Mapping 3D Underwater Environments with Smoothed Submaps

Mark VanMiddlesworth[1], Michael Kaess[2], Franz Hover[3], and John J. Leonard[1]

Abstract This paper presents a technique for improved mapping of complex underwater environments. Autonomous underwater vehicles (AUVs) are becoming valuable tools for inspection of underwater infrastructure, and can create 3D maps of their environment using high-frequency profiling sonar. However, the quality of these maps is limited by the drift in the vehicle's navigation system. We have developed a technique for simultaneous localization and mapping (SLAM) by aligning point clouds gathered over a short time scale using the iterative closest point (ICP) algorithm. To improve alignment, we have developed a system for smoothing these "submaps" and removing outliers. We integrate the constraints from submap alignment into a 6-DOF pose graph, which is optimized to estimate the full vehicle trajectory over the duration of the inspection task. We present real-world results using the Bluefin Hovering AUV, as well as analysis of a synthetic data set.

1 Introduction

Inspection of underwater infrastructure is currently a costly, time-consuming, and dangerous task performed manually by human divers. As autonomous underwater vehicles become more sophisticated, it will be increasingly feasible and desirable to automate these inspection tasks. However, there are many technical challenges that must still be overcome, particularly in the domain of localization and navigation in complex 3D environments.

Underwater localization is particularly difficult because globally-referenced satellite navigation such as GPS is rapidly attenuated in water. Underwater vehicles must therefore either rely on inertial measurements, observations of local features,

[1]Computer Science and Artificial Intelligence Laboratory (CSAIL), Massachusetts Institute of Technology (MIT), Cambridge, MA 02139, USA, {mvanmidd, jleonard}@mit.edu ·
[2] Robotics Institute, Carnegie Mellon University, Pittsburgh, PA 15213, USA, kaess@cmu.edu ·
[3] Department of Mechanical Engineering, MIT , Cambridge, MA 02139, USA, hover@mit.edu

or acoustic communication with a globally-referenced transponder for localization. Current AUVs, such as those used for seafloor photographic and bathymetric surveys, generally use a combination of these techniques.

Inspection of harbors, platforms, and other underwater infrastructure poses unique navigational challenges beyond those of a seafloor survey. Range-based acoustic localization such as LBL is subject to multi-path interference, which is exacerbated by the shallow depths and hard, flat walls of harbor environments. Additionally, large metallic objects such as ship hulls render magnetic compasses largely ineffective, requiring that heading be estimated with drift-prone inertial sensors. Finally, while seafloor surveys can often be performed using inertial navigation and corrected in post-processing, the collision hazards posed by underwater infrastructure require accurate navigation in real time.

2 Related Work

This paper builds upon a large body of prior research in underwater simultaneous localization and mapping (SLAM), 3D mapping, and dense point cloud alignment. The goal of SLAM is to correct for drift in the vehicle's dead reckoning by using repeated observations of static landmarks in the environment. There are two broad families of approaches: *filtering* and *smoothing*. Both approaches generally assume Gaussian process and measurement error models.

Filtering approaches track the robot's current pose by incrementally adding dead reckoning and loop closure constraints. Because constraints are added incrementally, this approach is naturally suited to real-time operation. Barkby et al. [2] used a particle filter along with a bathymetric sonar to produce a 2.5D map of the seafloor in real time. The extended Kalman filter (EKF) has been applied to imaging sonar data [16], and forward-looking sonar [11] collected by AUVs. The extended information filter (EIF) [21], in which the normal distribution is parameterized in terms of its information vector and information matrix rather than its mean and covariance, has a sparse structure which enables efficient computation. Walter et al. [22] used a filtering approach to survey an underwater structure using features manually extracted from an imaging sonar.

A disadvantage of filtering approaches is that they estimate only the current vehicle pose. Because information from loop closure constraints is not back-propagated to correct previous pose estimates, these approaches do not provide an accurate estimate of the entire vehicle trajectory. This is particularly problematic when adding constraints from large loop closures, which produces discontinuities in the estimated vehicle path.

Smoothing approaches also include all past poses into the optimization. Exploiting the fact that the information matrix is exactly sparse in view-based SLAM, Eustice et al. [9] applied the information filtering approach to camera data from the RMS Titanic, producing a 6-DOF trajectory estimate. Dellaert and Kaess [7] formulate the SLAM problem as a bipartite factor graph, and provide an efficient solution

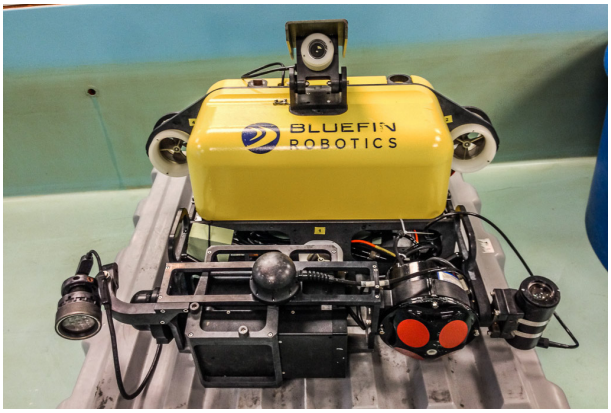


Fig. 1 Bluefin Hovering Autonomous Underwater Vehicle (HAUV) with Soundmetrics DIDSON sonar.

by smoothing and mapping (SAM). Incremental smoothing and mapping (iSAM) [13] incrementalizes the matrix factorization to efficiently integrate new constraints without re-factoring the information matrix.

In the underwater domain, pose graphs have been shown to produce more consistent maps due to their ability to correct prior navigation error and re-linearize around the corrected trajectory. Beall et al. [3] used an offline pose-graph based smoothing approach to estimate a full 6-DOF trajectory in a large-scale underwater photo survey. Kunz and Singh [14] applied offline pose graph optimization to visual and sonar data. Pose graphs have been used for real-time mapping of a locally planar complex structures such as ship hulls [12].

We chose to use a pose graph formulation for the improved handling of nonlinear error models and large loop closures, as we expect relatively large navigation drift in the absence of a magnetic compass. Additionally, the ability to reconstruct the full vehicle trajectory is particularly important for inspection tasks, to verify that the target has been fully covered by the vehicle's sensors.

In bathymetric and photomosaicing applications, a 2.5-dimensional representation of the environment (depth map) is sufficient, but complex environments require a full 3D representation. Fairfield et al. [10] use evidence grids inside a particle filter to perform real-time 3D mapping of a sinkhole with an imaging sonar.

Submap alignment requires generation of loop closure constraints, which, in visual SLAM, are commonly derived using viewpoint-invariant visual features such as SIFT. However, bathymetric and profiling sonars generally do not produce easily identifiable viewpoint-invariant features. If the vehicle dead reckoning is accurate over short time periods, as is the case with most IMU and DVL based systems, the sonar data can be aggregated into "submaps" for improved matching. The aggregated point cloud data is then treated as a single measurement. Point cloud-based approaches such as [15] and [5], which use iterative closest point (ICP) to align submaps, have been applied to scanning laser data. However, sonar data gener-

ally exhibits much higher noise than laser scanners, complicating registration. Prior work in the underwater domain has demonstrated success at aligning bathymetric (2.5D) submaps using cross-correlation [18] and ICP [17] to provide constraints for an EKF-based SLAM system; these are perhaps the most closely related to the work presented here. However, our work differs in a few key areas. Our contributions are as follows:

- A full 3D representation of arbitrarily complex marine environments
- Reconstruction of the entire vehicle trajectory using a pose graph
- A method of smoothing submaps for improved alignment

3 Problem Statement

Our complex area inspection missions begin with a long-range survey, which is used to construct a rough mesh of the inspection target. This mesh is used to plan an inspection path that covers the entire inspection target while avoiding collision. We use the sampling-based technique described in [8] for path generation.

Data for these experiments was collected on the Bluefin Hovering Autonomous Underwater Vehicle (HAUV), a vehicle specifically designed for ship hull inspection (see Fig. 1). Full details of the platform can be found in [12]; here we will briefly summarize the relevant attributes.

3.1 Navigation Sensors

The HAUV is equipped with a Honeywell HG1700 IMU, an RDI 1200kHz DVL, and a Keller pressure sensor. The DVL can be locked downward or rotated to point at the ship hull; in our complex-area operations, we keep the DVL locked downward. The DVL and IMU are combined to provide a position estimate. The x and y position is estimated by integrating velocities from the DVL and IMU, and are therefore subject to long-term drift. Because magnetic compasses are unreliable in the presence of steel structures, we do not use a magnetic compass; therefore, heading ψ is also subject to drift. Depth z, pitch θ, and roll ϕ are all measured directly, so they are subject to measurement noise, but not to drift.

3.2 Sonar

We use the DIDSON profiling sonar, which has an aperture 22° wide and 1° tall. There are 96 beams comprising the width of the sonar, each of which provides acoustic intensity in 512 bins representing range. To extract ranges from these intensities, we perform a median filter to reduce noise, then threshold the intensity and

accept the first (shortest-range) return along each beam. This somewhat reduces the effect of multipath interference, which often appears as an echo of close objects at a longer range. The range extraction at time t produces 2-D polar points $p_t^l = (\theta, r)$ in the sonar coordinate frame, which are transformed into Cartesian points $p_t = (x, y, z)$ in the vehicle coordinate frame.

We generally operate the sonar with a minimum range of 1m and maximum range of 6m for close inspection tasks, therefore each of the 512 range bins represents approximately 1cm. In practice, however, the range resolution is more coarse. Even in an ideal environment, measuring a hard, flat surface in a low-noise swimming pool, we observe that the return is generally "smeared" along several bins.

Harbor environments further degrade the quality of sonar data. Reflections from ship hulls, surface waves, and, surprisingly often, large fish lead to spurious returns even with aggressive filtering. From our experience, the error is generally on the order of 5cm, with occasional larger errors up to half a meter.

On the HAUV, the DIDSON is mounted to provide a horizontal "fan" of range returns, and can be swept 90°vertically. Due to the sonar's narrow field of view, we primarily use the fixed configuration for long-range surveys. Close range surveys consist of a series of waypoints, with the vehicle holding station while the sonar is swept vertically.

4 Pose Graph SLAM Using Submap Alignment

Fig. 3.2 shows the high-level architecture of our system. Our task is to estimate the vehicle pose $\mathbf{x}_t = [x, y, z, \psi, \theta, \phi]$ over the entire trajectory of a complex inspection task, $t = [1..n]$. As the vehicle moves along the inspection path, it aggregates sonar pings into submaps. The submaps are stored in the *submap catalog*, and are smoothed and aligned to provide loop closure constraints.

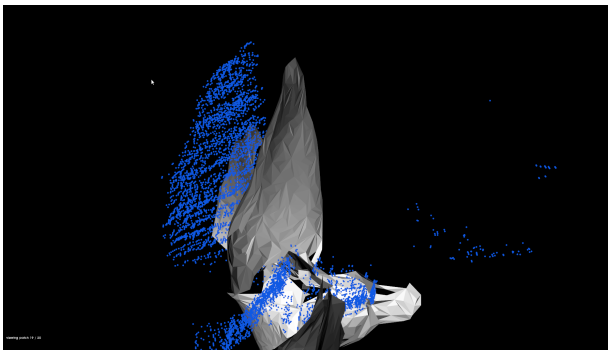


Fig. 2 An unfiltered submap projected alongside the prior mesh using dead reckoning, illustrating the approximate scale of dead reckoning error and some characteristics of unfiltered sonar data.

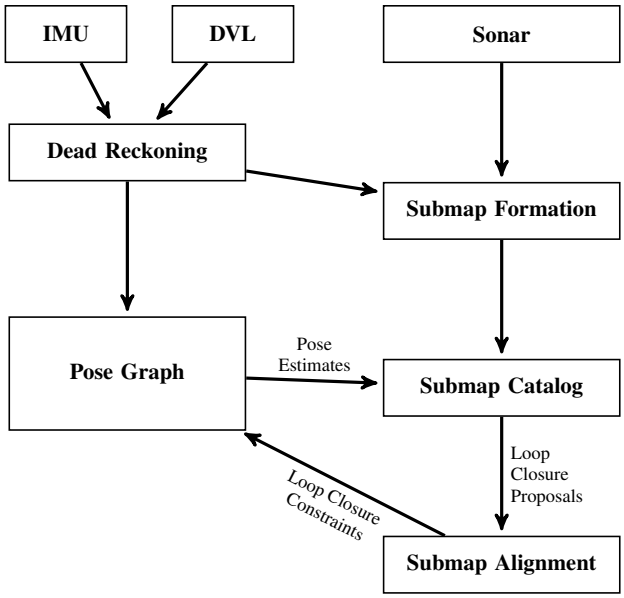


Fig. 3 Architecture of submap-based SLAM.

4.1 Submap Formation

To form submaps, we assume that dead reckoning is accurate over a short time scale, and use dead reckoning to aggregate groups of consecutive sonar pings into submaps. Each submap is assigned an *anchor time*, in this case halfway between the start and end time, and is treated as a single measurement taken at the anchor time.

Choosing submap size is a balance between the submaps being large enough to align with one another, but not so large as to incorporate significant navigation drift. With the DIDSON in the sweeping configuration, we simply define a submap as a single vertical sweep. When the DIDSON is locked in the horizontal configuration, we define a minimum number of points k_{sub} and a maximum time window for the submaps t_{sub}; when either threshold is reached, a submap is created. For a list of parameter values used in our experiments, see Table 1.

When the vehicle is actively scanning a target, the points threshold k_{sub} generally triggers submap formation well within the time window t_{sub}. When sonar returns are sparse (e.g. the vehicle is transiting between inspection waypoints), the time limit t_{sub} will trigger submap formation. In our experiments, these sparse submaps often produce poor alignments and are therefore not used for loop closures.

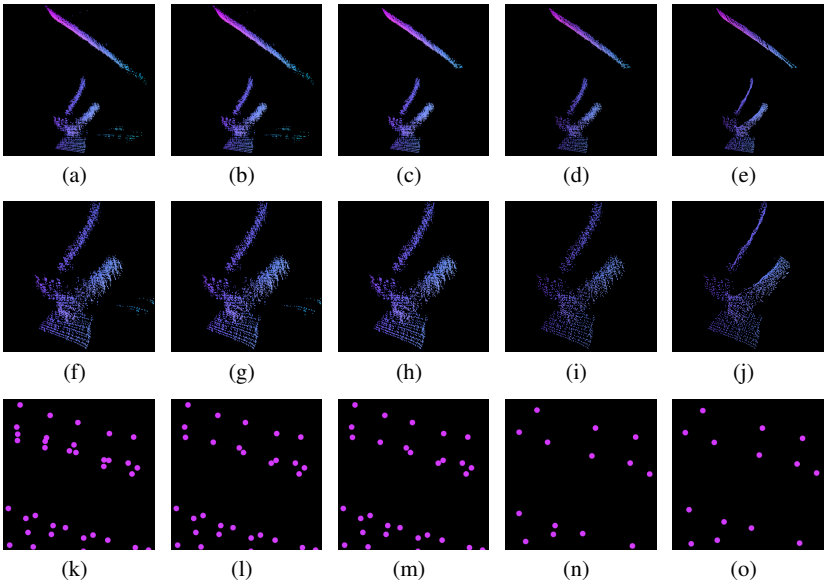


Fig. 4 A submap as it progresses through the four steps of the smoothing process. This submap consists of a vertical portion of the middle of the propeller and a segment of the hull, viewed from the starboard aft side. From left to right: the raw submap, submap after voxel filter 1, submap after outlier rejection, submap after voxel filter 2, and submap after parametric surface modeling. (a-d) wide view showing outliers. (f-j) detail shows smoothing of propeller blades. (k-o) illustrate the relative point density at each step.

4.2 Submap Smoothing

Before aligning the submaps, we perform a filtering and smoothing operation to reduce the sonar artifacts described above. This serves two purposes: to eliminate spurious returns caused by acoustic reflections, fish, etc., and to achieve a more uniform point density for better alignment. We rely primarily on the implementations found in the freely-available Point Cloud Library [19].

The first step is a voxel filter, in which the submap is divided into cubes of size v_1, and if multiple points occupy the same cube, they are removed and replaced with a single point at their centroid. We choose v_1 to be 2cm, which is approximately the spacing between DIDSON beams at medium range. Thus the first voxel filter serves to remove "redundant" data without significantly reducing the resolution of the submap.

Second, we perform k-nearest-neighbor outlier rejection, in which points are rejected if the average distance to their k_{nn} nearest neighbors is $> \sigma_{nn}$ standard deviations above the mean (for details and implementation, see [20]). This is effective at removing artifacts caused by electrical noise, minor multipath reflections, and small fish (we leave systematic multipath interference and large fish for future work).

The third step is a larger voxel filter of size v_2. The goal of this step is to produce a cloud of roughly uniform density, as the parametric surface modeling in step 4 is sensitive to variations in point density. Therefore, we choose v_2 to be roughly the distance covered between DIDSON frames (5-10 Hz) when the vehicle is moving at speed (.5-1 m/s), giving us $v_2 \approx 10$cm. This step has the effect of combining adjacent points from within DIDSON frames, so that their spacing more closely matches the spacing between frames.

Finally, the resulting points are smoothed using a local parametric approximation as described in [1]. For each point, a polynomial surface is constructed which minimizes the mean squared error to points within a radius r. The point normal is estimated as the normal to the parametric surface, and the point is projected onto the surface. The surface normal is also stored for each point, as it will be utilized in the alignment step. [1]

After smoothing, the smoothed submap and surface normals are stored in the submap catalog and used for alignment, while the full-resolution unprocessed submap is retained for reprojection into the final map. For an illustration of submaps at each step of the smoothing process, see Fig. 4.

4.3 Submap Alignment

We align submaps using Iterative Closest Point [4], an algorithm for aligning a set of *measurement points* (also called *source points*) to a *target model* by iteratively computing rigid transformations that minimize the sum of squared distances between the points and target model. Each measurement point is represented in Cartesian coordinates as $p = [x, y, z]$, and the target can be any model that allows computation of distances to a point, such as a parametric surface, line, or another point cloud. In our case, we have normal estimates for each submap, so we use point-to-plane distances in computing the transform.

For measurement points p'_i corresponding to target points p_i with normals n_i in the target, ICP computes the rigid transform T that minimizes the sum of square errors between the measurement set and the target set:

$$\sum_i ||(p'_i - T\,p_i) \cdot n_i||^2 \tag{1}$$

[1] While the previous steps were fairly well-grounded in sonar geometry and error characteristics, it is not immediately obvious why step 4, parametric surface modeling, is appropriate. In the general case, there is no reason to expect arbitrary input data to form a smooth manifold. However, in the underwater inspection domain, we find this technique justified for two reasons (beyond its empirical effectiveness). First, we make the general observation that the seafloor is, although not strictly polynomial, almost by definition a watertight manifold. Second, many of our inspection targets are anthropogenic structures which do, in fact, have a good polynomial approximation.

When a suitable alignment is found, we transform the anchor pose of the source submap, and formulate a relative constraint between the source and target anchor poses. This constraint is added as a factor in the pose graph.

ICP is known to be highly sensitive to initialization, due to the local minima which are often present in the cost function. If not initialized close to the correct solution, it will converge to the wrong local minimum. We address this issue in two ways. First, we initialize ICP with the most recent estimate of the relative position of the source and target poses. This is equivalent to assuming that the dead reckoning between the last correctly-aligned pose and the current pose is within the region of attraction of the correct alignment. Second, we assign a *fitness score* to each alignment that represents the normalized sum of squared distances between corresponding points. Smaller fitness scores represent better alignments. If the fitness score exceeds a threshold α, we reject the alignment and do not add a loop closure constraint to the pose graph. This threshold is dependent on the scope of the data set, point density, and noise levels; we found $\alpha = .1$ to be a reasonable value for our experiments.

For the purposes of this work, we assume data association is known; future work will address the issue of determining which pairs of submaps from the catalog to align.

4.4 Pose Graph Construction

We formulate the pose graph as a factor graph, in which nodes represent poses and factors represent constraints between poses, following the formulation in [13]. Specifically, a factor graph $G = (\mathscr{F}, \mathscr{Q}, \mathscr{E})$ is comprised of factor nodes $f_i \in \mathscr{F}$ and variable nodes $q_i \in \mathscr{Q}$. An edge $e_{ij} \in \mathscr{E}$ exists if factor node f_i depends on variable node q_j. Our goal is to find the maximizing variable assignment

$$\mathscr{Q}^* = \arg\max_{Q} \prod_i f_i(Q_i), \tag{2}$$

where Q_i is the set of variables adjacent to the factor f_i.

We construct a pose graph using constraints from dead reckoning and submap alignments. Dead reckoning constraints are periodically added, with a conservative covariance based on our sensor properties. When we get a match between two submaps, it is formulated as a relative constraint between the two corresponding anchor poses.

We estimate the covariance of a submap alignment constraint using a conservative heuristic based on the normalized covariance of the points in the source submap. While not exact, this provides an intuitively reasonable approximation. For example: alignment of the Y-Z plane would have higher certainty along the surface normal (X axis), and alignment of a uniform ball of points would be considered equally certain in all directions.

As constraints are added, they are incorporated incrementally, without costly variable re-ordering or redundant computation. The full batch optimization using Gauss-Newton, including variable reordering, is performed asynchronously in the background. This enables real-time operation, even for large pose graphs.

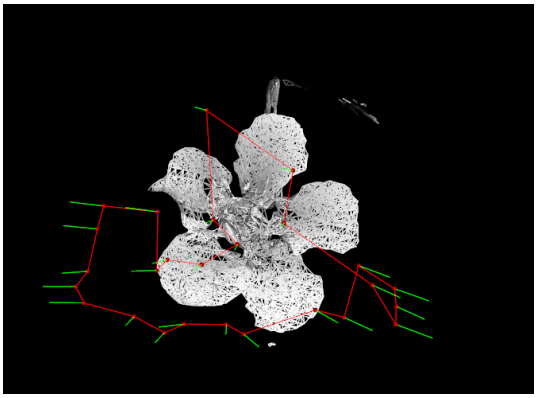


Fig. 5 Planned path for inspecting the propeller of the USS Saratoga.

5 Results

We tested our system in an underwater inspection scenario on the USS Saratoga, a decommissioned aircraft carrier in Newport, RI. We performed a survey of the running gear, including the approximately 7m diameter propeller and a section of the hull above the propeller. The survey consisted of three vertical track lines from 5-6m range with the DIDSON in fixed mode, followed by 24 close-range waypoints at which the DIDSON was swept vertically. The trajectory was generated by Englot and Hover's sampling-based coverage planning [8]; the planned path is shown in Fig. 5. The entire trajectory took approximately 1200 seconds to execute. For a summary of the parameters used for our submap smoothing and alignment, see Table 1.

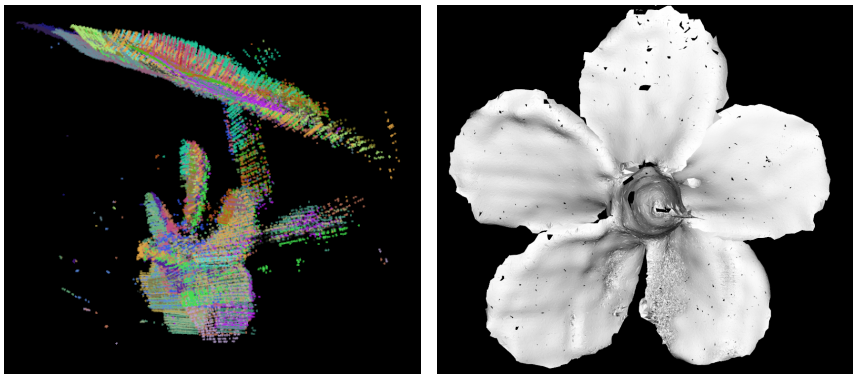
We do not have ground truth for this data set, but we estimate that the navigation drift was on the order of 30cm over the course of the survey. For an illustration of the navigation drift, see Fig. 2.

A mesh generated from our inspection trajectory is show in Fig. 6(b). The raw point cloud was denoised and smoothed using a variant of our submap smoothing algorithm described in Section 4.2. It was then meshed with a simple greedy triangulation. The raw sonar returns, reprojected from the optimized vehicle trajectory, can be seen in Fig. 6(a). Behind the propeller, the drive shaft is visible, along with a support strut. Note the high number of spurious sonar returns. Because the ship was

k_{sub}	500	Submap formation: min. points per submap
t_{sub}	25 sec	Submap formation: max. time per submap
v_1	2 cm	Voxel filter 1: size
k_{nn}	50	Outlier rejection: # neighbors considered
σ_{nn}	2	Outlier rejection: std. dev. limit
v_2	10 cm	Voxel filter 2: size
r	0.3 m	Polynomial surface modeling: radius
α	.1	Submap alignment: fitness threshold

Table 1 Key parameters for submap formation, smoothing, and alignment.

not operational, it had a significant amount of growth on the propeller and shaft, and had become a habitat for fish, oysters, and other marine life.



(a) (b)

Fig. 6 (a): The final point cloud from the propeller inspection trajectory. Raw submaps have been reprojected according to the SLAM-corrected trajectory, and points are colored by submap. (b): A smoothed mesh generated from the raw point cloud.

To isolate the effect of pose graph alignment, we also generated a partially synthetic data set based on the actual vehicle trajectory. We corrupted the vehicle trajectory with navigation drift, simulated by incrementally adding zero-mean Gaussian noise to dead reckoning measurements of x, y, and ψ state variables. We used a standard deviation of .01m for x and y, and .00001 radian for ψ, accumulating at 20Hz. We also added zero-mean Gaussian noise to the absolute measurements of z, θ, and ϕ to simulate increased measurement noise without drift. We also added synthetic sonar measurements created from the actual vehicle trajectory and a high-resolution sonar map of the propeller gathered in a previous experiment. To simulate sonar pings at each waypoint, we extracted points in the sonar field of view, downsampled them to the sonar resolution, and added Gaussian noise with a standard deviation of 5cm.

Fig. 7 shows the reprojected clouds from our synthetic data set, using the dead reckoning (left) and SLAM (right) trajectories. As is apparent, the dead reckon-

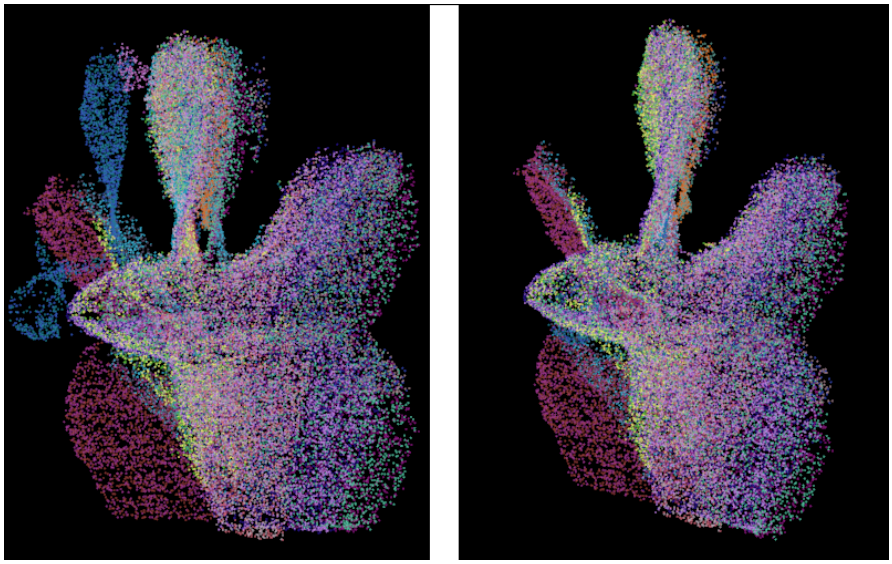


Fig. 7 Comparison of point clouds generated from dead reckoning (left) vs. SLAM (right).

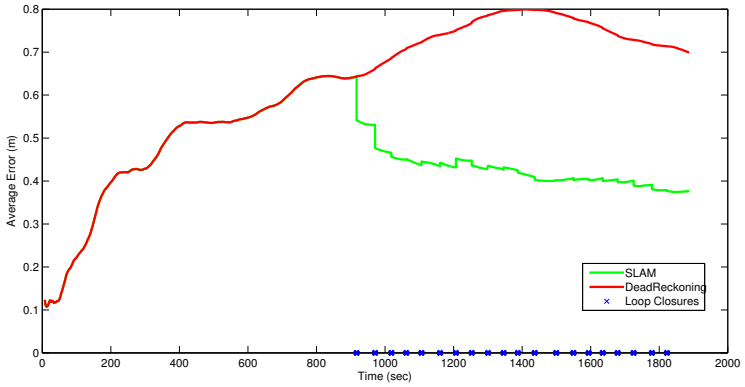


Fig. 8 Average per-pose error over time.

ing error caused misalignment in the reprojection, which is corrected in our SLAM framework. Fig. 8 shows the average per-pose trajectory error over time. At each loop closure event, the SLAM error is reduced, while the dead reckoning error continues to accumulate. Note that, in the SLAM reprojection, the sonar noise dominates the alignment error, even though the trajectory is estimated based on the noisy sonar returns. We attribute this to our smoothing procedure detailed in Section 4.2.

6 Conclusion

We have presented a system for submap-based loop closure in 3D underwater mapping, and demonstrated its use on real and synthetic data sets. By smoothing the submaps before alignment, we have reduced the effect of the spurious returns frequently found in cluttered and heavily biofouled environments. By regularizing point cloud density, our system is able to combine submaps of varying resolution, while retaining the full-resolution point clouds for reprojection into the final map. Using a pose graph framework, we are able to reconstruct the entire vehicle trajectory, which is necessary to ensure full coverage of the inspection target. Our system runs in real time, and it is robust to moderate amounts of sonar noise and navigation drift.

7 Future Work

This effort suggests many areas for future research. We have a simple threshold-based system for outlier rejection, but a fully probabilistic solution could potentially improve our results. We would also like to experiment with different alignment techniques, such as multi-scale ICP or the normal distribution transform.

We could potentially further reduce error by using a volumetric, rather than point-based, reconstruction. For example, the signed distance function [6] combines multiple measurements into a single implicit surface model. This has been used to great effect with RGBD cameras, and we suspect it may help reduce the sonar noise still present in our final models. A drawback of naive volumetric techniques is that misalignments are incorporated into the final model and affect all subsequent alignments. Perhaps a hybrid system, which used our pose graph technique for initial trajectory estimates and a volumetric reconstruction for further refinement, could combine the benefits of volumetric techniques with the flexibility of pose graph SLAM.

Acknowledgment

This work was partially supported by ONR grants N00014-12-1-0093, N00014-12-10020, N00014-10-1-0936, N00014-13-1-0588, and N00014-11-1-0688; NSF Award IIS-1318392; and the NSF Graduate Research Fellowship Program. The authors would also like to thank Josh Leighton and Pedro Vaz Teixeira at MIT, Ryan Eustice, Ayoung Kim and Paul Ozog at the University of Michigan, and Jerome Vaganay at Bluefin Robotics.

References

1. Alexa, M., Behr, J., Cohen-Or, D., Fleishman, S., Levin, D., Silva, C.T.: Computing and rendering point set surfaces. IEEE Transactions on Visualization and Computer Graphics **9**(1), 3 – 15 (2003)
2. Barkby, S., Williams, S., Pizarro, O., Jakuba, M.: An efficient approach to bathymetric SLAM. In: IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS) (2009)
3. Beall, C., Dellaert, F., Mahon, I., Williams, S.: Bundle adjustment in large-scale 3D reconstructions based on underwater robotic surveys. In: Proc. of the IEEE/MTS OCEANS Conf. and Exhibition (2011)
4. Besl, P.J., McKay, N.D.: A method for registration of 3-D shapes. IEEE Trans. Pattern Anal. Machine Intell. **14**(2), 239–256 (1996)
5. Borrmann, D., Elseberg, J., Lingemann, K., Nuchter, A., Hertzberg, J.: Globally consistent 3D mapping with scan matching. J. of Robotics and Autonomous Systems **56**(2), 130–142 (2008)
6. Curless, B., Levoy, M.: A volumetric method for building complex models from range images. In: SIGGRAPH, pp. 303–312 (1996)
7. Dellaert, F., Kaess, M.: Square Root SAM: Simultaneous localization and mapping via square root information smoothing. Intl. J. of Robotics Research **25**(12), 1181–1203 (2006)
8. Englot, B., Hover, F.: Sampling-based coverage path planning for inspection of complex structures. In: Proc. of Intl. Conf. on Automated Planning and Scheduling (2012)
9. Eustice, R., Singh, H., Leonard, J., Walter, M., Ballard, R.: Visually navigating the RMS Titanic with SLAM information filters. In: Robotics: Science and Systems (RSS) (2005)
10. Fairfield, N., Kantor, A.G., Wettergreen, D.: Real-time SLAM with octree evidence grids for exploration in underwater tunnels. J. of Field Robotics (2007)
11. Folkesson, J., Leonard, J.: Autonomy through SLAM for an underwater robot. In: Proc. of the Intl. Symp. of Robotics Research (ISRR) (2009)
12. Hover, F., Eustice, R., Kim, A., Englot, B., Johannsson, H., Kaess, M., Leonard, J.: Advanced perception, navigation and planning for autonomous in-water ship hull inspection. Intl. J. of Robotics Research **31**(12), 1445–1464 (2012)
13. Kaess, M., Ranganathan, A., Dellaert, F.: iSAM: Incremental smoothing and mapping. IEEE Trans. Robotics **24**(6), 1365–1378 (2008)
14. Kunz, C., Singh, H.: Map building fusing acoustic and visual information using autonomous underwater vehicles. J. of Field Robotics **30**(5), 763–783 (2013)
15. Nüchter, A., Hertzberg, J.: Towards semantic maps for mobile robots. J. of Robotics and Autonomous Systems **56**(11), 915–926 (2008). DOI http://dx.doi.org/10.1016/j.robot.2008.08.001
16. Ribas, D., Ridao, P., Tardós, J., Neira, J.: Underwater SLAM in man-made structured environments. Journal of Field Robotics **25**(11-12), 898–921 (2008)
17. Roman, C., Singh, H.: Improved vehicle based multibeam bathymetry using sub-maps and SLAM. In: IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS), pp. 3662–3669 (2005)
18. Roman, C., Singh, H.: A self-consistent bathymetric mapping algorithm. J. of Field Robotics **24**(1), 23–50 (2007)
19. Rusu, R., Cousins, S.: 3D is here: Point Cloud Library (PCL). In: IEEE Intl. Conf. on Robotics and Automation (ICRA). Shanghai, China (2011)
20. Rusu, R.B., Marton, Z.C., Blodow, N., Dolha, M., Beetz, M.: Towards 3D point cloud based object maps for household environments. J. of Robotics and Autonomous Systems **56**(11), 927–941 (2008)
21. Thrun, S., Liu, Y., Koller, D., Ng, A., Ghahramani, Z., Durrant-Whyte, H.: Simultaneous localization and mapping with sparse extended information filters. Intl. J. of Robotics Research **23**(7) (2004)
22. Walter, M., Hover, F., Leonard, J.: SLAM for ship hull inspection using exactly sparse extended information filters. In: IEEE Intl. Conf. on Robotics and Automation (ICRA), pp. 1463–1470 (2008)