

Identifying Structural Anomalies in Image Reconstructions of Underwater Ship Hulls

Paul Ozog and Ryan M. Eustice

Abstract—This paper reports on an algorithm enabling an autonomous underwater vehicle (AUV) to localize into a 3D computer aided design (CAD) model of a ship hull *in situ* using an optical camera and Doppler velocity log (DVL). The precision of our localization algorithm allows the identification of structural deviations between 3D structure inferred from bundle-adjusted camera imagery and the CAD model. These structural deviations are clustered into shapes, which allow us to fuse camera-derived structure into a CAD-derived 3D mesh. This augmented CAD model can be used within a 3D photomosaicing pipeline, providing a visually intuitive 3D reconstruction of the ship hull. We evaluate our algorithm on the Bluefin Robotics Hovering Autonomous Underwater Vehicle (HAUV) surveying the *SS Curtiss*, and provide a 3D reconstruction that fuses the CAD mesh with 3D information corresponding to underwater structure, such as biofouling.

I. INTRODUCTION

Within the autonomous underwater vehicle (AUV) community, simultaneous localization and mapping (SLAM) is a powerful technique to both correct for navigational drift and provide a visually intuitive 3D reconstruction of the environment [1–4]. In certain applications, such as *in situ* underwater ship hull inspection, AUVs face several important perceptual challenges while performing SLAM: the absence of acoustic beacons, lack of a global positioning system (GPS), and turbid water.

3D reconstruction remains a tremendous challenge in certain underwater environments due to several considerations for underwater optical imaging. In particular, the water column around ship hulls tends to be murky, making optical imaging difficult unless the camera is physically very close to the scene. Despite these challenges, the main benefit of optical vision-based approaches is high spatial resolution and cost savings as compared to acoustic-based systems.

To alleviate the challenges of underwater SLAM, our prior work [5] leveraged a computer aided design (CAD) model to assist in a 3D photomosaicing pipeline for an imaging sonar. However, this previous approach only applied texture to a 3D mesh, and did not identify structural differences detected from the perceptual sensors. We build upon this previous work by annotating a prior model with SLAM-derived structure. We show experimental results taken from

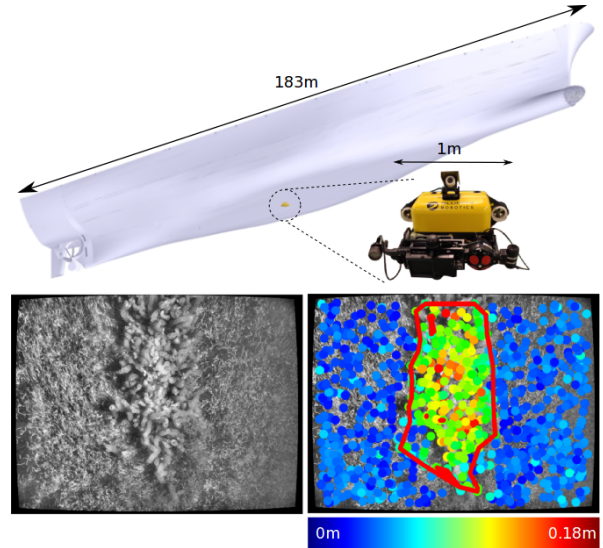


Fig. 1. Our algorithm allows an AUV, such as the Bluefin Robotics HAUV, to precisely localize into a nominal CAD mesh of the inspected ship (top). Unlike the approach by Trimble and Belcher [7], this localization takes place without any acoustic beacons or pingers. The sparse visual 3D reconstruction, shown on the bottom-right, shows that significant 3D structure exists in the imagery that does not exist in the CAD model (in this case, biofouling). 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.

the Hovering Autonomous Underwater Vehicle (HAUV) platform for automated ship hull inspection [6]. An overview of our approach can be found in Fig. 1. The contributions of this paper allow our AUV to:

- Localize into a 3D CAD model of the ship hull being surveyed *without* the use of long-baseline navigation [7].
- Label visually-derived 3D shapes based on their deviation from the nominal CAD mesh.
- Augment the nominal CAD mesh with visually-derived 3D information.

A. Related Work

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 [7]. 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 [8]. Some examples of field robots that use visual perception include work by Negahdaripour and Firoozfam [9], in which they used a

*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 N016970-UM-RCMOP.

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.

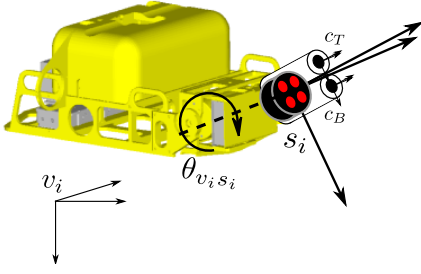


Fig. 2. Illustration of the various reference frames at time i . The vehicle has a sensor tray that houses both the DVL and vertically-oriented stereo rig. An onboard servo rotates the servo frame, s_i , which in turn rotates the DVL and camera. The vehicle controls this angle so that these sensors point 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.

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. [3] 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 [10] and acoustic sensors [11].

The computer vision and graphics community have studied fusing optical range measurements to form a reconstruction of a 3D surface for several decades [12–14]. The seminal work by Curless and Levoy [13] used running averaging 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 [15, 16]. 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 [17–19]. Third, we do not assume the availability of range images. Though we use a stereo camera for our experimental analysis (from which a disparity map can be easily converted to a range image), we instead use sparse feature-based registration so that this approach is also applicable to monocular (bearing-only) cameras.

B. Outline

This paper is organized as follows. In Section II, we describe the mathematical components of our localization and bundle adjustment framework, followed by an overview of a visual feature clustering algorithm. In addition, we describe an algorithm to annotate the nominal CAD-derived 3D mesh with visually-derived 3D structure. In Section III, we provide experimental results for our approach, and in Section IV we offer some concluding remarks.

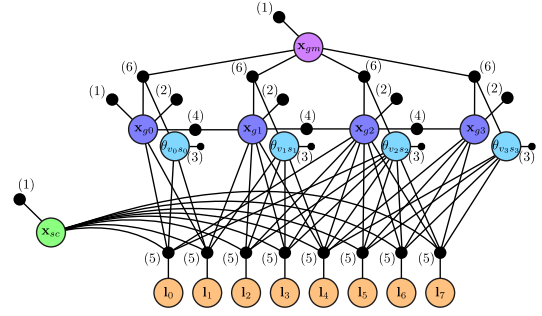


Fig. 3. Our localization and bundle adjustment pipeline can be represented as a factor-graph. Each factor node (black dot) is labeled with the corresponding conditional distribution (1 through 6) from Section II-A. The variable nodes correspond to the unknowns we are estimating: robot poses (blue), servo angles (cyan), vehicle-to-servo relative pose (green), visual features (orange), and pose of the prior CAD model (purple).

II. APPROACH

The notation for this section is as follows. Let g be the hull-relative global coordinate frame, and l_j be the 3D position of a visual feature indexed by j , as expressed in the global frame. We denote the 6-degree of freedom (DOF) relative-pose between frames i and j with the notation $\mathbf{x}_{ij} = [\mathbf{t}_{ij}^T, \phi_{ij}, \theta_{ij}, \psi_{ij}]^T$, where $\mathbf{t}_{ij}^T = [x_{ij}^i, y_{ij}^i, z_{ij}^i]^T$ is the translation vector from i to j as expressed in frame i . ϕ_{ij} , θ_{ij} , and ψ_{ij} are Euler angles about the x , y , and z axis, respectively, that describe the rotation from i to j . Let $\mathcal{M}$ denote a triangular mesh consisting of a set of vertices, edges between vertices, and triangular faces. $\mathcal{M}_{\text{prior}}$ is the prior (i.e., nominal) mesh, and $\mathcal{M}_{\text{new}}$ is the updated mesh (i.e., the final output of our algorithm).

A. Localization and Bundle Adjustment

Our localization and bundle adjustment pipeline estimates the pose of the vehicle at time i , $\mathbf{x}_{gvi}$, the static servo-to-camera transform, $\mathbf{x}_{scT}$, the angle of the servo at time i , $\theta_{v_i s_i}$, the pose of the CAD model, $\mathbf{x}_{gm}$, and the 3D position of scale invariant feature transform (SIFT) [20] features indexed by j , l_j . Here, the servo refers to a simple actuator that rotates both the camera and Doppler velocity log (DVL) to point orthogonal to the ship hull, as shown in Fig. 2. These unknown parameters are jointly estimated using a factor-graph SLAM framework [21]. An illustration of the factor-graph encoding the parameter’s probabilistic distribution is shown in Fig. 3.

Dellaert and Kaess [21] showed that a factor-graph can represent the joint distribution over all unknowns in the SLAM problem. A factor-graph is a bipartite graph consisting of variable nodes (unknowns) and factor nodes (observations) that encodes the joint distribution of all unknowns:

$$P(\mathbf{X}) \propto \prod_i \Psi_i(\mathbf{S}_i),$$

where $\mathbf{X}$ is a stacked vector of unknowns and $\mathbf{S}_i$ is the subset of unknowns that support (i.e., are connected to) the i^{th} factor node with potential Ψ_i . The product of these factors can be easily converted into a nonlinear least-squares optimization

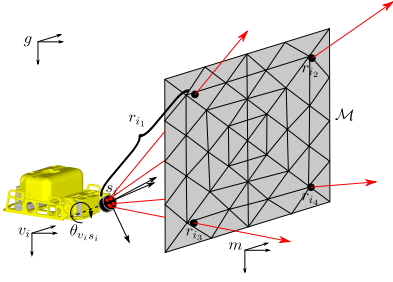


Fig. 4. Illustration of ray-casting constraint. Given pose of the vehicle frame at time i , $\mathbf{x}_{gv_i}$, the servo angle, $\theta_{v_i s_i}$, the pose of the CAD frame, $\mathbf{x}_{gm}$, and the prior mesh, $\mathcal{M}_{\text{prior}}$, the four DVL range returns can be computed with an efficient octree-based ray-casting approach. At time i , the four ranges are predicted as r_{i1} , r_{i2} , r_{i3} , and r_{i4} .

problem corresponding to a maximum *a posteriori* (MAP) estimate of $\mathbf{X}$:

$$\begin{aligned} \mathbf{X}^* &= \operatorname{argmax}_{\{\mathbf{X}\}} P(\mathbf{X}) \\ &= \operatorname{argmin}_{\{\mathbf{X}\}} - \sum_i \log \Psi_i(\mathbf{S}_i). \end{aligned}$$

The factor potentials represent conditional distributions of the observed measurement given the unknowns. Below, we elaborate on each factor used in our formulation (from Fig. 3). The covariance matrix for each factor is assumed known, as is standard practice. We use the Ceres library to solve the above nonlinear optimization problem [22].

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 $\mathbf{z}_{\mathbf{x}_{ij}}^{\text{full}}$:

$$P(\mathbf{z}_{\mathbf{x}_{ij}}^{\text{full}} | \mathbf{x}_{ij}) \sim \mathcal{N}(\mathbf{x}_{ij}, \Sigma_{\mathbf{z}_{\mathbf{x}_{ij}}^{\text{full}}}). \quad (1)$$

The initial guess for the pose of the CAD model, $\mathbf{x}_{gm}$, is determined using the generalized iterative closest point (GICP) algorithm for aligning two point clouds [19]. 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 using the method described in [2]. This GICP alignment is added as a prior factor on $\mathbf{x}_{gm}$.

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

$$P(\mathbf{z}_{\mathbf{x}_{ij}}^{\text{zpr}} | \mathbf{x}_{ij}) \sim \mathcal{N}([z_{ij}^i, \phi_{ij}, \theta_{ij}]^\top, \Sigma_{\mathbf{z}_{\mathbf{x}_{ij}}^{\text{zpr}}}). \quad (2)$$

Finally, we model the servo angle at time i as being directly observed with a prior factor. The corresponding observation model is simply:

$$P(v_{i s_i} | \theta_{v_i s_i}) \sim \mathcal{N}(\theta_{v_i s_i}, \sigma_{z_{s_i}}^2). \quad (3)$$

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

$$P(\mathbf{z}_{i(i+1)}^{\text{odo}} | \mathbf{x}_{gi}, \mathbf{x}_{g(i+1)}) \sim \mathcal{N}(\ominus \mathbf{x}_{gi} \oplus \mathbf{x}_{g(i+1)}, \Sigma_{\mathbf{z}_{i(i+1)}^{\text{odo}}}), \quad (4)$$

where $\oplus$ and $\ominus$ are pose composition operators following the conventions of Smith et al. [23].

3) *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:

$$\begin{aligned} P\left(\left[\mathbf{z}_{ik}^T \quad \mathbf{z}_{ik}^B\right]^\top \middle| \mathbf{x}_{gv_i}, \mathbf{x}_{sc_T}, \theta_{v_i s_i}, \mathbf{l}_k\right) \\ \sim \mathcal{N}\left(h_c(\mathbf{x}_{gv_i}, \mathbf{x}_{sc_T}, \theta_{v_i s_i}, \mathbf{l}_k), \sigma_c^2 \mathbf{I}_{4 \times 4}\right). \end{aligned} \quad (5)$$

The observation model, h_c , corresponds to two pinhole cameras in a calibrated and rectified vertical stereo configuration (from Fig. 2):

$$h_c(\mathbf{x}_{gv_i}, \mathbf{x}_{sc_T}, \theta_{v_i s_i}, \mathbf{l}_k) = \begin{bmatrix} \text{dehom}(K(R\mathbf{l}_k + \mathbf{t})) \\ \text{dehom}(K'(R'\mathbf{l}_k + \mathbf{t}')) \end{bmatrix}.$$

$\text{dehom}(\cdot)$ denotes the dehomogenization of a three-dimensional vector, R and $\mathbf{t}$ represent the transformation of points from the global frame to the top camera, (i.e., they correspond to 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 $\mathbf{x}_{v_i s_i} = [0, 0, 0, 0, \theta_{v_i s_i}, 0]^\top$), and R' and $\mathbf{t}'$ transform points in the global frame to the bottom camera (i.e., they correspond to the rotation and translation of the composed pose $(\mathbf{x}_{gv_i} \oplus \mathbf{x}_{v_i s_i} \oplus \mathbf{x}_{sc_T} \oplus \mathbf{x}_{sc_{CB}})$, where $\mathbf{x}_{sc_{CB}}$ is the transformation from the top camera frame to the bottom camera frame). This transformation is taken from stereo camera calibration.

4) *DVL raycast factors*: A critical component of our factor-graph formulation are 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:

$$\begin{aligned} P(z_{r_{in}} | \mathbf{x}_{gm}, \mathbf{x}_{gv_i}, \mathbf{x}_{sc_T}, \theta_{v_i s_i}) \\ \sim \mathcal{N}\left(h_{rn}(\mathbf{x}_{gm}, \mathbf{x}_{gv_i}, \mathbf{x}_{sc_T}, \theta_{v_i s_i}; \mathcal{M}_{\text{prior}}), \sigma_{z_{r_{in}}}^2\right), \end{aligned} \quad (6)$$

where h_{rn} corresponds to raycasting the n^{th} beam for a DVL in a four-beam Janus configuration [24]. This observation model is illustrated in Fig. 4. Since the prior mesh may consist of hundreds of thousands of triangles, we use an efficient octree-based raycast implementation [25]. In addition, when evaluating the Gaussian distribution from (5) and (6) we apply a Huber M-estimator to the squared loss term to automatically reject outliers.

B. Identifying Shapes by Clustering Features

Once the MAP estimate of all unknowns is computed from the previous section, we can easily compute the structural deviation from the CAD model for each feature as follows. If the 3D position of a visual feature is within the field of view of a given camera pose, then we compute a ray originating at the camera center of projection and extending toward infinity. We use a raycasting approach to compute the intersection on the CAD model surface. The structural deviation is the difference between the bundle-adjusted position and the position at which the ray intersects the CAD model.

Once the deviations of each feature are computed, we apply Density-Based Spatial Clustering of Applications with Noise (DBSCAN) to nonlinearly separate the features' positions into clusters [26]. These clusters of points are converted

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

Require: Camera pose (pose_i), visible features ($\mathcal{F}_{vi}$), and mesh ($\mathcal{M}_{\text{prior}}$)

- 1: $\mathcal{P}_i = \emptyset$ {Set of points to cluster.}
- 2: $\mathcal{C}_i = \emptyset$ {Set of clusters from DBSCAN.}
- 3: $\mathcal{S}_i = \emptyset$ {Set of detected shapes.}
- 4: **for** feature $\mathbf{f} = [f_x, f_y, f_z]$ in $\mathcal{F}_{vi}$, expressed in camera frame **do**
- 5: $d = \text{deviation}(\mathbf{f}, \text{raycast}(\text{pose}_i, \mathbf{f}, \mathcal{M}_{\text{prior}}))$
- 6: **if** $d > \tau$ **then**
- 7: $\mathcal{P}_i = \mathcal{P}_i \cup \{f_x, f_y, d\}$
- 8: **end if**
- 9: **end for**
- 10: $\mathcal{C}_i = \text{DBSCAN}(\mathcal{P}_i)$
- 11: **for** cluster in $\mathcal{C}_i$ **do**
- 12: $\mathcal{M}_{S_i} = \text{alpha_shape}(\text{cluster})$
- 13: $\mathcal{S}_i = \mathcal{S}_i \cup \mathcal{M}_{S_i}$
- 14: **end for**
- 15: **return** $\mathcal{S}_i$

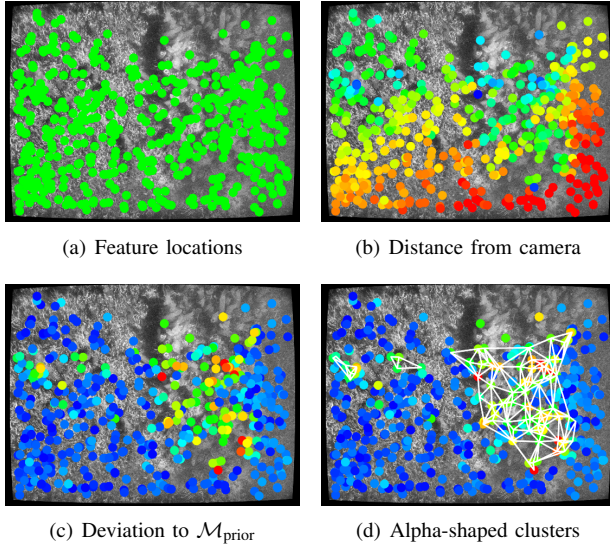


Fig. 5. Visual overview of Algorithm 1. For a particular keyframe, the bundle adjusted features, (a) and (b), are assigned a deviation by computing the intersection of the ray to the CAD model (c). These values are clustered using DBSCAN, and meshed using alpha-shapes. In (d), the three detected clusters have their alpha-shapes shown as white triangular meshes.

to shapes using a simple extension to Delaunay triangulation known as *alpha-shapes* [27]. This algorithm is summarized in Algorithm 1, with an accompanying example in Fig. 5. As shown in Fig. 5(d), the detected shapes can be projected as two-dimensional triangular meshes in the camera imagery. Note that 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 $\mathcal{M}_{\text{prior}}$.

C. CAD 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 and extending toward a vertex in $\mathcal{M}_{\text{prior}}$. Again, we use raycasting to compute the intersection with any detected shapes. Once

Algorithm 2 Fuse shapes into prior mesh

- 1: $\mathcal{M}_{\text{new}} = \mathcal{M}_{\text{prior}}$ {Make a deep copy of the prior mesh}
- 2: **for** pose in poses **do**
- 3: $\mathcal{F}_v = \text{is_visible}(\text{pose}, \mathcal{F})$ {Visible features}
- 4: $\mathcal{S}_d = \text{Algorithm1}(\text{pose}, \mathcal{F}_v, \mathcal{M}_{\text{prior}})$ {Detected shapes}
- 5: $\mathcal{V}_n = \text{nearby_vertices}(\text{pose}, \text{vertices}(\mathcal{M}_{\text{prior}}))$
- 6: **for** shape $\mathcal{M}_s$ in $\mathcal{S}_d$ **do**
- 7: **for** vertex $\mathbf{v}^i \in \mathcal{V}_n$ indexed by i **do**
- 8: $\text{ray} = \text{make_ray}(\text{pose}, \mathcal{V}_n[i])$
- 9: **if** $\text{ray.intersects_with}(\mathcal{M}_s)$ **then**
- 10: $\mathbf{p}^i = \text{ray.intersection}(\mathcal{M}_s)$
- 11: $\text{moving_avg}(\mathcal{M}_{\text{new}}, i, \mathbf{p}^i)$ {Using Eqn. (7)}
- 12: **end if**
- 13: **end for**
- 14: **end for**
- 15: **end for**
- 16: **return** $\mathcal{M}_{\text{new}}$

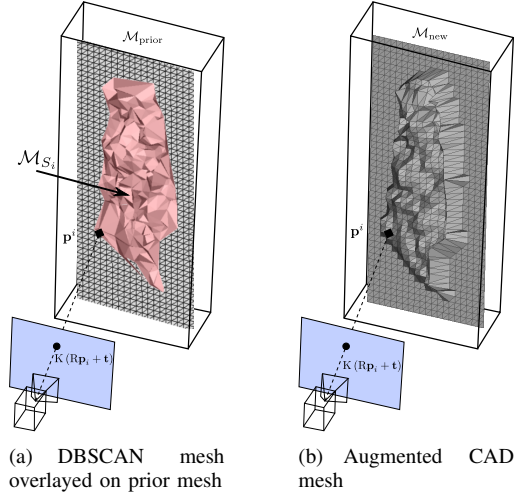


Fig. 6. Visualization of Algorithm 2. A 3D shape, $\mathcal{S}$, derived from Algorithm 1 can be fused into the prior CAD mesh by intersecting rays from the camera frame. For a particular camera pose, we choose a CAD mesh vertex, $\mathcal{V}_n[i]$, and then compute the intersection of the camera-to-vertex 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.

the intersection point corresponding to the i^{th} vertex, $\mathbf{p}^i$, is calculated, we update the corresponding vertex in $\mathcal{M}_{\text{new}}$, $\hat{\mathbf{v}}^i$, with a recursive moving average filter:

$$\hat{\mathbf{v}}_{n+1}^i = \frac{\mathbf{p}^i + n\hat{\mathbf{v}}_n^i}{n+1} \quad (7)$$

$$\hat{\mathbf{v}}_0^i = \text{get_ith_vertex}(\mathcal{M}_{\text{prior}}, i),$$

where the i^{th} vertex's counter, n , is incremented after every evaluation of line 11 from Algorithm 2.

This process is repeated for every pose. A summary of this algorithm is provided in Algorithm 2. Note that we conservatively and efficiently determine the visible features and nearby vertices (lines 3 and 5 of Algorithm 2), using a k-d tree-based radius search.

III. EXPERIMENTAL TRIALS

In this section, we provide an overview of the experimental trials used to analyze the methods proposed. Various statistics



(a)

Ship Length	183 m
Ship Beam	27 m
Ship Draft	9.1 m
AUV trajectory length	0.963 km
Number of images	44,868
Number of DVL raycasts	96,944
Number of feature reprojections	170,802
Number of vertices in $\mathcal{M}_{\text{prior}}$	2,573,366
Number of faces in $\mathcal{M}_{\text{prior}}$	5,052,456

(b)

Fig. 7. The field data in this paper was obtained from underwater surveys of the *SS Curtiss*, shown in (a). Various statistics for the amount of data processed, including the size of the bundle adjustment step, are shown in the table in (b).

for the physical size of the surveyed vessel, the *SS Curtiss*, and the amount of processed data is summarized in Fig. 7.

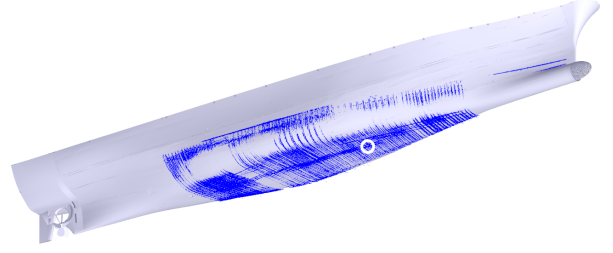
A. Localization and Shape Identification

Several characteristics of our localization and shape identification algorithms are summarized in Fig. 8. First, we consider the importance of incorporating DVL range measurements from (6). From Fig. 8(a) and Fig. 8(b), we can see noticeable misalignment particularly on the side of the hull. By examining the distribution of residuals in Fig. 8(c), we quantitatively confirm that the alignment of the DVL trajectory to $\mathcal{M}_{\text{prior}}$ is poor. However, by incorporating raycasting in the factor-graph framework, we can significantly tighten the distributions of these residuals.

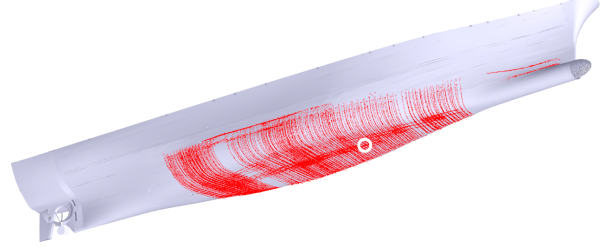
Second, we empirically analyze how the incorporation of raycasting factors directly translates to improved shape detection performance in Algorithm 1. An illustrative example is shown in Fig. 8(d) through Fig. 8(f). The algorithm tends to falsely assign features as having significant deviation to the nominal ship hull surface. Though we do not have ground-truth, we know from human divers that the green regions from Fig. 8(e) ought to be more flush with the hull. However, by incorporating raycasting factors, the results shown in Fig. 8(f) are more consistent with the geometry reported by divers. Indeed, by closely examining Fig. 8(d), we can see bare metal on the ship hull, suggesting little deviation from the ship hull on either side of the large biofouling in the center of the image.

B. Remeshing Results

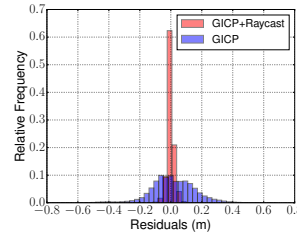
The remeshed CAD model for a visual survey of the *SS Curtiss* is provided in Fig. 9. We provide different results for the threshold, τ , used for determining the eligibility of features to be clustered with DBSCAN. For this application, the preferred approach is to keep the threshold zero (Fig. 9(b)), however for certain applications where false positives are a concern, this can be raised (Fig. 9(c)). In



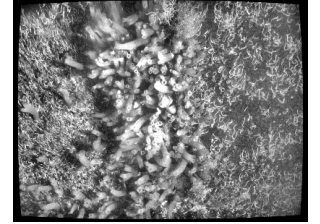
(a) DVL-to-CAD: GICP only



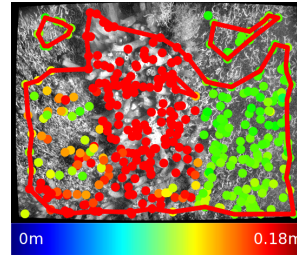
(b) DVL-to-CAD: GICP with raycasting



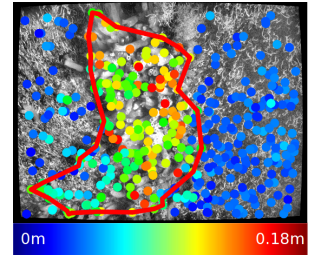
(c) DVL beam residuals: (a) vs (b)



(d) Sample raw image



(e) Clustering: GICP only



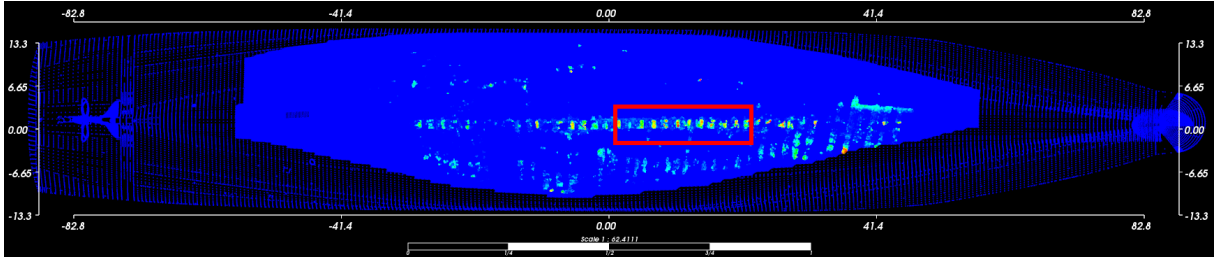
(f) Clustering: GICP with raycasting

Fig. 8. The improved consistency of our localization algorithm, shown in (a) through (c), allows for the identification of structural anomalies using underwater camera imagery, shown in (d) through (f). Note that (d) through (f) correspond to the same image, which is taken from the region of (a) and (b) circled in white. According to the CAD model, this region is near-perfectly flat.

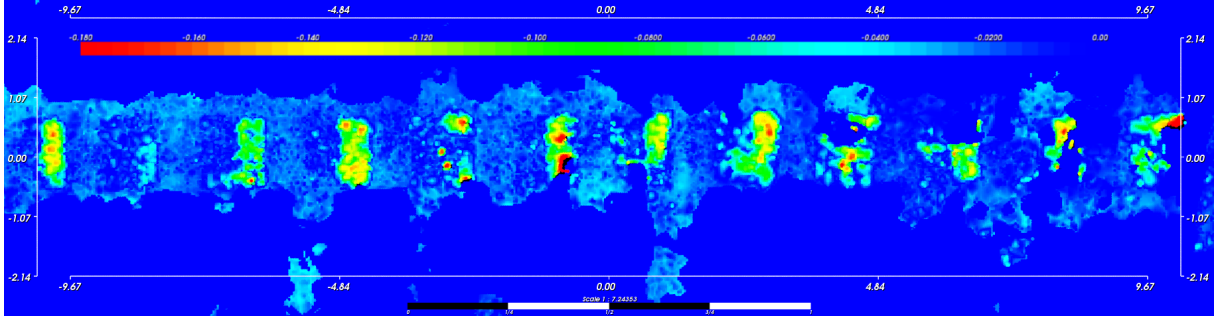
this figure, the rectangular-like structural anomalies shown in Fig. 9(c) correspond to support areas used in drydocking.

C. Utility in 3D Photomosaic Visualization

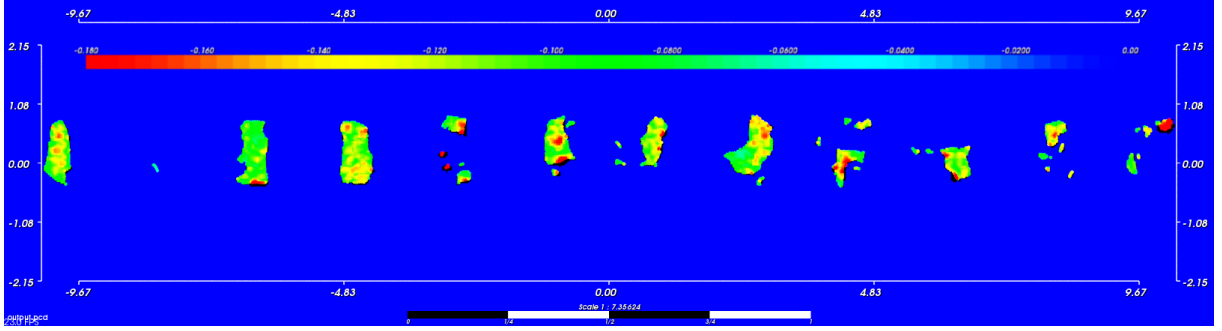
In addition to providing visually intuitive false color maps, the remeshed model can easily be used in a start-of-the-art 3D photomosaicing framework [28]. An example of this application is provided in Fig. 10. Unlike our previous work in [5] that applied texture to the ship's CAD model, this work allows additional structural details at a relatively small scale. In Fig. 10(b) and Fig. 10(c), we shade the regions of the 3D photomosaic according to height in the z -direction. Clearly,



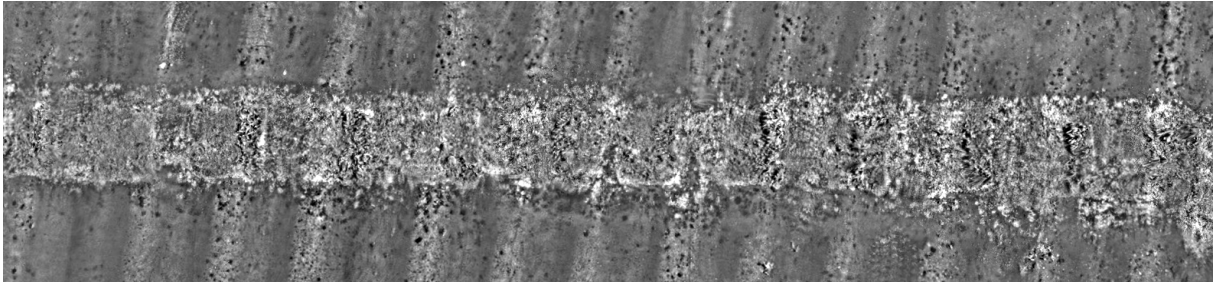
(a)



(b) $\tau = 0.0$ m



(c) $\tau = 0.06$ m



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

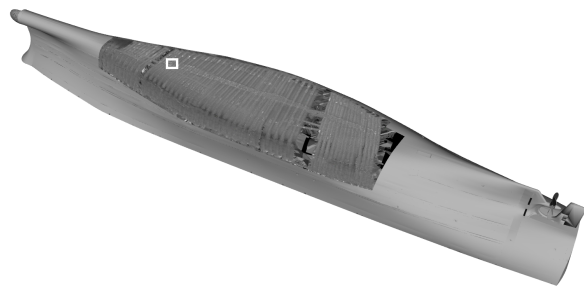
Fig. 9. In (a), we show a heatmap of the remeshed CAD vertices. The red region is expanded in (b) and (c). In (b), the clustering threshold from Algorithm 2, τ , is zero while in (c) it is relatively high. Lowering τ provides more details but potentially introduces false positives. We can see the red rectangular region from (a) corresponds to a strip of biofouling at the ship's centerline, shown in (d).

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.

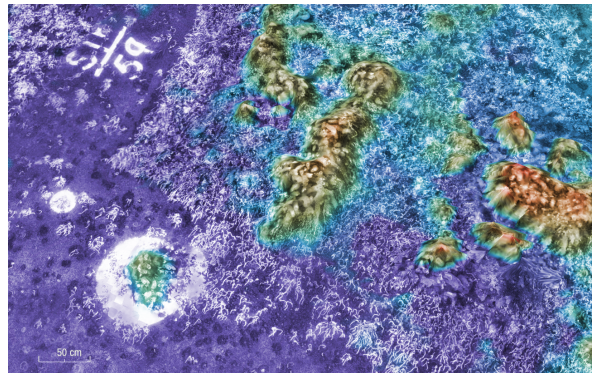
IV. CONCLUSION

We have shown that our localization and bundle adjustment framework provides precise capabilities for identifying visually-observed 3D structure that is absent from the CAD model. In addition, we provide a remeshing algorithm to

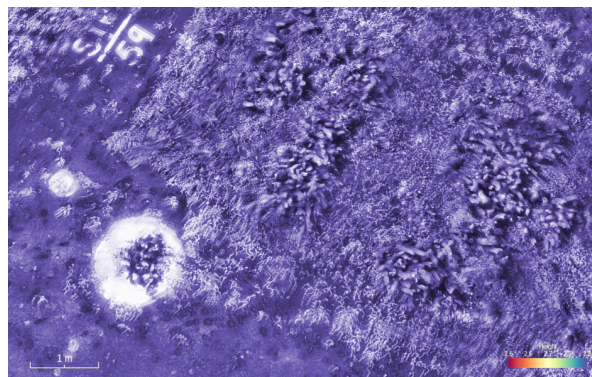
fuse these shapes into a prior mesh using approaches inspired from the computer graphics community. This newly remeshed model has several important benefits for data visualization. In particular, the false-color figures shown in this paper offer an intuitive visualization that is harder to discern from 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.



(a) Birds-eye view: (b) and (c) correspond to white region



(b) Close-up: proposed method



(c) Close-up: method from [5]

Fig. 10. Our approach allows 3D photomosaicing approaches to combine large-scale consistency in (a) with small-scale detail in (b). In (b), the mosaic is shaded according to height. Using the approach from [5], where a CAD model is used for photomosaicing, the small-scale details are lost as evidenced by the region in (c) being near-perfectly flat.

REFERENCES

- [1] 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, 2013.
- [2] 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," *J. Field Robot.*, 2015, in Press.
- [3] 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.
- [4] R. Campos, R. Garcia, P. Alliez, and M. Yvinec, "A surface reconstruction method for in-detail underwater 3D optical mapping," *Int. J. Robot. Res.*, vol. 34, no. 1, pp. 64–89, 2015.
- [5] 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 *Proc. IEEE Int. Conf. Robot. and Automation*, Seattle, WA, USA, May 2015, pp. 1137–1143.
- [6] F. 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 Tech. Soc. J.*, vol. 41, no. 2, pp. 44–55, 2007.
- [7] G. Trimble and E. Belcher, "Ship berthing and hull inspection using the CetusII AUV and MIRIS high-resolution sonar," in *Proc. IEEE/MTS OCEANS Conf. Exhib.*, vol. 2, Biloxi, MS, USA, 2002, pp. 1172–1175.
- [8] F. Bonin, A. Burguera, and G. Oliver, "Imaging systems for advanced underwater vehicles," *J. Maritime Res.*, vol. 8, pp. 65–86, 2011.
- [9] S. Negahdaripour and P. Firoozfam, "An ROV stereovision system for ship-hull inspection," *IEEE J. Ocean. Eng.*, vol. 31, no. 3, pp. 551–564, 2006.
- [10] F. Bonnín-Pascual and A. Ortiz, "Detection of cracks and corrosion for automated vessels visual inspection," in *Proc. Int. Conf. of the Catalan Assoc. Artif. Intell.*, Tarragona, Spain, Oct. 2010, pp. 111–120.
- [11] E. Belcher, B. Matsuyama, and G. Trimble, "Object identification with acoustic lenses," in *Proc. IEEE/MTS OCEANS Conf. Exhib.*, vol. 1, Kona, HI, USA, Sep. 2001, pp. 6–11.
- [12] E. Grosso, G. Sandini, and C. Frigato, "Extraction of 3-D information and volumetric uncertainty from multiple stereo images," in *Proc. European Conf. Artif. Intell.*, Munich, Germany, Aug. 1988, pp. 683–688.
- [13] 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*. New Orleans, LA: ACM, Aug. 1996, pp. 303–312.
- [14] M. Kazhdan, M. Bolitho, and H. Hoppe, "Poisson surface reconstruction," in *Proc. Symp. Geometry Processing*, vol. 7, Cagliari, Italy, Jun. 2006.
- [15] 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 *2011 10th IEEE Int. Symp. on Mixed and Augmented Reality*, Oct. 2011, pp. 127–136.
- [16] T. Whelan, M. Kaess, M. Fallon, H. Johannsson, J. Leonard, and J. McDonald, "Kintinuous: Spatially extended KinectFusion," in *RSS Workshop on RGB-D: Advanced Reasoning with Depth Cameras*, Jul. 2012.
- [17] Z. Zhang, "Iterative point matching for registration of free-form curves and surfaces," *Int. J. Comput. Vis.*, vol. 13, no. 2, pp. 119–152, 1994.
- [18] Y. Chen and G. Medioni, "Object modeling by registration of multiple range images," in *Proc. IEEE Int. Conf. Robot. and Automation*, Nice, France, May 1991, pp. 2724–2729.
- [19] A. Segal, D. Haehnel, and S. Thrun, "Generalized-ICP," in *Proc. Robot.: Sci. & Syst. Conf.*, Seattle, WA, USA, Jun. 2009.
- [20] D. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, 2004.
- [21] 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.
- [22] S. Agarwal, K. Mierle, and Others, "Ceres solver," <http://ceres-solver.org>.
- [23] R. Smith, M. Self, and P. Cheeseman, "Estimating uncertain spatial relationships in robotics," in *Proc. Uncertainty AI*. Philadelphia, PA, USA: Elsevier, 1986, pp. 267–288.
- [24] N. Brokloff, "Matrix algorithm for Doppler sonar navigation," in *Proc. IEEE/MTS OCEANS Conf. Exhib.*, vol. 3, Brest, France, 1994, pp. III–378.
- [25] W. J. Schroeder, B. Lorensen, and K. Martin, "The visualization toolkit," <http://vtk.org>, 2004.
- [26] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *Int. Conf. on Know. Disc. and Data Mining*, vol. 96, no. 34, Portland, OR, USA, 1996, pp. 226–231.
- [27] H. Edelsbrunner and E. P. Mücke, "Three-dimensional alpha shapes," *ACM Trans. on Graphics*, vol. 13, no. 1, pp. 43–72, 1994.
- [28] M. Johnson-Roberson, O. Pizarro, S. B. Williams, and I. Mahon, "Generation and visualization of large-scale three-dimensional reconstructions from underwater robotic surveys," *J. Field Robot.*, vol. 27, no. 1, pp. 21–51, 2010.