# Real-Time Dense 3D Mapping of Underwater Environments

Weihan Wang[a*], Bharat Joshi[b*], Nathaniel Burgdorfer[a], Konstantinos Batsos[c],
Alberto Quattrini Li[d], Philippos Mordohai[a], Ioannis Rekleitis[b]

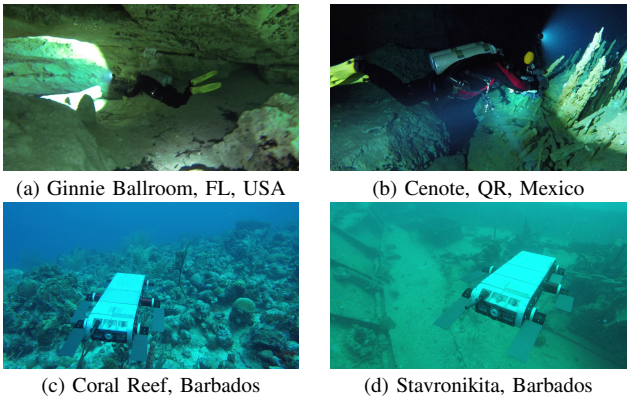
***Abstract*— This paper addresses real-time dense 3D reconstruction for a resource-constrained Autonomous Underwater Vehicle (AUV). Underwater vision-guided operations are among the most challenging as they combine 3D motion in the presence of external forces, limited visibility, and absence of global positioning. Obstacle avoidance and effective path planning require online dense reconstructions of the environment. Autonomous operation is central to environmental monitoring, marine archaeology, resource utilization, and underwater cave exploration. To address this problem, we propose to use SVIn2, a robust VIO method, together with a real-time 3D reconstruction pipeline. We provide extensive evaluation on four challenging underwater datasets. Our pipeline produces comparable reconstruction with that of COLMAP, the state-of-the-art offline 3D reconstruction method, at high frame rates on a single CPU.**

## I. INTRODUCTION

Mapping underwater environments is an important and challenging endeavor. Monitoring the coral reefs [1], exploring underwater caves [2] and recording the shape of *Cenotes* [3] have tremendous significance in our understanding and awareness of the environment. Underwater mapping is also crucial for marine archaeology, infrastructure maintenance, and during search and rescue missions. Automating mapping with Autonomous Underwater Vehicles (AUVs) reduces risks to divers, enables longer operations times and increases the frequency of mapping/exploration missions.

Unfortunately, as demonstrated in recent work on comparing numerous open-source visual and visual/inertial state estimation packages [4], [5], there are frequent failures underwater due to a variety of reasons. In contrast to above-water scenarios, GPS based localization is impossible. In addition to the traditional difficulties of vision based localization, the underwater environment is prone to rapid changes in lighting conditions, limited visibility, and loss of contrast and color information with depth.

* The first two authors have contributed equally to the paper.
[a]Stevens Institute of Technology, Hoboken, NJ, USA, 07030, {wwang103,nburgdor,pmordoha}@stevens.edu
[b]University of South Carolina, Columbia, SC, USA, 29208, bjshi@email.sc.edu, yiannisr@cse.sc.edu.
[c]Argo AI, Palo Alto, CA, USA, 94304 kbatsos@stevens.edu
[d]Dartmouth College, Hanover, NH, USA, 03755 alberto.quattrini.li@dartmouth.edu
This research has been supported in part by the National Science Foundation under grants 1943205, 1919647, 2024741, 2024541 and 2024653. The authors would also like to acknowledge the help of the Woodville Karst Plain Project (WKPP) and El Centro Investigador del Sistema Acuífero de Quintana Roo A.C. (CINDAQ) in collecting data, providing access to challenging underwater caves, and mentoring us in underwater cave exploration. K. Batsos's contributions were made while at Stevens.



(a) Ginnie Ballroom, FL, USA (b) Cenote, QR, Mexico

(c) Coral Reef, Barbados (d) Stavronikita, Barbados

Fig. 1: Datasets used in our experiments: (a) and (b) collected using a custom sensor suite. (c) and (d) collected using the Aqua2 AUV.

In this paper, we focus on real-time, scalable, detailed 3D mapping. These goals must be accomplished on a computational platform suitable for deployment on an AUV. Our software requires only a CPU and follows a pipeline architecture that incurs an almost constant computational load by processing fixed-length segments of the data at a time. The proposed approach includes: robust real-time camera pose estimation using SVIn2 [6] which fuses information from cameras, IMU, sonar, and a pressure sensor; two-stage depth map estimation based on multi-threaded CPU-based stereo matching followed by visibility-based depth map fusion; and colored point cloud generation.

We conducted a thorough evaluation comparing our method to COLMAP [7], [8], which is the state-of-the-art open-source 3D reconstruction framework. Our evaluation considers run-time, depth map estimation, and dense reconstruction on four challenging underwater datasets.

## II. RELATED WORK

The Structure-from-Motion (SfM) and Simultaneous Localization and Mapping (SLAM) literature is vast. Here, we focus on approaches tailored for underwater deployment. State estimation underwater is challenging due to color saturation, floating particulates, and limited visibility [5]. Vargas *et al*. [9] proposed robust visual SLAM underwater leveraging acoustic, inertial and altimeter/depth sensors in addition to cameras. Tightly coupled fusion of visual, inertial, and pressure sensors using forward and backward IMU preintegration is discussed in [10]. We use the approach of Rahman *et al*. [6] to obtain robust camera pose estimates by fusing visual, inertial, sonar and pressure information in real time. Beall *et al*. [11] demonstrate accurate sparse 3D

reconstruction of underwater structures from stereo videos. Joshi *et al*. [12] augment a visual SLAM algorithm so that, after loop closures, the map is deformed to preserve the relative pose between each point and its attached keyframes.

Among the few authors that tackle dense underwater stereo, Queiroz-Neto et al. [13] model light propagation to overcome poor contrast and illumination. Relevant to our method are general-purpose dense 3D reconstruction algorithms [14]–[23] operating in online mode on the frames of one or more video streams. They can achieve high throughput, in many cases by leveraging powerful GPUs. They have been evaluated qualitatively since appropriate benchmarks with ground truth, other than KITTI [24], are not available. Recently, learning-based approaches [25]–[30] have shown promising results at high frame rates. Considering the lack of ground truth data from relevant domains, training these algorithms in supervised mode is practically impossible.

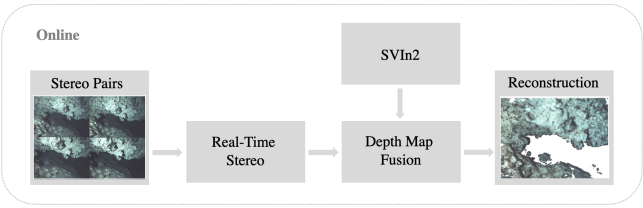



Fig. 2: A diagram of the components of the proposed pipeline. Given a pair of stereo images, the pose of the camera and a depth map can be estimated in parallel. Once both become available, depth maps can be fused to generate the final point cloud.

## III. Proposed Approach

Our approach to 3D reconstruction relies on two parallel components: (1) a multi-sensor SLAM system, SVIn2 [6], and (2) a real-time dense 3D mapping system, as shown in Fig. 2. In this paper, we focus on the latter, as well as a comprehensive comparison of our online system with COLMAP [7], [8]. Our approach requires a calibrated stereo camera rig, an IMU, and a pressure sensor to provide the necessary inputs.

### A. Pose Estimation

Robot pose estimation relies on our previous work, SVIn2 [6], [31], a tightly-coupled keyframe-based SLAM system that fuses data from cameras, an IMU, a scanning profiling sonar, and a water pressure sensor. We have demonstrated that SVIn2 performs well in underwater environments by explicitly addressing drift, loss of localization and poor illumination via robust initialization, loop-closing, and relocalization capabilities.

### B. Depth Map Estimation

The stereo matching module estimates depth for every pixel of the left image of a stereo pair of images. The cameras are placed with their image planes approximately parallel facilitating rectification in software via a pair of homographies [32] estimated using a calibration checkerboard. This configuration allows the use of very fast algorithms for dense correspondence estimation that operate on the epipolar lines, which are now horizontal. It also allows us to obtain metric depth at each time instant, due to the known baseline between the cameras, regardless of the accuracy in camera pose estimation.

We are able to process stereo pairs at high throughput leveraging multi-threaded CPU implementations (OpenMP) without relying on GPUs. We accomplish this by carefully designing every step of the stereo matching process. To our knowledge, ours is the fastest publicly available implementation of stereo matching[1].

**Matching cost computation.** Stereo matching operates by assigning a cost or score [33] to each possible disparity[2] that can be assigned to a given pixel of the reference image, typically the left. In this paper, we experimented with the Sum of Absolute Differences (SAD), which is a cost, and Normalized Cross Correlation (NCC), which is a score. Both are computed in matching windows centered around the pixels under consideration.

We use *integral images* to compute sums in rectangular sub-images in constant time [34] regardless of the matching window size. Further, memory coalescing and memoization are crucial for achieving high throughput. By ensuring that our implementation accesses nearby memory addresses in a predictable way, the compiler directly generates assembly code that takes advantage of the processor's vector instruction set. Moreover, coalesced memory accesses utilize cache more effectively leading to high global load and store efficiency. Memoization leads to further acceleration since it avoids redundant computations by retrieving previously computed results when the same input values re-appear. This happens frequently since image intensities are stored at one-byte precision.

**Optimization.** The output of the previous stage is a *cost volume* with dimensions equal to the width and height of the images and the number of disparity candidates for every pixel. The fastest way to obtain a disparity map from the cost volume is by selecting the disparity with the smallest cost for each pixel. To obtain higher accuracy the cost volume can be optimized by the widely-used Semi-Global Matching algorithm (SGM) [35].

SGM is used for extracting a disparity map that approximately optimizes a global energy function defined over 2D image neighborhood by combining multiple 1D minimization problems. Briefly, SGM favors constant disparity, imposes a small penalty for disparity differences equal to 1 between adjacent pixels along the minimization direction, and imposes a larger penalty for large discontinuities. This has the effect of allowing slanted surfaces and reducing the number of jumps in disparity. Here, we integrate the rSGM implementation of Spangenberg et al. [36] opting for the variant that considers only four 1D sub-problems to favor speed over a small loss of accuracy compared to the use of more directions. Disparity

[1] `https://github.com/kbatsos/Real-Time-Stereo`

[2] Disparity is defined as the difference between the horizontal coordinates of two potentially corresponding pixels in the same epipolar line (scanline) in the left and right image. Disparity is inversely proportional to depth.

is converted to depth using the known baseline and focal length of the cameras. Depth is then refined to sub-pixel precision by fitting a parabola in the vicinity of the minimum optimized cost [37].

**Confidence estimation.** Depth map fusion benefits from confidence values conveying which depths are more reliable. We attach a confidence to each depth after SGM using the PKRN measure [38], which is the ratio of the second smallest over the smallest cost for that pixel in the cost volume. PKRN can be computed in negligible time during the final disparity selection step, but it has been shown to be effective in discriminating reliable from unreliable depths.

### C. Depth Map Fusion

Depth maps estimated by the stereo matching module are reasonably accurate but contain noise due to lack of texture, occlusion, and motion blur. Under the assumption that these errors are not systematic and do not form hallucinated surfaces, but rather generate scattered noise that does not repeatedly appear at the same locations from frame to frame, we can improve the depth maps by fusing them. This scheme works as long as individual overlapping depth maps produce either relatively consistent 3D estimates for the same part of the scene or uncorrelated noisy estimates. Consensus among depths and violations of visibility constraints indicate which depths are correct and which are likely to be outliers.

Our approach operates in pipeline mode keeping a small number of recent depth maps in memory at a given time to support a single fusion operation. This decomposition allows the pipeline to operate at constant speed regardless of the size of the scenes and the number of frames that have been collected, and has been adopted by previous video-based 3D reconstruction systems [14], [22], [39]. At each time step, a depth map in the middle of the sliding window is used as reference and the remaining depth maps are rendered onto it along with the corresponding confidence maps. At the next time step, the oldest depth map in the sliding window is dropped and it is replaced by the most recently computed depth map. As discussed in Section IV, the sliding window we use is very short to keep latency low.

Similar to our previous work [40], [41], the input for computing a fused depth map for a given *reference view* is a set of $N_f$ depth maps and the corresponding confidence maps. The fusion process begins by rendering the depth and confidence maps to the reference view yielding a new set of $N_f$ depth and confidence maps from the perspective of the reference view. (When multiple depth estimates from input depth map $D_i$ fall on the same pixel of the rendered depth map $D_i^r$ the one nearest the camera is kept, and its corresponding confidence is kept in the rendered confidence map $C_i^r$.) At the end of the rendering stage, we have at most $N_f$ depth candidates per pixel of the reference view. For each of these depth candidates $d_j$ we accumulate *support* and *visibility violations*. Support comes from other depth candidates for the same pixel that are within a small distance of $d_j$, typically specified as a small fraction of the depth range $\epsilon$. $d_j$ is then replaced by the weighted average of the supporting depths, with confidence values serving as weights. The confidence of the blended depth estimate $d_j$ is set equal to the sum of the supporting confidences. See Fig. 3 (left).

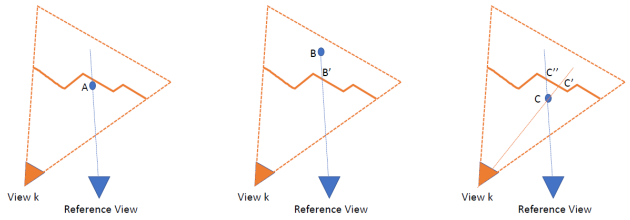



Fig. 3: Illustration of depth map fusion. Points A, B and C are depth candidates for pixels of the reference view, estimated directly or rendered to it from other views. The orange line marks the cross section of the surface estimated by view $k$. Left: point A is supported by the orange surface. Middle: point B is occluded by B', which is in front of B in the ray of the reference view. Right: point C violates the free space of C' on the ray of view $k$. (Note that there is no conflict between C and C" - imagine a tree in front of a wall.)

There are two types of violations of visibility constraints: occlusions and free space violations. An *occlusion* occurs when $d_j$ appears behind a rendered depth map from view $k$, $D_k^r$ on the ray of the reference view, as in Fig. 3 (middle), while a *free space violation* occurs when $d_j$ appears in front of an input depth map $D_l$ on the ray of view $l$, as in Fig. 3 (right). For each detected violation, we penalize the confidence of $d_j$ by subtracting the confidence of the conflicting depth estimate. We assign to each pixel the depth with the highest fused confidence, after adding support and subtracting conflicts. In practice, we threshold confidence to reject outliers. The fusion process is independent across pixels, and is thus parallelizable. Rendering depth candidates to the original depth maps to detect free space violations is the most expensive step.

## IV. Experimental Results

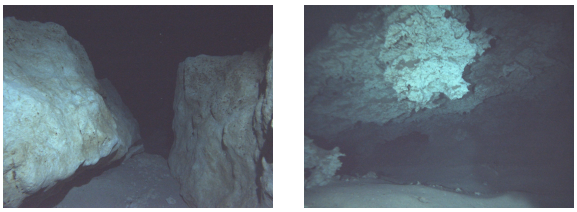
In this section, we present experimental results on challenging underwater sequences. We evaluate both the sparse and dense components of the 3D reconstruction system and compare them to the corresponding aspects of COLMAP, which operates much slower in offline mode.

### A. Datasets

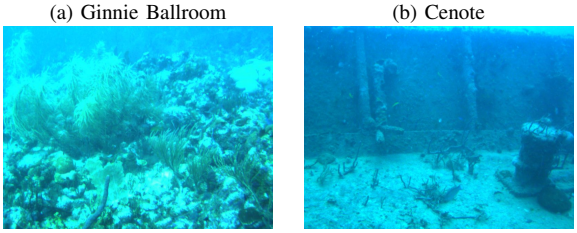
The datasets used in this paper were collected using a custom made sensor suite [42]; see Fig. 1(a) and (b), and an Aqua2 robot [43]; see Fig. 1(c) and (d). Both devices are equipped with two iDS USB3 uEye cameras as stereo pair, a MicroStrain 3DM-GX4-15 IMU, and a Bluerobotics Bar30 water pressure sensor. The stereo images are recorded at 15 Hz; inertial data at 100 Hz; and water pressure at 1 Hz using onboard Intel NUC. A video light is attached to the sensor suite unit to provide artificial illumination of the scene. The Aqua2 AUV is a hexapod robot which utilizes the motion from six flippers, each actuated independently by an electric motor, to move in 3D.

We perform experiments on four datasets:

- **Ginnie Ballroom, Gennie Springs, FL, USA**



(a) Ginnie Ballroom (b) Cenote



(c) Coral Reef (d) Stavronikita

Fig. 4: Sample images from the underwater datasets.

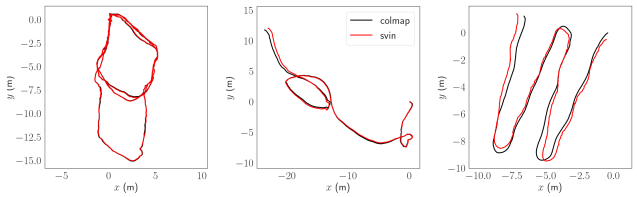
- **Cenote, QR, Mexico**
- **Coral Reef, Barbados**
- **Stavronikita Shipwreck, Barbados**

These underwater datasets present substantial challenges for 3D reconstruction. The *Ginnie Ballroom* and *Cenote* datasets are collected using a custom sensor suite [42] operated by a diver, Fig. 4(a) and (b); while the *Coral Reef* and *Stavronikita Shipwreck* datasets are collected using the Aqua2 AUV while performing a lawnmower pattern over the scene, Fig. 4(c) and (d). These datasets form a diverse set of underwater environments, including open, flat areas of the seafloor, dense and richly structured shipwrecks, and enclosed caverns with relatively uniform surfaces. In the *Coral Reef* and *Stavronikita Shipwreck* datasets, we can rely on natural light to illuminate the scene, but for the *Ginnie Ballroom* and *Cenote* datasets, we must rely on artificial illumination from the sensor suite.

### B. COLMAP

In this section, we briefly describe COLMAP [7], [8], a state-of-the-art, open-source Structure-from-Motion and dense 3D reconstruction software used as a baseline in our experiments. The sparse component of COLMAP takes as input an unordered image collection, extracts and matches features, builds the scene graph and performs bundle adjustment. All steps are carefully implemented to improve robustness, accuracy, and completeness. The dense reconstruction component jointly estimates depth and surface normals using a PatchMatch Multi-View Stereo (MVS) algorithm with pixel-wise view selection, photometric and geometric priors. The photometric and geometric depth maps are fused based on multi-view geometric consistency to produce dense reconstruction.

In our experiments, we pass the keyframes obtained from SVIn2 to COLMAP to obtain bundle-adjusted camera poses, dense depth maps and point clouds. Unless otherwise noted, all dense reconstruction experiments are performed using these camera poses as input.




(a) Ginnie Ballroom (b) Cenote (c) Coral Reef

Fig. 5: Camera trajectories, estimated by SVIn2 [44] and COLMAP [7] after *sim3* alignment.

### C. Camera Pose Estimation Results

We compare two drastically different approaches for estimating the trajectory: COLMAP which operates offline and performs global bundle adjustment and SVIn2 which runs online performing SLAM. Due to the known baseline of the stereo camera and the inertial data, SVIn2 produces trajectories with correct scale. However, these trajectories may suffer from drift, especially when loop closure opportunities are unavailable. Due to global bundle adjustment, COLMAP trajectories are more accurate, but scale may drift during optimization. During post-processing, the scale discrepancy is corrected by scaling the camera poses from COLMAP using the known stereo baseline.

To enable comparisons, we provide the keyframes selected by SVIn2 as input to COLMAP and align SVIn2 trajectory with COLMAP using *sim3* alignment [45]. We use the root mean square Absolute Trajectory Error (ATE) metric [46] to compare the trajectories. Note that in the absence of ground truth, we can only measure the discrepancy between the COLMAP and SVIn2. Fig. 5 shows the trajectories estimated by COLMAP and SVIn2 after *sim3* alignment, while Table I shows the root mean square ATE in meters. The trajectories are in general consistent up to a few cm in terms of ATE.

We were unable to obtain a complete trajectory on the Stavronikita dataset using SVIn2 due to segments in which the AUV maneuvered over the side of the wreck, thus facing open water, causing SVIn2 to lose track. COLMAP, on the other hand, attempts to match features over all images and registers a lot of the frames bridging gaps. We only use the COLMAP trajectory for Stavronikita in the remainder.

### D. Dense Reconstruction

Stereo matching is performed on 800×600 images with 100 disparity levels using the Sum of Absolute Differences (SAD) as the matching function in 3×3 windows. We generate a depth map for every frame after SGM optimization and sub-pixel fitting and fuse three depth maps using the middle frame as reference. We set the support radius during depth

TABLE I: Comparison of the SVIn2 and COLMAP trajectories based on root mean square ATE in meters.

| Dataset | length[m] | rmse |
|---|---|---|
| Ginnie Ballroom | 98.73 | 0.07 |
| Cenote | 67.24 | 0.19 |
| Coral Reef | 45.52 | 0.39 |
| Stavronikita | 106.30 | N/A |

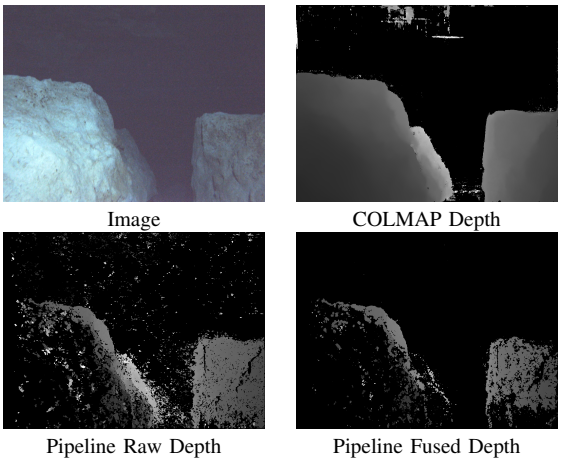



Fig. 6: **Ginnie Ballroom dataset**. Example depth maps from COLMAP and Pipeline.

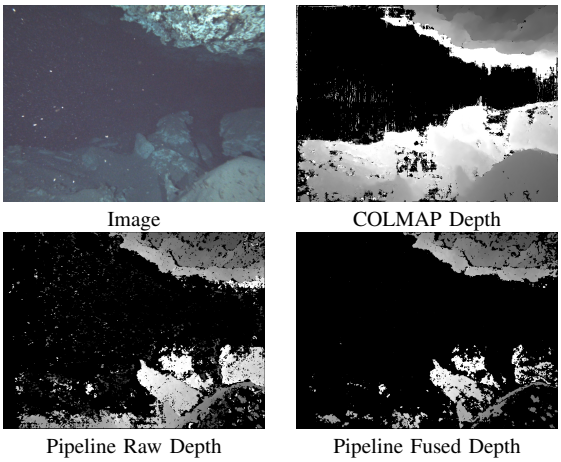



Fig. 7: **Cenote dataset**. Example depth maps from COLMAP and Pipeline.

map fusion, $\epsilon$, to 0.04 and the threshold on fused confidence for outlier rejection, $C_{\text{thres}}$, to 0.5.

**Evaluation**. In the absence of ground truth and of any practical system for acquiring ground truth underwater, we evaluate our online reconstruction pipeline by comparing the output depth maps and 3D point cloud with those generated offline by COLMAP. It should be noted that the latter are not perfect, but benefit from global optimization.

**Run-time comparison**. In the four datasets (Ginnie Ballroom, Cenote, Coral Reef, Stavronikita) that we consider, there are 13671, 3519, 3207, 8824 stereo pairs and 1519, 1401, 631 and 897 keyframes selected by SVIn2, respectively. We ran the pipeline on an Intel i7-10700K desktop with 32GB memory, while COLMAP is run on desktop with Intel i9-12900K CPU, 32GB memory, and an NVIDIA GeForce RTX 2080Ti GPU. Comparison of the run-times on the keyframes between COLMAP and our pipeline is listed in Table III. The pipeline outperforms COLMAP significantly with respect to the speed; it achieves a throughput ranging between 2.8 and 10.2 fps on the four datasets, whereas COLMAP runs at 0.05 - 0.3 fps, when considering all input frames (not just keyframes), which are a true measure of the length of the input videos.

TABLE II: Depth map evaluation between COLMAP and Pipeline.

| **Dataset** | **Depth Maps Comparison** | |
|---|---|---|
| | median-of-medians (m) | MAE (m) |
| Ginnie Ballroom | 0.063 | 0.079 |
| Cenote | 0.105 | 0.134 |
| Coral Reef | 0.327 | 0.402 |
| Stavronikita | 0.336 | 0.399 |

**2D Metrics**. Both COLMAP and the pipeline produce dense depth maps, which, however, contain some holes without depth estimates due to filtering. Avoiding to generate noise, especially since the same surface may be reconstructed from a different view, is desirable. Therefore, we only compare depth estimates that exist in both depth maps by recording the absolute depth error (AE). We then compute the mean (MAE) and median of these errors per depth map, as well as the MAE over an entire dataset and the median-of-medians as an approximation of the overall median over valid depths.

**3D Metrics**. To compare the point cloud reconstruction from the Pipeline with the offline reconstruction generated by COLMAP, we utilize Chamfer distance metrics between the two models. We refer to the pipeline point cloud as the *source* and the COLMAP point cloud as the *target*. *Accuracy* is the mean Chamfer distance from every point in the source model to the closest point in the target model. *Completeness* is the mean Chamfer distance from every point in the target model to the closest point in the source model. *Precision* and *Recall* are the percentage of points that have a Chamfer distance to the other set below a threshold; *Precision* is measured from *source-to-target* and *Recall* is measured from *target-to-source*. For our evaluation, we set the threshold to 0.1m.

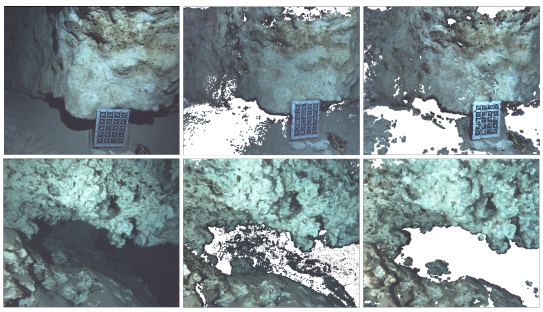
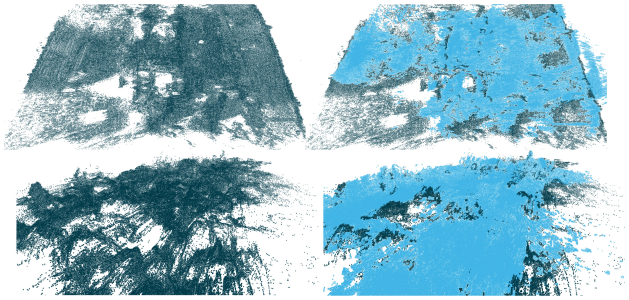



Fig. 8: Ginnie Ballroom (top), and Cenote (bottom).

**Reconstruction results** Table II summarizes the comparison of the *fused* depth maps from the pipeline with the geometric depth maps generated by COLMAP. The depth maps produced from COLMAP and the pipeline are similar for Gennie Ballroom and Cenote datasets with *median-of-medians* and MAE in the 0.06 - 0.14 m range. The depth maps for the Coral Reef and Stavronikita datasets differ more, with both metrics in the 0.3 - 0.4 m range. Fig. 6 and Fig. 7 show qualitative results, including raw and fused depth maps from the pipeline and depth maps from COLMAP. The COLMAP depth maps, while dense, contain noisy artifacts in open regions of the scene, typically resulting from floating

TABLE III: Run-time comparison between COLMAP and Pipeline.

| Dataset | COLMAP | | | | Pipeline | | | |
|---|---|---|---|---|---|---|---|---|
| | stereo pairs | vertices | MVS (min) | total time (min) | vertices | Stereo (ms/frame) | Fusion (ms/frame) | total time (min) |
| Ginnie Ballroom | 1519 | 8,056,361 | 607.61 | 858.10 | 54,034,304 | 398.88 | 374.93 | 22.36 |
| Cenote | 1401 | 8,909,774 | 568.52 | 1250.53 | 91,036,005 | 360.49 | 418.71 | 20.92 |
| Coral reef | 631 | 3,833,107 | 254.31 | 337.99 | 66,733,182 | 369.35 | 677.65 | 12.44 |
| Stavronikita | 897 | 4,552,326 | 367.17 | 482.57 | 29,851,344 | 323.57 | 553.37 | 14.95 |



COLMAP COLMAP+Pipeline

Fig. 9: **Stavronikita Shipwreck** (top). **Coral Reef** (bottom). COLMAP reconstruction results overlayed with the Pipeline reconstruction.

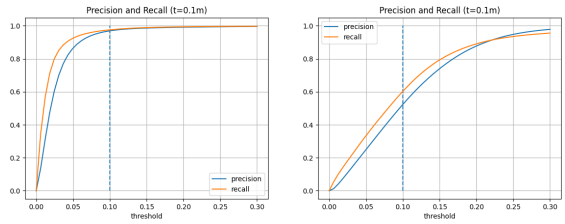
particles and changes in illumination. The raw depth maps generated by the pipeline are also noisy, but the fusion step removes noise and produces depths corresponding to surfaces in the environment.

In Table IV, we show a quantitative comparison between pipeline and COLMAP point clouds. For the Ginnie Ballroom and Cenote datasets, the pipeline generates models that are very close to COLMAP with both accuracy and completeness less than 0.05 m, and precision and recall over 90%. Fig. 8 shows a local perspective of the generated point clouds from the Ginnie Ballroom and Cenote datasets. Much of the detail is preserved in the pipeline reconstruction, with sparsity in areas of open water and smooth surfaces. The reconstruction results for Stavronikita and Coral Reef are inferior with accuracy and completeness less than 0.15 m, and precision and recall in the range 40%-60%. This can be explained in part by the sparsity of our models for these two datasets. The point clouds of the Stavronikita and Coral Reef datasets can be seen in Fig. 9 where the Pipeline model is overlayed on the COLMAP point cloud. Fig. 10 shows examples of the Precision and Recall curves for the Ginnie Ballroom and the Coral Reef as a function of the threshold. (The default 0.1 m are marked by the vertical line.)

It should be noted that the Ginnie Ballroom and Cenote datasets are collected by slowly moving human diver with artificial illumination. The Coral Reef and Stavronikita datasets are collected by a fast-moving Aqua2 AUV in deep ocean without any artificial lightning and contain irregular surfaces.

TABLE IV: Point cloud evaluation between COLMAP and Pipeline.

| Dataset | pipeline-to-colmap | | colmap-to-pipeline | |
|---|---|---|---|---|
| | Precision (%) | Accuracy (m) | Recall (%) | Completeness (m) |
| Ginnie Ballroom | 96.9 | 0.029 | 97.5 | 0.019 |
| Cenote | 94.4 | 0.037 | 92.4 | 0.047 |
| Coral Reef | 52.3 | 0.109 | 60.3 | 0.114 |
| Stavronikita | 43.6 | 0.134 | 40.2 | 0.143 |




(a) Ginnie Ballroom (b) Coral reef

Fig. 10: Precision and Recall plots for the Ginnie Ballroom and Coral Reef datasets.

Thus, the images in the latter two datasets suffer from motion blur and color saturation. This leads to noisy point clouds by both systems and larger discrepancies between them. COLMAP is somewhat more robust, but its models are far from perfect on these data.

**Real-Time Reconstruction using SVIn2 Poses** In the last experiment, we use poses obtained by SVIn2 as input to the pipeline to simulate actual deployment of our approach. (A comparison of pose estimation results between COLMAP and SVIn2 is presented in Section IV-C.) To account for discrepancy in dense reconstruction resulting from camera pose tracking error, we use the rmse error between SVIn2 and COLMAP poses as a threshold to compute precision and recall as shown in Table V. The results show that the dense reconstructions obtained using SVIn2 poses are accurate compared to those of COLMAP, with both precision and recall over 80% for all datasets. The pipeline results show that even with drift in SVIn2 poses we are able to produce comparable reconstruction with that of COLMAP. This paves the way for real-time reconstruction onboard an Aqua2 AUV.

TABLE V: Point cloud using SVIn2's poses evaluation between COLMAP and Pipeline.

| Dataset | threshold (m) | pipeline-to-colmap Precision (%) | colmap-to-pipeline Recall (%) |
|---|---|---|---|
| Ginnie Ballroom | 0.07 | 85.6 | 92.0 |
| Cenote | 0.19 | 91.2 | 89.0 |
| Coral Reef | 0.39 | 81.8 | 85.9 |

## V. CONCLUSIONS

We have shown on a variety of challenging datasets that an online 3D reconstruction system with robust VIO [47] can obtain results on par with a much slower offline system. Such an evaluation was missing from the literature and helps answering the question on whether real-time dense reconstruction is feasible onboard. Dense 3D representations of the environment estimated in real time will enable improved navigation [48] and autonomous operations for the Aqua2 AUV [49]. Furthermore, gaps and boundaries of the dense reconstruction will effectively guide the AUV towards frontier points [50].

## References


[1] S. Williams and I. Mahon, "Simultaneous localisation and mapping on the Great Barrier Reef," in *ICRA*, 2004, 1771–1776 Vol.2.

[2] D. N. Kernagis, C. McKinlay, and T. R. Kincaid, "Dive logistics of the turner to wakulla cave traverse," in *Diving for Science 2008. Proceedings of the American Academy of Underwater Sciences Symposium*, 2008.

[3] M. Gary, N. Fairfield, W. C. Stone, D. Wettergreen, G. Kantor, and J. M. Sharp Jr, "3D mapping and characterization of Sistema Zacatón from DEPTHX (DEep Phreatic THermal eXplorer)," in *Sinkholes and the Engineering and Environmental Impacts of Karst*, American Society of Civil Engineers, 2008, pp. 202–212.

[4] A. Quattrini Li, A. Coskun, S. M. Doherty, *et al.*, "Experimental comparison of open source vision based state estimation algorithms," in *International Symposium of Experimental Robotics (ISER)*, Tokyo, Japan, 2016.

[5] B. Joshi, S. Rahman, M. Kalaitzakis, *et al.*, "Experimental Comparison of Open Source Visual-Inertial-Based State Estimation Algorithms in the Underwater Domain," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Macau, 2019, pp. 7221–7227.

[6] S. Rahman, A. Quattrini Li, and I. Rekleitis, "SVIn2: A multi-sensor fusion-based underwater SLAM system," *The International Journal of Robotics Research*, 2022.

[7] J. L. Schönberger and J.-M. Frahm, "Structure-from-motion revisited," in *CVPR*, 2016, pp. 4104–4113.

[8] J. L. Schönberger, E. Zheng, J.-M. Frahm, and M. Pollefeys, "Pixelwise view selection for unstructured multi-view stereo," in *ECCV*, 2016, pp. 501–518.

[9] E. Vargas, R. Scona, J. S. Willners, T. Luczynski, Y. Cao, S. Wang, and Y. R. Petillot, "Robust underwater visual slam fusing acoustic sensing," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, 2021.

[10] C. Hu, S. Zhu, Y. Liang, and W. Song, "Tightly-coupled visual-inertial-pressure fusion using forward and backward imu preintegration," *IEEE Robotics and Automation Letters*, vol. 7, no. 3, 2022.

[11] C. Beall, B. J. Lawrence, V. Ila, and F. Dellaert, "3d reconstruction of underwater structures," in *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2010.

[12] B. Joshi, M. Xanthidis, S. Rahman, and I. Rekleitis, "High definition, inexpensive, underwater mapping," in *IEEE International Conference on Robotics and Automation (ICRA)*, Philadelphia, PA, USA, 2022, accepted.

[13] J. Queiroz-Neto, R. Carceroni, W. Barros, and M. Campos, "Underwater stereo," in *Proceedings. 17th Brazilian Symposium on Computer Graphics and Image Processing*, 2004, pp. 170–177.

[14] M. Pollefeys, D. Nistér, J. .-. Frahm, *et al.*, "Detailed real-time urban 3D reconstruction from video," *IJCV*, vol. 78, no. 2-3, pp. 143–167, 2008.

[15] J. Stühmer, S. Gumhold, and D. Cremers, "Real-time dense geometry from a handheld camera," in *DAGM*, 2010, pp. 11–20.

[16] R. Newcombe and A. Davison, "DTAM: Dense tracking and mapping in real-time," in *ICCV*, 2011.

[17] A. Wendel, M. Maurer, G. Graber, T. Pock, and H. Bischof, "Dense reconstruction on-the-fly," in *CVPR*, 2012, pp. 1450–1457.

[18] V. Pradeep, C. Rhemann, S. Izadi, C. Zach, M. Bleyer, and S. Bathiche, "Monofusion: Real-time 3d reconstruction of small scenes with a single web camera," in *International Symposium on Mixed and Augmented Reality (ISMAR)*, 2013, pp. 83–88.

[19] M. Pizzoli, C. Forster, and D. Scaramuzza, "Remode: Probabilistic, monocular dense reconstruction in real time," in *ICRA*, 2014, pp. 2609–2616.

[20] J. Zienkiewicz, A. Tsiotsios, A. Davison, and S. Leutenegger, "Monocular, real-time surface reconstruction using dynamic level of detail," in *International Conference on 3D Vision (3DV)*, 2016, pp. 37–46.

[21] T. Schöps, T. Sattler, C. Häne, and M. Pollefeys, "Large-scale outdoor 3D reconstruction on a mobile device," *CVIU*, vol. 157, pp. 151–166, 2017.

[22] L. Teixeira and M. Chli, "Real-time local 3D reconstruction for aerial inspection using superpixel expansion," in *ICRA*, 2017, pp. 4560–4567.

[23] P. Mordohai, K. Batsos, A. Makadia, and N. Snavely, "NBVC: A benchmark for depth estimation from narrow-baseline video clips," in *IROS*, 2020, pp. 10 076–10 083.

[24] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The KITTI dataset," *International Journal of Robotics Research (IJRR)*, vol. 32, no. 11, pp. 1231–1237, 2013.

[25] C. Liu, J. Gu, K. Kim, S. G. Narasimhan, and J. Kautz, "Neural RGB→D sensing: Depth and uncertainty from a video camera," in *CVPR*, 2019, pp. 10 986–10 995.

[26] J. Xie, C. Lei, Z. Li, L. E. Li, and Q. Chen, "Video depth estimation by fusing flow-to-depth proposals," in *IROS*, 2020, pp. 10 100–10 107.

[27] A. Duzceker, S. Galliani, C. Vogel, P. Speciale, M. Dusmanu, and M. Pollefeys, "DeepVideoMVS: Multi-view stereo on video with recurrent spatio-temporal fusion," in *CVPR*, 2021, pp. 15 324–15 333.

[28] X. Long, L. Liu, W. Li, C. Theobalt, and W. Wang, "Multi-view depth estimation using epipolar spatio-temporal networks," in *CVPR*, 2021, pp. 8258–8267.

[29] Z. Min and E. Dunn, "VOLDOR+SLAM: for the times when feature-based or direct methods are not good enough," in *ICRA*, 2021, pp. 13 813–13 819.

[30] J. Sun, Y. Xie, L. Chen, X. Zhou, and H. Bao, "NeuralRecon: Real-time coherent 3D reconstruction from monocular video," in *CVPR*, 2021, pp. 15 598–15 607.

[31] S. Rahman, A. Quattrini Li, and I. Rekleitis, "Contour based reconstruction of underwater structures using sonar, visual, inertial, and depth sensor," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Macau, 2019, pp. 8048–8053.

[32] R. I. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, Second. Cambridge University Press, ISBN: 0521540518, 2004.

[33] H. Hirschmüller and D. Scharstein, "Evaluation of cost functions for stereo matching," in *CVPR*, 2007.

[34] O. Veksler, "Fast variable window for stereo correspondence using integral images," in *CVPR*, 2003.

[35] H. Hirschmüller, "Stereo processing by semiglobal matching and mutual information," *PAMI*, vol. 30, no. 2, pp. 328–341, 2008.

[36] R. Spangenberg, T. Langner, S. Adfeldt, and R. Rojas, "Large scale semi-global matching on the cpu," in *IEEE Intelligent Vehicles Symposium Proceedings*, 2014, pp. 195–201.

[37] D. Scharstein and R. Szeliski, "A taxonomy and evaluation of dense two-frame stereo correspondence algorithms," *IJCV*, vol. 47, no. 1-3, pp. 7–42, 2002.

[38] X. Hu and P. Mordohai, "A quantitative evaluation of confidence measures for stereo vision," *PAMI*, vol. 34, no. 11, pp. 2121–2133, 2012.

[39] T. Schöps, T. Sattler, C. Häne, and M. Pollefeys, "Large-scale outdoor 3D reconstruction on a mobile device," *CVIU*, vol. 157, pp. 151–166, 2017.

[40] P. Merrell, A. Akbarzadeh, L. Wang, P. Mordohai, J.-M. Frahm, R. Yang, D. Nistér, and M. Pollefeys, "Real-time visibility-based fusion of depth maps," in *ICCV*, 2007.

[41] X. Hu and P. Mordohai, "Least commitment, viewpoint-based, multi-view stereo," in *3DIMPVT*, 2012, pp. 531–538.

[42] S. Rahman, N. Karapetyan, A. Quattrini Li, and I. Rekleitis, "A modular sensor suite for underwater reconstruction," in *MTS/IEEE OCEANS - Charleston*, IEEE, 2018, pp. 1–6.

[43] G. Dudek, M. Jenkin, C. Prahacs, *et al.*, "A visually guided swimming robot," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Edmonton AB, Canada, 2005, pp. 1749–1754.

[44] S. Rahman, A. Quattrini Li, and I. Rekleitis, "An Underwater SLAM System using Sonar, Visual, Inertial, and Depth Sensor," in *IROS*, Macau, (IROS ICROS Best Application Paper Award. Finalist), 2019, pp. 1861–1868.

[45] S. Umeyama, "Least-squares estimation of transformation parameters between two point patterns," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 13, no. 4, 1991.

[46] Z. Zhang and D. Scaramuzza, "A tutorial on quantitative trajectory evaluation for visual(-inertial) odometry," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018.

[47] S. Rahman, A. Quattrini Li, and I. Rekleitis, "SVIn2: A Multi-sensor Fusion-based Underwater SLAM System," *International Journal of Robotics Research*, Jul. 2022.

[48] M. Xanthidis, N. Karapetyan, H. Damron, S. Rahman, J. Johnson, A. O'Connell, J. O'Kane, and I. Rekleitis, "Navigation in the presence of obstacles for an agile autonomous underwater vehicle," in *IEEE International Conference on Robotics and Automation*, Paris, France, 2020, pp. 892–899.

[49] J. Sattar, G. Dudek, O. Chiu, *et al.*, "Enabling autonomous capabilities in underwater robotics," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Nice, France, 2008, pp. 3628–3634.

[50] M. Xanthidis, M. Kalaitzakis, N. Karapetyan, J. Johnson, N. Vitzilaios, J. O'Kane, and I. Rekleitis, "Aquavis: A perception-aware autonomous navigation framework for underwater vehicles," in *IROS*, Prague, Czech Republic, 2021, pp. 5387–5394.