Online Update of Communication Maps for Exploring Multirobot Systems under Connectivity Constraints

Francesco Amigoni[1], Jacopo Banfi[2], Nicola Basilico[2], Ioannis Rekleitis[3], and Alberto Quattrini Li[4]

[1] Politecnico di Milano, 20133 Milano, Italy,
francesco.amigoni@polimi.it
[2] Università degli Studi di Milano, 20122 Milano, Italy
{jacopo.banfi,nicola.basilico}@unimi.it
[3] University of South Carolina, Columbia, SC 29208, USA
yiannisr@cse.sc.edu
[4] Dartmouth College, Hanover, NH 03755, USA
alberto.quattrini.li@dartmouth.edu

Abstract. Multirobot systems for exploring initially unknown environments are often subject to communication constraints, due to the limited range of their transmission devices and to mission requirements. In order to make decisions about where the robots should move, a communication map that encodes knowledge of the locations from which communication is possible is usually employed. Typically, simple line of sight or circle communication models (that are rather independent of the specific environment in which the exploration is carried out) are considered. In this paper, we make a step forward and present a multirobot system that learns and updates a communication map *during* the exploration mission. In particular, we propose methods to incrementally update vertices, corresponding to the locations visited by robots, and edges, corresponding to communication links, of a graph according to the measured power of radio-frequency signals and to the predictions made by a model based on Gaussian Processes. Experimental results obtained in simulation show that the proposed methods build and update rich communication maps specific for the environments being visited and that the availability of these maps can improve the exploration performance.

Keywords: multirobot systems; exploration; communication.

1 Introduction

Exploration of initially unknown environments – namely the incremental discovery of their physical features – is a task involved in several applications, including map building, search, monitoring, and patrolling. Use of multirobot systems for this purpose is particularly challenging in the presence of constraints on communication [5,10,12,13,15,20,23]. Such constraints are due to limited ranges of the

Presented at the Distributed Autonomous Robotic Systems (DARS) 2018.
DOI: 10.1007/978-3-030-05816-6_36

robots' communication devices and to mission requirements, like guaranteeing a continuous or recurrent connection with a base station [1].

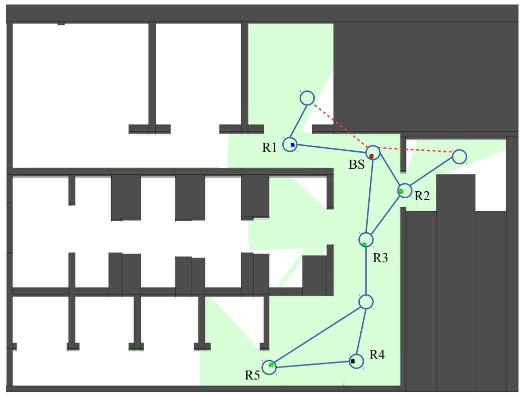

Fig. 1. Representation of the constrained exploration problem, where robots (R1-R5) need to be connected, either directly or in multihop, to the base station (BS). The explored area (green) together with the communication graph built by considering a conservative range line-of-sight communication model (blue) are illustrated. The red dashed lines indicate links that can be added with the approach proposed in this paper.

Multirobot systems for communication-constrained exploration make use of some form of knowledge about the locations from which communication is possible. This knowledge, encoded in a *communication map*, is used to decide where to move the robots in order to efficiently discover the unknown parts of the environment while satisfying the constraints. Usually, communication maps are built using simple rules that only partially account for the actual environment in which robots operate. For example, a communication is considered possible between two locations if they are visible from each other and closer than a threshold distance [5, 10, 21, 23]. In this way, communication between two close locations separated by a thin wall is not considered possible although it can be very likely in practice. Recently, a method that exploits Gaussian Processes for building more rich and reliable communication maps from measurements collected by robots has been proposed [4,17], but this approach considers the construction of communication maps as a stand-alone task not connected to exploration.

In this paper, we present a multirobot system for exploring initially unknown environments under communication constraints. Originally, the proposed system updates the communication map *during* the mission. In particular, we start from the multirobot system of [5] that explores environments under recurrent connectivity constraints (namely, robots must be connected to a base station when they acquire new information) and we enrich it with a communication map represented as a communication graph. This graph is incrementally updated, as the exploration progresses, with the vertices corresponding to the locations visited by the robots and with the edges generated according to the measured power of radio-frequency signals and to the prediction of possible communication links retrieved from the model of [4]. Fig. 1 shows an instance of such a problem, where online updates of the communication graph can allow the robots to be more efficient in exploring the environment. Indeed, with an enriched communication graph, the system can exploit the possibility of communicating directly between the base station and two locations (through the two dashed red links), without setting up any relay chain for multihop communication. Experimental results obtained in simulation show that our proposed methods maintain an updated communication graph that can be used to improve exploration performance.

The main original contribution of this paper is thus the introduction of a multirobot system for exploration that is able, at the same time, to learn and update a graph representing pairs of locations from where communication between robots is possible. Such a contribution allows a multirobot exploration system to be more effective and efficient in exploring environments and at the same time to keep a good situational awareness.

This paper is organized as follows. The next section reviews the relevant related work. Section 3 illustrates the proposed multirobot system for exploration and the methods it uses to build and update communication maps. Section 4 reports experimental results obtained in simulation. Finally, Section 5 concludes the paper.

2 Related Work

Constraints on communication for exploring multirobot systems are usually imposed by the limited communication range of robots and by mission requirements. Such requirements can impose that robots are connected at all times – see, e.g., [12–14, 20] – or recurrently – such as in [5, 10, 15, 16, 23]. A common trait of exploring multirobot systems that consider such constraints is the availability of some form of knowledge about where the robots can communicate. This knowledge – typically used for deciding where to move the robots – can be represented as a communication map: given two locations p and q in the environment, it returns whether a communication is possible between p and q.

Communication maps are often built (although not always explicitly stored) according to simple fixed communication models that tend to be conservative in establishing if a communication between two locations is possible. For instance, some papers, like [5, 10, 21, 23], consider a *line of sight* communication model in which two robots can communicate with each other only if the line segment connecting their locations is entirely contained in the known free space and it is not longer than a threshold value d. This knowledge is used to plan rendezvous between robots [21] or to asynchronously assign locations to robots [23]. Also, [10] uses a communication map built on a line of sight communication model to let robots regain connection with the base station after a fixed time interval.

A similar approach builds communication maps considering that a robot r_1 can communicate with a robot r_2 if r_2 is located within a circle centered in the current location of r_1 and with radius d (the communication range threshold), independently of the presence of obstacles. Papers that follow this approach include [5, 10, 13–16, 21]. Communication maps derived from the *circle* model are used to place explorer and relay robots in [15, 16] and to build a tree in which exploring robots are the leaves and link stations are the inner nodes in [13, 14].

A more complex and realistic way to build communication maps is to consider two robots able to communicate if the estimated power of the radio-frequency signal between their locations is large enough. For instance, an estimate of the signal power P (in dBm) at a distance d from a source can be calculated using the empirical formula of [3]:

$$P = P_0 - 10 \cdot N \cdot \log_{10}(d/d_0) - \min\{nW, C\} \cdot \mathit{WAF}, \tag{1}$$

where P_0 denotes the reference signal value at distance d_0, N is the ratio of power loss with the distance, nW is the number of walls traversed by the signal, C is the maximum number of walls, and WAF is the wall attenuation factor. Typically, it is assumed that two robots are able to communicate if $P > -93$ dBm. The validity of this empirical model is corroborated by works in which Eq. (1) is used to estimate signal power in order to assign robots to locations where they can communicate with the base station, and to arrange rendezvous between robots [21, 26].

The above systems consider communication maps built according to communication models (line of sight, circle, signal power) that basically depend on the distance between two locations. More realistic ways for building communication maps according to actual signal measurements collected in an environment have been recently proposed. For instance, Gaussian Processes are used to model signal power distribution [4, 17]. Such a model is updated incrementally as more measurements are acquired. However, this method has been used stand-alone and has not been integrated in any exploration system, as we do in this paper.

Finally, it is worth mentioning other systems for coordinated multirobot exploration under communication constraints not using any communication map. For example, in some systems, communication is considered as an unpredictable event that can be opportunistically exploited [2, 7, 8, 27]; others use communication devices that store messages and are spread in the environment [6, 12].

3 The Proposed System

We consider two-dimensional environments which are initially unknown and are explored by a system composed of multiple mobile robots $R = \{r_1, r_2, \ldots, r_m\}$ and of a fixed Base Station (BS). The presence of a BS is required in applications in which human operators have to supervise the exploration process (e.g., in search and rescue). Each robot r_i can perceive the surrounding environment using a laser range scanner and can communicate with other robots and with the BS using an onboard radio-frequency transceiver (such as WiFi). We assume that, when communication is possible, the bandwidth is enough to support information exchange within our system. Moreover, we assume that the possibility of a communication between two locations depends only on the locations and on the (generally unknown) environment; for instance, there are no transient disturbances of the signal.

Our system basically operates as follows (similarly to that of [5]).

(1) Each robot perceives the surrounding environment with its laser range scanner and sends the perceived data to the BS.
(2) The BS uses a graph-based SLAM algorithm [24] to localize the robots and to merge the perceived data in an occupancy grid map. (We assume that localization is accurate enough for our purposes.)
(3) The BS calculates a new assignment of robots either to frontiers (i.e., to locations between known and unknown portions of the environment) or to other locations to form a connected configuration in which every robot can communicate with the BS. Then, the BS sends the updated map and the assignment to the robots.
(4) Each robot receives the updated map and assigned location, plans a path to such location, moves to it, and, when the location is reached, iterates from Step (1). Possible collisions between robots are managed opportunistically when they are detected using procedures that locally adjust the paths of the robots.

The exploration strategy employed in Step (3) guarantees that the robots are connected when they reach their assigned locations. New plans are submitted by the BS as soon as a given number of robots (1 in our experiments) become *ready*. (Informally, a robot is ready if it has relayed all the information to the BS.) Assignment of robots to locations is performed by solving a constrained optimization problem. In particular, the algorithm, given a set of locations in the known portion of the environment, selects the robot-location assignments that maximize an objective function – a weighted sum of the distance the robot has to travel and of the amount of new area the robot is expected to perceive from that location. The robot-location assignments must satisfy the constraint that all the robots form a relay chain that allows multihop communication with the BS. The formulation as an Integer Linear Program is the same of [5] and is not reported here. Please refer to [5] for full details.

To evaluate the constraint mentioned above and to decide if robot-location assignments guarantee connection with the BS (before the robots actually move), a communication map is required. Differently from [5], where the assignment of robots to locations is performed assuming either a line of sight or a circle communication model (see previous section), here we build a communication map based on the actual measurements collected by the robots and we update it as the exploration progresses. Specifically, a (undirected) *communication graph* $G = (V, E)$ is maintained and updated on top of the occupancy grid map. Each vertex in V denotes a location of the environment, while each edge (u, v) in E denotes the presence of a bidirectional communication link between the locations represented by the vertices u and v. The vertices in V are updated, at each iteration of Step (3), by adding the locations reached by the robots.

In Step (4), while moving, robots poll each other (at a frequency of 1 Hz) to obtain radio-frequency signal strength measurements. These measurements are sent to the BS (as soon as the robots are in communication with it at their target locations), that adds them to a queue Q storing all the signal strength measurements performed from the start of the exploration. A generic element of Q is denoted by $q = \langle p_i, p_j, S_{i,j} \rangle$, where p_i and p_j are the locations of the transmitting robot r_i and the receiving robot r_j respectively, when the measurement is taken and $S_{i,j}$ is the received signal power. Note that, in general, communication links are not symmetric [9] – i.e., $S_{i,j} \neq S_{j,i}$ – therefore both measurements are stored.

A conservative prior on the communication model is used to derive the presence of the "safest" set of edges (communication links). We now present some methods able to enrich such a conservative communication graph with measurements collected in Q.

3.1 Edge addition

The first method is called *edge addition* and consists in adding edges to E directly from measurements q, ensuring that bidirectional communication is possible. Specifically, for each u and v in V, if there exist two measurements $q = \langle p_i, p_j, S_{i,j} \rangle$ and $q' = \langle p_k, p_l, S_{k,l} \rangle$ such that:

1. the locations p_i, p_j (p_k, p_l) are in line of sight and closer than a threshold α to vertices u and v (v and u) in V, respectively, which are currently not connected by any edge in E and
2. the measured signal powers $S_{i,j}$ and $S_{k,l}$ are larger than a threshold β,

then the edge (u, v) is added to E.

The first condition ensures that the measurements are correctly associated with the vertices u and v, and the second one ensures that there is enough signal strength to exchange messages in both directions. Note that the value of α determines the degree of spatial approximation that is introduced by considering the signal power $S_{i,j}$ ($S_{k,l}$) actually measured between p_i and p_j (p_k and p_l), as if it was measured between u and v (v and u).

3.2 Edge prediction

The second method for updating the graph G is called *edge prediction* and consists in predicting the presence of edges according to a Gaussian Process model trained with all the measurements Q collected so far. A *Gaussian Process* (GP) is a set of random variables for which each subset follows a Gaussian multivariate distribution [19]. Here, we adopt the approach of [4] that uses a GP to represent the signal strength distribution over the environment. By using GPs, each prediction is associated with a confidence value. Such extra information can be used for deciding which communication links can be considered reliable enough for addition to the graph G that is used for planning movements of robots.

In particular, we define a matrix $\mathbf{X} = [\mathbf{x}^1, \mathbf{x}^2, \ldots, \mathbf{x}^n]$ that contains all the $n = |Q|$ location pairs (each $\mathbf{x}$ is an ordered pair with the location of the transmitting robot and that of the receiving robot) between which the robots have measured the signal power, and a vector $\mathbf{Y} = [y^1, y^2, \ldots, y^n]$ that contains the corresponding measured values. Assume that the measured values are generated by an (unknown) noisy process $\mathbf{Y} = f(\mathbf{X}) + \epsilon$ (with $\epsilon \sim \mathcal{N}(0, \sigma_n^2)$). A GP is trained to estimate the posterior distribution over $f()$. The key idea is that the covariance between two function values, $f(\mathbf{x})$ and $f(\mathbf{x}')$, depends on the input values themselves, $\mathbf{x}$ and $\mathbf{x}'$. This dependency can be specified via a covariance, or kernel, function $k(\mathbf{x}, \mathbf{x}')$, which in this paper is assumed to be a squared exponential:

$$k(\mathbf{x}, \mathbf{x}') = \sigma_f^2 \exp\left(-\frac{1}{2l^2}(|\mathbf{x} - \mathbf{x}'|)^2\right), \tag{2}$$

whose parameters are the signal variance σ_f^2 and length scale l^2, which determine how strongly input values correlate. This choice is motivated by the simplicity and applicability of such a kernel function, when no a priori information is available on the structure of the underlying function The GP is then fully specified by the parameter array $\theta = [\sigma_f^2, l^2, \sigma_n^2]^T$. Notice that the covariance between function values decreases with the distance between their corresponding input values. The correlation between the observed measurements is given by:

$$\mathrm{cov}(\mathbf{Y}) = K(\mathbf{X}, \mathbf{X}) + \sigma_n^2 \mathbf{I}, \tag{3}$$

where $K(\mathbf{X}, \mathbf{X})$ is the covariance matrix of the input values and $\mathbf{I}$ is the identity matrix. As shown in [19], it is possible to learn parameters θ based on the training data $\mathbf{X}$ and $\mathbf{Y}$ using *hyperparameter* estimation. An estimate of θ can be computed by maximizing the observations log-likelihood:

$$\theta^\star = \arg\max_\theta \log p(\mathbf{Y}|\mathbf{X}, \theta), \tag{4}$$

where:

$$\log p(\mathbf{Y}|\mathbf{X}, \theta) = -\frac{1}{2}\left(\mathbf{Y}^T \mathrm{cov}(\mathbf{Y})^{-1}\mathbf{Y} - \log|\mathrm{cov}(\mathbf{Y})| - n \log 2\pi\right). \tag{5}$$

Note that optimizing Eq. (5) takes $O(n^3)$ because of the inversion of the covariance matrix. Therefore, to construct our GP in real time, we use only a subset of the measurements in Q – obtained by sub-sampling Q and taking pairs of measurements ($\langle p_i, p_j, S_{i,j}\rangle$ and $\langle p_j, p_i, S_{j,i}\rangle$) separated by at least 30 seconds and farther than 2 m from locations of any previous measurement pair.

We can now define a matrix $\mathbf{W}$ that represents the pairs of vertices (locations) where to predict the signal power values. In particular, called $\mathbf{W} = [\mathbf{w}^1, \mathbf{w}^2, \ldots, \mathbf{w}^\ell]^T$ a set of arbitrary pairs of vertices of G, $p(f(\mathbf{W}) \mid \mathbf{X}, \mathbf{Y}) \sim \mathcal{N}(\mu_\mathbf{W}, \mathbf{\Sigma}_\mathbf{W})$, where the mean vector is obtained as:

$$\mu_\mathbf{W} = K(\mathbf{W}, \mathbf{X})\mathrm{cov}(\mathbf{Y})^{-1}\mathbf{Y} \tag{6}$$

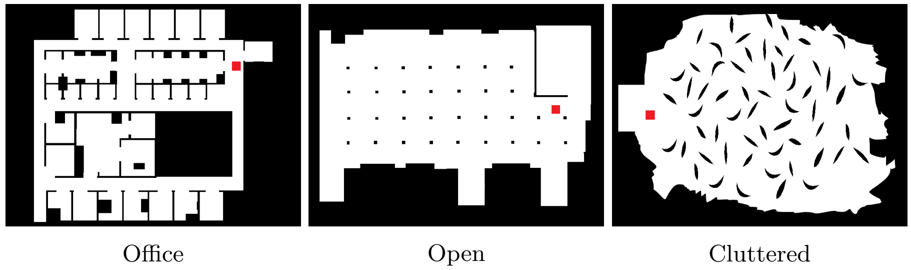

Fig. 2. Simulation environments, approximate size 80×60 m. Red squares denote the BS and the robots' starting locations.

and represents the estimate $\hat{f}(\mathbf{W})$, while the covariance matrix is given by:

$$\mathbf{\Sigma_W} = K(\mathbf{W}, \mathbf{W}) - K(\mathbf{W}, \mathbf{X})\mathrm{cov}(\mathbf{Y})^{-1}K(\mathbf{W}, \mathbf{X})^T. \tag{7}$$

In particular, the main diagonal of $\mathbf{\Sigma_W}$ is usually called predictive variance and is used to measure the uncertainty of estimates in $\mathbf{W}$.

Such a model is then used to add an edge (v_i, v_j) to E if the predicted signal powers $\hat{f}_{i,j} = \hat{f}([\mathbf{w}])$ between v_i and v_j and $\hat{f}_{j,i} = \hat{f}([\mathbf{w}'])$ between v_j and v_i are large enough and have enough low predictive variances $\delta^2_{i,j} = \Sigma_{[\mathbf{w}]}$, $\delta^2_{j,i} = \Sigma_{[\mathbf{w}']}$. More precisely, given a threshold γ, an edge (v_i, v_j) is added to E if:

$$\hat{f}_{i,j} - 2\delta_{i,j} \geq \gamma \quad \wedge \quad \hat{f}_{j,i} - 2\delta_{j,i} \geq \gamma. \tag{8}$$

In order to maintain up-to-date communication maps, this training process is re-started every 45 seconds and, when it ends, the communication graph G is updated.

3.3 Combination of edge addition and edge prediction

The above methods can be used independently of each other. However, it can also be useful to use them in sequence, starting from the same Q. In this way, the edge prediction method will add edges that are not initially added by the edge addition method.

4 Experiments

We implemented the proposed system in ROS [18] considering a team of 4 TurtleBots and a fixed BS, and run an extensive set of simulations in ROS/Stage [25] to assess its performance. The exploration strategy is implemented as a plugin of the "nav2d_exploration" package of the "nav2d" stack[5], which is also used for multirobot (graph-based) SLAM and path planning. By default, ROS/Stage assumes full communication among all the robots and the BS. Therefore, we

[5] http://wiki.ros.org/nav2d

implemented a communication simulator which is able to allow/forbid message passing between the robots according to the realistic communication model presented in [3] and reported as Eq. (1). For all the experiments, we set $P_0 = -38$ dBm, $d_0 = 1$ m, $N = 2.3$, $WAF = 3.37$, and $C = 5$ (which are realistic values for the WiFi transceivers as tested with real TurtleBots). To increase the realism of our simulations, we add a small amount of Gaussian noise with zero mean and unitary variance to the nominal signal strength calculated with Eq. (1). We consider the three environments shown in Fig. 2, displaying different features. Office and Open are from the Radish repository [11] ("sdr_site_b" and "acapulco_convention_center", respectively), while Cluttered is from the MRESim [22] repository ("grass"). For each environment, we complete 5 exploration runs with a deadline of 20 minutes.

Our experimental campaign aims at comparing the *number of edges* in the communication graphs built by *edge addition*, *edge prediction*, and their *combination* w.r.t. the *basic* exploration strategy presented in [5] – representative of the state-of-the-art multirobot exploration systems that consider communication maps not based on actual measurements. Moreover, we compare the amount of *average explored area* and of *average traveled distance* obtainable by using edge addition, edge prediction, and their combination and by the basic exploration strategy. In all cases, we assume that the BS has the following prior on the communication model to compute new plans: any two agents (robots and BS) can safely assume to be able to communicate with each other when their distance is not greater than 15 m, as empirically tested with TurtleBots. Such an assumption guarantees an initialization of the communication graph for the basic strategy. In general, this assumption can be changed and our proposed method is not affected by it. For edge addition and edge prediction, the following parameters are used: $\alpha = 3$ m, $\beta = -80$ dBm, and $\gamma = -93$ dBm. This choice of the parameters guarantees that each edge added to the communication graph represents an actual communication link.

Table 1 shows the average number of edges (over runs) of the communication graphs built by edge addition, edge prediction, and their combination for Office. The number of vertices is similar in the communication graphs built by all the methods. For the edges added in the case of combination, edge addition contributes to approx. 25% of the total number of edges while edge prediction adds the remaining 75%. It is clear that the proposed methods are able to build much richer communication graphs (i.e., with many more edges) than the basic exploration strategy. This means that the assignment of robots to locations to satisfy the connectivity constraint with the BS can exploit a larger number of opportunities (see Fig. 1 for an example of utility of richer communication maps). These results are consistent also for other environments (Tables 2 and 3) and we can conclude that the combination of edge addition and of edge prediction adds more edges to the communication graph, as expected.

Fig. 3 shows the exploration performance in Office. For each of the two metrics, the performance is shown as a set of time-stamped histograms displaying average values (over runs) with standard deviation bars. As the mission unfolds,

basic	edge addition	edge prediction	combination
833	997	2283	3123

Table 1. Average number of edges in the communication graphs of Office after 20 minutes of exploration.

basic	edge addition	edge prediction	combination
1124	1178	2252	3158

Table 2. Average number of edges in the communication graphs of Open after 20 minutes of exploration.

we can notice that the usage of the combination method provides advantages w.r.t. the state-of-the-art basic strategy, more in terms of explored area than of average traveled distance. The performance gains obtained by the two individual methods (edge addition and edge prediction), instead, is less evident. This suggests that exploration performance can be improved by a combination of measurement-grounded (edge addition) and speculative (edge prediction) updating of the communication graph. We note that the performance gain in terms of explored area of edge prediction and of combination increases as the exploration proceeds. This can be motivated by the fact that the GP is trained with more samples and returns more accurate predictions. As such, the BS can allocate robots more effectively in the environment. A nice feature of the GP-based approach to derive new communication links is that it provides a communication map that allows predictions over the whole environment.

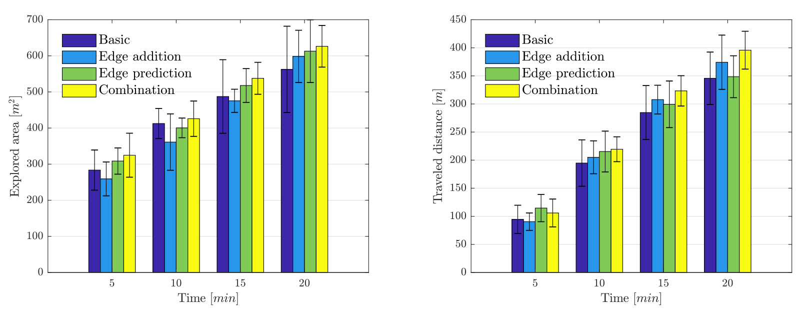

Fig. 3. Performance in Office: explored area (left) and average traveled distance (right). Each histogram reports average and standard deviation.

Fig. 4 shows the performance obtained in Open. In this environment, none of the proposed methods seems to be able to provide significant advantages w.r.t. the basic strategy in terms of explored area. However, the combination method results in a consistently shorter average traveled distance. Such a difference in performance gains compared to Office could be due to the simpler and less struc-

basic	edge addition	edge prediction	combination
1187	1424	2498	4115

Table 3. Average number of edges in the communication graphs of Cluttered after 20 minutes of exploration.

tured environment in Open: a robot can almost always reach a given location by moving directly there and robots can travel short distances to construct relay chains that connect new frontiers to the BS. This opportunity is better exploited by the combination method that leverages on a richer communication graph (see Table 2).

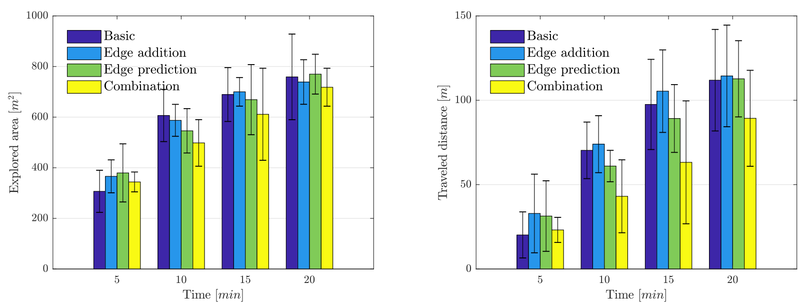

Fig. 4. Performance in Open.

Fig. 5 shows the performance obtained in Cluttered. In terms of explored area, the combination method seems to provide an advantage comparable to that obtained in Office. In terms of average traveled distance, instead, data do not allow to draw strong conclusions.

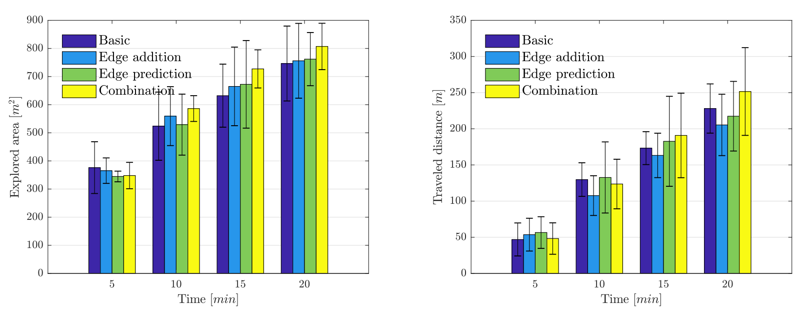

Fig. 5. Performance in Cluttered.

5 Conclusions

In this paper, we presented a multirobot system that explores initially unknown environments while maintaining recurrent connectivity with a base station under limited communication ranges. The system makes decisions about where the robots should move according to a representation, embedded in a graph, of the pairs of locations from which robots can communicate. The main original contribution of this paper is the online updating of the communication graph as the exploration progresses, according both to measurements of radio-frequency signal strength performed by the robots and to prediction of signal strength values performed using Gaussian Processes. This contrasts with other systems proposed in the literature that do not consider actual measurements collected in the environment to build the communication map used to assign robots to locations in the environment. Experimental activities performed in simulation show that the proposed methods are able to build richer communication maps, which can provide advantages to the exploration process, especially in structured environments. We are currently implementing and testing the system on real TurtleBots to further assess the outcomes of this paper (building on results that validate the GP model for WiFi signals [4, 17]).

Future work includes the enhancement of our methods for updating communication graphs; for instance, using correlated α and β values for edge addition, such that the closer a robot to a vertex (small α), the less restrictive the power threshold (large β). Other exploration strategies could be considered in addition to the frontier-based strategy we employed. Moreover, the extension of our methods to time-varying environments in which edges representing communication links could also be deleted is worth investigation. More generally, it would be interesting to develop exploration strategies that drive robots around to seek, at the same time, the coverage of the environment and the building of reliable communication maps.

References

1. Amigoni, F., Banfi, J., Basilico, N.: Multirobot exploration of communication-restricted environments: A survey. IEEE Intel Sy Mag 32(6), 48–57 (2017)
2. Andre, T., Bettstetter, C.: Collaboration in multi-robot exploration: To meet or not to meet? J Intell Robot Syst 82, 325—337 (2016)
3. Bahl, P., Padmanabhan, V.: Radar: An in-building RF-based user location and tracking system. In: Proc. INFOCOM. vol. 2, pp. 775–784 (2000)
4. Banfi, J., Quattrini Li, A., Basilico, N., Rekleitis, I., Amigoni, F.: Multirobot online construction of communication maps. In: Proc. ICRA. pp. 2577–2583 (2017)
5. Banfi, J., Quattrini Li, A., Rekleitis, I., Amigoni, F., Basilico, N.: Strategies for coordinated multirobot exploration with recurrent connectivity constraints. Auton Robot 42(4), 875–894 (2017)
6. Brass, P., Cabrera-Mora, F., Gasparri, A., Xiao, J.: Multirobot tree and graph exploration. IEEE T Robot 27(4), 707–717 (2011)
7. Burgard, W., Moors, M., Stachniss, C., Schneider, F.: Coordinated multi-robot exploration. IEEE T Robot 21(3), 376–386 (2005)

8. Fox, D., Ko, J., Konolige, K., Limketkai, B., Schulz, D., Stewart, B.: Distributed multirobot exploration and mapping. P IEEE 94(7), 1325–1339 (2006)
9. Heurtefeux, K., Valois, F.: Is RSSI a good choice for localization in wireless sensor network? In: Proc. AINA. pp. 732–739 (2012)
10. Hollinger, G., Singh, S.: Multirobot coordination with periodic connectivity: Theory and experiments. IEEE T Robot 28(4), 967–973 (2012)
11. Howard, A., Roy, N.: The robotics data set repository (Radish). http://radish.sourceforge.net/ (2003)
12. Jensen, E., Lowmanstone, L., Gini, M.: Communication-restricted exploration for search teams. In: Proc. DARS. pp. 17–30 (2016)
13. Mukhija, P., Krishna, K., Krishna, V.: A two phase recursive tree propagation based multi-robotic exploration framework with fixed base station constraint. In: Proc. IROS. pp. 4806–4811 (2010)
14. Pandey, R., Singh, A., Krishna, K.: Multi-robot exploration with communication requirement to a moving base station. In: Proc. CASE. pp. 823–828 (2012)
15. Pei, Y., Mutka, M.: Steiner traveler: Relay deployment for remote sensing in heterogeneous multi-robot exploration. In: Proc. ICRA. pp. 1551–1556 (2012)
16. Pei, Y., Mutka, M., Xi, N.: Connectivity and bandwidth-aware real-time exploration in mobile robot networks. Wirel Commun Mob Comput 13(9), 847–863 (2013)
17. Penumarthi, P., Quattrini Li, A., Banfi, J., Basilico, N., Amigoni, F., O'Kane, J., Rekleitis, I., Nelakuditi, S.: Multirobot exploration for building communication maps with prior from communication models. In: Proc. MRS. pp. 90–96 (2017)
18. Quigley, M., Gerkey, B., Conley, K., Faust, J., Foote, T., Leibs, J., Berger, E., Wheeler, R., Ng, A.: ROS: an open-source robot operating system. In: ICRA Workshop on Open Source Software (2009)
19. Rasmussen, C., Williams, C.: Gaussian Processes for Machine Learning. MIT Press (2006)
20. Rooker, M., Birk, A.: Multi-robot exploration under the constraints of wireless networking. Control Eng Pract 15(4), 435–445 (2007)
21. Spirin, V., Cameron, S.: Rendezvous through obstacles in multi-agent exploration. In: Proc. SSRR. pp. 1–6 (2014)
22. Spirin, V., de Hoog, J., Visser, A., Cameron, S.: MRESim, a multi-robot exploration simulator for the rescue simulation league. In: RoboCup 2014: Robot World Cup XVIII, pp. 106–117. Springer (2015)
23. Stump, E., Michal, N., Kumar, V., Isler, V.: Visibility-based deployment of robot formations for communication maintenance. In: Proc. ICRA. pp. 4498–4505 (2011)
24. Thrun, S., Burgard, W., Fox, D.: Probabilistic Robotics. MIT Press (2005)
25. Vaughan, R.: Massively multiple robot simulations in Stage. Swarm Intel 2(2-4), 189–208 (2008)
26. Visser, A., Slamet, B.: Including communication success in the estimation of information gain for multi-robot exploration. In: Proc. WiOPT. pp. 680–687 (2008)
27. Zlot, R., Stentz, A., Dias, M., Thayer, S.: Multi-robot exploration controlled by a market economy. In: Proc. ICRA. pp. 3016–3023 (2002)