2013 IEEE/RSJ International Conference on
Intelligent Robots and Systems (IROS)
November 3-7, 2013. Tokyo, Japan

Real-time SLAM with Piecewise-planar Surface Models and Sparse 3D Point Clouds

Paul Ozog and Ryan M. Eustice

Abstract— **This paper reports on the use of planar patches as features in a real-time simultaneous localization and mapping (SLAM) system to model smooth surfaces as piecewise-planar. This approach works well for using observed point clouds to correct odometry error, even when the point cloud is sparse. Such sparse point clouds are easily derived by Doppler velocity log sensors for underwater navigation. Each planar patch contained in this point cloud can be constrained in a factor-graph-based approach to SLAM so that neighboring patches are sufficiently coplanar so as to constrain the robot trajectory, but not so much so that the curvature of the surface is lost in the representation. To validate our approach, we simulated a virtual 6-degree of freedom robot performing a spiral-like survey of a sphere, and provide real-world experimental results for an autonomous underwater vehicle used for automated ship hull inspection. We demonstrate that using the sparse 3D point cloud greatly improves the self-consistency of the map. Furthermore, the use of our piecewise-planar framework provides an additional constraint to multi-session underwater SLAM, improving performance over monocular camera measurements alone.**

I. Introduction

The classic approach to the simultaneous localization and mapping (SLAM) problem uses landmark measurements, either bearing-only or coupled with range, to estimate both the locations of environment features and the trajectory of a mobile robot. These features are often included with the vehicle pose in a state vector that is estimated using an extended Kalman filter (EKF) [1], particle filter [2], or nonlinear least-squares solver [3]–[5].

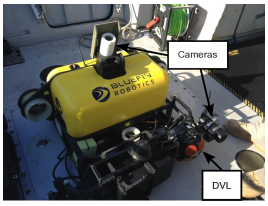
As robot sensors become more data-rich, like high-resolution light detection and ranging (LIDAR) scanners or depth cameras, including every point observed by these sensors into the state vector can quickly become burdensome. Instead, researchers have used feature-cloud matching so that landmark detections at two robot poses are used to compute a single relative transformation between the two poses. For 2D or 3D scanners, algorithms like iterative closest point (ICP) [6] or correlative scan matching [7] have been shown to work well for many widespread applications. Feature-cloud matching also preserves the sparsity of the information matrix in graph-based visual SLAM [8], which has inspired a number of successful works for camera-equipped autonomous underwater robots [9]–[14].

This work was supported by the Office of Naval Research under award N00014-12-1-0092.

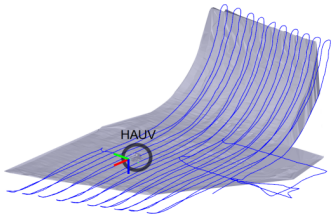
P. Ozog is with the Department of Electrical Engineering & Computer Science, University of Michigan, Ann Arbor, MI 48109, USA paulozog@umich.edu

R. Eustice is with the Department of Naval Architecture & Marine Engineering, University of Michigan, Ann Arbor, MI 48109, USA eustice@umich.edu

(a) SS Curtiss



(b) HAUV



(c) Piecewise-planar SLAM

Fig. 1. Treating large ship hulls as a collection of piecewise-planar surface features augments our visual SLAM system to produce more self-consistent maps, even with a very sparse 3D DVL point cloud (twenty 3D points per second). In (a), a 160 m vessel, the *SS Curtiss*, is surveyed by the HAUV [12] shown in (b), which is equipped with an actuated DVL sensor. A reconstructed underwater view of the HAUV's trajectory (solid line) and the ship hull (gray surface) is shown in (c).

The Doppler velocity log (DVL) sensor is common for precise navigation in underwater vehicles, and it can provide an accurate, though very sparse, 3D point cloud. For underwater robots, active range scanners like the DVL are simply not as data-rich as other sensors, like multibeam sonar or LIDAR scanners. For comparison, a RD Instruments Workhorse DVL [15] senses about twenty 3D points per second, whereas a high-end Velodyne HDL-64E scanner for ground vehicles [16] returns over a million points per second. Due to the sparsity of these point clouds, the DVL has simply not been able to take advantage of the aforementioned feature-cloud matching that is so widespread in modern mobile robotics. In this work, we leverage the sparse DVL point cloud by fitting local planar patches to a moving time window of points, and show how this can significantly constrain the trajectory of a Hovering Autonomous Underwater Vehicle (HAUV) performing automated ship hull inspection, as shown in Fig. 1.

A. Related Work

Despite the inability of underwater robots to perform feature-cloud-based SLAM using a DVL, there are still a wide variety of viable techniques for underwater SLAM. Barkby et al. [17] perform SLAM on the seabed using a

particle filter that weights particles based on sonar beam measurement agreement with previous observations. By fitting a Gaussian process to the past sonar beam observations, they are able to detect loop closures even with little overlap between the current sonar swath and past swaths.

Cameras have also proven to be a viable sensor for performing SLAM in underwater environments. In [13], Kim and Eustice use a monocular camera on a HAUV to efficiently bound odometry error by only considering visually salient keyframes. This works to great effect for good underwater visibility conditions and sufficiently feature-rich ship hulls, however these conditions are not always available. In that work, the DVL was used only for measuring velocity-derived odometry—the resulting 3D point cloud was not used. This work will show that incorporating this information into a SLAM backend is very useful for correcting odometry error, particularly when good imagery is not available.

Our algorithm relies on the use of planes as features for performing SLAM, rather than the standard Cartesian point-features or feature-clouds. Planar features have been used in real-time SLAM systems for a number of years [18]–[20]. In [18], Weingarten and Siegwart use a vehicle equipped with multiple SICK LIDAR scanners to detect planar feature patches in an office building using an EKF framework. They use a rotating SICK scanner to acquire a dense 3D scan, from which planar patches are detected and fused. These fused planes are used as a single feature measurement, keeping the number of features small in the resulting maps. Trevor et al. [19] implemented planar features as factors in a factor-graph-based SLAM framework. Effectively, these efforts have re-introduced feature-based maps for LIDAR-equipped robots rather than using feature-cloud scan-matching for laser odometry. In both of these works, a robot traverses an office building-like environment, and observes planar walls and tables with high-resolution 3D and 2D scanners. In unstructured environments using low-resolution scanners, it is unclear if these approaches would work.

In [21], Ruhnke et al. propose a system that avoids treating feature-clouds as rigid bodies and instead performs offline optimization of both the pose and environment surfaces in a joint optimization framework. Besides a laser range scanner, their sensor suite includes a RGB-D camera and they are able to construct accurate, high-resolution models. Similar to [21], we use planar patches to approximate a curved surface in a piecewise-planar fashion, however our approach is designed for real-time applications on much sparser 3D point clouds.

B. Outline

In §II, we cover the theory behind the use of planar features in our SLAM system. We frame the problem using factor-graphs, which can be integrated into the incremental smoothing and mapping (iSAM) framework [4], and we detail the weight matrices used in the sparse optimization. In §III, we present a simulation framework in which a robot in 3D surveys a spherical object, showing that our approach is capable of successfully modeling surfaces that are curved everywhere. In §IV, we apply this framework to a real-world autonomous ship hull inspection experiment, and present an improvement in the map consistency. Moreover, we provide initial results in a multi-session SLAM system in which the reliability is greatly improved by the addition of anchor node support [22] in our planar feature framework. Finally, in §V, we offer some concluding remarks and address areas for improvement and future work.

II. Approach

A. Parameterization of Planes in 6-DOF SLAM

Let $\mathbf{x}_i = [x_i, y_i, z_i, \phi_i, \theta_i, \psi_i]^\top$ be the 6-degree of freedom (DOF) relative-pose of the vehicle in the global frame at time i, where x, y, z are the Cartesian translation components, and ϕ_i, θ_i, and ψ_i denote the roll (x-axis), pitch (y-axis), and yaw (z-axis) Euler angles, respectively. Let $\boldsymbol{\pi}_k^i = [a_k^i, e_k^i, d_k^i]^\top$ be the plane indexed by k, expressed in the frame of $\mathbf{x}_i$. This plane is represented by the azimuth and elevation of the direction of the surface normal, and the orthogonal distance, or standoff, of the vehicle to the plane (i.e., a_k^i, e_k^i, and d_k^i, respectively). For convenience, let $\mathbf{n}_k^i$ be the Cartesian-coordinate unit-norm vector corresponding to the azimuth and elevation of the surface normal of $\boldsymbol{\pi}_k^i$. This normal vector is computed as

$$\mathbf{n}_k^i = \begin{bmatrix} n_{k_x}^i \\ n_{k_y}^i \\ n_{k_z}^i \end{bmatrix} = h\left(\begin{bmatrix} a_k^i \\ e_k^i \end{bmatrix}\right) = \begin{bmatrix} \cos(e_k^i)\cos(a_k^i) \\ \cos(e_k^i)\sin(a_k^i) \\ \sin(e_k^i) \end{bmatrix}. \quad (1)$$

The inverse of (1) is given by

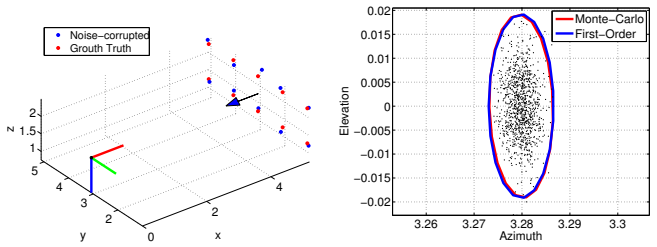
$$\begin{bmatrix} a_k^i \\ e_k^i \end{bmatrix} = h^{-1}\left(\begin{bmatrix} n_{k_x}^i \\ n_{k_y}^i \\ n_{k_z}^i \end{bmatrix}\right) = \begin{bmatrix} \operatorname{atan2}(n_{k_y}^i, n_{k_x}^i) \\ \operatorname{atan2}\left(n_{k_z}^i, \sqrt{n_{k_x}^{i^2} + n_{k_y}^{i^2}}\right) \end{bmatrix}. \quad (2)$$

Parameterizing a plane using a 3-DOF vector, $\boldsymbol{\pi}_k^i$, avoids problems associated with over-parameterized representations of variable nodes in iSAM. The plane, $\boldsymbol{\pi}_k^i$, is determined as the least-squares fit of a sliding-window of N stacked 3D points in pose i's frame, $\mathbf{p}_k^i \in \mathbb{R}^{3N}$. By reshaping $\mathbf{p}_k^i$ into a $3\times N$ data matrix and applying principal component analysis (PCA), the unit-norm can be determined, up to sign, as the left singular vector corresponding to the smallest singular value [23]. The standoff, d_k^i, is computed by dotting the unit normal with the centroid of $\mathbf{p}_k^i$. Finally, the azimuth and elevation are computed directly from (2). The process is illustrated in Fig. 2(a). We model the plane's uncertainty as a first-order approximation, described in §II-D.

B. Constraining Planes and Poses with Factors

The widely-used factor-graph representation of the SLAM problem uses variable nodes and factors in a bipartite graph representation [3]. We introduce a planar node, parameterized by the three values described in §II-A: azimuth, elevation and distance.

To initialize a planar node, we fit a least-squares plane as described in §II-A. This gives us a measurement, $\mathbf{z}_{\boldsymbol{\pi}_k^i}$, of the plane, indexed by k, in the frame of a pose indexed by i.



(a) Least-squares plane fit (b) First-order covariance

Fig. 2. Simple example of simulated least-squares fit of 3D points to a plane. In (a), we fit a plane to the sparse DVL point cloud by taking the principal component that captures the least variation in the data. This least-squares fit is shown by the blue arrow. In (b), the propagation of the point cloud covariance, $\Sigma_{\mathbf{p}}$, is well-approximated by numerically linearizing the least-squares fit described in §II-A, so long as the point cloud uncertainty is small.

Then, a unary factor is added to the initialized plane node with the following potential:

$$\Psi(\boldsymbol{\pi}_k^i; \mathbf{z}_{\boldsymbol{\pi}_k^i}, \Sigma_{\boldsymbol{\pi}_k^i}) = \|\mathbf{z}_{\boldsymbol{\pi}_k^i} - \boldsymbol{\pi}_k^i\|^2_{\Sigma_{\boldsymbol{\pi}_k^i}}, \tag{3}$$

where $\Sigma_{\boldsymbol{\pi}_k^i}$ is the measurement covariance matrix, to be discussed in §II-D.1.

Once two plane nodes are observed either from the same pose or two different poses and deemed sufficiently coplanar, a quaternary factor is added that constrains the two planes and the two poses from which they were observed. The potential for this factor, computed as in [19] by the difference between the predicted plane k expressed in pose i's frame and the plane l expressed in pose j's frame, is given by:

$$\Omega(\mathbf{x}_i, \boldsymbol{\pi}_k^i, \mathbf{x}_j, \boldsymbol{\pi}_l^j; \Sigma_{\boldsymbol{\pi}_{kl}}) = \|\widehat{\boldsymbol{\pi}}_k^j - \boldsymbol{\pi}_l^j\|^2_{\Sigma_{\boldsymbol{\pi}_{kl}}}, \tag{4}$$

where $\Sigma_{\boldsymbol{\pi}_{kl}}$ is a weight matrix, to be discussed in §II-D.2. The plane prediction, $\widehat{\boldsymbol{\pi}}_k^j$, is computed as follows:

$$\begin{aligned} \widehat{\mathbf{n}}_k^j &= \mathrm{R}_i^j \mathbf{n}_k^i \\ \widehat{d}_k^j &= \mathbf{t}_{ij}^{i\top} \mathbf{n}_k^i + d_k^i, \end{aligned}$$

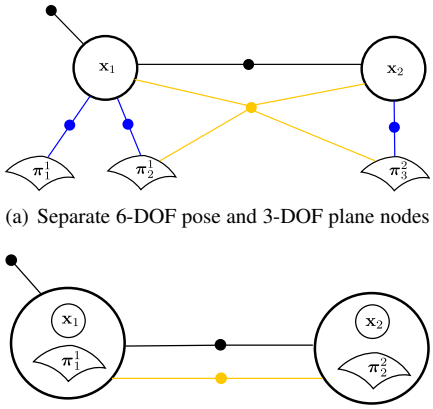
where R_i^j is the rotation matrix that rotates vectors in i's frame to vectors in j's frame, and $\mathbf{t}_{ij}^i$ is the translation from i to j expressed in i's frame. It is important to note that if this factor is left out of the graph, planar nodes have no impact on the optimization of the robot poses i and j.

A simple example of the graph layout for our approach is shown in Fig. 3(a). A detailed description of the weight matrices used in (3) and (4) is provided in §II-D.

C. 9-DOF Vehicle State Nodes

For the simulation and experimental results in §III and §IV, the planes are fit from a very sparse 3D scanner so we assume for convenience that there is only one observed planar patch per robot pose. We therefore combine the vehicle pose and planar patch into a single node in our factor-graph using the 9-DOF vehicle state nodes of the form

$$\mathbf{x}_i' = \begin{bmatrix} \mathbf{x}_i \\ \boldsymbol{\pi}_i^i \end{bmatrix}.$$



(a) Separate 6-DOF pose and 3-DOF plane nodes

(b) Combined 9-DOF state pose and plane nodes

Fig. 3. Small example of factor-graphs using our piecewise-planar approach (best viewed in color), with black representing unary prior factors and binary odometry factors. In (a), planes $\boldsymbol{\pi}_1^1$, $\boldsymbol{\pi}_2^1$, and $\boldsymbol{\pi}_3^2$ are represented as their own nodes in the factor-graph, along with pose nodes $\mathbf{x}_1$ and $\mathbf{x}_2$. In this example, pose 1 observes planes 1 and 2, while pose 2 observes plane 3 (blue factors). The addition of the quaternary yellow factor constrains the trajectory so that planes 2 and 3 are sufficiently coplanar. In (b), we conveniently assume that each pose observes one plane, so we lump the 6-DOF pose node together with the 3-DOF plane node into a 9-DOF vehicle state node. In this case, we can use binary factors (yellow) to constrain the robot trajectory.

The planar components of this vehicle state node are initialized in the same manner as (3). Then, the 9-DOF co-planar factor potential, Ω', is computed by evaluating (4) with $k = i$ and $l = j$:

$$\Omega'(\mathbf{x}_i', \mathbf{x}_j'; \Sigma_{\boldsymbol{\pi}_{ij}}) = \Omega(\mathbf{x}_i, \boldsymbol{\pi}_i^i, \mathbf{x}_j, \boldsymbol{\pi}_j^j; \Sigma_{\boldsymbol{\pi}_{ij}}) \tag{5}$$

D. Weight matrices

1) Least-Squares Plane Fit: The weight matrix $\Sigma_{\boldsymbol{\pi}_k^i}$ used in (3) can be interpreted as the covariance of a least-squares fitted plane. To compute this covariance matrix, we assume the sliding window of 3D points for the k^{th} plane as expressed in pose i's frame is corrupted by Gaussian noise. The propagation of this uncertainty is modeled by linearizing the function $f(\cdot)$ that fits $\boldsymbol{\pi}_k^i$ from $\mathbf{p}_k^i$:

$$\begin{aligned} \boldsymbol{\pi}_k^i &= f(\mathbf{p}_k^i) \\ \text{where } \mathbf{p}_k^i &\sim \mathcal{N}(\boldsymbol{\mu}_{\mathbf{p}_k^i}, \Sigma_{\mathbf{p}_k^i}). \end{aligned}$$

This function is the PCA-based method described in §II-A, which is numerically differentiable when care is taken to guarantee that the unit normal consistently faces toward the robot, since typical singular value decomposition (SVD) libraries may return singular vectors with flipped signs when the input matrix is slightly perturbed. Then, upon initialization, the plane's covariance is given by

$$\Sigma_{\boldsymbol{\pi}_k^i} \approx \mathrm{F} \Sigma_{\mathbf{p}_k^i} \mathrm{F}^\top, \tag{6}$$

where F is the Jacobian of $f(\cdot)$ evaluated at the observed value of $\mathbf{p}_k^i$. If $\Sigma_{\mathbf{p}_k^i}$ is small, this first-order approximation performs well as shown in Fig. 2(b). For our experimental underwater robot, the uncertainty of this point cloud was estimated to be $0.02\ \mathrm{m}$ by computing the sample variance when the sensor is pointed towards a known surface (for

instance, the very bottom of a ship hull where the depth error is small).

2) Co-planar Factor Potential : The norm $\|\widehat{\boldsymbol{\pi}}_k^i - \boldsymbol{\pi}_l^j\|^2$ based on the potential from (4) will be small if the two planes k and l are perfectly coplanar. Of course, these planes may not be perfectly coplanar for two reasons: (*i*) propagation of sensor noise discussed in the previous section, and (*ii*) the surface from which the two planes are measured is not exactly planar. Both of these, if not accounted for, will cause the backend optimizer to over-compensate for the factor's potential. For the remainder of this section, we use 9-DOF representation of poses and planes to keep the notation simpler; i.e., $k = i$ and $l = j$.

To account for nonplanar surfaces, we define an additional weight matrix to account for the surface curvature. This weight matrix is modeled by user-provided characteristic radii for both the azimuth and elevation dimensions. We compute the expected difference in azimuth, elevation, and distance given the circular curvature model (which constitutes a diagonal set of weights), and a forward-propagation of the covariance of the 9-DOF nodes i and j as follows:

$$\mathrm{W}_{\boldsymbol{\pi}_{ij}} = \mathrm{diag}(\Delta a^2, \Delta e^2, \Delta d^2) + \mathrm{C} \begin{bmatrix} \Sigma_{ii} & \Sigma_{ij} \\ \Sigma_{ji} & \Sigma_{jj} \end{bmatrix} \mathrm{C}^\top, \tag{7}$$

where C is the Jacobian of the function $c(\,\cdot\,)$ that is a simple curvature model of the surface being observed. It predicts the difference of the normal direction and standoff between two poses:

$$\begin{bmatrix} \Delta a \\ \Delta e \\ \Delta d \end{bmatrix} = c(\mathbf{x}_i', \mathbf{x}_j'; r_a, r_e),$$

$$\text{where } \begin{bmatrix} \Delta a \\ \Delta e \end{bmatrix} = \begin{bmatrix} a_i^i \\ e_i^i \end{bmatrix} - h^{-1}\left(\mathrm{R}_i^j h\left(\begin{bmatrix} a_i^i + t_{ij_y}^i / r_a \\ e_i^i + t_{ij_z}^i / r_e \end{bmatrix} \right) \right),$$

$$\Delta d = \left(h\left(\begin{bmatrix} a_i^i + t_{ij_y}^i / r_a \\ e_i^i + t_{ij_z}^i / r_e \end{bmatrix} \right) - h\left(\begin{bmatrix} a_i^i \\ e_i^i \end{bmatrix} \right) \right)^\top \mathbf{t}_{ij}^i.$$

Here, r_a and r_e are the user-defined characteristic radii in the azimuth and elevation axes, respectfully, $\mathbf{t}_{ij}^i$ is the translation vector and $h(\,\cdot\,)$ is defined in (1). Fig. 4 illustrates this simple curvature model in 2D.

The matrix used in the factor potential from (4) is the sum of the covariance of the j^{th} measured plane from (6) and the weight matrix from (7):

$$\Sigma_{\boldsymbol{\pi}_{ij}} = \Sigma_{\boldsymbol{\pi}_j^j} + \mathrm{W}_{\boldsymbol{\pi}_{ij}}.$$

Importantly, this matrix should be interpreted as a weight matrix, and not as a covariance matrix. The aim of this weight matrix is to avoid over-penalization of errors from (4) that are due to surface curvature, taking into account the uncertainty of the pose and planar measurements. This determination of weights clearly depends on the state of the vehicle and the uncertainty of the relevant robot poses *at the time the factor is added to the graph*. Thus, the weight matrix tends to provide conservative posterior covariances. Furthermore, the nonlinear least-squares problem we are providing to iSAM is clearly not the maximum likelihood estimate (MLE) of the robot trajectory and surface, because the measurement "covariance" matrix depends on the state, as formulated. However, so long as the trajectory does not drastically change during the optimization, we achieve good results. In particular, we have found that assuming depth, pitch, and roll uncertainty as bounded improves this approach, even with large translational and yaw uncertainty. This will be shown in our simulation environment in §III.

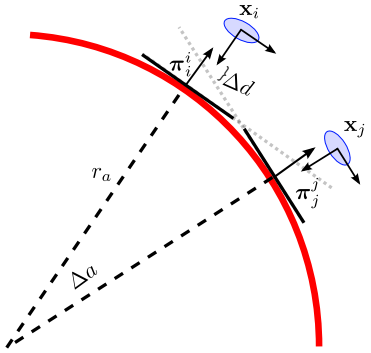


Fig. 4. An overview of how the weight matrix used in (4) is computed between two poses and two planes. Treating the translation from i to j as an approximate arc-length, the change in azimuth, Δa, between the two poses can be computed given the characteristic radius, r_a. The uncertainty of the poses, shown by the blue ellipses, also plays a role into the computation of Δa, so this is also included in the weight matrix.

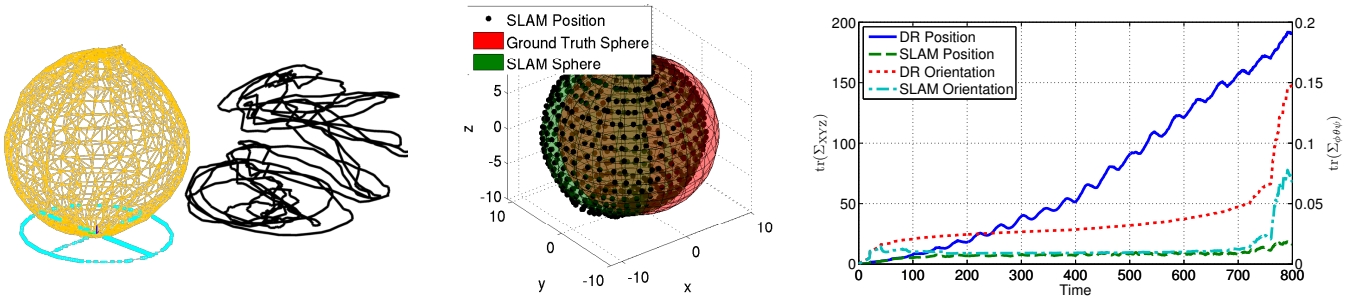
E. Data Association

Unlike the work from [18] and [19], we focus on experiments where the differences in plane positions and surface normals vary gradually on a smooth surface, rather than surface normals which belong to, say, walls in a building. To solve the data association problem, [18] checks the Mahalanobis distance of the candidate plane observed from the current pose to a plane that is already included in the EKF state vector. If it is below a threshold, the current planar measurement is associated with the filtered planar feature.

Finding the best association of planar patches on a smooth surface is significantly more challenging because (*i*) there is a large number of possible node pairs from which to add coplanar factors, and (*ii*) most neighboring planar features are all consistent in terms of Mahalanobis distance. Even so, we use the same basic method as [18] when it comes to adding a planar factor from (4) to the graph: for each nearest neighboring planar patch to the current pose, we check that the χ^2-error from (4) is below a threshold corresponding to a confidence region of 99%. We have found that this simple method performs well—even when the odometry is quite noisy as in §III.

III. Simulated Trials

To evaluate the proposed technique for modeling curved surfaces, we simulated a robot with a sparse 3D scanner surveying a sphere with radius $r = 8$ m at a standoff of 1 m. Clearly, this experiment motivates the use of a piecewise-planar model because the surface being observed is curved everywhere.



(a) SLAM vs. DR (b) SLAM vs. ground-truth (c) Covariance over time

Fig. 5. Results of the SLAM simulation using a spherical surface and a 3D robot. In (a), the left plot shows a visualization of the coplanar factors from (4) as yellow lines. The cyan region denotes the 99% positional confidence region. This confidence region grows at the beginning and end of the simulation, when the surface normals are not capable of correcting odometry error. This explains why the center of the reconstructed sphere in (b) does not align with ground-truth. In (c), the positional and orientational uncertainties are shown for each pose. For the first and last 100 poses, the surface normals are approximately aligned with gravity, and this explains why uncertainty grows at the start and end of the simulation, but is bounded during the interim.

In this simulation, the robot traverses the surface of the sphere in a spiral-like pattern in such a way that the robot never observes the sphere from the same pose more than once. We corrupt the ground-truth to provide odometry measurements, but have the z, pitch, and roll measurement uncertainties bounded (as would be the case from a pressure depth sensor and gravity-derived roll/pitch measurements in our application). The robot observes one planar patch for each pose, so we use the 9-DOF graph representation that is described in §II-C. The results of this experiment are shown in Fig. 5.

Using our approach, the robot is able to reasonably reconstruct an estimate of its trajectory and surface of the sphere. Interestingly, the robot accumulates most of the pose uncertainty at the top and bottom of the sphere, as shown in Fig. 5(c). At these portions of the sphere, the surface normal measurements are unable to correct for lateral and heading (x, y, yaw) error because the surface normals align with gravity, which bounds the z, pitch, and roll uncertainties. However, once the robot maps a sphere, the robot is well-localized relative to the sphere, and uncertainty in a global sense is bounded. This suggests the best approach for robot mapping would be to minimize time spent accumulating error in regions where the surface normals are not providing any way to correct odometry error.

Because initial odometry error in the simulation cannot be corrected with planar measurements, we avoid evaluating one-to-one error between ground-truth poses and SLAM poses. For this simulation, we instead evaluate the closeness of the SLAM map to the true sphere by fitting a least-squares sphere for the estimated robot trajectory. We compute the best-fit sphere center $\mathbf{c}$ and radius, ρ, by solving

$$\underset{r,\mathbf{c}}{\operatorname{argmin}} \sum_{i=1}^{N} \left(\|\mathrm{P}_i - \mathbf{c}\| - \rho \right)^2 . \tag{8}$$

For this simulation, $\mathrm{P} \in \mathbb{R}^{3\times N}$ consists of the positions for each of the N estimated poses. We evaluate this cost function using the true radius, which indicates the closeness of the positions to ground-truth. These values are shown in Fig. 6, where we also analyze the sensitivity of our approach to the characteristic radius of the sphere.

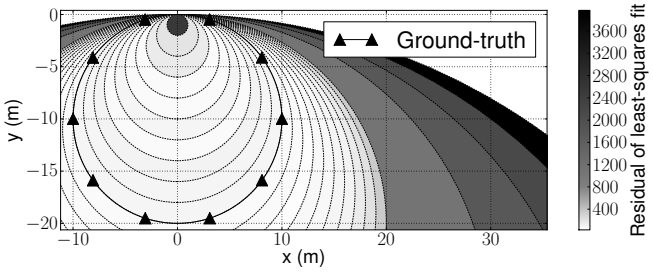


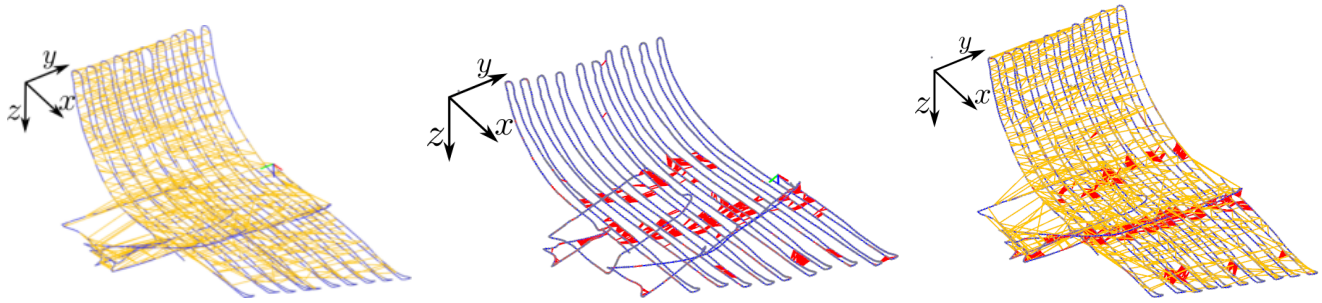
Fig. 6. The results of our simulation suggest that our approach produces good SLAM estimates even if the characteristic curvature is not the ground-truth value. We chose a variety of radii for our simulation, and we shaded the corresponding circular regions above according to how well the ground-truth sphere from Fig. 5(b) fits the estimated trajectory. All user-defined radii between 3 and 19 m produce good SLAM estimates compared to the true value, 8 m, which is marked with triangles.

IV. Experimental Trials

A. Improvements to Visual SLAM

The HAUV, shown in Fig. 1, continuously observes the underwater surface of a ship hull. The vehicle's DVL sensor is constantly pointed nadir to the hull so as to maintain a constant standoff and orientation. This provides velocity-derived hull-relative dead-reckoning, however in this experiment we also apply our piecewise-planar model to the sparse 3D point cloud produced by the sensor. The planar factors, as described in §II-B work concurrently with 5-DOF bearing-only camera measurements taken from the vehicle's monocular camera. These camera-derived measurements constitute the crux of the visual SLAM system used on the HAUV [12], [13].

The characteristic azimuth and elevation radii were user-determined from the physical lengths of the ship. We estimate these values by fitting least-squares circles to small side-to-side and top-to-bottom portions of DVL point clouds. For the



(a) Planar factors only (b) Camera factors only (c) Planar and camera factors

Fig. 7. The results of the HAUV dataset on a 25 m × 15 m × 10 m section of the hull are shown in (a), (b), and (c). Each plot shows the qualitative effects of the different factor types available to our SLAM system. Yellow lines denote the factors that enforce the piecewise-planar constraint from (4), and red lines represent monocular camera measurements. The piecewise-planar constraint provides noticeable translational correction along the x-axis and rotational correction about the z-axis. Furthermore, the use of piecewise-planar constraints offers far more visual loop closures, to be shown in Fig. 8.

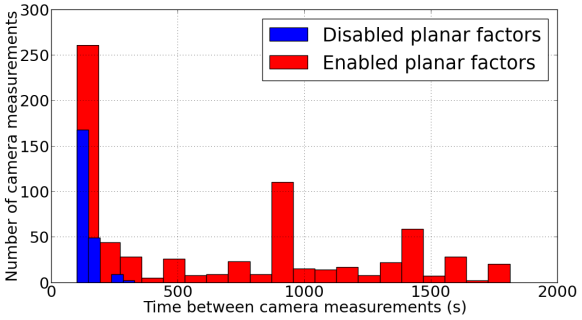


Fig. 8. The use of planar factors produces a more geometrically-consistent map, because we successfully capture more informative camera measurements. This histogram shows the time difference between nodes that are constrained by camera measurements. Larger times indicate larger loop closures. Using our piecewise-planar approach, we can add more loop closures to the graph because the reconstruction is more self-consistent. On the other hand, by disabling the planar factors the model becomes inconsistent and we can only make successful camera measurements between nodes that are closely separated in time.

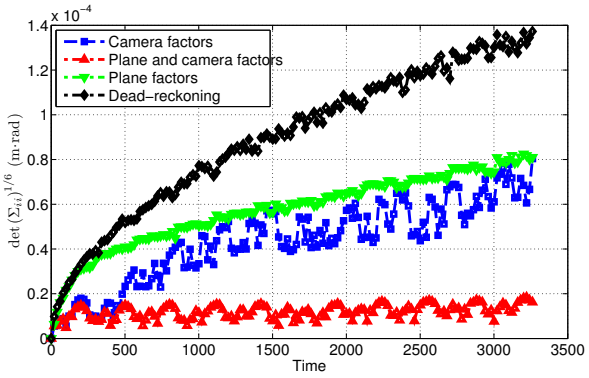


Fig. 9. Marginal covariance over time for the experiment shown in Fig. 7. The green, blue, and red curves denote the uncertainties of Fig. 7(a), Fig. 7(b), and Fig. 7(c), respectively. These uncertainties are computed based on the method from [24], which takes the sixth root of the 6-DOF pose covariance, and has units $\mathrm{m} \cdot \mathrm{rad}$. The addition of planar factors to the visual SLAM graph (yellow lines from Fig. 7(c)) lowers the uncertainty of the estimate substantially.

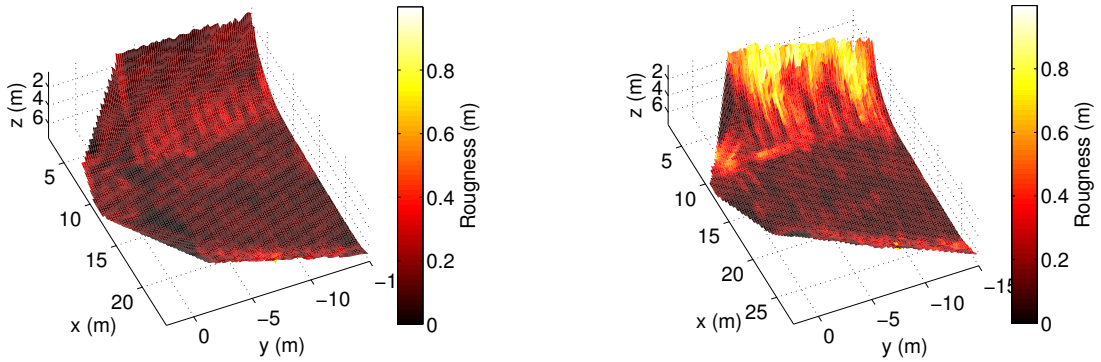
ship used in this experiment, the azimuth (side-to-side) and elevation (top-to-bottom) characteristic radii were 322 m and 7 m, respectively. In practice, we found that we can double or half these parameters before noticing any significant change in the performance, confirming that our approach is not particularly sensitive to the characteristic radii.

As shown in Fig. 7, incorporating our method into the SLAM system produces more consistent maps. We can quantitatively measure consistency simply by counting the number of large visual loop closures, as shown in Fig. 8. More precisely, we count the number of camera measurements between two nodes that are separated in time by more than 300 s. Using our planar factors, we successfully add 401 such loop closures to the graph. Without using our approach, we only add 2. This suggests that our approach produces more geometrically consistent maps, which can be understood by considering that the initial guess of the two-view camera measurements are derived using the current estimate of the vehicle poses. So, an inconsistent map leads to inconsistent initialization, which in turn leads to unsuccessful camera measurements. Moreover, the marginal pose covariances are also significantly improved using our planar factors. This is shown in Fig. 9, where covariances from Fig. 7(b) and Fig. 7(c) are plotted over time.

Finally, as a sanity check, we can define a roughness measure for the hull surface reconstruction by computing PCA over small neighborhoods of the DVL point cloud. The roughness measure around a point is defined as the smallest singular value of a data matrix consisting of the 20 nearest neighbors. Smaller values denote smoother surfaces. These values are overlaid onto a meshed point cloud in Fig. 10 for a visual comparison.

B. Multi-Session SLAM

The use of planar and camera factors extends naturally to the multi-session SLAM problem with the support of anchor nodes [22]. In short, we can co-register two sessions, 'A' and 'B', with the addition of two nodes to the factor-graph, each representing an unknown 6-DOF transformation. These unknown transformations simply relate the global frame to A's frame, and the global frame to B's frame. We then



(a) With planes (b) Without planes

Fig. 10. Roughness measures of the DVL point cloud after SLAM optimization, with smaller values encoding smoother surfaces (best viewed in color). (a) The addition of our planar factors to the graph makes the DVL point cloud roughness measures very small. (b) Disabling our planar factors, the roughness measure increases drastically because the planar patches are not being explicitly optimized in the SLAM backend.

incorporate the anchor node to the planar factor from (4) by applying a root-shift to each pose before evaluating the potential.

For the HAUV, each session is separated in time and the robot is tasked with aligning the current survey to a previously-completed survey. Anchor nodes require that the current SLAM session observes a full 6-DOF transformation to another session before rank-deficient measurements like the planar factors can constrain the anchor nodes. For the HAUV, this full transformation is approximated with a bearing-only monocular camera measurement by assigning a rough estimate of scale. The covariance of the camera measurement is properly inflated along the rank-deficient eigenvector to ensure that the matrix is full-rank.

The addition of anchor node support to planar factors provides improved results for multi-session SLAM with the HAUV. We globally aligned two surveys that occur on the same day, with each survey having its own global frame. The reset of the global frame happens each time the vehicle completes a survey. We initialize the global alignment with a camera measurement, and add planar factors between the two surveys to refine the alignment, which typically leads to the addition of more camera measurements. These results are shown in Fig. 11.

V. Conclusion

We introduced a technique using planar patches as a way to approximate smooth, curved surfaces for real-time factor-graph-based SLAM. We presented and verified a simple model to incorporate a notion of a surface's characteristic curvature when using our planar factors to constrain planar patches and 3D robot poses. This model was verified in simulation and showed that a robot with high odometry uncertainty can reconstruct a sphere using only planar surface measurements.

The commonly-used DVL sensor for underwater vehicle navigation produces a useful by-product: a sparse 3D point cloud. We have shown that this information can be quite useful for SLAM applications involving continuous observations of mostly-smooth surfaces, like ship hulls. Using data from the HAUV, we have shown an increase in map consistency and a decrease in pose uncertainty using our approach. Finally, the use of anchor nodes for multi-session SLAM can be easily incorporated into our planar factors, and we have shown a noticeable improvement in multi-session SLAM using the HAUV.

Our approach models the characteristic curvature of the surface being observed by the robot. The two user-provided parameters, the characteristic radii in the azimuth and elevation dimensions, are held as fixed for the simulation and experimental trials in this paper. Though this achieves good results for our experiments, future work could involve the online segmentation of the observed planar patches into separate characteristic radii. This would be necessary if the observed surface curvature is not well-approximated by a fixed parameter.

References

[1] R. Smith, M. Self, and P. Cheeseman, "Estimating uncertain spatial relationships in robotics," in *Autonomous Robot Vehicles*, I. Cox and G. Wilfong, Eds. Springer-Verlag, 1990, pp. 167–193.

[2] S. Thrun, M. Montemerlo, D. Koller, B. Wegbreit, J. Nieto, and E. Nebot, "FastSLAM: An efficient solution to the simultaneous localization and mapping problem with unknown data association," *J. Machine Learning Res.*, vol. 4, no. 3, pp. 380–407, 2004.

[3] F. Dellaert and M. Kaess, "Square root SAM: Simultaneous localization and mapping via square root information smoothing," *Int. J. Robot. Res.*, vol. 25, no. 12, pp. 1181–1203, 2006.

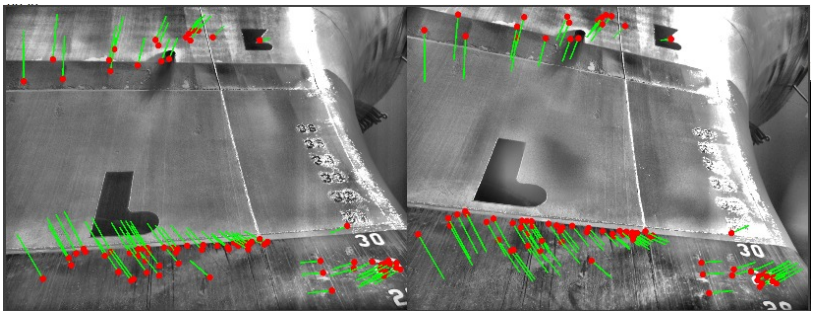
[4] M. Kaess, A. Ranganathan, and F. Dellaert, "iSAM: Incremental smoothing and mapping," *IEEE Trans. Robot.*, vol. 24, no. 6, pp. 1365–1378, Dec. 2008.

[5] G. Grisetti, D. Lodi Rizzini, C. Stachniss, E. Olson, and W. Burgard, "Online constraint network optimization for efficient maximum likelihood map learning," in *Proc. IEEE Int. Conf. Robot. and Automation*, Pasadena, CA, May 2008, pp. 1880–1885.

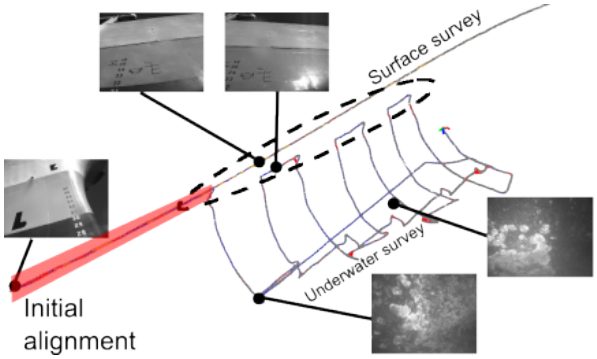
[6] Z. Zhang, "Iterative point matching for registration of free-form curves and surfaces," *Int. J. Comput. Vis.*, vol. 13, no. 2, pp. 119–152, Oct. 1994.

[7] E. Olson, "Real-time correlative scan matching," in *Proc. IEEE Int. Conf. Robot. and Automation*, Kobe, Japan, June 2009, pp. 4387–4393.

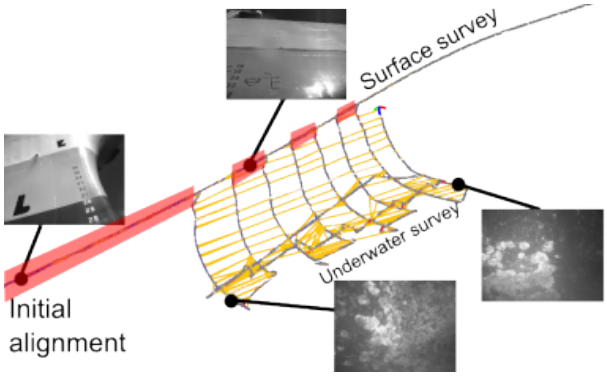
[8] R. M. Eustice, H. Singh, and J. Leonard, "Exactly sparse delayed-state filters," in *Proc. IEEE Int. Conf. Robot. and Automation*, Barcelona, Spain, Apr. 2005, pp. 2417–2424.



(a) Initial camera registration



(b) Alignment without planar factors



(c) Alignment with planar factors

Fig. 11. Overview of our multi-session SLAM system, where a 130 m surface survey is globally aligned with an underwater survey. In (a), two views from each survey are used to compute a relative-pose from which to initialize the global alignment. This alignment is refined with the addition of more camera measurements and planar factors. In (b) and (c), we attempt this experiment with and without the aid of planar factors. Without planar factors, the robot is only initially aligned to the previous survey, with the red highlighted region denoting areas with frequent camera measurements between the two dives. After submerging and surfacing, however, uncorrected odometry error causes the failed detection of loop closures at subsequent surfacings, shown in the dotted circled region. In (c), the addition of planar factors enforces additional constraints on the global alignment, which leads to a more geometrically consistent map. In this setting, the robot regularly adds camera measurements between the two surveys in the regions highlighted in red.

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

[10] I. Mahon, S. B. Williams, O. Pizarro, and M. Johnson-Roberson, "Efficient view-based SLAM using visual loop closures," *IEEE Trans. Robot.*, vol. 24, no. 5, pp. 1002–1014, Oct. 2008.

[11] P. Ridao, M. Carreras, D. Ribas, and R. Garcia, "Visual inspection of hydroelectric dams using an autonomous underwater vehicle," *J. Field Robot.*, vol. 27, no. 6, pp. 759–778, Nov. 2010.

[12] F. S. Hover, R. M. Eustice, A. Kim, B. Englot, H. Johannsson, M. Kaess, and J. J. Leonard, "Advanced perception, navigation and planning for autonomous in-water ship hull inspection," *Int. J. Robot. Res.*, vol. 31, no. 12, pp. 1445–1464, October 2012.

[13] A. Kim and R. M. Eustice, "Real-time visual SLAM for autonomous underwater hull inspection using visual saliency," *IEEE Trans. Robot.*, vol. 29, no. 3, pp. 719–733, June 2013.

[14] C. Kunz and H. Singh, "Map building fusing acoustic and visual information using autonomous underwater vehicles," *J. Field Robot.*, 2013, In Print.

[15] Teledyne RD Instruments. (2006) RD Instruments Workhorse Navigator DVL. Specification sheet and documentations Available at `www.rdinstruments.com`.

[16] Velodyne, Inc. (2007, October) Velodyne HDL-64E: A high definition LIDAR sensor for 3D applications. Specification sheet and documentations Available at `www.velodyne.com`.

[17] S. Barkby, S. B. Williams, O. Pizarro, and M. V. Jakuba, "Bathymetric SLAM with no map overlap using gaussian processes," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots and Syst.*, 2011, pp. 1242–1248.

[18] J. Weingarten and R. Siegwart, "3D SLAM using planar segments," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots and Syst.*, 2006, pp. 3062–3067.

[19] A. J. B. Trevor, J. G. Rogers, and H. I. Christensen, "Planar surface SLAM with 3D and 2D sensors," in *Proc. IEEE Int. Conf. Robot. and Automation*, 2012, pp. 3041–3048.

[20] T. Pietzsch, "Planar features for visual SLAM," *Advances in Artificial Intelligence*, pp. 119–126, 2008.

[21] M. Ruhnke, R. Kummerle, G. Grisetti, and W. Burgard, "Highly accurate 3D surface models by sparse surface adjustment," in *Proc. IEEE Int. Conf. Robot. and Automation*, 2012, pp. 751–757.

[22] B. Kim, M. Kaess, L. Fletcher, J. J. Leonard, A. Bachrach, N. Roy, and S. Teller, "Multiple relative pose graphs for robust cooperative mapping," in *Proc. IEEE Int. Conf. Robot. and Automation*, Anchorage, Alaska, May 2010, pp. 3185–3192.

[23] N. J. Mitra, A. Nguyen, and L. Guibas, "Estimating surface normals in noisy point cloud data," *Int. J. of Comput. Geom. and Applicat.*, vol. 14, pp. 261–276, 2004.

[24] H. Carrillo, I. Reid, and J. A. Castellanos, "On the comparison of uncertainty criteria for active SLAM," in *Proc. IEEE Int. Conf. Robot. and Automation*, 2012, pp. 2080–2087.