COOPERATIVE MAPPING FOR LUNAR EXPLORATION BY A TEAM OF HETEROGENEOUS ROBOTS

Matteo De Benedetti, Fabio Polisano, and Shashank Govindaraj

Space Applications Services, Leuvensesteenweg 325, 1932 Brussel, Belgium

ABSTRACT

Cooperation between robots is emerging as a promising technique to improve performances, but it still remains unused for robots in space exploration missions. This paper presents a modular approach in which Cooperative Localization and Mapping based on Graph Optimisation and inter-robot loop closure detections for heterogeneous lunar robots is proposed. The approach relies on a smoothing technique, based on graph optimisation and a module to perform features-based matching between sub-maps of different robots. A global map is then built using the optimised poses. The method proposed shows a significant decrease in the error of the pose estimation in different scenarios. Thanks to the design choices, the number of nodes increases less than in the classical approaches, making the optimisation process faster. It has also been shown how the optimised trajectories can be used to obtain a qualitatively improved representation of the environment using multiple robots. This solution is developed within the scope of Cooperative Robots for Extreme Environments (CoRob-X) which is an EC-funded project with the objective of demonstrating how a multi-agent robotic team can explore a lunar lava tube. Three Robotic Explorer Units (REUs) form a team of heterogeneous robots, with different locomotion systems and sensors. SLAM is still a crucial problem in extreme environments such as the Moon surface.

Key words: Collaborative Robotics, Mapping, SLAM.

1. INTRODUCTION

The paper first introduces the CoRob-X project and its scenarios, that are the context in which the Collaborative Mapping (CMAP) component was developed. This sections also presents the state of the art of Collaborative SLAM, to give additional context to the work described in the paper. Next, Section 2 describes the proposed architecture, from a general overview of its interfaces, to a detailed explanation of each sub-component. In Section 3, the results of the tests with different datasets and in simulation are presented. The performances are analysed both in terms of mapping and localisation performances, trying to identify the main differences and challenges. Lastly, Section 4 collects all the conclusions drawn from the previously shown tests and how the component performs in the various cases. Additionally, possible future and interesting activities are presented.

1.1. CoRob-X Project

Cooperative Robots for Extreme Environments (CoRob-X) is an EC-funded project within the framework of the SRC Space Robotics Technologies. The project is carried out in cooperation with several companies, universities, and research centers in Europe. The project objective is to demonstrate that a multi-agent robotic team could achieve tasks otherwise impossible or carries high risk for a single robot. In particular, the field of interest is the exploration of lunar lava tubes, volcanic caves accessible from the Moon surface. Such caves are of great scientific interest as they are the main candidates to establish a human settlement, protected from space radiations. Similarly, the development of these technologies could easily find Earth applications in the future, like the exploration of human safety-critical places, like mines and tunnels.

The ground demonstration of the project reproduces a lunar mission, further divided into several Mission Phases:

- Mission Phase 1 (MP1): all the rovers explore and map the surrounding of the lava tube to find the optimal entry location.
- Mission Phase 2 (MP2): the LUVMI-X rover deploys its Payload Cube to generate a preliminary map of the skylight and floor of the lavatube.
- Mission Phase 3 (MP3): the Coyote-III is lowered into the lava tube from a rappelling system mounted on Sherpa-TT.
- Mission Phase 4 (MP4): the Coyote-III explores the tunnel of the lava tube.

For the Collaborative Mapping component described in this paper, the most relevant phase is the mission Phase 1 (MP1). In this scenario, the objective is to explore the

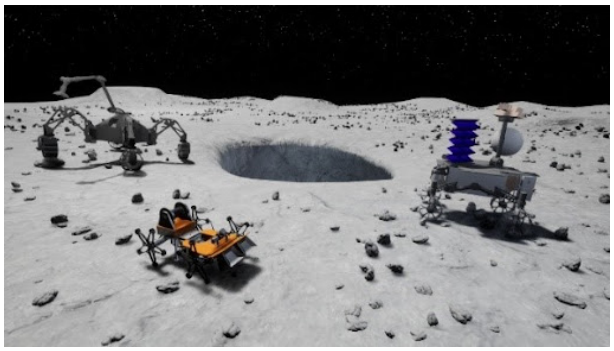

Figure 1. REUs mapping the surroundings of the lava tube.

surroundings of the lava tube and to build a map of the environment around the access of the lava tube. To do so, three Robotic Explorer Units (REUs) will cooperate, sharing sensors data, to fuse independent rover maps and improve the quality of map estimation and to speed up the map building process using overlapping information. The involved rovers are SherpaTT (REU-1) and Coyote-III (REU-2) from DFKI, and LUVMI-X (REU-3) from Space Applications Services. Each rover is equipped with a variety of sensors. In particular, all of them have either a stereo/depth cameras or a Lidar, wheel odometry, and an additional stereo camera for visual odometry. Each REU processes data from its sensors to create a detailed 3-D map of the environment and estimate its pose. During the whole operation, the robots communicate with each other and exchange map and pose data. Thus, after the exploration, a joint map is built. A rendering of the scenario, showing the different REUs, is showed in Fig. 1, while the final demonstration will be held in Lanzarote in January 2023, in a test site with an actual lava tube.

1.2. State of the Art Collaborative SLAM

In the Autonomous Mobile Robotics field, a crucial issue is to give the robot the capacity to recognize its position and orientation in space and to build a map of the environment, to perform several kinds of tasks. The problem to simultaneously localize the pose of a robot and map the environment around it is known as Simultaneously Localization And Mapping (SLAM). It is a complex problem since it is necessary for a good pose estimation to make a good map estimation and vice versa. Single robot SLAM is a well-addressed problem in literature [1]. Depending on environmental conditions and hardware architecture of the robot different approaches are taken. One of the popular widespread approach is the dynamic filtering through Extended Kalman Filter (EKF) [2]. In the last years, however, other approaches become popular, like Unscented Kalman Filters (UKF) [3], Rao–Blackwellized Particle (RBPF) Filters [4], Sparse Information Filters (SIF) [5]. The most popular today is probably the Pose Graph Optimization approach [6] [7] for its scalability to large systems and representation flexibility. While these solutions are widely used and robust,

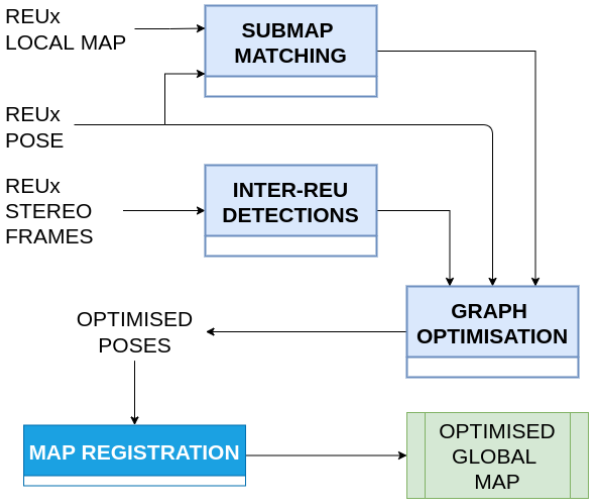

Figure 2. High Level Architecture of CMAP.

the problem of different robots performing SLAM at the same time and place has fewer studies and applications around it. This kind of problem is usually addressed as Cooperative SLAM (C-SLAM). A systematic review is taken in [8]. C-SLAM seems to be a very promising approach to overcome the single-robot SLAM limitations. It could be useful in obtaining better estimation accuracy, especially in extreme environmental conditions, like space exploration, difficult-to-reach, or dangerous places. Solutions to decrease computational time can be explored through distributed approaches. Also, it can be a robust approach in case of a single robot failure. C-SLAM approaches are usually an extension of a single robot SLAM problem. So classical approaches like EKF [9], UKF [10], RBPF [11], SIF [12] are studied. The most promising one, anyway, seems to be the Pose Graph approach [13], which will be the object of study of this paper.

2. COLLABORATIVE MAPPING

This section will present the development of the Collaborative Mapping solution, referred to as CMAP, starting from an high-level overview of the overall architecture in ROS and the interfaces between the various subcomponents. Then, each subcomponent will be described in details.

2.1. Software Architecture

The Collaborative Mapping component (CMAP) was developed win the ROS framework to make use of its data-handling and communications functionalities, and take advantage of its ease of integration and deployment. A high-level architecture diagram, Fig. 2, shows the various components and their interfaces.

The main idea behind this architecture is to use various inputs, namely maps, poses and inter-robots detections (meaning a relative pose from a robot to another), to generate constraints between the poses of the REUs. These constraints and poses are used to populate a pose-graph, that is then optimized to generate a more accurate trajectory. The maps collected from each REUs are then registered in these new collection of optimised poses to generate a global map, that is more precise than the single ones, and brings together the data from all the REUs.

To better understand the logic of the CMAP component, this can be first divided into two main blocks. A first block, referred to as Submap Matching, is executed for each REU. It takes as inputs the local maps (that can be either Lidar scans or a pointclouds from a stereo camera), REU pose and inter-REU detections, and generates the appropriate constraints. These constraints are then used in the next block, the Graph Optimization component, which creates a pose-graph based on the constraints and poses of the REUs, and optimises it to generate the better poses.

2.2. Graph Optimisation Component

SE(3) objects have been chosen to model the vertices of the graph robots poses and R^3-vectors for landmarks. The same parametrisation has been chosen for edges, with SE(3) objects for edges between robot poses and R^3-vectors for edges between landmarks. In particular, all positions considered in the Graph are considered with respect to a fixed navigation frame East-North-Up (ENU). It's usual to perform error minimization on a manifold for blocks that span over a non-Euclidean space. A manifold is a topological space that locally resembles Euclidean space near each point. Mathematical details are reported in [7][14]. Indeed in the SLAM problem, the state lives in SE(3), belonging to the Lie algebra. To define a cost function is necessary to define an error function and measurement function. The measurement function between two REU pose nodes $\mathbf{X}_x(t)$ and $\mathbf{X}_y(t)$, can be expressed as:

$$\mathbf{h}^p_{xy}(t) \doteq \mathbf{X}_y(t) \ominus \mathbf{X}_x(t) \tag{1}$$

and then the error function is simply given by the relative transformation between the measurement and the measurement function:

$$\mathbf{e}^p_{xy}(t) \doteq \mathbf{z}^p_{xy}(t) \ominus \mathbf{h}^p_{xy}(t) \tag{2}$$

Given the error function then it is possible to evaluate different cost functions. In general, given a cost function **F(x)** a minimization problem tries to find $\mathbf{x}^*$ such that:

$$\mathbf{x}^* = \underset{x}{\operatorname{argmin}}\, \mathbf{F}(\mathbf{x}) \tag{3}$$

In the scope of this work three cost functions have been investigated. The first one is the classical Least Squares, described by:

$$\mathbf{F}(x) = \sum_t \mathbf{e}^T_{xy}(t)\Omega_{xy}(t)\mathbf{e}_{xy}(t) =$$
$$= \sum_t \rho_2\left(\sqrt{\mathbf{e}^T_{xy}(t)\Omega_{xy}(t)\mathbf{e}_{xy}(t)}\right)$$

With:

$$\rho_2(x) \doteq x^2 \tag{4}$$

This cost function has been widely applied in different fields and in vary range of problems. Thus, the error vector **e** has a quadratic influence on **F**, so that outliers have a major negative impact on the cost, while they should be rejected. In order to reject outliers it has been considered the Huber robust cost function. It can be defined changing the ρ parameter like this:

$$\rho_H(x) \doteq \begin{cases} x^2 & if\ \ |x| < b \\ 2b|x| - b^2 & else \end{cases} \tag{5}$$

where b is a threshold for outlier rejection, so that **F** is quadratic for small $|x|$, but linear for large $|x|$. It has been considered also more robust cost function like Cauchy one. It is described by:

$$\rho_C(x) \doteq \frac{c^2}{2}\left(1 + \left(\frac{x}{c}\right)^2\right) \tag{6}$$

With respect to Cauchy cost function, Huber kernel has to advantage that it is still convex and thus does not introduce new local minima in **F**. Constraints have already been defined from a mathematical point of view. However, it is necessary to distinguish between different kinds of constraints, even if they are the same kind of mathematical object. The first kind of constraint is the *odometry edge*. It gives the relative transformations between 2 consecutive poses of the same REU. Then it could be possible to use also as constraints the relative transformations between poses and landmarks as between landmarks themselves. However, it has been deciding to analyze also a different approach. As discussed above, the computation time for the minimization of the cost function grows exponentially with the growth of the state vector. So it has been decided to pre-process the information coming from landmarks into the Submap Matching module to extrapolate a direct constraint coming from the *base node*, centered in the origin of ENU frame, to a particular node. This allows lowering the rate of growth of the calculation time significantly. The last kind of constraint, proper of a C-SLAM problem, is the Inter Robot Detection constraint. In fact, through the detection coming from recognizing the other robot in the space and

computing the relative transform, it is possible to give the robot a new kind of constraint that wouldn't be available in a single SLAM approach. It is modeled like an edge between a REU_x node to a REU_y node, with x=1,2,3, y=1,2,3 and $x \neq y$.

2.3. Submap Matching Component

The role of the Submap Matching component is to generate pose constraints, for each REU, from the local maps collected over time. This is accomplished in a sequence of steps, as shown in Fig. 3.

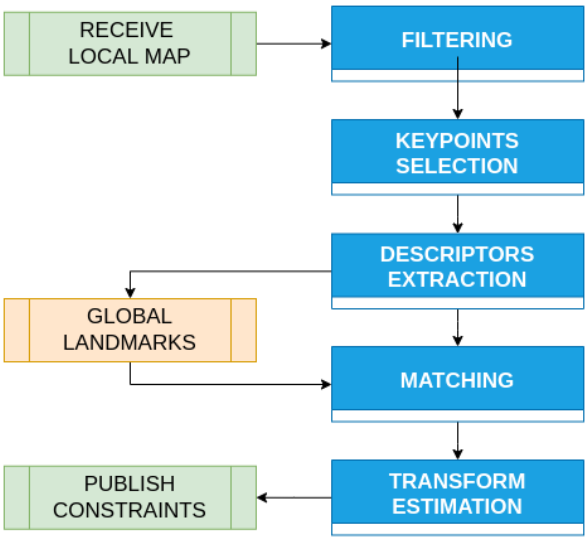

Figure 3. Flowchart of the Submap Matching Component.

Once a local map is received (meaning sensors data from a stereo camera or Lidar), they undergo a filtering step to remove NaN, down-sampled using voxellisation and prepare the appropriate data structures for the next steps, such as computing their surface normals.
The pointcloud is a very heavy structure, so to ease computation only a handful of points are selected using Keypoints Selection methods, such as the Harris Detector [15], which selects interesting points based on a corner mask. For each of these points then a descriptor is computed. As the name suggest, a descriptor is used to describe the neighborhood of a point, and allows to compare them, to find the same point in a different pointcloud. Several Descriptor Extraction algorithms were tested, the final selection lead to the use of the Fast Point Feature Histograms (FPFH) [16] if the pointcloud is purely geometrical (meaning the points only contain XYZ data), and the Color SHOT (CSHOT) descriptor [17] if the points contain also color information (XYZRGB), which is often the case for data coming from stereo-cameras.
At this point, the sensor data is now a much smaller, and information rich, data structure made of keypoints, normals and descriptors. This data is saved to progressively build a collection of landmarks as the REU moved. It is also shared with the other REU, so that each robot has information from all the places it and the other REU visited. These are referred to as Global Landmarks. To reduce the size of this data, if a point is close to an already saved point, and has a similar descriptor, it means it is likely a duplicate of the same landmark and will not be added. This step greatly helps to keep a manageable size and avoids an overload of data in a small region of space, which would not help when performing matching and pose estimation.
The last two remaining steps are the core of this component. The keypoints and descriptors computed from the last received local map are matched against the Global Landmarks. To ease computation, only a small window of the Global Landmarks is used, selected around the last pose of the REU. The matching is performed looking for the most similar pair of descriptors. It is further refined with a method called *Mutual Consistency Check*: for a match to be valid, both the points have to mutually have each other as their best match.
Once a collection of matches between a target and source pointclouds are found, the transformation between them can be estimated with several methods. After a first investigation phase, it was decided to use a combination of the Singular Value Decomposition (SVD) and Iterative Closest Point (ICP). The SVD is an optimisation algorithm and was used to provide a first guess of the transformation as additional input of the ICP, that would then refine it.
Lastly, this transformation is converted into a constraint, and covariance data is added based on a score metric from the ICP.
These steps were implemented using the Point Cloud Library (PCL) [18] in C++ and wrapped in a ROS node, along with all the necessary parameters and interfaces to be fully integrated in the CMAP architecture.

2.4. Inter Robots Detection Component

This component provides an additional constraint in the form of a relative transformation from one REU to the other. At the moment this component is simulated from ground truth data and adding a white noise on all position and orientation components.
There are various strategies to implement this. Space Applications Services is maintainer of Infuse, which provides three possible solutions: Aruco pose detection, rover wheeled pose estimation, and model based detection from camera images [19], [20]. Additionally, a Deep Learning based alternative is being tested, based on PoseCNN [21].
This component is intended to be developed and tested for the first field test campaign of CoRob-X in September 2022, where all the robots will be together and real images from their respective caeras can be used.

2.5. Optimization Approaches

Smoothing problems require an intensive computational resource exploitation to be solved and the solution to the problem could still require a time not suitable to run the algorithm online. In order to guarantee the modularity of the architecture, the first approach is to run the Submap Matching and the Graph Optimization components sequentially. In this way all the constraints coming from the first module go to the second and the optimization can be fully performed offline, eventually changing parameters as the cost function or the solver. A second approach is to run the two components in parallel, providing in feedback the latest optimized poses to the Submap Matching, which, using a more accurate pose estimation, could perform a better estimation of the correct transform. This last approach is then also suitable to be performed online. The delay provided by the computational time required by the Graph Optimization could nevertheless bring to a larger error if it is not small enough. Furthermore, another aspect to take into consideration is that in this case the two components are strictly coupled and then changing the parameters is not possible, therefore less flexibility.

3. RESULTS

In this section simulation and experimental results will be discussed. To test the entire CMAP stack, it has been necessary to build a simulation environment suited for perception purposes, so carrying on both dynamics models of a wheeled robot and different sensors. The most suitable solution has been identified in Webots simulator [22]. To further validate the CMAP stack, datasets from real scenarios were used. In particular, it has been initially tested with a dataset collected on a previous project of the company, Pro-Act. Then, datasets from the ADE project were used, which are of particular interest since they employ the same robot that will be used in CoRob-X, the Sherpa-TT. In all of the datasets presented only Submap Matching constraints have been considered. A standard metric for evaluating the performance has been considered. In particular the Mean Absolute Error (MAE) and the Mean Squared Error (MSE) of Position and Attitude estimation are considered. For each of the test cases proposed, the estimation of the trajectory of the REUs is evaluated in a standard graph that represents the MAE across the components X, Y and Z of the position or Roll, Pitch and Yaw over the distance travelled. To evaluate the performances of the stack, every test is presented on several graphs. A first type that allows a quantitative evaluation of the accuracy of the solution. It plots the squared norm of the position error over the distance travelled. This allows for a quick evaluation of the overall accuracy, and the slope of the curve shows the drift as the robot moves. The usual comparison is made against a localisation drift of 2% which is the standard in the state of the art. Another graph that is used frequently to evaluate localisation algorithms shows the planar path of the robot from a ground truth source and the estimation. This gives a better idea of the trajectory the robot is performing and allows a quick evaluation of its accuracy, both in terms of the error and the shape. To evaluate the improvements obtained in mapping, a qualitative approach has been followed. A standard feature easily recognizable has been placed into the map and the distortion can be evaluated. The results will be discussed in detail in the next subsections.

3.1. Simulation

Webots is an open-source three-dimensional mobile robot simulator. It comes equipped with different robot models and sensors combined with an intuitive interface to build custom environments, called Worlds, and with a complete and well-documented set of API to interface with C++ code and ROS framework. To evaluate the performance of the stack, an unstructured World poor of features has been built. The terrain has been chosen particularly uneven and jagged. It has been used a model of Husky A200^{TM} Robot from Clearpath Robotics to simulate two REUs. Each REUs is equipped with a GPS sensor to extract data on position, an Inertial Unit sensor to obtain the orientation, a model of Puck LITE^{TM} lidar from Velodyne Lidar, and a model of Multisense 21 stereo-camera from Carnegie Robotics. Data on position and orientation has been post-processed after the simulation to add White Gaussian Noise and simulate also the drift in odometry. The only features in the environment are rocks of a different dimension, as it could be a possible lunar scenario. An overview of the custom World can be seen in Fig. 4.

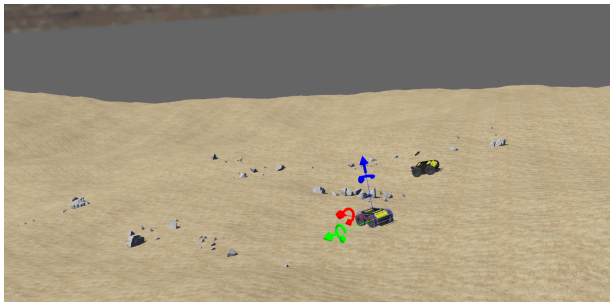

Figure 4. Overview on custom Webots World

In this scenario, REUs performed a trajectory with the shape of an X, that can be appreciated in Fig. 5. The Mean Squared Error (MSE) on REUs trajectory before the optimization was 0.877 m on REU1 and 0.849 m on REU2. After the optimization, the MSE on REU1 is reduced to 0.224 m and to 0.237 m on REU2. As shown in Fig. 6, the MAE is considerably reduced.

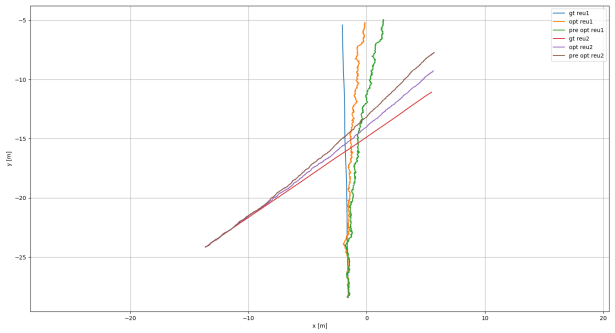

Figure 5. Overview on the XY plane of Groundtruth trajectories (Blue and Red), pre-optimization (Green and Brown) and after-optimization (Orange and Purple)

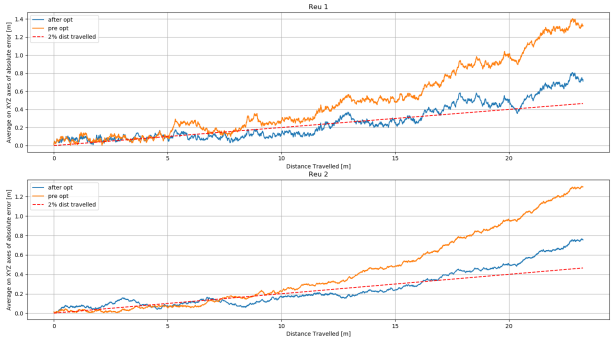

Figure 6. Average of the absolute error on XYZ before and after optimization, compared with the 2% of the distance travelled

3.2. PRO-ACT

Pro-Act is an EC-funded project within the Horizon 2020 programme that aimed to develop and demonstrate cooperation and manipulation capabilities between three robots for assembling an in-situ resource utilisation (ISRU) plant [23], [24]. Several multi-robot datasets were colleted during the project that could be used to start testing the CMAP on real world data. The selected dataset was collected outdoors and the robots used mounted both a lidar and stereo camera, which allowed to test several descriptors for both geometrical and also RGB pointclouds. The localisation used was coming from the wheel odometry and the groundtruth from an ultrasound localisation system. In this case, only one of the robots managed to find constraints along the trajectory and this resulted in a reduced localisation error, as show in Fig. 7, while the orientation error remained very similar.

3.3. ADE

The ADE (Autonomous DEcision Making in very long traverses) project belongs to the same programme as PRO-ACT. During this project various datasets were collected from long trajectories executed by the SherpaTT

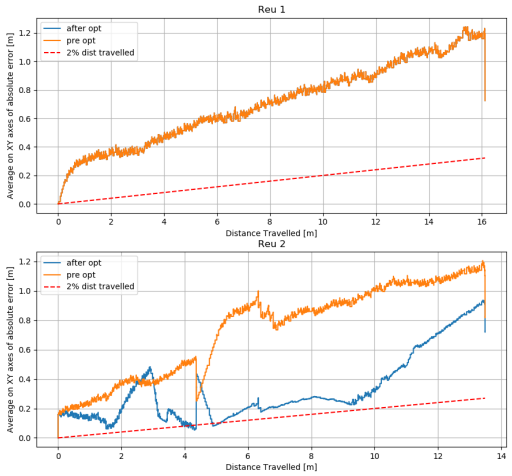

Figure 7. Norm of the localisation error in the PRO-ACT dataset.

rover from DFKI, and Magellium, one of the partners of both ADE and CoRob-X, provided the dataset to further evaluate the performances of CMAP. This data comes from only a single rover, but it is interesting to see how constraints are still found, often from consecutive maps or loop closures in the trajectory. Fig. 8 shows the trajectory and optimised poses in the planar plane, while in Fig. 9 a reduction in the error around the constraints can be seen, both in position and attitude estimation.

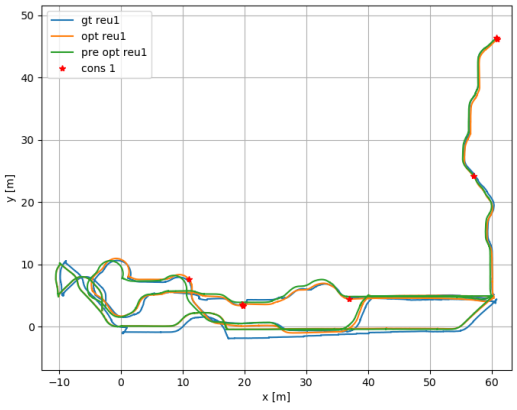

Figure 8. Planar trajectory of the ADE dataset.

3.4. Mapping

The first thing to take into account is how to build a global map using different sensor sources. This is handled by the open-source ROS Elevation Mapping package, giving as inputs the point clouds generated by Lidar and/or

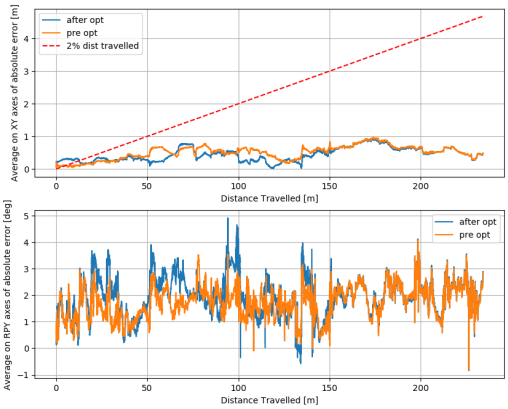

Figure 9. Norm of the localisation error in the ADE dataset.

Cameras. While the performances of a localisation algorithm can be precisely evaluated with various metrics and comparison with the ground-truth, the evaluation of the mapping component is often carried out more qualitatively. For this, the Webots simulation shown in Fig, 10 was mostly used, comparing the global map built over the noisy poses from wheel odometry in Fig. 11, with the one coming from the optimised poses as presented in Fig. 12. It can be noticed how the distortion in the mapping of the box is smaller in the second case, where the new poses are more precise.

Figure 10. Webots overview with the reference box highlighted in the red rectangle

4. CONCLUSIONS

The outcomes obtained both in simulation and with datasets from past project show promising results for an improvement in both localisation and mapping performances. However many challenges are still open and further improvement is possible. The results in the PRO-ACT dataset revealed how manual tuning of various parameters, thresholds, and descriptors for the submap

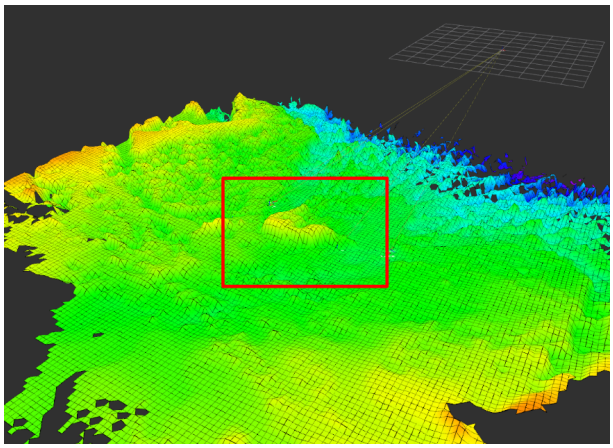

Figure 11. Global Cooperative Map from odometry poses with the reference box highlighted in the red rectangle

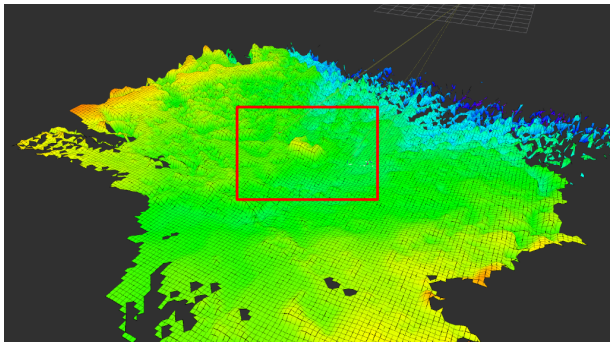

Figure 12. Global Cooperative Map from optimized poses with the reference box highlighted in the red rectangle

matching is necessary to obtain constraints, as for example it was not possible to find any in one of the robots. Encouraging results were obtained in the ADE dataset, showing how this approach can lead to an improved localisation also in a single robot, exploiting constraints within consecutive maps or loop closures. The test were performed with only two robots but it could be easily scaled to larger robotics teams by just running additional submap matching and inter-robot detection components on them. This should not have negative effects on the size of the map, thanks to various checks and measures. The submaps are matched only in a neighborhood of the robot's pose. Additionally, the size of the global map itself does not grow depending on the number of robots but only on the area it covers, since new points are only added if they do not overlap with previous ones, resulting in a bounded resolution. As a further development, the optimized poses could be provided in feedback to the submap matching module, achieving more accurate constraints by starting from already better poses. The CMAP stack will be intensively tested in the months to come as the CoRob-X project enters its final phase, providing a larger variety of scenarios and datasets to test it and further improve it.

REFERENCES

[1] C. Cadena, L. Carlone, H. Carrillo, Y. Latif, D. Scaramuzza, J. Neira, I. Reid, and J.J. Leonard. Past, present, and future of simultaneous localization and mapping: Towards the robust-perception age. *IEEE Transactions on Robotics*, 32(6):1309–1332, 2016.

[2] Silvère Bonnabel. Symmetries in observer design: Review of some recent results and applications to ekf-based slam. In Krzysztof Kozłowski, editor, *Robot Motion and Control 2011*, pages 3–15, London, 2012. Springer London.

[3] Guoquan P. Huang, Anastasios I. Mourikis, and Stergios I. Roumeliotis. On the complexity and consistency of ukf-based slam. In *2009 IEEE International Conference on Robotics and Automation*, pages 4401–4408, 2009.

[4] Giorgio Grisetti, Gian Diego Tipaldi, Cyrill Stachniss, Wolfram Burgard, and Daniele Nardi. Fast and accurate slam with rao–blackwellized particle filters. *Robotics and Autonomous Systems*, 55(1):30–38, 2007. Simultaneous Localisation and Map Building.

[5] Viorela Ila, Josep M. Porta, and Juan Andrade-Cetto. Information-based compact pose slam. *IEEE Transactions on Robotics*, 26(1):78–93, 2010.

[6] F. Lu and E. Milios. Globally consistent range scan alignment for environment mapping. *Autonomous Robots*, 4:333–349, 1997.

[7] Giorgio Grisetti, Rainer Kümmerle, Cyrill Stachniss, and Wolfram Burgard. A tutorial on graph-based slam. *IEEE Transactions on Intelligent Transportation Systems Magazine*, 2:31–43, 12 2010.

[8] Sajad Saeedi, Michael Trentini, Mae Seto, and Howard Li. Multiple-robot simultaneous localization and mapping: A review. *Journal of Field Robotics*, 33(1):3–46, 2016.

[9] Xun S. Zhou and Stergios I. Roumeliotis. Multi-robot slam with unknown initial correspondence: The robot rendezvous case. In *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1785–1792, 2006.

[10] Shi Xingxi, Huang Bo, Wang Tiesheng, and Zhao Chunxia. Cooperative multi-robot localization based on distributed ukf. In *2010 3rd International Conference on Computer Science and Information Technology*, volume 6, pages 590–593, 2010.

[11] Luca Carlone, Miguel Kaouk Ng, Jingjing Du, Basilio Bona, and Marina Indri. Rao-blackwellized particle filters multi robot slam with unknown initial correspondences and limited communication. In *2010 IEEE International Conference on Robotics and Automation*, pages 243–249, 2010.

[12] Sebastian Thrun and Yufeng Liu. Multi-robot slam with sparse extended information filers. In Paolo Dario and Raja Chatila, editors, *Robotics Research. The Eleventh International Symposium*, pages 254–266, Berlin, Heidelberg, 2005. Springer Berlin Heidelberg.

[13] Been Kim, Michael Kaess, Luke Fletcher, John Leonard, Abraham Bachrach, Nicholas Roy, and Seth Teller. Multiple relative pose graphs for robust cooperative mapping. In *2010 IEEE International Conference on Robotics and Automation*, pages 3185–3192, 2010.

[14] Rainer Kümmerle, Giorgio Grisetti, Hauke Strasdat, Kurt Konolige, and Wolfram Burgard. G2o: A general framework for graph optimization. In *2011 IEEE International Conference on Robotics and Automation*, pages 3607–3613, 2011.

[15] Christopher G. Harris and M. J. Stephens. A combined corner and edge detector. In *Alvey Vision Conference*, 1988.

[16] Radu Bogdan Rusu, Nico Blodow, and Michael Beetz. Fast point feature histograms (fpfh) for 3d registration. In *2009 IEEE International Conference on Robotics and Automation*, pages 3212–3217, 2009.

[17] Federico Tombari, Samuele Salti, and Luigi Di Stefano. A combined texture-shape descriptor for enhanced 3d feature matching. In *2011 18th IEEE International Conference on Image Processing*, pages 809–812, 2011.

[18] Radu Bogdan Rusu and Steve Cousins. 3D is here: Point Cloud Library (PCL). In *IEEE International Conference on Robotics and Automation (ICRA)*, Shanghai, China, May 9-13 2011.

[19] Shashank Govindaraj et al. Infuse a comprehensive framework for data fusion in space robotics.

[20] Iyas Dalati Irene Sanz Nieto. Simplistic model-based wheeled robot detection and tracking with full pose estimation.

[21] Yu Xiang, Tanner Schmidt, Venkatraman Narayanan, and Dieter Fox. Posecnn: A convolutional neural network for 6d object pose estimation in cluttered scenes, 2017.

[22] Webots. http://www.cyberbotics.com. Open-source Mobile Robot Simulation Software.

[23] Pro-act project. https://www.h2020-pro-act.eu/.

[24] Shashank Govindaraj, Irene Nieto, Alexandru But, Wiebke Brinkmann, Alexander Dettmann, Leon Danter, Nabil Aouf, Masoud Sotoodeh Bahraini, Abdelhafid Zenati, Heitor Savino, Jakub Stelmachowski, Francisco Colmenero, Enrique Heredia Aguado, Mercedes Alonso, Joe Purnell, Kevin Picton, and Luís Lopes. Multi-robot cooperation for lunar base assembly and construction. 10 2020.