Vision-based 3D Reconstruction for Deep-Sea Environments: Practical Use for Surveys and Inspection

Maxime Ferrera, Aurélien Arnaubec, Clémentin Boittiaux, Inès Larroche, Jan Opderbecke

Vision-based 3D Reconstruction for Deep-Sea Environments: Practical Use for Surveys and Inspection

Maxime Ferrera
Ifremer
Underwater Robotics Lab
La Seyne-sur-mer, 83500, France
maxime.ferrera@ifremer.fr

Aurélien Arnaubec
Ifremer
Underwater Robotics Lab
La Seyne-sur-mer, 83500, France
aurelien.arnaubec@ifremer.fr

Clémentin Boittiaux
Ifremer
Underwater Robotics Lab
La Seyne-sur-mer, 83500, France
clementin.boittiaux@ifremer.fr

Inès Larroche
Ifremer
Underwater Robotics Lab
La Seyne-sur-mer, 83500, France
ines.larroche@eleves.enpc.fr

Jan Opderbecke
Ifremer
Underwater Robotics Lab
La Seyne-sur-mer, 83500, France
jan.opderbecke@ifremer.fr

***Abstract*—This paper presents a combination of real-time and offline 3D reconstruction methods for remotely operated vehicles (ROVs) equipped with cameras used in underwater inspection and survey tasks. The real-time component is based on a stereo visual simultaneous localization and mapping algorithm and a truncated signed distance field representation for producing a coarse online 3D reconstruction. The offline component uses structure-from-motion techniques to create a dense point cloud representation of the scene which is then meshed and textured to produce a high-quality textured 3D mesh. The paper highlights the feasibility of using ROVs for vision-based 3D reconstruction in real-world scenarios and the potential of combining real-time and offline processing in practice for a range of underwater applications.**

I. Introduction

Remotely operated vehicles (ROVs) equipped with cameras are increasingly being used for underwater inspection and survey tasks. In these scenarios, the ability to generate 3D reconstructions of the environment can be valuable for a variety of purposes, including navigation, mapping, inspection, and monitoring. However, generating high-quality 3D reconstructions in real-time can be a challenging task due to the limited computing power and bandwidth available on ROVs, as well as the complex and dynamic nature of underwater environments [1]. Yet, acquiring the correct data required to later produce high quality 3D reconstruction in an offline manner is also challenging if there is no live feedback for the ROV's pilots.

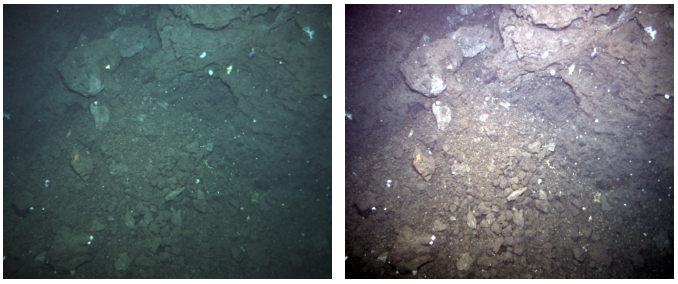
For instance, ensuring the complete coverage of a site of interest is highly time-consuming and error prone when performed without any feedback on what part of the site has been captured and what remains to be imaged.

To address these challenges, we present a combination of both real-time and offline 3D reconstruction methods. The real-time 3D reconstruction component is based on a stereo visual simultaneous localization and mapping (VSLAM) algorithm which is in charge of providing highly accurate navigation. Leveraging on the produced trajectory estimation, depth maps are integrated in a 3D model using a truncated signed distance field (TSDF) representation based on a voxel hash map [2]. This method allows us to create a coarse 3D reconstruction online that is already extremely useful for the ROV's pilots in order for them to more efficiently navigate around a structure of interest. Furthermore, the produced 3D models could allow the ROV to navigate in a more informed manner, allowing the development of autonomous navigation features.

Once the acquisition is completed, the recorded images and videos are processed with our offline 3D reconstruction software Matisse [3]. This software integrates structure-from-motion (SfM) techniques [4]–[6] to create a sparse point cloud and estimate the 3D pose of the camera for all the provided images. The output of the SfM are then processed with a multi-view stereo (MVS) algorithm [7] in order to create a dense point cloud representation of the scene which is finally meshed and textured in order to produce a high quality textured 3D mesh.

Current limitations are the quality of the depth maps estimation for the 3D reconstruction and the ability to re-localize on a previous reconstruction of the same site. Perspectives brought by deep learning approaches that could help solving these issues are also discussed.

The remaining of this paper is organized as follow. First, we present the offline vision-based 3D reconstruction solution. Next, we detail the real-time 3D reconstruction pipeline. Finally, we outline the current limitations of our methods and



(a) Raw image. (b) Stretched image.

Fig. 1: Comparison of acquired raw image and enhanced image through histogram stretching.

discuss about the perspectives brought by deep learning based algorithms to overcome them.

Overall, this paper demonstrates the feasibility of using ROVs for vision-based 3D reconstruction in real-world scenarios, and highlights the potential of combining real-time and offline processing for practical use in surveys and inspection tasks.

II. Offline underwater structure-from-motion for 3D reconstruction

In order to create faithful 3D reconstructions of underwater environments explored with ROVs, we present an offline vision-based structure-from-motion solution that can be enhanced when navigation data are also available.

In a first step, images acquired with either a still camera or a video camera are pre-processed. This pre-processing consists in extracting images at a regular rate from the recorded videos and applying an image processing enhancement in order to increase the underwater images' contrast [8]. This enhancement is performed through an histogram stretching for each channel independently that account for the typical underwater visual perturbations that are due to the back-scattering and color absorption effects. To do so, for each channel, the lowest and highest quantiles in terms of pixel intensity are first computed. Then, the pixel intensities are stretched using these quantiles as the minimum and maximum intensity values :

$$\mathbf{I}^c(\mathbf{x}) = \frac{\mathbf{I}^c(\mathbf{x}) - \mathbf{I}^c_{q_{min}}}{\mathbf{I}^c_{q_{max}} - \mathbf{I}^c_{q_{min}}} \cdot \mathbf{I}^c_{q_{max}}$$

where $\mathbf{I}^c(\mathbf{I})$ is the intensity of the image at the 2D pixel coordinate $\mathbf{I}$ for the color channel c and $\mathbf{I}^c_{q_{min}}$ and $\mathbf{I}^c_{q_{max}}$ are respectively the lowest and highest quantiles intensity values. The result of this histogram stretching technique is illustrated in Fig. 1.

In a second step, the collection of pre-processed images is fed to a classical features extraction and matching pipeline. As in most of the state-of-the-art SfM techniques, SIFT features [9] are extracted in every images and used for matching. When using pre-calibrated cameras, the features matching step is applied in a guided matching fashion to improve the quality of the found matches. This guided matching is performed thanks to the known intrinsics and distortion parameters of the cameras and to the relative motion computed between paired images from the initial matches found through a brute-force matching. In a nutshell, an initial brute-force matching is applied to find the set of images to pair and to compute either a two-view fundamental matrix or a homography transformation for each pair [10], finding the most likely one based on the number of resulting inliers after a 2D-2D filtering RANSAC scheme [11]. The two-view transformation is then used to match again the features of paired images through an epipolar guidance or a homography projection to increase the number of correct matches. Furthermore, when navigation data is available, the coarse navigation is used to limit the number of candidate images for the features matching, testing only images that are close enough to the currently processed image. This allows to significantly reduce the computation time of this time-consuming step while also limiting the number of possible outliers and erroneously established images pairs.

Once the features matching step performed, we obtain a graph of connected images that is used to bootstrap the structure-from-motion. We use an incremental SfM where images are incrementally added to the reconstructed model [4]. They are first registered through a P3P-RANSAC [12] using the observed features already triangulated. Then, their estimated pose is refined with a local Bundle Adjustment [13].

Defining camera's pose as $\mathbf{T_{wc_i}} \in \mathbb{SE}(3)$, where $\mathbb{SE}(3)$ denotes the 3D Special Euclidean group [14], the set of 3D map points $\boldsymbol{\lambda}^w_j \in \mathbb{R}^3$ and a set of 3D map points 2D observations per image $\mathbf{x}_{ij} \in \mathbb{R}^2$ and considering a calibrated camera, these state parameters are related by the projection function $\boldsymbol{\pi} : \mathbb{R}^3 \mapsto \mathbb{R}^2$ such that we have:

$$\mathbf{x}_{ij} = \boldsymbol{\pi}(\mathbf{T_{c_iw}} \odot \boldsymbol{\lambda}^w_j)$$
$$\mathbf{x}_{ij} = \boldsymbol{\pi}(\mathbf{R_{c_iw}} \cdot \boldsymbol{\lambda}^w_j + \mathbf{t_{c_iw}})$$

where $\mathbf{T_{c_iw}} = \mathbf{T^{-1}_{wc_i}}$ and $\mathbf{R_{c_iw}} \in \mathbb{SO}(3)$ and $\mathbf{t_{c_iw}} \in \mathbb{R}^3$ are respectively the rotation matrix and the translation part of image i inverse pose $\mathbf{T_{c_iw}}$.

The local Bundle Adjustment is then solved as a nonlinear least-squares problem:

$$\underset{\mathbf{T_{c_iw}}, \boldsymbol{\lambda}^w_j}{\arg\min} \sum_i \sum_j \left\| \mathbf{x}_{ij} - \boldsymbol{\pi}(\mathbf{T_{c_iw}} \odot \boldsymbol{\lambda}^w_j) \right\|^{\boldsymbol{\gamma}}_{\boldsymbol{\Sigma}}$$

where the most highly connected images with the current image are set free, as well as their set of observed 3D points. The other connected images are kept fixed during the local Bundle Adjustment in order to fix the gauge of the model. The Bundle Adjustment problems are solved with the Levenberg-Marquardt algorithm [15], using a robust Huber cost function model $\boldsymbol{\gamma}(\cdot)$ [10] during the BA optimization to limit the impact of potential outliers and remove the detected ones. Additionally, the residuals are weighted by their covariance $\boldsymbol{\Sigma}$ through the minimization of the Mahalanobis distance [13].

Once the pose refined, new 3D points are estimated through triangulation [16] between the 2D features matched between registered images and not yet triangulated. In order to ensure consistent estimations of images' poses and 3D points, a global Bundle Adjustment is also regularly applied, where the intrinsics calibration parameters $\mathbf{K}$ and distortion coefficients $\mathbf{D}$ of the camera are also optimized :

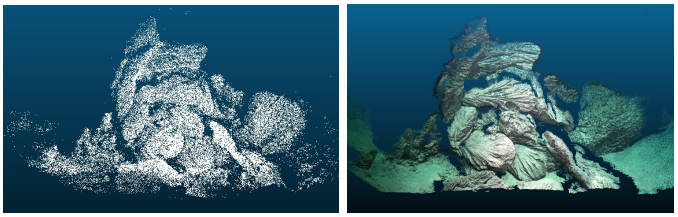
$$\underset{\mathbf{T_{c_iw}},\boldsymbol{\lambda}_j^w,\mathbf{K},\mathbf{D}}{\arg\min} \sum_i \sum_j \left\| \mathbf{x}_{ij} - \boldsymbol{\pi}(\mathbf{T_{c_iw}} \odot \boldsymbol{\lambda}_j^w) \right\|_{\boldsymbol{\Sigma}}^{\gamma}$$

In cases where navigation data is available, the navigation priors are used as position measurements $\mathbf{p}_i$, whose covariance is denoted as $\boldsymbol{\Sigma}'$, to constrain the global Bundle Adjustment optimization problem :

$$\underset{\mathbf{T_{c_iw}},\boldsymbol{\lambda}_j^w,\mathbf{K},\mathbf{D}}{\arg\min} \sum_i \sum_j \left\| \mathbf{x}_{ij} - \boldsymbol{\pi}(\mathbf{T_{c_iw}} \odot \boldsymbol{\lambda}_j^w) \right\|_{\boldsymbol{\Sigma}}^{\gamma} + \sum_i \left\| \mathbf{p}_i - \mathbf{t_{wc_i}} \right\|_{\boldsymbol{\Sigma}'}^{\gamma}$$

The advantage of tightly including the navigation priors as measurements are two-fold: they allow to obtain scaled and geo-referenced SfM reconstruction model, allowing us to get rid of the well-known scale ambiguity issue of monocular SfM techniques, and they help in detecting visual outliers, early limiting their impact in the reconstruction process.

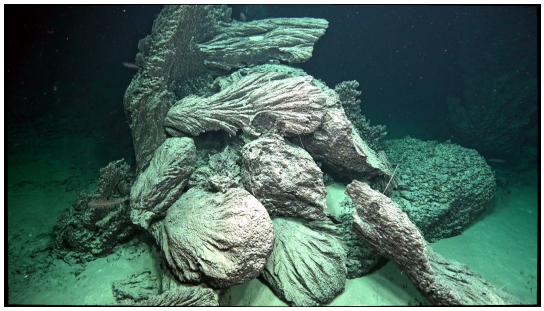
Once the structure-from-motion done, we obtain a sparse point cloud that represent the 3D structure of the imaged environment (see Fig. 2a) as well as highly accurate poses estimation of the camera for each image.



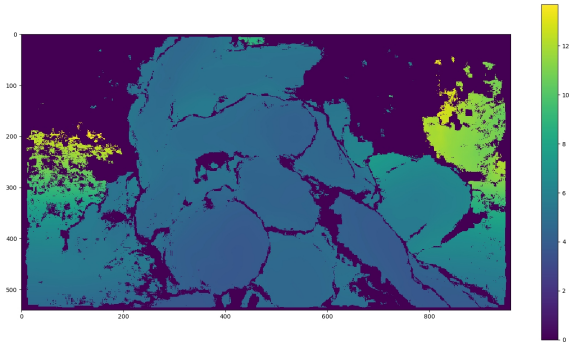
(a) Sparse point cloud. (b) Dense point cloud.

Fig. 2: (left) Structure-from-motion sparse point cloud. (right) Dense point cloud from depth maps estimation.

This sparse model is then used to perform a Multi-View Stereo processing. This step consists in computing depth maps for each image in order to densify the 3D reconstruction. To do so, a patchmatch algorithm [17] is applied to estimate, for each image, a dense matching with its covisible images that will finally allow us to estimate a 3D point per matched pixel thanks to the known images' poses and camera's calibration parameters (see Fig. 3). Highly dense point clouds are then extracted for each image from their estimated depth maps and thus are fused with some visibility checks to remove as best as possible the remaining outliers [18] (see Fig. 2b).



(a) Raw image used in the structure-from-motion.



(b) Resulting depth map.

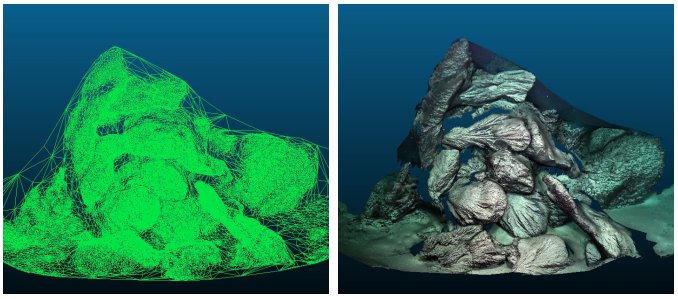
Fig. 3: Depth map computed for an image processed by the structure-from-motion through multi-view stereo.

Finally, a Delaunay meshing algorithm is applied to compute a 3D mesh (see Fig. 4a) over the dense point cloud [19]. This mesh is then textured (see Fig. 4b) through the projection of the 3D mesh faces into their viewing images and optimizing the choice of the texture to obtain a seamless texturing of the mesh [20].

While highly accurate 3D models can be obtained through this offline pipeline (see Fig. 5 and 6), the inherent difficulty lies in the correct acquisition of the initial images. Indeed, ROV's are manually piloted and the acquisition of the images when inspecting the scene of interests are hence performed in a dead-reckoning way. This issue leads to very careful survey of these scenes, navigating in an exhaustive manner around the scene to scan, that are extremely time consuming and always exhibit a risk of missing some parts of the scene that will only be detected post-mission through the apparition of holes in the reconstruction when processed through this offline reconstruction pipeline. To overcome this issue, we next present a method for performing real-time 3D reconstruction that provides a live feed-back to the ROV's pilots, thus allowing them to optimize the survey of the scene to scan and to ensure its full coverage.

III. Real-time underwater visual 3D reconstruction

The generation of a 3D model in real-time is extremely valuable in underwater environments. Indeed, while this topic



(a) 3D reconstructed mesh. (b) 3D textured mesh.

Fig. 4: Depth map computed for an image processed by the structure-from-motion through multi-view stereo.

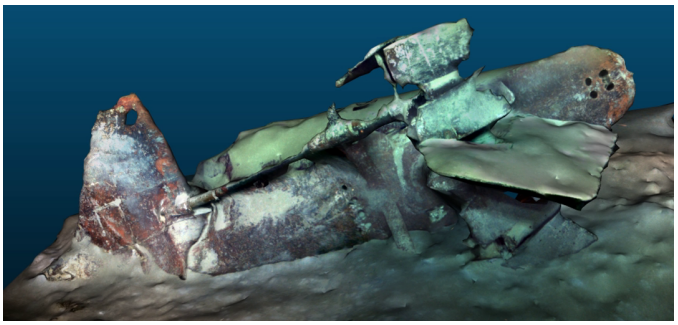


Fig. 5: Offline 3D reconstruction of a shipwreck.

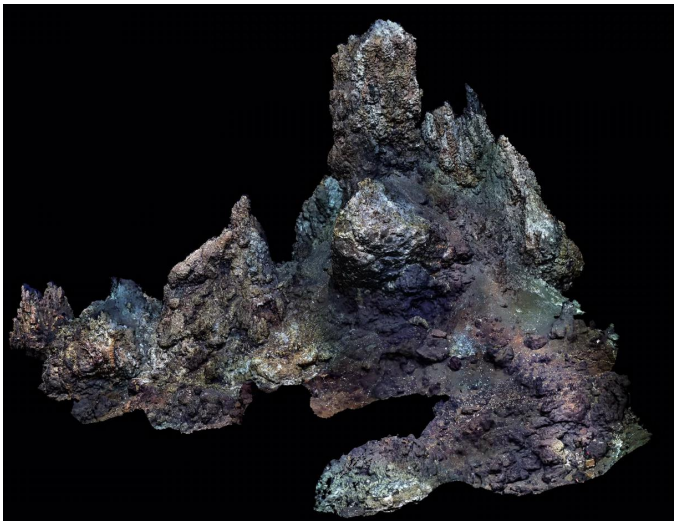


Fig. 6: Offline 3D reconstruction of a hydro-thermal vent.

is also of high interest in ground and aerial robotics for autonomous navigation mostly, it is additionally of direct use for the ROV's pilot as an assistance feature for a better and safer navigation.

In opposition to offline 3D reconstruction methods such as the one presented in the previous section, the real-time case is highly more challenging. First, only past and present data can be used for the current estimations. Second, the involved algorithms must cope with the real-time constraint, that is the processing of the data must be fast enough to perform the estimations at approximately the data flow rate. Last, the computation a 3D model in real-time requires that both the 3D position and orientation of the ROV are very accurately known also in real-time as this knowledge is required to correctly and accurately position the 3D measurements to be used to reconstruct the model.

Here, we present a method based on the use of stereo camera embedded on an ROV. The stereo camera is calibrated in order to use it as a 3D sensor with metric measurements. A naive and easy solution to build a 3D model in real-time with such sensor would be to use the navigation data that are produced by the Inertial Navigation System (INS) that is usually embedded on ROVs to estimate its position and orientation in real-time and use this information to project 3D point clouds computed through stereo matching in a 3D space. However, the accuracy of INS for navigation is at best in the order of tens of centimeters and more often around one meter of uncertainty. Thus, using these raw navigation data would lead to very inaccurate 3D models which, in addition to being geometrically inaccurate, would also appear extremely blurred as even errors of a few centimeters in the pose of the 3D sensor create bad alignments between the different 3D points clouds to integrate. Moreover, the point clouds that are computed from stereo matching are always noisy and corrupted by outliers. This means that, even if an ideal system could provide a perfect navigation, simply projecting the computed point clouds in the 3D space would lead to inaccurate and blurry 3D models. To overcome all these issues, we follow the monocular real-time 3D reconstruction presented in [21]. A visual SLAM algorithm is used to estimate in real-time the trajectory followed by the stereo camera embedded on the ROV. This visual SLAM allows the estimation of centimetric accurate 3D pose of the camera and is thus way more suited to the 3D reconstruction task. Additionally, a truncated signed distance field (TSDF) 3D model [22] is employed in order to integrate the 3D measurements computed through stereo matching in a probabilistic way.

The stereo visual SLAM that we use is based on OV²SLAM [23]. This SLAM algorithm follows the modern multi-threading approach [24], [25] where a front-end thread is dedicated to the processing of all the received images in real-time, providing a pose estimation of the camera at its acquisition frame-rate, and a back-end thread ensure a continuous optimization of a subsample of the most recent camera's poses, *i.e.* the keyframes, and the triangulated 3D points observed by these keyframes through a local Bundle Adjustment.

The front-end thread tracks features in the video stream through a sparse optical flow method based on the KLT algorithm [26] and estimates the pose of the camera with a PnP method [10] through the tracked observations of previously triangulated 3D points. When the front-end thread detects a high motion or a significant drop in the number of tracked

features, it triggers the creation of a new keyframe and detect new features into it. On the other side, the back-end thread is in charge of processing the created keyframe. It ensures the triangulation of new 3D points and perform a local BA over the newly created keyframes, along with their neighbors and observed 3D points:

$$\underset{\mathbf{T_{c_iw}},\boldsymbol{\lambda}_j^w}{\arg\min} \sum_i \sum_j \left\| \mathbf{x}_{ij} - \boldsymbol{\pi}(\mathbf{T_{c_iw}} \odot \boldsymbol{\lambda}_j^w) \right\|_{\boldsymbol{\Sigma}}^{\gamma}$$

As only a subsample of all the received images are pinned as keyframes, the back-end thread has a higher amount of time available for performing these local Bundle Adjustment optimizations. This back-end thread thus provides very accurate pose estimations for the keyframes.

For the 3D reconstruction, we set an additional thread that receives the keyframes processed by the back-end thread once they have been optimized. This thread is in charge of applying a stereo matching to produce dense depth maps. This stereo matching is done with the SGM algorithm [27] on the keyframe' rectified stereo images. The computed depth map is then converted into a 3D point cloud that is projected into the 3D space using the known pose of the keyframe. The projected point cloud is finally integrated in a 3D TSDF model.

We use the Voxblox implementation of TSDF [28] to integrate the produced point clouds. Voxblox integrates the 3D rays that define the point cloud into voxels and updates a distance to the surface value for each voxel in probabilistic way, keeping an estimate of this value and a confidence weight for each voxel that gets updated every time a ray get through it or close enough. In order to ensure a real-time capacity and to stay tractable in terms of memory footprint, chunks of voxels are allocated dynamically, any time a new part of the 3D space gets covered [29]. This way, there is no requirement for any prior knowledge on the size of the scene to scan. Finally, a 3D mesh can be extracted from all the allocated voxels at a user-defined rate. The mesh extraction is performed using a marching cube algorithm [30] in charge of searching the iso-contours of the 3D model through the distance to the surface value stored in the TSDF voxels.

The results of this method can be appreciated in Fig. 8a. As one can see in Fig. 7, parts of the scene that have not yet been acquired are directly noticeable, allowing the pilots of the ROV to navigate more efficiently for completing the coverage of the site. Furthermore, when comparing the final 3D model produced in real-time to the reconstruction obtained with the offline method presented in section II, we can see that the result is quite faithful in the reconstruction of the 3D structures (see Fig. 8b).

In our case, a stereo camera is used for this real-time 3D pipeline. While the stereo visual SLAM employed here shows a great accuracy [31], underwater vision can be easily perturbed by high level of turbidity or extensive light absorption

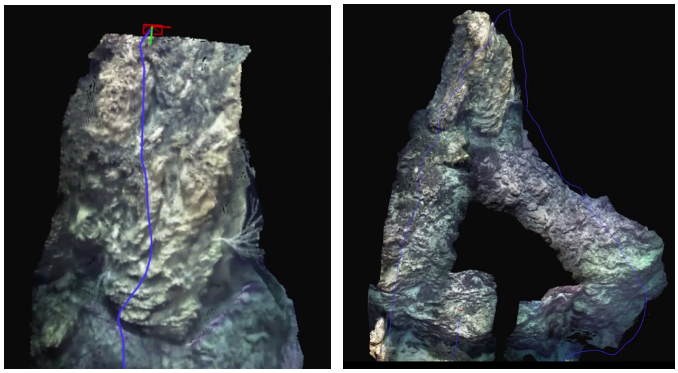


Fig. 7: Real-time 3D reconstruction live views.

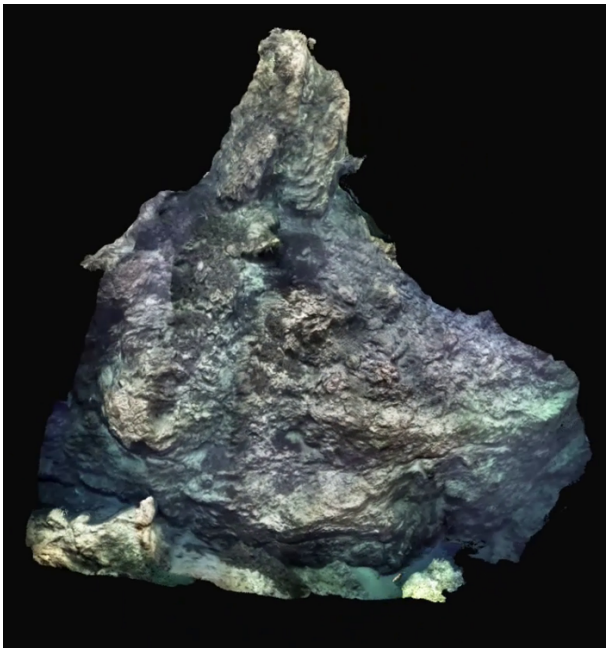
(when the ROV is navigating too far away from the structure of interest for instance). To overcome this limitation, it could be interesting to fuse the measurements of additional navigation sensors [32] and add a sonar sensor [33] in order to be able to continuously estimate the trajectory of the ROV through SLAM, even when there are loss of visual information.

Another important feature, both for offline and real-time 3D reconstruction, lies in the ability to automatically relocate from images only, either for loop-closure or to register images acquired with a different camera on an existing 3D model. For real-time processing, the quality of the estimated depth maps is also something that could be improved. These topics are discussed in the next section.

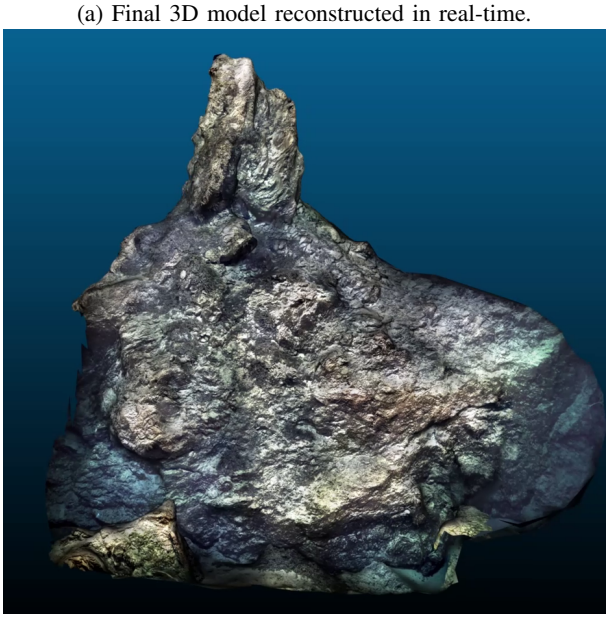
IV. Limitations and Perspectives

The ability to relocalize on previously reconstructed 3D models is of high interest for many applications such as long-term visual localization [34] and temporal monitoring of underwater sites of interest. Yet, when matching images acquired by a different camera setup, possibly with a different ROV and different lightning condition, the classical SIFT features tend to fail. However, recent neural network for robust features detection and matching like SuperPoint [35] and SuperGlue [36] have proven to generalize well to underwater images [37]. Furthermore, the use of deep pose regression methods [38]–[40] have also shown interesting results on this task [41].

Another topic of interest for deep learning approaches is the estimation of more accurate depth maps. Indeed, while stereo matching or multi-view stereo methods are usable in the underwater context, the resulting depth maps are often noisy and require strong filtering for optimal results in real-time 3D reconstructions [42]. Estimation and refinement of stereo matching based depth maps using deep convolutional neural networks [43]–[45] could prove to be more accurate than classical model-based stereo matching methods [27], [46]. Additionally, because of the inherent difficulty in acquiring ground-truth underwater, the use of self-supervised methods are quite appealing [47]–[50] and seems applicable to the underwater context [51].



(a) Final 3D model reconstructed in real-time.



(b) Offline 3D reconstruction of the same scene as above.

Fig. 8: Final 3D models produced by: (top) the real-time reconstruction method, (bottom) the offline method.

V. Conclusion

In conclusion, the proposed method demonstrates the feasibility of using ROVs equipped with cameras for vision-based 3D reconstruction in real-world underwater scenarios. By combining real-time and offline processing, the method offers an efficient and accurate solution for acquiring and generating high-quality 3D reconstructions that can be used for navigation, mapping, inspection, and monitoring purposes. Overall, the presented methods highlight the practical usage of vision-based 3D reconstructions with ROVs and discuss their current limitations as well as some perspectives brought by deep learning approaches for further improvement.

References

[1] Ferrera, Maxime and Moras, Julien and Trouvé-Peloux, Pauline and Creuze, Vincent, "Real-time monocular visual odometry for turbid and dynamic underwater environments," *Sensors*, vol. 19, no. 3, p. 687, 2019.

[2] M. Nießner, M. Zollhöfer, S. Izadi, and M. Stamminger, "Real-time 3d reconstruction at scale using voxel hashing," *ACM Transactions on Graphics (ToG)*, vol. 32, no. 6, pp. 1–11, 2013.

[3] A. Arnaubec, M. Ferrera, J. Escartin, M. Matabos, N. Gracias, and J. Opderbecke, "Underwater 3d reconstruction from video or still imagery: Matisse & 3dmetrics processing and exploitation software," *Journal of Marine Science and Engineering*, 2023.

[4] Moulon, Pierre and Monasse, Pascal and Perrot, Romuald and Marlet, Renaud, "Openmvg: Open multiple view geometry," in *International Workshop on Reproducible Research in Pattern Recognition*, pp. 60–74, Springer, 2016.

[5] Schönberger, Johannes Lutz and Frahm, Jan-Michael, "Structure-from-Motion Revisited," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[6] Rupnik, Ewelina and Daakir, Mehdi and Deseilligny, Marc Pierrot, "MicMac–a free, open-source solution for photogrammetry," *Open Geospatial Data, Software and Standards*, vol. 2, no. 1, pp. 1–9, 2017.

[7] Y. Furukawa, C. Hernández, *et al.*, "Multi-view stereo: A tutorial," *Foundations and Trends® in Computer Graphics and Vision*, vol. 9, no. 1-2, pp. 1–148, 2015.

[8] Arnaubec, Aurelien and Opderbecke, Jan and Allais, Anne-Gaelle and Brignone, Lorenzo, "Optical mapping with the ARIANE HROV at IFREMER: The MATISSE processing tool," in *OCEANS 2015-Genova*, pp. 1–6, IEEE, 2015.

[9] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International journal of computer vision*, vol. 60, pp. 91–110, 2004.

[10] Hartley, Richard and Zisserman, Andrew, *Multiple view geometry in computer vision*. Cambridge university press, 2003.

[11] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[12] M. Persson and K. Nordberg, "Lambda twist: An accurate fast robust perspective three point (p3p) solver," in *Proceedings of the European conference on computer vision (ECCV)*, pp. 318–332, 2018.

[13] B. Triggs, P. F. McLauchlan, R. I. Hartley, and A. W. Fitzgibbon, "Bundle adjustment—a modern synthesis," in *Vision Algorithms: Theory and Practice: International Workshop on Vision Algorithms Corfu, Greece, September 21–22, 1999 Proceedings*, pp. 298–372, Springer, 2000.

[14] Barfoot, Timothy D, *State estimation for robotics*. Cambridge University Press, 2017.

[15] S. Wright, J. Nocedal, *et al.*, "Numerical optimization," *Springer Science*, vol. 35, no. 67-68, p. 7, 1999.

[16] S. H. Lee and J. Civera, "Triangulation: Why optimize?," in *Proceedings of the British Machine Vision Conference (BMVC)*, September 2019.

[17] M. Bleyer, C. Rhemann, and C. Rother, "Patchmatch stereo - stereo matching with slanted support windows," in *Proceedings of the British Machine Vision Conference (BMVC)*, vol. 11, pp. 1–11, January 2011.

[18] Cernea, Dan, "OpenMVS: Multi-View Stereo Reconstruction Library." https://cdcseacave.github.io/openMVS, 2020.

[19] M. Jancosek and T. Pajdla, "Exploiting visibility information in surface reconstruction to preserve weakly supported surfaces," *International scholarly research notices*, vol. 2014, 2014.

[20] Waechter, Michael and Moehrle, Nils and Goesele, Michael, "Let there be color! Large-scale texturing of 3D reconstructions," in *European conference on computer vision*, pp. 836–850, Springer, 2014.

[21] M. Ferrera, *Monocular Visual-Inertial-Pressure fusion for Underwater localization and 3D mapping.* PhD thesis, Université de Montpellier, 2019.

[22] 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*, pp. 303–312, 1996.

[23] Ferrera, Maxime and Eudes, Alexandre and Moras, Julien and Sanfourche, Martial and Le Besnerais, Guy, "OV^{2}SLAM: A Fully Online and Versatile Visual SLAM for Real-Time Applications," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 1399–1406, 2021.

[24] Klein, Georg and Murray, David, "Parallel tracking and mapping for small AR workspaces," in *6th IEEE and ACM International Symposium on Mixed and Augmented Reality*, pp. 225–234, IEEE, 2007.

[25] Mur-Artal, Raul and Montiel, Jose Maria Martinez and Tardos, Juan D, "ORB-SLAM: a versatile and accurate monocular SLAM system," *IEEE Transactions on Robotics*, vol. 31, no. 5, pp. 1147–1163, 2015.

[26] Baker, Simon and Matthews, Iain, "Lucas-kanade 20 years on: A unifying framework," *International Journal of Computer Vision*, vol. 56, no. 3, pp. 221–255, 2004.

[27] H. Hirschmüller, M. Buder, and I. Ernst, "Memory efficient semi-global matching," *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 1, pp. 371–376, 2012.

[28] H. Oleynikova, Z. Taylor, M. Fehr, R. Siegwart, and J. Nieto, "Voxblox: Incremental 3d euclidean signed distance fields for on-board mav planning," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1366–1373, IEEE, 2017.

[29] M. Klingensmith, I. Dryanovski, S. Srinivasa, and J. Xiao, "Chisel: Real time large scale 3d reconstruction onboard a mobile device using spatially hashed signed distance fields," in *Robotics: Science and Systems*, (Rome, Italy), July 2015.

[30] W. E. Lorensen and H. E. Cline, "Marching cubes: A high resolution 3d surface construction algorithm," *ACM siggraph computer graphics*, vol. 21, no. 4, pp. 163–169, 1987.

[31] M. Ferrera, A. Arnaubec, K. Istenič, N. Gracias, and T. Bajjouk, "Hyperspectral 3d mapping of underwater environments," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 3703–3712, 2021.

[32] M. Ferrera, V. Creuze, J. Moras, and P. Trouvé-Peloux, "Aqualoc: An underwater dataset for visual–inertial–pressure localization," *The International Journal of Robotics Research*, vol. 38, no. 14, pp. 1549–1559, 2019.

[33] S. Rahman, A. Q. Li, and I. Rekleitis, "Svin2: An underwater slam system using sonar, visual, inertial, and depth sensor," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1861–1868, IEEE, 2019.

[34] P.-E. Sarlin, C. Cadena, R. Siegwart, and M. Dymczyk, "From coarse to fine: Robust hierarchical localization at large scale," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12716–12725, 2019.

[35] D. DeTone, T. Malisiewicz, and A. Rabinovich, "SuperPoint: Self-supervised interest point detection and description," in *Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pp. 337–33712, 2018.

[36] P.-E. Sarlin, D. DeTone, T. Malisiewicz, and A. Rabinovich, "SuperGlue: Learning feature matching with graph neural networks," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4938–4947, 2020.

[37] C. Boittiaux, C. Dune, M. Ferrera, A. Arnaubec, R. Marxer, M. Matabos, L. Van Audenhaege, and V. Hugel, "Eiffel tower: A deep-sea underwater dataset for long-term visual localization," *The International Journal of Robotics Research*, 2023.

[38] A. Kendall, M. Grimes, and R. Cipolla, "Posenet: A convolutional network for real-time 6-dof camera relocalization," in *Proceedings of the IEEE international conference on computer vision*, pp. 2938–2946, 2015.

[39] P.-E. Sarlin, A. Unagar, M. Larsson, H. Germain, C. Toft, V. Larsson, M. Pollefeys, V. Lepetit, L. Hammarstrand, F. Kahl, *et al.*, "Back to the feature: Learning robust camera localization from pixels to pose," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 3247–3257, 2021.

[40] C. Boittiaux, R. Marxer, C. Dune, A. Arnaubec, and V. Hugel, "Homography-based loss function for camera pose regression," *IEEE Robotics and Automation Letters*, vol. 7, no. 3, pp. 6242–6249, 2022.

[41] C. Boittiaux, C. Dune, A. Arnaubec, R. Marxer, M. Ferrera, and V. Hugel, "Long-term visual localization in deep-sea underwater environment," in *ORASIS*, 2023.

[42] T. Schöps, T. Sattler, C. Häne, and M. Pollefeys, "Large-scale outdoor 3d reconstruction on a mobile device," *Computer Vision and Image Understanding*, vol. 157, pp. 151–166, 2017.

[43] A. Tonioni, F. Tosi, M. Poggi, S. Mattoccia, and L. D. Stefano, "Real-time self-adaptive deep stereo," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 195–204, 2019.

[44] J.-R. Chang and Y.-S. Chen, "Pyramid stereo matching network," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 5410–5418, 2018.

[45] M. Ferrera, A. Boulch, and J. Moras, "Fast stereo disparity maps refinement by fusion of data-based and model-based estimations," in *2019 International Conference on 3D Vision (3DV)*, pp. 9–17, IEEE, 2019.

[46] A. Geiger, M. Roser, and R. Urtasun, "Efficient large-scale stereo matching," in *Asian Conference on Computer Vision (ACCV)*, 2010.

[47] C. Godard, O. Mac Aodha, and G. J. Brostow, "Unsupervised monocular depth estimation with left-right consistency," in *CVPR*, 2017.

[48] C. Godard, O. Mac Aodha, M. Firman, and G. J. Brostow, "Digging into self-supervised monocular depth estimation," in *The International Conference on Computer Vision (ICCV)*, October 2019.

[49] R. Ranftl, K. Lasinger, D. Hafner, K. Schindler, and V. Koltun, "Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 3, 2022.

[50] M. Poggi, F. Aleotti, F. Tosi, and S. Mattoccia, "Towards real-time unsupervised monocular depth estimation on cpu," in *2018 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, pp. 5848–5854, IEEE, 2018.

[51] S. Amitai, I. Klein, and T. Treibitz, "Self-supervised monocular depth underwater," *arXiv preprint arXiv:2210.03206*, 2022.