Mission-time 3D Reconstruction with Quality Estimation

Klemen Istenič*, Viorela Ila†, Lukáš Polok‡, Nuno Gracias*, and Rafael García*
* University of Girona, † Australian National University, ‡ Brno University of Technology
{klemen.istenic,rafael.garcia}@udg.edu, viorela.ila@anu.edu.au, ngracias@silver.udg.edu, ipolok@fit.vutbr.cz

***Abstract*—Accurate and detailed 3-dimensional (3D) models of the underwater environment are becoming increasingly important in modern marine surveys, since they convey immense information that can be easily interpreted. Techniques such as bundle adjustment (BA) and structure from motion (SfM), which jointly estimate sparse 3D points of the scene and camera poses, have gained popularity in underwater mapping applications. However, for large-area surveys these methods are computationally expensive and not intended for online application. This paper proposes an SfM pipeline based on solving the BA problem in an incremental and efficient way. Furthermore, the new system can provide not only the solution of the optimization (camera trajectory along time and the 3D points of the environment), but also the estimate of the uncertainty associated with the 3D reconstruction. This system is able to produce results in mission-time, *i.e.* while the robot is in the water or very shortly afterwards. Such quick availability is of great importance during survey operations as it allows data quality assessment *in-situ*, and eventual replanning of missions in case of need.**

I. Introduction

Underwater exploration and inspection is a fundamental way to improve our knowledge of the oceans. Accurate and detailed 3D models of the environment obtained from the data acquired underwater yield high added value to any marine survey, as such results convey immense information easily interpretable by humans. The wealth of information enables experts (biologists, archaeologists and geologists, among others) to perform further in-depth investigation of the areas of interest after the missions, and can also serve as base map for long term environmental monitoring.

Recent advances in technology enabled scientists to capitalize on the use of unmanned underwater vehicles (UUVs) to gain access to large marine areas and deep sea regions. While underwater 3D mapping usually relies on multibeam echosounders and sidescan sonars, the relatively coarse resolution of acoustic sensing prevents highly detailed representation of complex structures with concavities. Optical sensing, on the other hand, can be used to recover quality 3D representation of smaller areas of interest in higher resolution.

Image based 3D reconstruction techniques have been studied extensively in the computer vision community. Sparse techniques such as BA and SfM, which jointly estimate sparse 3D points of the scene and camera poses, have gained popularity in underwater surveying and are currently used for producing 3D representations from data provided by commercial and custom built camera systems (e.g., [1]–[4]). To obtain an optimal solution, a nonlinear optimization is performed on a complete set of camera poses and 3D points observed by the camera [5] in a stochastic estimation framework which accounts for Gaussian noise models in the observations. For large scale applications, this is an expensive procedure and it is normally performed offline, after the acquisition process. In this sense, the reconstruction phase is decoupled from the acquisition, and performed offline post-mission.

Due to the unfavourable properties of the underwater medium (such as the rapid attenuation of light, scattering effects, and non-uniform lighting), the outcome of the underwater 3D reconstruction is vastly dependent on the conditions and the strategy applied in the acquisition process. The current offline nature of the processing prevents any feedback about the quality of the reconstruction during the mission. This consequently demands a strong human intervention during the surveys and careful mission planning to ensure the capture of adequate data. Despite the best efforts, several deployments over the same area are still commonly required, significantly increasing the total expenditure of the mission due to high costs of ship-time and highly trained personnel.

Currently, some techniques capable of online reconstruction decouple the problem into a local BA step optimizing over parameters of a few recently added cameras and 3D points and a global camera pose graph optimization [6], [7]. Alternatively, the complexity of the problem can be reduced by marginalizing out the 3D structure from the optimization process: a technique called light bundle adjustment [8]. While the former achieves real-time results in structured environments, the reconstruction is only locally consistent during the execution of large loops, whereas the latter obtains the solution solely for camera poses. Neither case is able to provide an estimate of the uncertainty of the solution encoded in the covariance matrix associated to each of the variables (camera poses and points). This covariance is a valuable indicator of the quality of the reconstruction, which can be highly beneficial for the acquisition process if it can be estimated during the mission.

This paper analyzes several solutions for solving the

BA problem in an efficient way, during the mission, in an incremental fashion. Furthermore, it proposes a new system that is able to provide not only the solution of the optimization (pose of the cameras and location of the 3D points in the environment), but also the value of the associated uncertainty of the 3D reconstruction at mission-time. The ability to obtain the reconstruction and its associated uncertainty during the time of the mission enables the possibility of concurrent assessment of the quality of the acquired data (and the 3D structure) as well as the identification of poorly mapped or even missing areas. Endowed with this additional information, skilled pilots and autonomous planning schemes will be able to alter the mission in progress. By guiding the vehicle towards the problematic areas, the quality of the final representation will be significantly improved. As such, these quality-aware surveys not only improve the survey efficiency but also help reduce the need for additional deployments of the vehicle, further reducing the mission time and cost.

II. Related Works

When performing online 3D reconstruction, the state, containing the 3D structure and the camera trajectory so far, is continuously growing, leading to a highly computationally demanding estimation process. There are several solutions to speed up the online processing. One is to reduce the problem to a pose graph optimization where only the poses of key frames are globally optimized and local BA is used to adjust the cameras and the points [6]. Incremental light bundle adjustment [8] is another technique proposed for solving BA incrementally, which it is based on marginalizing out the structure while solving only for the camera poses. Those techniques are not suitable for applications where a feedback about the structure is needed during the acquisition, where a globally consistent solution is required every step.

Efficient incremental NLS methods have been developed in the simultaneous localization and mapping (SLAM) community [9], [10]. Those methods exploit the fact that adding new information into the system only affects a part of the solution. SLAM structure facilitates identifying the affected part. The matters in bundle adjustment are more complicated, since the increments have much higher rank than in SLAM and sometimes affecting large part of the system (e.g. when the same points are seen by most of the cameras).

In general, the existing solutions to NLS provide only the estimate of the mean state vector, its associated covariance being computationally too expensive to recover. Nevertheless, in SLAM applications, knowing only the mean vector is not enough. Quality estimation, active decisions and next best view are only a few of the applications that require fast state covariance recovery. Several approximations for marginal covariance recovery have been proposed in the literature. Thrun et al. [11] suggested using conditional covariances, which are inversions of sub-blocks of the system matrix, called the Markov blankets. The result is an overconfident approximation of the marginal covariances. Online, conservative approximations were proposed in [12], where at every step, the covariances corresponding to the new variables are computed by solving the augmented system with a set of basis vectors. An exact method for sparse covariance recovery was proposed in [13], based on a recursive formula which calculates any covariance elements on demand from other covariance elements and elements of the Cholesky factorization result. An incremental technique to obtain exact marginal covariances has recently been proposed by Ila et al. [14], and it is based on incremental updates of marginal covariances every time new variables and observations are integrated into the system, and on the fact that, in practice, the changes in the linearization point are often small and can be ignored. However, the BA and SfM problems have a slightly different structure where the number of points is in general much larger than the number of cameras and there are more efficient methods to solve the linearized system. Polok et al. [15] proposed an efficient method to calculate the point covariances in the context of BA. An improved version of this method is integrated in our pipeline.

III. Pipeline overview

The following section presents an overview of the proposed approach for a robust, globally-consistent, large-scale 3D reconstruction. Conceptually we can understand the pipeline as divided into two parts; the *front-end* and the *back-end*. The front-end is in charge of tracking 3D points and obtaining, at every time step, an initial estimate for the camera pose, associations with the existing 3D points, and creating newly observed 3D points. For that, features in every new frame are matched with features in the previous frames and based on that an initial estimate of the current pose of the camera and new 3D points is obtained. The new camera and points are refined by the back-end that implements incremental bundle adjustment system to obtain a globally consistent estimate at every step. In order to account for the high level of noise in underwater image processing, the BA is formulated as a probabilistic framework and provides not only the mean estimate but also the uncertainty of each camera pose and 3D points.

An important characteristic of the proposed pipeline is its ability to eliminate outliers which is implemented at several stages, when initializing the camera pose as well as in the global optimization stage. We found this is mandatory when processing noisy underwater images.

The state of our 3D reconstruction is given by the camera poses, $\mathbf{c} = [c_1 \dots c_{nc}]$ and the 3D points in the environment $\mathbf{p} = [p_1 \dots p_{np}]$. The camera poses can be parameterized using 6D vectors. It is common to consider a camera pose as an element of the Lie algebra $\widehat{c}_i \in \mathfrak{se}(3)$ of the special Euclidean group $SE(3)$ with $\widehat{c}_i$ being the matrix form of the pose $c_i = [v, \omega]^\top$, $c_i \in \mathbb{R}^6$, with $\omega \in \mathbb{R}^3$, the rotation

component and $v \in \mathbb{R}^3$ the translation component. The scale can be better estimated during the optimization process by considering the camera poses as elements of the Lie algebra $\widehat{c}_i \in \mathfrak{sim}(3)$ of the Similarity group $Sim(3)$ with:

$$\widehat{c}_i = \begin{bmatrix} [\omega]_\times + qI_{3\times 3} & v \\ 0 & 0 \end{bmatrix}, \quad (1)$$

where $q \in \mathbb{R}$ and $\sigma = \exp(q)$ being the scale factor [16]. Thus, now the pose becomes $c_i = [v, \omega, \sigma]^\top$, $c_i \in \mathbb{R}^7$. Estimating for the scale component alleviates the scale-drift effect when constructing the map incrementally [7].

The *3D points* can be parameterized either in *Euclidean coordinates* $P = [x, y, z]^T$, or using local *inverse depth* $P = [x/z, y/z, 1/z]^T$. Such point parameterization, as shown by [17], bounds the number of variables affected by updating the system with new measurements, thus reducing the incremental processing time.

IV. Tracking and Mapping

In order to obtain a good estimation, the points are tracked along a sequence of images and tested whether or not they are outliers. Through the process, the map points are assigned with *confidence* values depending on the number of successful observations. The confidence levels are used to decide whether a point is added or not to the *global map* or kept into a *local map* for further processing. In particular, every new 3D point is initially added to a local map, and remains there until its confidence reaches the threshold of minimum number of observations before being moved to the global map. The confidence of the points is increased with successful observation from any frame. The local map points are discarded after a period of inactivity, as they are considered outliers. If their confidence increases in the meantime (seen by new frames), they are added to the global map. By decoupling the two sets, all points are still used in the tracking process while only well observed points are utilized for global map estimation.

A. Feature Extraction

As the estimation of the 3D points together with the motion of the camera is inferred entirely from the sparse features matched across the set of 2D images, it is important to identify distinctive and repeatable features in each frame. Features that can not be matched across multiple frames do not contribute to the localization and mapping efforts and are therefore discarded. The particularities of the underwater medium induces several effects (such as light attenuation, blurring and low contrast) which can deteriorate the performance of some feature detectors/descriptors [18], [19]. In our approach, we currently use scale-invariant feature transform (SIFT) [20] features, which can be extracted using graphics processing unit (GPU) (e.g. Wu [21]) and are widely accepted as one of the highest quality feature descriptors [22] due to their high degree of invariance to scale and rotation, as well as being partially invariant to changes in illumination, noise, occlusions and small changes in the viewpoint.

B. Initialization

In order to start the tracking, the relative pose between two frames (not necessarily consecutive) has to be estimated, together with an initial set of triangulated points. Photo-metric correspondences between extracted features in both candidate frames are computed using Cheng et al. [23] cascade hashing approach based on the Euclidean distance between the descriptors. Ambiguous matches are discarded using Lowe's ratio test [20].

Relative poses are then computed through the estimation and decomposition of a geometric model. The selection of the most appropriate model should depend on the structure of the viewed scene, the type of motion and the knowledge of the intrinsic parameters of the camera. While homography (4-point algorithm [24]) should be used if the scene is planar/distant or motion is pure rotation, the selection between fundamental matrix (8-point algorithm [24]) and essential matrix (5-point algorithm [25]) depends on the knowledge of the intrinsic parameters of the camera.

To select the best model, we estimate both, homography and fundamental/essential matrix using all-to-all photometric feature matching and a parameter-free robust AC-Ransac [26] statistical method. The best model and its confidence level (automatically adapted to the noise) is estimated by following the Helmholtz principle of meaningful deviations and by regarding any model that is unlikely to be explained by chance as conspicuous. By comparing the confidence levels and number of inliers obtained from the estimation of both geometric models, we select the more appropriate. At the same time, using a robust estimation method also diminishes the influence of outliers on the estimation process.

As a final step of the initialization, the selected model is used to evaluate the behavior of individual matches with respect to the epipolar constraints [24]. Outliers and points with low parallax are omitted, while the rest are triangulated and added to the initial local map. Once the points are seen by a sufficient number of frames they are introduced to the global map.

C. Motion Tracking

Given that the tracking has been successful for the previous frame, the constant motion model is used to predict the pose of a new camera. Based on that, successfully tracked points from a previous frame are projected onto the new frame obtaining a prediction of where the correspondence in the new frame should be $\hat{z}_k = \text{proj}_k(c_i, p_j)$. The features extracted in the new image, which we call observations and denote with z_k, are potentially matched with the predicted features in their vicinity. The matching is successful if the

difference of their descriptors is below a threshold and passes the χ^2 test at 95% ($\text{TH}_m = 5.991$):

$$\left\|\hat{z}_k - zk\right\|^2_{\Sigma_k} < \text{TH}_m \,. \tag{2}$$

It is important to note that if the motion model does not describe well the real motion of the camera, the matches will not be found and the system can easily lose track of the points. In case too many points are unable to be tracked, the system automatically adjusts by widening the search area around the projections $\hat{z}_k$, and in the case that no matches were found, the system uses the last frame added to the global system to re-localize the current frame. This is done by using 3D-2D correspondences, and the pose of the new frame can be estimated using EPnP algorithm [27].

Once the matching is successful, the camera pose is optimized through a camera optimization step, where only the parameters of the camera pose are allowed to change. The newly optimized pose is further improved by attempting to match 3D points seen in neighboring frames (i.e. frames which share a sufficient number of 3D points with current frame) by projection (2) followed by another pose-points refinement.

D. Frame Insertion and Outlier Rejection

In order to maintain a scalable representation, only the frames which exhibit sufficient motion [6] are added to the global system. If the number of tracked features, compared to the last inserted frame, significantly decreases (below 70%), we introduce the frame to the global map and triangulate new points to strengthen the tracking. Similarly, in case of small number of tracked features (e.g. due to poor quality of images) we introduce new frames more frequently to increase the probability of successful tracking.

Once the frame is selected for insertion, previously unmatched features in neighboring frames are tested to match unmatched features in the current frame. As the poses of all the frames have been previously estimated, the search can be restricted to only pairs of features satisfying epipolar constraints. Successful matches are triangulated and inserted only if all matched observations from neighboring frames are consistent with the triangulated point.

In order to eliminate possible point outliers, a local refinement test using BA is required prior to the global map insertion. This includes the points visible in the current frame and all the camera poses from where the points in the local map have been previously observed. The optimization is restricted to the parameters of the current camera and local points, as this step is only used to eliminate possible point outliers before introducing them to the global map. Local points with high re-projection error in frames that they are observed are considered outliers and removed. The remaining points are added to the global map if and only if they achieved sufficient confidence (e.g. seen by sufficient number of frames).

In [6] the goal is to obtain real-time tracking, and thus global BA process is run concurrently with the tracker on distinct processing thread. When the tracker adds new points and cameras to the map, the global BA process is stopped to promote the real time operation. This can prevent the pipeline to provide a global optimization at every step. In contrast, our pipeline is concerned with providing the best estimate all the time so that the result can be used either on-board an autonomous underwater vehicle (AUV) to localize the robot and generate a good representation of the environment, or presented on a remotely operated vehicle (ROV) mission control panel to help the pilot control the acquisition in real-time.

V. Incremental Processing

Bundle adjustment is used to refine the camera poses and the 3D structure. In order to deal with the uncertainty, BA is formulated as probabilistic estimation and solved using non-linear least squares (NLS). Available BA software and applications are able to assemble and process the information from large amount of images and produce accurate solutions; *Bundler*, *Open MVG*, *Visual SFM*, to name just a few. Nevertheless, the majority of the existing applications are designed to be used offline, post-acquisition and do not provide any feedback about the uncertainty of the reconstruction.

SLAM++ [28], [29] is an open source library we are developing and which implements nonlinear least squares solvers for SLAM and SfM applications. The main advantage of the SLAM++ is that it implements incremental solutions for SLAM [10] using sparse block Cholesky factorization and recently also incremental solutions for BA [17]. Those are based on the highly efficient block matrix data-structure that facilitates structural and numerical changes of block matrices as well as arithmetic operations. Both the CPU and the GPU versions were shown to be faster than the SuiteSparse variants of element-wise implementations [30].

A. Bundle Adjustment Step

Formulating the BA as a probabilistic estimation method accounts for the uncertainties in the image measurements. It is common to assume that the point measurements are characterized by zero mean Gaussian noise and to formulate the BA problem as an optimization over a set of variables $\boldsymbol{\theta} = [\theta_1 \dots \theta_n]$, the camera poses and the 3D points forming the state $\boldsymbol{\theta} = [\mathbf{c}, \mathbf{p}]$. We want to find the optimal configuration satisfying a set of measurements, $\mathbf{z} = [z_1 \dots z_m]$, given by the re-projected points on the image. This can be done by finding the maximum a posteriori probability (MAP) estimate:

$$\boldsymbol{\theta}^* = \underset{\boldsymbol{\theta}}{\text{argmax}}\ P(\boldsymbol{\theta} \mid \mathbf{z}) = \underset{\boldsymbol{\theta}}{\text{argmin}}\ -\log(P(\boldsymbol{\theta} \mid \mathbf{z}))\,. \tag{3}$$

Each point observation is assumed to have zero-mean Gaussian noise with the covariance Σ_k and we measure the re-projection error: $e_k(c_i, p_j, z_k) = z_k - \text{proj}_k(c_i, p_j)$, with $[c_i, p_j] \subseteq \boldsymbol{\theta}$ where $\text{proj}(\cdot)$ is the projection function of a point, p_j, onto the camera c_i, and z_k is the actual pixel measurement. Note that even if this paper considers only

3D point observations, other measurement such as IMU or altitude sensors can be easily integrated into the estimation problem. The solution is obtained by solving the following NLS:

$$\boldsymbol{\theta}^* = \underset{\boldsymbol{\theta}}{\operatorname{argmin}} \frac{1}{2} \sum_{k=1}^{m} \left\| z_k - \operatorname{proj}_k(c_i, p_j) \right\|^2_{\Sigma_k} . \qquad (4)$$

Iterative methods such as Gauss-Newton (GN), Levenberg-Marquardt (LM) or Dog leg trust region are often used to find the solution of (4). In brief, these methods compute, at every iteration, a linear approximation of the problem, given a linearization point $\boldsymbol{\theta}^0$ and find a correction $\boldsymbol{\delta}$ towards the solution by solving a linear system $\Lambda\boldsymbol{\delta} = \boldsymbol{\eta}$. At every iteration, the linear system can be solved either using matrix factorization methods or gradient methods and obtain an update for the current linearization point, $\boldsymbol{\theta}^{i+1} = \boldsymbol{\theta}^i \oplus \boldsymbol{\delta}$. The process iterates until convergence.

B. Incremental Bundle Adjustment with Fast Covariance Recovery

In online applications, at every time step, new measurements are integrated into the system and a globally consistent solution can be found by solving the updated system. In our previous work [10], [29], we showed that, in SLAM applications, the updates only affect a small part of the system. Based on that, we proposed methods to solve the system in an incremental fashion, which translated into very efficient SLAM algorithms that can run onboard vehicles with limited computational capabilities. The proposed methods were based on techniques such as partial, block Cholesky factorization, which, at every step, performs matrix factorization on a small part of the system matrix affected by the update [10].

In general, the solution of the NLS is providing a mean estimate $\boldsymbol{\theta}^*$. In a probabilistic framework, it is important to consider the uncertainty associated with this solution, the covariance matrix in our case. It is a well known fact that the covariance matrix is very computationally expensive to obtain, given that it requires inversion of large matrices. Identifying which variables are affected by the updates led to very efficient solutions for the calculation of important elements of the covariance matrix in SLAM [14]. Our previous work shows that the marginal covariances, representing the uncertainty of each variable, and the cross-covariances of the last pose of the robot and all the other variables can be computed in a time which is a fraction of the solving time. The proposed algorithm outperformed the existing methods by two orders of magnitude and enabled SLAM applications where the state representation is maintained compact and the loop closure is obtained based on the state estimates [29].

Nevertheless, SLAM has a much simpler structure than BA. A typical BA problem is more *dense* and this is due to the fact that many (sometimes up to thousands) of points are seen from a single camera view. This makes incremental methods which are efficient when solving a SLAM problem to become inefficient when applied to BA. This is also the case of the matrix factorization, in SLAM Cholesky factorization has been shown to be very efficient. Whereas in BA, it is well known that the underlying variable graph is bipartite and can be separated in two parts, one corresponding to the camera variables and one to the point variables. An algebraic trick called Schur complement (SC) can be applied in this case. Given that the point measurements are independent and the fact that the number of camera poses is in general much smaller than the number of 3D points ($nc << np$), solving first for the camera poses and then for the points divides the problem to solving a small relatively dense system first and a large sparse system after. This partitioning of the computation is at the core of speeding up most of the batch BA solvers, but can make the matters difficult in incremental processing. Recently, we integrated into SLAM++ a method that can directly update the Schur complement representation with the new measurements obtained at every time step. This method brings up to threefold reduction in solving time in steps where the 3D points are seen from a small amount of cameras and when the size of the update increases, it gracefully degenerates to batch solving.

Moreover, a highly efficient covariance recovery technique was integrated. This method is based on an algebraic manipulation of the operations involved, so that resulting calculations take advantage of the sparsity of the problem, the previously calculated elements of the SC, and requires similar storage as the solving of the SC system. Details about the method can be found in our previous work [15] and an improved version of that will be made publicly available with the new release of the SLAM++ code http://sf.net/p/slam-plus-plus. This method was shown to provide marginal covariances at a time comparable with the solving time and to be more than one order of magnitude faster than the existing implementations.

Although the tracking system in the front-end implements several stages of outlier rejection based on local reprojection errors, the global optimizer can further check for the global consistency of the observations. This is done by using using *robust estimators*. The appealing property of robust estimators or M-estimators (maximum likelihood type estimators) [31] is their simple integration into the ordinary nonlinear least squares framework. The only change is that each observation z_k is assigned a weight. These weights then multiply the measurement covariances Σ_k in (4). With that, our back-end adds another level of robustness to the pipeline.

VI. Results

The proposed pipeline was tested on a large-scale underwater data set acquired by a setup comprising five GoPro Hero 4 cameras, while inspecting a shipwreck near the coast of Palamos, Spain. The total duration of the acquisition was 11 minutes. Views containing plain water and no structure (figure 1b) were automatically omitted by the tracker, as

(a) Structure (b) Open water

Figure 1. Captured frames: (*a*) registered and (*b*) discarded.

they convey no information to be integrated into the system. Binary masks were used to conceal the view of the robot seen on two cameras. For the back-end performance analysis in section VI-A we aimed to maximize the size of the BA system, therefore we use the images form all five cameras and, from that, a total of 1772 images were successfully registered offline to produce 455776 observations of a total of 170018 3D points. However, our current version of the tracker only supports single camera systems. Therefore for the tracking performance analysis we used the video sequence of the camera that captures the most of the visible structure. In total, the sequence contains 5480 frames out of the total of 16000. The extension to handle a multi-camera system will be implemented as future work.

A. Back-end Performance

We first tested the incremental optimization and covariance recovery introduced in section V and implemented in SLAM++. For that, we compared performance with two popular NLS solvers in computer vision, g2o [32] and Ceres [33]. The first one, g2o, can solve BA and SLAM problems out-of-the-box. TheBA implementation is restricted to batch solving. The Ceres solver received much attention, as it is used in Google's 3D Maps and Street View applications. It is mostly focused on batch solving. SLAM++, on the other hand, implements incremental solutions for SLAM and recently for BA.

In order to guarantee repeatability and fairness of the evaluation, the same input data was used by all the solvers. Therefore, for time comparison we processed the images with a similar pipeline as described in section IV but instead of actually feeding an incremental, global BA optimizer, the measurements were stored in a file. This file was then parsed incrementally by all three solvers and the data processed incrementally. Here we need to make the distinction between incremental processing and incremental solving. While the former refers to performing the global optimization every time new information is available, the latter refers to actually updating and solving the system incrementally (partially). For example with SLAM++ we process and solve incrementally but with g2o and Ceres we process incrementally and solved batch.

The main characteristic of this dataset is that each point is seen in only a few images, and that makes the updates on the incremental optimizer very efficient. Table I shows how the incremental solver in SLAM++ outperforms the other solvers by a factor of 1.5× while having comparable root-mean-square error (RMSE). Observe that the covariance recovery times using g2o and Ceres are fairly high, clearly showing that those solvers are not suitable for this purpose.

Figure 4 shows an example of how the covariance value helps identifying poorly sampled regions of the reconstruction. For that, we simply color-coded the values of the determinant of marginal covariances for each point in the global map (purple–high uncertainty, red–low uncertainty). This can help in re-planning trajectories of the robot to re-sample high uncertainty regions.

B. Tracking Performance

The tracking approach in section IV successfully reconstructed a scene (figure 5) containing 65355 global 3D points and 801 key-frames out of a total of 5480 frames. Observe that the tracker samples the key-frames, only maintaining the informative ones in the global representation. This is very important for the efficiency of the on-board processing where the computational resources are limited. We further analyze the global vs. local maps. As shown in figure 2, we can see that the number of global 3D points is constantly rising, while the number of local 3D points is limited as inactive points are continuously removed from the map.

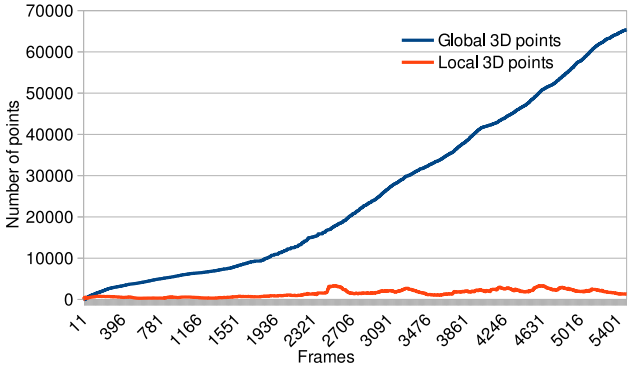

Figure 2. Number of global/local points with respect to number of frames processed

As already mentioned earlier, our tracking relies on both, matching with motion model and subsequent matching with points in a local neighborhood map. Both steps are highly important for obtaining an accurate pose estimation which enables better outlier rejection. This can be seen in figure 3, where the percentage of successfully matched points with each strategy is shown. After each key-frame insertion into global map, the proportion of points matched using the motion model increases, as the motion and points have been recently optimized by global optimization. As frame-to-frame tracking gradually accumulates error, the number of matches using a motion model decreases. However, the camera refinement step enables better matching with the local neighborhood map, resulting in an increased percentage of matched points.

Solver	g2o	Ceres	Batch SLAM++	Incremental SLAM++
Solve Time	19578.700 sec	15138.994 sec	20876.609 sec	541.680 sec
Covariance Recovery Time	25.65 hours	2478.973 hours	2.839 hours	2.839 hours
RMSE	2.616 px	8.466 px	12.575 px	6.022 px

Table I
EVALUATION OF THE INCREMENTAL PROCESSING ON THE BOREAS DATASET. NUM. OF CAMERAS: 1772, NUM. OF POINTS: 170018, NUM. OF OBSERVATIONS: 455776.

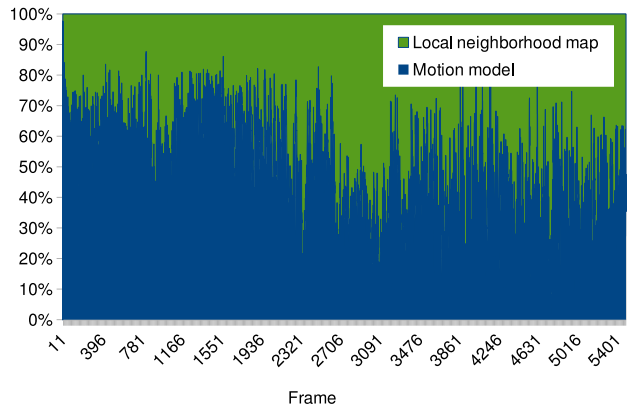

Figure 3. Relation between matches obtained using motion model/local neighborhood map

Our current implementation was developed by extending the open-source library OpenMVG [34]. The CPU implementation of the SIFT feature extractor, taking on average 0.2 s per frame could be significantly improved using GPU implementation (e.g. [21]) which we plan to integrate in the future. Matching and triangulation take 0.15 s and the rest of the tracking additional 0.1 s. While the numbers do not indicate real-time performance it is worth noting that the code could be further optimized. Some parts such as feature extraction and matching can be parallelized to gain on time performance. This experiment was performed on Intel Core i7-5500U processor and 8 GB RAM.

VII. Conclusions

This paper contributes to the field by demonstrating the feasibility of mission-time 3D reconstruction and uncertainty estimation. This is the critical component missing for performing quality-aware data acquisitions, which will increase the quality of both the final acquired data and the survey efficiency as well as concurrently diminish the possibility of performing unsatisfactory optical surveys. In the future we plan to speed up the tracking part by parallelizing parts of the pipeline to obtain real-time, and to exhaustively test the entire incremental pipeline on different scenarios. The final goal is to integrate this system into our Girona500 and SparusII underwater robots and use the uncertainty-aware 3D reconstruction to guide their missions.

Acknowledgments

This research was supported by the Australian Research Council Centre of Excellence for Robotic Vision (project number CE140100016), the Spanish National Project OMNIUS (Lightweight robot for OMNidirectional Underwater Surveying and telepresence MINECO CTM2013-46718-R), the Robocademy European project (FP7-PEOPLE-2013-ITN-608096) and the Czech Republic Technological Agency project TE01020415 V3C. The authors would also like to thank the developers of openMVG [34] library for the support and providing the library as open-source.

References

[1] N. Gracias, P. Ridao, R. Garcia, J. Escartin, M. L'Hour, F. Cibecchini, R. Campos, M. Carreras, D. Ribas, N. Palomeras *et al.*, "Mapping the Moon: Using a lightweight AUV to survey the site of the 17th century ship 'La Lune'," in *In Proc. MTS/IEEE OCEANS,*. IEEE, 2013, pp. 1–8.

[2] C. Beall, B. J. Lawrence, V. Ila, and F. Dellaert, "3D reconstruction of underwater structures," in *Int. Conf. on Intelligent Robots and Systems (IROS)*. IEEE, 2010, pp. 4418–4423.

[3] M. Johnson-Roberson, O. Pizarro, S. B. Williams, and I. Mahon, "Generation and visualization of large-scale three-dimensional reconstructions from underwater robotic surveys," *Journal of Field Robotics*, vol. 27, no. 1, pp. 21–51, 2010.

[4] J. D. Hernández, K. Istenič, N. Gracias, N. Palomeras, R. Campos, E. Vidal, R. García, and M. Carreras, "Autonomous underwater navigation and optical mapping in unknown natural environments," *Sensors*, vol. 16, no. 8, p. 1174, 2016.

[5] B. Triggs, P. F. McLauchlan, R. I. Hartley, and A. W. Fitzgibbon, "Bundle adjustment—a modern synthesis," in *Int. workshop on vision algorithms*. Springer, 1999, pp. 298–372.

[6] R. Mur-Artal, J. Montiel, and J. D. Tardós, "ORB-SLAM: a versatile and accurate monocular SLAM system," *IEEE Transactions on Robotics*, vol. 31, no. 5, pp. 1147–1163, 2015.

[7] H. Strasdat, J. Montiel, and A. J. Davison, "Scale drift-aware large scale monocular SLAM," *Robotics: Science and Systems VI*, 2010.

[8] V. Indelman, R. Roberts, C. Beall, and F. Dellaert, "Incremental light bundle adjustment," in *British Machine Vision Conf. (BMVC)*. BMVA Press, 2012, pp. 134.1–134.11.

[9] M. Kaess, H. Johannsson, R. Roberts, V. Ila, J. J. Leonard, and F. Dellaert, "iSAM2: Incremental smoothing and mapping using the Bayes tree," *Intl. J. of Robotics Research*, vol. 31, pp. 217–236, Feb. 2011.

[10] L. Polok, V. Ila, M. Šolony, P. Smrž, and P. Zemčík, "Incremental block Cholesky factorization for nonlinear least squares in robotics," in *Robotics: Science and Systems (RSS)*, 2013.

[11] S. Thrun, Y. Liu, D. Koller, A. Ng, Z. Ghahramani, and H. Durrant-Whyte, "Simultaneous localization and mapping with sparse extended information filters," *Intl. J. of Robotics Research*, vol. 23, no. 7–8, pp. 693–716, 2004.

[12] R. Eustice, H. Singh, J. Leonard, and M. Walter, "Visually mapping the RMS Titanic: Conservative covariance estimates for SLAM information filters," *Intl. J. of Robotics Research*, vol. 25, no. 12, pp. 1223–1242, Dec 2006.

[13] M. Kaess and F. Dellaert, "Covariance recovery from a square root information matrix for data association," *Robotics and Autonomous Syst.*, 2009.

[14] V. Ila, L. Polok, M. Šolony, P. Smrž, and P. Zemčík, "Fast covariance recovery in incremental nonlinear least square solvers," in *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, May 2015, pp. 4636–4643.

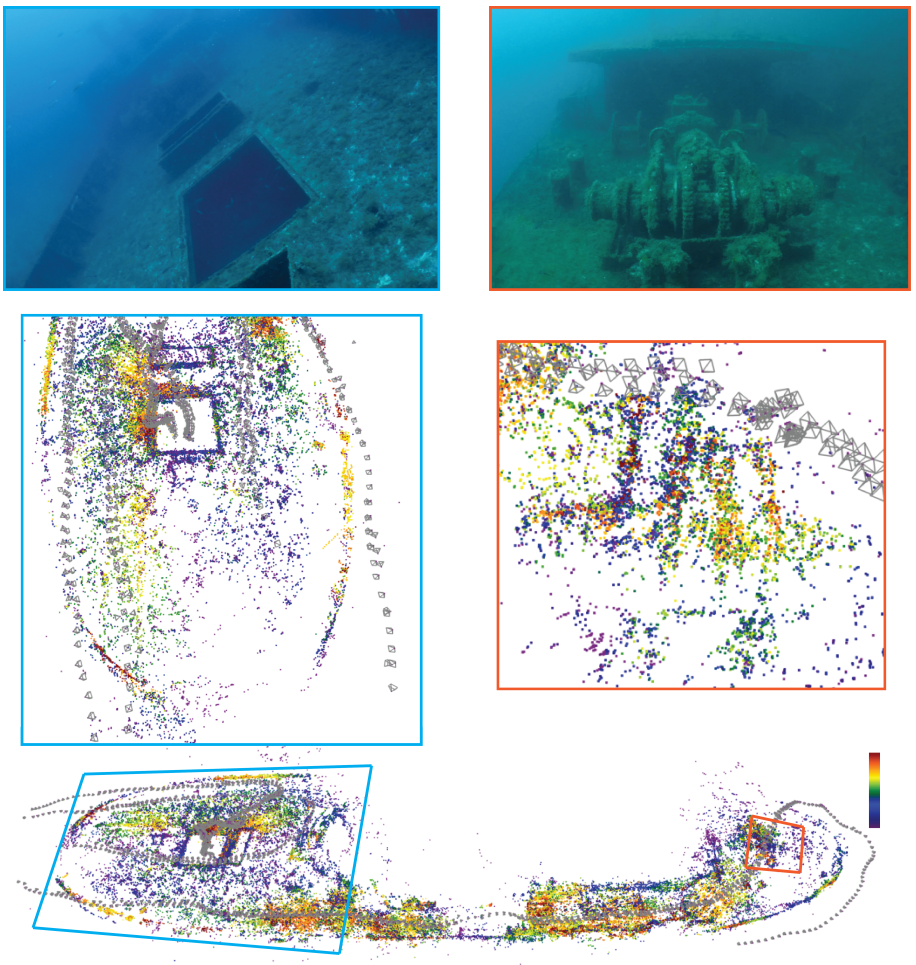

Figure 4. Final sparse 3D reconstruction using *multi-camera dataset* with color-coded magnitude of the uncertainty estimation (violet-high uncertainty, red-low uncertainty)

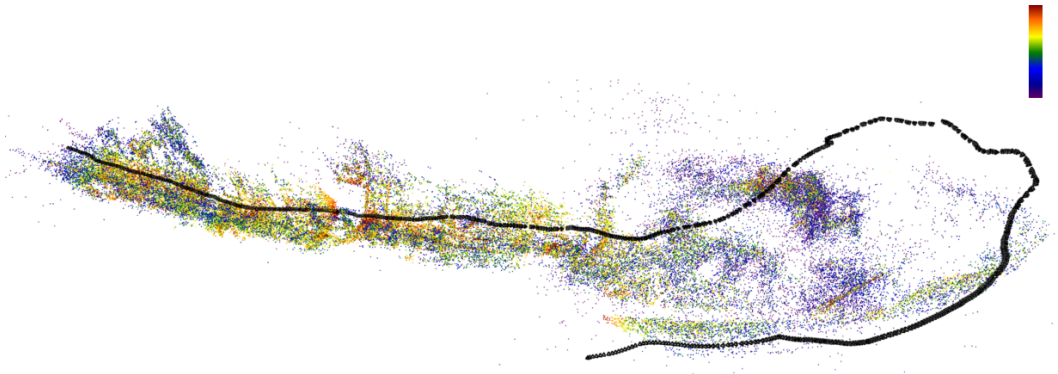

Figure 5. Final sparse 3D reconstruction *obtained with tracking* with color-coded magnitude of uncertainty estimation (violet-high uncertainty, red-low uncertainty)

[15] L. Polok, V. Ila, and P. Smrž, "3D reconstruction quality analysis and its acceleration on GPU clusters," in *Proc. of the European Signal Processing Conf.* IEEE, 2016.

[16] H. Strasdat, "Local accuracy and global consistency for efficient visual SLAM," Ph.D. dissertation, Imperial College London, UK, 2012.

[17] L. Polok, V. Lui, V. Ila, T. Drummond, and R. Mahony, "The effect of different parameterisations in incremental structure from motion," in *Australian Conf. on Robotics and Automation (ACRA)*, December 2015.

[18] R. Garcia and N. Gracias, "Detection of interest points in turbid underwater images," in *OCEANS, 2011 IEEE-Spain.* IEEE, 2011, pp. 1–9.

[19] A. Q. Li, A. Coskun, S. M. Doherty, S. Ghasemlou, A. S. Jagtap, M. Modasshir, S. Rahman, A. Singh, M. Xanthidis, J. M. O'Kane *et al.*, "Vision-based shipwreck mapping: On evaluating features quality and open source state estimation packages," in *OCEANS 2016 MTS/IEEE Monterey.* IEEE, 2016, pp. 1–10.

[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] C. Wu, "SiftGPU: A GPU implementation of scale invariant feature transform (SIFT)," 2007.

[22] K. Mikolajczyk and C. Schmid, "A performance evaluation of local descriptors," *IEEE transactions on pattern analysis and*

machine intelligence, vol. 27, no. 10, pp. 1615–1630, 2005.
[23] J. Cheng, C. Leng, J. Wu, H. Cui, and H. Lu, "Fast and accurate image matching with cascade hashing for 3D reconstruction," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 1–8.
[24] R. Hartley and A. Zisserman, *Multiple view geometry in computer vision*. Cambridge university press, 2003.
[25] D. Nistér, "An efficient solution to the five-point relative pose problem," *IEEE transactions on pattern analysis and machine intelligence*, vol. 26, no. 6, pp. 756–770, 2004.
[26] L. Moisan, P. Moulon, and P. Monasse, "Automatic homographic registration of a pair of images, with a contrario elimination of outliers," *Image Processing On Line*, vol. 2, pp. 56–73, 2012.
[27] L. Kneip, D. Scaramuzza, and R. Siegwart, "A novel parametrization of the perspective-three-point problem for a direct computation of absolute camera position and orientation," in *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*. IEEE, 2011, pp. 2969–2976.
[28] L. Polok, M. Šolony, V. Ila, P. Zemčík, and P. Smrž, "Efficient implementation for block matrix operations for nonlinear least squares problems in robotic applications," in *IEEE Intl. Conf. on Robotics and Automation (ICRA)*. IEEE, 2013.
[29] V. Ila, L. Polok, M. Šolony, and P. Svoboda, "SLAM++-A highly efficient and temporally scalable incremental SLAM framework," *Intl. J. of Robotics Research*, vol. Online First, no. 0, pp. 1–21, 2017.
[30] L. Polok, V. Ila, and P. Smrž, "Fast sparse matrix multiplication on GPU," in *Proc. of the High Performance Computing Symp.* ACM, 2015.
[31] P. J. Huber, *Robust statistics*. Springer Heidelberg, 2011.
[32] R. Kümmerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard, "g2o: A general framework for graph optimization," in *Proc. of the IEEE Int. Conf. on Robotics and Automation (ICRA)*, Shanghai, China, May 2011.
[33] S. Agarwal and K. Mierle, "Ceres solver," http://ceres-solver.org/, 2012.
[34] P. Moulon, P. Monasse, R. Marlet, and Others, "Openmvg. an open multiple view geometry library." https://github.com/openMVG/openMVG.