

Experiments on Surface Reconstruction for Partially Submerged Marine Structures

Georgios Papadopoulos

Department of Mechanical Engineering, Massachusetts Institute of Technology, 77 Massachusetts Avenue, Cambridge, Massachusetts 02139

e-mail: gpapado@mit.edu

Hanna Kurniawati

School of Inf. Technology & Electrical Engineering, University of Queensland, St Lucia, Brisbane, QLD, Australia

e-mail: hannahkur@ua.edu.au

Ahmed Shafeeq Bin Mohd Shariff

Department of Computer Science, National University of Singapore, 21 Lower Kent Ridge Road, Singapore

e-mail: shafeeq@nus.edu.sg

Liang lie Wong

Liang Jie Wong, Tropical Marine Science Institute, National University of Singapore, 21 Lower Kent Ridge Road, Singapore

Tropical Marine Science Institute
e-mail: tmszvli@nus.edu.sg

e-mail: tmswi@nus.edu.sg

Department of Mechanical Engineering, Massachusetts Institute of Technology, 77 Massachusetts Avenue, Cambridge, Massachusetts 02139

02139

Received 7 January 2012; accepted 14 July 2012

Over the past 10 years, significant scientific effort has been dedicated to the problem of three-dimensional (3-D) surface reconstruction for structural systems. However, the critical area of marine structures remains insufficiently studied. The research presented here focuses on the problem of 3-D surface reconstruction in the marine environment. This paper summarizes our hardware, software, and experimental contributions on surface reconstruction over the past few years (2008–2011). We propose the use of off-the-shelf sensors and a robotic platform to scan marine structures both above and below the waterline, and we develop a method and software system that uses the Ball Pivoting Algorithm (BPA) and the Poisson reconstruction algorithm to reconstruct 3-D surface models of marine structures from the scanned data. We have tested our hardware and software systems extensively in Singapore waters, including operating in rough waters, where water currents are around 1–2 m/s. We present results on construction of various 3-D models of marine structures, including slowly moving structures such as floating platforms, moving boats, and stationary jetties. Furthermore, the proposed surface reconstruction algorithm makes no use of any navigation sensor such as GPS, a Doppler velocity log, or an inertial navigation system. © 2013 Wiley Periodicals, Inc.

1. INTRODUCTION

Surface reconstruction of marine structures is an important problem with several applications in marine environments, including marine vehicle navigation, marine environment inspection, and harbor patrol and monitoring. A marine structure is a structure that is fully submerged or partially submerged in the sea. In this paper, we are interested in surface reconstruction of partially submerged marine structures. The surface reconstruction problem is an essential

Direct correspondence to: Georgios Papadopoulos, e-mail: gppapado@mit.edu.

part of the inspection problem that we are interested in. Depending on the application, the required resolution can vary from 3 in (e.g., mine detection tasks) to 20 in (missing or broken parts) or even to 1 m for navigation tasks.

Marine structures are exposed to challenging conditions such as water currents, corrosion, and several hurricanes per year. This exposure results in unpredictable damages to the marine structures, which could be life threatening for the people on and around the structures. Therefore, to ensure safety, thorough structural inspections are required regularly or after a platform is affected by natural disasters. For example, after hurricane Dennis in 2005, structural inspection of Thunder Horse oil platform



Figure 1. Thunder Horse oil platform after Hurricane Dennis in 2005.

(Figure 1) was needed as soon as possible to identify damages on the platform and to reevaluate the platform's condition. Currently, such inspections are performed by human workers through visual observation. They inspect the above-water part of the structures aboard small boats, and use diving equipment to inspect the submerged portions of structures. In both situations, inspectors often lack the time and comfort to inspect the structures and reevaluate the structures' status properly. In addition, visual examination does not give inspectors the ability to track important structural changes. Scanning the structures using robots and automatically reconstructing three-dimensional (3-D) models of the structures from the scanned data will give the inspectors the opportunity to inspect structures from the comfort of their offices, thereby reducing the safety risks of the inspectors and increasing the accuracy of the inspection results.

There is a plethora of scientific work (Newman et al., 2009; Nüchter et al., 2007) dedicated to the reconstruction of 3-D models for land-based structures, however very few works have been dedicated to marine environments. Unpredictable disturbances (such as water currents) and the marine environment itself pose significantly more challenges compared to the challenges posed by terrestrial (shore) environments. Water currents generate disturbance forces on marine vehicles, resulting in large roll and pitch angles. Furthermore, many marine structures are floating platforms, which deflect under strong currents. On top of that, GPS access close to big marine structures is not reliable and other localization sensors developed for marine vehicles lag behind the ones developed for ground robots.

The 3-D surface reconstruction problem often requires gathering 3-D point clouds and registering them under the same coordinate frame. In the case of stationary structures, 2-D laser scanners and localization sensors [GPS, inertial measurement unit (IMU), Doppler velocity log (DVL), or wheel encoders] can be used to gather 3-D data from

stationary environments (Howard et al., 2004). In addition, advanced simultaneous localization and mapping (SLAM) techniques can be used to optimize the estimated trajectory and the resulting map (Scherer et al., 2012; Tong and Barfoot, 2012). On the other hand, in the case of slowly moving structures, special care should be taken in both gathering the 3-D point clouds and registering them under the same coordinate frame. In the absence of knowledge of the motion of the structure, data gathered using 2-D laser scanners combined with localization units cannot be used to reconstruct point cloud representations of structure views.

From our discussion above, it is clear that when dealing with moving structures, 3-D laser scanners with a wide field of view that allows significant overlap between subsequent laser scans should be used to gather point clouds. Then, registration algorithms that are invariant to motion, such as the Iterative Closest Point (ICP) (Besl and McKay, 1992), can be used to register all data gathered under the same coordinate frame, resulting in a point cloud representation of the structure of interest. In the case of stationary structures, researchers commonly use estimates of vehicle positions to initialize the ICP algorithm (Nüchter et al., 2007), however this is not always feasible in the case of moving structures.

This paper presents the hardware and software designs for scanning and reconstructing 3-D surface models of marine structures. We alleviate the difficulty caused by disturbances, moving structures, and sensor errors through hardware and software design. In terms of hardware, we select sensors that would ease construction of 3-D models from scanned data when no positioning sensors are available. Furthermore, we develop a software system that constructs 3-D point cloud models of marine structures from the scanned data and, by using known surface reconstruction techniques, reconstructs 3-D surface models of marine structures. We have successfully tested our system in various conditions in Singapore waters between 2009 and 2011, including operating in the water with up to 2 m/s water currents, and constructing 3-D models of slowly moving structures. All the structures are constructed *without* any information from positioning sensors, such as GPS and DVL.

The problem of reconstructing surfaces from registered point clouds still remains an open issue. There are several algorithms that reconstruct surfaces from point clouds; some of them reconstruct exact surfaces (interpolating surface) by a triangulation that uses a subset of the input point cloud as vertices (Amenta and Bern, 1998; Bernardini et al., 1999). These approaches perform well in clean point clouds and present problems when the point clouds are noisy. Some other approaches reconstruct approximating surfaces by using best-fit techniques (Alliez et al., 2007; Hoppe et al., 1993; Kazhdan et al., 2006). These approaches are robust to noise in the point clouds, however they tend to oversmooth the surfaces and thus an important part of the structure

geometry may be lost. A comparison between several state-of-the-art surface reconstruction techniques can be found in Berger et al. (2011).

In this paper, our main focus is on gathering, registering, and cleaning real 3-D point clouds that can be used to reconstruct surfaces. We also reconstruct 3D surfaces using known techniques, such as the ball pivoting algorithm (Bernardini et al., 1999) and the Poisson surface reconstruction algorithm (Kazhdan et al., 2006). Developing a *new* algorithm that uses noisy point clouds and builds a surface model of structures is crucial, but it is beyond the scope of this paper.

The main contribution in this paper is on the experimental side of robotics. Using standard sensors such as the Velodyne sensor, the micro-bathymetry sonar, and well-known algorithms such as the ICP and the Poisson Surface Reconstruction algorithm, we were able to reconstruct real models of marine structures in the ocean (where 1–2 m/s water currents substantially accentuate the control and sensing errors). The proposed framework can reconstruct surfaces from both parts (the above waterline part and the below waterline part) of marine structures. It is also important to say that the vehicle we assembled is a novel vehicle.

In Section 2, we describe related work. In Section 3, we describe the vehicle and the sensors used in the experiments. Section 4, explains our surface reconstruction algorithms. In Section 5, we present our experimental results, and we close with conclusions and future work in Section 6.

2. RELATED WORK

The 3D model reconstruction problem can be considered to be a special category of the Simultaneous Localization and Mapping (SLAM) problem (Thrun et al., 2005). SLAM was originally introduced by Smith and Cheeseman (1986) and solved using extended Kalman filter (EKF) approaches and feature based maps (Smith et al., 1990). Dissanayake et al. proved that the solution to the SLAM problem is possible, and they proposed another solution to the EKF-SLAM problem (Dissanayake et al., 2001). Thrun, Montemerlo et al. approached the problem from a probabilistic point of view (Montemerlo et al., 2003; Thrun et al., 2004), which was the foundation of the modern approaches that followed: batch smoothing least-squares optimization methods (Dellaert and Kaess, 2006), their incremental equivalent (Kaess et al., 2008), and incorporating loop closures (Cummins and Newman, 2007, 2009).

2.1. 3D Model Reconstruction using Ground Robots

The 3-D model reconstruction problem has attracted substantial research interest over the past 10 years. Although 3-D model reconstruction by marine robots has not been done

sufficiently due to its difficulty, comparable processes have been researched using ground robots. Several robotic platforms were used and different mapping algorithms were proposed. Two different types of sensors (visual sensors and laser sensors) have been used, and each type has strengths and weaknesses. Visual sensors are less expensive, but they do not provide data in \mathbb{R}^3 . However, by using machine learning techniques or vehicle motion or stereo vision, this method can obtain 3-D data, making reconstruction feasible (Izadi et al., 2011; Newcombe and Davison, 2010). Currently, vision-based 3D model reconstruction can be mainly accomplished in indoor environments where the distances are small and the brightness is limited.

In the early stages of large-scale outdoor mapping research, 2-D laser scanners were used to gathered 3-D laser data. Howard et al. gathered 3-D data by mounting a 2-D laser scanner on a segway (Howard et al., 2004). They fused GPS and inertial navigation system (INS) measurements to get estimates of vehicle trajectory and reconstructed 3-D point cloud representations of outdoor environments. Thrun et al. reconstructed 3-D point cloud maps of indoor environments using two laser scanners (2-D). The first laser scanner was mounted horizontally and was used, combined with odometry measurements, to localize the vehicle. The second laser scanner was mounted vertically to scan the environment (Thrun et al., 2000). Because the vertical scanner is not able to scan horizontal surfaces, Zhao and Shibasaki used 2-D laser scanners mounted vertically and shifted 45 degrees to be able to capture horizontal surfaces. In their work, they used GPS and an expensive INS to localize the vehicle; estimates of vehicle trajectory and the laser scanner data were used to reconstruct 3-D maps of outdoor environments (Zhao and Shibasaki, 2001).

In later research, vehicles were capable of directly gathering 3-D point clouds by coupling a 2-D laser scanner with a low-cost nodding actuator or by using a 2-D spinning laser scanner. Harrison and Newman developed a system that was able to gather 3-D point clouds by coupling a 2-D laser scanner with a low-cost nodding actuator. They used odometry measurements and feature extraction algorithms to localize the robot and finally to reconstruct point cloud maps of outdoor environments (Harrison and Newman, 2008). A similar approach was taken by Cole and Newman (2006). Recently, Bosse and Zlot used a 2-D spinning laser scanner to reconstruct 3-D point cloud maps of outdoor environments. They used scan matching algorithms such as the ICP without using any other navigation sensor (Bosse and Zlot, 2009). A similar approach was taken by Holenstein et al. in reconstructing watertight surfaces of caves using 3-D laser data (Holenstein et al., 2011).

A similar approach was also taken by Nuchter et al. They used a ground robot equipped with dead reckoning sensors and a rotating 2-D laser scanner to reconstruct occupancy grid-based maps of outdoor environments. The algorithm proposed was based on the ICP algorithm and dead

reckoning measurements obtained onboard that were used to initialize the ICP algorithm. In addition, efficient data structures such as KD trees, were used to accelerate the ICP algorithm (Nüchter et al., 2007). In another work, Nüchter et al. proposed an algorithm that does not use any navigation sensor. This new approach is based on the existence of skyline features. The skyline features were extracted from panoramic 3-D scans and encoded as strings enabling the use of string matching algorithms for merging the scans. Initial results of the proposed method in the old city center of Bremen are presented. They also use the ICP algorithm for fine registration (Nüchter et al., 2011).

Significant progress on 3D model reconstruction was made by Newman and his group (Newman et al., 2009). They utilized a combination of stereo vision and laser data to first localize the vehicle and then reconstruct 3-D maps of the environment. They employed the Sliding Window Filter developed by Sibley (2006) (that marginalizes poses that are farther away as opposed to full pose batch optimization methods) to approximate the locally optimal trajectory. The proposed system also identifies and utilizes loop closures; their loop closure system, FAB-MAP, is described in work by Cummins and Newman (2007, 2009). After estimating a good trajectory, they used the laser scanner data obtained and the estimated trajectory to produce a 3-D point cloud map of the environment. In a different contribution, they managed to illustrate good localization results using stereo cameras and loop closures for missions over 142 km (Sibley et al., 2010).

The majority of the works presented in the previous paragraphs reconstruct point cloud maps. Konolige et al., instead of using a point cloud representation of the environment, proposed the use of occupancy grid-based maps. They developed a ground vehicle equipped with stereo cameras, GPS, and wheel encoders. Using feature extraction methods, they computed visual odometry. The visual odometry, the GPS, and dead reckoning were used to localize the vehicle and reconstruct occupancy grid maps (Konolige et al., 2006) of outdoor environments. In an expansion of their work, they incorporated loop closures and formulated the problem as a Bayes graph estimation problem, which was solved using nonlinear least-squares techniques, which compute the trajectory that minimizes the squares error (Konolige and Agrawal, 2008).

In recent research, Tong et al. designed a vehicle and algorithms for mapping planetary work site environments. The vehicle used was a rover equipped with a 2-D laser scanner mounted on a pan-tilt mechanism to utilize 3-D data capture, a stereo camera to provide visual odometry measurements, and IMU. Advanced feature extraction algorithms were used to extract features. Extracted features and IMU measurements were used to localize the vehicle using advanced full pose nonlinear least-squares SLAM techniques (Tong and Barfoot, 2012).

2.2. 3D Model Reconstruction using Marine Vehicles

What little research has been done on 3-D model reconstruction by marine robots deals mostly with underwater surfaces and reconstruction of bathymetric maps. In the early years of bathymetric mapping, Singh et al. (2000) presented results taken from deep seas in Italy. Their approach utilized a Long Base Line (LBL) system that localized the underwater vehicle and a set of sonar sensors (side-scan sonar and an actuated pencil profiling sonar) to scan the seabed. At the same time, Bebbett and Leonard reconstructed bathymetric maps from the Charles River using the AUV Odyssey II. They used dead reckoning sensors onboard and a single altimeter sonar (Bennett and Leonard, 2000).

In more recent years of bathymetric research, researchers have been using SLAM algorithms to localize the vehicle. Ruiz et al. used side-scan sonar and onboard sensors to reconstruct bathymetric maps. They manually extracted features out of the sonar data and they used SLAM techniques to localize the vehicle and get better estimates of bathymetric maps (Tena Ruiz et al., 2004). A similar approach was taken by Shkurti et al. (2011). They designed a new vehicle called The AQUA Underwater Robot, which is equipped with IMU and cameras. By extracting features on the seabed and using IMU measurements, they were able to reconstruct the bathymetric map of small areas. Their approach utilizes an EKF SLAM approach. Roman et al. reconstructed bathymetric maps by fusing navigation measurements and relative poses given by the ICP algorithm (Roman and Singh, 2007).

Johnson et al., using an underwater vehicle, presented results on large-scale 3-D model reconstruction and visualization of bathymetric maps from Australian coastal waters. The proposed vehicle was equipped with stereo cameras, sonars, a DVL, accelerometers, depth sensors, GPS (it surfaces to acquire GPS fixes), and it also used Ultra-short Base Line (USBL). Using stereo cameras, they extracted features that are used to compute visual odometry. The visual odometry was combined with readings from other sensors onboard to estimate vehicle trajectory, and then the vehicle trajectory was used to reconstruct the map of the seafloor using readings from the cameras (Johnson-Roberson et al., 2010). They also used meshing algorithms to reconstruct mesh representations of the seafloor.

In a different direction of the surface reconstruction problem using marine vehicles, Fairfield et al. used a hovering underwater vehicle to reconstruct surfaces of caves and tunnels. Their method consisted of a Rao-Blackwellized particle filter with a 3-D evidence grid map representation. In addition, they used occupancy-based grid representation of the environment (Fairfield et al., 2007; Fairfield and Wettergreen, 2008). More recently, significant progress has

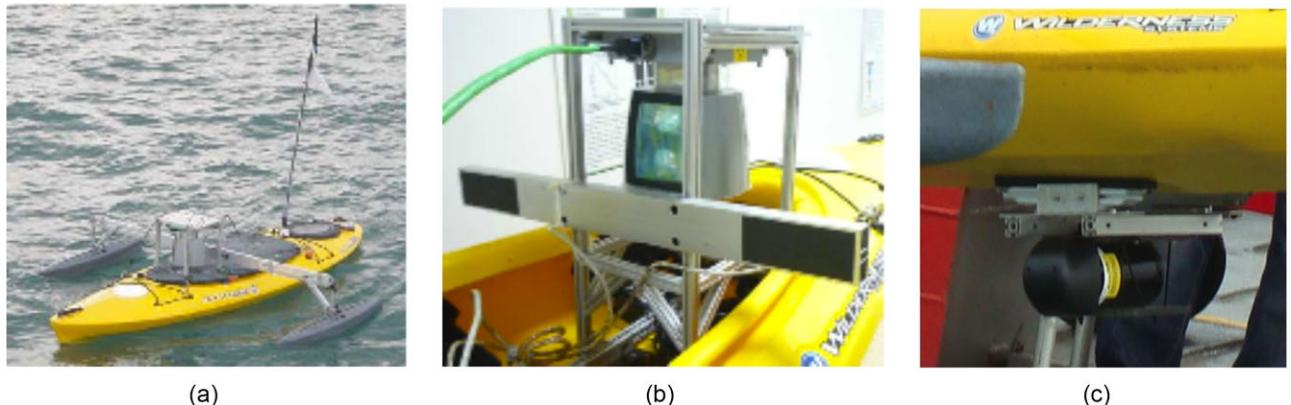


Figure 2. (a) The ASV with Velodyne LiDAR mounted on top. An additional pontoon is attached to the ASV to improve stability. (b) The Velodyne LiDAR and the mounting platform for placing the LiDAR in an inverted configuration on the ASV. (c) The BlueView MB2250 micro-bathymetry sonar mounted in a sideways configuration.

been made on surface reconstruction for ship hull inspection tasks by Hover et al. (2012).

The closest work to that presented in this paper is Leedekerken's recent work (Leedekerken et al., 2010). In Leedekerken's work, a set of 2-D laser scanners is combined with a high-accuracy localization unit that combines GPS-IMU and DVL. In contrast, our work uses a powerful LiDAR Velodyne laser scanner and assumes a GPS-denied environment without using any other localization sensors such as IMU or DVL. In addition, Leedekerken uses a forward-looking sonar that mostly provides bathymetry data rather than data on the submerged part of the structure, whereas we scan the below-water part of the marine structure of interest with side-looking sonar.

Another key difference between our approach and other surface reconstruction approaches (including (Leedekerken et al., 2010)) is that our approach can reconstruct 3-D models of moving structures such as floating platforms and boats. Results, presented later in the paper, indicate that the proposed algorithm can reconstruct 3-D models of slowly moving structures. Slowly moving structures are commonly found in the oil industry (e.g. Figure 20). Reconstructing models of large 3-D moving structures is not easily done using known SLAM techniques that first compute vehicle trajectory and then project laser scanners or cameras data using vehicle trajectory. In the case of moving structures, this cannot be done because vehicle trajectory and laser data obtained from different views are not sufficient to describe the structure. In this work, we are not interested in vehicle trajectory; instead we compute an equivalent trajectory (set of transformation matrices) that minimizes registration error between different scans. In the case of stationary structures, the equivalent trajectory obtained should be close to the actual one.

3. PROPOSED ROBOTIC PLATFORM

Our goal is to generate 3-D models for marine structures. Given the complexity of the marine environment, we would like to use a small but powerful vehicle with high maneuverability that will be able to access hidden places of the structure without crashing into the structure (due to water currents). For this purpose, we use a SCOUT Autonomous Surface Vehicle (ASV)—a kayak with a 3 m length, 0.5 m width, and 90 kg mass. The ASV is equipped with a 245N thruster for propulsion, as well as a steering servo (Curcio et al., 2005) [Figure 2(a)].

No localization sensors such as GPS, INS, or DVL are used in the current paper, but the vehicle is nevertheless equipped with a GPS (Garmin GPS-18) and a compass (Ocean Server OS5000) so that, in the future, it will be able to expand our current work in all possible directions.

To facilitate data capturing and autonomous control capability, the main compartment of the ASV is equipped with a Main Vehicle Computer (MVC). The MVC consists of a pair of single-board computers connected through an Ethernet cable. Each single-board computer is equipped with 1 GB RAM. In addition, one of the single-board computers is equipped with a 120 GB hard drive to facilitate large data capturing capability. The ASV can be controlled remotely using a remote control or autonomously using the well-known autonomy software MOOS (Benjamin et al., 2009; Newman, 2003).

Registration algorithms frequently fail because the two point clouds to be combined have no common features. Given that our goal is to perform surface reconstruction without using any navigation sensors, we use a laser scanner with a wide field of view that allows significant overlap between subsequent scans. In addition, we use a laser scanner that completes a scanning action much faster than the

vehicle and structure motion, so that we do not need to incorporate vehicle and structure motion within a single scan. This makes the procedure simpler and reduces the computational cost of the algorithm. One sensor that meets the desired specifications is the Velodyne HDL-64E S2, shown in Figure 2(b). The Velodyne HDL-64E S2 is a 3-D LiDAR (Light Detection and Ranging) that completes each scanning action in 0.1 s (scanning frequency = 10 Hz).

The Velodyne LiDAR was initially developed for the 2007 Urban DARPA Grand Challenge. Its original configuration was supposed to be mounted on car roofs to perform scanning actions in which a full 360-degree horizontal picture with a vertical arc of 26.80° (from 2° to –24.80°) is captured. The Velodyne is mounted on the kayak in an inverse configuration to maximize the scanning surface. To reduce the amplitude of the rolling motion caused by the marine environment, we installed pontoons [see Figure 2(a)].

To scan the below-water part of the marine structure of interest, we use a 3-D micro-bathymetry sonar (BlueView MB2250 micro-bathymetry sonar). The MB2250-45 sonar uses 256 beams with one degree beam width in elevation. Since we are interested in mapping marine structures, instead of mounting the sonar in a forward-looking configuration such as in Leedekerken et al. (2010), we mount it sideways on the vehicle [Figure 2(c)].

4. SURFACE RECONSTRUCTION ALGORITHMS

The goal of this paper is to illustrate that a 3-D model of marine slow-moving structures in a sea environment can be constructed through the use of a single LiDAR without using any other scanning or navigation sensor. Since this work is an early one to be done on surface reconstruction of marine structures that are partially submerged in a GPS-denied environment, we want to keep things simple; thus, we do not use advanced techniques such as feature extraction and loop closures. In addition, as explained in Section 1, advanced SLAM techniques that optimally localize the agent and reconstruct maps, in their current state, are not easily used to reconstruct the shape of moving structures.

Based on the vehicle design described in the previous section, we propose algorithms to construct the 3-D model of partially submerged marine structures. We propose three different algorithms: the first one (Algorithm 1), is used to reconstruct point cloud representations of the above-water part of marine structures, the second one (Algorithm 2) is used to reconstruct mesh models of the above-water part of marine structures, and the third (Algorithm 3) is used to reconstruct 3-D models (point cloud representations, mesh representations, and occupancy grid-based maps) of both parts (the above- and below-waterline parts) of partially submerged marine structures.

We use scan matching techniques to construct the 3-D model for the above-water part. We then use the trans-

formations, computed by the scan matching algorithm for the above-water part, to construct the 3-D model from a sequence of 2-D sonar data from the below-water part. We then combine the 3-D model of above- and below-water parts to construct a complete 3-D model of the partially submerged marine structure. We construct two types of maps: a low-quality and a high-quality map. The low-quality map can be constructed online and would be useful for navigation purposes. The high-quality map is constructed offline and can be used for inspection purposes.

4.1. Registration Algorithm

To scan a marine structure, the vehicle is driven around the structure of interest, gathering 3-D laser data. The data are logged and saved in the ASV's computer in data structures called point clouds. Each point cloud represents a set of points in \mathbb{R}^3 gathered by a complete 360° rotation of the LiDAR. Since the scanning action is performed much faster than the vehicle's speed, all points within a point cloud are expressed within a common orthogonal reference frame that is aligned to the center of the LiDAR at the starting point of the scanning cycle.

Given two point clouds M_i in $\mathbb{R}^{M \times 3}$ and M_j in $\mathbb{R}^{D \times 3}$ that include common features, we need to compute the transformation ${}^i T_j$ that transforms each point m_k of M_i to each point d_l of M_j . This problem was originally proposed and solved by Besl and McKay in 1992 using the ICP algorithm (Besl and McKay, 1992) by minimizing the following metric:

$$E({}^i R_j, {}^i b_j) = \sum_{k=1}^M \sum_{l=1}^D [w_{k,l} ||m_k - ({}^i R_j d_l + {}^i b_j)||^2]. \quad (1)$$

Here, $w_{k,l}$ is a binary variable as follows: $w_{k,l} = 1$ if m_k is the closest point to d_l within a close limit and is equal to zero otherwise. ${}^i R_j$ and ${}^i b_j$ are the rotation matrix and the translation vector defined in Nüchter et al. (2007). The minimization is done using nonlinear optimization tools such as that of Levenberg-Marquardt.

4.2. Above-Water Surface Reconstruction Algorithm

The vehicle gathers point clouds ($M_1, M_2, \dots, M_{n-1}, M_n$) with a frequency of 10 Hz. We sequentially merge these 3-D point clouds, each in their respective coordinate systems, into a single coordinate system. The transformations between sequential point clouds (${}^0 T_1, {}^1 T_2, \dots, {}^{n-2} T_{n-1}, {}^{n-1} T_n$) is given by the ICP algorithm described in the preceding section. The first point cloud in the sequence is transformed into the coordinate system of the second point cloud. The union of the first two point clouds is then transformed into the coordinate system of the third point cloud, and so on. This process continues until we transform all the point clouds into the coordinate system of the last point cloud in the sequence.



Figure 3. The online map: we use big simplification cell size.

The Velodyne LiDAR generates around 8 MB of data per second (250,000 points per scanning cycle and 10 scanning cycles per second). The computational cost and memory requirements to deal with this amount of data are huge, making ICP impossible to run online. The time for a single merging process in the worst case is $O(N \log N)$, where N is the number of points in the current map. This complexity is dominated by searching for correspondence points. For online mapping, we speed up the search process by using the ASV maximum speed to bound the maximum possible displacement d , and hence limit our search space. Furthermore, we fix a maximum number of possible iterations to ensure termination within the required time.

In addition, we reduce these computational demands in two other ways. First, instead of using the raw data as gathered, we use data from scanning actions performed every Δ_t seconds. Second, we perform spatial subsampling on each point cloud by discretizing the bounding box of the point cloud into a regular grid with user-specified resolution; thus, all the points inside a single grid cell are represented by a single point. By limiting cells to a given size (resolution), we both reduce the amount of data to a reasonable quantity and also cancel out the errors (assuming zero mean noise).

To get the online map, we simplify the data as described above using a large simplification cell size and large Δ_t , as shown in Figure 3 and Algorithm 1. This online map is a low-resolution map that can be used for navigation.

Algorithm 1 Construct3DModel(P, s, t) Construct a 3-D model from a sequence of point clouds P . The inputs s and t are the user-specified spatial and temporal resolution, respectively.

```

1: MergedData = SpatialSubSampling( $P[1], s$ ).
2: for  $i = t$  to  $|P|$  step  $t$  do
3:    $P'$  = SpatialSubSampling( $P[i], s$ ).
4:   Let  $T_0$  be the identity transformation matrix.
5:    $T = \text{ICP}(MergedData, P', T_0)$ .
6:    $MergedData = \text{Transform}(MergedData, T)$ .
7:    $MergedData = MergedData \cup P'$ .
8: Return MergedData.
  
```

To get a higher-quality map, we want to use as much data as we can handle. To do so, we generate an occupancy grid-based map under a probabilistic framework such as Oc-

toMap (Wurm et al., 2010). Because the LiDAR generates a huge amount of data, we still use simplified data (albeit less simplified than above), rather than raw data, in order to get the transformation matrices. We then use the transformation matrices to merge the raw data, resulting in a single, dense 3-D point cloud. An occupancy grid-based map is then generated using OctoMap. The resulting grid-based map is used to get the mesh of the structure using the ball-pivoting algorithm (Bernardini et al., 1999). This process is shown in Figure 4 and in Algorithm 2. This high-resolution map must be generated offline, and can be used for inspection or for further analysis (depending on the application).

Algorithm 2 Construct3DModel (P, s, t) Construct a 3-D high-quality models (mesh representation, dense point clouds, occupancy grid) from a sequence of point clouds P . The inputs s and t are the user-specified spatial and temporal resolution, respectively.

```

1: MergedData = SpatialSubSampling( $P[1], s$ ).
2: for  $i = t$  to  $|P|$  step  $t$  do
3:    $P'$  = SpatialSubSampling( $P[i], s$ ).
4:   Let  $T_0$  be the identity transformation matrix.
5:    $T = \text{ICP}(MergedData, P', T_0)$ .
6:    $DenseMergedData = \text{Transform}(P[i], T)$ .
7:    $DenseMergedData = DenseMergedData \cup P[i]$ .
8:    $MergedData = \text{Transform}(MergedData, T)$ .
9:    $MergedData = MergedData \cup P'$ .
10:  $OccupancyGrid = \text{Octomap}(DenseMergedData)$ 
11:  $MeshModel = \text{BallPivoting}(OccupancyGrid)$ .
12: Return MergedData, OccupancyGrid, MeshModel.
  
```

Two parameters drastically affect the computational cost of our method: Δ_t and the cell size. A small Δ_t results in big computational costs, leaving time intervals between scans that are too small to solve the problem online. On the other hand, given that we are not using other localization sensors, a large Δ_t may result in the failure of the ICP algorithm, since the algorithm cannot merge point clouds gathered from sequential locations that are not sufficiently close to each other (i.e., they get stuck in a local minimum).

In Figure 5, below, we can see the trajectories that the ICP algorithm yields for different values of Δ_t ($C_1, C_2 \dots C_5$). We observe that trajectories corresponding to different values of Δ_t form a sequence with decreasing differences as Δ_t goes to zero (i.e., the sequence trajectories have the Cauchy

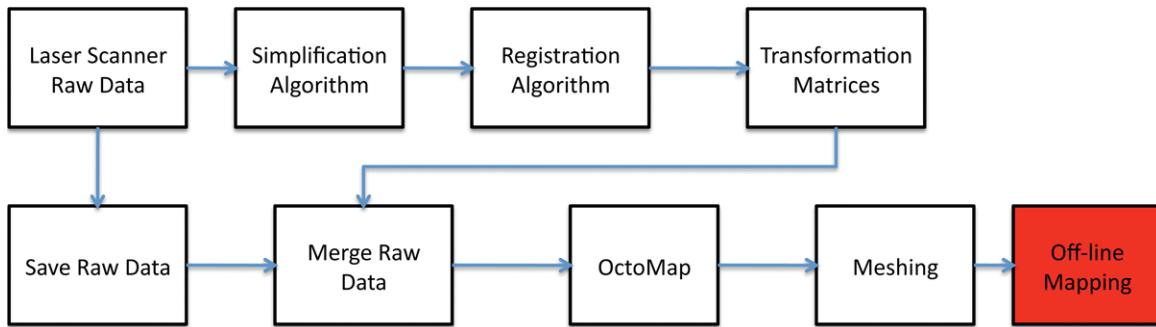


Figure 4. The offline map: Less simplified data are used to get the vehicle's trajectory, which is then used to project raw data resulting in a dense point cloud.

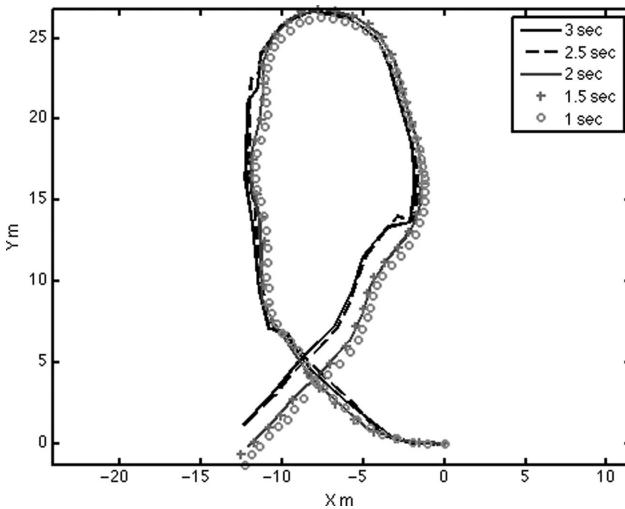


Figure 5. Trajectories generated during the first 100 s of one of the experiments: Trajectories corresponding to different Δ_t , form a sequence with decreasing deference.

property and thus there exists a limit). Therefore, for this particular dataset, the benefit of reducing Δ_t below 1 s is not worth the computational cost for either the online mapping or the offline mapping.

Regardless of cell size, as long as it is small enough to capture geometrical features that are important to the ICP algorithm, it does not affect the localization. For the offline map, simplification cell size generally does not matter, as long as the localization works properly, since we are using the vehicle's trajectory to project dense raw data.¹ However, for the online map, the cell size is bounded by the accuracy we want to have in the map.

¹In the offline map, the voxel size and the probability threshold given in the OctoMap algorithm are important and reflect the resolution we want to capture.

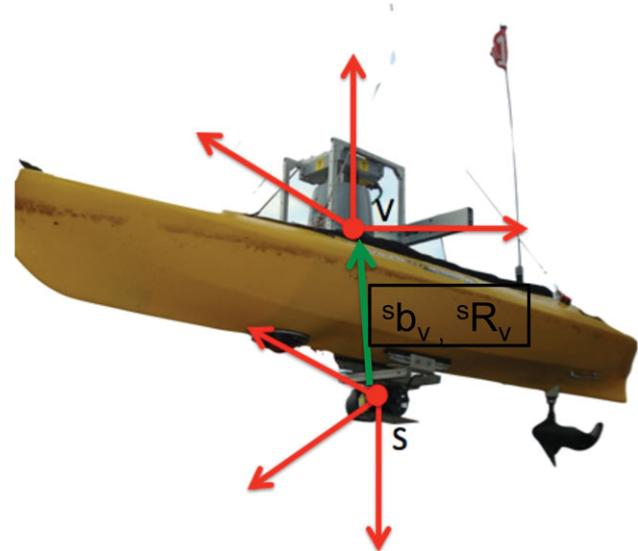


Figure 6. The coordinate systems of LiDAR and sonar. $s b_v$ and $s R_v$ are the translation vector and the rotation matrix that transform the sonar data in the Velodyne coordinate system

4.3. Combined Map

To create a complete 3-D model of the entire partially submerged marine structure, we use the 3-D micro-bathymetry sonar described in the previous section to get the below-water part of the marine structure, and then we combine these data with the model of the above-water part of the structure. The vehicle's trajectory generated by our above-water mapping algorithm and the transformation matrix from the sonar coordinate frame to the LiDAR one (see Figure 6) are used to register the 2-D sonar data into the global 3-D map.

The vehicle's sonar generates 2-D data in the polar coordinate system (r, ϕ, I) , where r and ϕ are the ranges and the angles of the returns and I is the intensity (see Figure 7). Since we want to project 2-D sonar data into the Cartesian global coordinate frame generated by the above-water

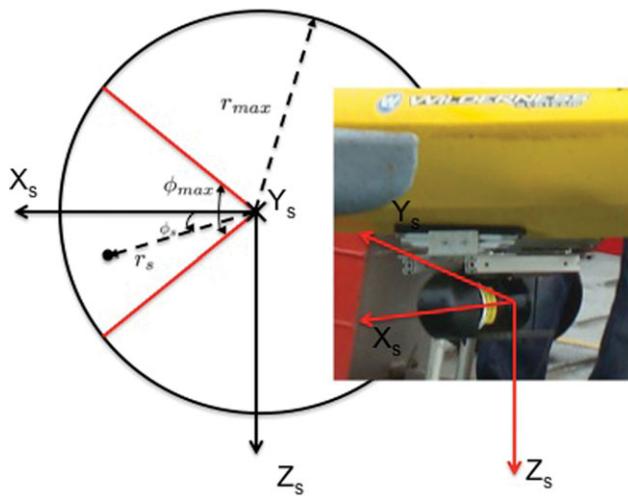


Figure 7. Sonar coordinate system: Sonar gives returns up to 10 m with an angle of 45 degrees.

surface reconstruction algorithm, we transform the 2-D sonar data to an equivalent local Cartesian frame using the following equations:

$${}^s x_s = r \cos \phi, \quad (2)$$

$${}^s z_s = r \sin \phi, \quad (3)$$

$${}^s y_s = 0. \quad (4)$$

The sonar data are then transformed into the current LiDAR coordinate system using Eq. (5). This allows the sonar data to be treated and propagated as equivalent to Velodyne data, as described in Section 4.2,

$${}^v X_s = [{}^s T_v] [{}^s X_s], \quad (5)$$

where $[{}^s T_v]$ is the transformation matrix from the sonar coordinate system to the Velodyne system, as shown in Figure 6, and ${}^s X_s = [{}^s x_s, {}^s z_s, {}^s y_s]^T$. At present, the registration between sonar and LiDAR data is done by manually measuring the transformation from the sonar frame to the LiDAR frame. In the near future, we intend to implement a registration algorithm to do the registration between LiDAR and sonar data (Figure 7). The proposed algorithm for surface reconstruction of the combined model is shown in Figure 8 and Algorithm 3.

Sonar data are noisy, so to clean up the data, we extract objects from the raw sonar data using clustering filtering methods. The main concern is to separate the object from the noise. For this purpose, we use simple background removal on the 2-D intensity map. Initially, we capture a sonar reading when no objects are within the sonar's range. The 2-D intensity map [Figure 9(b)] of this scan becomes the "background" intensity map. For robustness, we do not compare the object intensity map and the background intensity map per pixel. Instead, we use the background map

to find a good threshold to determine if an intensity at a particular pixel can be considered as an object or just noise. This is done by dividing the background intensity map into six clusters [Figure 9(b)] based on the background intensity map and then using the most frequent intensity in each region as the threshold. Given an object intensity map, we divide the map into six regions as in the background intensity map, and we consider a point to be part of an object whenever its intensity is higher than the threshold for the region. Figure 9(a) compares the raw sonar data to the data that have been cleaned using clustering filtering methods.

5. EXPERIMENTAL RESULTS

To evaluate our algorithms, we performed a set of experiments in the Singapore area between January of 2009 and August of 2010. Results presented in this section were initially presented in our ISOPE 2011 and IROS 2011 conference contributions (Kurniawati et al., 2011; Papadopoulos et al., 2011). Results given here are post-processing results, since none of our surface reconstruction algorithms were running on-line.

Our goal was to test our system in rough water environments. However, before testing our system in rough water, we performed preliminary experiments in calm water to ensure the system is ready for rough water testing. To test the system in rough water, first we need to decide the location and the time of testing. We would like to find marine structures, such as jetties, that would be as close as possible to open waters. Therefore, we chose a jetty in a small island (of size less than a square kilometer). To decide the time and the dates of the experiments, we look at the tide and water current predictions to ensure that the environment is challenging enough but not so extreme that it could possibly put our lives and equipment in danger (water currents with speeds greater than 4 m/s are considered, for our case, to be dangerous environments). To go to the operational area, we had to consult a ship and a few boats. We ran the experiments onboard a boat (e.g., Pandan Reservoir experiment, or the boat reconstruction experiment) or from a workstation at the marine structure of interest (e.g., Selat Pauh experiment).

The first experiment was performed in a calm water environment in Pandan Reservoir, Singapore and we reconstructed point clouds and alpha shape representations of the above part of a jetty [see Figure 10(a)]. The vehicle was also equipped with a sonar micro-bathymetry sensor, but for technical reasons we did not manage to gather reliable sonar data to reconstruct the below-water part of the structure.

To give an illustration of the difficulty in merging the scanned data, Figure 10(c) shows the resulting 3-D point clouds scanned by the LiDAR, plotted in one coordinate system. Figure 10(d) shows the resulting point clouds representation of the structure when the coordinate systems

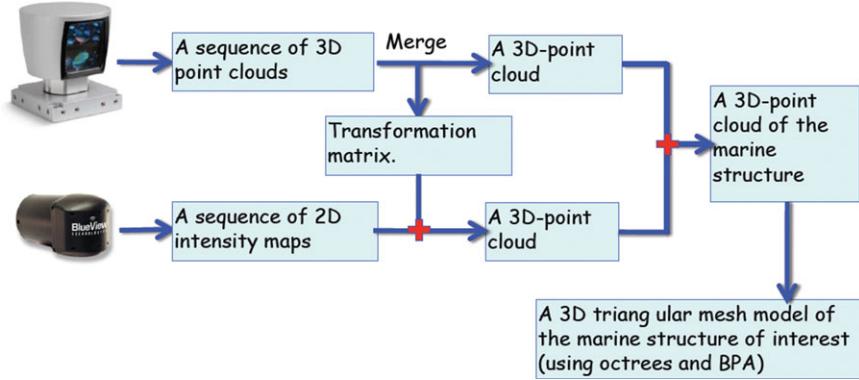


Figure 8. Surface reconstruction of both parts of marine structures, above and below waterline. The ICP algorithm is used to find the transformation matrices between different poses (using the laser scanner data). Then we use the transformation matrices to register the laser scanner data under the same coordinate frame and register the unregistered 2-D sonar data as 3-D data to the same coordinate frame. Then we use a probabilistic framework such as the occupancy grid-based maps to clean up the point clouds, and then we use known surface reconstruction algorithms (ball pivoting algorithm and the Poisson reconstruction algorithm) to reconstruct a 3-D surface model of the marine structure.

Algorithm 3 Construct3DModel(P, S, s, t, b^s_v, R^s_v) Construct a 3-D model (of the above- and below-water parts of marine structures) from a sequence of point clouds P . The inputs s and t are the user-specified spatial and temporal resolution, respectively.

```

1:   MergedData = SpatialSubSampling( $P[1], s$ ).
2:   for  $i = t$  to  $|P|$  step  $t$  do
3:      $P' = \text{SpatialSubSampling}(P[i], s)$ .
4:     Let  $T_0$  be the identity transformation matrix.
5:      $T = \text{ICP}(MergedData, P', T_0)$ .
6:      $MergedData = \text{Transform}(MergedData, T)$ .
7:      $MergedData = MergedData \cup P'$ .
8:      $\{S_{temp}\} = \text{FindSonarLogs}(t, t - 1)$ 
9:     SonarMergedData = SonarRegistration( $\{S_{temp}\}, T, b^s_v, R^s_v$ ).
10:    SonarMergedData = SonarMergedData'  $\cup$  SonarMergedData
11:    OccupancyGrid = Octomap(DenseMergedData)
12:    OccupancyGridSonar = Octomap(SonarMergedData)
13:    SonarAndVelodyneGrid = OccupancyGrid  $\cup$  OccupancyGridSonar
14:    MeshModel = BallPivoting(SonarAndVelodyneGrid).
15:  Return MeshModel, SonarAndVelodyneGrid.
  
```

are transformed to the first coordinate system based on the GPS and compass information alone. Figures 10(b) and 10(e) show the results of our 3-D model reconstruction algorithm on the above data set.

The second experiment was performed in a rough sea water environment in Selat Pauh at the Singapore Straits [Figure 11(a)]. The water currents in Selat Pauh are around 2 m/s. In addition, Selat Pauh is a busy strait with a significant amount of ship traffic, causing high-frequency water wakes that significantly disturb the motion of small marine vehicles. In that particular experiment, we reconstruct a point cloud representation for a slowly moving boat, illustrated

in Figure 11(b). Although the boat is moving slowly, the water currents and wakes cause the boat to drift and move up and down significantly. As an illustration of the effect of water currents and wakes on the scanned data, Figures 11(c) and 11(f) show the 3-D point clouds scanned by the LiDAR over a 2-s period, plotted in the same coordinate system. Figures 11(d) and 11(g) show the data when the coordinate systems are transformed to the first coordinate system based on GPS and compass information alone. Figures 11(e) and 11(h) show the results of applying our 3-D model reconstruction algorithm to the above data set.

In our last experiment, we present results from two missions performed with a jetty located at Pulau Hantu

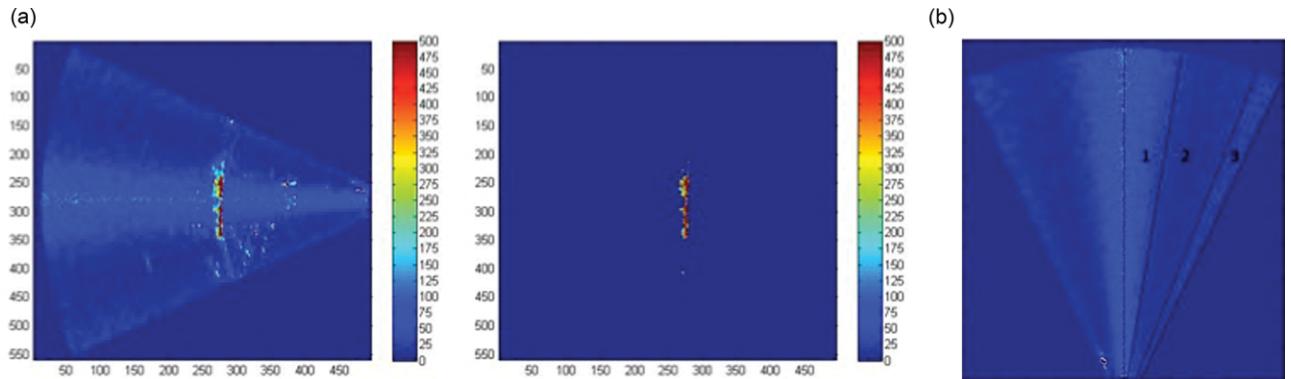


Figure 9. Filtering the sonar data. (a) In the left figure we can see a frame of raw noisy sonar data, in the figure in the right we can see the filtered sonar data using a filter based on clustering. (b) An empty sonar frame divided into 6 clusters.

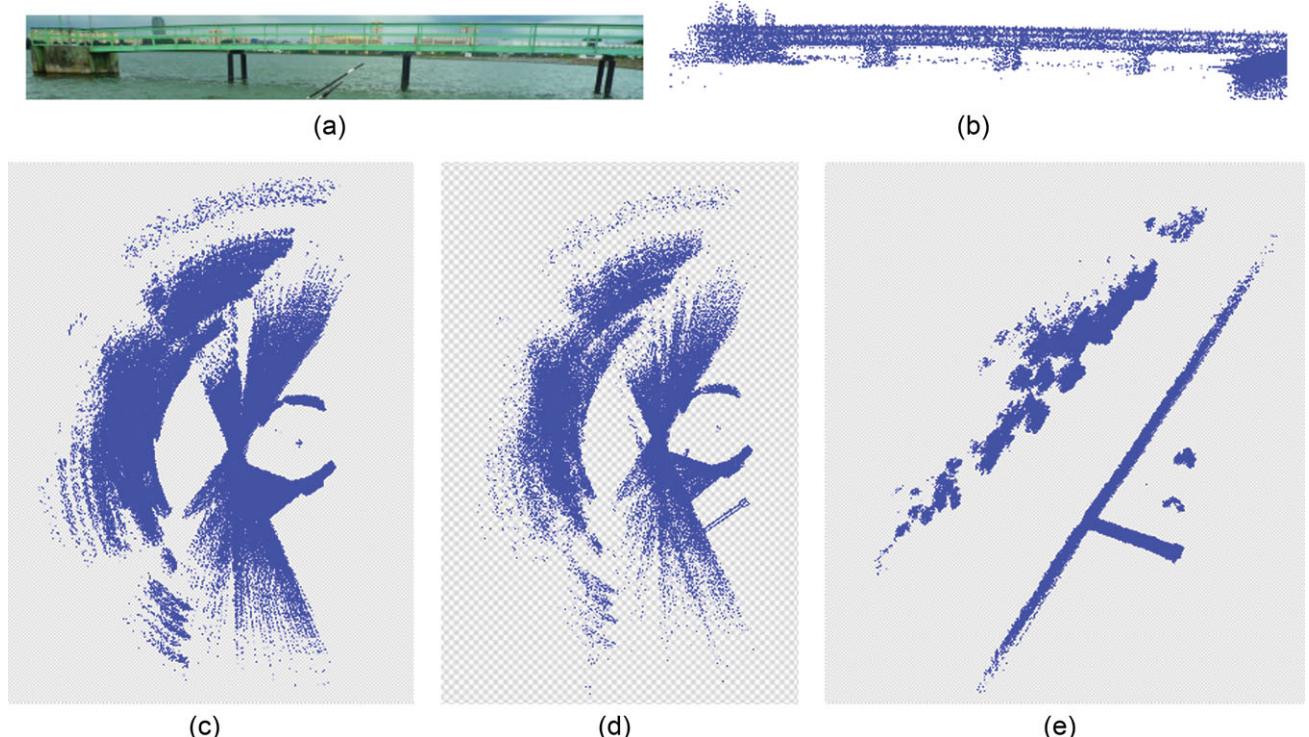


Figure 10. Reconstruction of a jetty in Pandan Reservoir, Singapore. (a) The target jetty. (b) Side view of the jetty model constructed by our algorithm. (c) Top view of multiple frames of the 3-D LiDAR data before processing. (d) Top view of the constructed 3-D model based on GPS and compass information. (e) Top view of the constructed 3-D model generated by our algorithm.

(a small island a few kilometers away from Singapore). We deployed our vehicle from a ship near Pulau Hantu and drove it about the jetty to gather data. We present two missions. The first mission lasted 3 min and gathered data for the above-water part of the jetty. In this mission, we drove the vehicle a distance of about 200 m, making sure that the vehicle approached the structure from different views to recover all the hidden parts of the structure. The second

mission lasted 1 min and gathered data from the above- and below-water parts of the floating platform that was located in front of the jetty [see Figure 17(a)].

We present three different maps of the jetty and the floating platform. The first map is a low-quality point cloud-based map that could be generated online and can be used for navigational purposes [Figures 12(b) and 12(c)]. The second map is a higher-quality mesh-based map

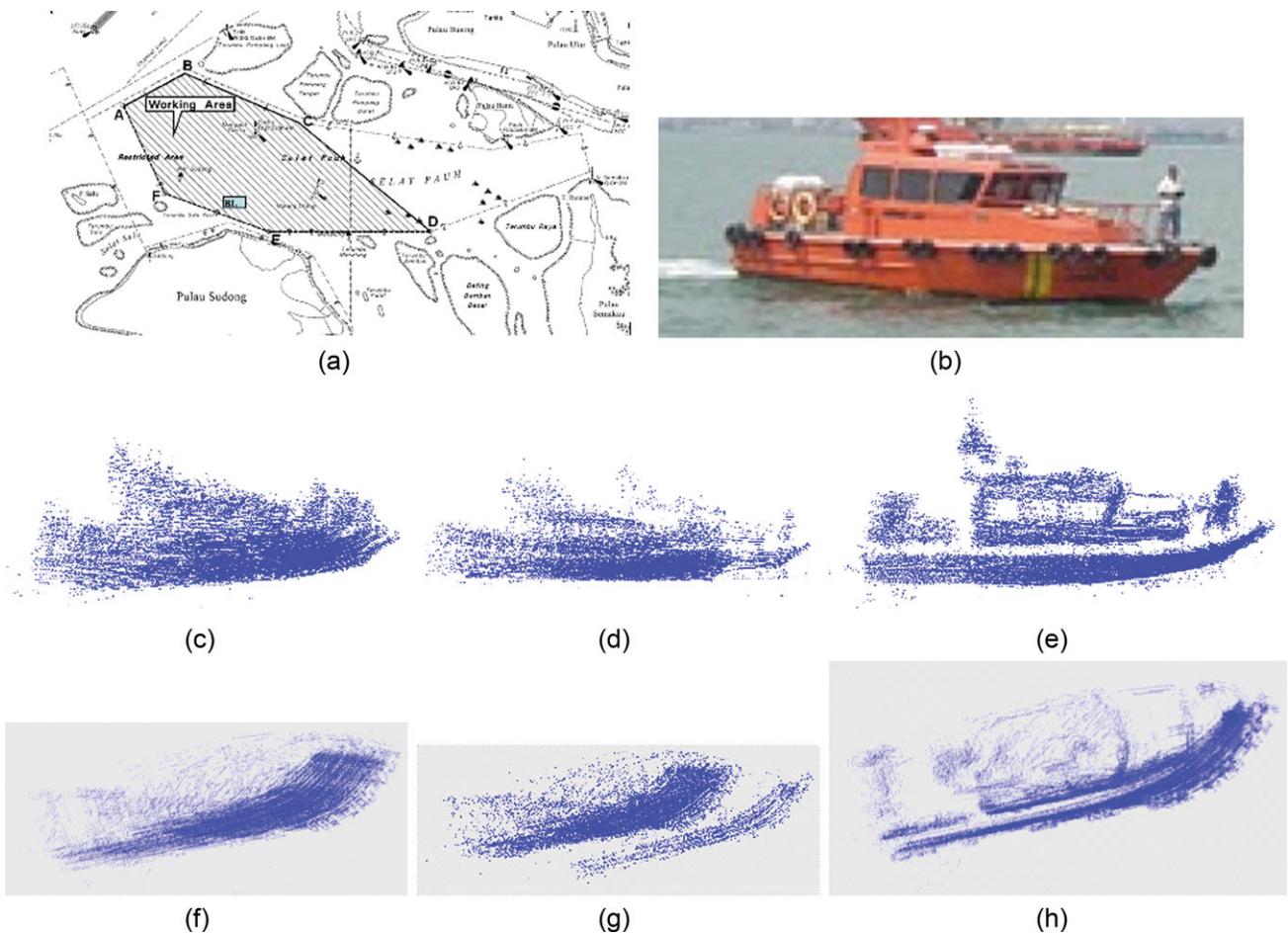


Figure 11. Reconstruction of a slowly moving boat in a rough sea water environment, in Selat Pauh, Singapore. (a) Our operating area. (b) The target boat. (c) Side view of multiple frames of the 3-D LiDAR data before processing. (d) Side view of the constructed 3-D model based on GPS and compass information. (e) Side view of the constructed 3-D model generated by our method. (f) Top view of multiple frames of the 3-D LiDAR data before processing. (g) Top view of the constructed 3-D model based on GPS and compass information. (h) Top view of the constructed 3-D model generated by our method.

(Figures 13–16). The third one combines both the above- and the below-water parts of a single marine structure [Figures 17(b) and 17(c)].

In Figures 12(b) and 12(c), we can see the low-resolution point cloud-based maps of the the jetty for different cell sizes. We can verify that the one that was generated with 30 cm cell size can be generated online. For both cases, $\Delta_t = 1$ s.

In Figures 13 and 14, we present different views of the high-quality mesh-based maps of the jetty (the mesh was reconstructed using the ball pivoting surface reconstruction algorithm). The voxel size used in the occupancy grid generation was 8 cm and $\Delta_t = 1$ s. Here we present two different high-quality maps; the first one (Figure 13) was generated using a high probability threshold for occupied cells, and the second one (Figure 14) was generated using

a low probability threshold for occupied cells resulting in a dense map.

In Figure 15, we can see the mesh-based high-quality map for the above-water part of the jetty, using a low probability threshold for occupied cells and the Poisson surface reconstruction algorithm. From our results we can see that the Poisson surface reconstruction algorithm produces better results than the results the ball pivoting algorithm gives. In addition the computation time of the Poisson surface reconstruction algorithm is on the order of minutes for a point cloud that includes about 1 million points. On the other hand, the ball pivoting algorithm took several hours to run. In Figure 16, we can see the Poisson surface reconstruction results focused on certain parts of the structure. The upper part shows the pillars of the structure; we can see that the algorithm can capture the pillars' geometry pretty well. The

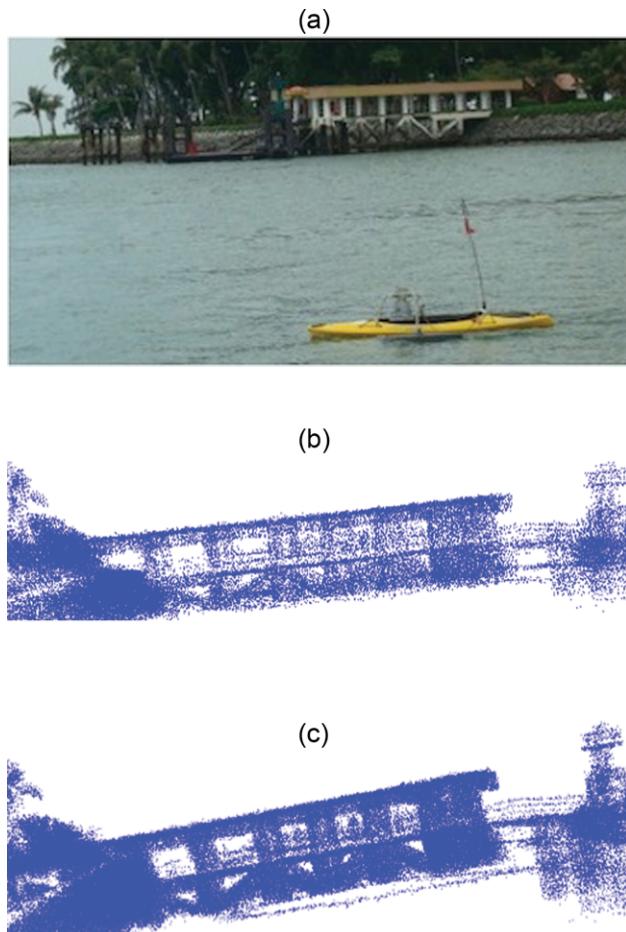


Figure 12. Low-resolution “on-line” model. (a) The marine structure of interest. (b) Low-resolution map, cell size = 30 cm, the mission lasts 3 mins, is generated in 3 mins (3 GHz CPU). (c) Low-resolution map, cell size = 18 cm, the mission lasts 3 mins, is generated in 8 mins (3 GHz CPU).

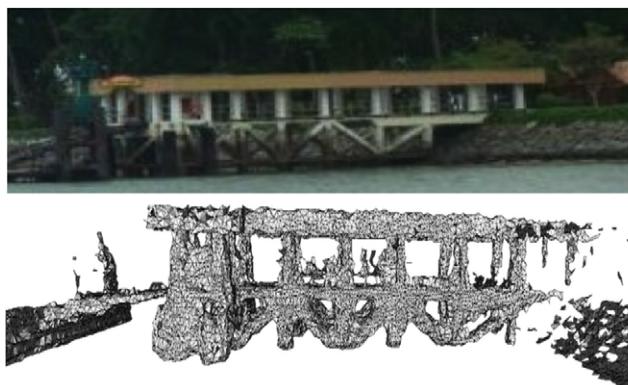


Figure 13. The mesh-based high-quality map for the above-water part of the jetty, using a high probability threshold for occupied cells and the ball pivoting surface reconstruction algorithm. Cell size = 8 cm.

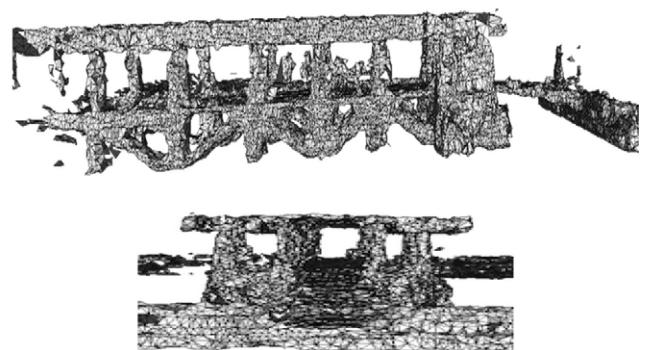
below left part shows the surface of a human body; during the experiments we have people sitting and possibly moving on the jetty, thus the inside part of the structure presents some anomalies. In the below right part of Figure 16, we can see two large buoys that are used to avoid direct collision of boats on the jetty.

Figure 17 shows results from both parts of a marine structure. Specifically, Figures 17(b) and 17(c) show the point-based map for both the above- and below-water parts for the floating platform. Figure 17(d) is a zoomed-in view of the combined mesh-based map. We notice that the buoy that supports the floating platform is flattened due to regular contact with boats. In all cases, the mesh is generated using meshlab (a tool developed with the support of the 3D-CoForm project) (Visual Computing Lab ISTI - CNR, 2010).

5.1. Results Validation

In this section, we validate our experimental results. Probably the best way to evaluate our results is to use a mesh comparison method, such as the one developed by Roy et al. that compares 3D models and produces statistics on mesh similarity (Roy et al., 2004). Unfortunately, we have no access to 3-D models of the jetties we reconstructed, however we infer our results quality in the following ways:

- We compute the quality of all ICP registrations using as a metric the ICP “goodness” as defined in Blanco-Claraco (2009). For the cases we studied in this paper, the mean “goodness” for all ICP registrations is 96.5% with standard deviation about 2%. At the same time, the ICP took on average 277 iterations to converge. Another way to capture the numerical quality of the accuracy of the registration is described by Douillard et al. (2012).
- To show consistent reconstruction from both parts of the marine structure of interest, we present an occupancy



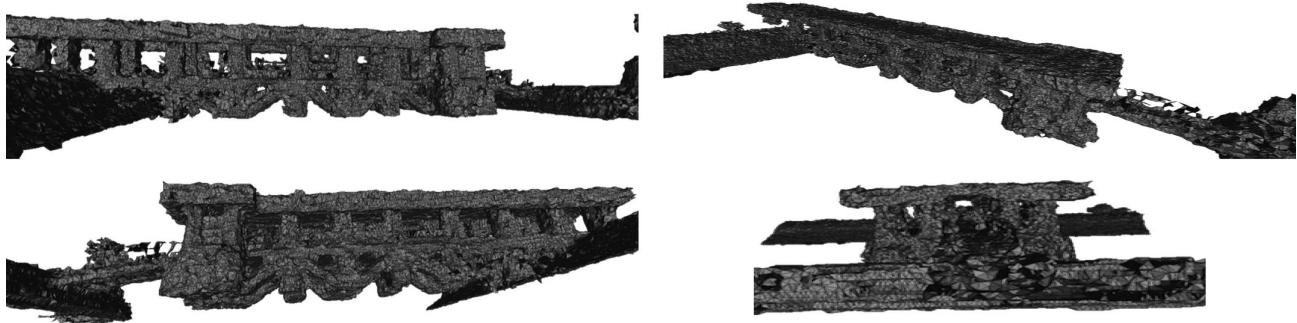


Figure 14. The mesh-based high-quality map for the above-water part of the jetty, using a low probability threshold for occupied cells and the ball pivoting surface reconstruction algorithm. Cell size = 8 cm.

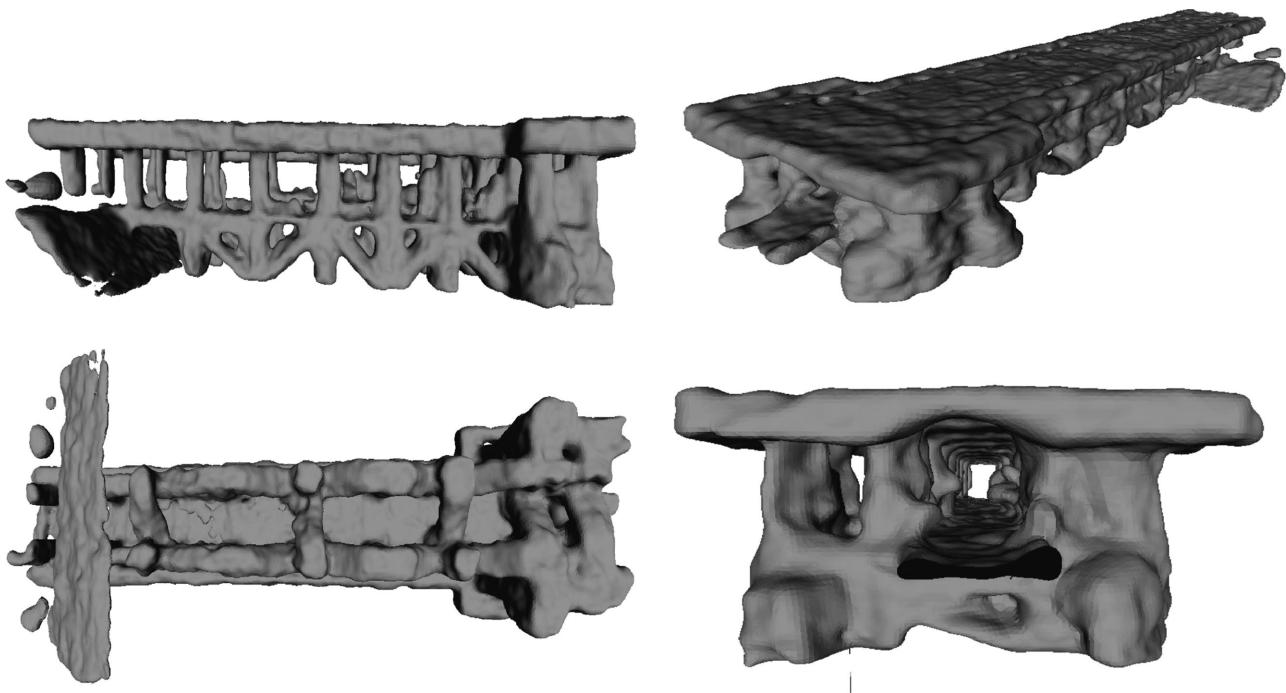


Figure 15. The mesh-based high-quality map for the above-water part of the jetty, using a low probability threshold for occupied cells and the Poisson surface reconstruction algorithm. Cell size = 8 cm.

grid-based map of the above- and below-water parts of the structure and the probability of each cell to be occupied. The color indicates the probability of the cells to be occupied, Figure 18. Low probability gives blue colors and high probability gives red colors. From this figure we can clearly see the waterline and three areas: The above-water part of the structure, the below-water part of the structure, and the interface area between the LiDAR and the sonar data.

The probability of the above-water part of the structure cells to be occupied is close to uniform and less dense than the ones from the below-water part of the structure.

Generally, laser scanners produce denser point clouds than the sonars, however this is not the case here because to avoid huge data sets we simplify/regularize our raw data using very small simplification cell-size (e.g., a few millimeters to a few centimeters). Still we can see that the above-water part of the structure is reconstructed consistently.

The underwater part of the structure is also reconstructed consistently, and the probability of the cells of the underwater part of the structure to be occupied is high (the cells with the highest probability are located in the far left side of the structure where the mission started

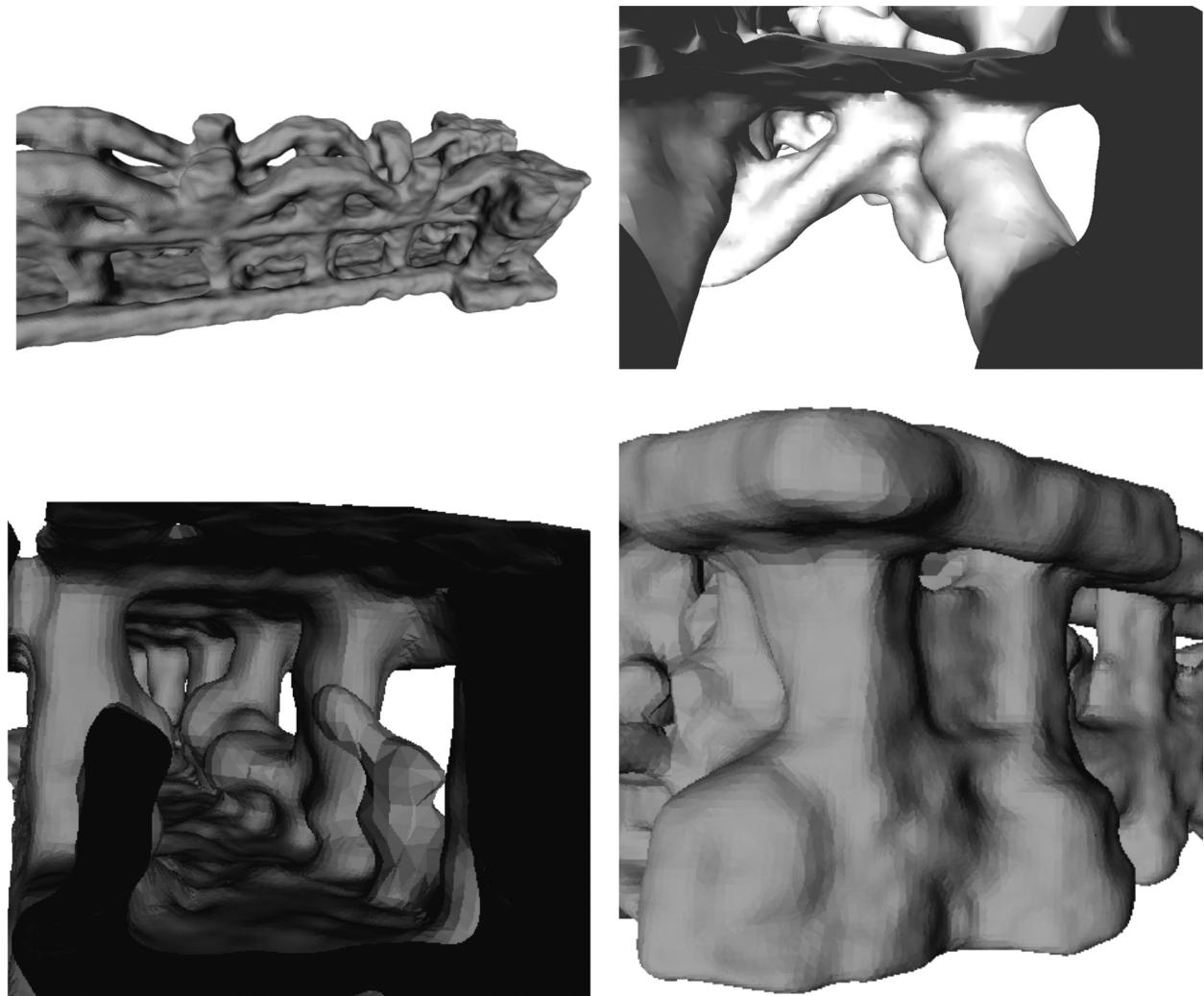


Figure 16. Zoom-in of the mesh-based high-quality map for the above-water part of the jetty, using a low probability threshold for occupied cells and the Poisson reconstruction algorithm. Upper part shows the pillars of the structure. Below left part shows the surface of a human body (during the experiments we have people sitting and possibly moving on the jetty). Below right shows two large buoys that are used to avoid direct collision of boats on the jetty. Cell size = 8 cm.

and the vehicle probably was sitting there accumulating sonar data from the same position for a few seconds). In the interface area between the LiDAR and sonar data, the probability of a cell to be occupied, as expected, is lower than the below-water part of the structure but is high enough to achieve surface reconstruction in the interface area. In the far right area of the underwater part of the structure, we do not get enough sonar returns. This is probably due to the fact that in this particular location, the vehicle started turning toward the left and thus the side-looking sonar (with a very narrow beam) instead of pointing next to where the kayak was pointing toward

the previously scanned areas (e.g., in the left side of the platform, behind the kayak), and then we stopped the mission. This problem can be solved using a better sonar sensor such as the Didson sensor [used by Hover et al. (2012)] or designing trajectories—that take into account vehicle dynamics—to be informative enough to reconstruct structures (Papadopoulos et al., 2013).

- We measure the characteristic lengths of the jetty using Google Maps and compare them to the ones we get from the reconstructed jetty. To reduce the effect of constant errors such as conversion from “Velodyne units” to metric units and errors due to the direction the aerial picture was

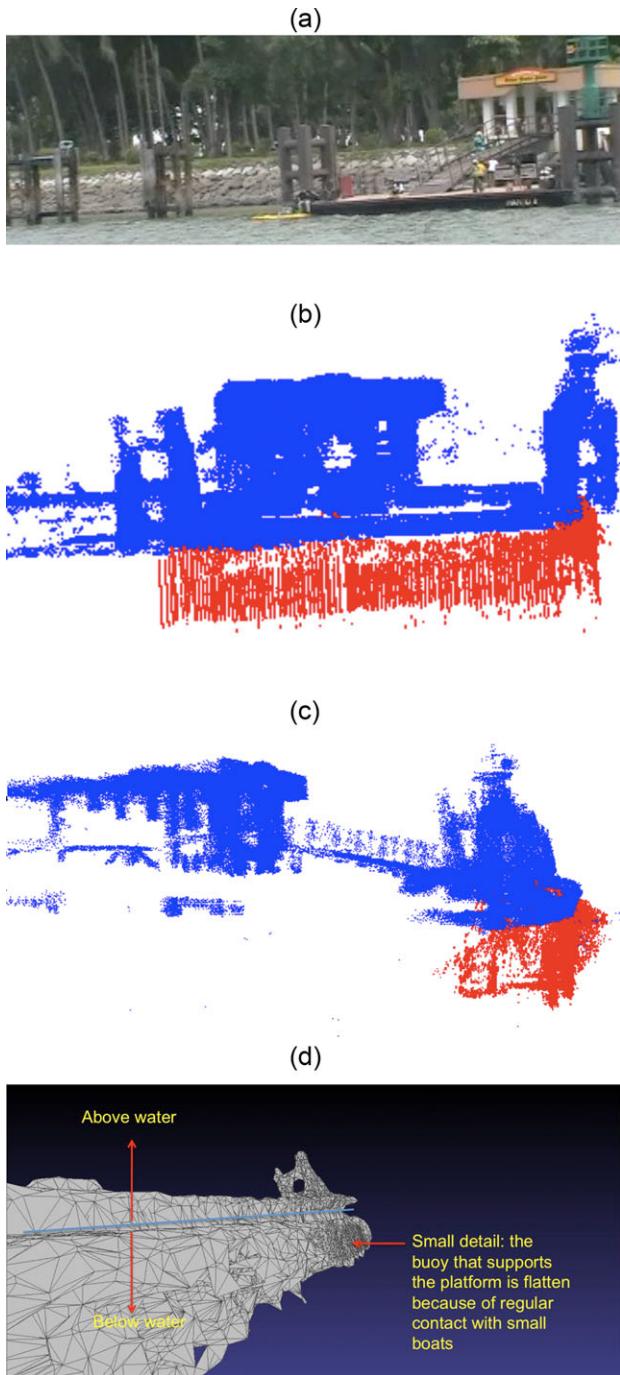


Figure 17. Combined map: (a) Picture of the front part of the structure. (b) Above- and below-water parts of marine structure, point cloud-based map, front view. In red color below water part, in blue color above water part. (c) Above- and below-water parts of marine structure, point cloud-based map, side view. In red color below water part, in blue color above water part. (d) Above- and below-water mesh-based map: Small details.

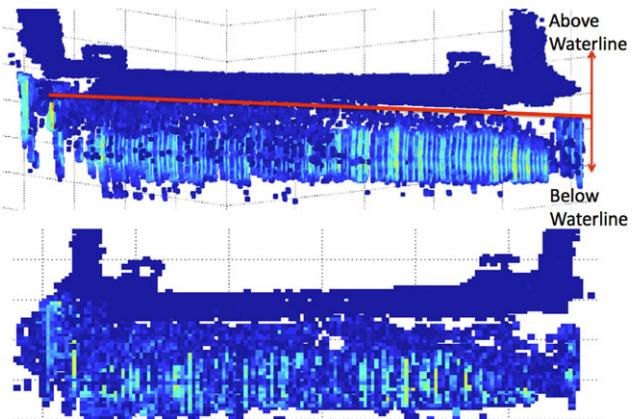


Figure 18. Upper part: A point cloud representation of the above- and below-water parts of the structure. The color indicates the density of the points. Lower part: An occupancy grid-based map of the above- and below-water parts of the structure. The color represents the probability of the cells to be occupied. Low probability gives blue colors and high probability gives red colors.

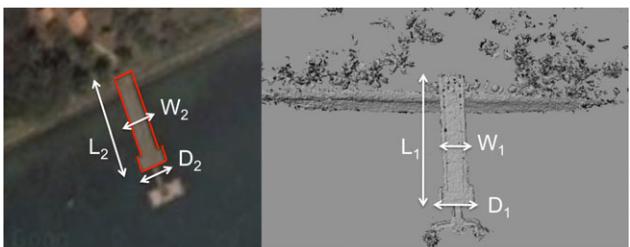


Figure 19. Comparison between our results and pictures of the actual jetty taken from Google Maps.

taken from, instead of comparing the actual lengths we reconstruct dimensionless numbers that characterize the structure (Figure 19). In our case, we have $L_1/W_1 = 4.9$, which is close to $L_2/W_2 = 4.8$ (error 2%), i.e., the ones obtained using the Google Maps. In addition, $L_1/D_1 = 3.5$, which is also close to $L_2/D_2 = 3.4$ (error 2.8%), i.e., the ones obtained using the Google Maps, where L_1 , W_1 , D_1 are the characteristic lengths of the reconstructed jetty and L_2 , W_2 , D_2 are the characteristic lengths of the jetty as measured using Google Maps.

6. CONCLUSIONS AND FUTURE WORK

In this paper, we present a hardware and software system to reconstruct 3-D surface models of marine structures that are partially submerged. In particular, this paper made the following contributions.

- Using off-the-shelf sensors and a commercial robotic kayak developed by Curcio et al. (2005), we assembled a novel surface vehicle that is capable of using a powerful 3-D laser scanner (Velodyne) and a side-looking sonar to scan marine structures both above and below the water-line.
- Using Velodyne and sonar data, without using any other navigation sensor such as GPS, DVL, or INS, we propose a method to reconstruct 3-D models of partially submerged slowly moving marine structures.
- We present various surface reconstruction experimental results including results from sea environments with water currents around 1–2 m/s. More specifically, we present three experimental results of mapping the above-water part of marine structures and one experimental result of mapping both the above- and below-water parts of marine structures. To the best of our knowledge, these are the first results on 3-D surface reconstruction experiments from rough waters and slowly moving structures. Experiments presented here are as realistic as possible. We present results in rough waters with moving structures (floating platforms and boats) under tidal effects (1–3 m) and water disturbances arising from big ships moving in the busy Singapore waters.

The results show that our robotic system for 3-D model construction of marine structures is reliable to operate in rough sea water environments. The resulting scanned data indicate that the LiDAR's mounting platform does not pose significant degradation in the quality of the scanned data. Furthermore, because of the high scanning frequency of the Velodyne LiDAR and sufficient overlap between point clouds generated by different scanning cycles, the simple merging algorithm we propose is sufficient to construct a rough 3-D model of marine structures.

We also show results indicating that the proposed algorithm can reconstruct 3-D models of slowly moving structures. Slowly moving structures are commonly found in the oil industry (e.g., Figure 20). Reconstructing models of large 3-D moving structures is not easily done using standard SLAM techniques that first compute vehicle trajectory

and then project laser scanners or cameras data using vehicle trajectory. In the case of moving structures, this cannot be done because vehicle trajectory and laser data obtained from different views are not sufficient to describe the structure. In contrast to the SLAM problem, we are not interested in vehicle trajectory. Instead we compute an equivalent trajectory (set of transformation matrices) that minimizes the registration error between different scans. In the case of nonmoving structures, the equivalent trajectory obtained should be close to the actual one.

One thing we notice is that extremely strong currents excite vehicle roll and pitch motions, causing some of the LiDAR scans to be empty (taken with the LiDAR looking at the sky).² If we try to merge one of these LiDAR scans, the ICP algorithm is likely to fail. To avoid this problem, before we call the surface reconstruction module we filter out LiDAR scans with an extremely low amount of data.

Despite the above promising results, there is still plenty of room for improvement. In this work, we have assumed a GPS-denied environment, without using any other navigation sensors such as DVL or INS. Of course whenever possible, we are interested in using GPS and other navigation sensors. Furthermore, we would like to use more SLAM advanced techniques such as feature extraction and loop closures to bound localization accuracy, which crucially affects the quality of the map. However, as indicated above, special treatment should be considered in the case of moving structures such as floating platforms (Lin and Wang, 2010; Hsiao and Wang, 2011). In addition, more research needs to be done on integrating data from above- and below-water parts of marine structures. More specifically, we would like to use registration algorithms to better align the above-water part of the structure with the below-water parts. Another possible direction of future work will be the combination of the ICP solver with randomization methods to find the global minimum in Eq. (1). We believe that as the number of sample points increases, ICP solvers combined with randomization methods will converge to the globally optimal transformation matrix (given that the samples are chosen correctly).

ACKNOWLEDGMENTS

This work was supported by the Singapore-MIT Alliance for Research and Technology (SMART) Center for Environmental Sensing and Modeling (CENSAM). We would also like to thank Dr. Leedekerken and Professor John Leonard for thoughtful discussions, and Andrew Patrikalakis for his help in the initial vehicle design. We thank the anonymous reviewers for their input.



Figure 20. Different types of oil platforms; the majority of the platforms are floating and slowly moving. The picture is from the Office of Ocean Exploration and Research.

²In our case we try to minimize this effect by installing pontoons on the kayak.

APPENDIX A: INDEX TO MULTIMEDIA EXTENSIONS

Extension	Media Type	Description
1	Video	It shows some of the experiments presented in this paper

REFERENCES

- Alliez, P., Cohen-Steiner, D., Tong, Y., & Desbrun, M. (2007). Voronoi-based variational reconstruction of unoriented point sets. In ACM International Conference Proceeding Series (vol. 257, pp. 39–48).
- Amenta, N., & Bern, M. (1998). Surface reconstruction by voronoi filtering. *Discrete and Computational Geometry*, 22, 481–504.
- Benjamin, M., Leonard, J., Schmidt, H., & Newman, P. (2009). An overview of MOOS-ivp and a brief users guide to the ivp helm autonomy software. Technical Report CSAIL-2009-028, MIT.
- Bennett, A. A., & Leonard, J. J. (2000). A behavior-based approach to adaptive feature detection and following with autonomous underwater vehicles. *IEEE Journal of Oceanic Engineering*, 213–226.
- Berger, M., Levine, J. A., Nonato, L. G., Taubin, G., & Silva, C. T. (2011). A Benchmark for Surface Reconstruction. *ACM Transactions on Graphics*, 32(2), 20:1–20:17 (to be presented at SIGGRAPH 2013).
- Bernardini, F., Mittleman, J., Rushmeier, H., Silva, C., Taubin, G., & Member, S. (1999). The ball-pivoting algorithm for surface reconstruction. *IEEE Transactions on Visualization and Computer Graphics*, 5, 349–359.
- Besl, P. J., & McKay, N. D. (1992). A method for registration of 3D shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2), 239–256.
- Blanco-Claraco, J. (2009). Contributions to localization, mapping and navigation in mobile robotics. Ph.D. thesis, University of Malaga, Malaga, Spain.
- Bosse, M., & Zlot, R. (2009). Continuous 3D scan-matching with a spinning 2D laser. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA) (pp. 4312–4319).
- Cole, D., & Newman, P. (2006). Using laser range data for 3D SLAM in outdoor environments. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA).
- Cummins, M., & Newman, P. (2007). Probabilistic appearance based navigation and loop closing. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA) (pp. 2042–2048).
- Cummins, M., & Newman, P. (2009). Highly scalable appearance-only SLAM-FAB-MAP 2.0. In Proceedings of the Robotics: Science and Systems (RSS), Seattle.
- Curcio, J., Leonard, J., & Patrikalakis, A. (2005). SCOUT—A low cost autonomous surface platform for research in cooperative autonomy. In Proceedings of the IEEE/MTS OCEANS Conference and Exhibition, Washington, DC.
- Dellaert, F. and Kaess, M. (2006). Square Root SAM: Simultaneous localization and mapping via square root information smoothing. *International Journal of Robotics Research*, 25(12), 1181–1203.
- Dissanayake, M. W. M. G., Newman, P., Clark, S., Durrant-Whyte, H. F., & Csorba, M. (2001). A solution to the simultaneous localization and map building (SLAM) problem. *IEEE Transactions on Robotics and Automation*, 17, 229–241.
- Douillard, B., Nourani-Vatani, N., Johnson-Roberson, M., Williams, S., Roman, C., Pizarro, O., Vaughn, I., & Inglis, G. (2012). Fft-based terrain segmentation for underwater mapping. In Proceedings of Robotics: Science and Systems, Sydney, Australia.
- Fairfield, N., Kantor, G. A., & Wettergreen, D. (2007). Real-time SLAM with octree evidence grids for exploration in underwater tunnels. *Journal of Field Robotics*, 24, 3–21.
- Fairfield, N., & Wettergreen, D. (2008). Active localization on the ocean floor with multibeam sonar. In Proceedings of the IEEE/MTS OCEANS Conference and Exhibition (pp. 1–10).
- Harrison, A., & Newman, P. (2008). High quality 3D laser ranging under general vehicle motion. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA).
- Holenstein, C., Zlot, R., & Bosse, M. (2011). Watertight surface reconstruction of caves from 3D laser data. In Proceedings of the IEEE/RSJ Intl. Conference on Intelligent Robots and Systems (IROS) (pp. 3830–3837).
- Hoppe, H., DeRose, T., Duchamp, T., McDonald, J., & Stuetzle, W. (1993). Mesh optimization. In Proceedings of the 20th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH '93 (pp. 19–26), New York, ACM.
- Hover, F. S., Eustice, R., Kim, A., Englot, B., Johannsson, H., Kaess, M., & Leonard, J. J. (2012). Advanced perception, navigation and planning for autonomous in-water ship hull inspection. *International Journal of Robotics Research*, 31(12), 1445–1464.
- Howard, A., Wolf, D. F., & Sukhatme, G. S. (2004). Towards 3D mapping in large urban environments. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (vol. 3, pp. 419–424).
- Hsiao, C.-H., & Wang, C.-C. (2011). Achieving undelayed initialization in monocular slam with generalized objects using velocity estimate-based classification. In Robotics and Automation (ICRA), 2011 IEEE International Conference (pp. 4060–4066).
- Izadi, S., Kim, D., Hilliges, O., Molnyneaux, D., Newcombe, R., Kohli, P., Shotton, J., Hodges, S., Freeman, D., Davison, A., & Fitzgibbon, A. (2011). Kinectfusion: Real-time 3D reconstruction and interaction using a moving depth camera. In Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology (pp. 559–568), ACM.
- Johnson-Roberson, M., Pizarro, O., Williams, S. B., & Mahon, I. (2010). Generation and visualization of large-scale three-dimensional reconstructions from underwater robotic surveys. *Journal of Field Robotics*, 27(1), 21–51.

- Kaess, M., Ranganathan, A., & Dellaert, F. (2008). iSAM: Incremental smoothing and mapping. *IEEE Transactions in Robotics*, 24(6), 1365–1378.
- Kazhdan, M., Bolitho, M., & Hoppe, H. (2006). Poisson surface reconstruction. In *Proceedings of the Fourth Eurographics Symposium on Geometry Processing*.
- Konolige, K., & Agrawal, M. (2008). FrameSLAM: From bundle adjustment to real-time visual mapping. *IEEE Transactions in Robotics*, 24(5), 1066–1077.
- Konolige, K., Agrawal, M., Bolles, R. C., Cowan, C., Fischler, M., & Gerkey, B. (2006). Outdoor mapping and navigation using stereo vision. In *Proceedings of the International Symposium on Experimental Robotics*.
- Kurniawati, H., Schulmeister, J., Bandyopadhyay, T., Papadopoulos, G., Hover, F., & Patrikalakis, N. (2011). Infrastructure for 3D model reconstruction of marine structures. In *Proceedings of the International Offshore and Polar Engineering Conference*, International Society of Offshore and Polar Engineers (ISOPE).
- Leedekerken, J. C., Fallon, M. F., & Leonard, J. J. (2010). Mapping complex marine environments with autonomous surface craft. In *Proceedings of the International Symposium on Experimental Robotics (ISER)*, Delhi, India.
- Lin, K. H., & Wang, C. C. (2010). Stereo-based simultaneous localization, mapping and moving object tracking. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (pp. 3975–3980).
- Montemerlo, M., Thrun, S.,oller, D., & Wegbreit, B. (2003). FastSLAM 2.0: An improved particle filtering algorithm for simultaneous localization and mapping that provably converges. In *Proceedings of the 18th International Joint Conference on Artificial Intelligence* (pp. 1151–1156).
- Newcombe, R., & Davison, A. (2010). Live dense reconstruction with a single moving camera. In *Proceedings of the IEEE International Conference Computer Vision and Pattern Recognition* (pp. 1498–1505).
- Newman, P. (2003). MOOS—A mission oriented operating suite. Technical Report OE2003-07, Department of Ocean Engineering, MIT.
- Newman, P., Sibley, G., Smith, M., Cummins, M., Harrison, A., Mei, C., Posner, I., Shade, R., Schroter, D., Murphy, L., Churchill, W., Cole, D., & Reid, I. (2009). Navigating, recognizing and describing urban spaces with vision and lasers. *International Journal of Robotics Research*, 28, 1406–1433.
- Nüchter, A., Gutev, S., Borrmann, D., & Elseberg, J. (2011). Skyline-based registration of 3D laser scans. *Geo-spatial Information Science*, 14, 85–90.
- Nüchter, A., Lingemann, K., Hertzberg, J., & Surmann, H. (2007). 6D SLAM-3D mapping outdoor environments. *Journal of Field Robotics*, 24(8-9), 699–722.
- Papadopoulos, G., Kurniawati, H., & Patrikalakis, N. (2013). Asymptotically optimal inspection planning using systems with differential constraints. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, May.
- Papadopoulos, G., Kurniawati, H., Shariff, A. S. B. M., Wong, L. J., & Patrikalakis, N. M. (2011). 3D-surface reconstruction for partially submerged marine structures using an Autonomous Surface Vehicle. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (pp. 3551–3557).
- Roman, C., & Singh, H. (2007). A self-consistent bathymetric mapping algorithm. *Journal of Field Robotics*, 24(1), 23–50.
- Roy, M., Foufou, S., & Truchetet, F. (2004). Mesh comparison using attribute deviation metric. *International Journal of Image and Graphics*, 4, 1–14.
- Scherer, S., Rehder, J., Achar, S., Cover, H., Chambers, A., Nuske, S., & Singh, S. (2012). River mapping from a flying robot: State estimation, river detection, and obstacle mapping. *Autonomous Robots*, 33(1-2), 189–214.
- Shkurti, F., Rekleitis, I., & Dudek, G. (2011). Feature tracking evaluation for pose estimation in underwater environments. In *Proceedings of the 2011 IEEE Canadian Conference on Computer and Robot Vision* (pp. 160–167).
- Sibley, G. (2006). Sliding window filters for SLAM. Technical report, University of Southern California, Center for Robotics and Embedded Systems.
- Sibley, G., Mei, C., Reid, I., & Newman, P. (2010). Vast-scale outdoor navigation using adaptive relative bundle adjustment. *International Journal of Robotics Research*, 29, 958–980.
- Singh, H., Whitcomb, L., Yoerger, D., & Pizarro, O. (2000). Microbathymetric mapping from underwater vehicles in the deep ocean. *Computer Vision and Image Understanding*, 79(1), 143–161.
- Smith, R., & Cheeseman, P. (1986). On the representation and estimation of spatial uncertainty. *International Journal of Robotics Research*, 5, 56–68.
- Smith, R., Self, M., & Cheeseman, P. (1990). Estimating uncertain spatial relationships in robotics. In *Autonomous robot vehicles* (pp. 167–193). Springer-Verlag, New York.
- Tena Ruiz, I., de Raucourt, S., Petillot, Y., & Lane, D. (2004). Concurrent mapping and localization using sidescan sonar. *Journal of Oceanic Engineering*, 29(2), 442–456.
- Thrun, S., Burgard, W., & Fox, D. (2000). A real-time algorithm for mobile robot mapping with applications to multi-robot and 3D mapping. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)* (pp. 321–328).
- Thrun, S., Burgard, W., & Fox, D. (2005). *Probabilistic Robotics*. Cambridge, MA, MIT Press.
- Thrun, S., Montemerlo, M., Koller, D., Wegbreit, B., Nieto, J., & Nebot, E. (2004). FastSLAM: An efficient solution to the simultaneous localization and mapping problem with unknown data association. *Journal of Machine Learning Research*, 4(3), 380–407.
- Tong, C. H., & Barfoot, T. D. (2012). Three-dimensional SLAM for mapping planetary work site environments. *Journal of Field Robotics* (pp. 381–412).
- Visual Computing Lab ISTI - CNR (2010). Meshlab. <http://meshlab.sourceforge.net/>.

Wurm, K. M., Hornung, A., Bennewitz, M., Stachniss, C., & Burgard, W. (2010). OctoMap: A probabilistic, flexible, and compact 3D map representation for robotic systems. In Proceedings of the ICRA 2010 Workshop on Best Practice in 3D Perception and Modeling for Mobile Manipulation, Anchorage, AK.

Zhao, H., & Shibasaki, R. (2001). Reconstructing textured CAD model of urban environment using vehicle-borne laser range scanners and line cameras. In Proceedings of the Second International Workshop on Computer Vision Systems (pp. 284–297), London, Springer-Verlag.