CoMapping: Multi-robot Sharing and Generation of 3D-Maps applied to rural and urban scenarios
Luis Contreras-Samamé, Salvador Domínguez-Quijada, Olivier Kermorgant, Philippe Martinet

To cite this version:

HAL Id: hal-01867743
https://hal.archives-ouvertes.fr/hal-01867743
Submitted on 4 Sep 2018

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L’archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d’enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.
CoMapping: Multi-robot Sharing and Generation of 3D-Maps applied to rural and urban scenarios

Luis F. Contreras-Samamé, Salvador Domínguez-Quijada, Olivier Kermorgant and Philippe Martinet

Abstract—We present an experimental study for the generation of large 3D maps using our CoMapping framework. This framework considers a collaborative approach to efficiently manage, share, and merge maps between vehicles. The main objective of this work is to perform a cooperative mapping for urban and rural environments denied of continuous-GPS service. The study is split in to 2 stages: Pre-Local and Local. In the first stage, each vehicle builds a Pre-Local map of its surroundings in real-time using laser-based measurements, then relocates the map in a global coordinate system using just the low cost GPS data from the first instant of the map construction. In the second stage, vehicles share their pre-local maps, align and merge them in a decentralized way in order to generate more consistent and larger maps, named Local maps. To evaluate performance of all the cooperative system in terms of map alignments, tests are conducted using 3 cars equipped with LiDARs and GPS receiver devices in urban and rural outdoor scenarios of the École Centrale Nantes campus and rural environments.

I. INTRODUCTION

Urban and rural outdoor scenarios can be a real challenge to perform mapping, since environment complexity, such as terrain roughness or lack of structure or dimensions can affect negative the task performance. Other aspects as access to unexplored area, dimension of the region or communication constraints can be considered in the process. In the case of constructing maps for large areas, the idea of using a set of vehicles for building accurate maps in a reasonable amount of time is feasible [1], because a cooperative mapping extends the capability of a single robot by sharing and merging data between group members. When all the data is analysed and merged in a single computation unit, the process is called centralized multi-robot mapping [2]. On the opposite, the goal here is to have each mobile unit build its own map and merge them upon rendezvous, which is a decentralized process [3], [1], [4].

This approach is considered in this article in order to obtain the 3D map of an urban environment. Two vehicles, the ZOE and FLUENCE, work independently during the exploration and define a meeting point to allow direct exchange of data (such as pose, size, limits and maps) during the map reconstruction. The mapping of a rural scenario was also considered, in which was used Lidar-measurement data performed by GOLFCAR robot. The mapping was based on multi-robot approach because the GOLFCAR path was split in two, simulating two robots for the task (see Fig. 1).

Following this direction, we present the development and validation of a Cooperative Mapping framework (CoMapping) applied to outdoor scenarios based on 2 stages, where:

- In the first stage named “Pre-Local Mapping”, each platform constructs its own map by processing range measurements from a 3D LiDAR for a six degrees of freedom (6-DOF) movement and using GPS data (GPS/GGA) only in the beginning for representing the map in robot common frame.
- In the second stage named “Local Mapping”, the robots define which part of their pre-local maps are shared with the other robots based on a Sharing algorithm from the CoMapping framework. Finally the registration process is executed, including an intersecting technique of maps to accelerate the process.

We also propose some indicators to analyze the alignment process in both stages. Our proposal has been tested and validated in real situations. The results include maps developed with data acquired on the surroundings of the ECN (École Centrale Nantes) campus for the urban environment and a farm for the rural experiment, corresponding to the ALFS project.

II. RELATED WORKS

An important topic of the mapping problem is the type of representation used for the environment. Generally 3 types of representations are encountered: feature, grid and topological maps [5]. Our system uses two types of map representation: 3D point cloud (feature) for the pre-processing in Pre-Local Mapping and Octree format (grid) for the exchange of maps in the Local Mapping stage later.

In a situation of collaborative mapping, the registration method from a single robot is really important. Several registration applications use Laser Range-finder sensor to build 3D maps [6] [7]. In that case, a high laser scan rate compared to its tracking can be harmful for this task, since it may lead to distortion in the map construction, in which methods based on Iterative Closest Point (ICP) [8] can be used to match laser returns for different scans. Implementations, with 2-axis and 3-axis lidar and matches geometric structures of a set of local point generated to finally get a point cloud, were presented in [9]. In all those cases, the proposed methods used batch processing to build...
the maps with accuracy, hence are not applicable to real-time map construction. Regarding our Pre-Local Mapping stage, we reconstruct the map of the environment as a point cloud in real-time using a 3-axis lidar by extraction and matching of geometric features in Cartesian space based on a modified version of the registration method from [10]. Then our system uses GPS position data to re-localize the cloud in a global frame.

Several methods have been proposed to merge maps. In [2] and [11] techniques were proposed for 3D merging of occupancy grid maps based on octrees [12]. Their merging process refined the transformation estimate between maps by ICP registration [8]. Specifically, in [11] an ICP version was performed including an efficient technique to exchange maps between robots in order to optimize the bandwidth resources of a multi-robot network for decentralized cases. Those last cases are experimentally studied in this paper.

### A. Merging Indicators

To evaluate the alignment post-ICP, we have first to determine the transformation matrices between frames (see Figure 2). Let us say the correct transformation matrix (provided by Ground-Truth GPS data for instance) is obtained by a matrix product, which is mathematically denoted by \( T \).

\[
T_{\text{metrics}} \implies T_{\text{ICP}} = (T_{\text{ICP}}^2)^{-1} \cdot (T_{\text{ICP}}^1)^{-1} \cdot T_{\text{exact}}^1, \quad (1)
\]

where \( T_{\text{ICP}}^2 \) is the ICP alignment transformation matrix and \( T_{\text{ICP}}^1 \) is the initial transformation estimate (provided by GAA low lost GPS data for instance). \( T_{\text{exact}}^1 \) will represent the transformation matrix used as metric to evaluate the refinement. That matrix \( T_{\text{exact}}^1 \) will be denoted \( T_{\text{metrics}} \).

Very often, alignments are evaluated using Euler representation of 6 parameters. The matrix elements \( \varepsilon_x, \varepsilon_y \) and \( \varepsilon_z \) could be used for the translation evaluation in the \( x, y, z \) axis; and \( \varepsilon_{roll}, \varepsilon_{pitch} \) and \( \varepsilon_{yaw} \) for the orientation. However in order to reduce the complexity of alignment analyse, we propose two indicators: \( \varepsilon_r^* \) and \( \varepsilon_t^* \). We can have a better idea of the orientation evaluation by re-expressing \( T_{\text{metrics}} \) in terms of an Axis-Angle representation \((\theta, u)\) as is shown in (2). In the same way, translation evaluation can be reduced to analyse only the module of the vector with components \( \varepsilon_x, \varepsilon_y \) and \( \varepsilon_z \) (see (3)).

\[
\begin{align*}
T_{\text{metrics}} &= \begin{bmatrix}
R_{3 \times 3} & \varepsilon_x \\
\varepsilon_y & \varepsilon_z \\
0 & 0 & 1
\end{bmatrix}, \quad R_{3 \times 3} \rightarrow (\theta, u) \quad (2)
\end{align*}
\]

\[
\varepsilon_t^* = \sqrt{\varepsilon_x^2 + \varepsilon_y^2 + \varepsilon_z^2}, \quad \varepsilon_r^* = \theta, \quad (3)
\]

**Table I** shows the ICP alignment evaluation considering a known initial transformation. We also made the comparison between metrics before and after ICP. Analogously, in order to obtain indicators for correct ICP refinement, it was determined that values of \( \varepsilon_t^* < 0.25 \) and \( \varepsilon_r^* < 6.14 \) are considered as coherent merging results. In the real experiments where the exact correct transformation is not known, we can use the ICP matrix as a criterion of the error evaluation, in other words: \( T_{\text{metrics}} = T_{\text{ICP}}^1 \). In the following sections, our metrics from (3) will be renamed as \( \varepsilon_t^* \) and \( \varepsilon_r^* \), associated to experiments with unknown maps transformations.

### III. METHODOLOGY

#### A. Pre-Local Mapping Stage

In the Pre-Local Mapping stage, inspired by the work in [11], its versatile configuration does not depend on a specific LidarSLAM method. A modified version of [10] was used as LidarSLAM method for this article, considering its good place in the KITTI ranking [9]. Each mobile platform runs

---

1. LOAM: [https://github.com/laboshinl/loam_velodyne](https://github.com/laboshinl/loam_velodyne)
2. KITTI ranking: [http://www.cvlibs.net/datasets/kitti/eval_odometry.php](http://www.cvlibs.net/datasets/kitti/eval_odometry.php)
this Pre-Local Mapping node. The behaviour of this node is illustrated in Figure 3, where \( \hat{P} \) represents the raw point cloud obtained by a laser scan initially. The accumulated cloud during each sweep \( k \) generates \( P_k \), which is processed by the Lidar-Odometry algorithm at a frequency around 10Hz. The algorithm receives \( P_k \) and computes the lidar motion between two successive sweeps, obtaining the transformation \( T_k \). Deformation in \( P_k \) is corrected using the estimated lidar motion in order to use it at a frequency of 1Hz by the Lidar-Mapping algorithm. This algorithm executes the matching and registration of the non-distorted cloud onto a map. Both algorithms (Lidar-Odometry and Lidar-Mapping) are solved with a non-linear optimization, specifically the Levenberg-Marquardt method [13]. Finally, using the GPS information for identifying the initial vehicle pose, it is possible to coarsely project the map of each robot into a common coordinate frame. This projected cloud is denoted as the Pre-Local map. This Pre-Local mapping node can work with different kind of GPS data, either a Ground-Truth option (DGPS-RTK) or an approach most economical, for example using GPS-GGA type information (Global Positioning System Fix Data).

B. Local Mapping Stage

The architecture of Local Mapping Stage is described in Figure 4, where the process run on a robot referenced as “i”, which shared its map with a robot “n”.

1) Map Sharing Step: Here lies the key of the efficiency of the collaborative framework [11]. Once the Pre-Local Maps are generated, the robots have to exchange their point-clouds to start the map merging task. However in many cases, the sharing and processing of big map data can penalize the team performance with respect to memory usage and execution time. The sharing technique overcomes this problem, since each mobile platform just transfers a certain volume of its Pre-Local map to the other mobile units. Before the transfer of the map Previous the transference, a map compression step is executed in each vehicle, using octree structure from the octomap package [12] as a format of compression, optimizing the car-communication.

An overview of this sharing technique is showed in Figure 5 in which the point clouds \( A \) and \( B \) are the Pre-Local Maps from two different cars “i” and “n” respectively. Both cars receiving simple data about the 3D limits of the maps (i.e. bounding cubic lattice of the point clouds) and the sharing node determines what part of its map will be sent to the other car. These limits are expressed as two vectors: in the first one \( A_{\text{min}} \), the components represent the lowest displacement from the origin along each axis in the point cloud; and in the second one, \( A_{\text{max}} \), is related to the point of highest displacement.

![Fig. 5. Graphical representation of the Map Sharing technique (Top view of plane \( XY \)). \( A_{\text{min}}, A_{\text{max}}, B_{\text{min}}, B_{\text{max}} \) represent the point cloud limits along the \( z \)-axis.](image)

Algorithm 1: Selection of point cloud to share with another robot.

The pseudo-code of the map sharing step is depicted in Algorithm 1. First of all, the functions as \( \text{GetValues()} \) sort in ascending order the array of components along each axis of the vectors \( A_{\text{min}}, A_{\text{max}}, B_{\text{min}}, B_{\text{max}} \) and returns the 2nd and 3rd values from this sorted array, named \( (V'2) \) and \( (V'3) \) respectively. Then for each axis, the mean of \( (V'2) \) and \( (V'3) \) establishes the Cartesian coordinates \((C_x,C_y,C_z)\)
of the geometric center of the exchanged region \((S)\). This region \(S\) is a cube whose edge length is \(2L\) and the points from \(A\) contained in this region are extracted to generate a new point cloud \(A_{sel}\). At each iteration, the cube region is adjusted until the number of points from \(A_{sel}\) is less than a defined threshold (maximum number of points desired to exchange \(Np_{max}\)). Once the condition is accomplished the transfer begins. All the sharing process is analogous on the other robot “n”. Finally, encoding of \(A_{sel}\) and \(B_{sel}\) in octree format is done to reduce the usage of bandwidth resources of the network.

2) Registration step with ICP: The intersecting volumes of \(A_{sel}\) and \(B_{sel}\) are calculated and renamed as \(A_{int}\) and \(B_{int}\), which are down-sampled in order to reduce the time execution of the registration. In the feature descriptors estimation step, the surface normals and curvature of these input clouds are computed in order to improve the feature estimation step, the surface normals and curvature of these applied to source cloud \(B\) for estimating the best transformation to align a source cloud \(B\) with \(A\). The ICP refines an coarse alignment between clouds, \(B\) and \(A\), by iterative minimization of a cost function. Corresponding pairs \((b',a')\) from \(A_{int}\) and \(B_{int}\) are determined iteratively. Least squares registration is executed and the mean squared distance \(E\) is minimized with respect to translation \(t\) and rotation \(R\):

\[
E(R,t) = \frac{1}{Np_{b'}} \sum_{i=1}^{Np_{b'}} \| a'_{i} - (R b'_{i} + t) \|^2,
\]

(4)

where \(Np_{b'}\) is the number of points \(b'\).

The resultant rotation matrix \(R\) and translation vector \(t\) are applied to source cloud \(B_{int}\). Again the matching between points from \(A_{int}\) and \(B_{int}\) is re-computed, until the variation of the mean square error between two consecutive iterations is less than a threshold. The final ICP refinement for \(n\) iterations is obtained by multiplying the individual transformations: \(T_{ICP} = \prod_{j=1}^{n} T_j\). \(T_{ICP}\) is applied to \(B_{sel}\) to align and merge with the original cloud \(A\), generating finally the Local Map \(A_L\). Similarly, this merging process is executed individually in the other mobile unit.

IV. RESULTS

Our experiments considered 3 vehicles: a Renault ZOE, a Renault FLUENCE and a GOLFCAR equipped with a Velodyne VLP-16 3D LiDAR of 360° horizontal and 30° vertical field of view.

A. Urban scenarios: ECN campus case

1) For one robot: First of all, we analysed the impact of GPS quality on the 3D map generation of each vehicle. For that, we took as reference some tests executed with the FLUENCE car in Table II and we performed randomly a path with the robot around the surroundings of the ECN campus. The generated map was twice re-projected on a global frame using two types of GPS data: a high-accuracy DGPS-RTK and standard low cost GPS-GGA. The map with the 1st type of GPS was assumed as referential map, using this DGPS-RTK data as correct transformation matrix \(T_{exact}\). On the other hand, the map re-located with GPS-GGA information was considered as a map with a coarsely placement, used to fill the initial transformation \(T_{j}\). Results of ICP alignment between those 2 maps were addressed and substituted in (1), to obtain our metrics proposed in (3).

<table>
<thead>
<tr>
<th>Robot</th>
<th>(\varepsilon^*_{t})</th>
<th>(\varepsilon^*_{r})</th>
<th>(\varepsilon^*_{t}) [2]</th>
<th>(\varepsilon^*_{r}) [2]</th>
</tr>
</thead>
<tbody>
<tr>
<td>FLUENCE</td>
<td>1.781047</td>
<td>0.003453</td>
<td>3.005938</td>
<td>0.014086</td>
</tr>
</tbody>
</table>

The first two values in Table II corresponds to translation and orientation evaluation. Regarding the analysis performed in Table II, we have coherent alignment results using GGA vs Ground-Truth data because the values comply with the conditions \(\varepsilon_{r}^* \leq 6.14\) m and \(\varepsilon_{t}^* \leq 0.25\) rad. For the moment, we only show our indicator results with indicators proposed by [2], corresponding to the 3rd and 4th values in Table II. According to [2], coherent alignments must be \(\varepsilon_{r}^* \leq 0.9\) and \(\varepsilon_{t}^* \leq 0.21\), so their metrics showed only correct alignments in rotation.

2) For multi-robot team: In that case, our proposed system was validated considering two vehicles: FLUENCE and ZOE car. For both robots, ENU (East-North-Up) coordinate system was considered as external reference of the world frame \(\{W\}\), where \(x\)-axis and \(z\)-axis correspond to the North and East respectively, but coinciding its origin with the GPS coordinates [Longitude: -1.547963; Latitude: 47.250229].

Experiments were realized in urban outdoor environment for an area of approximately 1000m x 700m. For the validation, the robots build clouds from different paths (see Figure 7), running pre-local mapping process in real-time.

![Fig. 6. ZOE (left), FLUENCE (center) and GOLFCAR (right).](image)

![Fig. 7. Paths followed by the FLUENCE (green one) and the ZOE (red one) during experiments. Image source: Google Earth.](image)

Results of the Pre-Local Mapping of this experiment are shown in Figure 8, which also illustrates the “sharing region” determined during the map exchange process in each robot. Pre-Local maps from the FLUENCE and ZOE cars were
projected on the global frame using Ground-Truth and GGA GPS data respectively. In order to simulate constraints on the network bandwidth and memory usage of the robot-team, the maximum number of points exchanged between cars was set to 410000.

In the decentralized case, a meeting point for the robots was defined so they can transfer their maps and perform individually the relative registration process assuming its Pre-Local map as target cloud for alignment reference. Each mobile unit runs an intersecting algorithm, then an ICP refinement obtains an improved transform between maps. Figure 9 depicts the intersection between the shared point clouds during the ICP-alignment process for each robot.

Table III

<table>
<thead>
<tr>
<th>Cars</th>
<th>x</th>
<th>y</th>
<th>z</th>
<th>roll</th>
<th>pitch</th>
<th>yaw</th>
</tr>
</thead>
<tbody>
<tr>
<td>F-Z</td>
<td>0.62208</td>
<td>-1.07885</td>
<td>5.03453</td>
<td>-0.00907</td>
<td>0.00294</td>
<td>-0.00252</td>
</tr>
</tbody>
</table>

Quantitative refinement results are shown in Tables III and IV. ICP transformations were converted in an Euler parametrization \((x, y, z, \text{roll}, \text{pitch}, \text{yaw})\) in meters and radians. Table III corresponds to the refinement process for the FLUENCE car, when it received the map from ZOE, and that map is aligned to the own pre-local map from FLUENCE.

Table IV

<table>
<thead>
<tr>
<th>Car</th>
<th>x</th>
<th>y</th>
<th>z</th>
<th>roll</th>
<th>pitch</th>
<th>yaw</th>
</tr>
</thead>
<tbody>
<tr>
<td>Z-F</td>
<td>0.27346</td>
<td>0.48724</td>
<td>-6.05394</td>
<td>0.00546</td>
<td>0.00091</td>
<td>-0.00145</td>
</tr>
</tbody>
</table>

\(\bar{\varepsilon}_r^* = 0.009814\) and \(\bar{\varepsilon}_t^* = 6.079865\) (Our indicators)
\(\bar{\varepsilon}_r^* = 3.023563\) and \(\bar{\varepsilon}_t^* = 6.814636\) (Jessup’s indicators [2])

We can also see either in Table III or Table IV that our proposed indicators confirm coherent merging results because the values are in the range \(\bar{\varepsilon}_r^* \leq 0.25\) and \(\bar{\varepsilon}_t^* \leq 6.14\) established from Table I. On the other hand, indicators from [2] do not work for this application because their results are not even in the ranges of translation \(\bar{\varepsilon}_t^* \leq 0.9\) or rotation \(\bar{\varepsilon}_r^* \leq 0.21\) established by themselves.

Finally, Figure 10 depicts one of the final merging results, specifically the final 3D Local map from the ZOE projected on a 2D map in order to make qualitative comparisons.

B. Rural scenarios: farm case

For that case, our proposed system was validated using the GOLFCAR robot, but in order to do a collaborative mapping approach, the robot path was split in two for simulating two mobile units (see Figure 11(a)). All data come from the rural outdoor environment in an area of approximately 300m x 150m. The external reference of that world frame was also parallel to ENU coordinate system, with a GPS origin given [Longitude: -1.355357; Latitude: 46.809106].

Initial Pre-Local maps and “sharing region” for each simulated robot are exposed in Figure 11(b). Those maps were projected on the global frame using GGA GPS data in both cases. For this scenario, the maximum number of points was set to 50000. Similar to the urban case, each platform mobile executes the intersecting algorithm and then an ICP alignment (see Figure 12).

Table V

<table>
<thead>
<tr>
<th>Cars</th>
<th>x</th>
<th>y</th>
<th>z</th>
<th>roll</th>
<th>pitch</th>
<th>yaw</th>
</tr>
</thead>
<tbody>
<tr>
<td>G1-G2</td>
<td>0.1461</td>
<td>0.3570</td>
<td>0.5180</td>
<td>-0.01526</td>
<td>-0.04404</td>
<td>0.00158</td>
</tr>
</tbody>
</table>

Regarding the ICP transformation, quantitative alignment results relatives to GOLFCAR1(G1) are shown in Table V. All the ICP transformations are also expressed in Euler.
In this context, experiments also showed the relevance of registration by reducing the number of points for processing. In this context, experiments also showed the relevance of the map sharing algorithm by optimizing the performance of the robot-team by reducing the amount of data transferred on the network. Finally, the CoMapping framework remains an appropriate candidate to exchange and generate efficiently large maps with an approach multi-robot suitable for different kind of environments.

V. Conclusion and Future Work

A cooperative framework for large scale maps applied to urban and rural environments was presented. In this framework, each robot generates its Pre-Local map using 3D-Lidar range measurements and executes individually the merging process considering initially a coarse map alignment with an optimum data exchange. The experimental results highlight the efficiency and versatility of the framework for cooperative mapping with three vehicles in different scenarios. Indicators were also proposed to demonstrate the success of the map merging process for the group of robots. Future work will be addressed to analyze the Sharing step and propose metrics to evaluate the map compression.

Acknowledgment

The authors gratefully acknowledge support of this work by Lionel Génievê and all the members of the ECN and LS2N institutions.

References