2013 IEEE/RSJ International Conference on
Intelligent Robots and Systems (IROS)
November 3-7, 2013. Tokyo, Japan

Multi-Robot SLAM using Condensed Measurements

M.T. Lázaro, L.M. Paz, P. Piniés, J.A. Castellanos and G. Grisetti

Abstract— **In this paper we describe a Simultaneous Localization and Mapping (SLAM) approach specifically designed to address the communication and computational issues that affect multi-robot systems. Our method utilizes condensed measurements to exchange map information between the robots. These measurements can effectively compress relevant portions of a map in a few data. This results in a substantial reduction of both the data to be transmitted and processed, that renders the system more robust and efficient. As documented by our simulated and real world experiments, these advantages come with a very little decrease in accuracy compared to ideal (but not realistic) methods that share the full data among all the robots.**

I. INTRODUCTION

Simultaneous Localization and Mapping (SLAM) is an essential skill for mobile robots that have to execute complex tasks in articulated environments. This problem has been actively investigated by the community for over 20 years, and effective solutions for single-robot systems are nowadays available. In contrast, very few works address the problem of building a map with multiple robots. Within single-robot SLAM, a prominent family of techniques are the so called graph-based algorithms [1], [2], [3], [4], [5]. Solving a graph-based SLAM problem involves to construct a graph whose nodes represent robot poses or landmarks and in which an edge between two nodes encodes a sensor measurement that constrains the connected poses. Once such a graph is constructed by a *front-end* algorithm, the crucial problem is to find a configuration of the nodes that is maximally consistent with the measurements. This involves solving a large error minimization problem which is often done by means of modern least-squares optimization approaches, also called *back-ends* in the SLAM context.

In principle, using multiple robots to acquire the map is more robust, since the failure of a single system does not necessarily compromise the whole result. Furthermore, the parallel acquisition of data by multiple robots might result in less time needed for building the map. Despite these attractive properties, multi-robot systems for SLAM presents substantial challenges of both theoretical and practical nature. Ideally, existing algorithms for single-robot graph-based SLAM could be extended to handle the multi-robot case just by constructing and optimizing the graph based on all measurements gathered by the robots. Unfortunately, such an approach presents several challenges. First, determining constraints between pairs of robots' graphs requires a re-localization scheme without any initial guess. This might dramatically increase the chances of adding wrong edges to the graph, and would compromise the entire process. Second, assuming to have an ideal error-free front-end, the graph obtained by each robot would rapidly increase its size. In the worst case, each robot would add a set of edges to the graph with a quadratic dependency on the number of robots. Consequently it would limit the on-line performance of any state-of-the-art optimizer whose complexity roughly increases with the number of edges.

Furthermore, the above scenario assumes the robot can perfectly communicate with each other, which is typically not the case. Wireless communications in large environments are usually brittle and depend on the positions of the nodes. Furthermore they present bandwidth limitations, that would prevent the robots to share large amounts of data.

In this paper we propose an approach for multi-robot SLAM (MR-SLAM) that addresses the issues raised above. Our method is designed to operate with very limited commu-

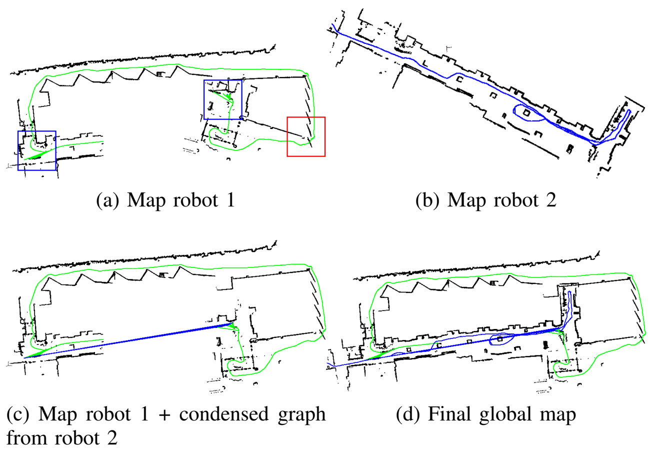

(a) Map robot 1 (b) Map robot 2
(c) Map robot 1 + condensed graph from robot 2 (d) Final global map

Fig. 1: This figure illustrates a motivating example of our approach. Two robots cooperate to construct a map of a building containing a loop of 250m. $a), b)$ Each robot is in charge of mapping one part of the large loop. Due to the lack of enough observations robot 1 commits a big error and fails in the estimation of its part of the loop (a), red square). However, it meets and localizes robot 2 at two points of its trajectory (a), blue squares) who sends a compressed version of its map that contains measurements relating these two locations. $c)$ When robot 1 adds these measurements (blue edges) to its map, it improves its estimation. $d)$ Since the maps become interconnected, we are able to reconstruct the global map by merging the individual maps and optimizing them together.

This work was partially supported by Grupo DGA T04-FSE, the Spanish projects MICINN-FEDER DPI2009-13710 and DPI2012-36070 and by the EU project ROVINA FP7 ICT-2011.8.2 Project Reference 600890.

M.T. Lázaro, L.M. Paz, P. Piniés and J.A. Castellanos are with Instituto de Investigación en Ingeniería de Aragón, Universidad de Zaragoza (Spain) {mtlazaro,linapaz,ppinies,jacaste}@unizar.es

G. Grisetti is with the Department of Computer, Control and Management Engineering, La Sapienza University of Rome (Italy) grisetti@dis.uniroma1.it

nication facilities and allows to dynamically add and remove robots from the system.

Each robot in the team computes its own map, but it refines it by integrating a set of virtual or condensed measurements coming from the other robots. These condensed measurements can be seen as a reduced version of the graph constructed by the other robots, that contains only the information relevant to the receiver in order to refine its own map. This allows to substantially reduce the size of the optimization problem that each member of the team has to solve, thus increasing the efficiency. To localize a robot in any other robot's graph with high presence of outliers, we propose a robust RANSAC-based approach that substantially decreases the chances of wrong data associations and loop closings. Our system has been tested both on real robots and on simulated environments, and it will become available as an open source ROS package. Figure 1 illustrates a motivating example of our approach.

II. RELATED WORK

Most of the algorithms proposed for multi-robot SLAM during the last years have been motivated by the success achieved by their corresponding single-robot SLAM counterparts. It is not surprising then that many of the first distributed implementations [6], [7], [8] were based on the filtering (EKF/EIF) paradigm and inherited some of its drawbacks: overconfident estimates due to linearization errors, quadratic computational cost and difficulty to recover from wrong data association decisions. However, these works already described the main challenges encountered in multi-robot systems: bandwidth limitations, asynchronous communications, coherent information integration and data association between different robots. In the same filtering context we can find more recent works like [9] and [10]. In [9] a Rao-Blackwellized Particle Filter is implemented as estimation kernel that works in simple scenarios with unknown initial correspondences. Each time a pair of robots communicate they have to calculate their relative transformation and interchange all the information gathered since the last meeting. In [10], a distributed and decentralized cooperative SLAM algorithm based on the EKF is presented. In order to not duplicate the information transmitted between the robots the algorithm relays on a complex set of rules and theorems that guarantee a coherent and consistent interchange of information.

Graph-based optimization algorithms have become the most successful techniques to solve the SLAM problem. In [3] we can find the first comprehensive graph-based approach to distributed SLAM with landmarks. The algorithm uses a multifrontal QR factorization in which no measurements are communicated between robots or robots and a server. Instead, the communication is limited to QR update messages, which condense the entire measurement history on the individual robots into small upper trapezoidal matrices. The data association problem is not considered and the measurements are processed off-line for each robot. In [11] the authors present a Collaborative Smoothing and Mapping (C-SAM) algorithm to build a joint map from a team of robots without initial knowledge of their relative positions. Therefore C-SAM does not present a proper distributed SLAM solution but a centralized version of the problem. Only simulated results are provided in the paper. A recursive solution for multi-robot pose graph SLAM is presented in [12]. The main novelties of this approach reside on its incremental nature, i.e. the solution does not depend on a batch optimization after all measurements are taken, and on the introduction of anchor nodes that allow each robot to use its own reference frame whereas inter-robot measurements and graphs obtained from other robots can be easily managed in the same framework. Therefore in each encounter the robots interchange their graphs which do not need to be transformed to a common frame since can be tackled using the anchor nodes. The paper does not take into account any issues of communication bandwidth constraints between robots.

The closest approach to our proposed method is presented in [13] and [14]. The authors address in [13] the multi-robot problem with an extended Smoothing and Mapping approach called Decentralized Data Fusion (DDF) which is represented using a factor graph. Each robot optimizes its own trajectory and its landmark map and then creates a *condensed map* formed exclusively by the marginalization of *common landmarks*. These condensed maps are mutually interchanged among neighboring robots to create a simplified neighborhood graph of landmarks that is optimized by each robot. To correct the local map with the information obtained from the optimization of the neighborhood map a set of hard equality constraints are established between each neighborhood landmark and its corresponding local version. In summary, robots get mutually connected by sending graph nodes of shared features that must be hard-linked with their corresponding local representations. In [14] the work is extended with a novel multi-robot data association method for robust decentralized mapping. The data association is based on a triangulation algorithm that provides matching between maps.

Our multi-robot SLAM system is based on the concept of *condensed measurements* [15]. During map construction, robots meet and exchange data in different parts of the environment. The messages are governed by a protocol explained in detail later in the paper and results in each robot augmenting its pose graph with a measurement about the relative position of the encountered partner. After the first encounter, each time a pair of robots meet they additionally interchange a set of *condensed measurements*, which is just a factor graph of the shared variables obtained from an approximation of their respective global graphs at the equilibrium. The advantages of this approach are three-fold: 1- Each robot only carries its own graph that gets minimally augmented whenever an encounter with other robot of the team takes place; 2- The mutual influence between the team of robots is easily tackled by using the *condensed measurements* since only new virtual factors (edges in the graph) between the shared nodes must be taken into account in the optimization process; Neither special constraints nor different graph representations are required. 3- The communication bandwidth is efficiently used since a summarized (condensed) representation of the required constraints is transmitted between robots. In addition, we propose a technique to robustly find alignments between local maps. This technique is used to find loop closures or alignments between local maps from different robots.

III. SINGLE ROBOT LEAST SQUARES SLAM AND CONDENSED MEASUREMENTS

In a pose-graph based approach, the problem of SLAM is reduced to determine the positions of a robot along its trajectory $\mathbf{x} = (\mathbf{x}_1, ..., \mathbf{x}_n)^T$. This problem can be represented by a graph. Each node of the graph represents a position $\mathbf{x}_i$ of the robot, together with a measurement (image or laser scan) acquired at that position. An edge between two nodes encodes a measurement about the relative transformation of the two connected nodes. These measurements can be computed directly from the odometry or indirectly by computing relative transformations from the observations e.g. by using a visual place recognition system or scan matching. Assuming that the measurements are affected by Gaussian noise, a measurement between $\mathbf{x}_i$ and $\mathbf{x}_j$ is characterized by its mean $\mathbf{z}_{ij}$ and its information matrix $\mathbf{\Omega}_{ij}$.

In other words, in a pose-graph SLAM approach, we assume that our robot is equipped with a simple sensor capable of measuring the transformation between robot locations in the trajectory when they are either temporally or spatially close.

Given a pair of nodes $\mathbf{x}_i, \mathbf{x}_j$ and a measurement $\mathbf{z}_{ij}$ connecting both nodes, it is possible to compute the error committed in the estimation:

$$\mathbf{e}_{ij}(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{z}_{ij} - \hat{\mathbf{z}}_{ij} \tag{1}$$

where $\hat{\mathbf{z}}_{ij} = \mathbf{h}(\mathbf{x}_i, \mathbf{x}_j)$ is the expected measurement given the current configuration of nodes $\mathbf{x}_i, \mathbf{x}_j$. In our case $\mathbf{h}(\mathbf{x}_i, \mathbf{x}_j)$ computes the position and the orientation of $\mathbf{x}_j$ in the frame of $\mathbf{x}_i$.

Let $\mathcal{C} = \{\langle i, j\rangle\}$ be the set of pairs of nodes for which a measurement $\mathbf{z}$ exists. The goal of a maximum likelihood approach is to find the configuration of nodes $\mathbf{x}^*$ which minimizes the overall error:

$$F(\mathbf{x}) = \sum_{\langle i,j\rangle \in \mathcal{C}} \mathbf{e}_{ij}^T \Omega_{ij} \mathbf{e}_{ij} \tag{2}$$

$$\mathbf{x}^* = \arg\min_{\mathbf{x}} F(\mathbf{x}) \tag{3}$$

where Ω_{ij} is the information matrix of measurement $\mathbf{z}_{ij}$. Thus, the SLAM problem is formulated as a nonlinear least squares problem which can be solved by using standard optimization methods like the Gauss-Newton or the Levenberg-Marquardt algorithms.

To solve this problem, modern optimization approaches like $\mathbf{g^2o}$ [1] or SAM [16] require a time that depends on the number of edges, and their success in finding the correct solution is affected by the initial guess available to the system. In the single robot case, this initial guess is typically good, since the robot can rely on an estimate that is constructed incrementally, and that at each point in time contains all the information acquired so far. Conversely, in the multi-robot case it might happen that when two robots meet and want to share their map, the individual estimates are affected by a large error. Furthermore, to carry out the optimization by using one of these approaches, the two robot would have to share their entire graph, which is potentially large.

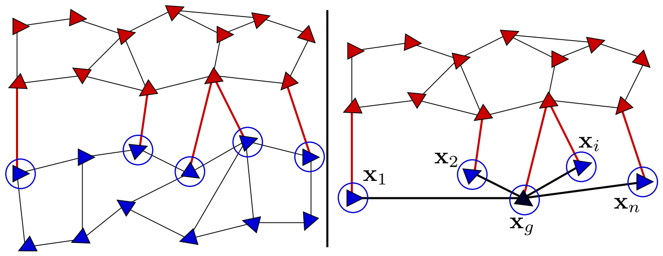


Fig. 2: In this figure, we illustrate the use of condensed measurements to share information between two robots. The graph of Robot A is illustrated in red and the graph of Robot B is illustrated in blue. Red edges show the measurements between nodes of the Robot A's graph and the Robot B's graph. Instead of sending to Robot A all its graph, Robot B sends a condensed version, consisting of a central node (gauge, $\mathbf{x}_g$), and a set of condensed factors connecting the gauge with each of the nodes ($\mathbf{x}_i, i = 1...n$) seen from Robot A. Notice that $\mathbf{x}_g$ can also be selected from the nodes already seen by Robot A.

To lessen this problem, in this paper we use an alternative approach based on condensed measurements [15]. When two robots A and B meet, they share a reduced graph so that each robot receives from the other only the information needed to refine its own estimate. In this section, let us assume that robot A knows which nodes of the other robot B's graph he observed. In order to optimize its own graph, by taking into account the information from B, robot A should know how these shared nodes are related in the space. This information is clearly contained in the graph of B, but it is too large to be sent over the network. Instead of sending the full graph, B sends a "condensed" version that has substantially less nodes, but that captures the information necessary to A to perform this optimization. To this end, we regard the graph of B as a local map. In Figure 2 we illustrate the process.

Once we have a minimal error configuration for the graph of Robot B, we can compute the condensed factors. To this end we need to select an arbitrary node $\mathbf{x}_g$ in the graph. We then need to compute a measurement between $\mathbf{x}_g$ and each other node $\mathbf{x}_i$ which is seen by Robot A. This means that we need to compute a "measurement" between $\mathbf{x}_i$ and $\mathbf{x}_g$, that incorporates the knowledge in the original graph of Robot B that $\mathbf{x}_g$ has about the position of $\mathbf{x}_i$. This is done by projecting the marginal covariance of $\mathbf{x}_i$, with respect to $\mathbf{x}_g$ through the measurement function $\mathbf{h}(\mathbf{x}_g, \mathbf{x}_i) = \mathbf{x}_i \ominus \mathbf{x}_g$. For more details we refer the reader to [15].

IV. MULTI-ROBOT SLAM USING CONDENSED GRAPHS

In this section we will describe in detail our multi robot SLAM system. Our approach operates on raw sensor measurements acquired by mobile robots equipped with a laser scanner. Inter-robot communication is based on a wireless ad-hoc network that dynamically adapts depending on the mutual locations of the robots. Section IV-A presents the details of our communication model.

Each robot executes a standard laser-based SLAM pipeline: the state of the system is stored in a pose-graph which is constantly optimized by the $\mathbf{g^2o}$ optimizer. When the robot moves for a certain distance, a new node is added to the graph, and the odometry measurement is used to label the edge between the new and the previous robot positions. The laser scan acquired at the new position is matched against a set of candidate scans stored in the nodes of the graph. The candidate nodes are selected if the current robot position falls in their uncertainty ellipses. This gives a set of candidate loop closing edges between non temporally subsequent nodes, that are inserted in the graph upon validation by a RANSAC based procedure described in Section IV-C.

To extend this single-robot SLAM algorithm to the multi-robot case, we need to augment the graph-construction method described above to handle information coming from other robots. The multi-robot front-end will be in charge of:

- robustly localizing other robots into the current robot's map, based on their raw sensor measurements.
- integrating the condensed measurements of the other robots in the current graph. This is achieved simply by including the set of condensed edges.

In the remainder of this section we describe in detail our communication model and how we address the problems outlined above, by taking into account the limitations of the communication infrastructure.

A. Communication Model

We assume that no infrastructure is present. Thus the communication between robots is point-to-point. Robots can communicate only when they are within a certain distance, and the communication graph changes dynamically based on the current configuration of our multi-robot system.

This models conservatively the behaviour of wireless ad-hoc networks. Wireless communication has a limited range and bandwidth which will vary depending on the protocol (WiFi, Bluetooth, ...), the IEEE standard used (e.g. 802.11b/g/n...) and also the structure of the environment. Not relying in infrastructure has substantial practical advantages.

The communication model proposed works in a robot-independent way, where the messages are transmitted asynchronously and contain the most up-to-date information available. The probability that a message sent is correctly delivered decreases with its size. To maximize the probability that the messages are correctly delivered, in our algorithm we kept the size of the single messages as small as possible, possibly fitting within an Ethernet frame (1400 bytes). Each robot periodically sends a ping and, based on the ping messages received by the other robots, it determines its neighbors. When two robots are within communication range they send two kind of messages: to transmit their local maps and to manage the graphs.

The local map is transmitted through a message containing the following information:

- The last measurement (laser scan) acquired, and the current id of the node containing the laser scan in the graph.
- The up-to-date estimated locations of the last N nodes.

With this information each robot is able to reconstruct the local maps of the team mates in range. Notice that a robot sends only the most recent laser scan, which is the bulky part of the message. To determine a local map consisting of N scans we need to buffer the last N messages from each sender, and render the scans according to the most recent list of estimates of the nodes. The latter is transmitted each time a new node is added to the graph. This allows to update the local maps with minimal communication overhead, even if the graph changes its configuration. The local maps of other robots are used to localize them in the current robot's map. This is done by using a robust RANSAC-based outlier rejection scheme in combination with a correlative scan matching algorithm, which is described in detail in Section IV-C.

To manage the graph, a robot sends the following types of messages:

- A list of nodes of the other robot's local map it has matched against its own local map.
- A condensed graph extracted by its own graph and containing the edges among the nodes that have been matched by some other robot.

These messages are sent whenever a new node is added to the graph, based on the number of mates in range.

B. Multi Robot SLAM

In this section, we illustrate how these messages are used to implement our multi-robot SLAM approach. To simplify the description, we refer to Figure 3 and without loss of generality we assume having only two robots: A (red) and B (blue).

Initially (see Figure 3a), each robot constructs its own map with a single-robot SLAM algorithm. When a communication is available A starts receiving the current local map of B, by storing its most recent readings. A matching procedure is executed to align the two local maps, and results in a set of candidate edges connecting the map of A and the map of B (see Figure 3b). When reasonably confident about the correctness of these edges, robot A sends to B this list (see Figure 3c). B then computes a condensed graph containing only the nodes of its map that appear in the candidate edges found by A (see Figure 3d), and sends it to A. Finally A, includes these measurements in its own graph to get a more consistent map (see Figure 3e).

This algorithm can be implemented within a robot A in a straightforward way by maintaining the following data structures:

- the graph $\mathcal{G}_A$ obtained by single-robot SLAM
- for each other robot B:
  - the most recent local map $\mathcal{M}_B$ consisting of the last N nodes, that is used for cross-localization.

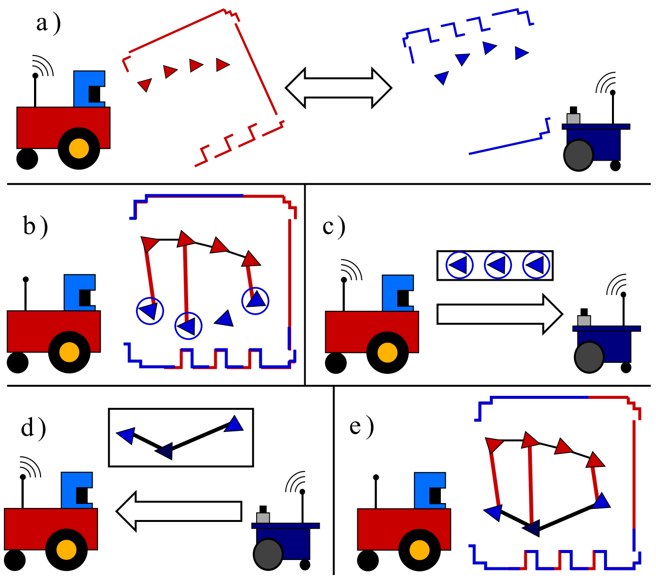


Fig. 3: Illustration of our Multi-Robot SLAM algorithm in a two robots scenario. Robot A is depicted in red and Robot B in blue. Triangles represent the nodes of the graph. *a)* Each robot runs a graph-based SLAM algorithm and constructs its own map. When they are within a communication range, they share their current local maps; *b)* A localizes B and determines a set of candidate edges connecting the two maps; *c)* A informs to B which of its nodes it has matched; *d)* B computes condensed measurements that connect the nodes in its own map that appear in the edges found by A; *d)* A includes these edges in its own graph.

- the list $\mathcal{E}_A^B$ of candidate edges between the map of A and the map of B that have been found by A.
- the list of $\mathcal{E}_B^A$ edges received from B, that connect the map of A and the map of B and that have been found by B.
- the condensed graph $\mathcal{G}_A^B$ sent by B.

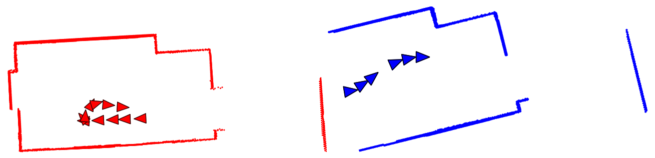
Robot A updates the local maps of each other robot $\mathcal{M}_B$ and the list of edges $\mathcal{E}_B^A$ whenever a new message is received. Each time the single-robot SLAM algorithm running on A adds a new node to the graph, the estimate of the last N nodes and the last laser scan are sent to allow the other robots to construct the local map of A. Subsequently, A runs a map-alignment algorithm between its local map and each $\mathcal{M}_B$, and updates the list of candidate edges by using the procedure described in the next section.

Finally, by knowing $\mathcal{E}_B^A$ Robot A computes which nodes of its own map are relevant for Robot B, and sends the corresponding condensed measurements. In computing the condensed measurements Robot A considers only the portion of the graph acquired with its own sensors, thus avoiding multiple integration of information.

C. Robust Map Alignment

In this section, we describe our approach to robustly align two local maps $\mathcal{M}_A$ and $\mathcal{M}_B$ onto each other. A local map consists of a portion of the graph. We recall that each node consists of a robot pose and a laser scan acquired at that pose. Figure 4 illustrates the problem.

Our goal is to find a set of edges between the nodes of the two local maps such that they are maximally consistent, given the scans. To this end we match each scan $\mathbf{s}_i^B$ of $\mathcal{M}_B$ with each scan $\mathbf{s}_j^A$ of $\mathcal{M}_A$, by using a correlative scan matcher. Note that each matching can result in zero or more solutions. Each of these solutions is then converted in an edge between $\mathbf{s}_i^B$ and $\mathbf{s}_j^A$, and added to a pool of candidate edges.


(a) Local map $\mathcal{M}_A$. (b) Local map $\mathcal{M}_B$.

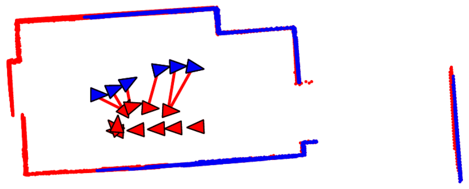

(c) Alignment of the two local maps.

Fig. 4: Example of map alignment between two local maps after finding a set of edges jointly consistent.

Given this pool of edges, we run a RANSAC based procedure to determine which of them are inliers. The idea is the following: to determine a translation between the two local maps it is sufficient to translate them so that one candidate edge e_i is satisfied (its error is $\mathbf{0}$). Applying this translation affects the error of all other candidate edges, and their error will be small if they are consistent with e_i, while it will be large otherwise. Based on these errors we determine inliers and the outliers and we decide whether to accept a match or not. Figure 5 illustrates the procedure.

The bottleneck of this schema is the scan matching routine, since the RANSAC requires typically very few iterations to provide a consistent solution. Accordingly, we need to limit the number of times we perform scan-matching. By considering that the local maps can be assumed to be consistent, and that one of the two local maps is acquired incrementally one scan at a time, we can implement the above procedure in an efficient way. Each time we receive a new scan $\mathbf{s}_i^B$, we match it against the local map constructed by the union of all $\mathbf{s}_j^A$. The scan matcher results in a set of transformations between $\mathbf{s}_i^B$ and the map $\mathcal{M}_A$. These transformations are converted in edges between $\mathbf{s}_i^B$ and the closest node in $\mathcal{M}_A$, after applying the transformation. The resulting edges are inserted in a pool. The RANSAC validation is done at every step, and the candidate edges that are marked as outliers for a certain number of times are removed from the pool.

V. EXPERIMENTS

The multi-robot SLAM approach proposed in this paper has been validated through simulations and real world experiments. Our system is implemented in C++ as a ROS package, and the simulations have been conducted with the Stage simulator.

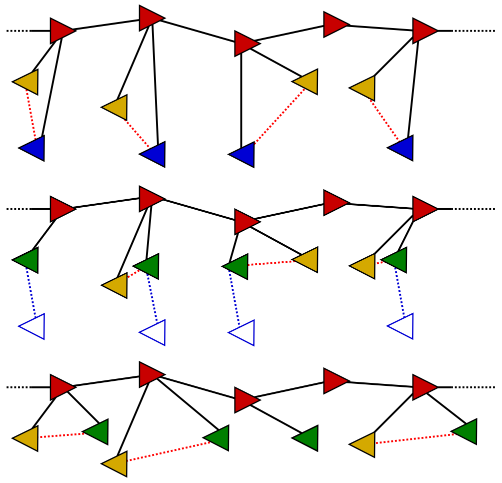
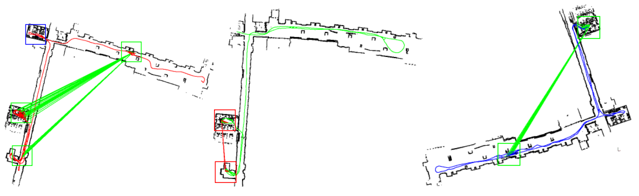


Fig. 5: Top: In red, nodes belonging to a local map, in blue, current estimation of the received nodes and in yellow, the same nodes with respect to the candidate closure edges. Dashed red lines represent the error in the estimation for each edge. Middle: Green, position of the nodes after applying the transformation (blue dashed line) that makes the error of the first node equal to zero. With this configuration, the error in the second and fourth nodes is small (they could be selected as inliers if the error is lower than a threshold) whereas the error in the third node is large (outlier). Bottom: Configuration of the nodes if the transformation to make the error of the third node equal to zero is applied. Since it is a wrong closure, the error in the rest of nodes is large, they are selected as outliers and this configuration of nodes is rejected. Notice that this procedure can be used whether the local maps are from different robots or from the same robot trying to compute loop closing edges.

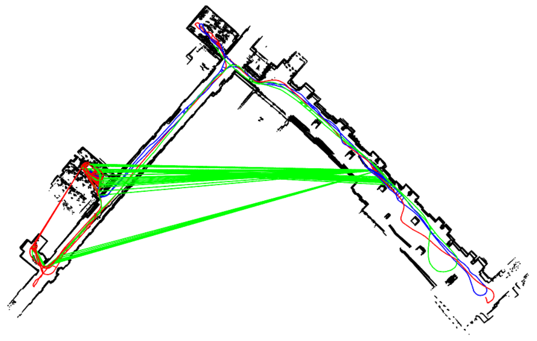
A. Real World

We conducted a real world experiment by using three Pioneer 3-AT robots, equipped with SICK laser rangefinders. The robots were simultaneously controlled by three persons that steered them manually in the environment shown in Figure 6. The robots communicated through an ad-hoc network by sending UDP packets and each of them was running the algorithm described in this paper. We previously synchronized the clocks of all robots with NTP. To be able to reproduce the experiment, we recorded a dataset containing the own measurements each robot logged its own measurements (odometry and laser), and the ping received by other robots. This allow us to reproduce off-line the connectivity of the communication network, and repeat the experiment off-board.

The results of this experiment are shown in Figure 6. The individual maps obtained by each robot together with the condensed graphs received from other robots are depicted in Figures 6a, 6b and 6c. During their navigation, each robot was able to meet and localize some other robot into its own map. The meeting points are depicted with squares in the individual views. These intra-robot localizations make that all maps become interconnected which allows us to reconstruct the global map shown in Figure 6d. In addition to the experiment described here, we executed additional tests with Erratic robots equipped with an Hokuyo UTM laser rangefinder with two robots. The result after merging the individual maps is shown in Figure 7.



(a) Map robot 1 (b) Map robot 2 (c) Map robot 3



(d) Global map

Fig. 6: Experiment at the Ada Byron building of the University of Zaragoza. $a), b), c)$ show the individual maps obtained by each robot, depicted in red, green and blue respectively. The condensed graphs received from other robots are depicted in the colour of the sender robot. $d)$ shows the global map after all individual maps are merged and jointly optimized.

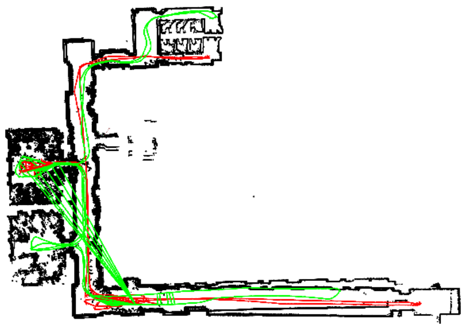
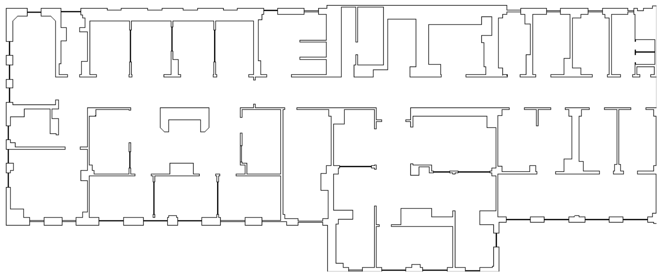


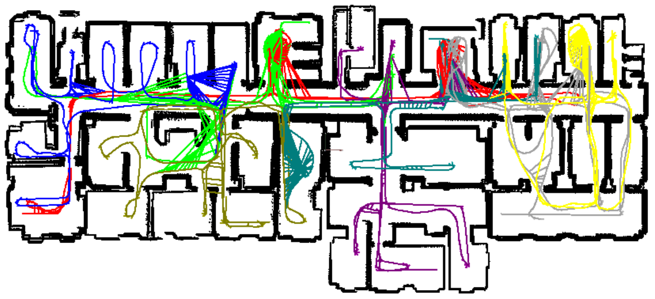
Fig. 7: Experiment at the DIS building of La Sapienza University of Rome. The misalignment observed in the bottom right corner originates from the fact that the robots never meet in that region, thus they are unable to determine constraints between that part of their trajectories. This can be recovered when the two robots meet in that region, or in a post-processing phase.

B. Simulation results

We quantitatively evaluated the performance of our system through simulation experiments. In particular, we measure how the proposed multi-robot system performs in terms of optimization time, bytes transmitted by each robot and accuracy with respect an ideal implementation in which the robots share their whole graph instead of the condensed version. Additionally, we want to analyze how these aspects scale with the number of robots and therefore we tested our approach with 2, 4 and 8 robots. The simulation environment is shown in Figure 8a. We designed trajectories such that each robot met at least once with another robot. As an example, the trajectories and final map obtained in the 8 robots simulation are shown in Figure 8b.



(a)



(b)

Fig. 8: *a*) Simulation environment. *b*) Trajectories and final map in a 8 robots experiment.

Figure 9 shows the results for the optimization times and communication overload obtained in the simulations. Clearly, the more robots are used for mapping the less time is needed to cover the entire environment and the smaller will be the map of each robot. Figures 9a, 9b and 9c show the optimization times for both approaches. It can be seen how, in the condensed approach (green), the optimization times increase linearly as the map grows. Receiving a condensed graph implies adding a few edges to their graphs and this does not affect substantially the computation. In the ideal implementation (red), times grow also linearly with the number of edges. However, this number has a substantial increment when the robots meet and receive the whole graph from the others. This happens, for example at time 350 in the two robots simulation.

The communication overload is shown in Figures 9d, 9e and 9f. As explained in section IV-A, two kind of messages are sent, one containing the local map and another one to send the condensed graph. Transmitting the local map has a constant size if the number of nodes to send is fixed. In our implementation, we transmit both the updated estimates and ids of the last 5 nodes plus the last laser scan obtaining a message of constant size of 1580 bytes. Since this value is the same for both condensed and ideal approaches, this type of message is not taken into account in the results. However, as it can be seen in the figures, the size of the messages to send the graph in the ideal approach differs substantially from the messages in the condensed approach, where the size of the messages stays below 1000 bytes in most of cases. The size of the message that a robot has to send in the condensed approach will grow with the number of nodes of its own map another robot has matched.

By using the ground truth of the simulation, we compared the accuracy of our multi-robot SLAM approach with the ideal implementation. We created a ground-truth graph by extracting a set of virtual edges between neighboring nodes, by using the approach described in [17]. Table I shows the overall mean Chi2 error per edge for each one of the simulations. The number of edges of each individual map varies with the simulation, and from one robot to another. For this reason we use the mean error per edge as a measure of accuracy for both approaches. As it can be seen in Table I the mean errors are very similar and therefore, we can conclude that the accuracy is not sacrificed when sharing the condensed graphs instead of the whole version. This result is confirmed by the visual inspection of the maps.

	Accuracy	
	Condensed Graphs	Ideal
2 robots	1.404	1.442
4 robots	1.572	1.548
8 robots	1.884	1.899

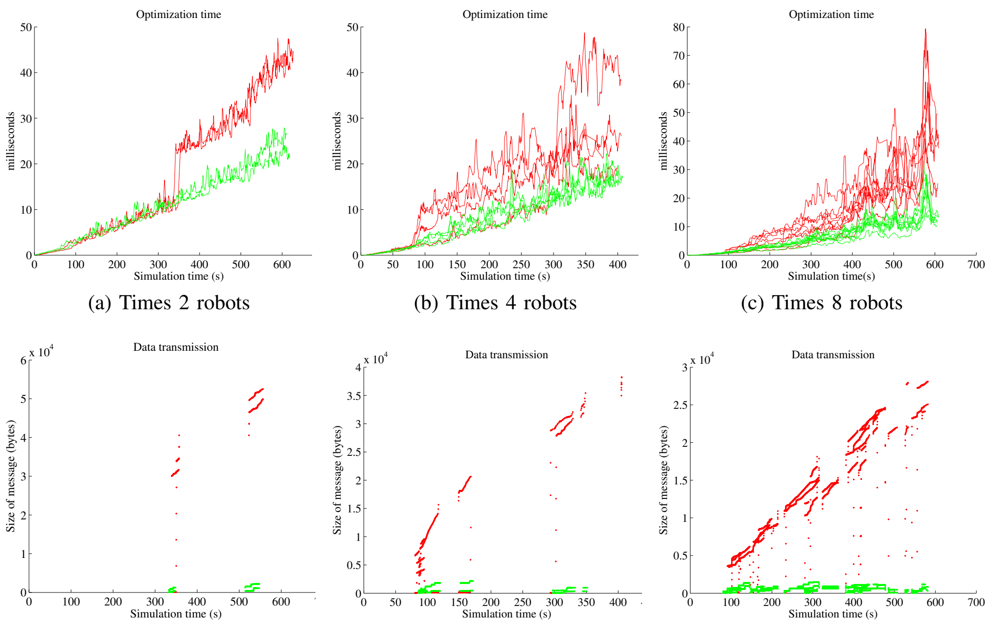
TABLE I: Comparison of the accuracy obtained by our condensed measurement multi-robot SLAM approach and the ideal implementation. The numbers are the χ^2 error of the edges in the ground-truth graph, evaluated with nodes placed as reported by the algorithm.

C. Post Processing

The procedures described above are the core of our multi-robot SLAM. Compared with a centralized approach that has access to all information of all robots, our system leads to a higher error in positions where the robots do not meet. This arises from the fact that robots only share local maps around their current position, thus they cannot relocalize. This is visible in the right hand side of Figure 7. Solving this problem would require transmitting substantial more information, since the robots would have to share all the measurements. Despite this limitation, our schema produces solutions that are sufficient for the robots to navigate. In a subsequent processing stage, a global accurate map can be obtained by merging the solutions of all robots and optimizing them including the condensed measurements. This aligns the map in a global frame. This map can be further improved by adding a set of constraints by matching scans between neighboring nodes. Due to the good initial guess obtained by the map alignment, this step is relatively straightforward, leading to results illustrated in Figure 10.

VI. CONCLUSIONS

In this paper we proposed an approach for multi-robot SLAM that specifically addresses the limitations in network and computation affecting multi-robot systems. We have shown that our method allows to obtain consistent estimates by adding a relatively limited complexity to the traditional single-robot SLAM methods. This results in an overall increase of robustness and computational efficiency with respect to naive MR-SLAM implementations.



(a) Times 2 robots (b) Times 4 robots (c) Times 8 robots

(d) Bytes 2 robots (e) Bytes 4 robots (f) Bytes 8 robots

Fig. 9: Timings for the optimization of the graph and bytes transmitted by each robot. The results are shown in green for the condensed graph approach and in red for the ideal implementation.

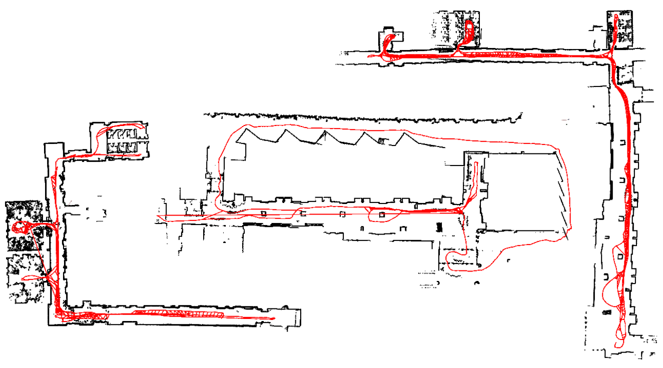


Fig. 10: Results of the three real world experiments performed for this paper, using a straightforward centralized processing of the joint estimates obtained by our MR-SLAM method.

References

[1] R. Kümmerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard, "g2o: A general framework for graph optimization," in *2011 IEEE Int. Conf. on Robotics and Automation (ICRA)*, Shangai, China, 2011, pp. 3607–3613.

[2] G. Grisetti, R. Kümmerle, C. Stachniss, and W. Burgard, "A Tutorial on Graph-Based SLAM," *IEEE Intell. Transport. Syst. Mag.*, pp. 31–43, 2010.

[3] F. Dellaert, A. Kipp, and P. Krauthausen, "A Multifrontal QR Factorization Approach to Distributed Inference applied to Multi-robot Localization and Mapping," in *AAAI National Conf. on Articial Intelligence*, 2005.

[4] E. Olson, J. Leonard, and S. Teller, "Fast Iterative Optimization of Pose Graphs with Poor Initial Estimates," in *Proceedings of the IEEE Inter. Conf. on Robotics and Automation (ICRA)*, 2006, pp. 2262–2269.

[5] U. Frese, P. Larsson, and T. Duckett, "A Multilevel Relaxation Algorithm for Simultaneous Localisation and Mapping," *IEEE Transactions on Robotics*, vol. 21, no. 2, pp. 1–12, 2005.

[6] S. Grime and H. Durrant-Whyte, "Data fusion in decentralized sensor networks," in *Control Engineering Practice*, vol. 2, no. 1, 1994, pp. 849–863.

[7] E. Nettleton, S. Thrun, H. Durrant-Whyte, and S. Sukkarieh, "Decentralised SLAM with low-bandwidth communication for teams of vehicles," in *Proc. Int. Conf. on Field and Service Robotics*, Lake Yamanaka, Japan, 2003, pp. 179–188.

[8] S. Roumeliotis and G. Bekey, "Distributed Multi-Robot Localization," *IEEE Transactions on Robotics and Automation*, vol. 18, no. 5, pp. 781–795, 2002.

[9] L. Carlone, M. Ng, J. Du, B. Bona, and M. Indri, "Rao-Blackwellized Particle Filters multi robot SLAM with unknown initial correspondences and limited communication," in *2010 IEEE Int. Conf. on Robotics and Automation (ICRA)*, Anchorage, Alaska, 2010, pp. 243–249.

[10] K. Leung, T. Barfoot, and H. Liu, "Distributed and decentralized cooperative simultaneous localization and mapping for dynamic and sparse robot networks." in *2011 IEEE Int. Conf. on Robotics and Automation (ICRA)*, Shangai, China, 2011, pp. 3841–3847.

[11] L. Andersson and J. Nygards, "C-SAM: Multi-Robot SLAM using square root information smoothing," in *2008 IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2008, pp. 2798–2805.

[12] B. Kim, M. Kaess, L. Fletcher, J. Leonard, A. Bachrach, N. Roy, and S. Teller, "Multiple relative pose graphs for robust cooperative mapping," in *2010 IEEE Int. Conf. on Robotics and Automation (ICRA)*, Anchorage, Alaska, 2010, pp. 3185–3192.

[13] A. Cunningham, M. Paluri, and F. Dellaert, "DDF-SAM: Fully Distributed SLAM using Constrained Factor Graphs," in *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, Taipei, Taiwan, 2010, pp. 3025–3030.

[14] A. Cunningham, K. Wurm, W. Burgard, and F. Dellaert, "Fully Distributed Scalable Smoothing and Mapping with Robust Multirobot Data Association," in *2012 IEEE Int. Conf. on Robotics and Automation (ICRA)*, St. Paul, Minnesota, USA, 2012, pp. 1093–1100.

[15] G. Grisetti, R. Kümmerle, and K. Ni, "Robust Optimization of Factor Graphs by using Condensed Measurements," in *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, Vilamoura, Portugal, 2012, pp. 581–588.

[16] M. Kaess, A. Ranganathan, and F. Dellaert, "iSAM: Incremental smoothing and mapping," *Robotics, IEEE Transactions on*, vol. 24, no. 6, pp. 1365–1378.

[17] W. Burgard, C. Stachniss, G. Grisetti, B. Steder, R. Kümmerle, C. Dornhege, M. Ruhnke, A. Kleiner, and J. D. Tardós, "A Comparison of SLAM Algorithms Based on a Graph of Relations," in *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2009.