

Exploration of Indoor Environments through Predicting the Layout of Partially Observed Rooms

Matteo Luperto
Università degli Studi di Milano
Milano, Italy
matteo.luperto@unimi.it

Luca Fochetta
Politecnico di Milano
Milano, Italy
luca.fochetta@mail.polimi.it

Francesco Amigoni
Politecnico di Milano
Milano, Italy
francesco.amigoni@polimi.it

ABSTRACT

We consider exploration tasks in which an autonomous mobile robot incrementally builds maps of initially unknown indoor environments. In such tasks, the robot makes a sequence of decisions on where to move next that, usually, are based on knowledge about the observed parts of the environment. In this paper, we present an approach that exploits a prediction of the geometric structure of the unknown parts of an environment to improve exploration performance. In particular, we leverage an existing method that reconstructs the layout of an environment starting from a partial grid map and that predicts the shape of partially observed rooms on the basis of geometric features representing the regularities of the indoor environment. Then, we originally employ the predicted layout to estimate the amount of new area the robot would observe from candidate locations in order to inform the selection of the next best location and to early stop the exploration when no further relevant area is expected to be discovered. Experimental activities show that our approach is able to exploit the predicted layout of partially observed rooms in order to speed up the exploration.

KEYWORDS

robot exploration; map prediction; inaccurate knowledge

ACM Reference Format:

Matteo Luperto, Luca Fochetta, and Francesco Amigoni. 2021. Exploration of Indoor Environments through Predicting the Layout of Partially Observed Rooms. In *Proc. of the 20th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2021)*, Online, May 3–7, 2021, IFAAMAS, 8 pages.

1 INTRODUCTION

For autonomous mobile robots, *exploration* is a task that incrementally builds maps of initially unknown environments [28]. Typically, at each stage of the exploration process, a robot selects the next best location (often on a frontier between known and unknown space in the current map) according to an *exploration strategy* [8]. The robot iteratively reaches the selected location, acquires new knowledge on the environment, updates the map, and selects a new next best location, until the environment is fully observed. Decisions made by most exploration strategies are only informed by the knowledge of the observed parts of the environment that are represented by the current map [5, 8]. Structured indoor environments show regularities and symmetries [18] that could be exploited to predict what could be observed in unexplored parts of the environment

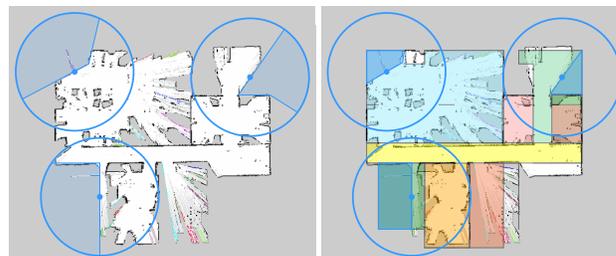


Figure 1: An example application of the proposed method, where the exploring robot has to choose between three candidate locations (blue dots) in a real-world cluttered map (from [24]). Without any knowledge on the unexplored parts of the environment (left) the robot selects the next best location to reach on the basis of what is already represented in the map. For instance, the robot could select the location that maximizes the amount of area ideally covered with a perception (blue circles). (For visualization purposes, we reported a shorter perception range of 2.5 m.) Predicting the shape of the unexplored parts of the environment and, in particular, of the partially observed rooms (right) allows the robot to make a more informed choice. For example, the robot can infer that the two candidate locations on top can provide little information about the environment. However, predictions of rooms' shapes could be inaccurate and lead the robot to discard promising locations.

and, ultimately, to make better decisions when selecting the next best locations to reach. However, such knowledge could be *inaccurate*, in the sense that predictions about unexplored portions could not mirror the configurations found in the actual environment. While the use of prior inaccurate and incomplete knowledge that is available *before* the exploration starts (e.g., the floor plan of an environment) has been recently shown to provide advantages to robot exploration [16, 21], the potential of using knowledge predicted on the basis of the currently explored parts of the environment has yet been assessed (to the best of our knowledge).

In this paper, we address the above issue by presenting a method that exploits the on-line prediction of the geometric structure of the unknown parts of an indoor environment to select the next best location for an exploring mobile robot and to terminate the exploration early, namely before all the area of the environment has been actually perceived by the robot, if no further relevant area is expected to be observed.

More precisely, we consider a mobile robot that explores an initially unknown indoor environment in order to build its 2D grid map (Section 3.1). At each stage of the exploration process, we reconstruct, from the current grid map, the layout of the observed parts and we predict a (possibly inaccurate) layout of the rooms of the environment that have been only partially observed. The *layout* is an abstract geometrical representation in which rooms are modeled as polygons, capturing their shape and filtering out noisy data (e.g., misalignment of walls and small pieces of furniture) that are present in grid maps [3, 13, 14, 19].

The shape of partially observed rooms is predicted following the insight that different parts of the building share common features. For example, rooms connected to the same corridor likely share a common wall and have similar shapes, as it usually happens in large schools and offices [20]. By retrieving such structural knowledge from the current map, we provide an estimate of the shape of partially observed rooms. For layout reconstruction and prediction, we employ methods we previously developed [14][17] (summarized in Section 3.2). The predicted layout is then used to evaluate the amount of new area that the robot expects to perceive from candidate locations and to inform its decision when selecting where to go next (Section 3.3 and Fig. 1). Moreover, if the exploration process is expected to be nearly over, namely if the amount of unobserved area in the predicted layout is below a threshold, the exploration is stopped and the predicted layout is used to complete missing map portions.

Experimental activities, mainly conducted in several simulated large-scale indoor environments, show that our use of possibly inaccurate predictions of partially observed rooms could effectively speed up the exploration (Section 4). In general, the more the exploration progresses, the more knowledge about the structure of the environment the robot acquires, the more the layout prediction becomes accurate, and the more gain our proposed method provides wrt classical exploration strategies that only consider knowledge about the observed parts of the environment that are contained in the grid map.

The main original contribution of this paper is thus a method that employs the predicted layout of a partially observed environment to speed up the exploration. We explicitly note that this paper deals with a problem different from the one we addressed in a recent paper [16], where the robot is assumed to have knowledge of indoor environments (like floor plans and footprints) in advance, before starting the exploration. In this paper, instead, we assume to have zero initial knowledge and we predict, from the current map, the shape of partially observed rooms on-line, at each stage of the exploration process.

2 RELATED WORK

In several applications, a robot (or a multirobot system) is placed in an initially unknown environment and needs to acquire relevant knowledge by incrementally covering the free space of the entire environment with its sensors. This task is called exploration. Exploration approaches could be grouped in two main families. *Frontier-based* exploration approaches move the robot to the geometrical boundaries between known and unknown portions of environments [31]. *Information-based* exploration approaches move

the robot to the most informative locations, according to some information measure (e.g., [27]). This paper focuses on frontier-based approaches, as they usually do not assume to know in advance all the locations the robot can possibly reach and thus better deal with the task we consider, that of incrementally building a map of the explored environment.

In frontier-based approaches, robots usually choose the next best frontier to visit according to on-line greedy exploration strategies [30]. For instance, the method in [8] evaluates each candidate frontier considering both its distance from the robot (closest frontiers are preferred) and the expected amount of information obtainable from the frontier (most informative frontiers are preferred). The two criteria are then combined in an *ad hoc* utility function. A similar method is reported in [29]. Other approaches, like [2] and [5], present more principled methods, based on multi-objective optimization, to aggregate the criteria that are used to evaluate frontiers. A complete survey is beyond the scope of this paper and can be found in [11]. Differently from ours, all these methods evaluate candidate frontiers only according to the knowledge of the parts of the environment the robot has directly observed.

The use of other forms of knowledge to integrate the information from the current map has been investigated with the aim of improving the performance of exploration. In [22], the possible aspect of the unexplored parts of an environment is predicted by exploiting a database of previously mapped environments, in order to complete the partial map obtained by the robot. A similar approach, but extended to multirobot settings, is that of [26]. Unlike our approach, both [22] and [26] use knowledge relative to environments different from the one where the robot operates. Hence, while the above methods rely on the presence of libraries of environments observed in the past, our approach can be applied also when such data are not available.

The authors of [21] propose an exploration approach that, knowing a representation of the environment in terms of a topological graph, finds an efficient exploration path. Similarly to ours, this method exploits the knowledge of the same environment in which the robot operates. However, in [21] (and in [16] mentioned before), the robot is provided with prior knowledge about the environment, while in the approach presented in this paper the robot updates and exploits the knowledge as the exploration progresses.

A method that shares some similarities with our approach is that of [7], which predicts the structure of an unexplored region of an environment to improve SLAM performance. The method reconstructs the neighborhood of a frontier by identifying similar structures in the known map. The prediction of [7] considers the local similarity between different parts of the same environment, while our approach considers more abstract global features like the fact that rooms aligned along the same corridor share the same wall.

Another recent method that is similar to ours is that of [25], where a variational autoencoder (VAE) is employed to predict unknown regions of an environment starting from a partial map. The prediction is then used to compute the expected information gain for candidate locations. The good performance of this approach is related to the fact that the authors consider buildings that are very similar to each other. The generalization of the approach to other environments requires to use more diversified data. Moreover,

the method in [25] considers empty environments (a VAE could be trained with real-world maps with clutter and furniture, but this would require a significantly larger training set). Our method, instead, can be used with any map (in which the walls can be identified) acquired in any environment.

We finally mention some methods that infer the presence and location of specific elements in the unknown parts of environments, like emergency exits [6], labels of unseen rooms [23], and portions of environments represented as graphs [4][15]. These methods do not provide the geometrical information we exploit for speeding up exploration.

3 OUR APPROACH

In this section, we describe the exploration process we consider (Section 3.1), then we detail the methods we use for reconstructing and predicting the layout starting from a partial grid map (Section 3.2) and for exploiting the predicted layout to estimate the amount of information the robot can acquire from a frontier (Section 3.3). Finally, we illustrate the use of the predicted layout to implement an early stopping criterion for the exploration (Section 3.4).

3.1 Frontier-based exploration overview

In frontier-based exploration, a robot incrementally perceives an environment E by iteratively reaching frontiers at the boundaries between known and unknown space. The knowledge acquired about the environment is collected in a map M . In this paper, we consider a robot equipped with a 2D laser range scanner that moves in an initially unknown planar indoor environment E . To simplify our illustration, we assume that the laser range scanner has a field of view of 360° , so we can just consider the location of the robot, ignoring its orientation. We will remove this assumption in Section 4.1. The robot builds a *grid map* M obtained with a SLAM algorithm, where each cell of M can be known (either free or obstacle) or unknown. In our experiments, we use GMapping [9] and we threshold the occupancy grid it returns so that each cell is labeled as free, obstacle, or unknown. Note that our method only requires the availability of an abstract grid map M , disregarding how it has been obtained. We tested our approach with other SLAM methods (Karto¹ and Hector SLAM² [12]) observing no significant change in robot's behavior (results are not reported here).

The robot performs the following steps until no frontier is left and M represents all the free space of E reachable from the initial location of the robot:

- (i) it initializes the map M with a perception from its initial location;
- (ii) it extracts from M a list of frontiers and, for each frontier, it selects a candidate location p ;
- (iii) it selects the next best location p^* using an exploration strategy;
- (iv) it plans and follows a path towards p^* , integrating the perceptions acquired along the way into the map M ; and
- (v) once it reaches p^* , it restarts from (ii).

A *frontier* is a chain (i.e., an ordered set) of adjacent (with a common edge) free cells such that each one of them is adjacent to

at least an unknown cell. The middle cell of a frontier is a *candidate location* p . If the chain has an even number of cells, ties are broken randomly. Given a map M , the set of candidate locations is denoted by C . Candidate locations are ranked according to an utility function $u(p)$ that aggregates distance $d(p)$ and information gain $i(p)$, similarly to [5][8].

The *distance* utility value $d(p)$ is used to prefer closest candidate locations that minimize the travel cost for the robot to reach them:

$$d(p) = \frac{D_{\max} - D(p, p_R)}{D_{\max}}, \quad (1)$$

where $D(p, p_R)$ is the Euclidean distance between the current location of the robot p_R and the candidate location p and D_{\max} is the maximum $D(p, p_R)$ over all the candidate locations $p \in C$.

The *information gain* utility value $i(p)$ represents the new knowledge that the robot expects to acquire from visiting a given candidate location:

$$i(p) = \frac{I(p)}{I_{\max}}, \quad (2)$$

where $I(p)$ is the estimate of the amount of new (unexplored) area visible from p (calculated as described in Section 3.3) and I_{\max} is the maximum value of $I(p)$ over all the candidate locations $p \in C$.

To select locations that are both close and informative, a tradeoff is computed by linearly combining the distance and utility values using a parameter $\alpha \in [0, 1]$:

$$u(p) = \alpha \cdot d(p) + (1 - \alpha) \cdot i(p). \quad (3)$$

The next best candidate location p^* is selected as follows:

$$p^* = \operatorname{argmax}_{p \in C} u(p). \quad (4)$$

Given a value of α , p^* represents the best balance between closeness and expected new area visible and, as such, is considered the best greedy choice for efficient exploration of the environment [1].

Each candidate location $p \in C$ lies on a frontier and, since we consider indoor environments, it is in some partially observed room (defined below). Our method infers the possible geometrical shape (layout) of partially observed rooms, on the basis of structural features of the observed parts of the environment. In particular, given the current grid map M , we retrieve its layout \mathcal{L} by identifying the rooms and labeling them as fully or partially observed. While the layout of fully observed rooms is known, that of partially observed rooms is predicted (Section 3.2). Then, the predicted layout of a partially observed room containing p is used to provide an informed value for $I(p)$ in equation (2) (Section 3.3).

3.2 Predicting the layout of partially observed rooms

To predict the shape of partially observed rooms, we use the environment layout \mathcal{L} obtained from a partial grid map M of the environment using the method presented in [14][17]. (In principle, any other method that provides a geometrical estimate of the shape of partially observed environments could be used.) We provide here a summary of the algorithm using a running example (Fig. 2). Please refer to the original papers for full details. The main idea of the method is to identify the shape of fully observed parts of the environments and to use such structural knowledge to predict the possible shape of partially observed rooms.

¹<http://www.ros.org/wiki/karto>

²http://www.ros.org/wiki/hector_slam

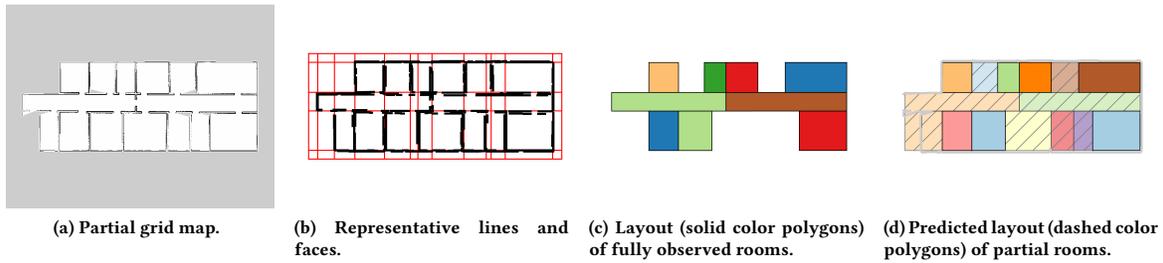


Figure 2: An example run of the approach we use for retrieving the layout starting from a partial grid map. In the map of Fig. 2a 95% of the area has been explored. In Fig. 2d, the layout of partially observed rooms is dashed and the layout of fully observed rooms is solid. The map known to the robot is superimposed with gray lines. The same color code is used for layouts \mathcal{L} in the rest of the paper.

The method starts from a partial grid map M of the environment, like that of Fig. 2a. From M , a set of edges is extracted and used to identify walls. Each wall is then associated to a *representative line*, which indicates the direction of collinear (along the same direction), but possibly spatially separated, walls. Representative lines, reported in red in Fig. 2b, segment the environment into a set of *faces*. (Although faces are rectangular in the example of Fig. 2b, they are in general polygonal because representative lines are not constrained to intersect orthogonally, for example when the environment has diagonal walls.) Faces can be of three types: *fully observed*, if their area has been completely observed in M , *partially observed*, if their area has been partially observed in M , and *unknown*, if no point of their area has been observed in M .

Then, the faces are clustered in groups, each one representing a room. Intuitively, two adjacent faces are clustered together if their common edges does not correspond to any observed wall. The polygon representing the layout of a room is obtained by merging the faces of the group corresponding to the room. Rooms are distinguished in *fully observed* rooms, only composed of fully observed faces, and *partially observed* rooms, also composed of partially observed or unknown faces. An example of fully observed rooms identified from the partial grid map of Fig. 2a is in Fig. 2c.

Finally, the estimated layout of partially observed rooms is computed by using information from representative lines, faces, and fully observed rooms. The idea is to find out the set of faces (not belonging to any room) that will form a room whose shape is maximally “consistent” with the structure of the rest of the environment. For example, if one side of a room is bounded by a corridor, the opposite side of that room likely shares a wall with adjacent rooms along the same corridor. Practically, starting from a partially observed face belonging to a partially observed room r containing a frontier, all the sets of adjacent faces (fully observed, partially observed, and unknown, which have been not clustered in any room) that can complete r (e.g., that are connected) are iteratively considered. The set of faces that maximizes a function that measures the expected quality of the predicted shape (according to its consistency wrt to fully observed rooms, to its convexity, and to the number of its walls) is then found and considered as the most likely shape for r . An example of the predicted layout of partially observed rooms is

in Fig 2d. The retrieved layout $\mathcal{L} = \{r_1, r_2, \dots\}$ is eventually composed of the layout of both fully observed and partially observed rooms. While the layout of fully observed rooms comes from direct observations made by the robot and, consequently, is accurate, the layout of partially observed rooms can be inaccurate and rather different from the shape of the actual environment.

A particular situation is encountered when a partially observed room is at the border of the map M . In this case, one or more sides of the room are not bounded by any representative line derived from M and the layout of the room cannot be predicted, as we cannot exploit any structural knowledge for this end. When this happens, we label the room as containing an *open frontier* and we highlight the corresponding edges in red, as in the first three examples of Fig. 5 (discussed later). This particular situation usually occurs at early stages of exploration, where only a limited portion of the environment has been explored.

3.3 Expected information gain $I(p)$

In this paper, we originally exploit the retrieved layout \mathcal{L} to calculate $I(p)$ in equation (2), namely to estimate the amount of unexplored area visible from a candidate location p (see Fig. 3a). The mainstream approaches for calculating $I(p)$ measure the maximum visible area from p given the footprint of the robot’s laser range scanner (as done, e.g., in [5, 8]) or the length of the frontier on which p lies (as partially done, e.g., in [29]). These approaches are designed for settings where no knowledge about the unobserved parts of the environment is taken into account. Fig. 3b shows an example in which $I(p)$ is calculated as the area of the maximum number of unknown cells that can be perceived by the laser range scanner from p . This estimate is optimistic and implicitly assumes that the area beyond the frontier on which p is located is free.

In our approach, we calculate $I(p)$ as follows. Given a map M and its retrieved layout \mathcal{L} , the walls identified in \mathcal{L} (corresponding to the edges of the polygons representing the rooms) are projected on M as obstacles (occupied cells), thus obtaining a new map $M^{\mathcal{L}}$. Given a cell $p \in M$, we find the corresponding cell $p^{\mathcal{L}} \in M^{\mathcal{L}}$. Then, for each unknown cell $c \in M$ that is within the footprint of the laser range scanner when the robot is in p , we find the corresponding cell $c^{\mathcal{L}} \in M^{\mathcal{L}}$. The cell c contributes to calculating the expected area $I(p)$ visible from p when both the following conditions are satisfied:

- $c^{\mathcal{L}}$ is free,
- $c^{\mathcal{L}}$ is visible from $p^{\mathcal{L}}$ in $M^{\mathcal{L}}$, namely the line segment connecting their centers does not touch any obstacle cell in $M^{\mathcal{L}}$.

Eventually, given the cells c that satisfy the above conditions, $I(p)$ is calculated by summing the areas of those cells. Fig. 3d shows an example in which the method just described is used to calculate $I(p)$. It is interesting to contrast it with Fig. 3b. Although it is a sort of informed variant of the classical frontier-based exploration approaches, the proposed method provides benefits to the performance of exploration also when the retrieved layout \mathcal{L} and the grid map $M^{\mathcal{L}}$ in which it is embedded are inaccurate, as we show in the next section. In case of open frontiers, where the information about \mathcal{L} cannot be used to estimate $I(p)$, the maximum area that can be perceived from p is considered (as in [5, 8]).

3.4 Early stopping of exploration

Exploration missions are usually performed until the entire area of the environment is mapped by the robot. As a consequence, classical exploration techniques as [5, 8] often result in the following behavior: at the beginning of exploration, the robot quickly increments its map M by visiting close locations with high information gain. However, it usually leaves behind small scattered frontiers across different rooms, as in the example of Fig. 2. Hence, in the final stages of the exploration process (e.g., at 90% or 95% of the total area explored), the robot has to reach all such remaining frontiers and perceive the environment from there. These residual frontiers are particularly costly to visit, being usually in rooms far away from each other, and often result in small information gains, as they typically represent small gaps like corners.

The retrieved layout \mathcal{L} can be used to estimate the missing parts of partially observed rooms and to automatically fill the small gaps without actually observing them. Considering the example of Fig. 2, our method is able to provide a correct estimate of the shape of the

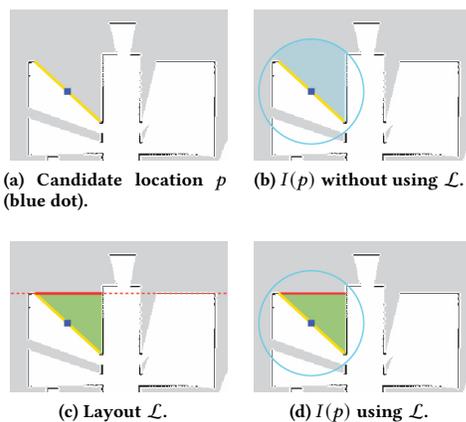


Figure 3: An example of how $I(p)$ is calculated without (Fig. 3b) and with (Fig. 3d) the knowledge of \mathcal{L} . Fig. 3c shows the representative line (red, dashed) of the wall used to predict the layout (red, solid) of the partially observed room in \mathcal{L} .

area visible from all of the remaining frontiers. In order to exploit this feature, we introduce a criterion for stopping the exploration early, which is based on \mathcal{L} . More precisely, *Early Stopping* (ES) ends the exploration if the estimated unexplored area visible from all the current candidate locations in C is less than a threshold. When ES is triggered, we consider the exploration complete and discard the remaining frontiers, as we can easily predict the area that can be perceived from them.

4 EXPERIMENTAL EVALUATION

In this section, we evaluate the proposed approach in exploring large-scale simulated buildings. Note that, we resort to simulations to perform repeated experiments in different environments, as publicly available maps from repositories do not account for the online decisions made by the exploration strategies.

4.1 Experimental setting

We implemented our method in ROS, using the ROS navigation stack. Explorations are performed in 10 large-scale buildings (from 1000 m² to 3500 m²) simulated in Stage³, using a simulated robot equipped with a laser range scanner with a field of view of 180° and a range of 6 m. To address a field of view that is narrower than 360°, as assumed in Section 3, we consider that the robot, at a frontier location, heads towards the unknown space along the perpendicular to the tangent to the frontier, in order to maximize the perceived area. Environments are selected as examples of complex real-world office and school environments with more than 15 rooms. They have polygonal structures, like the vast majority of indoor environments, and are large-scale, to have meaningful predictions.

We perform, for each environment, 10 runs using our method for estimating the information gain $I(p)$ (“with \mathcal{L} ”) and using a baseline method similar to those of [5, 8], which is representative of methods currently most used in the literature (“no \mathcal{L} ”), in which the information gain $I(p)$ is calculated as in Fig. 3b. We measure, as exploration progresses, the percentage of *explored area* (*exp*), namely the percentage of free area of E mapped in M , as a function of the *time*. We compute mean and standard deviation (over all the runs) of the time required to perform a full exploration for each environment (namely time at *exp* = 100%). The starting locations of the robot are realistically placed at the entrances of the buildings and are fixed for all the runs. The maps obtained in different runs are slightly different from each other, because of the noise introduced in the simulation (translational error up to 0.01 m/m and rotational error up to 2°/rad), resulting in different frontiers being detected and, ultimately, in different choices being made by the robot.

The layout \mathcal{L} is retrieved on-line during the exploration mission starting from the grid map M provided by the ROS implementation of GMapping⁴. At the beginning of the exploration, as only few rooms have been fully observed, the predicted layout of partially observed rooms could be highly inaccurate, and several open frontiers could be detected. Note that, differently from methods like [25], we do not require any prior data to learn a model. With the progression of the exploration, however, \mathcal{L} becomes more stable and accurate (confirming the findings of [17]). The computation of \mathcal{L} and of $I(p)$,

³<http://wiki.ros.org/stage>

⁴<http://wiki.ros.org/gmapping>

| no \mathcal{L} | | with \mathcal{L} | | with $\mathcal{L} + \text{ES}$ | | gain | |
|------------------|----------|--------------------|----------|--------------------------------|----------|--------------------|--------------------------------|
| t | σ | t | σ | t | σ | with \mathcal{L} | with $\mathcal{L} + \text{ES}$ |
| 3440 | 313 | 3090 | 316 | 3003 | 321 | 10.1% | 12.8% |

Table 1: Exploration results over 10 runs in 10 simulated large-scale buildings. t (in s) is the average time and σ is the corresponding standard deviation. “no \mathcal{L} ” is the baseline method, “with \mathcal{L} ” is our proposed method, and “with $\mathcal{L} + \text{ES}$ ” is the variant of our method with ES. The percentage gain of our method is reported in the last two columns.

for all the candidate locations on the frontiers, takes few seconds at early stages of exploration and less than 10 s at final stages, and is performed by a dedicated ROS node so that an updated version of \mathcal{L} is always available without introducing delays. After performing several tests, we set $\alpha = 0.5$ in (3) to equally balance distance and information gain and guarantee a good overall performance in exploration. With $\alpha \approx 1$, the robot tends to select the closest frontier, without considering the information gain and, ultimately, slowing down the exploration process by choosing frontiers with a limited information gain. With $\alpha \approx 0$, the robot tends to select the most informative frontier even if it is far from its current position. This results in an exploration pattern where the robot could travel to the extremities of the environment to reach frontiers with a high $i(p)$, ultimately slowing down the exploration process and increasing the travel cost. The balance between the two components of equation (3) obtained with values of α around 0.5 produces a more stable and more efficient performance across different environments.

Exploration runs are concluded when no frontier is left and the set of candidate locations C is empty. However, when we employ the ES variant of Section 3.4 (“with $\mathcal{L} + \text{ES}$ ”), the exploration is stopped when the estimated unexplored area visible from all the current candidate locations in C is less than 1 m^2 . We set this threshold to a very conservative value. If increased, the exploration process could terminate earlier, at the cost of possible inaccuracies in estimating the unobserved parts of the environment. This issue is discussed later. Although we do not perform any quantitative evaluation of the quality of the maps obtained in our experiments, visual inspection does not identify any relevant imperfection.

4.2 Experimental results

Table 1 reports the average time required for performing 10 complete exploration runs in all the 10 large-scale environments. The time includes retrieving the layout \mathcal{L} and computing $I(p)$ for all candidate locations at each stage. The use of \mathcal{L} for estimating the information gain brings a speedup of the total exploration time of approximately 10%. More detailed results on the average time (over 10 runs) required to completely explore ($exp = 100\%$) each one of the 10 environments are reported in Fig. 4. For all 10 environments, our method results in a speedup of exploration. The use of ES further decreases the time for almost all of the environments.

We now look at the progression of the exploration during one run performed in the environment of Fig. 2. Fig. 5 reports the partial grid maps and the corresponding retrieved layouts \mathcal{L} at different exploration stages. As expected, \mathcal{L} becomes more stable

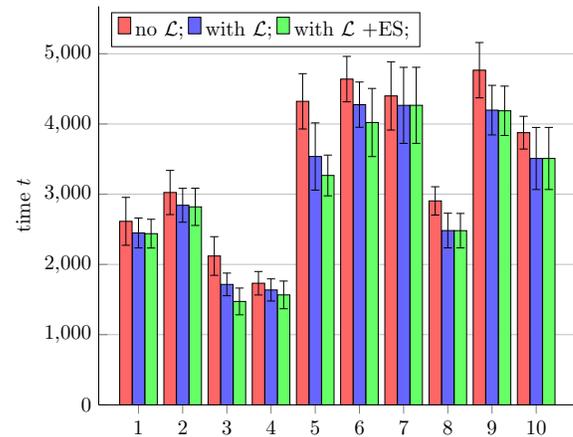


Figure 4: Average exploration time (and standard deviation) for all the 10 environments (on the x axis) with $exp = 100\%$.

and accurate as the exploration progresses. Reliable estimates of $I(p)$ can be obtained at $exp = 60\%$ and even at $exp = 20\%$. Fig. 6 reports the explored area (averaged over 10 runs) as a function of time. Our method performs similarly to the baseline until the explored area is around 90%, when we are able to provide a better estimate of $I(p)$ and, consequently, to speed up the final part of the exploration. For this example, “with \mathcal{L} ” has a gain of 19.1% over “no \mathcal{L} ”, while “with $\mathcal{L} + \text{ES}$ ” has a gain of 30.5%.

Fig. 7 shows an example of a reconstructed layout from a grid map with $exp = 80\%$, where most of the partially observed rooms are correctly predicted, despite the fact that relatively large portions of the building are still unobserved. In settings like this one, our proposed method could stop the exploration early, because the reconstructed layout correctly represents the actual environment and ES can be activated. Fig. 8 shows a partial map obtained during the early stages of an exploration run. It can be noticed how the layout of the rooms is a rough estimate of the correct one, as little information about the shape of the rooms is known at this point. Nevertheless, even in this case, our method can provide a reasonable estimate for $I(p)$ for all frontiers.

Further results, including some in more complex environments, are reported in the video available at: <https://amigoni.faculty.polimi.it/research/AAMAS2021-exploration-through-prediction.html>.

4.3 Discussion

An interesting, although intuitively expected, result of our experimental analysis is that the availability of a more accurate $I(p)$, using the predicted layout \mathcal{L} , produces a speedup in exploration. This fact is remarkably evident at the end of the exploration runs, when \mathcal{L} is more accurate. Indeed, the speedup obtained by our method is not uniformly distributed over the entire exploration process (see Fig. 6). Our method performs similarly to the baseline method until, approximately, the 90% of the total area has been explored. From that moment on, it performs consistently better than the baseline method, eventually resulting in the final gain of Table 1. On the one hand, the use of an inaccurate prediction

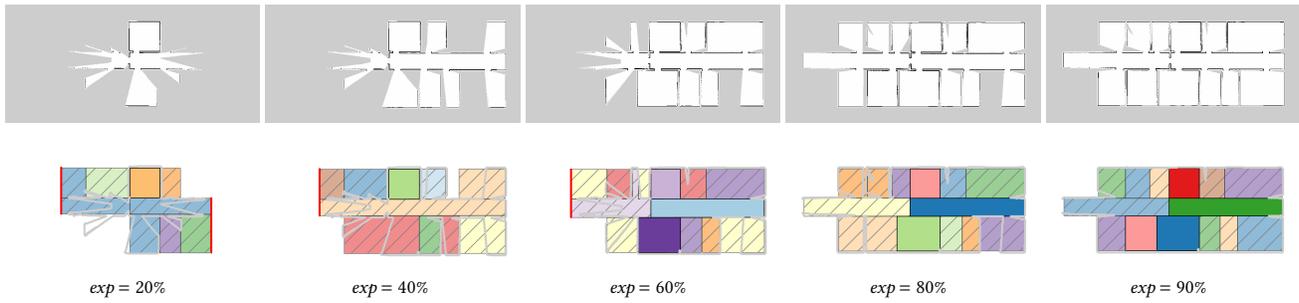


Figure 5: Example of the predicted layout \mathcal{L} built from partial grid maps M obtained at different exploration stages of an environment. The layout obtained in the environment with $exp = 95\%$ is in Fig. 2.

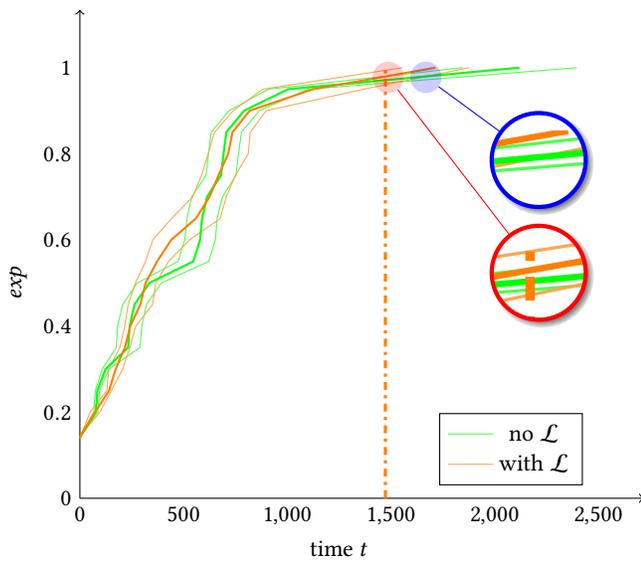


Figure 6: The progression of the explored area (averaged over runs) as a function of time for the environment of Fig. 5. Early stopping (ES) time is indicated by a dashed vertical line. For each method, the solid line in the middle is the average value, while the other two lines are the standard deviation.

about the environment at the early stages of exploration does not jeopardize the gain obtained at the end with an accurate prediction. On the other hand, the use of an inaccurate \mathcal{L} for estimating the information gain produces results similar to those obtained with the mainstream approaches that consider $I(p)$ equal to the footprint of the laser range scanner. When the predicted layout \mathcal{L} becomes enough accurate, the proposed method starts to speed up and, in some cases, the robot does not need to explore until all area of the environment has been actually perceived, because of ES.

We use a very conservative threshold (1 m^2) for the ES criterion, with the aim of avoiding to stop the exploration too early when potentially interesting frontiers could still be present and, consequently, of completing the map with an accurate prediction. In fact,

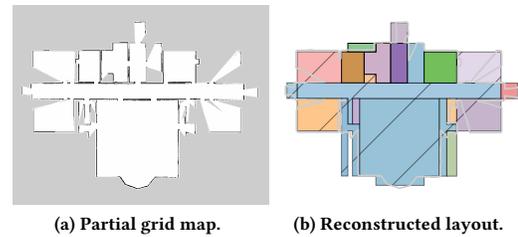


Figure 7: An example of reconstructed layout \mathcal{L} obtained at $exp = 80\%$. In this case, the exploration could be considered complete.

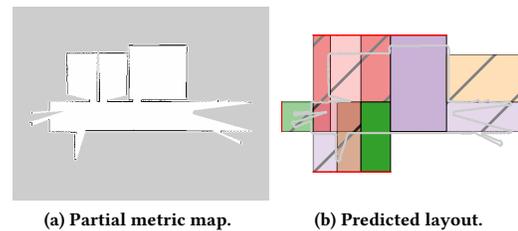


Figure 8: An example of a predicted layout \mathcal{L} obtained with $exp = 30\%$. Despite the fact that only a limited portion of the building is known at this point, the layout \mathcal{L} can provide a good estimate of $I(p)$.

ES is triggered in 7 of the 10 environments we consider: in 3 environments its effects are negligible, while in the other 4 environments it has a significant impact on exploration performance. In these latter environments, ES is activated when the 99% of the area is explored (on average) and provides a remarkable gain in terms of exploration time of 20% (on average). However, observing the runs, we note that \mathcal{L} reliably represents the layout of partially observed rooms when exp is 80 – 90%. Setting ES to stop exploration at $exp = 95\%$, we successfully explore our buildings with only minor inaccuracies in \mathcal{L} and an average gain of 38% in exploration time. Setting ES to terminate at $exp = 90\%$ results in an impressive average gain of 50% in exploration time and in missing one room (which is left

unexplored because not included in \mathcal{L} for each environment, on average. A more aggressive ES criterion that discards candidate locations with low estimated $I(p)$ (according to \mathcal{L}) could potentially provide higher gains in exploration time at the risk (related to the accuracy of \mathcal{L}) of not exploring some relevant frontiers. It is intriguing that, while manual exploration stops when the map is “good enough” for the human operator, automated exploration stops when no frontier is left. In this sense, a criterion for ES should stop exploration when the map quality is adequate for the robot’s tasks.

Finally, our approach can be reliably applied to partial grid maps acquired in real environments. An example of $I(p)$ computed in a map obtained by a real robot (from [24]) is shown in the initial example of Fig. 1. The process of layout reconstruction filters out the clutter and provides a good estimate of the missing parts of the rooms, for all the three frontiers of the example. Similar results have been obtained on maps from [10] (see also results of [17]). In principle, our approach could be applied to irregularly-shaped environments (e.g., with diagonal walls), as it does not assume a Manhattan world, and could also be used in presence of nested rooms. Moreover, the information gain $I(p)$ could be computed effectively even in maps whose reconstructed layout \mathcal{L} is inaccurate due to under- or over-segmentation of rooms, because the overall shape of the environment provided by \mathcal{L} is generally accurate.

5 CONCLUSIONS

In this paper, we have presented a method that shortens the time required by a robot for exploring an initially unknown indoor environment by selecting the next locations according to the predicted layout of the partially observed rooms. Experimental results show that our method outperforms state-of-the-art exploration strategies, similar to those of [5, 8], especially when the predicted layout is accurate. The use of an Early Stopping (ES) criterion, which ends exploration when only uninteresting frontiers are left, could further improve performance.

In addition to devising a more aggressive ES criterion as discussed in the previous section, future work will study a dynamic switch from a classical exploration strategy to our method when the layout \mathcal{L} becomes enough accurate. Moreover, combining prior knowledge (as in [16, 21]) and layout prediction could be investigated, especially in exploring environments for which previous partial maps are available. Finally, experiments with real robots will further assess the improvement provided by our approach to the exploration process.

REFERENCES

- [1] F. Amigoni. 2008. Experimental evaluation of some exploration strategies for mobile robots. In *Proc. ICRA*. 2818–2823.
- [2] F. Amigoni and A. Gallo. 2005. A Multi-Objective Exploration Strategy for Mobile Robots. In *Proc. ICRA*. 3850–3855.
- [3] I. Armeni, O. Sener, A. Zamir, H. Jiang, I. Brilakis, M. Fischer, and S. Savarese. 2016. 3D Semantic Parsing of Large-Scale Indoor Spaces. In *Proc. CVPR*. 1534–1543.
- [4] A. Aydemir, P. Jensfelt, and J. Folkesson. 2012. What can we learn from 38,000 rooms? Reasoning about unexplored space in indoor environments. In *Proc. IROS*. 4675–4682.
- [5] N. Basilico and F. Amigoni. 2011. Exploration strategies based on multi-criteria decision making for searching environments in rescue operations. *Auton Robot* 31, 4 (2011), 401–417.
- [6] J. Caley, N. Lawrance, and G. Hollinger. 2016. Deep learning of structured environments for robot search. In *Proc. IROS*. 3987–3992.
- [7] J. Chang, G. Lee, Y. Lu, and C. Hu. 2007. P-SLAM: Simultaneous localization and mapping with environmental-structure prediction. *IEEE T Robot* 23, 2 (2007), 281–293.
- [8] H. González-Baños and J. Latombe. 2002. Navigation Strategies for Exploring Indoor Environments. *Int J Robot Res* 21, 10-11 (2002), 829–848.
- [9] G. Grisetti, C. Stachniss, and W. Burgard. 2007. Improved Techniques for Grid Mapping with Rao-Blackwellized Particle Filters. *IEEE T Robot* 23 (2007), 34–46.
- [10] A. Howard and N. Roy. 2003. The Robotics Data Set Repository (Radish). <http://radish.sourceforge.net/>
- [11] M. Julia, A. Gil, and O. Reinoso. 2012. A comparison of path planning strategies for autonomous exploration and mapping of unknown environments. *Auton Robot* 33, 4 (2012), 427–444.
- [12] S. Kohlbrecher, J. Meyer, O. von Stryk, and U. Klingauf. 2011. A Flexible and Scalable SLAM System with Full 3D Motion Estimation. In *Proc. SSRR*.
- [13] Z. Liu and G. von Wichert. 2014. A generalizable knowledge framework for semantic indoor mapping based on Markov logic networks and data driven MCMC. *Future Gener Comp Sy* 36 (2014), 42–56.
- [14] M. Luperto and F. Amigoni. 2018. Extracting Structure of Buildings using Layout Reconstruction. In *Proc. IAS-15*.
- [15] M. Luperto and F. Amigoni. 2019. Predicting the global structure of indoor environments: A constructive machine learning approach. *Auton Robot* 43, 4 (2019), 813–835.
- [16] M. Luperto, M. Antonazzi, F. Amigoni, and N. A. Borghese. 2020. Robot exploration of indoor environments using incomplete and inaccurate prior knowledge. *Robot Auton Syst* 133 (2020), 103622.
- [17] M. Luperto, V. Arcerito, and F. Amigoni. 2019. Predicting the Layout of Partially Observed Rooms from Grid Maps. In *Proc. ICRA*. 6898–6904.
- [18] M. Luperto, A. Quattrini Li, and F. Amigoni. 2013. A System for Building Semantic Maps of Indoor Environments Exploiting the Concept of Building Typology. In *Proc. RoboCup*. 504–515.
- [19] C. Mura, O. Mattausch, A. Villanueva, E. Gobetti, and R. Pajarola. 2014. Automatic room detection and reconstruction in cluttered indoor environments with complex room layouts. *Comput Graph* 44 (2014), 20–32.
- [20] E. Neufert and P. Neufert. 2012. *Architects’ data*. John Wiley & Sons.
- [21] S. Obwald, M. Bennewitz, W. Burgard, and C. Stachniss. 2016. Speeding-Up Robot Exploration by Exploiting Background Information. *IEEE RA-L* 1, 2 (2016), 716–723.
- [22] D. Perea Ström, I. Bogoslavskyi, and C. Stachniss. 2017. Robust exploration and homing for autonomous robots. *Robot Auton Syst* 90 (2017), 125 – 135.
- [23] A. Pronobis and P. Jensfelt. 2012. Large-scale semantic mapping and reasoning with heterogeneous modalities. In *Proc. ICRA*. 3515–3522.
- [24] J. Ruiz-Sarmiento, C. Galindo, and J. González-Jiménez. 2017. Robot@Home, a Robotic Dataset for Semantic Mapping of Home Environments. *Int J Robot Res* 36, 2 (2017), 131–141.
- [25] R. Shrestha, F. Tian, W. Feng, P. Tan, and R. Vaughan. 2019. Learned Map Prediction for Enhanced Mobile Robot Exploration. In *Proc. ICRA*. 1197–1204.
- [26] A. Smith and G. Hollinger. 2018. Distributed inference-based multi-robot exploration. *Auton Robot* 42, 8 (2018), 1651–1668.
- [27] C. Stachniss, G. Grisetti, and Wolfram Burgard. 2005. Information Gain-based Exploration Using Rao-Blackwellized Particle Filters. In *Proc. RSS*. 65–72.
- [28] S. Thrun. 2003. Robotic Mapping: A Survey. In *Exploring Artificial Intelligence in the New Millennium*, G. Lakemeyer and B. Nebel (Eds.). Morgan Kaufmann, 1–35.
- [29] B. Tovar, L. Munoz, R. Murrieta-Cid, M. Alencastre, R. Monroy, and S. Hutchinson. 2006. Planning exploration strategies for simultaneous localization and mapping. *Robot Auton Syst* 54, 4 (2006), 314–331.
- [30] C. Tovey and S. Koenig. 2003. Improved analysis of greedy mapping. In *Proc. IROS*. 3251–3257.
- [31] B. Yamauchi. 1997. A frontier-based approach for autonomous exploration. In *Proc. ICRA*. 146–151.