Mapping underwater ship hulls using a model-assisted bundle adjustment framework

Paul Ozog[a,1,*], Matthew Johnson-Roberson[a,2], Ryan M. Eustice[a,2]

[a] *University of Michigan, Ann Arbor, Michigan, USA 48109*

Abstract

This paper reports on a model-assisted bundle adjustment (BA) framework in which visually-derived features are fused with an underlying three-dimensional (3D) mesh provided *a priori*. By using an approach inspired by the expectation-maximization (EM) class of algorithms, we introduce a hidden binary label for each visual feature that indicates if that feature is considered part of the nominal model, or if the feature corresponds to 3D structure that is absent from the model. Therefore, in addition to improved estimates of the feature locations, we can identify visual features that correspond to foreign structure on the ship hull. We show that this framework is a special case of the Gaussian max-mixtures framework, which can be efficiently incorporated into state-of-the-art graph-based simultaneous localization and mapping (SLAM) solvers.

In addition, the precision of our bundle adjustment framework allows the identification of structural deviations between 3D structure inferred from bundle-adjusted camera imagery and the prior model. These structural deviations are clustered into shapes, which allow us to fuse camera-derived structure back into the 3D mesh. This augmented model can be used within a 3D photomosaicing pipeline, providing a visually intuitive 3D reconstruction of the ship hull. We evaluate our pipeline using the Bluefin Robotics hovering au-

[*] Corresponding author

Email addresses: paulozog@umich.edu (Paul Ozog), mattjr@umich.edu (Matthew Johnson-Roberson), eustice@umich.edu (Ryan M. Eustice)

[1] Electrical Engineering and Computer Science Department

[2] Naval Architecture and Marine Engineering Department

tonomous underwater vehicle (HAUV) surveying the *SS Curtiss*, where a 3D mesh derived from computer aided design (CAD) drawings serves as the prior model. In addition to more consistent visual reconstructions, we can update the prior mesh with 3D information corresponding to underwater structure, such as biofouling or manually-placed cylindrical shapes with known dimensions.

Keywords: SLAM, AUVs, underwater inspection, mapping, visualization

1. Introduction

Bundle Adjustment (BA) is a special case of the simultaneous localization and mapping (SLAM) problem; it is an estimation problem whose unknowns consist of camera poses and the positions of visually-observed features. Thus, BA is a reconstruction technique that seeks to estimate the three-dimensional (3D) structure of the scene and the egomotion of the camera. It is a widespread technique used throughout computer vision and mobile robotics, due mainly to the low cost and high reconstruction quality of digital cameras [2, 10, 17]. A major drawback of using optical cameras in field robotics is the susceptibility to environmental noise and varying lighting conditions [15]. Despite these challenges, the main benefit of vision-based 3D reconstruction is high spatial resolution and cost savings as compared to laser and acoustic-based sensing.

To mitigate the challenges of vision-based 3D reconstruction, researchers have previously proposed modifications to BA that leverage 3D models of the scene (provided *a priori*). This practice is sometimes referred to as *model-assisted bundle adjustment*. The reconstruction of human faces has been a particularly prevalent application domain, however these techniques have certain shortcomings that are ill-suited for their application in large-scale robotic surveillance. For instance, mobile robots typically survey areas that are much larger than themselves, unlike the relative sizes of a camera and human faces. In addition, the scene will almost surely consist of 3D structure that is absent from the provided model.

Underwater environments are especially challenging for optical imaging and 3D reconstruction. In particular, a phenomenon known as *back-scatter* is an issue that researchers must consider when deploying autonomous underwater vehicles (AUVs) that are equipped with optical cameras [19, 32, 4, 11]. Despite these challenges, the benefits of the aforementioned model-assisted BA have yet to be explored in a large scale underwater setting. The facial reconstruction methods are designed for standard in-air imaging so these methods do not easily transition to underwater environments.

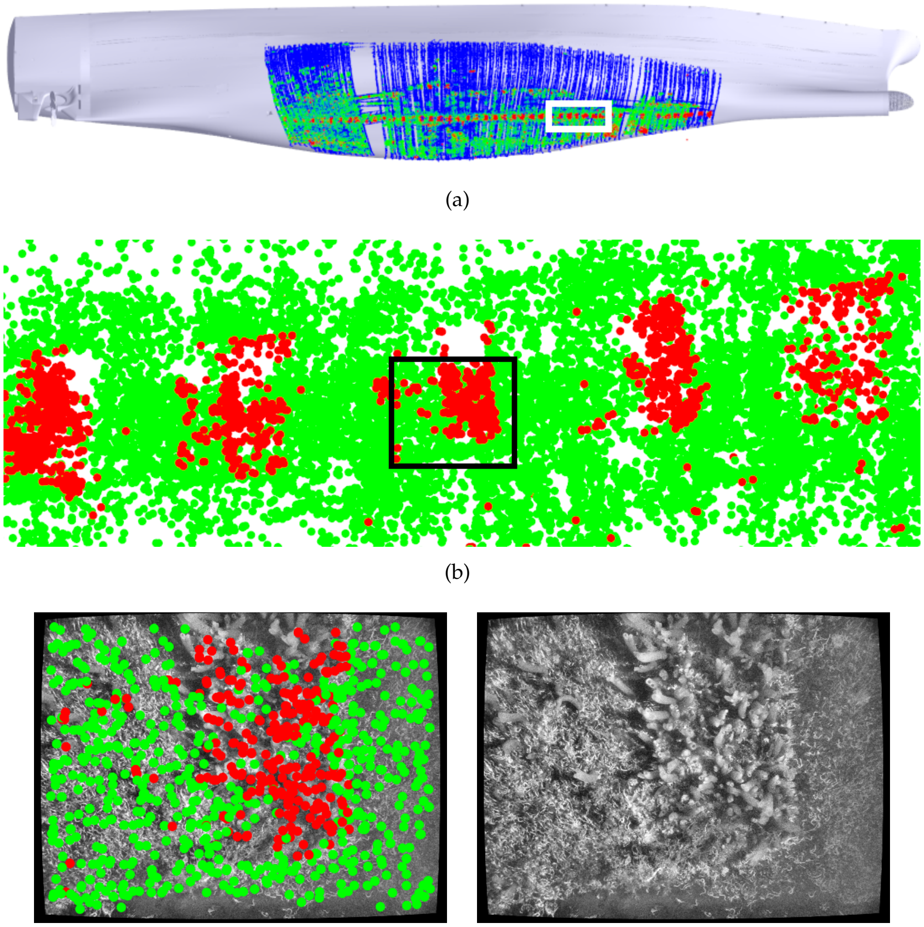
To address these challenges, we have developed an improved model-assisted BA framework that is easily applicable to underwater ship hull inspection, as shown in Fig. 1. In addition, we have leveraged this BA framework in a mapping pipeline that can identify foreign 3D structure and fuse it back into the prior model, as shown in Fig. 2. The contributions of Section 2 are as follows:

- We propose a expectation-maximization (EM) algorithm that assigns hard binary labels to each visual feature and solves for the optimal 3D locations of cameras and features accordingly. This approach is therefore capable of identifying 3D structure that is absent from the prior model.

- We show that this algorithm is a special case of the Gaussian max-mixture framework, which was originally intended for handling non-Gaussian error models in graphical SLAM [47].

- To our best knowledge, our datasets provide the largest evaluation of a fielded robot performing model-assisted BA, both in physical scale and number of images.

In addition, in Section 3 we explore mapping techniques that fuse the model-assisted bundle-adjusted structure (in addition to texture, similar to our method presented in [48]) back into the low-fidelity prior mesh. In this paper, we are interested in identifying structural differences detected from the underwater camera. We build upon this previous work by annotating a prior model with



(a)

(b)

(c)

(d)

Figure 1: (a) DVL ranges, shown in blue, allow us to localize to the prior model of the ship being inspected, shown in gray. In (b) and (c), visual features that are hypothesized to lie on the nominal surface of the prior model are shown in green. Features that correspond to 3D structure that is absent from the model are shown in red. In this example, red features correspond to tubular biogrowth emanating from docking blocks along the hull's centerline, in (d).

SLAM-derived structure. We show experimental results taken from the Bluefin Robotics hovering autonomous underwater vehicle (HAUV) platform for automated ship hull inspection [29]. The contributions of Section 3 allow our AUV to:

- Label visually-derived 3D shapes based on their deviation from the nominal *a priori* mesh.

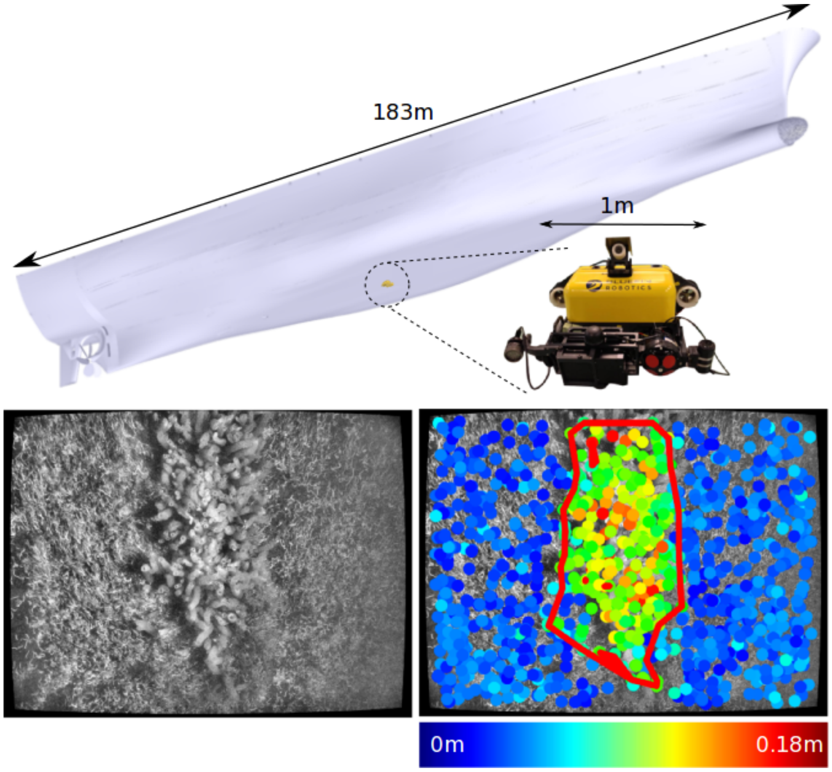


Figure 2: In addition to an improved BA framework, we propose a mapping technique that can identify 3D shapes in the imagery (bottom) and fuse these back into a low-fidelity prior model derived from CAD drawings (top). The color of each feature encodes the structural deviation from the CAD model. An outline of clusters of these features is shown in red, computed using DBSCAN.

- Augment the nominal mesh with visually-derived 3D information, producing a high-fidelity map.

1.1. Related Work: Model-assisted BA

Model-assisted visual reconstruction methods were particularly popular during the late 1990's and early 2000's, particularly in the domain of human face reconstruction [24, 35, 55, 14, 23]. Works by Fua [24] and Kang and Jones [35] are similar to traditional BA: a least-squares minimization over reprojection error. However, they introduce regularization terms that essentially enforce triangulated points lying close to the model's surface. Shan et al. [55] introduce an optimization problem over a set of model parameters—rather than a regularization over features—that allows the generic face model to more closely match the geometry of the subject's face. Fidaleo and Medioni [23] noted that these methods are rarely able to integrate 3D structure present in the subject that is absent from the model (such as facial hair and piercings). Instead, their approach used a prior model strictly for pose estimation, but the reconstruction of the face was entirely data-driven.

The primary application domain of these methods is in the reconstruction of human faces, however they have largely been overshadowed by modern, highly accurate, dense reconstruction methods that use either commodity depth cameras [45, 63], patch-based multiview stereopsis using high-quality imagery [25], or photometric stereo reconstruction techniques [37, 52]. These more recent methods have shown impressive reconstructions of both small-scale objects (human faces), and large scale objects (indoor environments and outdoor structures).

Recently, however, model-assisted methods have seen some re-emergence in particularly challenging areas of mobile robotics, such as the work by Geva et al. [26] in which an unmanned aerial vehicle (UAV) surveys a remote area. They used digital terrain models (DTMs) to regularize the position of 3D features observed from the camera mounted on the UAV, in a very similar fashion to the work in [24, 35]. These DTMs are freely available from the Shuttle Radar Topography project [22], and act as the prior model used in their approach. This approach is most similar to ours, however we differentiate our approach in three important ways: (i) our approach is capable of incorporating visual information that is absent from the nominal *a priori* model by assigning a hidden binary random variable for each visual feature; (ii) we use an orthogonal signed distance, rather than raycasting, to evaluate a feature's surface constraint likelihood; (iii) we evaluate our approach on a dataset with several orders of magnitude more bundle-adjusted keyframes.

1.2. Related Work: Underwater Visual Mapping

Early work in ship hull inspection includes the use of long-baseline navigation, where a robot localizes to a ship hull using manually-deployed acoustic pingers [61]. More recently, researchers have instead used underwater visual perception and SLAM techniques, rather than acoustic localization beacons, for AUV navigation. A survey of underwater visual sensing modalities was provided by [7]. Some examples of field robots that use visual perception include work by Negahdaripour and Firoozfam [44], in which they used a stereo camera rig on a remotely operated vehicle (ROV) to inspect the underwater portion of a ship hull. Visual mapping of underwater infrastructure was also explored by Ridao et al. [50] using an AUV with a calibrated underwater monocular vision system. In addition to mapping tasks, several researchers have explored automated object identification (such as corrosion or underwater mines) using both visual sensors [8] and acoustic sensors [6].

The computer vision and graphics community have studied fusing optical range measurements to form a reconstruction of a 3D surface for several decades [27, 16, 36]. The seminal work by Curless and Levoy [16] used running averages to fuse range measurements into an implicit surface. This simple approach is still used in state-of-the-art surface reconstruction and pose tracking algorithms using a commodity depth camera [45, 63]. We differentiate our work in three ways. First, we assume the availability of a nominal mesh of the surface being constructed and that the camera pose with respect to this mesh is unknown. Second, we assume that the object can only be observed at a very close distance—i.e., the observations are locally planar and so using iterative closest point (ICP) (or its variants) to estimate relative poses between keyframes is ill-constrained [65, 13, 54]. Third, we do not assume the availability of dense depth images during the pose estimation stage. Though we use a stereo camera for some of our experimental analysis (from which a dense disparity map can be easily converted to a dense depth image), we instead use sparse feature-based registration so that this approach is also applicable to monocular (bearing-only) cameras.

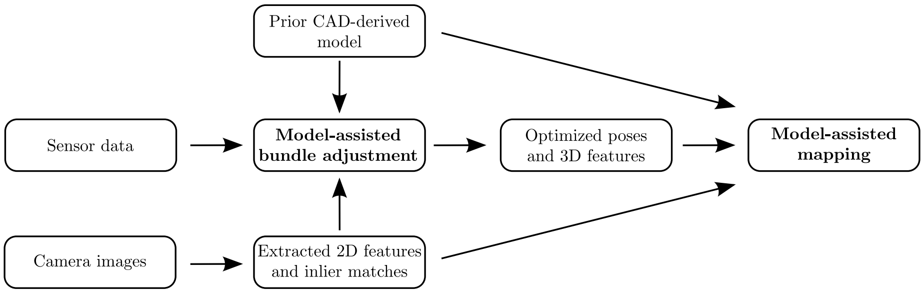

Prior CAD-derived model
Sensor data
Model-assisted bundle adjustment
Optimized poses and 3D features
Model-assisted mapping
Camera images
Extracted 2D features and inlier matches

Figure 3: Flowchart of the model-assisted mapping pipeline.

1.3. Outline

This paper is organized into two major components: a model-assisted BA framework discussed in Section 2 and a underwater visual mapping pipeline discussed in Section 3. In Fig. 3, we provide a flowchart of our processing pipeline and show that the output of model-assisted BA serves as the input to model-assisted mapping. In Section 2, we describe the mathematical model for a robust model-assisted optimization framework, and we show that the approach is a special case of the Gaussian max-mixture models from Olson and Agarwal [47]. In Section 3, we describe a mapping framework that fuses the BA result back into the prior model in the form of triangulated 3D shapes derived from a clustering algorithm. Results and discussion are provided in Section 4. In Section 5 we offer some concluding remarks.

1.4. Sensor Payload

The HAUV acts as the experimental platform for the methods proposed in this paper. An illustration of the HAUV's sensor configuration is shown in Fig. 4. The light-emitting diode (LED) light, underwater stereo camera, Dual frequency IDentification SONar (DIDSON) imaging sonar, and Doppler velocity log (DVL) are mounted on a sensor tray at the front of the vehicle, while the periscope camera is placed on top. In this configuration, the robot can easily capture either below-water (stereo or monocular) or above-water (monocular periscope) images.

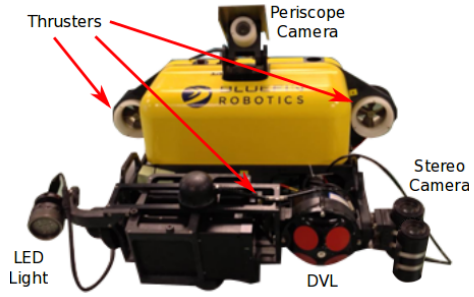


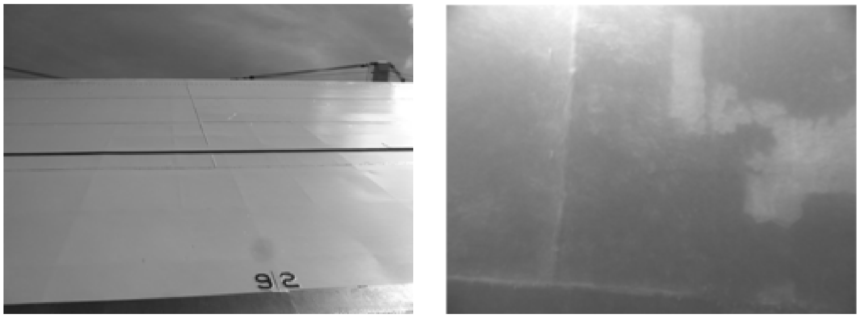
Figure 4: Overview of the HAUV sensor configuration.

Table 1: Payload characteristics of the HAUV

Prosilica GC1380	12-bit digital stills, fixed-focus, monochrome, 1 Megapixel
Periscope Camera	Monocular Prosilica GC1380 in water-proof housing
Underwater Camera (monocular, pre-2013)	Monocular Prosilica GC1380 in water-proof housing
Underwater Camera (stereo, post-2013)	Two Prosilica GC1380s in separate water-proof bottles, linked via Fast Ethernet
Lighting	520 nm (green) LED
IMU	Honeywell HG1700
Depth	Keller pressure sensor
DVL	RDI 1200 kHz Workhorse; also provides four range beams
Imaging Sonar	Sound Metrics 1.8 MHz DIDSON
Communication	Fiber-optic Ethernet cable
Thrusting	Five rotor-wound thrusters
Battery	1.5 kWh lithium-ion
Dry Weight	79 kg
Dimensions	1 m × 1 m × 0.45 m

During a mission, the sensor tray is servoed such that both the DVL and camera point nadir to the hull while the robot keeps a fixed distance. In late 2013, we upgraded the underwater monocular camera to a stereo configuration. The periscope camera, on the other hand, is fixed with a static angle allowing the robot to image superstructure to localize previous missions, as shown in [49].

The specifics of the sensor suite are tabulated in Table 1 and reported by Hover et al. [30]. The inertial measurement unit (IMU), DVL velocities, and depth sensors are used for dead-reckoning (DR), and the cameras, sonar, and DVL ranges are used for perceptual information. In Fig. 5, we provide typical examples of both above-water (periscope) and underwater imagery.


(a) Periscope camera

(b) Underwater camera

Figure 5: Sample periscope and underwater imagery.

2. Model-assisted Bundle Adjustment

2.1. Notation

We denote the set of all unknowns, $\mathbf{X}$, as consisting of N_p poses, $\mathbf{x}_{g1} \dots \mathbf{x}_{gN_p}$, the relative transformation to the model frame, $\mathbf{x}_{g\mathcal{M}}$, and N_l landmarks, $\mathbf{l}_1 \dots \mathbf{l}_{N_l}$

$$\mathbf{X} = \{\underbrace{\mathbf{x}_{g1} \dots \mathbf{x}_{gN_p}}_{\text{robot poses}}, \underbrace{\mathbf{x}_{g\mathcal{M}}}_{\text{model pose}}, \underbrace{\mathbf{l}_1 \dots \mathbf{l}_{N_l}}_{\text{visual landmarks (features)}}\},$$

where $\mathbf{x}_{ij}$ denotes the 6-degree-of-freedom (DOF) relative-pose between frames i and j. The common, or global frame, is denoted as g. Visually-derived features, denoted as $\mathbf{l}_i$, are the 3D positions of features as expressed in the global frame. Finally, $\mathcal{M}_{\text{prior}}$ denotes a prior triangular mesh consisting of a set of vertices, edges between vertices, and triangular faces.

Note that $\mathbf{X}$ may consist of additional variables, such as extrinsic parameters of the robot sensors. We omit these values now for the sake of clarity, however we re-introduce them in Section 2.6.

Let $\mathbf{Z}$ denote the set of all measurements, which consists of all odometry measurements, priors, surface range measurements (e.g., from an active range scanner), visual feature detections, and surface constraints (which will be described in Section 2.3),

$$\mathbf{Z} = \{\mathcal{Z}_{\text{odo}}, \mathcal{Z}_{\text{prior}}, \mathcal{Z}_{\text{range}}, \mathcal{Z}_{\text{feat}}, \mathcal{Z}_{\text{surf}}\}.$$

We assign a hidden binary feature label to each visual feature,

$$\mathbf{\Lambda} = \{\lambda_1 \dots \lambda_{N_l}\}, \lambda_i \in \{0, 1\},$$

where a value of one encodes that a visually-derived feature lies on the nominal surface of $\mathcal{M}_{\text{prior}}$. A value of zero encodes that the visually-derived feature corresponds to physical structural that is absent from $\mathcal{M}_{\text{prior}}$.

2.2. Formulation as Expectation-Maximization

The goal of our work is to estimate $\mathbf{X}$ using a simplified variant of the EM algorithm, known as *hard EM*:

1. Initialize $\mathbf{X}$
2. Repeat the following until $p(\mathbf{Z}, \mathbf{\Lambda}|\mathbf{X})$ converges:
    (a) $\mathbf{\Lambda}^* = \underset{\mathbf{\Lambda}}{\operatorname{argmax}}\, p(\mathbf{Z}, \mathbf{\Lambda}|\mathbf{X})$
    (b) $\mathbf{X}^* = \underset{\mathbf{X}}{\operatorname{argmax}}\, p(\mathbf{Z}, \mathbf{\Lambda}^*|\mathbf{X})$

Similar to previous work, we introduce a set of prior measurements, $\mathcal{Z}_{\text{surf}}$, that regularize the positions of 3D visual features so that they lie on the surface of $\mathcal{M}_{\text{prior}}$. We expand the likelihood function using Bayes' rule and note that the odometry, prior, and feature detection observations are independent of the feature labels (and conditionally independent of each other):

$$\begin{aligned} p(\mathbf{Z}, \mathbf{\Lambda}|\mathbf{X}) &= p(\mathbf{Z}|\mathbf{\Lambda}, \mathbf{X})p(\mathbf{\Lambda}|\mathbf{X}) \\ &= p(\mathcal{Z}_{\text{odo}}, \mathcal{Z}_{\text{prior}}, \mathcal{Z}_{\text{range}}, \mathcal{Z}_{\text{feat}}|\mathbf{X})p(\mathcal{Z}_{\text{surf}}|\mathbf{\Lambda}, \mathbf{X})p(\mathbf{\Lambda}|\mathbf{X}). \end{aligned} \tag{1}$$

If we conservatively assume that $p(\lambda_i|\mathbf{X})$ is uninformative, then we can express the likelihood as proportional to a simpler expression:

$$p(\mathbf{Z}, \mathbf{\Lambda}|\mathbf{X}) \propto p(\mathcal{Z}_{\text{odo}}, \mathcal{Z}_{\text{prior}}, \mathcal{Z}_{\text{range}}, \mathcal{Z}_{\text{feat}}|\mathbf{X})p(\mathcal{Z}_{\text{surf}}|\mathbf{\Lambda}, \mathbf{X})$$

Therefore, Step 2(a) in the hard EM algorithm simplifies to

$$\underset{\mathbf{\Lambda}}{\operatorname{argmax}}\, p(\mathbf{Z}, \mathbf{\Lambda}|\mathbf{X}) = \underset{\mathbf{\Lambda}}{\operatorname{argmax}}\, p(\mathcal{Z}_{\text{surf}}|\mathbf{\Lambda}, \mathbf{X}), \tag{2}$$

where $p(\mathcal{Z}_{\text{surf}}|\mathbf{\Lambda}, \mathbf{X})$ is described in Section 2.3. In addition, Step 2(b) simplifies to

$$\begin{aligned}
&\operatorname*{argmax}_{\mathbf{X}} p(\mathbf{Z}, \mathbf{\Lambda}|\mathbf{X}) = \\
&\operatorname*{argmax}_{\mathbf{X}} p(\mathcal{Z}_{\text{odo}}, \mathcal{Z}_{\text{prior}}, \mathcal{Z}_{\text{range}}, \mathcal{Z}_{\text{feat}}|\mathbf{X}) p(\mathcal{Z}_{\text{surf}}|\mathbf{\Lambda}, \mathbf{X}),
\end{aligned} \tag{3}$$

which is equivalent to a least-squares optimization problem when the measurements are corrupted by additive Gaussian noise.

2.3. Modeling the Surface Constraint

Consider the set of all surface constraints $\mathcal{Z}_{\text{surf}} = \{z_{s_1} \ldots z_{s_{N_l}}\}$, we model the conditional distribution of these constraints as Gaussian:

$$p(z_{s_i}|\lambda_i, \mathbf{X}) = \begin{cases} \mathcal{N}(h\left(\mathbf{x}_{g\mathcal{M}}, \mathbf{l}_i\right), \sigma_0^2), & \lambda_i = 0 \\ \mathcal{N}\left(h\left(\mathbf{x}_{g\mathcal{M}}, \mathbf{l}_i\right), \sigma_1^2\right), & \lambda_i = 1 \end{cases}, \tag{4}$$

where $h(\,\cdot\,)$ computes the orthogonal signed distance of the i^{th} feature to the model. The values σ_0^2 and σ_1^2 denote the variance of the surface constraint when λ_i is 0 or 1, respectively. Intuitively, these variances are chosen such that $\sigma_1^2 \ll \sigma_0^2$, i.e., features that lie close to the model surface are more tightly pulled toward it, while features that lie away from the model are free to vary with approximately zero cost. For certain applications, features may be biased toward the exterior or interior of the prior model by setting z_{s_i} to some positive or negative value, respectively. However, for our experiments, we assign $z_{s_i} = 0$ for all of the features so that the camera-derived 3D structure tends to coincide with the surface of the prior model.

The orthogonal signed distance function $h(\,\cdot\,)$ is a nonlinear function of the pose of the model and the position of the visual feature:

$$h\left(\mathbf{x}_{g\mathcal{M}}, \mathbf{l}_i\right) = \frac{\left(\mathrm{H}_g^{\mathcal{M}} \underline{\mathbf{l}_i} - \mathbf{p}\right)^\top \mathbf{n}}{\sqrt{\mathbf{n}^\top \mathbf{n}}}, \tag{5}$$

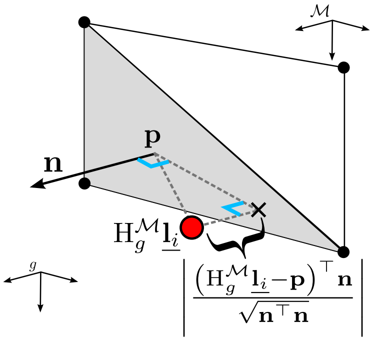

$\mathcal{M}$
$\mathbf{p}$
$\mathbf{n}$
$\mathrm{H}_g^{\mathcal{M}}\underline{\mathbf{l}_i}$
g
$\left|\frac{\left(\mathrm{H}_g^{\mathcal{M}}\underline{\mathbf{l}_i}-\mathbf{p}\right)^\top\mathbf{n}}{\sqrt{\mathbf{n}^\top\mathbf{n}}}\right|$

Figure 6: Overview of the surface constraint using a simple triangular mesh $\mathcal{M}$ consisting of two triangles. The constraint converts the distance to the closest face, d_{s_i}, to a signed distance (depending on if the feature is inside or outside the triangular face).

where $\mathrm{H}_g^{\mathcal{M}} = \left[{}^{\mathcal{M}}_{g}\mathrm{R} \,|\, {}^{\mathcal{M}}\mathbf{t}_{\mathcal{M}g}\right]$ is the transformation (orthonormal rotation matrix, ${}^{\mathcal{M}}_{g}\mathrm{R}$, followed by three-vector translation, ${}^{\mathcal{M}}\mathbf{t}_{\mathcal{M}g}$) of points in the global frame to points in the model frame and $\underline{\mathbf{u}} = [\mathbf{u}^\top 1]^\top$ represents a vector expressed in homogeneous coordinates.

Intuitively, $h(\cdot)$ returns the orthogonal signed distance of a visual feature $\mathbf{l}_i$ to the surface of the closest triangular face in $\mathcal{M}$. This triangle is determined by using a KD tree to return the closest vertex in the mesh, then arbitrarily choosing a triangle that contains this vertex. When computing $h(\cdot)$, this triangle is characterized by any point, $\mathbf{p}$, that lies on the surface of the triangle, and its surface normal, $\mathbf{n}$. This calculation is illustrated in Fig. 6.

2.4. *Relation to Gaussian Max-Mixture Models*

In this section, we show how the previous formulation is a special case of Gaussian max-mixture models proposed by Olson and Agarwal [47]. This approach was mainly introduced in the area of robust SLAM backends as a probabilistically motivated approach to rejecting incorrect loop closures [47, 1, 59] and detecting wheel slippage in ground robots [47]. More recently, it has been applied in learning robust models for consumer-grade global positioning system (GPS) measurements that can reject outliers [43].

Similar to [47], we note that the surface constraint likelihood $p(z_{s_i}|\mathbf{X})$ is not Gaussian because this density does not condition on λ. Even so, we can still apply Bayes' rule to the conditional distribution of the unknowns given the measurements. By assuming an uninformative prior on $\mathbf{X}$, we have:

$$\log p(\mathbf{X}|\mathbf{Z}) \propto \log \prod_i p(z_i|\mathbf{X}), \tag{6}$$

where $p(z_i|\mathbf{X})$ denotes the i^{th} factor potential that corresponds to an odometry, prior, range, feature, or surface information (we describe these factors in greater detail in Section 2.6). By maximizing this distribution, we arrive at a maximum *a posteriori* (MAP) estimate for $\mathbf{X}$, as shown by [18].

Though the labels, $\mathbf{\Lambda}$, are absent from (6), we can adapt the Gaussian max-mixture distribution proposed by Olson and Agarwal [47] to model the surface constraint likelihood. In our case, the surface constraint likelihood then takes the form

$$p(z_{s_i}|\mathbf{X}) = \eta \max_{\lambda_i} p(z_{s_i}|\lambda_i, \mathbf{X}). \tag{7}$$

The logarithm can be brought inside the product from (6), and again inside the $\max$ operator from (7). This distribution can therefore be thought of as a binary Gaussian max-mixture with equal weights for each component of the mixture.

This conditional distribution essentially combines Steps 2(a) and 2(b) from the hard EM algorithm so that the labels are determined whenever the likelihood term is evaluated. The distribution from (7) is therefore equivalent to a binary max-mixture of Gaussians with equal weights. This conforms to our earlier formulation from Section 2.2 that assigns equal prior probability to a surface lying on or off the mesh's surface. The only two parameters used in our approach are therefore σ_0^2 and σ_1^2 from (4). We illustrate this distribution in Fig. 7 using typical values for these parameters. These values are fixed, i.e., they are not at all dependent on the amount of structural deviation from the nominal surface.

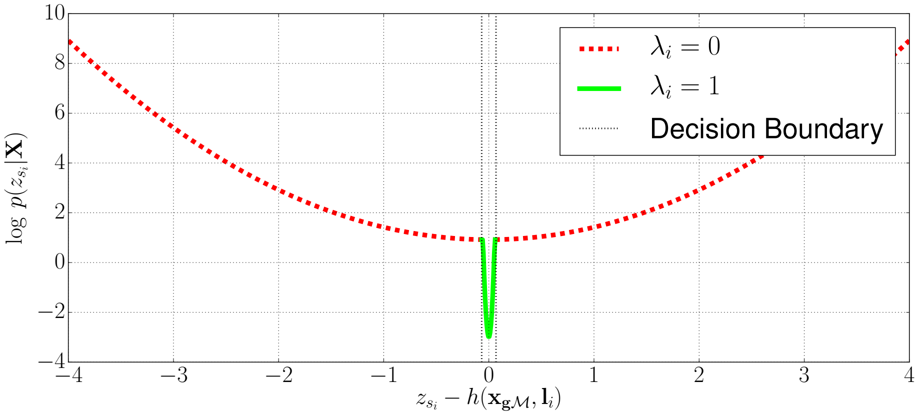


Figure 7: Decision boundary for $\sigma_0 = 1$ m, $\sigma_1 = 0.02$ m overlayed on the log probability (i.e., cost computed during optimization). The decision boundary represents approximately the minimum size of foreign objects our system can identify. For these values of σ_0 and σ_1, the decision boundary is at ± 5 cm.

The choice of σ_0 makes little difference, however σ_1 must be tuned depending on the expected variation of small-scale matter that is attached along the immediate surface of the hull. We have found for the ship hull inspection, values between 1 cm and 5 cm provide acceptable results. However, we use $\sigma_i = 2$ cm for the results in Section 4.

Note that the distribution from (7) contains an unknown normalization constant, η, that ensures a valid probability distribution. However, for the purposes of maximizing the likelihood, computing the specific value of this scale factor is not necessary [47]. Additionally, we represent the distribution from (6) using a factor graph [18], as shown in Fig. 8. To solve the corresponding least-squares problem, we use the freely-available Ceres library [3].

2.5. *Localizing to the Prior Model*

Our approach, like all model-assisted BA frameworks, requires a good initial guess of the alignment between the camera trajectory and the prior model. This is typically done using triangulated features from the camera imagery, but for autonomous ship hull inspection there are many portions of the ship where no visual features can be detected because the hull is not uniformly salient [39, 12, 40].

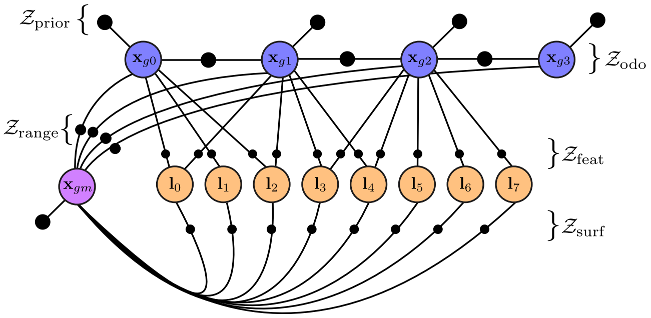


Figure 8: Representation of our method as a factor graph. The factor nodes denoted with $\mathcal{Z}_{\text{surf}}$ denote the surface constraints, which are implemented using binary Gaussian max-mixtures distributions from Section 2.4. These factors constrain the pose of the prior model $\mathbf{x}_{gm}$ and the location of visual features $\mathbf{l}_i$.

In our case, the underwater robot observed sparse range measurements using a DVL, which are a by-product of underwater navigation with a Doppler sonar [33, 42]. These range returns are rigidly aligned to the prior model using generalized iterative closest point (GICP) [54], which serves as an initial guess (i.e., the prior factor connected to node $\mathbf{x}_{gm}$ in Fig. 8). In our case, one point cloud consists of vertices in $\mathcal{M}_{\text{prior}}$, while the other point cloud consists of DVL range returns expressed in the global frame, which are SLAM-corrected in real-time using the method described in [49]. Individual poses can be further optimized using raycasting techniques to compute the likelihood of $\mathcal{Z}_{\text{range}}$ and the surface constraint measurements $\mathcal{Z}_{\text{surf}}$ from Section 2.3.

In general, we cannot use the DVL for detecting foreign objects on the hull for two reasons. First, particularly gelatinous or spongy species of biogrowth my be acoustically transparent to the DVL beams [62]. Second, the range returns are extremely sparse and thus do not provide reliably sufficient coverage of the ship hull. However, in Fig. 15 we provide results where DVL beams intersect with a solid, metallic foreign object. This is coincidental and not the general use-case.

2.6. Application to the HAUV

In Section 2.3 we described the general evaluation of the surface constraint measurements, $\mathcal{Z}_{\text{surf}}$. In this section we describe in detail the measurement models used on the HAUV and relate them to the factors illustrated in Fig. 8. Each subsection will describe $\mathcal{Z}_{\text{prior}}$, $\mathcal{Z}_{\text{odo}}$, $\mathcal{Z}_{\text{feat}}$, and $\mathcal{Z}_{\text{range}}$ in terms of the conditional distributions of the observations given the unknowns. The covariance matrix for each factor is assumed known, as is standard practice.

Each of these factor potentials assumes that the sensor observations are corrupted by zero-mean additive Gaussian noise. Though this assumed model is mathematically and computationally convenient, it has been shown to be invalid for certain applications in underwater robotics [51, 57]. We note, however, that our results do not exhibit characteristics often seen with force-fitting a Gaussian noise model to non-Gaussian data, such as inconsistent 3D reconstructions and high residual errors. We therefore conclude that, in our case, the Gaussian assumption is valid.

As suggested earlier, we also include extrinsic sensor parameters in the factor-graph formulation. In particular, we assume that the time-varying servo angle that actuates the HAUV's sensor tray is instrumented but uncertain. Similarly, the static servo-to-camera transform is also not precisely calibrated and treated as uncertain. Therefore, we denote these unknowns as $\theta_{v_i s_i}$ and $\mathbf{x}_{sc_T}$, respectively, and include them as variable nodes in our factor-graph implementation. These additional unknowns are illustrated in Fig. 9.

We used a graphical processing unit (GPU) implementation of the scale-invariant feature transform (SIFT) feature descriptor to detect visual features on the hull [41, 64]. We treat these features as landmarks in our BA pipeline described above. Putative correspondences were established using a standard appearance-based nearest neighbor search. This was implemented on a GPU in the interest of computational performance. Data association between these features and the prior model is simply a lookup to the nearest triangle in the prior model, as described in Section 2.3.

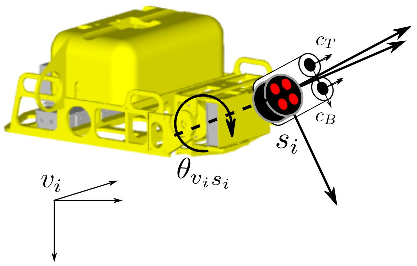


Figure 9: Illustration of the various reference frames at time i. The vehicle has a sensor tray that houses both the DVL and vertically-oriented stereo rig. An onboard servo rotates the servo frame, s_i, which in turn rotates the DVL and camera. The vehicle controls this angle so that these sensors point approximately orthogonal to the ship hull surface. This angle is instrumented, but must be treated as uncertain in our estimation framework due to the mechanical slop in the servo.

The prior model used in our application is a 3D triangular mesh that is derived from computer aided design (CAD) drawings of the ship hull being surveyed. This mesh can be read into memory and efficiently rendered on a consumer-grade laptop.

2.6.1. *Prior Factors*

A full-state prior on all six degrees of freedom for a particular variable node, $\mathbf{x}_{ij}$, is given by the conditional distribution of the measurement $\mathbf{z}^{\text{full}}_{\mathbf{x}_{ij}}$:

$$p\left(\mathbf{z}^{\text{full}}_{\mathbf{x}_{ij}} \middle| \mathbf{x}_{ij}\right) = \mathcal{N}\left(\mathbf{x}_{ij}, \Sigma_{\mathbf{z}^{\text{full}}_{\mathbf{x}_{ij}}}\right). \tag{8}$$

The initial guess for the pose of the prior model, $\mathbf{x}_{gm}$, is determined using the GICP algorithm for aligning two point clouds [54], as discussed in Section 2.5. This GICP alignment is added as a prior factor on $\mathbf{x}_{gm}$ with high variance ($\Sigma_{\mathbf{z}^{\text{full}}_{\mathbf{x}_{gm}}} = \mathrm{I}_{6\times 6}$).

The onboard depth and IMU sensors allow us to directly observe a bounded-error measurement of the vehicle's depth, pitch, and roll. This observation, denoted $\mathbf{z}^{\text{zpr}}_{\mathbf{x}_{ij}}$, has the following conditional distribution:

$$p\left(\mathbf{z}^{\text{zpr}}_{\mathbf{x}_{ij}} \middle| \mathbf{x}_{ij}\right) = \mathcal{N}\left(\left[z^{i}_{ij}, \phi_{ij}, \theta_{ij}\right]^{\top}, \Sigma_{\mathbf{z}^{\text{zpr}}_{\mathbf{x}_{ij}}}\right). \tag{9}$$

For this factor, we chose $\Sigma_{\mathbf{z}^{\text{zpr}}_{\mathbf{x}_{ij}}}$ to be diagonal corresponding to standard deviations of 0.1 m, 0.1°, and 0.1° for depth, pitch, and roll, respectively.

Finally, we model the servo angle at time i as an uncertain observation as discussed in Fig. 9. The corresponding observation model is simply:

$$p\left(z_{v_i s_i} \middle| \theta_{v_i s_i}\right) = \mathcal{N}\left(\theta_{v_i s_i}, \sigma_{z_{s_i}}^2\right), \tag{10}$$

where $\sigma_{z_{s_i}} = 5°$.

2.6.2. Odometry Factors

Our factor-graph formulation models odometry measurements as a sequential relative-pose observation, $\mathbf{z}_{i(i+1)}^{\text{odo}}$. The conditional distribution of this measurement is

$$p\left(\mathbf{z}_{i(i+1)}^{\text{odo}} \middle| \mathbf{x}_{gi}, \mathbf{x}_{g(i+1)}\right) = \mathcal{N}\left(\ominus\mathbf{x}_{gi} \oplus \mathbf{x}_{g(i+1)}, \Sigma_{\mathbf{z}_{i(i+1)}^{\text{odo}}}\right), \tag{11}$$

where $\oplus$ and $\ominus$ are pose composition operators following the conventions of Smith et al. [56]. We model the covariance matrix, $\Sigma_{\mathbf{z}_{i(i+1)}^{\text{odo}}}$, as a diagonal, where the entries' standard deviations are proportional to the time difference between i and $(i+1)$. The translational noise is proportional to a rate of $5\ \mathrm{mm}$ per second and the rotational noise is proportional to a rate of $80°$ per hour.

2.6.3. Stereo Camera Factors

The observed pixel locations at time i corresponding to the k^{th} feature are denoted as $\mathbf{z}_{ik}^T$ and $\mathbf{z}_{ik}^B$ for the top and bottom cameras, respectively:

$$p\left(\left[\mathbf{z}_{ik}^{T\,\top}\ \mathbf{z}_{ik}^{B\,\top}\right]^\top \middle| \mathbf{x}_{gv_i}, \mathbf{x}_{sc_T}, \theta_{v_i s_i}, \mathbf{l}_k\right) = \mathcal{N}\left(h_c\left(\mathbf{x}_{gv_i}, \mathbf{x}_{sc_T}, \theta_{v_i s_i}, \mathbf{l}_k\right), \sigma_c^2 \mathrm{I}_{4\times4}\right), \tag{12}$$

where we choose $\sigma_c = 2\ \mathrm{pixels}$. The observation model, h_c, corresponds to two pinhole cameras in a calibrated and rectified vertical stereo configuration (from Fig. 9):

$$h_c\left(\mathbf{x}_{gv_i}, \mathbf{x}_{sc_T}, \theta_{v_i s_i}, \mathbf{l}_k\right) = \begin{bmatrix} \mathrm{K}_{c_T}\left[{}^{c_T}_{g}\mathrm{R} \,\middle|\, {}^{c_T}\mathbf{t}_{c_T g}\right]\underline{\mathbf{l}_k} \\ \mathrm{K}_{c_B}\left[{}^{c_B}_{g}\mathrm{R} \,\middle|\, {}^{c_B}\mathbf{t}_{c_B g}\right]\underline{\mathbf{l}_k} \end{bmatrix}.$$

In general, $\left[{}^{j}_{i}\mathrm{R} \,|\, {}^{j}\mathbf{t}_{ji}\right]$ denotes the transformation of a point from frame i to frame j. In this case, it can represent the transformation of points from the global frame, g, to the top camera, c_T (i.e., the rotation and translation of the composed pose $(\mathbf{x}_{gv_i} \oplus \mathbf{x}_{v_i s_i} \oplus \mathbf{x}_{sc_T})$, where $\mathbf{x}_{v_i s_i} = [0, 0, 0, 0, \theta_{v_i s_i}, 0]^\top$). It also can describe the transformation of points in the global frame to the bottom camera, c_B (i.e., the rotation and translation of the composed pose $(\mathbf{x}_{gv_i} \oplus \mathbf{x}_{v_i s_i} \oplus \mathbf{x}_{sc_T} \oplus \mathbf{x}_{c_T c_B})$, where $\mathbf{x}_{c_T c_B}$ is the transformation from the top camera frame to the bottom camera frame). This transformation is taken from stereo camera calibration.

Note that our notation for $h_c\left(\,\cdot\,\right)$ omits the dehomogenization of each camera projection for the sake of clarity.

2.6.4. Monocular Camera Factors

In the event a stereo camera is not available, our framework can also use a calibrated monocular camera. Similar to the above stereo camera factor from (12), we have

$$p\left(\mathbf{z}^{c}_{ik} \,|\, \mathbf{x}_{gv_i}, \mathbf{x}_{sc_T}, \theta_{v_i s_i}, \mathbf{l}_k\right) = \mathcal{N}\left(\mathrm{K}\left[{}^{c}_{g}\mathrm{R} \,|\, {}^{c}\mathbf{t}_{cg}\right]\mathbf{l}_k, \sigma_c^2 \mathrm{I}_{2\times 2}\right). \tag{13}$$

Similar to (12), we omit the extra dehomogenization step for the sake of clarity.

2.6.5. DVL Raycast Factors

A critical component of our factor-graph formulation is factors modeling the intersection of DVL beams to the prior mesh—doing so allows us to significantly constrain both the unknown vehicle poses and servo angles. The conditional distribution takes the form:

$$p\left(z_{r_{in}} \,|\, \mathbf{x}_{gm}, \mathbf{x}_{gv_i}, \theta_{v_i s_i}\right) = \mathcal{N}\left(h_{rn}\left(\mathbf{x}_{gm}, \mathbf{x}_{gv_i}, \theta_{v_i s_i}; \mathcal{M}_{\text{prior}}\right), \sigma^2_{z_{r_{in}}}\right), \tag{14}$$

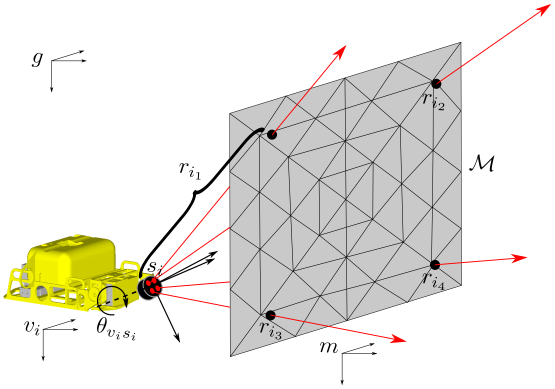


Figure 10: Illustration of ray-casting constraint. Given pose of the vehicle frame at time i, $\mathbf{x}_{gv_i}$, the servo angle, $\theta_{v_i s_i}$, the pose of the prior mesh frame, $\mathbf{x}_{gm}$, and the prior mesh, $\mathcal{M}_{\text{prior}}$, the four DVL range returns can be computed with an efficient octree-based ray-casting approach. At time i, the four ranges are predicted as r_{i_1}, r_{i_2}, r_{i_3}, and r_{i_4}.

where h_{rn} corresponds to raycasting the n^{th} beam for a DVL in a four-beam Janus configuration [9]. This observation model is illustrated in Fig. 10. Since the prior mesh may consist of hundreds of thousands of triangles, we use an efficient octree-based raycast implementation [53]. In addition, when evaluating the Gaussian distribution from (12) and (14) we apply a Huber M-estimator to the squared loss term to automatically reject outliers [31].

We assume that the uncertainty of DVL range returns is constant regardless of range because the HAUV maintains a fixed standoff when inspecting ship hulls. The standoff typically varies between 1 m and 2 m, and we have found that in this range the DVL is corrupted by relatively small noise. We therefore choose $\sigma_{z_{r_{in}}} = 3$ mm.

2.7. Frontend Details

The focus of this paper is on the optimization backend so we purposefully omit some details of the visual frontend that establishes feature correspondences between keyframes. However, the approach we use is quite standard: a random sample consensus (RANSAC)-based algorithm to reject outliers established during putative feature descriptor matching. In the case of a stereo camera, we use a standard triangulated point-cloud alignment technique to ensure inlier feature matches are consistent (in a Euclidean sense) under a 6-DOF rigid transformation [46, 5].

For a monocular camera, we fit an essential matrix and measure the RANSAC fitness score using Sampson (i.e., epipolar) distance. We opt for the essential matrix model because it generalizes to scenes that are not locally planar, unlike a plane-induced homography-based model [60, 38].

The features used in our BA pipeline are only SIFT visual features. We do not extract any features on the prior model because we assume the prior model is locally featureless. Indeed, for a CAD-derived model of a large ship hull, we find that small-scale manmade features such as weld lines, intake ports, frame numbers, etc. are absent.

In our case, data association between visual features is assigned each iteration by performing a lookup to the nearest triangle in the prior model, as described in Section 2.3. This depends on a good initial guess of the alignment between the SLAM coordinate frame and the prior model's coordinate frame. As discussed in Section 2.6.1, we derive the initial guess by performing a one-time alignment (using GICP) between the DVL-derived range returns and the vertices of the CAD.

3. Model-assisted Visual Mapping

3.1. Notation

In addition to the notation introduced in Section 2.1, we will let $\mathcal{M}_{\text{new}}$ denote the updated mesh (i.e., the fusion of camera-derived structure with the prior model, $\mathcal{M}_{\text{prior}}$). This is the final output of our algorithm.

3.2. Identifying Shapes by Clustering Features

Once we estimate all unknowns in the factor-graph formulation from Section 2, we can easily compute the structural deviation from the prior model using the formula for signed distance that is provided in (5).

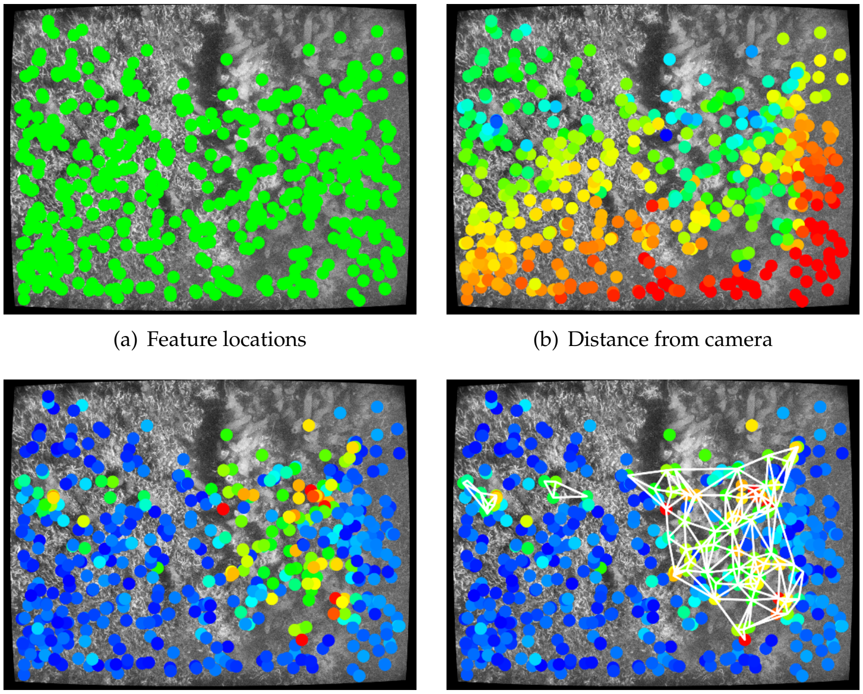
Algorithm 1 Detect shapes at a given camera pose at time i

```
Require: Camera pose (pose$_i$), visible features ($\mathcal{F}_{vi}$), and mesh ($\mathcal{M}_{\text{prior}}$)
$\mathcal{P}_i = \emptyset$                                   //Set of points to cluster.
$\mathcal{C}_i = \emptyset$                                   //Set of clusters from DBSCAN.
$\mathcal{S}_i = \emptyset$                                   //Set of detected shapes.
for feature $\mathbf{l} = [l_x, l_y, l_z]$ in $\mathcal{F}_{vi}$, expressed in global frame do
    $d = |h(\mathbf{x}_{g\mathcal{M}}, \mathbf{l})|$                        //See Eqn. (5).
    if $d > \tau$ then
        $[f_x\ f_y\ f_z]^\top = \mathrm{H}^{g}_{c_T}\underline{\mathbf{l}}$             //Transform to camera frame
        $\mathcal{P}_i = \mathcal{P}_i \cup \{f_x, f_y, d\}$
    end if
end for
$\mathcal{C}_i$ = DBSCAN($\mathcal{P}_i$)
for cluster in $\mathcal{C}_i$ do
    $\mathcal{M}_{S_i}$ = alpha_shape(cluster)        //2.5D triangulation over $f_x$, $f_y$, and $f_z$
    $\mathcal{S}_i = \mathcal{S}_i \cup \mathcal{M}_{S_i}$
end for
return $\mathcal{S}_i$
```

For this section, we adapt the notion of a bundle adjusted feature's *structural deviation*. This is simply the orthogonal signed distance of each feature; these values were computed during BA. Using these feature deviations, we apply density-based spatial clustering of applications with noise (DBSCAN) to nonlinearly separate the features' positions (as expressed in each camera's coordinate frame) into clusters [21]. In short, DBSCAN visits each point in the dataset, queries its neighbors using a distance metric, and assigns the neighbors to a new cluster. This process is repeated until all points are visited.

Note that in general DBSCAN clusters a dataset consisting of column vectors. In our case, the dataset consists of 3D points corresponding to the positions of visual features as expressed in the camera coordinate frame.

These clusters of points are converted to shapes using a simple extension to Delaunay triangulation known as *alpha-shapes* [20]. This algorithm is summarized in Algorithm 1, with an accompanying example in Fig. 11.



(a) Feature locations

(b) Distance from camera

(c) Deviation to $\mathcal{M}_{\text{prior}}$

(d) Alpha-shaped clusters

Figure 11: Visual overview of Algorithm 1. For a particular keyframe, the bundle adjusted features, (a) and (b), are assigned a deviation by computing the intersection of the ray to the prior model (c). These values are clustered using DBSCAN, and meshed using alpha-shapes. In (d), the three detected clusters have their alpha-shapes shown as white triangular meshes.

As shown in Fig. 11(d), the detected alpha-shapes can be projected as two-dimensional triangular meshes in the camera imagery. The choice of "alpha" in determining these shapes is quite application-dependent [20]. We use a fixed value of 12 cm for all of our experiments, but we suspect this value should be tuned depending on the application.

We note that line 13 of Algorithm 1 computes alpha-shapes using 2D Delaunay triangulation on the points' x, y coordinates as expressed in the camera frame. Each vertex in the 2D triangulation is then assigned the camera-relative z value, lifting the detected shape a 2.5D triangular mesh (when expressed in the camera frame). The use of 2.5D shapes as opposed to fully 3D shapes effectively prevents triangles from occluding each other. In addition, these shapes are determined for every camera view, and therefore a single physical object on the hull will have multiple shapes associated with it. In Algorithm 2, these shapes are combined across multiple views and fused into $\mathcal{M}_{\text{prior}}$.

Algorithm 2 Fuse shapes into prior mesh

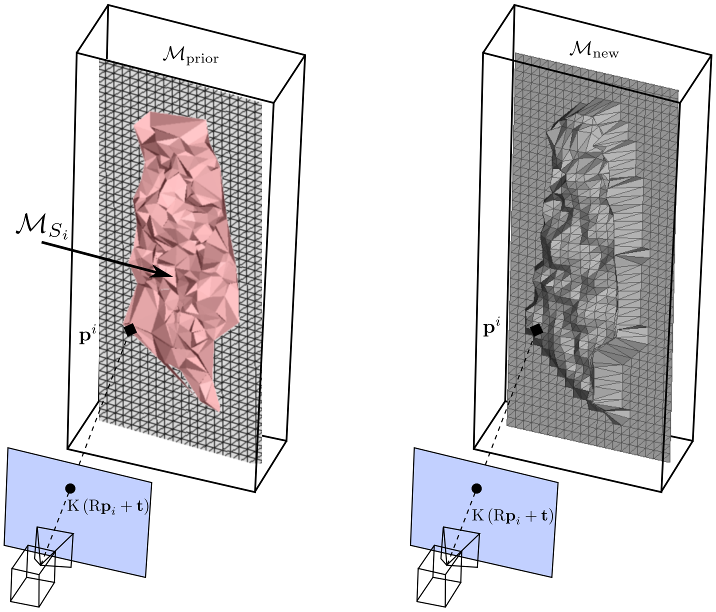
```
$\mathcal{M}_{\text{new}} = \mathcal{M}_{\text{prior}}$                    //Make a deep copy of the prior mesh
for pose in poses do
    $\mathcal{F}_v$ = is_visible(pose, $\mathcal{F}$)                           //Visible features
    $\mathcal{S}_d$ = Algorithm1(pose, $\mathcal{F}_v$, $\mathcal{M}_{\text{prior}}$)                //Detected shapes
    $\mathcal{V}_n$ = nearby_vertices(pose, vertices($\mathcal{M}_{\text{prior}}$))
    for shape $\mathcal{M}_s$ in $\mathcal{S}_d$ do
        for vertex $\mathbf{v}^i \in \mathcal{V}_n$ indexed by $i$ do
            ray = make_ray(pose, $\mathcal{V}_n[i]$)
            if ray.intersects_with($\mathcal{M}_s$) then
                $\mathbf{p}^i$ = ray.intersection($\mathcal{M}_s$)
                moving_avg($\mathcal{M}_{\text{new}}$, $i$, $\mathbf{p}^i$)                      //Using Eqn. (15)
            end if
        end for
    end for
end for
return $\mathcal{M}_{\text{new}}$
```

3.3. Model Remeshing Step

The final step of our approach is to fuse the shapes detected in Algorithm 1 with the prior mesh, $\mathcal{M}_{\text{prior}}$, resulting in a new mesh, $\mathcal{M}_{\text{new}}$. To this end, we compute a ray originating from the top camera's center (c_T in Fig. 9) and extending toward a vertex in $\mathcal{M}_{\text{prior}}$. Like the DVL range observation model from (14), we use a raycasting approach to compute the intersection with any detected shapes. Once the intersection point corresponding to the i^{th} vertex, $\mathbf{p}^i$, is calculated, we update the corresponding vertex in $\mathcal{M}_{\text{new}}$, $\widehat{\mathbf{v}}^i$, with a recursive moving average filter:

$$\widehat{\mathbf{v}}^i_{n+1} = \frac{\mathbf{p}^i + n\widehat{\mathbf{v}}^i_n}{n+1} \tag{15}$$

$$\widehat{\mathbf{v}}^i_0 = \texttt{get\_ith\_vertex}(\mathcal{M}_{\text{prior}}, i),$$


$\mathcal{M}_{\text{prior}}$
$\mathcal{M}_{S_i}$
$\mathbf{p}^i$
$\mathrm{K}\left(\mathrm{R}\mathbf{p}_i + \mathbf{t}\right)$
$\mathcal{M}_{\text{new}}$
$\mathbf{p}^i$
$\mathrm{K}\left(\mathrm{R}\mathbf{p}_i + \mathbf{t}\right)$

(a) DBSCAN mesh overlayed on prior mesh

(b) Augmented CAD mesh

Figure 12: Visualization of Algorithm 2. A 3D shape, $\mathcal{S}$, derived from Algorithm 1 can be fused into the prior mesh by intersecting rays from the camera frame. For a particular camera pose, we choose a mesh vertex, $\mathcal{V}_n[i]$, and then compute the intersection of the camera-to-vertex ray as $\mathbf{p}^i$. The various intersections at different camera poses are fused into a new mesh, $\mathcal{M}_{\text{new}}$, using a moving-average filter.

where the i^{th} vertex's counter, n, is incremented after every evaluation of line 11 from Algorithm 2. Note that because the shapes are all 2.5D when expressed in the camera frame, *all* rays from the camera center to the vertices will intersect the detected shapes no more than once.

This process is repeated for every pose. A summary of this algorithm is provided in Algorithm 2 along with an accompanying visualization in Fig. 12. The function `nearby_vertices()` simply returns a set of vertices that are some fixed distance from the camera pose (we conservatively use $5\ \mathrm{m}$). Because the number of vertices in the prior mesh is very large, we use a k-dimensional (KD)

Ship Length	183 m
Ship Beam	27 m
Ship Draft	9.1 m

Table 2: Size characteristics of the *SS Curtiss*.

tree to make this search computationally efficient. The function `is_visible()` returns the set of visual features that are contained in the camera's frustum. Note that the vertices are only updated if a line segment between the camera center and the vertex intersects with a detected shape (line 9 of Algorithm 2).

4. Results

The field data used in our experimental evaluation is taken from the Bluefin Robotics HAUV surveying the *SS Curtiss*, shown in Fig. 13. A 3D triangular mesh, shown previously in Fig. 1 and Fig. 2, was derived from computer aided design (CAD) drawings, and serves as the prior model in our model-assisted framework. In this section, we evaluate the performance of three approaches when processing this single large dataset:

1. A naive BA framework where the measurements consists of $\mathcal{Z}_{\text{prior}}$, $\mathcal{Z}_{\text{odo}}$, $\mathcal{Z}_{\text{range}}$, and $\mathcal{Z}_{\text{feat}}$. All surface constraints are disabled, i.e., $\lambda_i = 0$ for every feature.

2. The approach based on Geva et al. [26], which consists of the measurements $\mathcal{Z}_{\text{prior}}$, $\mathcal{Z}_{\text{odo}}$, $\mathcal{Z}_{\text{range}}$, and $\mathcal{Z}_{\text{feat}}$, in addition to surface constraints, $\mathcal{Z}_{\text{surf}}$, such that $\lambda_i = 1$ for every feature.

3. The proposed algorithm discussed in Section 2.2, implemented using Gaussian max-mixtures, where each hidden label λ_i is assigned from (7).

The size of the bundle adjustment problem is shown in Table 2.

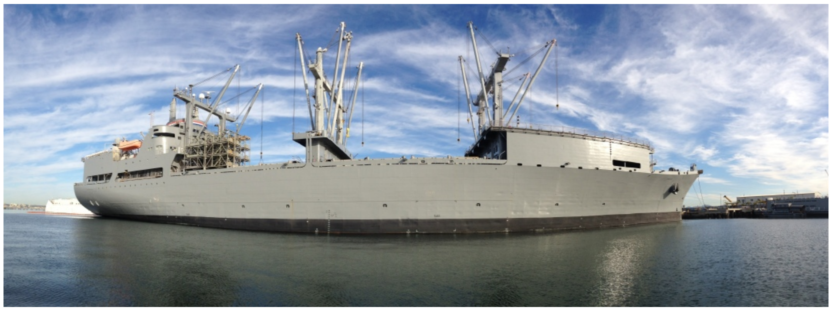
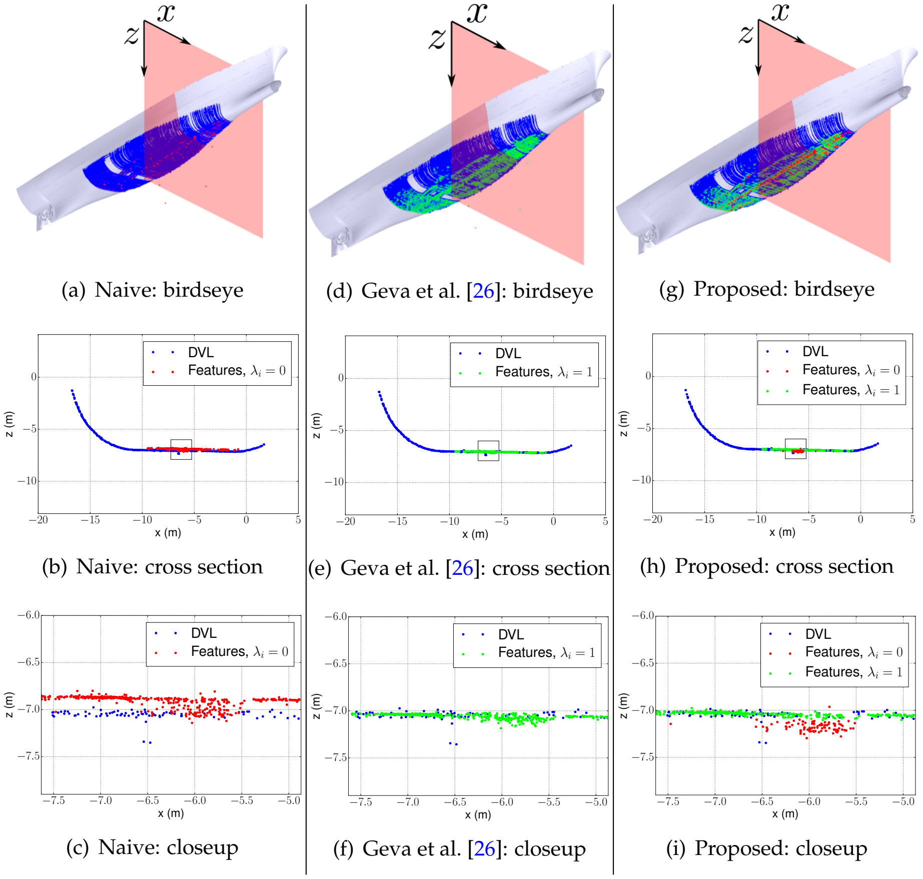


Figure 13: The vessel being surveyed is the *SS Curtiss*, for which we have access to a CAD-derived 3D mesh. The size of the dataset used in this paper is summarized in Table 2.

4.1. Model-assisted Bundle Adjustment

We provide a visualization of the reconstruction that highlights the advantages of our approach in Fig. 14 and Fig. 15. These plots show cross sections of the ship hull, from starboard-to-port, to highlight the relevant portions of the visual features and range returns from the DVL. Because of the raycasting factors from (14), these range returns act as a proxy for the true profile of the hull form. We achieve this as follows: in the top rows of Fig. 14 and Fig. 15, we plot a narrow cross-sectional 3D rectangle in pink. In the second row, we only plot the DVL and visual features that are contained within this pink rectangle. Finally, the third row shows a zoomed-in portion of the second row to highlight small-scale details.

Fig. 14 shows the reconstruction using 2014 field data, which uses an underwater stereo camera. The portion of the ship hull that is captured in the bottom row of Fig. 14 corresponds to the biofouling detection shown in Fig. 1. Fig. 15 is taken from a 2011 survey, during which the HAUV was equipped with a monocular underwater camera. This scene corresponds to a cylindrical shape with known dimensions. This shape will be shown later in Fig. 22. The factor-graph representations (discussed in Section 2.6) between the stereo and monocular datasets are quite similar, except that the reprojection error is computed using (12) in the case of a stereo camera, and (13) in the case of a monocular camera.



(a) Naive: birdseye

(d) Geva et al. [26]: birdseye

(g) Proposed: birdseye

(b) Naive: cross section

(e) Geva et al. [26]: cross section

(h) Proposed: cross section

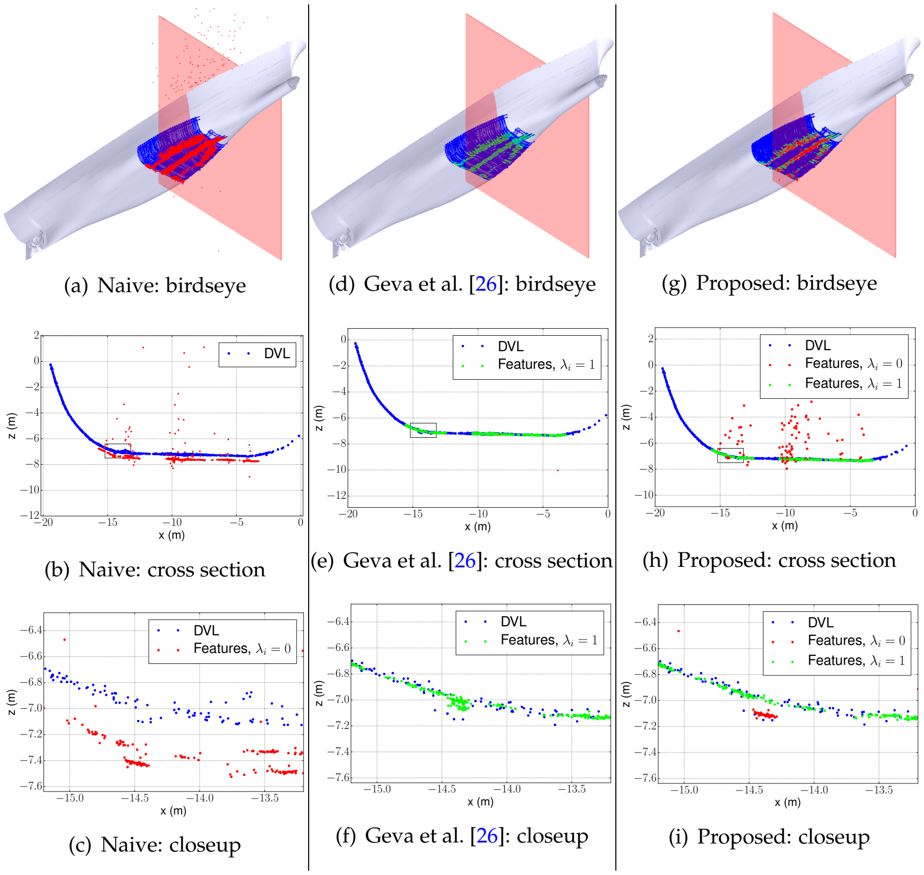
(c) Naive: closeup

(f) Geva et al. [26]: closeup

(i) Proposed: closeup

Figure 14: Visual inspection from 2014 using a stereo camera. Each column represents a different method, with each column showing the same representative cross section of the reconstruction. For the naive approach shown in (a) through (c), there are no factors constraining the visual features and prior model, resulting in a noticeable misregistration with the DVL. Using the method from Geva et al. [26], shown in (d) through (f), the features and DVL ranges are well-registered, but the biofouling is visibly "squished" onto the surface. Our method, shown in (g) through (i), combines the favorable qualities of each method, aligning the visual features and DVL returns while also preserving 3D structure that is absent from the prior model (i.e., the red points in (i)).

In these figures, we see some general trends. (i) In the reconstruction derived from the naive approach, the visual features do not lie on the same surface as the range returns from the DVL. The features are underconstrained in the naive case because there is zero information (i.e., connected factors) between the pose of the prior model and the position of the visual features. (ii) Using the approach from Geva et al. [26], the visual features lie on the same surface as the



(a) Naive: birdseye

(d) Geva et al. [26]: birdseye

(g) Proposed: birdseye

(b) Naive: cross section

(e) Geva et al. [26]: cross section

(h) Proposed: cross section

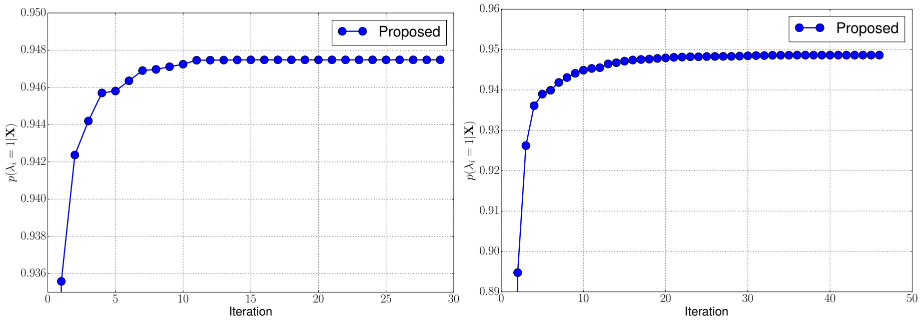
(c) Naive: closeup

(f) Geva et al. [26]: closeup

(i) Proposed: closeup

Figure 15: Visual inspection from 2011 using a monocular camera. The general trends in this result are nearly identical to the trends shown in Fig. 14. In this case, the known cylindrical shape's 3D structure is preserved in our method (shown in (i)), while it is *not* well-preserved using the approach from Geva et al. [26] (shown in (f)).

DVL range returns, as expected. Because *all* features are constrained to lie on the surface, the algorithm does not capture 3D structure present on the actual ship hull that is not present in the prior model (e.g., the docking blocks along the bottom of the hull, or manually-placed cylindrical shapes). (iii) Our approach combines the benefits of both approaches: the visual features and DVL-derived point cloud lie on the same surface, and our visual reconstruction yields 3D structure that would have been heavily regularized using the approach from Geva et al. [26].



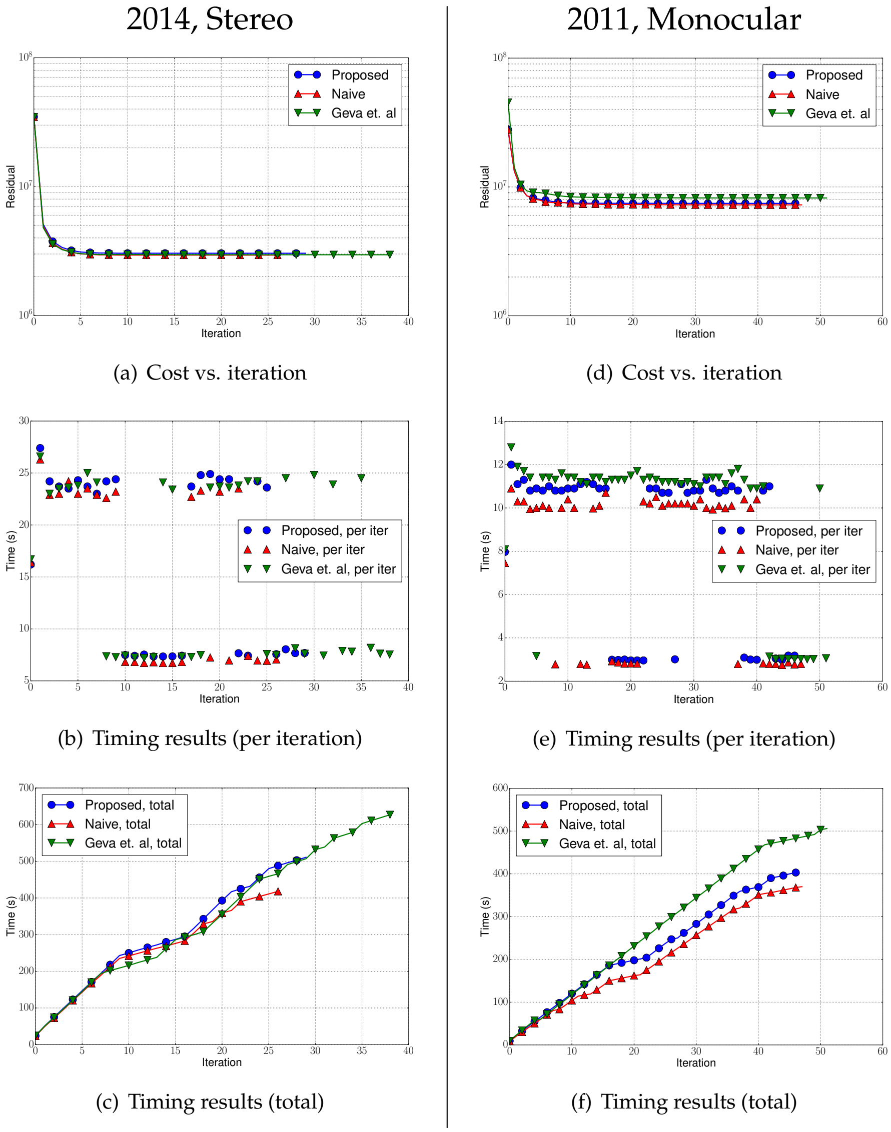
(a) 2014 (stereo)

(b) 2011 (monocular)

Figure 16: Here we show the relative frequency of all features with $\lambda_i = 1$ as a function of the current least-squares iteration. The left and right plots corresponds to the right columns of Fig. 14 and Fig. 15, respectively. Note the difference in scale for each plots y-axis, and that the first several datapoints were omitted to highlight the subtle changes in subsequent iterations. The initial probabilities for (a) and (b) are 0.82 and 0.34, respectively.

In addition, there exists several outlier features in the sparse monocular reconstruction compared to the stereo reconstruction, as shown in Fig. 14(h) and Fig. 15(h). This is not surprising; the geometric verification step is relatively weak for monocular cameras as compared to stereo (as briefly mentioned in Section 2.7). This has little to no effect on our model-assisted BA framework; because they are outliers (i.e., they lie far from the surface of the prior mesh) they impose very little cost when evaluating the surface constraint likelihood from (4).

The results in Fig. 16 suggest the identification of feature labels stabilizes after about twenty iterations. Clearly, for this application, the vast majority of features lie on the prior model, suggesting that the weights in each mixture (or, equivalently, the last multiplicand in (1)) can be optionally tuned to reflect this trend, rather than assigning an uninformative prior probability for $p(\lambda_i = 1)$. However, we prefer the latter approach because it is more appropriate for conservatively detecting foreign objects (an important capability for automated ship hull inspection).



(a) Cost vs. iteration

(b) Timing results (per iteration)

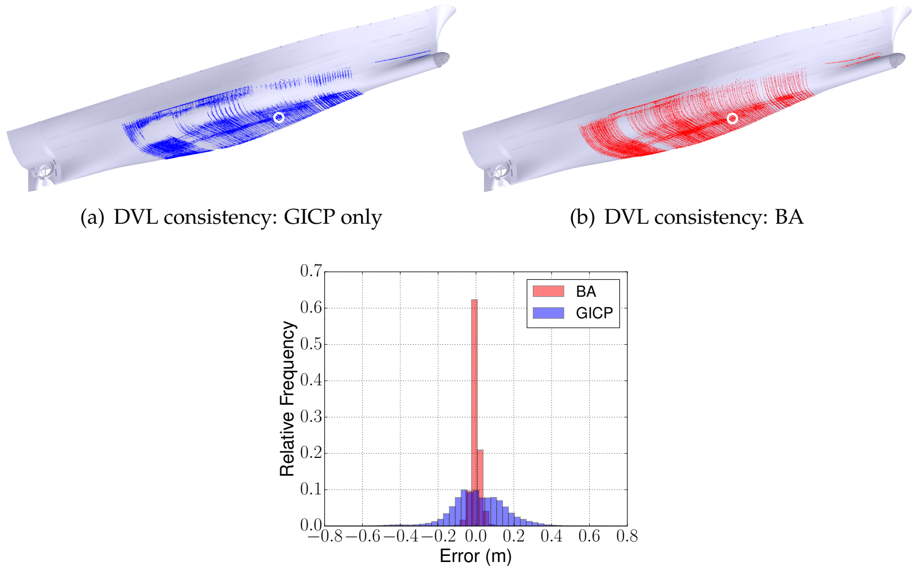
(c) Timing results (total)

(d) Cost vs. iteration

(e) Timing results (per iteration)

(f) Timing results (total)

Figure 17: Cost and timing comparison for each method illustrated in Fig. 14 (left) and Fig. 15 (right). Our algorithm imposes some extra computation time compared to the naive approach, but the overall execution time is comparable. For the stereo dataset, the naive, Geva et al. [26], and proposed approaches converged in 26, 38, and 29 iterations, respectively. For the monocular dataset, each approach converged in 47, 51, and 46 iterations, respectively.



(a) DVL consistency: GICP only

(b) DVL consistency: BA

(c) DVL beam residuals: (a) vs (b)

Figure 18: These figures show that the distribution of residual error for DVL range returns is relatively large unless using our model-assisted BA framework. Here, *residual* refers to the difference between observed range and the range predicted by the observation model, h_{rn}, provided in (14).

We assessed the computational performance by performing timed trials on a consumer-grade four-core 3.30 GHz processor. These results were obtained using the freely-available Ceres nonlinear least-squares solver. In each iteration of Levenburg-Marquardt, we used sparse Cholesky decomposition to solve the normal equations [3].

We report the timing results of each of the three methods in Fig. 17. From this plot we draw two conclusions: (i) the computational costs of our method impose total performance loss of 22.3% (stereo) and 8.9% (monocular) compared to the naive approach; (ii) the computational costs of the approach from Geva et al. [26] imposes a performance loss of 50.0% (stereo) and 36.8% (monocular) compared to the naive approach. This behavior can be explained by the optimizer having to perform more iterations until convergence for the approach from Geva et al. [26]. Intuitively, by forcing visual features that protrude from the prior model to lie flush, the optimizer must perform more iterations to satisfy the corresponding reprojection errors. Even though our method devotes additional processing time when evaluating (7), this is overcome by the added cost of performing additional iterations.

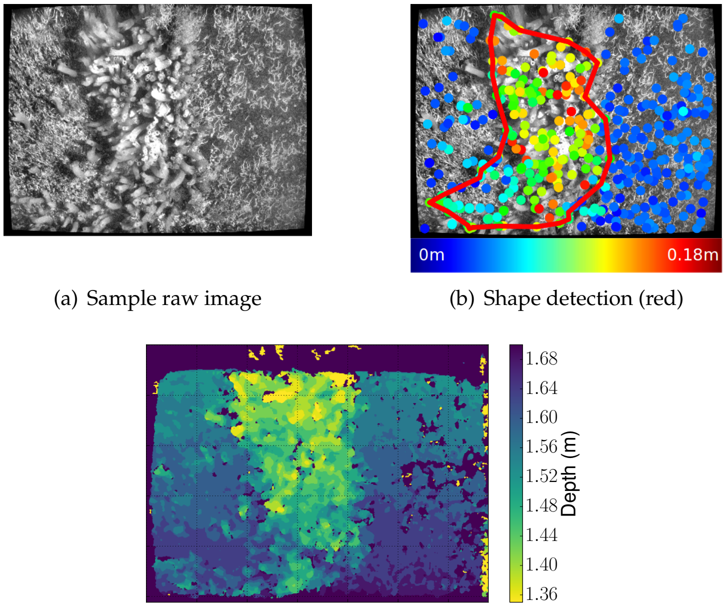
Finally, if we examine the consistency of DVL range measurements, we can see a noticeable improvement using our model-assisted BA framework. From Fig. 18(a) and Fig. 18(b), we can see several inconsistencies particularly on the side of the hull. By examining the distribution of error when evaluating (14) (shown in in Fig. 18(c)), we quantitatively confirm that the alignment taken from GICP alone yields relatively large error distribution of DVL returns, however our model-based BA framework can significantly tighten the distributions of these residuals. The main source of error is the slop in the servo that rotates the DVL. By solving for this angle, we can have a much tighter distribution of error, as shown in Fig. 18(c).

4.2. *Model-assisted Mapping Results*

Our evaluation of our model-assisted mapping pipeline consists of a metric evaluation of the structures detected from Algorithm 1 and the utility of the remeshed prior model from Algorithm 2. These will be discussed in Section 4.2.1 and Section 4.2.2, respectively.

4.2.1. *Shape Detection and 3D Accuracy*

Illustrative examples of our shape detection approach from Algorithm 1 are shown in Fig. 19 (stereo) and Fig. 20 (monocular). The former shows biofouling emanating from a docking block, while the later shows a manually-placed cylindrical shape. These examples correspond to the cross-sections shown in Fig. 14 and Fig. 15. The features contained in the shape detected in Fig. 19(b) have deviations ranging between $0.06\ \mathrm{m}$ and $0.18\ \mathrm{m}$. Though we do not have ground-truth, we can use a dense stereo matching algorithm [28] to get a rough sense of how much the biofouling protrudes compared to the rest of the scene.



(a) Sample raw image

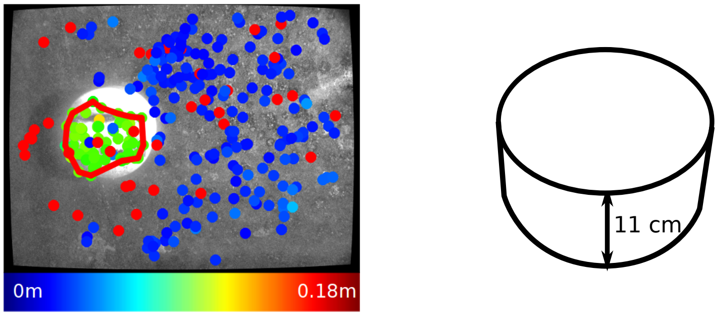
(b) Shape detection (red)

(c) Camera-relative depth (dense stereo)

Figure 19: Shape detection example from the 2014 stereo dataset. In this example, the foreign object corresponds to biofouling along the ship hull centerline that is captured in the bottom row of Fig. 14. The dimensions of this object, as derived from dense stereo matching, agree closely to the proposed method's reconstruction from Fig. 14(i).

In Fig. 19(c), we see that the biofouling is about 0.10 m to 0.20 m closer to the camera than the rest of the scene. Though this comparison is relatively coarse, this serves as a promising indication that the 3D structure inferred using our approach is reasonably correct.

The monocular example we show in Fig. 20(a) indicates a cluster of features centered on the cylindrical shape with a mean deviation of 0.10 m to the CAD model. In this case we know that the ground-truth height of the cylindrical shape is 0.11 m, which suggests a reconstruction error of approximately 0.01 m. In addition, in this figure there are several outlying features (their corresponding red colors were capped at 0.18 m). Because DBSCAN requires a minimum number of datapoints per cluster (in our case, we choose 3), these features are not detected as foreign shapes.



(a) Shape detection (red)

(b) Ground-truth height

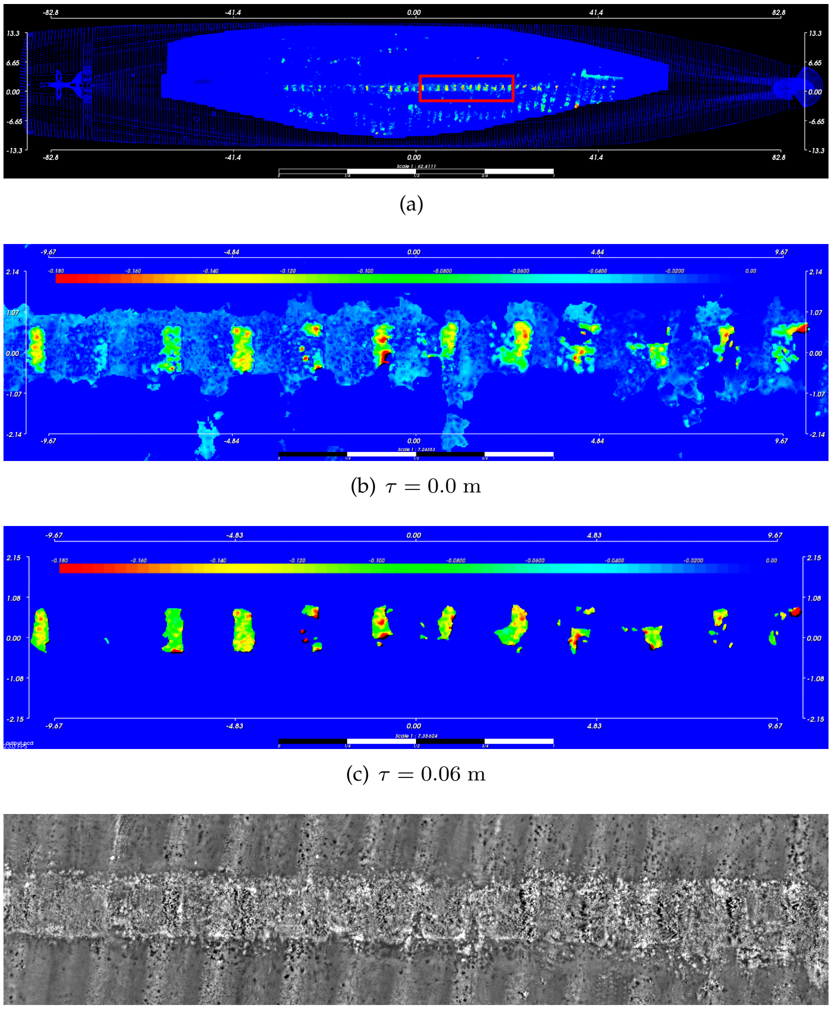
Figure 20: Shape detection from the 2011 monocular dataset. In this example, this object corresponds to the cylindrical shape that is captured in the bottom row of Fig. 15. There is approximately 1 cm of error between the height of the detected 3D structure (i.e., green cluster in (a)) and the ground-truth height of the cylindrical shape in (b). In addition, the dimensions of this object, as derived from ground-truth, agree closely to the proposed method's reconstruction from Fig. 15(i).

Note that Fig. 19 and Fig. 20 act as evidence that our shape detection algorithm is general enough to handle both monocular and stereo sensor configurations, but they should not be compared against each other. A thorough comparison between stereo and monocular reconstructions using BA can be found in in [58].

4.2.2. Remeshing Results

Algorithm 2 provides us with a remeshed CAD model for a visual survey of the *SS Curtiss*, which is shown in Fig. 21. We present results for two different values of τ, the threshold used for determining the eligibility of features to be clustered with DBSCAN. For this application, the preferred approach is to keep the threshold zero (Fig. 21(b)), however for certain applications where false positives are a concern, this can be raised (Fig. 21(c)). Like the other visualizations for this dataset, rectangular-like foreign 3D structures shown in Fig. 21(c) correspond to biofouling along the centerline.

In addition to providing visually intuitive false color maps, the remeshed model can easily be used in a state-of-the-art 3D photomosaicing framework [34]. An example of this application is provided in Fig. 23. Unlike our previous work in [48] that applied texture to the ship's CAD model, this work allows additional



(a)

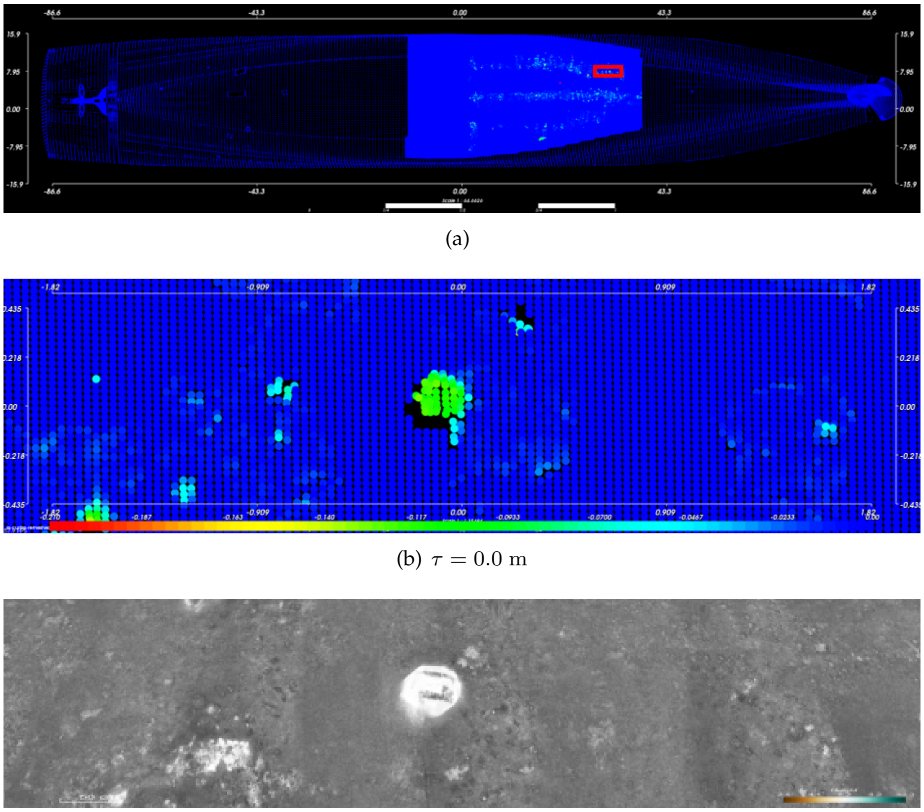
(b) $\tau = 0.0$ m

(c) $\tau = 0.06$ m

(d) Portion of photomosaic corresponding to (b) and (c)

Figure 21: Results of Algorithm 1 and Algorithm 2. In (a), we show a heatmap of the remeshed CAD vertices for the 2014 stereo dataset. The red region is expanded in (b) and (c). In (b), the clustering threshold from line 6 of Algorithm 1, τ, is zero while in (c) it is relatively high. Lowering τ provides more details but potentially introduces false positives. We can see the red rectangular region from (a) corresponds to a strip of biofouling at the ship's centerline, shown in (d).

structural details at a relatively small scale. In Fig. 23, we shade the regions of the 3D photomosaic according to height in the z-direction. Clearly, the approach proposed in this paper captures significantly more information that is otherwise discarded if the ship hull is assumed to match the CAD model shape exactly.

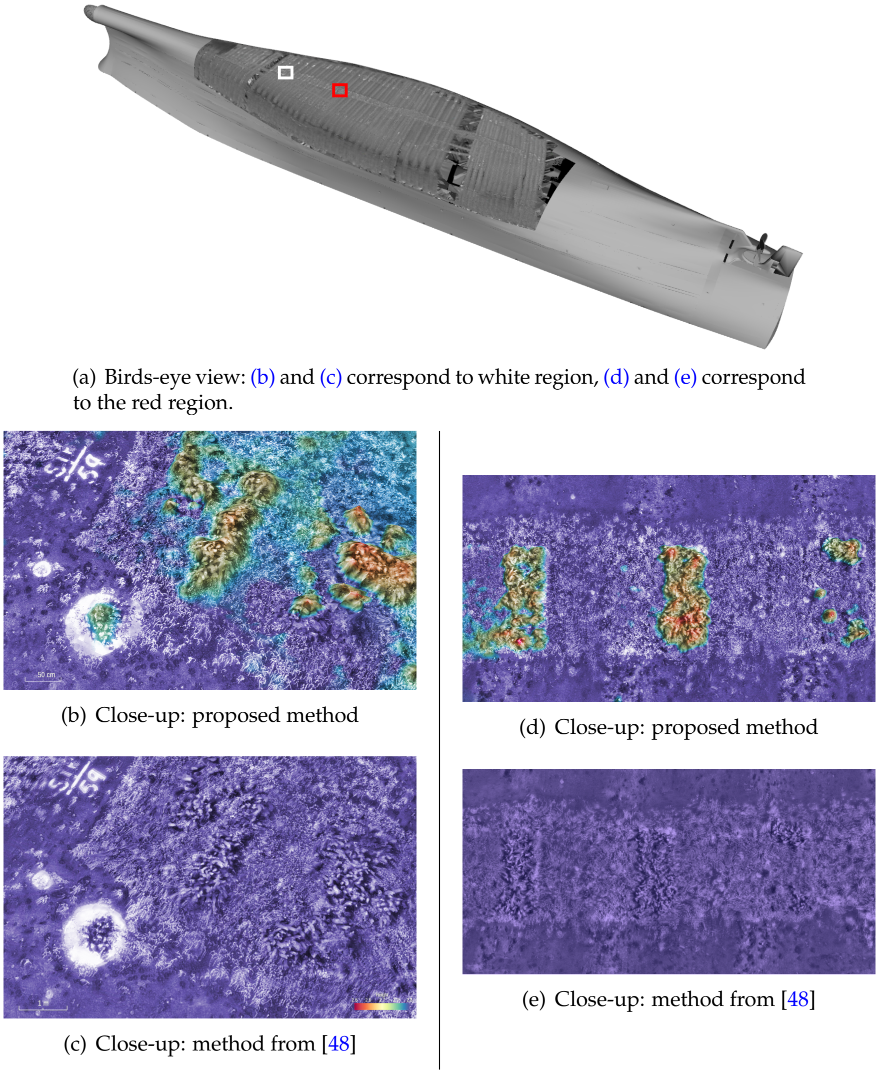


(a)

(b) $\tau = 0.0$ m

(c) Portion of photomosaic corresponding to (b)

Figure 22: Remeshed CAD vertices shown as a heatmap for the 2011 monocular dataset. The red region in (a) is expanded in (b), were we show the 3D structure corresponding to the cylindrical shape shown in Fig. 20. In (c) we show the corresponding region in 3D photomosac.



(a) Birds-eye view: (b) and (c) correspond to white region, (d) and (e) correspond to the red region.

(b) Close-up: proposed method

(c) Close-up: method from [48]

(d) Close-up: proposed method

(e) Close-up: method from [48]

Figure 23: Application to large-scale 3D photomosaicing. Our approach allows 3D photomosaicing approaches to combine large-scale consistency in (a) with small-scale detail in (b) and (d). In (b) and (d), the mosaic is shaded according to height. Using the approach from [48], where a CAD model is used for photomosaicing, the small-scale details are lost as evidenced by the regions in (c) and (e) being near-perfectly flat.

5. Conclusion

We proposed a model-assisted bundle adjustment framework that assigns binary labels to each visual feature. Using an EM algorithm with hard hidden label assignments, we iteratively update these labels and refine the current state estimate. We show that this algorithm is a special case of the Gaussian max-mixtures framework from earlier work in robust pose graph optimization. We compared our approach to recent work in model-assisted methods, and showed our algorithm has favorable properties when evaluated in the context of autonomous ship hull inspection.

In addition, we propose a shape identification and mapping algorithm that provides precise capabilities for identifying visually-observed 3D structure that is absent from the CAD model. The mapping algorithm fuses these shapes into the prior mesh, resulting in a newly remeshed model. This newly remeshed model has several important benefits for data visualization. In particular, the false-color figures shown in this paper offer an intuitive visualization that is harder to discern from image mosaics alone. In addition, the remeshed model can easily be used in a 3D photomosaicing framework such that the overall consistency of the ship hull reconstruction is preserved, but captures details at a small scale.

We evaluated these techniques using field data collected from the HAUV hull inspection robot—the largest model-assisted BA evaluation to date. We have shown that our BA framework introduces only slight computational overhead, while producing accurate reconstructions for both stereo and monocular camera sensors.

Acknowledgments

This work was supported in part by the Office of Naval Research under award N00014-12-1-0092, and in part by the American Bureau of Shipping under award number N016970-UM-RCMOP.

References

[1] P. Agarwal, G. D. Tipaldi, L. Spinello, C. Stachniss, and W. Burgard. Robust map optimization using dynamic covariance scaling. In *Proceedings of the IEEE International Conference on Robotics and Automation*, Karlsruhe, Germany, May 2013.

[2] S. Agarwal, N. Snavely, I. Simon, S. M. Seitz, and R. Szeliski. Building Rome in a day. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 72–79, Kyoto, Japan, Oct. 2009.

[3] S. Agarwal, K. Mierle, and Others. Ceres solver. `http://ceres-solver.org`, 2014.

[4] F. Aguirre, J. Boucher, and J. Jacq. Underwater navigation by video sequence analysis. In *Proceedings of the International Conference Pattern Recognition*, volume 2, pages 537–539, Atlantic City, NJ, USA, June 1990.

[5] K. S. Arun, T. S. Huang, and S. D. Blostein. Least-squares fitting of two 3-D point sets. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 9 (5):698–700, 1987.

[6] E. Belcher, B. Matsuyama, and G. Trimble. Object identification with acoustic lenses. In *Proceedings of the IEEE/MTS OCEANS Conference and Exhibition*, pages 6–11, Kona, HI, USA, Sept. 2001.

[7] F. Bonin, A. Burguera, and G. Oliver. Imaging systems for advanced underwater vehicles. *Journal of Maritime Research*, 8:65–86, 2011.

[8] F. Bonnín-Pascual and A. Ortiz. Detection of cracks and corrosion for automated vessels visual inspection. In *Proceedings of the International Conference of the Catalan Association for Artificial Intelligence*, pages 111–120, Tarragona, Spain, Oct. 2010.

[9] N. Brokloff. Matrix algorithm for Doppler sonar navigation. In *Proceedings of the IEEE/MTS OCEANS Conference and Exhibition*, volume 3, pages 378–383, Brest, France, Sept. 1994.

[10] M. Bryson, M. Johnson-Roberson, O. Pizarro, and S. Williams. Colour-consistent structure-from-motion models using underwater imagery. In *Proceedings of the Robotics: Science & Systems Conference*, Sydney, Australia, July 2012.

[11] R. Campos, R. Garcia, P. Alliez, and M. Yvinec. A surface reconstruction method for in-detail underwater 3D optical mapping. *International Journal of Robotics Research*, 34(1):64–89, 2015.

[12] S. M. Chaves, A. Kim, and R. M. Eustice. Opportunistic sampling-based planning for active visual SLAM. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3073–3080, Chicago, IL, USA, Sept. 2014.

[13] Y. Chen and G. Medioni. Object modeling by registration of multiple range images. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 2724–2729, Nice, France, May 1991.

[14] A. K. R. Chowdhury and R. Chellappa. Face reconstruction from monocular video using uncertainty analysis and a generic model. *Computer Vision and Image Understanding*, 91(12):188–213, 2003.

[15] P. Corke, R. Paul, W. Churchill, and P. Newman. Dealing with shadows: Capturing intrinsic scene appearance for image-based outdoor localisation. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2085–2092, Tokyo, Japan, Nov. 2013.

[16] B. Curless and M. Levoy. A volumetric method for building complex models from range images. In *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques*, pages 303–312, New Orleans, LA, Aug. 1996. ACM.

[17] S. Daftry, C. Hoppe, and H. Bischof. Building with drones: Accurate 3D facade reconstruction using MAVs. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 3487–3494, Seattle, WA, USA, May 2015.

[18] F. Dellaert and M. Kaess. Square root SAM: simultaneous localization and mapping via square root information smoothing. *International Journal of Robotics Research*, 25(12):1181–1203, 2006.

[19] S. Duntley. Light in the sea. *Journal of the Optical Society of America*, 53(2): 214–233, Feb. 1963.

[20] H. Edelsbrunner and E. P. Mücke. Three-dimensional alpha shapes. *ACM Transactions on Graphics*, 13(1):43–72, 1994.

[21] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the International Conference on Knowledge Discovery and Data Mining*, pages 226–231, Portland, OR, USA, 1996.

[22] T. G. Farr, P. A. Rosen, E. Caro, R. Crippen, R. Duren, S. Hensley, M. Kobrick, M. Paller, E. Rodriguez, L. Roth, et al. The shuttle radar topography mission. *Reviews of Geophysics*, 45(2), 2007.

[23] D. Fidaleo and G. Medioni. Model-assisted 3D face reconstruction from video. In *Proceedings of the IEEE Workshop on Analysis and Modeling of Faces and Gestures*, pages 124–138, Rio de Janero, Brazil, Oct. 2007.

[24] P. Fua. Using model-driven bundle-adjustment to model heads from raw video sequences. In *Proceedings of the IEEE International Conference on Computer Vision*, volume 1, pages 46–53, Corfu, Greece, 1999.

[25] Y. Furukawa and J. Ponce. Accurate, dense, and robust multiview stereopsis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(8): 1362–1376, 2010.

[26] A. Geva, G. Briskin, E. Rivlin, and H. Rotstein. Estimating camera pose using bundle adjustment and digital terrain model constraints. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 4000–4005, Seattle, WA, USA, May 2015.

[27] E. Grosso, G. Sandini, and C. Frigato. Extraction of 3-D information and volumetric uncertainty from multiple stereo images. In *Proceedings of the European Conference on Artificial Intelligence*, pages 683–688, Munich, Germany, Aug. 1988.

[28] H. Hirschmüller. Stereo processing by semiglobal matching and mutual information. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(2):328–341, 2008.

[29] F. S. Hover, J. Vaganay, M. Elkins, S. Willcox, V. Polidoro, J. Morash, R. Damus, and S. Desset. A vehicle system for autonomous relative survey of in-water ships. *Marine Technology Society Journal*, 41(2):44–55, 2007.

[30] 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. *International Journal of Robotics Research*, 31(12):1445–1464, 2012.

[31] P. J. Huber. *Robust Statistics*. Wiley, New York, 2011.

[32] J. Jaffe. Computer modeling and the design of optimal underwater imaging systems. *IEEE Journal of Oceanic Engineering*, 15(2):101–111, 1990.

[33] B. Jalving, M. Mandt, O. K. Hagen, and F. Pøhner. Terrain referenced navigation of AUVs and submarines using multibeam echo sounders. In *Proceedings of the European Conference on Undersea Defense Technology*, Nice, France, June 2004.

[34] M. Johnson-Roberson, O. Pizarro, S. B. Williams, and I. Mahon. Generation and visualization of large-scale three-dimensional reconstructions from underwater robotic surveys. *Journal of Field Robotics*, 27(1):21–51, 2010.

[35] S. B. Kang and M. Jones. Appearance-based structure from motion using linear classes of 3D models. *International Journal of Computer Vision*, 49(1): 5–22, 2002.

[36] M. Kazhdan, M. Bolitho, and H. Hoppe. Poisson surface reconstruction. In *Proceeings of the Symposium on Geometry Processing*, volume 7, pages 61–70, Cagliari, Italy, June 2006.

[37] I. Kemelmacher-Shlizerman and S. M. Seitz. Face reconstruction in the wild. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1746–1753, Barcelona, Spain, 2011.

[38] A. Kim and R. M. Eustice. Pose-graph visual SLAM with geometric model selection for autonomous underwater ship hull inspection. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1559–1565, St. Louis, MO, USA, Oct. 2009.

[39] A. Kim and R. M. Eustice. Real-time visual SLAM for autonomous underwater hull inspection using visual saliency. *IEEE Transactions on Robotics*, 29(3):719–733, June 2013.

[40] J. Li, R. M. Eustice, and M. Johnson-Roberson. High-level visual features for underwater place recognition. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 3652–3659, Seattle, WA, USA, May 2015.

[41] D. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.

[42] D. Meduna, S. Rock, and R. McEwen. Low-cost terrain relative navigation for long-range AUVs. In *Proceedings of the IEEE/MTS OCEANS Conference and Exhibition*, pages 1–7, Woods Hole, MA, USA, Oct. 2008.

[43] R. Morton and E. Olson. Robust sensor characterization via max-mixture models: GPS sensors. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 528–533, Tokyo, Japan, Nov. 2013.

[44] S. Negahdaripour and P. Firoozfam. An ROV stereovision system for ship-hull inspection. *IEEE Journal of Oceanic Engineering*, 31(3):551–564, 2006.

[45] R. A. Newcombe, A. J. Davison, S. Izadi, P. Kohli, O. Hilliges, J. Shotton, D. Molyneaux, S. Hodges, D. Kim, and A. Fitzgibbon. Kinectfusion: Real-time dense surface mapping and tracking. In *Proceedings of the IEEE International Symposium on Mixed and Augmented Reality*, pages 127–136, Basel, Switzerland, Oct. 2011.

[46] D. Nistér, O. Naroditsky, and J. Bergen. Visual odometry for ground vehicle applications. *Journal of Field Robotics*, 23(1):3–20, 2006.

[47] E. Olson and P. Agarwal. Inference on networks of mixtures for robust robot mapping. *International Journal of Robotics Research*, 32(7):826–840, 2013.

[48] P. Ozog, G. Troni, M. Kaess, R. M. Eustice, and M. Johnson-Roberson. Building 3D mosaics from an autonomous underwater vehicle, Doppler velocity log, and 2D imaging sonar. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 1137–1143, Seattle, WA, USA, May 2015.

[49] P. Ozog, N. Carlevaris-Bianco, A. Kim, and R. M. Eustice. Long-term mapping techniques for ship hull inspection and surveillance using an autonomous underwater vehicle. *Journal of Field Robotics*, 33(3):265–289, 2016.

[50] P. Ridao, M. Carreras, D. Ribas, and R. Garcia. Visual inspection of hydroelectric dams using an autonomous underwater vehicle. *Journal of Field Robotics*, 27(6):759–778, 2010.

[51] D. M. Rosen, M. Kaess, and J. J. Leonard. Robust incremental online inference over sparse factor graphs: Beyond the Gaussian case. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 1025–1032, Karlsruhe, Germany, May 2013.

[52] J. Roth, Y. Tong, and X. Liu. Unconstrained 3D face reconstruction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2606–2615, Boston, MA, USA, June 2015.

[53] W. J. Schroeder, B. Lorensen, and K. Martin. The visualization toolkit. http://vtk.org, 2004.

[54] A. Segal, D. Haehnel, and S. Thrun. Generalized-ICP. In *Proceedings of the Robotics: Science & Systems Conference*, Seattle, WA, USA, June 2009.

[55] Y. Shan, Z. Liu, and Z. Zhang. Model-based bundle adjustment with application to face modeling. In *Proceedings of the IEEE International Conference on Computer Vision*, volume 2, pages 644–651, Vancouver, Canada, 2001. IEEE.

[56] R. Smith, M. Self, and P. Cheeseman. Estimating uncertain spatial relationships in robotics. In *Proceedings of Uncertainty in AI*, pages 435–461. Elsevier, 1986.

[57] M. Stojanovic and J. Preisig. Underwater acoustic communication channels: Propagation models and statistical characterization. *IEEE Communications Magazine*, 47(1):84–89, 2009.

[58] H. Strasdat, J. M. M. Montiel, and A. J. Davison. Real-time monocular SLAM: Why filter? In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 2657–2664, May 2010.

[59] N. Sunderhauf and P. Protzel. Switchable constraints for robust pose graph SLAM. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1879–1884, Algarve, Portugal, Oct. 2012.

[60] P. H. Torr. Model selection for two view geometry: A review. In *Shape, Contour and Grouping in Computer Vision*, pages 277–301. Springer, 1999.

[61] G. Trimble and E. Belcher. Ship berthing and hull inspection using the CetusII AUV and MIRIS high-resolution sonar. In *Proceedings of the IEEE/MTS OCEANS Conference and Exhibition*, pages 1172–1175, Biloxi, Mississippi, 2002.

[62] R. J. Urick. *Principles of underwater sound for engineers*. Tata McGraw-Hill Education, 1967.

[63] T. Whelan, M. Kaess, M. Fallon, H. Johannsson, J. Leonard, J. Mcdonald, and J. J. Leonard. Kintinuous : Spatially extended KinectFusion. In *RSS Workshop on RGB-D: Advanced Reasoning with Depth Cameras*, Sydney, Australia, July 2012.

[64] C. Wu. SiftGPU: A GPU implementation of scale invariant feature transform (SIFT). `http://cs.unc.edu/~ccwu/siftgpu`, 2007.

[65] Z. Zhang. Iterative point matching for registration of free-form curves and surfaces. *International Journal of Computer Vision*, 13(2):119–152, 1994.