IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2023, London, UK. Cite as RA-L paper.

Hybrid Visual SLAM for Underwater Vehicle Manipulator Systems

Gideon Billings[1], Richard Camilli[2], Matthew Johnson-Roberson[3]

***Abstract*— This paper presents a novel visual feature based scene mapping method for underwater vehicle manipulator systems (UVMSs), with specific emphasis on robust mapping in natural seafloor environments. Our method uses GPU accelerated SIFT features in a graph optimization framework to build a feature map. The map scale is constrained by features from a vehicle mounted stereo camera, and we exploit the dynamic positioning capability of the manipulator system by fusing features from a wrist mounted fisheye camera into the map to extend it beyond the limited viewpoint of the vehicle mounted cameras. Our hybrid SLAM method is evaluated on challenging image sequences collected with a UVMS in natural deep seafloor environments of the Costa Rican continental shelf margin, and we also evaluate the stereo only mode on a shallow reef survey dataset. Results on these datasets demonstrate the high accuracy of our system and suitability for operating in diverse and natural seafloor environments. We also contribute these datasets[3] for public use.**

***Index Terms*—SLAM, Sensor Fusion, Computer Vision for Automation, Computer Vision for Other Robotic Applications, Data Sets for Robotic Vision**

I. Introduction

Exploration vehicles for remote environments, such as rovers, planetary landers, or underwater Remotely Operated Vehicles (ROVs) are often equipped with manipulator systems for collecting samples, placing sensors, or otherwise interacting with the environment. These systems largely rely on direct tele-operation or manually scripted commands to execute manipulation tasks, due to the risks associated with acting in unstructured and often complex remote environments. Despite these risks, there are some remote environments, such as Europa, the icy moon of Jupiter, so distant that any kind of tele-operation or pre-scripted manipulator control is highly impractical. Considering environments closer to home, the deep ocean is a domain of intensive scientific research and exploration, but operation of deep submergence ROVs with attendant support ships and pilots is costly and availability is limited. Also, human error during tele-operated control can lead to damage to the vehicle or loss of valuable mission time. These considerations motivate the automation of manipulator systems for exploration vehicles, to enable complex workspace interactions in communication limited or denied environments and to support tele-operated procedures. This advancement can reduce the overhead requirements of supporting tele-operation infrastructure (including associated human labor and hotel functions), thereby increasing the availability of robotic systems for scientific research. Critical to achieving safe and robust autonomy of such vehicle-manipulator systems is scene perception and reconstruction. In this work, we address the problem of feature based 3D scene mapping for underwater vehicle-manipulator systems (UVMSs). A key innovation of our mapping system is the fusion of feature points from both a vehicle mounted stereo camera and a dynamically positioned manipulator mounted fisheye camera into the same mapping framework. In situations where a UVMS's movement is limited or risky, our method addresses the challenge of limited viewpoints from the vehicle mounted cameras and incomplete scene reconstruction due to shadowing from scene structure by enabling the wrist mounted camera to dynamically extend the map beyond the vehicle fixed camera views and fill in shadowed areas of the scene.

This work makes the following contributions:

1) To our knowledge, the first SLAM system, designed for manipulator systems, that fuses a manipulator mounted fisheye camera into the same map with a vehicle mounted stereo camera.
2) An adaptation of the ORB-SLAM2 framework to GPU accelerated SIFT features, with improved odometer based tracking and real-time performance.
3) An evaluation of our method on both shallow reef and natural deep seafloor environments, where our method achieves good performance and standard ORB-SLAM2 fails. The evaluation datasets are also published with this paper[4].

II. Related Work

3D scene mapping is a very mature problem in computer vision and robotics, and a rich body of literature has been generated from decades of study on the topic. Here we present a review of the works which we consider most relevant to our developed method and from which we took inspiration in our approach.

A. Feature Based Visual SLAM

Since its inception, ORB-SLAM2 [1], remains one of the most widely adopted and complete feature based SLAM systems, demonstrating that a bundle adjustment approach can

This work was supported by NASA grant NNX16AL08G and by the National Science Foundation under grants IIS-1830660 and IIS-1830500.

[1]Gideon Billings is with Department of Naval Architecture and Marine Engineering, University of Michigan, 2600 Draper Dr. Ann Arbor, MI 48109, USA gidobot@umich.edu

[2]Richard Camilli is with Applied Ocean Physics and Engineering, Woods Hole Oceanographic Institution, Deep Submergence Laboratory Woods Hole, MA 02543, USA

[3]Matthew Johnson-Roberson is with The Robotics Institute, Carnegie Mellon University, 5000 Forbes Avenue Pittsburgh, PA 15213-3890, USA

[4]https://github.com/gidobot/UWslam_dataset

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2023, London, UK. Cite as RA-L paper.

attain more accurate camera localization than direct methods or ICP, with the advantage of being less computationally expensive. Given the established robustness of ORB-SLAM2 across a variety of applications and camera systems, the efficient computational performance based on a parallel thread architecture, and the demonstrated accuracy of keyframe based bundle adjustment for pose estimation, we chose to develop our method based on this framework.

CoSLAM [2] proposed an innovative solution for fusing multiple synchronized but independently moving monocular cameras into a single framework that can also differentiate between dynamic and static feature points. We took inspiration from this approach in our method design, with the key differences being our use of stereo features to constrain the map scale, our fusion of independent hybrid camera frames into the same map (i.e. the manipulator mounted fisheye camera and a vehicle mounted perspective stereo camera), and the specific adaptations of our method to underwater environments.

B. Underwater SLAM

Significant progress has been made in underwater vision applied to large scale survey reconstructions [3], terrain aided navigation [4]–[6], and ship hull inspection [7]. However, dense scene reconstruction methods generally process the image data offline, and methods designed for navigation generally provide very sparse feature maps if any. In contrast, our method emphasises real-time scene mapping, suitable for natural seafloor environments, that is robust to underwater visual effects and provides an optimized feature map and camera pose graph that can underlie dense reconstruction methods.

Negre *et al.* [4] proposed a stereo based SLAM method specifically designed for operating in underwater feature-poor environments. The map is constructed as a pose graph connecting to feature clusters. For inter-frame pose estimation of non keyframes, they used the VISO2 stereo odometer [8], which they found to perform better than the tracking stage in ORB-SLAM. For detecting loop closures, this method generated a HALOC [9] signature for each feature cluster, which can be efficiently matched across very large image sets and does not require a prior training step like a bag of words representation. This work informed our choice of using a modified version of VISO2 for the initial inter-frame pose estimations. While their method was tailored specifically to the problem of localization through the optimization of keypoint cluster locations, our method, based on ORB-SLAM2, optimizes the location of the individual map points, which is desirable for scene reconstruction. Hidalgo *et al.* [6] studied off the shelf monocular ORB-SLAM applied in different shallow oceanic underwater environments. Their results showed that ORB-SLAM performed well in structured or feature rich environments with adequate lighting and low flickering. However, ORB-SLAM performed poorly in areas with highly dynamic lighting, large numbers of moving objects, or low textured regions such as sand beds.

C. Kinematics in SLAM

Prior research has utilized eye-in-hand based SLAM, where a camera is mounted near the end effector of a manipulator. ARM-SLAM [10] used a Kinect depth sensor mounted on a manipulator with a fixed base to capture point clouds of the scene and fuse them into a reconstruction using a method based on Kinect Fusion. SKCLAM [11] used feature based pose tracking with an RGB-D camera on the endeffector to calibrate the full kinematic parameters of an industrial manipulator with a fixed base. Point clouds from the RGB-D camera were integrated to construct a 3D map. Chen *et al.* [12] used ORB-SLAM3 and a stereo camera on a mobile manipulator to map an orchard. Unlike these prior works, our method fuses features from both an independent manipulator mounted fisheye camera and a vehicle mounted stereo in a common feature graph. We use a monocular camera on the wrist rather than relying on a depth sensor, which would be very bulky to fit in a pressure rated housing for mounting on the manipulator. Das *et al.* [13] proposed a method for calibrating a dynamic camera cluster, where one camera is articulated with respect to the other cameras in the system. They demonstrated multi-camera SLAM with one camera mounted on a pan-tilt unit, assuming accurate calibration of the pan-tilt unit's extrinsics. In contrast, our method is demonstrated with the manipulator camera having 5-DoF actuation using very high baseline to the vehicle camera, and without relying on accurate extrinsic measurement of the articulated camera. To our knowledge, this is the first successful demonstration of an eye-in-hand SLAM method on mobile underwater manipulator platforms in natural deep ocean environments.

III. Method

In this section, we highlight the innovations made to adapt the ORB-SLAM2 system to SIFT features, the underwater environment, and the hybrid imagery. For details on the system architecture that remain unchanged from ORB-SLAM2, we defer the reader to [1].

Figure 1 shows a high level block diagram of the hybrid SLAM system, where our method retains the same four threaded architecture as the original ORB-SLAM2. Crucial modifications were made in the tracking thread, which follows the top horizontal flow of the diagram, with separate functional flow branches for stereo and monocular fisheye frames. Both stereo and fisheye frames share a common keyframe representation which is processed through the local mapping, loop closing, and full bundle adjustment threads. The core of our system is the feature based stereo mapping framework, which can be operated stand-alone or in a hybrid mode, where frames from an independently moving fisheye camera are fused into the same map.

The constructed map is represented as a covisibilty graph of optimized keyframe and keypoint poses, with factors between keyframes formed through common keypoint observations. Like ORB-SLAM2, the covisibility graph is used to retrieve a local neighborhood of keypoints for the tracking and local mapping stages and forms the graph structure for the bundle adjustment optimizations. A minimum spanning tree is also maintained, which connects every keyframe to the neighbor with the maximum number of shared keypoint observations. The spanning tree is used to propagate keyframe pose optimizations from full bundle adjustment to new keyframes

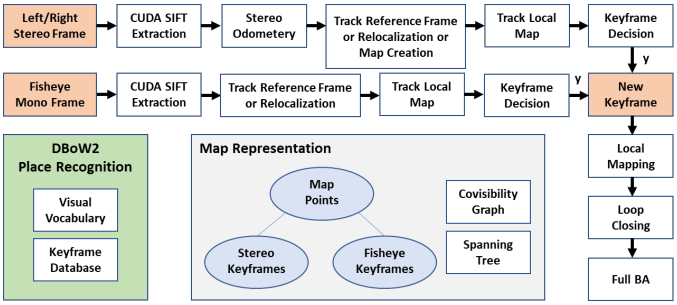


Fig. 1: System block diagram

that were not included during the optimization. A DBoW2 module [14], that we adapted to SIFT features, is used for place recognition during relocalization and loop closing.

A. Hybrid Camera System

The hybrid camera system is specifically tailored to mobile manipulator systems, with a stereo camera mounted on the vehicle and an independent but synchronized fisheye camera mounted on the manipulator wrist. In our evaluations, the stereo pair uses a pinhole camera model, calibrated with the ROS [15] camera calibration package, and the fisheye camera uses the Kannala-Brandt [16] model, calibrated with the Kalibr toolbox [17]. Underwater checkerboard imagery was collected for both cameras to perform the calibrations. We adapt the camera model code from ORB-SLAM3 [18] to support our hybrid camera system. In our notation, we represent the camera projection function as $\pi_{(\cdot)}$, where π_m and π_s are the left monocular stereo camera and rectified stereo projection functions respectively as defined in [1] and π_f is the calibrated fisheye projection function from [16].

B. Feature Representation

While ORB-SLAM2 uses ORB [19] features, ORB performs poorly in many underwater environments compared to other feature types. We conducted an analysis of the matching performance of different feature types in the underwater domain, presented in the results section, which motivated our choice of the SIFT [20] feature for our system. We adopted CudaSIFT [21], which is currently one of the fastest GPU accelerated SIFT implementations, for real-time feature extraction.

C. System Initialization

On system startup, the first keyframe K_0 is created from the first stereo frame $F_{s,0}$ that retains at least 8% of the maximum number of features N extracted from the images. K_0 is set as the origin of the initial map, which is constructed from all of the stereo keypoints. After the map is initialized, new keyframes are added from both the stereo and monocular fisheye frames, with the map scale constrained by the initial and new stereo map points.

D. Stereo Odometery

Similar to [4], we found that the tracking stage of ORB-SLAM2 failed on our underwater datasets, even when adapted to SIFT features. A considerable limitation of the ORB-SLAM2 tracking stage is a constant velocity model, which has poor accuracy at the low frame rates typical for underwater imaging systems. Negre *et al.* [4] used the VISO2 stereo odometer for initial frame pose estimation. We took inspiration from this and also adopted VISO2 for our system. However, we found that off-the-shelf VISO2 failed to track our underwater stereo dataset, due to poor performance of the simple blob and corner response features, described in the Sobel operator space. We modified VISO2 to use CudaSIFT features, which are extracted once for each image and then propagated through the rest of the SLAM pipeline for efficient computation. While the original VISO2 implementation used a search window to circularly match features across the current and previous stereo pair, we use GPU accelerated brute force matching, followed by circular filtering for improved computational performance. In this scheme, brute force matching is applied between the *previous left* and *previous right* frames, *previous right* and *current right* frames, *current right* and *current left* frames, and *current left* and *previous left* frames. A feature is accepted only if the same feature is matched across all image pairs in a circular fashion. Like in the original VISO2 implementation, feature matches between a *left* and *right* stereo image pair are further filtered by an epipolar constraint of 1 pixel error tolerance. However, we found the outlier removal step of the original VISO2 by 2d Delaunay triangulation to be too restrictive in high rugosity coral reef imagery, resulting in the filtering of many correct feature correspondences. Through extensive experimentation, we found the circular matching and epipolar constrained filtering steps were sufficient for removing the majority of outliers before processing the matches through the ego-motion estimation stage. The output transform $\widetilde{T}_{s,i}$ of the odometer is the estimated ego-motion of the current stereo frame $F_{s,i}$ with respect to the previous stereo frame $F_{s,i-1}$.

E. Tracking

Given an initialized system, the current **stereo frame** $F_{s,i}$ is processed through the tracking stage in the following steps:

1) Process $F_{s,i}$ through the odometer to extract CudaSIFT features and estimate the ego-motion $\widetilde{T}_{s,i}$ with respect to the previous stereo frame.
2) Initialize pose of $F_{s,i}$ as $T_{s,i} = \widetilde{T}_{s,i} T_{s,i-1}$, where $T_{s,i} \in SE(3)$.
3) Project the map points tracked in previous frame into the current frame using projection function $x = \pi_m(T_{s,i} X)$, where X is the vector coordinate of the map point in the world reference frame, and search feature matches within window regions around the projected points in a brute force manner.
4) Formulate the iterative pose optimization:

$$\underset{T_{s,i}}{argmin} \sum_{j \in \chi} \rho\left(\left\| x^j_{(\cdot)} - \pi_{(\cdot)}(T_{s,i} X^j) \right\|^2_{\Sigma} \right) \qquad (1)$$

where $j \in \chi$ is the set of all map point matches, ρ is the robust Huber cost function, Σ is the covariance matrix associated with the map point scale, and keypoints can be monocular with $x_m^j \in R^2$ for projection π_m or stereo with $x_s^j \in R^3$ for projection π_s. Only retain inlier map point matches following the optimization.

5) If steps 3-4 fail to find enough map point correspondences, track $F_{s,i}$ to the map points tracked in the reference stereo keyframe K_s using BoW vocabulary levels to guide matching followed by pose optimization as in step 4.
6) If few map point matches are found in steps 3-5, set $T_{s,i}$ to the odometery estimated pose of step 2.
7) Search additional map point matches in a local window of keyframes that share observations of the current map point matches and may include both stereo and fisheye frames, and repeat the pose optimization of step 4.

If the stereo tracking stage fails completely, the system enters relocalization mode until tracking is recovered for a stereo frame, as in [1].

In hybrid mode, the current monocular **fisheye frame** $F_{f,i}$ is processed through the tracking stage in the following steps, but only after the current stereo frame is successfully tracked:

1) Track $F_{f,i}$ to the map points tracked in the reference fisheye keyframe K_f using BoW vocabulary levels to guide matching followed by pose optimization as in step 4 of the stereo tracking. The optimization formulation is the same as Eq. (1), with fisheye projection π_f and $x_f^j \in R^2$.
2) If step 1 fails, $F_{f,i}$ is tracked to the map points in the reference stereo keyframe K_s.
3) Search additional map point matches in a local window of keyframes, that may include both stereo and fisheye frames, and repeat the pose optimization step.

If tracking fails for the fisheye frame but not the current stereo frame, the system enters relocalization mode for only the fisheye camera, while continuing mapping of the stereo frames. In this relocalization mode, the current fisheye frame is first attempted to be matched against all keyframes in the map using the BoW place recognition to identify match candidates. If place recognition fails, tracking of the current fisheye frame is then attempted against the current stereo reference keyframe, proceeding from step 2 above. During the local mapping stage for both stereo and fisheye frames, the reference keyframe for each is updated to the keyframe that shares the most feature matches, agnostic to the type of keyframe (i.e. stereo or fisheye). When a new keyframe is inserted, it is made the reference keyframe for the next frame of the same type.

During relocalization or when the fisheye frame is tracked against the reference stereo frame, a perspective-n-point (PnP) solver is constructed to estimate an initial pose. Like [18], we adopt the Maximum Likelihood Perspective-n-Point algorithm (MLPnP) [22], which uses projective rays in the optimization that are agnostic to the camera model, in order to accurately optimize the feature correspondences between the hybrid fisheye and perspective stereo frames.

F. Inserting New Keyframes

New stereo and fisheye keyframes are decided following the same scheme as [1] for stereo and monocular keyframes respectively, with some thresholds tuned for lower framerates and higher keypoint counts. When a new keyframe is inserted, new map points are triangulated and added into the map. For each of these keypoints the maximum and minimum distances that the point can be detected in a frame are calculated based on the scale of the keypoint in the reference keyframe. With the hybrid camera system, the scale of the keypoint can be different at the same distance, depending on which type of frame observes the keypoint. We resolve this ambiguity by normalizing the keypoint scale factor by the focal length of the observing frame. This normalization enables consistent keypoint scale prediction and comparison between hybrid frames. As in [1], local bundle adjustment is performed on a set of covisible keyframes $\mathcal{K}_L$ and all points seen in those keyframes $\mathcal{P}_L$ after a new keyframe is inserted into the map, with the optimization formulated as follows:

$$\underset{X^i, T_l}{argmin} \sum_{k \in \mathcal{K}_L \cup \mathcal{K}_F} \sum_{j \in \chi_k} \rho(\left\| x_{(\cdot)}^j - \pi_{(\cdot)}(T_k X^j) \right\|_\Sigma^2) \quad (2)$$

where $i \in \mathcal{P}_L$, $l \in \mathcal{K}_L$, χ_k is the set of point matches between $\mathcal{P}_L$ and keypoints in keyframe k, and $\mathcal{K}_F$ is all other keyframes observing $\mathcal{P}_L$ but not in $\mathcal{K}_L$ which contribute to the cost but are fixed in the optimization. Keyframes can be from fisheye images, in which case the projection function is π_f, or from stereo images in which case the projection function is π_m or π_s, depending on whether the keypoint is monocular or stereo, respectively.

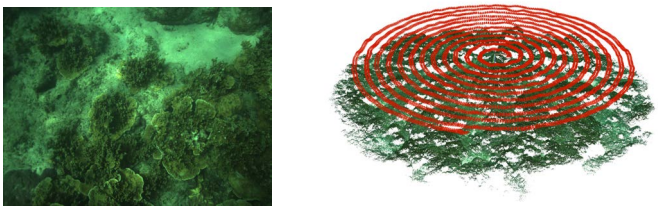
G. Loop Closing

For place recognition, we adapted DBoW2 to SIFT features and trained a million word vocabulary with ten branching factors and six levels, like the ORB vocabulary used in ORB-SLAM2. The vocabulary was trained on an extensive set of underwater imagery data, including the UWHandles and LizardIsland datasets presented in this paper, plus three large imagery datasets from the Australian Center for Field Robotics: Tasmania CSP [23], Scott Reef 25 [24], and Tasmania O'Hara 7 [25]. 2000 CudaSIFT features were extracted per image, with the images upscaled by a factor of 2 for the first scale pyramid level, the initial blur set to 1.6, and the difference of Guassian threshold set to 1.0.

After a loop closing event, a full bundle adjustment optimization is initiated, following the same optimization as Eq. (2), but including all keyframes and points in the map except for the origin stereo keyframe, which remains fixed.

H. Datasets

1) Stereo Survey Dataset: A stereo SLAM evaluation dataset was collected with a diver operated camera rig on a shallow coral reef of Lizard Island in Australia (fig. 2). The dataset was collected using a spiral survey technique [26] that fully covered a circular area of approximately 14m in diameter, with natural sunlight providing the only illumination.



(a) Left stereo image (b) COLMAP reconstruction

Fig. 2: The LizardIsland spiral survey dataset was collected with a diver operated stereo rig. The ground truth reconstruction was generated with COLMAP.

The rectified stereo image size is 1355x1002 pixels and the images were collected at 5Hz. We refer to this dataset as **LizardIsland**.

To obtain a ground truth comparison for evaluating our stereo SLAM method, we processed the dataset through COLMAP [27] to generate a sparse 3D reconstruction with optimized camera poses. COLMAP does not fix the scale during optimization, so the reconstruction was scaled in post-process to match the mean left and right stereo pair baseline to the calibrated value.

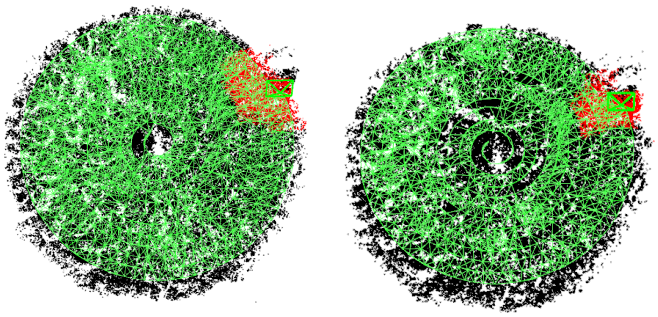
2) Hybrid Vehicle-Manipulator Dataset: During a cruise in 2019, a hybrid dataset of synchronized vehicle mounted stereo and wrist mounted fisheye imagery was collected in natural deep ocean environments of the Costa Rican continental shelf margin with the SuBastian ROV, operated by Schmidt Ocean Institute [28]. We have previously published the fisheye imagery portion of this data as the **UWHandles** dataset [29]. For this work, we have extended this dataset by further processing four environmentally unique stereo and fisheye image sequences for evaluation of our hybrid SLAM method. We refer to these sequences as Mounds1, Mounds2, Seeps1, and Seeps2. For these sequences, TagSLAM [30] was used to obtain ground truth pose estimates for the stereo and fisheye cameras, based on the detection of AprilTags [31] distributed in the scenes.

IV. Evaluation

All evaluations were run on a desktop computer with an AMD Ryzen Threadripper 2990WX CPU and an NVIDIA Titan V GPU.

A. Comparative Feature Analysis

We conducted an evaluation to determine which feature representation is best adapted to the visual degradation of underwater environments and can be robustly matched between hybrid perspective and fisheye frames with variable relative poses. We sampled every fifth hybrid frame from each of the UWHandles dataset sequences and, to reduce any bias from artificial features, we used the tracked AprilTag poses from TagSLAM to project circular masks over the tags in the image frames. For each feature type, 2000 features were extracted from each image, and the features were brute force matched across each hybrid fisheye and left stereo image pair. Lowe's ratio test was applied to remove ambiguous



(a) 4k features/frame (b) 2k features/frame

Fig. 3: Final stereo SLAM maps on the LizardIsland dataset, showing the densely connected keyframe graphs.

matches, with a ratio threshold of 0.8 for all feature types except ContextDesc, which achieved significantly improved performance with a ratio of 0.9. OpenCV's RANSAC based essential matrix fitting was used to filter the matches and recover a relative pose estimate between each fisheye and stereo frame. Table I shows the results of this evaluation. Given that an essential matrix based pose estimate does not provide scale, both the orientation and translation errors of the pose estimates were evaluated as angular errors. For translation, this error is the angular difference between the translation direction vector from the left stereo frame to the fisheye frame. The performance was evaluated using the area under the accuracy-threshold curve (AUC) with a max angular error of 180°. While most of the tested feature types were popular conventional features, we also tested two deep learned feature variants: ContextDesc and SuperPoint. We note that the learned features were used with their provided model weights and were not fine-tuned on underwater data. Of the conventional feature types, ROOT_SIFT and SIFT perform the best, achieving significantly better performance than ORB. Of the deep learned features, SuperPoint had highly variable performance across the different sequences, and the mean number of inlier matches was lower than other conventional features. Interestingly, ContextDesc performed the best overall out of all the feature types, consistently matching more than double the features of ROOT_SIFT and achieving very high AUC scores. It is noteworthy that ContextDesc uses SIFT interest points but learns the descriptor, so all of the best performing features are based on the SIFT detector. These results merit further investigation into the application of learned features for underwater vision. For our initial implementation in this work, we chose to use a highly optimized GPU accelerated implementation of SIFT, but we note that the learned descriptors of ContextDesc are 128-d, like SIFT, and are directly compatible with our entire method pipeline.

B. Stereo SLAM

The core of our system is a stereo SLAM pipeline, which must be robust to underwater environments. We used the LizardIsland survey dataset to evaluate the stereo SLAM performance. We tested ORB-SLAM2 on this dataset, both with and without loop closing enabled, but it lost track after

TABLE I: Area under accuracy-threshold curve evaluation of feature matching performance on the UWHandles underwater hybrid image sequences. Accuracy is evaluated as angular error in the predicted rotation (AUC Rot) and translation direction vector (AUC Trans) between each hybrid left stereo and fisheye image pair. Also reported is the mean number of inlier feature matches across each sequence.

Sequence		SIFT [20]	ROOT SIFT[32]	ORB [33]	SURF [34]	AKAZE [35]	CONTEXTDESC [36]	SUPERPOINT [37]
Mounds1	AUC Trans	0.949	0.936	0.856	0.877	0.922	0.970	**0.98**
	AUC Rot	0.937	0.948	0.750	0.812	0.858	0.951	**0.964**
	Mean Matches	91	101	25	34	46	**210**	74
Mounds2	AUC Trans	0.946	0.940	0.864	0.886	0.906	**0.976**	0.947
	AUC Rot	0.810	0.853	0.488	0.676	0.629	**0.959**	0.873
	Mean Matches	31	33	14	21	21	**80**	41
Seeps1	AUC Trans	0.964	0.980	0.885	0.869	0.904	**0.986**	0.938
	AUC Rot	0.944	0.965	0.745	0.770	0.811	**0.980**	0.885
	Mean Matches	64	73	23	28	43	**150**	53
Seeps2	AUC Trans	0.960	0.964	0.935	0.930	0.954	**0.974**	0.917
	AUC Rot	0.942	0.953	0.894	0.891	0.926	**0.965**	0.763
	Mean Matches	89	100	60	50	92	**146**	39

(a) Tracking (b) SLAM

Fig. 4: Stereo SLAM results (green) for the LizardIsland dataset when loop closing is disabled (a) and enabled (b), plotted against the ground truth (black).

only a few frames and was unable to relocalize. We also tested the vanilla VISO2 stereo odometer, but, even with extensive tuning, VISO2 failed to track the dataset with sensible accuracy. As a result, the stereo SLAM method of [4], specifically designed for underwater environments, also failed on our dataset, as it runs on top of VISO2. We evaluated our stereo SLAM method with both 2000 and 4000 CudaSIFT features extracted each frame, with an interest point Difference of Gaussian threshold of 1.2. Figure 4 shows the results for 4000 features, both with and without loop closing enabled, and table II gives the performance of the system in all tests. The test trajectories were aligned with the COLMAP ground truth using the Horn method [38] without scaling. The figure shows that our visual odometer based tracking method without loop closing tracks very well in the horizontal plane with most of the drift error being accumulated in the z-depth estimate. For both extracted feature counts, the table shows that a high accuracy, with less than 2cm root mean squared absolute trajectory error (RMSE), is attained by the full SLAM system with loop closing. For 4000 features per frame, the number of map points in the final map is approximately double the map point count for 2000 features per frame, showing that the system scales well with the number of extracted features. The system can achieve $>$10Hz for 2000 features per frame, which is a high framerate for underwater systems. Figure 3 shows the densely connected keyframe graphs for the final SLAM maps, demonstrating consistent loop closing between neighboring spiral trajectories.

C. Hybrid SLAM

We evaluated the performance of our hybrid SLAM system on the four sequences of the UWHandles dataset. The results are reported in table IV. For all sequences, every stereo frame was successfully registered in the SLAM map. Given that the stereo camera is mostly stationary across these image sequences, and to reduce the effect of noise in the imperfect

TABLE II: Stereo SLAM performance on the LizardIsland dataset, with the number of extracted features is set to 4000 and 2000. Performance is evaluated as RMSE of the absolute trajectory error. Results are reported with and without loop closing enabled. Also reported is the number of keyframes (KFs) and map points (MPs) in the final map and the average frame processing time in the tracking thread.

System Mode	RMSE (cm)	KFs	MPs	Avg Time (ms)
Tracking Only (4000)	49.1	-	-	94.2
Loop Closing (4000)	1.4	562	190,474	117.9
Tracking Only (2000)	58.2	-	-	55.7
Loop Closing (2000)	1.8	622	98,812	64.8

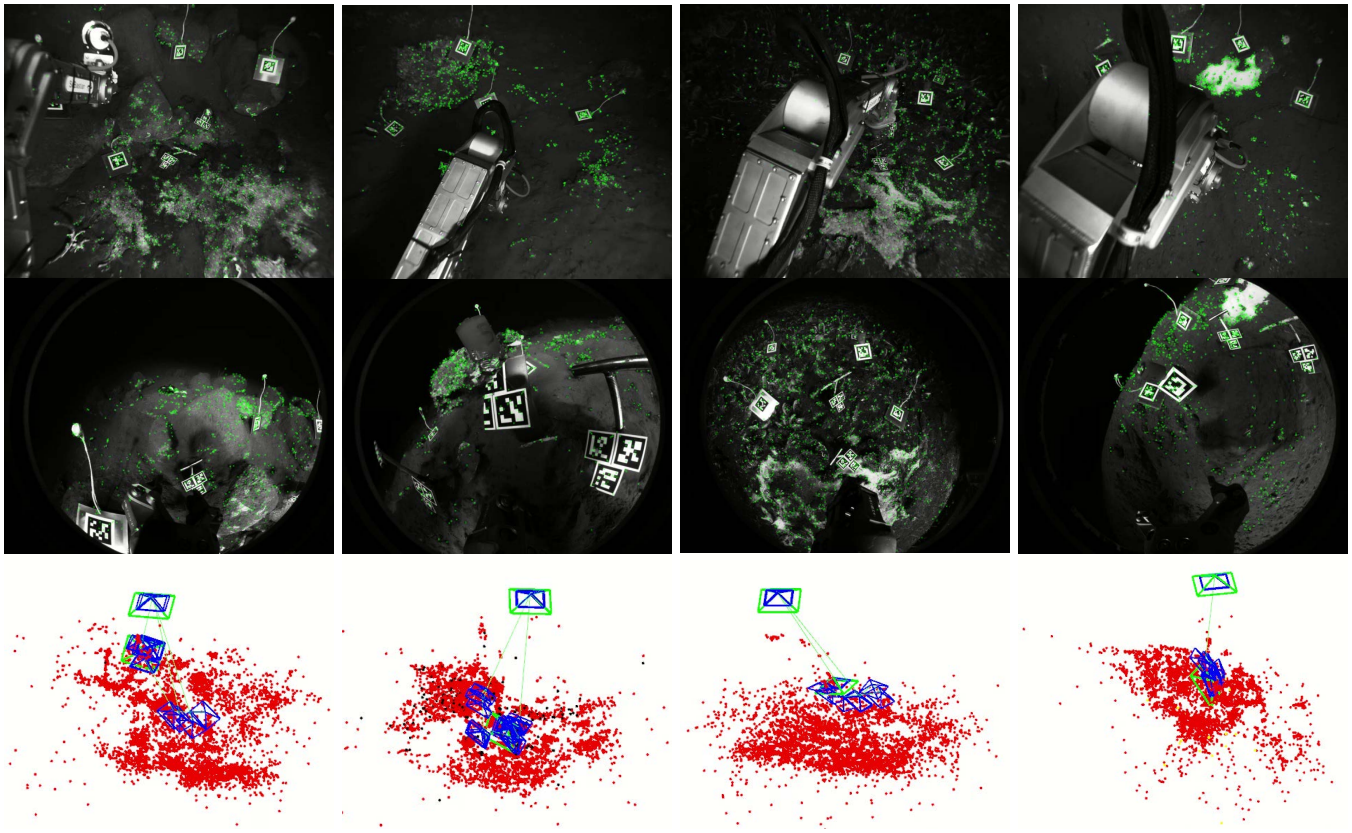
TABLE III: Hybrid SLAM timing evaluation, measured as the mean frame processing time in the tracking thread.

# Features / Frame	2k	4k	6k
Mean Time	179ms	249ms	314ms

ground truth, we evaluated the hybrid SLAM error using the relative pose estimates between the left stereo and fisheye cameras for each synchronized hybrid frame. Only hybrid frames where the fisheye frame was successfully registered into the map were included in the error evaluation. As is detailed in the table, the system generated approximately twice as many keyframes and map points when running in hybrid mode versus stereo only mode, demonstrating the ability to extend the map beyond the limited stereo camera viewpoint. Also, the majority of fisheye frames were successfully registered into the map for all sequences. Despite the sequences varying significantly in environment type, the hybrid SLAM mode is able to generate a similar amount of keyframes and map points for each sequence, and the estimated pose errors are very similar across each sequence, demonstrating our system can operate in challenging and diverse, natural seafloor environments. Figure 5 shows a frame capture from running hybrid SLAM on each of the sequences. Table III provides the timing evaluation for processing a hybrid stereo and fisheye frame pair through the tracking thread for different feature count settings. For 4000 features extracted per image, the system can easily attain 3Hz, which is the rate that the UWHandles data was collected.

V. Conclusion

We have presented a novel hybrid SLAM method, targeting deployment on underwater vehicle manipulator systems, that



(a) Mounds1 (b) Mounds2 (c) Seeps1 (d) Seeps2

Fig. 5: Snapshots of hybrid SLAM running on the UWHandles sequences. Top row is the left stereo camera frame, middle row is the manipulator mounted fisheye frame, and bottom row is the map with the keypoints and keyframes.

TABLE IV: Evaluation of hybrid SLAM on the UWHandles dataset. Error is evaluated on the estimated pose difference between the left stereo and fisheye cameras for each synchronized hybrid frame, where Δt is translation error and Δq is rotation error. The "hybrid matches" column gives the number of fisheye frames registered in the map over the total number of frames in the sequence. The error is only evaluated over the registered frames. The "KFs" column is the number of keyframes in the final map for the hybrid SLAM mode versus stereo only mode, and the "MPs" column is the same format for the number of final keypoints in the map.

Sequence	Δt mean (cm)	Δt median (cm)	Δq mean (deg)	Δq median (deg)	hybrid matches	KFs hybrid/stereo	MPs hybrid/stereo
Mounds1	2.04	2.06	0.58	0.50	652 / 783	21 / 11	4271 / 2671
Mounds2	1.38	1.22	0.98	0.82	713 / 756	24 / 11	4086 / 1773
Seeps1	2.82	2.02	1.17	0.46	1059 / 1089	24 / 11	4636 / 2847
Seeps2	2.12	1.84	1.38	0.97	778 / 802	23 / 16	4320 / 2365

can operate in real-time. The method can fuse features from both a vehicle mounted stereo camera and a manipulator mounted fisheye camera into the same map, enabling dynamic viewpoint acquisition and map extension with the manipulator mounted camera. We have demonstrated the robustness of our method on both a shallow reef stereo image survey dataset and on four hybrid image sequences captured in natural, deep seafloor environments. In this work, the system has been tested on a desktop computer, which is suitable for ROV operations, where the compute can run topside through a tethered connection with the vehicle. In future work, we will optimize the method to run on embedded GPU devices which can be deployed onboard the vehicle. Currently, the system fuses the fisheye data into the map without consideration of the manipulator state. In future work, we will also explore the formulation of a kinematic factor from the manipulator joint states between the fisheye and the stereo camera to improve registration of the fisheye camera into the map and the overall robustness of the mapping method. This factor would also enable real-time feedback for the kinematic calibration of the manipulator, which is a challenging problem for the imprecise hydraulic manipulators common for underwater systems. We will also explore the use of learned feature descriptors such as ContextDesc to improve system performance. Finally, we will explore methods for generating dense reconstructions based on the sparse feature maps and optimized camera poses to build a complete real-time scene reconstruction method for UVMSs.

REFERENCES

[1] R. Mur-Artal and J. D. Tardós, "Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras," *IEEE Transactions on Robotics*, vol. 33, no. 5, pp. 1255–1262, 2017.

[2] D. Zou and P. Tan, "Coslam: Collaborative visual slam in dynamic environments," *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 2, pp. 354–366, 2012.

[3] M. Johnson-Roberson *et al.*, "High-resolution underwater robotic vision-based mapping and 3d reconstruction for archaeology," *Journal of Field Robotics*, 2016.

[4] P. L. Negre, F. Bonin-Font, and G. Oliver, "Cluster-based loop closing detection for underwater slam in feature-poor regions," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, May 2016, pp. 2589–2595.

[5] M. Ferrera, J. Moras, P. Trouvé-Peloux, and V. Creuze, "Real-time Monocular Visual Odometry for Turbid and Dynamic Underwater Environments," *arXiv:1806.05842 [cs]*, Feb. 2020, arXiv: 1806.05842.

[6] F. Hidalgo, C. Kahlefendt, and T. Bräunl, "Monocular orb-slam application in underwater scenarios," in *2018 OCEANS-MTS/IEEE Kobe Techno-Oceans (OTO)*, IEEE, 2018, pp. 1–4.

[7] P. Ozog, M. Johnson-Roberson, and R. M. Eustice, "Mapping underwater ship hulls using a model-assisted bundle adjustment framework," *Robotics and Autonomous Systems*, vol. 87, pp. 329–347, 2017.

[8] A. Geiger, J. Ziegler, and C. Stiller, "Stereoscan: Dense 3d reconstruction in real-time," in *Intelligent Vehicles Symposium (IV)*, 2011.

[9] P. L. N. Carrasco, F. Bonin-Font, and G. Oliver-Codina, "Global image signature for visual loop-closure detection," *Autonomous Robots*, vol. 40, no. 8, pp. 1403–1417, 2016.

[10] M. Klingensmith, S. S. Sirinivasa, and M. Kaess, "Articulated robot motion for simultaneous localization and mapping (arm-slam)," *IEEE robotics and automation letters*, vol. 1, no. 2, pp. 1156–1163, 2016.

[11] J. Li, A. Ito, and Y. Maeda, "A slam-integrated kinematic calibration method for industrial manipulators with rgb-d cameras," in *2019 19th International Conference on Control, Automation and Systems (ICCAS)*, IEEE, 2019, pp. 686–689.

[12] M. Chen, Y. Tang, X. Zou, Z. Huang, H. Zhou, and S. Chen, "3d global mapping of large-scale unstructured orchard integrating eye-in-hand stereo vision and slam," *Computers and Electronics in Agriculture*, vol. 187, p. 106 237, 2021.

[13] A. Das and S. L. Waslander, "Calibration of a dynamic camera cluster for multi-camera visual slam," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2016, pp. 4637–4642.

[14] D. Gálvez-López and J. D. Tardós, "Bags of binary words for fast place recognition in image sequences," *IEEE Transactions on Robotics*, vol. 28, no. 5, pp. 1188–1197, 2012.

[15] M. Quigley *et al.*, "Ros: An open-source robot operating system," in *ICRA workshop on open source software*, Kobe, Japan, vol. 3, 2009, p. 5.

[16] J. Kannala and S. S. Brandt, "A generic camera model and calibration method for conventional, wide-angle, and fish-eye lenses," *IEEE transactions on pattern analysis and machine intelligence*, vol. 28, no. 8, pp. 1335–1340, 2006.

[17] J. Maye, P. Furgale, and R. Siegwart, "Self-supervised calibration for robotic systems," in *2013 IEEE Intelligent Vehicles Symposium (IV)*, IEEE, 2013, pp. 473–480.

[18] C. Campos, R. Elvira, J. J. G. Rodríguez, J. M. Montiel, and J. D. Tardós, "Orb-slam3: An accurate open-source library for visual, visual–inertial, and multimap slam," *IEEE Transactions on Robotics*, 2021.

[19] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "Orb: An efficient alternative to sift or surf," in *2011 International conference on computer vision*, Ieee, 2011, pp. 2564–2571.

[20] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International journal of computer vision*, vol. 60, no. 2, pp. 91–110, 2004.

[21] M. Björkman, N. Bergström, and D. Kragic, "Detecting, segmenting and tracking unknown objects using multi-label mrf inference," *Computer Vision and Image Understanding*, vol. 118, pp. 111–127, 2014.

[22] S. Urban, J. Leitloff, and S. Hinz, "Mlpnp – a real-time maximum likelihood solution to the perspective-n-point problem," *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. III-3, pp. 131–138, Jun. 2016.

[23] L. Meyer, N. Hill, P. Walsh, and N. Barrett, "Methods for the processing and scoring of auv digital imagery from south eastern tasmania," *Institute for Marine and Antarctic Studies Internal Report*, 2011.

[24] D. M. Steinberg, S. B. Williams, O. Pizarro, and M. V. Jakuba, "Towards autonomous habitat classification using gaussian mixture models," in *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, IEEE, 2010, pp. 4424–4431.

[25] D Steinberg, A. Friedman, O. Pizarro, and S. B. Williams, "A bayesian nonparametric approach to clustering data from underwater robotic surveys," in *International Symposium on Robotics Research*, Citeseer, vol. 28, 2011, pp. 1–16.

[26] O. Pizarro, A. Friedman, M. Bryson, S. B. Williams, and J. Madin, "A simple, fast, and repeatable survey method for underwater visual 3d benthic mapping and monitoring," *Ecology and Evolution*, vol. 7, no. 6, pp. 1770–1782, 2017.

[27] J. L. Schönberger and J.-M. Frahm, "Structure-from-motion revisited," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[28] P. Vrolijk *et al.*, "Using a ladder of seeps with computer decision processes to explore for and evaluate cold seeps on the costa rica active margin," *Frontiers in Earth Science*, vol. 9, p. 143, 2021.

[29] G. Billings and M. Johnson-Roberson, "Silhonet-fisheye: Adaptation of a roi based object pose estimation network to monocular fisheye images," *IEEE Robotics and Automation Letters*, vol. 5, no. 3, pp. 4241–4248, 2020.

[30] B. Pfrommer and K. Daniilidis, "Tagslam: Robust slam with fiducial markers," *arXiv preprint arXiv:1910.00679*, 2019.

[31] J. Wang and E. Olson, "Apriltag 2: Efficient and robust fiducial detection," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2016, pp. 4193–4198.

[32] R. Arandjelović and A. Zisserman, "Three things everyone should know to improve object retrieval," in *2012 IEEE Conference on Computer Vision and Pattern Recognition*, IEEE, 2012, pp. 2911–2918.

[33] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "ORB: An efficient alternative to SIFT or SURF," in *2011 International Conference on Computer Vision*, ISSN: 2380-7504, Nov. 2011, pp. 2564–2571.

[34] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, "Speeded-up robust features (surf)," *Computer vision and image understanding*, vol. 110, no. 3, pp. 346–359, 2008.

[35] P. F. Alcantarilla and T Solutions, "Fast explicit diffusion for accelerated features in nonlinear scale spaces," *IEEE Trans. Patt. Anal. Mach. Intell*, vol. 34, no. 7, pp. 1281–1298, 2011.

[36] Z. Luo *et al.*, "Contextdesc: Local descriptor augmentation with cross-modality context," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 2527–2536.

[37] D. DeTone, T. Malisiewicz, and A. Rabinovich, "Superpoint: Self-supervised interest point detection and description," in *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, 2018, pp. 224–236.

[38] B. K. Horn, "Closed-form solution of absolute orientation using unit quaternions," *Josa a*, vol. 4, no. 4, pp. 629–642, 1987.