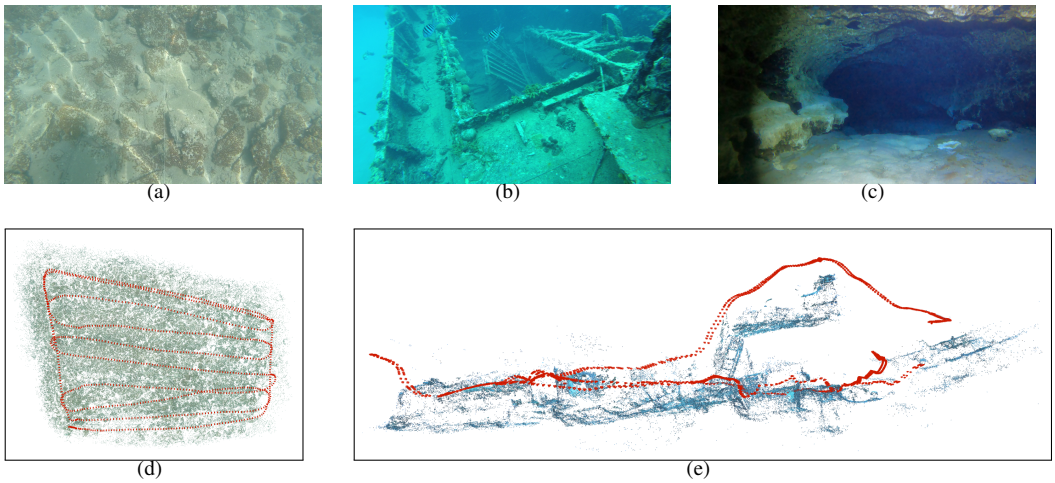



Fig. 2. Three different kinds of environments: (a) Mostly flat; a sunken Bronze era village, Greece. (b) Three dimensional structure; Pamir shipwreck, Barbados. (c) Overhead environment; Devil cave system, FL, USA. (d) Reconstruction utilizing global bundle adjustment [11] of the stereo data collected over a patch of the submerged village (a); (e) Partial reconstruction of the Pamir shipwreck (b). In both (d,e) the camera poses are depicted in red.

downward looking camera was presented in [16]. Accurate reconstructions of an underwater dataset via a bundle adjustment approach was presented in [17]. Recently, Eisele *et al.* [18] demonstrated the use of a plenoptic camera based navigation system on AUV; using Dual Extended Kalman Filter (DEKF) to fuse camera and IMU measurements. Multiple submaps are combined with a global stochastic map after each loop closure for large scale underwater SLAM in [19]. Kim *et al.* [20] presented a real-time monocular visual SLAM algorithm for underwater ship hull inspection using image saliency for the keyframe selection, gain-based link hypothesis, and novelty detection. Vargas *et al.* [21] proposed robust visual SLAM in underwater leveraging acoustic, inertial, depth sensors and acoustic odometry estimates as motion priors.

Similarly, underwater navigation is very difficult, due to state estimation challenges discussed previously, impact of water currents or inaccurate hydrodynamics assumed by the controls, and the high-dimensionality of the motion planning problem. At the same time, the limited range sensor capabilities in the underwater terrain and the presence of dynamic obstacles that are observed only in close proximity require the AUVs to quickly decide to guarantee safe operation. Past research has focused mostly on search-based and sampling-based approaches [22], [23], due to their popularity, analysis, intuitive understanding, and guarantees on finding a solution. Unfortunately, they are either bounded to 2D, do not deal effectively with motion uncertainty, or assume very simple environments due to being very computationally expensive. Deep-learning navigation techniques [24] are logistically very demanding to develop, and are limited only to similar environments. Additional underwater path planning techniques are presented in [25]. Previous work [26] addressed such shortcomings by providing a computationally-light optimization-based framework that deals effectively with motion uncertainty, enabling 3D navigation through challenging underwater environments, without imposing any constraints on the motion range of the robot. Extensions have been presented [27] towards developing navigation behaviors that maximize visibility of target areas, in support of state estimation.

## III. Environment Classification and Camera Configuration

Underwater environments can be classified in three broad categories, each requiring different motion strategies and changes in camera configuration to maximize the perception of the surroundings. The most commonly explored environment is characterized by mostly flat terrain. Such environments include coral reefs, archaeological sites, sea-grass beds, etc. Figure 1 and Figure 5 display the Aqua2 AUV [28] swimming over a coral reef, while Figure 2(a) shows a view of a Bronze era submerged village at "Bay of Koilada", at Lampayannas, Greece. The standard motion strategy is a lawnmower pattern over the area, at a height that results in acceptable resolution while maximizing the area covered by the camera footprint; see Figure 2(d) for the sparse reconstruction of Figure 2(a), with the camera poses marked in red.

Underwater 3D structures such as shipwrecks, oil-rigs, or underwater pinnacles and rocky formations, present a number of challenges. The target structure sits on the seafloor and extends upwards. Thus, the vehicle has to move around and over the structure, utilizing different motion strategies – see [29] for a deep learning approach for moving around a shipwreck. Figure 2(b) presents part of the deck and the insides of the Pamir shipwreck, Barbados, while Figure 5(b) shows an Aqua2 AUV navigating around the superstructure of the same wreck. Visible in both figures is a collapsed crane extending downwards. While such structures often present partial overhead structures, there is always a path that takes the AUV above the overhead – see Figure 2(e) for a partial sparse reconstruction of Pamir, where the camera, depicted in red, moved under and over the top deck.

The most complex underwater environments are characterized by complete overheads such as the inside of shipwrecks and underwater caves. The environment presents the challenges of the previous two types, a floor that needs to

Fig. 3. The structure inside an underwater cave, Mexico, and sparse reconstruction.

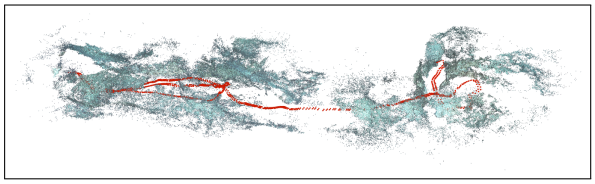


Fig. 4. Sparse reconstruction from inside a cave in Mexico.

be mapped, together with 3D structures, and in addition there is a ceiling and surrounding walls – see Figure 3, where there are interesting features all around. In contrast, Figure 2(c) presents an underwater cave in Florida which is much less decorated. In addition, due to the overhead there is no ambient light and the scene is illuminated only by the light-sources carried by the AUV. The moving light-sources result in constantly moving shadows that can be used for producing a denser scene reconstruction [30]. Navigating around such an environment is extremely challenging and complete coverage is near impossible to achieve. Figure 4 presents a partial sparse reconstruction of an underwater cave, Mexico; the camera trajectory, in red, moves among the structures.

Different camera configurations provide advantages given the target environment and the motion planning strategies employed. In the most popular case of a fly-over, the downward facing camera maximizes the visibility of the terrain – see Figure 6(a). When the camera sees the scene from an angle, as in Figure 6(b,c), the objects' opposite side is not visible; however, when only a downward facing camera is used, the AUV has no perception in the direction of motion. In such a case a forward facing camera tilted partially downwards ensures the AUV has situational awareness of where it is going, while at the same time avoiding looking out into open water – see Figure 6(b) for the current configuration of the Aqua2 AUV utilized by the authors. Please note that due to light absorption by water and limited visibility by particulates, the range of the cameras is limited, thus, the parts of the image that cover distant objects are of limited use. Finally, when the target environment is defined by an overhead, the forward looking cameras should be vertically centered to observe the ceiling together with the floor – see Figure 6(c) for the standard camera configuration of the Aqua AUV. Please note, utilizing multiple cameras provides a generalized field of view at the cost of large data volumes, which affect the bandwidth of the embedded system's data bus and introduces synchronization challenges and processing bottlenecks.

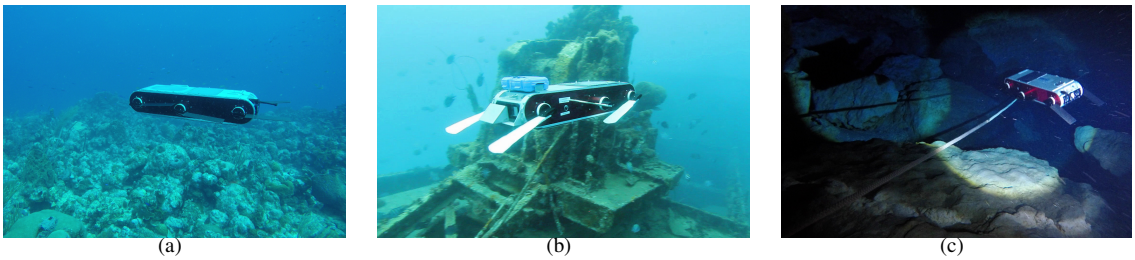
## IV. Exploration, Coverage, and Mapping

Mapping is the process of fusing measurements over time producing a consistent representation of the environment. Traditionally, mapping uses the data collected over an arbitrary trajectory without concern about completeness of representation. The areas of active perception and SLAM [31]–[33] are considering the areas which are left uncovered. However, there is no guarantee for completeness. The guiding principle behind reaching unknown areas is frontier based exploration [34], where the robot finds the areas bordering the unknown and collects new data. More generally, the AUV has to consider which parts of the environment are not sensed yet and then select where is the best place to take the next measurement [35].

Non-systematic exploration can be achieved by learned [24], [29] or curiosity-driven [36], [37] behaviors. In such cases, the robot looks for new areas to map without ensuring completeness of coverage or pursuing loop closures.

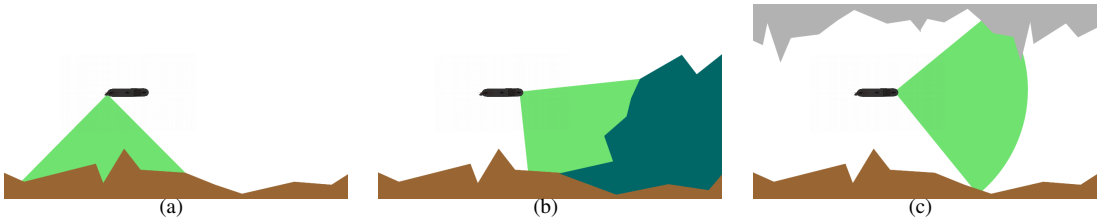
A related concept is coverage, defined as finding a path that would take the sensor's footprint over all free space. The main premise of most coverage algorithms is completeness, that is, there is no available area that is not scanned. While coverage always considers a bounded space (area or volume), exploration does not necessarily have that constraint. Another main difference between coverage and exploration is based on the sensor's range. Historical exploration approaches assume that the sensor's field of view is bounded by the obstacles of the environment, while coverage assumes a limited sensor footprint. Most of the coverage approaches underwater focus on flat environments [38], utilizing a lawnmower pattern, also known as boustrophedon, grid, or seed-spreader – please refer to [39], [40] for a review of traditional coverage algorithms. If there are depth changes, online replanning can modify the altitude of the AUV over the terrain [41]. Englot and Hover [42] proposed a 3D coverage algorithm for the inspection of underwater structures. Kim and Eustice [20] utilized the visual features for hull inspections.

Coverage inside an overhead environment is particularly challenging due to the typical structural complexity of such environments. Most human-visited environments, such as caves and wrecks, are marked with a guide-line traversing the main passages – in Figure 3 the yellow guideline can be seen in the left/middle part of the image. Traversing and mapping the environment following such guide-lines, still remains future work, as most current approaches are teleoperated [43]. Another subject that has yet to be investigated in depth is the relation between fundamental exploration strategies, the platforms considered, and the target domain. Often research focuses only on two of the above, which limits the options for the third factor: a choice of an exploration strategy and available platforms determines the environment of application; available platforms and the target domain determine the exploration strategy to be developed; and a choice of an exploration strategy and a target environment determines the robot to be considered. Rarely, though, all these factors are considered holistically at the planning stage due to logistics and finite



(a) (b) (c)

Fig. 5. AUV deployed in different environments: (a) performing a lawn-mowing pattern over a coral reef; (b) exploring over the Pamir wreck; (c) collecting data inside the Ballroom cavern; Ginnie Springs, FL.



(a) (b) (c)

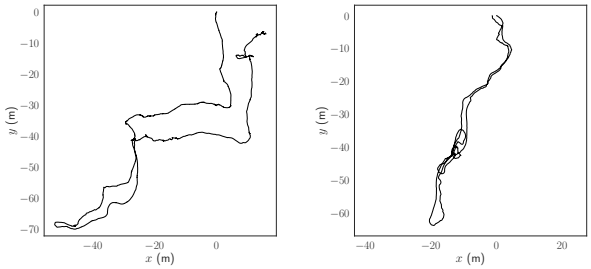
Fig. 6. The FOV resulting from different camera placement: (a) downward facing camera, utilized in "fly"-overs of flat terrains, (b) forward camera tilted downwards, utilized for mapping 3D structures, and (c) forward facing camera for mapping overhead environments.

material, human, and time resources.

For example, when exploring a cave area prone to silting, is it better to deploy an agile robot such as Aqua2 [28] that constantly moves and requires a minimal propulsion system but is unable to hover stably, or a more stable platform such as BlueROV2 that can maintain position but requires constant thruster operation, potentially disturbing the surroundings? Would an exploration strategy for producing a map of desired quality be possible with an agile robot or is it necessary to move slow and maintain accurate robot positioning? Similarly, in a shipwreck environment with currents, is it better to use a constantly moving agile robot that handles currents more effectively but at the potential expense of map quality, or use a hover-able robot to achieve more complete exploration at the increased collision risk and lowered battery life due to raised energy consumption? It is expected that a meta-analysis of future research will address such trade-offs and form a holistic consensus towards best practices, necessary platform capabilities, and exploration and mapping strategies for operation in different underwater settings.

## V. Loop Closure

Central to maintaining accurate positioning and producing accurate representations of the environment is loop-closure. All online incremental state estimation algorithms suffer from drift, where errors slowly accumulate and estimated trajectory deviates over time. Loop-closures occur when the vehicle visits the same location and perceives similar data. Vision based approaches utilize the bag-of-words [44] approach. For practical applications, descriptor matching suffer from view point and illumination changes. When the camera perceives the environment at an angle, see Figure 6(c), objects are seen from a specific direction each time. During a traversal inside the Devil's system, FL – see Figure 2(c) – even though the camera followed approximately the same trajectory, as seen in Figure 7(a), the return part of the trajectory drifted over time. This observation inspired a new idea on motion planning:




(a) (b)

Fig. 7. Trajectory estimation inside the Devil cave system, FL. Both trajectories followed the same tunnel. (a) Accumulated drift overtime had the return path diverging. (b) Shorter trajectory, with loop closure resulted in non-diverging path.

when the cave passage is wide enough, a loop is performed resulting in the environment being sensed from the same orientation during the return trajectory. Figure 7(b) presents a trajectory from the same cave, where the loop-closure was achieved, as can be seen the return trajectory matches the inbound one. Please note, utilizing absolute orientation sensors, such as a magnetometer, will eliminate the orientation drift resulting in more accurate trajectories.

## VI. Dense 3D Reconstruction

In this section, we discuss dense 3D reconstruction of the visible surfaces given images and the corresponding camera poses. We distinguish between *online* and *offline* dense reconstruction algorithms and representations. We also discuss the suitability of the representations for each type of environment. Online dense 3D reconstruction is crucial for obstacle avoidance, navigation and planning. It requires efficient and incremental computation. Offline reconstruction, on the other hand, enables visualization, inspection and 3D shape analysis. It can generate accurate, detailed, photorealistic models with a high degree of completeness.

At the core of most 3D reconstruction approaches is a module that estimates a depth map for a reference image. In some cases, especially when high throughput is required,

the depth maps themselves serve as the output representation, while in other cases 3D points or meshes are generated from the depth maps. The minimal number of images for computing a depth map is two; they can be acquired either by two synchronized cameras with a known relative transformation between them, or by a single camera as it moves through the scene. Depth for the pixels of the reference image is estimated by establishing pixel correspondences between the images and triangulating the rays. We refer readers to surveys of the large variety of conventional [45] and learning-based [46] methods due to lack of space.

For 3D mapping using long image sequences as input, it is beneficial to use more than the minimum number of images to improve robustness. Depth map estimation techniques can be extended to the *multi-baseline* setting by considering more images in the computation of matching costs or scores for potential depths of the pixels of the reference image. The most popular paradigms for this computation are plane-sweeping [47] and PatchMatch [48] stereo, followed by depth map fusion [49], [50]. Conventional methods that represent the scene as a collection of depth maps have been effective in a broad range of settings [48], [51], [52], while alternatives based on deep learning surpass them given training data from the target domain [53]–[59], but are slower.

The inherent disadvantage of depth maps is that they are 2.5-D and thus restricted to the viewpoint of the reference image. They can be useful for collision avoidance, but they are not suitable for other tasks. Therefore, world-based 3D representations are needed. The simplest among them is a point cloud obtained by generating a point for each pixel with estimated depth. Point clouds, however, are not effective for visualization, collision avoidance or motion planning, because they do not capture connectivity information.

Occupancy grids, or volumetric representations in general, are well suited for motion planning because they delineate free, occupied and unknown space, but suffer from low resolution due to space complexity. Despite their large memory requirements, occupancy grids are suitable for online operations because they allow incremental updates [60], [61]. Methods such as voxel planes [62] and VoxBlox [63] also maintain approximate surface patches per voxel. Learning-based volumetric representations [64], [65] do not allow incremental updates but benefit from learned priors.

Alternatively, triangles, partially or fully connected to form meshes, can be used as primitives in the representation. Meshes are irregular, which makes accessing them more complex than voxels, but provide visualization, inspection and motion planning capabilities. The footprint of mesh-based 3D models is much smaller than voxel-based ones of similar resolution. Meshes without guarantees of watertightness can be generated from depth maps [66] or by generating surface patch hypotheses in 3D directly [67]. Globally consistent and watertight meshes can be retrieved by solving a volumetric problem, typically approximating the signed distance function from the nearest surface at each voxel. Then manifold surfaces can be extracted via Marching Cubes [68], or using the Poisson Surface Reconstruction algorithm [69]. *Implicit* representations that aim to compute a function whose zero-level set is the surface have been employed by conventional systems [70], [71] yielding high-quality models in real time, but are limited to small volumes. Slower, learning-based methods have also been presented [72]–[74]. (We consider NeRF [75] out of scope since it does not generate explicit 3D structure.)

In our context, the strengths and weaknesses of each representation per environment type are as follows. Flat environments are essentially 2.5-D and thus amenable to be represented by *Digital Elevation Models* (DEMs) or *heightmaps*, which exploit the fact that there is a dominant horizontal surface without holes that separates the occupied space below it from the free space where the robot moves. This assumption lends itself to fast algorithms that can generate watertight meshes from depth map collections [76], [77]. This concept has been extended to scenes with overhangs, such as canopies or balconies, via the introduction of $n$-Layer Heightmaps [78] that can model an a priori unknown number of overlapping layers at several frames per second.

On the other hand, heightmaps are inadequate for modeling the other types of environments which are not 2.5-D. While occupancy grids and octrees can be updated in real-time, their visualization capabilities are limited. Point clouds can also be generated at high rates but fail as a form of visualization. We would argue for surface approximations such as voxel planes [62] or VoxBlox [63] for online operation and for slower algorithms that can generate high-quality watertight meshes for offline applications. Observing an object from a large number of viewpoints oriented towards the object itself, covering most of its surface, is ideal for generating the necessary inputs for Marching Cubes [68] or Poisson Surface Reconstruction [69].

The choice is more challenging when the camera moves in the interior of a cave or a shipwreck. Viewpoints in this case point outward and provide limited coverage on the surfaces, which is suboptimal when the surface extraction algorithm [68], [69] must generate a watertight surface. This leads to hallucinated, invisible completions of the visible surfaces. These completions can be removed with various heuristics, but artifacts often remain. Alternatively, methods that generate partial meshes [66], [67] can be deployed, if holes are preferable to unsupported connections. An example of a partial 3D reconstruction from a wreck with the robot moving among the various surfaces can be seen in Fig. 8.

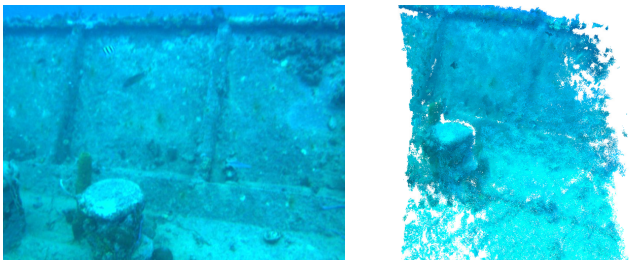


Fig. 8. An input image and a segment of a reconstructed point cloud from the Stavronkita wreck, Barbados.

## VII. Vision with Acoustic Sensor Configuration

Camera sensors capture rich information (i.e., color, texture) about the scene which can be integral in navigation and reconstruction. However, the underwater image formation model, more complex than the in-air counterpart, introduces blue/green color monotony, haze, and further challenges that will affect the quality of state estimation and reconstruction.

Range sensors can be used to complement imaging sensors. Here, we discuss different sensor configurations, including their challenges, trade-offs, and where they are best utilized in the three underwater environments. Differently from out-of-the water, LiDAR solutions underwater are extremely expensive, in the order of US$100,000+, and require a bulky configuration with a laser scanner and a camera [79]. Thus, the main underwater sensors used for navigation and reconstruction are acoustic-based. They consist of single-beam echosounders, mechanical scanning sonars, multi-beam sonars, and sidescan sonars. Unlike cameras, sonars measure azimuth and range, but not elevation. Commercially-available sensors return an image where the intensity of each pixel represents the return strength of the signal. Acoustic sensors, though cheaper than underwater LiDAR, are generally expensive as well, in the order of US$10,000+, and are quite power demanding. There is thus a trade-off between constraints on cost and power and specifications on accuracy for accomplishing exploration tasks.

For flat terrains or straight navigation, side-scan sonars are powerful in covering large areas and providing information to build bathymetric maps. For navigating the insides of overhead environments, multi-beam sonars or mechanical scanning sonars facing forward can provide the robot a view of the obstacles. Note, having a mechanical scanning sonar collect measurements parallel to the image plane can also be beneficial to capture the structure, as shown in [14]. For navigating around an object, a combination of sonars pointing forwards and downwards can help an AUV maintain a safe distance as well as map the object when hovering above it.

Working with sonar data is also not trivial; measurements can be noisy due to multi-path effects, type of objects, etc. However, there have been a few works that have effectively used such sensors. Most of the literature focused only on acoustic sensors for real-time feature tracking, exploiting the imaging data returned by those sensors (e.g., [43], [80], [81]). Recently, a few works appeared fusing both acoustic and visual images to exploit the absolute scale measurements from the acoustic sensors and correct inaccuracies from the cameras [14], [82]. The challenge in fusing both acoustic and visual data is in the difference in sparseness and noise present in the data, requiring more research to effectively use them together for underwater exploration tasks. In addition, as the azimuth measurement is missing in sonar data, to properly match multi-beam or sidescan sonar's measurements and features (i.e., shadows, peaks) to camera image features, one requires knowledge of how the AUV is transitioning (velocity and trajectory changes). Fusing correctly both sensor streams will be beneficial for augmenting the AUV robustness.

## VIII. Conclusions

This paper presented the problem of underwater exploration and mapping of different types of environments by an AUV and highlighted the main subproblems – i.e., exploration/coverage strategies, environment representation, and acoustic-visual sensor fusion – potential solutions, and open challenges to enable a fully-autonomous system.

We posit that the general principle for future work is having algorithmic and system design and development that considers state estimation, representation, and planning holistically rather than separately (as typically done in current work), for AUVs to be capable of exploring underwater environments.

## References


[1] R. M. Eustice, H. Singh, J. J. Leonard, and M. R. Walter, "Visually mapping the RMS Titanic: Conservative covariance estimates for SLAM information filters," *IJRR*, vol. 25, no. 12, pp. 1223–1242, 2006.

[2] M. Gary, N. Fairfield, W. C. Stone, D. Wettergreen, G. Kantor, and J. M. Sharp, Jr, "3d mapping and characterization of sistema zacatón from depthx (DEep Phreatic THermal eXplorer)," in *Sinkholes and the Engineering and Environmental Impacts of Karst*. American Society of Civil Engineers, 2008, pp. 202–212.

[3] D. N. Kernagis, C. McKinlay, and T. R. Kincaid, "Dive logistics of the turner to wakulla cave traverse." 2008.

[4] S. Williams and I. Mahon, "Simultaneous localisation and mapping on the great barrier reef," in *ICRA*, 2004, pp. 1771–1776 Vol.2.

[5] S. Skaff, J. Clark, and I. Rekleitis, "Estimating surface reflectance spectra for underwater color vision," in *BMVC*, 2008, pp. 1015–1024.

[6] M. Roznere and A. Quattrini Li, "Real-time model-based image color correction for underwater robots," in *IROS*, 2019, pp. 7191–7196.

[7] F. Shkurti, I. Rekleitis, and G. Dudek, "Feature tracking evaluation for pose estimation in underwater environments," in *CRV*, 2011.

[8] L. Paull, S. Saeedi, M. Seto, and H. Li, "AUV navigation and localization: A review," *IEEE J. Ocean. Eng.*, pp. 131–149, 2013.

[9] F. Maurelli, S. Krupiński, X. Xiang, and Y. Petillot, "AUV localisation: a review of passive and active techniques," *Int. Journal of Intelligent Robotics and Applications*, pp. 1–24, 2021.

[10] D. Ribas, P. Ridao, J. D. Tardós, and J. Neira, "Underwater SLAM in man-made structured environments," *JFR*, vol. 25, no. 11-12, 2008.

[11] J. L. Schönberger and J.-M. Frahm, "Structure-from-motion revisited," in *CVPR*, 2016.

[12] B. Joshi *et al.*, "Experimental Comparison of Open Source Visual-Inertial-Based State Estimation Algorithms in the Underwater Domain," in *IROS*, 2019, pp. 7221–7227.

[13] A. Quattrini Li *et al.*, "Experimental comparison of open source vision based state estimation algorithms," in *ISER*, 2016.

[14] S. Rahman, A. Quattrini Li, and I. Rekleitis, "SVIn2: A Multi-sensor Fusion-based Underwater SLAM System," *IJRR*, 2022.

[15] A. I. Mourikis and S. I. Roumeliotis, "A multi-state constraint Kalman filter for vision-aided inertial navigation," in *ICRA*, 2007.

[16] F. Shkurti, I. Rekleitis, M. Scaccia, and G. Dudek, "State estimation of an underwater robot using visual and inertial information," in *IROS*, 2011, pp. 5054–5060.

[17] C. Beall, F. Dellaert, I. Mahon, and S. B. Williams, "Bundle adjustment in large-scale 3d reconstructions based on underwater robotic surveys," in *OCEANS, Spain*, 2011, pp. 1–6.

[18] J. Eisele, Z. Song, K. Nelson, and K. Mohseni, "Visual-inertial guidance with a plenoptic camera for autonomous underwater vehicles," *IEEE RAL*, vol. 4, no. 3, 2019.

[19] J. Aulinas, X. Lladó, J. Salvi, and Y. R. Petillot, "Selective submap joining for underwater large scale 6-dof slam," in *IROS*, 2010.

[20] A. Kim and R. M. Eustice, "Real-time visual SLAM for autonomous underwater hull inspection using visual saliency," *IEEE TRO*, 2013.

[21] E. Vargas, R. Scona, J. S. Willners, T. Luczynski, Y. Cao, S. Wang, and Y. R. Petillot, "Robust underwater visual SLAM fusing acoustic sensing," in *ICRA*, 2021.

[22] C. Petres, Y. Pailhas, P. Patron, Y. Petillot, J. Evans, and D. Lane, "Path planning for autonomous underwater vehicles," *IEEE TRO*, 2007.

[23] J. D. Hernández, E. Vidal, M. Moll, N. Palomeras, M. Carreras, and L. E. Kavraki, "Online motion planning for unexplored underwater environments using autonomous underwater vehicles," *JFR*, vol. 36, no. 2, pp. 370–396, 2019.

[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," *RSS*, 2020.

[25] M. Panda, B. Das, B. Subudhi, and B. B. Pati, "A comprehensive review of path planning algorithms for autonomous underwater vehicles," *Int. Journal of Automation and Computing*, pp. 321–352, 2020.

[26] M. Xanthidis, N. Karapetyan, H. Damron, S. Rahman, J. Johnson, A. O'Connell, J. O'Kane, and I. Rekleitis, "Navigation in the presence of obstacles for an agile autonomous underwater vehicle," in *ICRA*, 2020, pp. 892–899.

[27] M. Xanthidis, M. Kalaitzakis, N. Karapetyan, J. Johnson, N. Vitzilaios, J. O'Kane, and I. Rekleitis, "Aquavis: A perception-aware autonomous navigation framework for underwater vehicles," in *IROS*, 2021.

[28] G. Dudek *et al.*, "A visually guided swimming robot," in *IROS*, 2005.

[29] N. Karapetyan, J. 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.

[30] S. Rahman, A. Quattrini Li, and I. Rekleitis, "Contour based reconstruction of underwater structures using sonar, visual, inertial, and depth sensor," in *IROS*, 2019, pp. 8048–8053.

[31] R. Bajcsy, Y. Aloimonos, and J. K. Tsotsos, "Revisiting active perception," *Auton. Robot.*, vol. 42, no. 2, pp. 177–196, 2018.

[32] C. Leung, S. Huang, and G. Dissanayake, "Active SLAM in structured environments," in *ICRA*, 2008, pp. 1898–1903.

[33] S. Suresh, P. Sodhi, J. G. Mangelson, D. Wettergreen, and M. Kaess, "Active slam using 3d submap saliency for underwater volumetric exploration," in *ICRA*, 2020, pp. 3132–3138.

[34] B. Yamauchi, "Frontier-based exploration using multiple robots," in *Int. Conf. on Autonomous agents*, 1998, pp. 47–53.

[35] M. Sheinin and Y. Y. Schechner, "The next best underwater view," in *CVPR*, 2016, pp. 3764–3773.

[36] Y. Girdhar, P. Giguere, and G. Dudek, "Autonomous adaptive exploration using realtime online spatiotemporal topic modeling," *IJRR*, vol. 33, no. 4, pp. 645–657, 2014.

[37] Y. Girdhar and G. Dudek, "Modeling curiosity in a mobile robot for long-term autonomous exploration and monitoring," *Auton. Robot.*, vol. 40, no. 7, pp. 1267–1278, 2016.

[38] L. Paull, S. Saeedi, M. Seto, and H. Li, "Sensor-driven online coverage planning for autonomous underwater vehicles," *IEEE/ASME Trans. Mechatronics*, vol. 18, no. 6, pp. 1827–1838, 2012.

[39] E. Galceran and M. Carreras, "A survey on coverage path planning for robotics," *RAS*, vol. 61(12), pp. 1258 – 1276, 2013.

[40] H. Choset, "Coverage for robotics – a survey of recent results," *Ann. Math. Artif. Intel.*, vol. 31(1-4), pp. 113–126, 2001.

[41] E. Galceran, R. Campos, N. Palomeras, M. Carreras, and P. Ridao, "Coverage path planning with realtime replanning for inspection of 3d underwater structures," in *ICRA*, 2014, pp. 6586–6591.

[42] B. Englot and F. S. Hover, "Three-dimensional coverage planning for an underwater inspection robot," *IJRR*, pp. 1048–1073, 2013.

[43] A. Mallios, P. Ridao, D. Ribas, M. Carreras, and R. Camilli, "Toward autonomous exploration in confined underwater environments," *JFR*, vol. 33, no. 7, pp. 994–1012, 2016.

[44] D. Gálvez-López and J. D. Tardós, "Bags of binary words for fast place recognition in image sequences," *IEEE TRO*, vol. 28, no. 5, 2012.

[45] D. Scharstein and R. Szeliski, "A taxonomy and evaluation of dense two-frame stereo correspondence algorithms," *IJCV*, vol. 47, no. 1-3, pp. 7–42, 2002.

[46] M. Poggi, F. Tosi, K. Batsos, P. Mordohai, and S. Mattoccia, "On the Synergies between Machine Learning and Binocular Stereo for Depth Estimation from Images: a Survey," *PAMI*, 2021.

[47] D. Gallup, J.-M. Frahm, P. Mordohai, Q. Yang, and M. Pollefeys, "Real-time plane-sweeping stereo with multiple sweeping directions," in *CVPR*, 2007.

[48] E. Zheng, E. Dunn, V. Jojic, and J.-M. Frahm, "Patchmatch based joint view selection and depthmap estimation," in *CVPR*, 2014.

[49] S. Galliani, K. Lasinger, and K. Schindler, "Massively parallel multiview stereopsis by surface normal diffusion," in *ICCV*, 2015.

[50] P. Merrell, A. Akbarzadeh, L. Wang, P. Mordohai, J.-M. Frahm, R. Yang, D. Nistér, and M. Pollefeys, "Real-time visibility-based fusion of depth maps," in *ICCV*, 2007.

[51] A. Kuhn, H. Hirschmüller, D. Scharstein, and H. Mayer, "A TV prior for high-quality scalable multi-view stereo reconstruction," *IJCV*, vol. 124, no. 1, pp. 2–17, 2017.

[52] J. L. Schönberger, E. Zheng, J.-M. Frahm, and M. Pollefeys, "Pixelwise view selection for unstructured multi-view stereo," in *ECCV*, 2016.

[53] S. Cheng, Z. Xu, S. Zhu, Z. Li, L. E. Li, R. Ramamoorthi, and H. Su, "Deep stereo using adaptive thin volume representation with uncertainty awareness," in *CVPR*, 2020.

[54] X. Gu, Z. Fan, S. Zhu, Z. Dai, F. Tan, and P. Tan, "Cascade cost volume for high-resolution multi-view stereo and stereo matching," in *CVPR*, 2020.

[55] J. Y. Lee, J. DeGol, C. Zou, and D. Hoiem, "PatchMatch-RL: Deep MVS with Pixelwise Depth, Normal, and Visibility," in *ICCV*, 2021.

[56] X. Ma, Y. Gong, Q. Wang, J. Huang, L. Chen, and F. Yu, "Epp-mvsnet: Epipolar-assembling based depth prediction for multi-view stereo," in *ICCV*, 2021, pp. 5732–5740.

[57] F. Wang, S. Galliani, C. Vogel, P. Speciale, and M. Pollefeys, "PatchmatchNet: Learned Multi-View Patchmatch Stereo," in *CVPR*, 2021.

[58] J. Yang, W. Mao, J. M. Alvarez, and M. Liu, "Cost volume pyramid based depth inference for multi-view stereo," in *CVPR*, 2020.

[59] Y. Yao, Z. Luo, S. Li, T. Fang, and L. Quan, "MVSNet: depth inference for unstructured multi-view stereo," in *ECCV*, 2018.

[60] S. Thrun, "Learning occupancy grid maps with forward sensor models," *Auton. Robot.*, vol. 15, no. 2, pp. 111–127, 2003.

[61] A. Hornung, K. M. Wurm, M. Bennewitz, C. Stachniss, and W. Burgard, "OctoMap: An efficient probabilistic 3D mapping framework based on octrees," *Auton. Robot.*, 2013.

[62] J. Ryde, V. Dhiman, and R. Platt, "Voxel planes: Rapid visualization and meshification of point cloud ensembles," in *IROS*, 2013.

[63] H. Oleynikova, Z. Taylor, M. Fehr, R. Siegwart, and J. Nieto, "Voxblox: Incremental 3d euclidean signed distance fields for on-board mav planning," in *IROS*, 2017, pp. 1366–1373.

[64] G. Riegler, A. Osman Ulusoy, and A. Geiger, "Octnet: Learning deep 3d representations at high resolutions," in *CVPR*, 2017, pp. 3577–3586.

[65] M. Tatarchenko, A. Dosovitskiy, and T. Brox, "Octree generating networks: Efficient convolutional architectures for high-resolution 3d outputs," in *ICCV*, 2017, pp. 2088–2096.

[66] M. Pollefeys *et al.*, "Detailed real-time urban 3D reconstruction from video," *IJCV*, vol. 78, no. 2-3, pp. 143–167, 2008.

[67] Y. Furukawa and J. Ponce, "Accurate, dense, and robust multiview stereopsis," *PAMI*, vol. 32, no. 8, pp. 1362–1376, 2010.

[68] W. E. Lorensen and H. E. Cline, "Marching cubes: A high resolution 3d surface construction algorithm," *Proc. of ACM SIGGRAPH*, vol. 21, no. 4, pp. 163–169, 1987.

[69] M. Kazhdan and H. Hoppe, "Screened poisson surface reconstruction," *ACM Trans. on Graphics (TOG)*, vol. 32, no. 3, p. 29, 2013.

[70] R. Newcombe and A. Davison, "DTAM: Dense tracking and mapping in real-time," in *ICCV*, 2011.

[71] G. Vogiatzis and C. Hernández, "Video-based, real-time multi-view stereo," *IVC*, vol. 29, no. 7, pp. 434–441, 2011.

[72] J. Chibane and G. Pons-Moll, "Neural unsigned distance fields for implicit function learning," *NeurIPS*, vol. 33, 2020.

[73] S. Peng, M. Niemeyer, L. Mescheder, M. Pollefeys, and A. Geiger, "Convolutional occupancy networks," in *ECCV*, 2020, pp. 523–540.

[74] V. Sitzmann, E. Chan, R. Tucker, N. Snavely, and G. Wetzstein, "MetaSDF: Meta-Learning Signed Distance Functions," in *NeurIPS*, 2020, pp. 10 136–10 147.

[75] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng, "NeRF: Representing scenes as neural radiance fields for view synthesis," in *ECCV*. Springer, 2020, pp. 405–421.

[76] R. Pajarola, "Overview of quadtree-based terrain triangulation and visualization," No. 02-01, Inf. & Comp. Sc., UCI, Tech. Rep., 2002.

[77] F. Remondino, "Heritage recording and 3d modeling with photogrammetry and 3D scanning," *Remote sensing*, vol. 3, no. 6, 2011.

[78] D. Gallup, M. Pollefeys, and J.-M. Frahm, "3d reconstruction using an n-layer heightmap," in *Joint Pattern Recognition Symposium*, 2010.

[79] D. McLeod, J. Jacobson, M. Hardy, and C. Embry, "Autonomous inspection using an underwater 3D LiDAR," in *OCEANS*, 2013.

[80] J. Folkesson, J. Leonard, J. Leederkerken, and R. Williams, "Feature tracking for underwater navigation using sonar," in *IROS*, 2007.

[81] E. Westman, A. Hinduja, and M. Kaess, "Feature-based SLAM for imaging sonar with under-constrained landmarks," in *ICRA*, 2018.

[82] M. Roznere and A. Quattrini Li, "Underwater monocular depth estimation using single-beam echo sounder," in *IROS*, 2020.