J Intell Robot Syst (2018) 91:313–331
https://doi.org/10.1007/s10846-017-0709-0

Information-Driven Rapidly-Exploring Random Tree for Efficient Environment Exploration

Jhielson M. Pimentel[1] · Mário S. Alvim[1,2] · Mario F. M. Campos[1] · Douglas G. Macharet[1]

Received: 16 December 2016 / Accepted: 23 August 2017 / Published online: 5 October 2017

Abstract Exploration of unknown environments using autonomous mobile robots is essential in various scenarios such as, for instance, search and rescue missions following natural disasters. The task consists essentially in transversing the environment to build a complete and accurate map of it, and different applications may demand different exploration strategies. In the literature, the most used strategy is a simple greedy approach which visits closest unknown sites first, without considering whether they will likely yield significant information gain about the environment. In this paper, we propose a navigation strategy for efficient exploration of unknown environments that, based on local structures present in the map built so far, uses Shannon entropy to estimate the expected information gain of exploring each candidate frontier. A key advantage of our method over the state of the art is that it allows for the robot to simultaneously (i) select a destination likely to be most informative among all candidate frontiers; and (ii) compute its own path to that destination. This unified approach balances priority among candidate frontiers with highly expected information gain and those closer to the current position of the robot. We thoroughly evaluate our methodology in several experiments in a simulated environment, showing that our approach provides faster information gain about the environment when compared to other exploration strategies.

Keywords Autonomous exploration · Information theory · RRT · Mobile robots

1 Introduction

The task of exploring unknown environments using autonomous mobile robots is essential in many applications. The task consists essentially in perceiving the environment with the robot's sensors and building a complete and accurate map. However, different applications demand different exploration strategies. For example, rescue robots are required to act quickly, covering as much of the environment as possible in a short period of time, while other applications demand maps with high quality without necessarily imposing stringent time constraints.

In general, an autonomous exploration strategy can be divided into three steps: (i) determining a best viewpoint, (ii) navigating to the defined target, and (iii) updating the map and localization estimates. Typically a strategy is characterized by how it performs the first step, which is responsible for the robot's decision about where to go next in order to enhance its knowledge about the environment (map completeness).

One of the most well-known exploration strategies is Yamauchi's Near-Frontier Exploration (NFE) [1], which always selects for exploration the closest frontier (i.e., a boundary between known and unknown areas in the map) to the robot's position. Despite its simplicity, NFE is often not adequate in scenarios in which time is critical, as it drives the robot to new locations without considering whether they are expected to significantly reduce map uncertainty.

In fact, a critical issue is to, given the current map and the robot's position, determine a next destination for exploration that most efficiently leads to an accurate map of

✉ Douglas G. Macharet
doug@dcc.ufmg.br

1 Computer Vision and Robotics Laboratory, Department of Computer Science, Universidade Federal de Minas Gerais, Belo Horizonte, MG, Brazil

2 Laboratory of Information Security, Cryptography, Privacy, and Transparency, Universidade Federal de Minas Gerais, Belo Horizonte, MG, Brazil

the environment. Figure 1 illustrates the problem, showing the paths to three possible frontiers a robot can choose to explore first.

In recent years, information-based approaches have been proposed to determine a next best destination for more efficient exploration [2–4]. Such approaches compute the *(Shannon) entropy* [5] of the map at the current time, and the entropy of the robot's belief about what the map would be updated to if the robot were to visit a certain destination. The difference between these entropy values is called *mutual information*, or the *information gain* [6] of that destination. The higher its value, the more likely it is that a visit to that destination will improve the accuracy of the map faster.

However, estimating what the entropy of the map would be after a potential destination is visited in the future is not a trivial task, and computing all potential observations before the robot reaches a candidate location is not feasible in practice. To make predictions possible, in this work we consider a simple assumption: most indoor environments are composed of straight walls with intersections. Under this assumption, in previous work [7] we proposed a novel method to anticipate information gain of candidate frontiers by analyzing the local structure around each frontier and predicting how its walls are expected to propagate into unknown areas. This strategy makes it viable to estimate the information gain of the unexplored areas.

In the literature, it is common to use information gain in order to select among destinations for exploration. Typically, information gain is combined with other criteria in order to build a *utility function* to guide the robot's decision-making. In our previous work [7], we estimate information gain and use it to recurrently select the region with the highest associated uncertainty. We then deterministically calculate a path to the target region using the Wavefront Propagation algorithm, which can have a high computational cost depending of the environment size. However, in that work we do not consider other relevant information such as the distance from the frontier to the robot's current position.

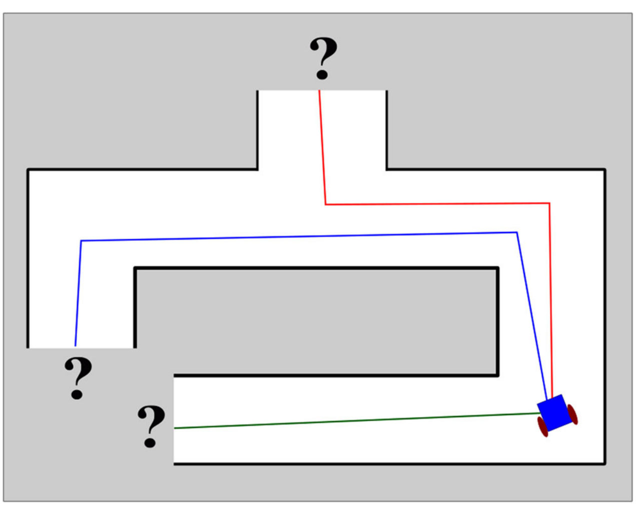


Fig. 1 A fundamental problem in exploration is the determination of a next destination for exploration that is expected to most efficiently cover the environment

Our main contribution in this work is a novel exploration methodology which uses the previously proposed estimated information gain to explore relevant regions. We have improved the robot's decision-making, as well as its approach to determining a viable path to reach a chosen destination. The traditional Rapidly-exploring Random Tree (RRT) algorithm [8] was modified to expand the tree over the map with a bias toward frontiers with higher estimated information gain. As a result, the robot selects a destination and, at the same time, generates a feasible path to reach it. Our approach automatically considers a trade-off between the distance to be traveled and the information gain. Furthermore, the use of a probabilistic path planning algorithm allows for more efficient navigation in larger environments.

The remainder of this paper is organized as follows: A discussion on related works regarding autonomous exploration is presented in Section 2. The proposed methodology is presented in Section 3, and validated by simulated experiments discussed in Section 4. Finally, in Section 5 we draw the conclusions and discuss paths for future investigation.

2 Related Work

Autonomous exploration is of great importance, and has been the object of research in several fields, especially in Robotics. It is a non-trivial task that can be solved with many distinct approaches. In the following we review work related to our aim of increasing map completeness (i.e. reduce uncertainty) in the shortest amount of time.

Among the first autonomous exploration strategies were those based on fixed and random trajectories [9, 10]. While offering low complexity, these classical methods were rather rudimentary, and motivated efforts to improve the robot's decision-making. An improved approach takes into account the cost of reaching candidate destinations [1]. Others attempt to further reduce exploration time by devising utility functions that estimate not only the cost of visiting a particular goal, but also the expected information gain therein [2–4].

Information-based strategies have proved to be a better alternative for calculating information gain. A few such methods guide the robots to frontiers that potentially maximize the reduction of the map's uncertainty. One technique for doing this is to use knowledge of previously seen environments to predict and, consequently, to better estimate the information gain.

Information Gain [5] has been used in many different ways, which can be classified into three main categories: (i) map approaches that use information obtained only from the map to calculate uncertainty [11, 12]; (ii) geometric approaches that use the robot's sensor model and the map to predict areas beyond the frontiers [13–15]; and (iii) prediction approaches that use the robot's sensor model, the map, and sometimes environment knowledge to predict information beyond the frontiers [16, 17].

In the work of Ström et al. [17], it is presented a method that tries to match the area beyond the frontiers with local known maps. The best match indicates the best prediction. As mentioned by the authors, this approach seeks to find similarities between the current surroundings of the robot and previously acquired maps stored in a database in order to predict how the environment may expand in the unknown area. This method yields good results for areas with repetitive structure but can suffer in more unstructured environments. Alternatively, in the work of Oßwald et al. [18] background information is considered in the form of a topo-metric graph to improve the task performance. Based on a rough structure of the environment, the robot exploits this information to cover the environment faster.

There are other approaches that also incorporate environmental knowledge in the robot's decision-making in order to speed up the exploration task. For example, Stachniss et al. [19] use knowledge about the structure of the environment and enhance exploration by identifying corridors and giving them higher utility values. This encourages the robot to move through corridors first, and only then into adjoining rooms. Wurm et al. [20] use knowledge about the environment to coordinate a group of robots during exploration. However, these techniques do not use information gain for its utility function.

For a more in depth discussion related to different autonomous exploration strategies we refer the reader to Juliá et al. [21], which presents a recent overview of the main challenges and existing techniques to the problem.

The main contribution of this work is a complete exploration methodology built upon an adapted version of the Rapidly-exploring Random Tree (RRT) algorithm [8]. The tree is expanded from the robot's current position throughout the environment biased by a simple yet novel technique to predict the uncertainty beyond the frontiers presented in our previous work [7]. We focus on the uncertainty terms but other criteria such as travel distance and frontier size are analyzed indirectly. Further details are presented in the next sections.

3 Methodology

In this section, we present our information-based approach to autonomous exploration. We start by formalizing the problem, describing the robotic system used, the exploration objectives, and the environmental constraints. We, then, present techniques to: (i) predict the information gain of each destination potentially chosen by the robot for exploration; and (ii) generate paths to navigate to the selected destination.

3.1 Problem Statement

The problem of autonomous exploration can be described as follows. Given a robot's current pose in a (partially) unknown environment, and the information collected by the robot about the environment so far, the goal is to produce a map of the environment that is as informative as possible. The robot's main task at each step is to determine a next most informative destination in the environment to move to in order to build the map. In this work we focus on instances of the problem with critical time constraints, such as rescue missions.

We consider a single autonomous mobile robot, which is equipped with a limited-visibility range scanner, and can move in an arbitrary continuous two-dimensional bounded environment $\xi \subset \mathbb{R}^2$ of interest. Time is measured in discrete steps $t \in \{1, 2, 3, \ldots, n\}$, starting from the moment the robot begins the exploration ($t = 1$), and finishing when the robot completes it ($t = n$). We assume no object is moving through the environment except for the robot itself, whose location is represented at any instant t as a point $R_t \in \xi$.

Within its visibility range, the robot's scanner can distinguish free spaces and the outer boundary of obstacles. At each time t, the location of the surfaces scanned by the robot's sensor are represented by a set $O_t = \{o_{t,0}, o_{t,1}, \ldots, o_{t,u_t}\}$ of points in ξ. More precisely, a point $o_{t,i}$ belongs to O_t if at time t the robot's sensor determines that the particular physical location $0 \leq i \leq u_t$ contains an *obstacle* that should be avoided by the robot during navigation (e.g., a wall or piece of furniture). We assume that the sensor model interprets as *free space* all points between the robot's position R_t and those in O_t. In particular, if there are no obstacles detected (i.e., $O_t = \emptyset$), the whole area scanned by the sensor at that time is interpreted as free space. Points lying beyond the area between the robot's location R_t and the set O_t are ignored at time t, and may be evaluated as free or occupied in future time steps.

During exploration, the robot keeps a representation of all the information about the environment acquired so far. We translate from the continuous environment to a discrete space by using an *Occupancy Grid* model [22], which consists in a discrete spatial grid that reflects the occupancy of the environment as a matrix M in which each cell m_{ij} represents a discrete portion of the space ξ. Each m_{ij} is filled with one of three possible values, indicating whether the corresponding points in environment ξ are considered *occupied* (represented by the value “1”, or the color black), *free* (value “0”, or the color white), or *unknown* (value “-1”, or the color gray). All cells are initialized as unknown at time $t = 1$, and their values are updated by the algorithm responsible for generating the map and updating the robot’ pose (Simultaneous Localization and Mapping problem). We use the Rao-Blackwellized particle filter to learn grid maps from laser range data [23].

During exploration the robot must decide which destinations to visit. In our work, the robot considers as a candidate destination every *frontier*, a concept introduced by Yamauchi [1], which corresponds to all cells in M that are on the boundary between free and unknown cells.

We denote by $\mathcal{F}_t$ the set of all frontiers at time t. Both the distance between the robot and the frontier, and the estimated information gain of the frontier, are considered for choosing destinations.

We now define at high-level the exploration problem; we will make it mathematically precise in the next section.

Problem 1 (Information-driven Single-robot Efficient Autonomous Exploration) Consider a robot with a limited visibility-range scanner, an occupancy grid M representing the robot’s knowledge about an environment ξ, a discrete time interval $t = \{1, 2, \ldots, n\}$, and an initial pose $R_1 \in \xi$ for the robot. Assuming that at time t each frontier $f \in \mathcal{F}_t$ can be identified by the robot, find a sequence of destinations $\mathcal{D} = \mathrm{d}_1, \mathrm{d}_2, \ldots, \mathrm{d}_n$ in ξ, such that each d_t:

a) belongs to a free area close to a frontier $f \in \mathcal{F}_t$, and
b) is expected to have maximum information gain at time t for the map M.

3.2 Information-Theoretic Approach

During exploration the robot can perform *actions* (e.g., move in a certain direction at a certain speed, turn around itself by a certain angle, come to a complete stop, etc.) from a set $\mathcal{X}$. In general, actions are taken to make the robot move to a desired location. We let $x_t \in \mathcal{X}$ denote the robot’s action at time t. We assume that the action x_t and the observations O_t of the robot at any given time t are executed with minimal error, so the robot always knows its position perfectly.

At any time t, we are interested in determining which robot’s actions will most reduce the map’s uncertainty, given all observations $O_1, \ldots, O_t$ collected and all actions $x_1, \ldots, x_n$ taken by the robot so far. For that, we estimate the information gain of each frontier in $\mathcal{F}_t$, under the assumption that those yielding highest expected value of information gain are more likely to aggregate more information to the map M if scanned by the robot.

We denote by $z_t = (O_t, x_t)$ the *probe* realized by the robot at time t, which is the ordered pair of the observations O_t made and the action x_t taken by the robot at that time. We denote by Z_t the random variable associated with a probe at time t, and by $z_{1:t} = z_1, z_2, \ldots z_t$ the sequence of probes between instants 1 and t. The estimated *information gain*, also called *mutual information*, of a predicted probe Z_{t+1} at time $t + 1$ with respect to the map M at time t, given all probes $z_{1:t}$ realized so far, is defined as:

$$I(M; Z_{t+1}|z_{1:t}) = H(M|z_{1:t}) - H(M|Z_{t+1}, z_{1:t}), \qquad (1)$$

where $H(M|z_{1:t})$ is the *current entropy* of the map at time t given all probes from times 1 through t, and $H(M|Z_{t+1}, z_{1:t})$ is the *posterior entropy* of the map at time $t + 1$ conditioned on all past probes from times 1 through t and the probe Z_{t+1} projected to be made at time $t + 1$. To calculate the current entropy of the map conditioned on all sensor readings it is necessary to adapt the values from the occupancy grid map to apply on the following equation [24]:

$$H(M|z_{1:t}) = -\sum_{i,j} p_{ij} \log p_{ij} + (1 - p_{ij}) \log(1 - p_{ij}), \qquad (2)$$

where p_{ij} is the probability of cell m_{ij} in M being occupied, defined as: $p_{ij} = 0$ if $m_{ij} = 0$ (the cell is for certain free), $p_{ij} = 1$ if $m_{ij} = 1$ (the cell is for certain occupied), and $p_{ij} = 0.5$ if $m_{ij} = -1$ (the state of the cell is undetermined, and it considered occupied with probability 50%).

Unfortunately, it is not feasible to calculate the posterior entropy $H(M|Z_{t+1}, z_{1:t})$ without making an adaptation or prediction [17]. This is expected since there exists a vast number of potential measurements for the probe Z_{t+1}, which grows exponentially over time with the size of the environment.

In this work, we propose a suitable approximation that depends only on the current map. With carefully made predictions, it becomes possible to calculate the posterior entropy and then generate the paths.

The methodology can be represented by a diagram with an exploration cycle, as illustrated in Fig. 2.

3.3 Prediction of Unvisited Areas

In this section, we review the approach introduced in our previous work [7] for predicting areas beyond frontiers, and which will be used to guide the robot to locations with the highest information gain.

The posterior entropy $H(M|Z_{t+1}, z_{1:t})$ is a key value in determining a next best destination to be explored by the robot. To compute its value precisely, however, it would be necessary that the robot knew the contents of areas beyond each candidate frontier, and for that the robot would have had to move to those frontiers to collect data in the first place. Since we want the robot to use the posterior entropy to decide the best location to explore *before moving there*, the robot must somehow *predict* the expected changes in the

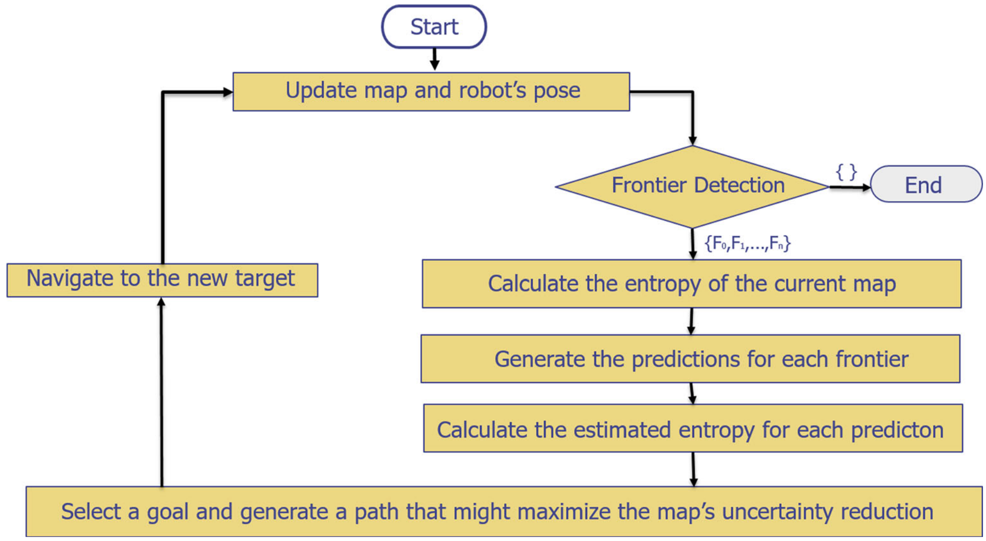


Fig. 2 Diagram that represents the exploration cycle proposed by our methodology

map caused by the robot eventually reaching each candidate frontier and collecting data. Posterior entropy is, then, estimated using these predictions.

In this work, we make predictions by extrapolating the local structure of the occupancy grid around each frontier to the area beyond it.

The key insight is that indoor environments are typically regular and can be characterized by straight walls and their intersections (e.g., corridors and rooms). Based on this observation, we consider that there are two main ways walls can be expected to propagate beyond a frontier: in a straight line, or in a straight-angle corner, as illustrated in Fig. 3.

Our predictions of what lies beyond frontiers is made using these considerations. It is important to note, however, that as the distance beyond frontiers increases, the number of possible combinations of straight segments and corners dramatically increases, and it becomes less likely that our predictions would actually resemble the true state of the environment ξ. In other words, predictions can not be reliably extended to large areas beyond frontiers.

To avoid inaccurate predictions, we use a *prediction zone* $\phi_i \subseteq \xi$ around each frontier $\mathrm{f}_i \in \mathcal{F}$. Our method extrapolates the information of a frontier only to the corresponding prediction zone. To determine the size of each ϕ_i we use the minimum-perimeter bounding box technique to build a geometric structure designed as the smallest bounding which lies within all the frontier's cells. Then we expand it to incorporate the walls that restrain the frontier. In other words, the area ϕ_i covered by the predicted propagation of wall segments is proportional to the size of the frontier f_i, as Fig. 4 illustrates.

Finally, to determine the direction in which the wall is most likely to propagate, the local area around the frontier is analyzed. First each occupied cell is classified as part of a wall, and then walls bordering unknown territory are extended. If during propagation a wall meets cells known to be free, the wall bends perpendicularly to the direction of the unknown area. Next, cells of unknown status are updated to free cells using a wave propagation method [25], and so are cells in contact with the frontier. Then, the cells most recently updated to free become the new frontier, and the

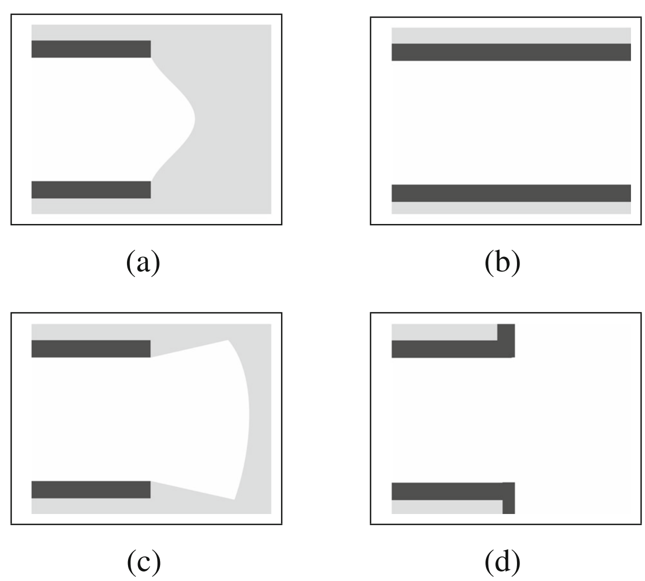


Fig. 3 Two different ways to propagate walls beyond a frontier. **a** and **c** are maps with their detected frontiers; whereas **b** and **d** are the corresponding predictions of what lies beyond those frontiers according to our assumptions about walls propagation. In **b** the walls propagate in straight lines, whereas in **d** they turn in a straight angle because they cannot propagate through a free area

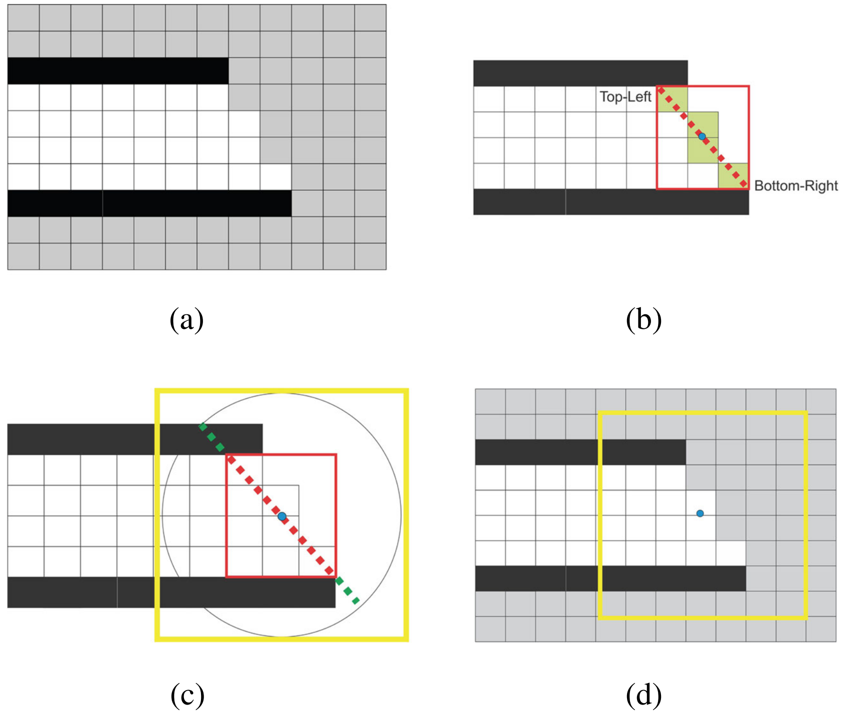


Fig. 4 Design of a prediction zone ϕ_i based on the corresponding frontier f_i's size. **a** Portion of the map with a detected frontier. **b** Creation of a minimum-perimeter bounding box by exploiting the two extremes of the frontier's cells. **c** Expansion of this box adding an extra fixed value represented by the green line. **d** Prediction zone designed for the detected frontier

process is repeated until the new frontier cells reach the limits of the prediction zone. The predictions are made with a linear reduction of the certainty to avoid big propagation in big zones. Figure 5 shows a few examples of predictions during exploration.

As it can be seen in Fig. 5, prediction zones are larger when the size of the frontiers are larger. In Fig. 5a, one wall propagates straight, hits a free cell, and turns right, while the others walls propagate in a straight line until the limit of the prediction zone. In a second step, Fig. 5b, the difference in size of the frontiers and subsequently the prediction zones can clearly be seen. In the same step, the reduction in the certainty of the prediction can also be clearly seen. Finally, in the last step, Fig. 5c, demonstrates the fact that even when there exists a cluster of frontiers in a small area, the method can still predict the behavior of the walls and select the frontier with the highest estimate information gain. This is possible because each prediction is executed separately, and one does not interfere with the other. The filling of a prediction zone corresponds to the estimation of probe Z_{t+1} on the map, which is then used to estimate posterior entropy.

Algorithm 1 describes the steps of estimating the information gain for each frontier detected on map. First, it needs to adapt the map transforming the occupied cells in line segments (line 4). To do that, it is employed the Hough Line Transform [26] with some adjustments to acquire these lines. Then, it calculates the current entropy of the map (line 6) using Eq. 2 necessary to estimate information gain of each frontier further. For each frontier, it designs the prediction zone (lines 10 and 11), detects the walls (line 13) and initiates the prediction within the prediction zone (lines 15 and 16). With the prediction completed, the posterior entropy is estimated (line 18) using the same Eq. 2 but the map already updated with the prediction. Based on the

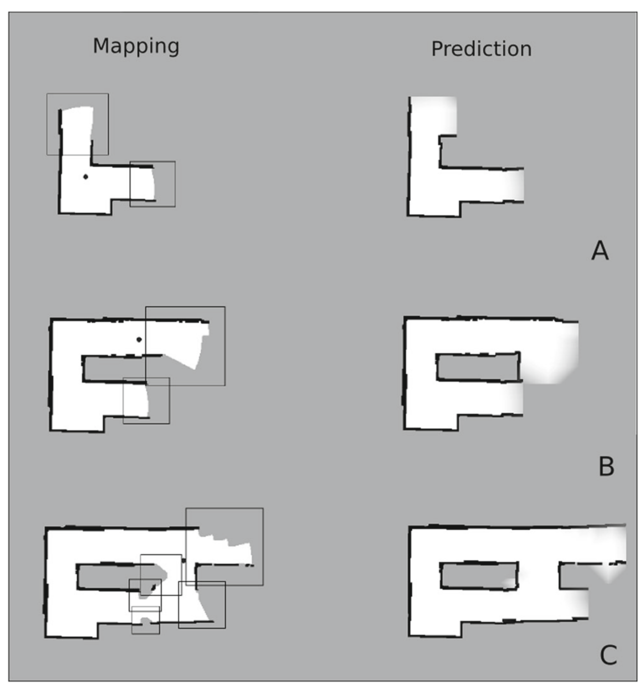


Fig. 5 Examples of predictions [7]. The left column represents the candidate map frontiers, while the right column shows the predictions based on the structure of the walls around each frontier. The squares covering the frontier indicate the extents of the prediction zones

values for both entropies, the information gain is obtained (line 20) using Eq. 1. After applying repeatedly this process for all frontiers, we obtain the estimated information gain values necessary to bias the tree expansion until it selects a new destination and generates a new path. It technique will be better explained in the next section.

Algorithm 1 CalculateIG(F, M)

1: **// Initiate IG vector for each frontier**
2: $IG \leftarrow \emptyset$
3: **// Apply Hough Line Transform** [26]
4: $S \leftarrow$ *identifyLineSegment(M)*
5: **// Calculate the current entropy of the map**
6: $e \leftarrow$ *calculateEntropy(M)* ▷ Eq. 2
7: **// For each frontier**
8: **for** $i = 0$ *to size(F)* **do**
9: **// Create a prediction zone**
10: $h_i \leftarrow$ *sizeFrontier*($F[i]$, M)
11: $l_i \leftarrow$ *createPredictionZone*($F[i]$, h_i, M)
12: **// Detect the walls connected to frontier**
13: $W \leftarrow$ *identifyWalls*($F[i]$, S, M)
14: **// Initiate the prediction propagation**
15: $\hat{M} \leftarrow$ *expandWalls*(l_i, W, M)
16: $\hat{M} \leftarrow$ *expandFreeCells*(l_i, $F[i]$, $\hat{M}$)
17: **// Calculate estimated entropy**
18: $\hat{e} \leftarrow$ *calculateEntropy*($\hat{M}$) ▷ Eq. 2
19: **// Calculate estimated information gain**
20: $ig \leftarrow$ *informationGain*(e,ê) ▷ Eq. 1
21: $IG[i] \leftarrow ig$
22: **end for**
23: **return** IG

3.4 Decision-Maker and Navigation Approach

In this section, we describe the novel combined technique to select the most relevant destinations and generate feasible paths for the robot's navigation in order to efficiently explore the environment.

After calculating the potential information gain given by each frontier at a time t, a destination needs to be selected and then explored by the robot. In the literature, a utility function is often used to guide the robots' decisions. The utility function traditionally ranks actions according to reduction on uncertainty they would cause (the more uncertainty is reduced, the higher the utility of an action), and the action with best expected utility is picked by the robot. Although such an approach can generate good results, in this work we plan the robot's destinations in a different manner. More precisely, we combine in a same algorithm both steps of choosing a destination and how to navigate to it. This novel approach allows for the selection of best destinations at the same time it creates routes free of obstacles from the robot to these destinations.

To implement our approach, we adapted the common Rapidly-Exploring Random Tree (RRT) algorithm [8] for the decision-making of the robot. It is a sample-based method and can be classified as probabilistically complete, which means that the chances of finding a path in a map increases when more effort is applied. For the RRT algorithm, if a path exists, it will be eventually found as the number of samples increases.

The original RRT algorithm, Algorithm 2, generates a path from the robot's location into a unknown area. The algorithm generates a tree T rooted at the robot's position (x_{init}) and incrementally increases this tree iteratively. At each iteration, a new configuration (x_{rand}) is sampled uniformly where $m_{ij} = 0$. A nearest configuration (x_{near}) to x_{rand} in T is selected, and it is attempted to make progress from x_{near} toward x_{rand}. Usually this entails moving x_{near} a distance in the straight line defined by x_{near} and x_{rand}. If it is collision-free, this newly generated configuration, x_{new}, is then added to the vertices of T and the edge (x_{near}, x_{new}) is added to the edges of T.

Algorithm 2 RRT(x_{init}, K, Δt) [8]

$\mathcal{T}$.init(x_{init});
for $k = 1$ to K **do**
$x_{rand} \leftarrow$ *RANDOM_State()*;
$x_{near} \leftarrow$ *NEAREST_NEIGHBOR*(x_{rand}, x_{init});
$u \leftarrow$ *SELECT_INPUT*(x_{rand}, x_{near});
$x_{new} \leftarrow$ *NEW_State*(x_{near}, u, Δt);
$\mathcal{T}$.add_vertex(x_{new});
$\mathcal{T}$.add_edge(x_{near}, x_{new}, u);
end for
return $\mathcal{T}$

The main difference between the traditional RRT and our proposed method is the function employed to generate the new points x_{rand} that might be appended to the tree T. More precisely, we introduce a decision-maker method based on the Genetic Algorithm known as Roulette Wheel Selection. Figure 6 demonstrates this sample function. For each frontier detected on map, a section of the wheel will be configured based on the estimated information gain previously calculated and normalized. We used a simple linear normalization where we could transform the data to a specific range (0 to 100%). Larger frontiers will be assigned larger sections of the wheel, and hence will have higher probability to be selected.

In our algorithm, the function receives as arguments the destinations followed by their weights (IG). The probability of selecting a specific destination is proportional with its weight. For the RRT algorithm, it is important to add some

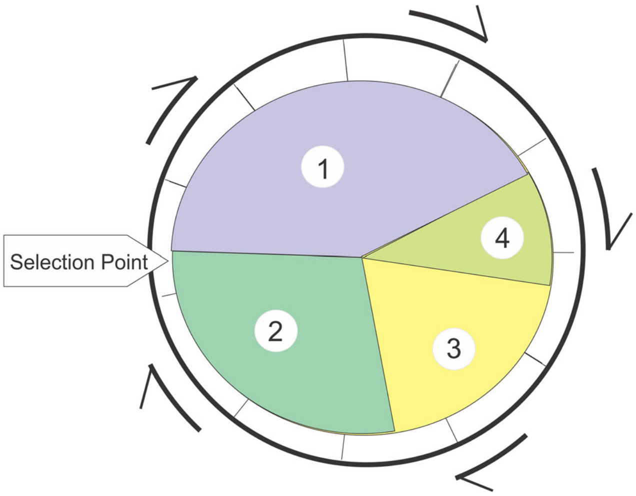


Fig. 6 Decision-maker function based on the Genetic Algorithm Roulette Wheel Selection. The candidate destinations are set as weighted sections of the wheel (Section 1 = 40%, Section 2 = 30%, Section 3 = 19%, Section 4 = 11%). The element selected will be used as bias on a RRT expansion. In this example wheel there is no section for a random destination

weight to a random destination to turn possible some obstacles to be circumvent avoiding some *Local Minima* issue. The size of the random destination section is set based on the number of candidates frontiers. If there is only 1 frontier then the size will be 25% of the wheel, if there is 5 frontiers it will be 10% of the wheel and with 10 or more frontiers it will get closer to 1% of the wheel.

The proposed approach to select the samples for the RRT algorithm is presented on Algorithm 3.

Algorithm 3 SelectSamples(F,IG)

// Set the size of the random section
$a \leftarrow randomSection(size(F))$
// Set the size of the shares section
$c \leftarrow 100 - a$
// Normalize the values (Shared Section)
$W \leftarrow normalizeValues(IG, c)$
$v \leftarrow selectRandomValue(0,100)$
// Identify sample to be used as bias on the RRT
$b \leftarrow identifySection(W, v, a)$
return b

In this way, the samples might expand the tree into regions with the highest estimated information gain because the probability of these sections being selected is higher. However, it is still possible to expand the tree to frontiers that yield lower information gain. When there is more than one frontier with a high probability to be select as a bias on our approach, the tree might expand into different directions. Consequently, frontiers close to the robot have also good chances to be reached and explored. As it can be noticed, the process to select a next best destination and the planning to create a path free of obstacles becomes as an unique process. At the same time the algorithm chooses a next best destination it completely generates a path toward it.

3.5 Environment Exploration

Algorithm 4 presents a pseudo code of the complete proposed methodology for the exploration process.

Algorithm 4 AutonomousExporation(n)

1: **// Iniciate vector: path to be navigated**
2: $C \leftarrow \emptyset$
3: **// For each cicle of time**
4: **for** timestep t **do**
5: $d \leftarrow checkDestination()$
6: $M \leftarrow updateMap()$
7: $r \leftarrow checkRobotPose()$
8: **if** $reachedGoal(r, d)$ **then**
9: $F \leftarrow identifyFrontiers(M)$
10: **if** $size(F) == 0$ **then**
11: $endExploration()$
12: **end if**
13: $IG \leftarrow \boldsymbol{CalculateIG}(F, M)$ ▹ Alg. (1)
14: $C \leftarrow \boldsymbol{RRT}(IG, r, M, n)$ ▹ Alg. (2)
15: **else**
16: navigate(C)
17: **end if**
18: **end for**

At each time step, the robot checks if the current goal has been reached (line 8). If it has, the robot calculates a next best destination (lines 9–14); otherwise the robot keeps navigating until arriving at the desired location (line 16). To calculate the new target, it is necessary to select each frontier from the map and evaluate them by the criteria previously determined (lines 9 and 13). In this work, we defined as main criteria the Information Gain, being the travel cost (distance) analyzed indirectly. Afterwards, a path will be generated selecting samples in the environment from the robot's pose until a new sample reaches an unknown area on map (line 14). Finally, when there are no more frontiers to be explored, the algorithm is interrupted (line 11).

4 Experiments

In this section, we present an experimental evaluation of the methodology in order to assess the benefits of our combined predictive exploration and path planning approach.

Consequently, we support our claim that by estimating the information gain for each candidate destination it is possible to cover an unknown environment faster than traditional approaches.

We performed a series of experiments in two simulated scenarios, an office and a cave-like environment, similar to those used by other works in the literature [24]. These environments are an interesting choice since they have different properties. The exploration framework was written in C++ using ROS (Robot Operating System) and executed on a computer running 64-bit Ubuntu 14.04 with an Intel i7 processor and 8GB of RAM.

As mentioned earlier, in this work, we mainly focus on the robot's decision-making and navigation process, leaving aside the localization and map generation problems. Therefore, we decided to use the ROS gmapping SLAM package to provide the robot's localization and to generate the maps during the experiments. This is a well-known package providing a high efficient Rao-Blackwellized particle filter to learn grid maps from laser range data [27]. In this way, it is possible to build accurate maps by setting the parameters of the sensors to minimum error.

The agent model applied on the simulation is a nonholonomic differential robot. However, it moves through the environment only in straight lines. Then, in order to change directions it needs to turn around by its own axis and then keep moving. The robot is attached with a laser sensor with a standard 5 m range and an opening of approximately 60°. The navigation of the agent uses an dynamic control system (PID) that allows the robot to move trough the path generated by the RRT algorithm. By the end of the navigation, the robot needs to turn around and face the unexplored region. Otherwise, it will not be able to update and expand the map.

4.1 Experimental Procedure

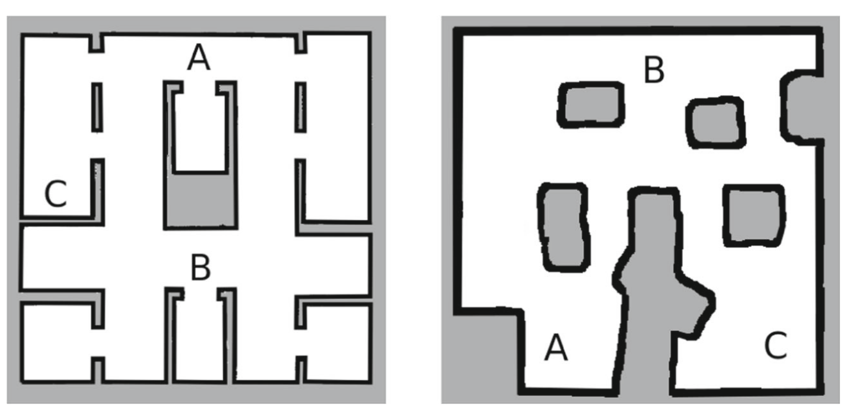
To evaluate the proposed algorithm through simulation, three trials are run in each environment with the robot's initial position being randomly chosen for each trial. Figure 7 presents the ground-truth maps, as well as the initial positions (marked as A, B and C). The left side shows an office with some rooms and corridors and the right side shows a cave with large open areas containing some rocks and columns.

To better evaluate the proposed approach, we compared the results with three different and common methods. All these techniques were used in the decision-making step about where to go. In this way, we considered methods that always reach and explore the

- **Nearest frontiers (NFE)**: based on the euclidean distance between the robot's location with the frontiers center of mass, ignoring the existence of obstacles and unknown areas;
- **Biggest frontiers (BF)**: based on the number of cells that belongs to each frontier;
- **Highest estimated information gain frontiers (HIG)**: based on the approach presented in the paper to calculate the estimated information gain using the predictions of areas beyond frontiers.

These three strategies can be classified as greedy because they are used to optimize the exploration problem always selecting the destinies that might be better than the others in terms of a factor or criteria in analysis. In this case, the criteria of the three methods are: distance, size and estimated information gain, respectively. The strategy known as NFE [1], even though it is a simple method, is one of the technique most used in the literature to compare with novel approaches because it brings interesting results.

Overall, the methodology proposed in this paper corresponds to a cycle which needs to be repeated while there are frontiers to be explored. This cycle is represented by the diagram on Fig. 2. Thus, when an experiment is initiated, the robot needs to gather some information about the environment and initiate the map's construction. When the model is updated, the robot needs to search for frontiers and then



(a) Office (b) Cave [28]

Fig. 7 Referential maps with the three initial positions for each map, represented by the letters A, B and C execute the predictions in order to calculate the estimated information gain. With these information, it will select and generate a path to a region of the map. Afterwards, the robot navigates through the path until reaches the desired destination. This whole process is repeated until the environment be completely covered.

Some common metrics, such as quality of the map, were not applied in the experiments because of the high precision of the robot's location. To asses the methodology, we used a rate of completeness by time and the total cost to transverse the environment while building the map. In order to calculate this rate, we used two maps as referential for each environment and defined the total information values of each ideal complete map.

4.2 Illustrative Example

In this section, an experiment is presented to illustrate our methodology during an exploration task. Pictures were captured in different moments in order to describe the main steps of our technique, such as frontiers detection, predictions and our adapted tree expansions.

The robot was initialized at the position represented by letter A on the Office scenario. At first instance, the robot executed a 360° observation over the environment and generated a map using all information acquired. It was detected three different frontiers on map and, consequently, three different prediction zones were designed based on the characteristics of each frontier, as it can be seen in Fig. 8.

Figure 9 presents the predictions and the values corresponding to the estimated information gains already calculated for each frontier. The propagation of the walls generated open areas inside the prediction zones with numbers 1 and 3. On the other hand, the prediction zone with number 2 propagated using straight lines preserving the local structure.

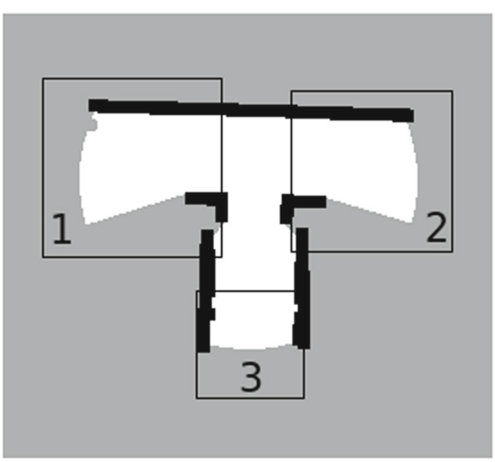
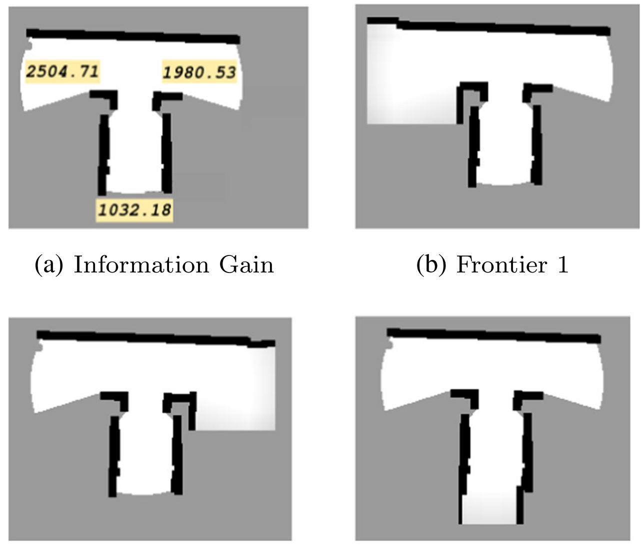


Fig. 8 Initial map containing frontiers represented by numbers 1, 2 e 3 with their own prediction zones



(a) Information Gain (b) Frontier 1

(c) Frontier 2 (d) Frontier 3

Fig. 9 In **a** the Information Gain values are shown for each frontier. Images **b**, **c** and **d** represent the prediction for frontiers number 1, 2 and 3, respectively

The last step of the exploration cycle is characterized by the RRT propagation in order to select a destination to be explored. The new path will be used during the navigation to reach the desired location. In this moment, the path was expanded with more samples in direction of the frontiers 1 and 2 until it reaches the number 1, illustrated in Fig. 10. In this way, the selected frontier will be explored first. During the tree expansion, the frontier with number 2 had only one sample in its direction even though it brings an equivalent weight based on the estimated information gain values. The reason is because a sample cannot not be generated when it is too close to obstacles. Hence, from the closest node of the tree in direction to the frontier number 2 there is no space to generate a new sample. Besides, the percentage of

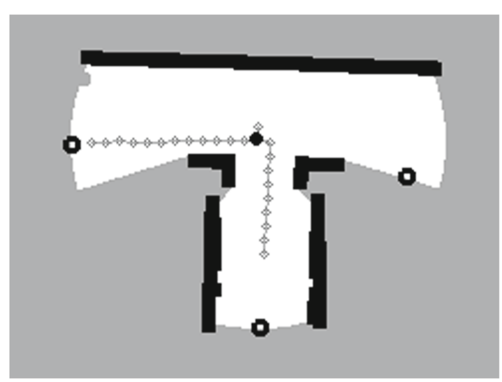


Fig. 10 Tree propagation to create an exploration route. The circles, one on each frontier, indicate the goal bias for the tree expansion. The tree is represented using small gray circles with the root in black. The final route is the only path between the robot and the Frontier 1

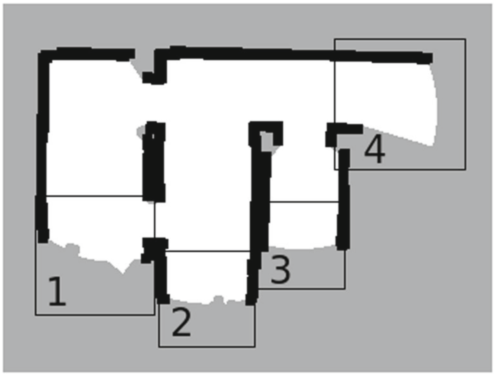


Fig. 11 Detected frontiers and prediction zones after five minutes of exploration

samples generated on random positions was low, not enough to circumnavigate the obstacle.

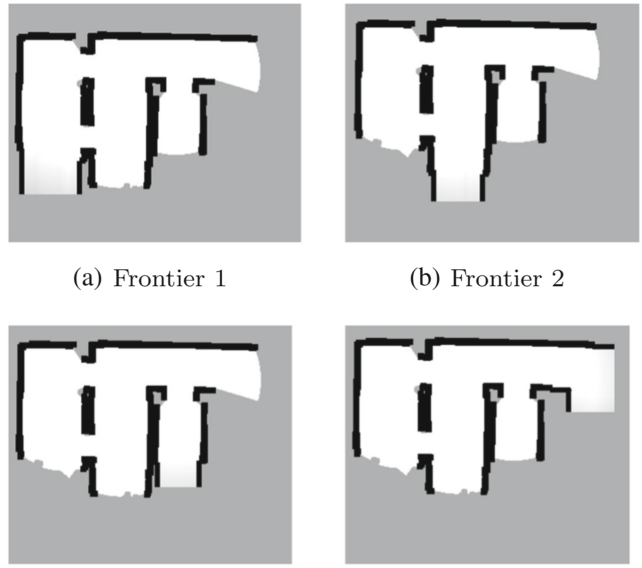
After five minutes of exploration, the robot updates the map and identifies four new frontiers. The prediction zones were projected as it can be seen on Fig. 11. After the initial position, the robot explored the left side of the map until it reaches the current position. The prediction zones that were formed have similar dimensions and characteristics.

Figure 12 shows the current map with values related to each estimated information gain obtained by the predictions shown in Fig. 13. From the highest to the lowest values, the frontiers can be ranked as follows: 4, 1, 2 and 3.

At this moment, the walls were propagated in straight lines inside the predictions zone 1, 2 and 3, as presented in Fig. 13a, b and c, respectively. Frontier 4 was propagated changing the direction in one of the walls creating an open area. It brought a higher estimated information gain compared to the others.

In order to finish another cycle, the robot needs to make a new decision about where to go next. Figure 14 shows the tree propagation, which will result in a path to a relevant destination (frontier) on the map. At this moment, the path was created in the direction of Frontier 1. Even though Frontier 4 got the highest information gain on the current map, most of the samples are blocked by obstacles when they are generated by the adapted RRT algorithm. The same thing happens for the other frontiers except by Frontier 1. As a result, the tree expanded into Frontier 1 more frequently. The tree expansion is interrupt only when a new sample is located close to the bias of a frontier being attached to the tree.

Comparing the results from the different strategies presented earlier after 15 minutes of exploration, it is possible to spot the effects of each approach on the built map. For example, the map which represents the outcome of the strategies that always explore the nearest frontiers has not much gaps with low information, as it can be seen in Fig. 15a. It is common on this strategy explore parts of the



(a) Frontier 1 (b) Frontier 2

(c) Frontier 3 (d) Frontier 4

Fig. 13 Predictions for **a** Frontier 1, **b** Frontier 2, **c** Frontier 3, and **d** Frontier 4

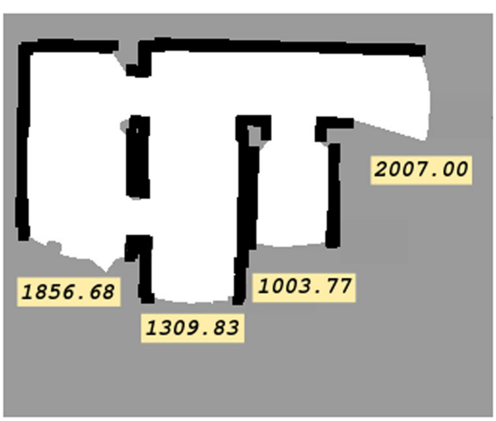


Fig. 12 The values related to each estimated information gain on current map

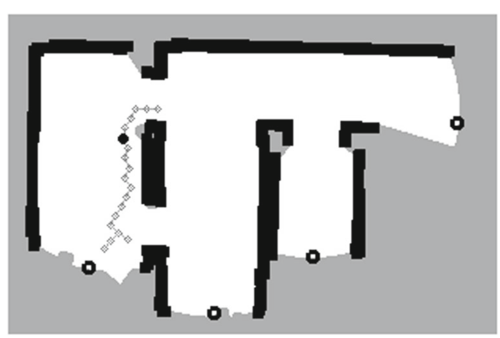


Fig. 14 Tree generated to create an exploration route. The final route will be the only path between the robot and Frontier 1

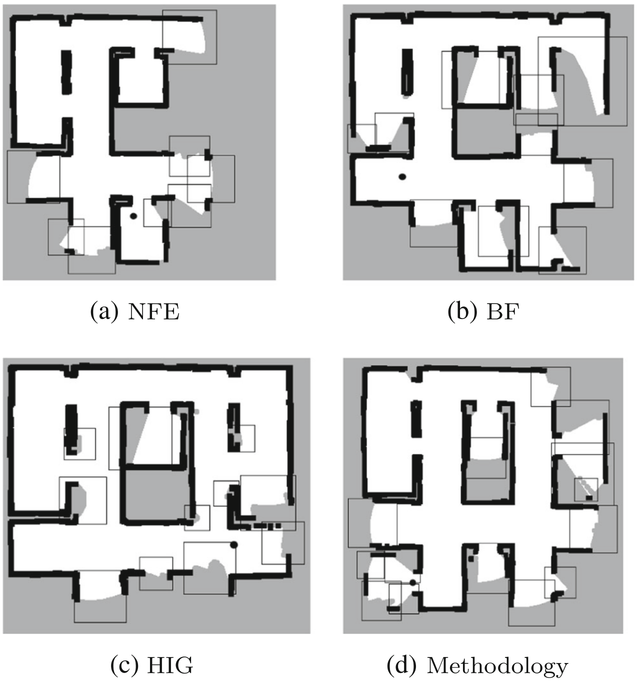


Fig. 15 Maps generated based on different strategies after 15 minutes of exploration. The small black circle indicates the robot's current position. The actual detected frontiers as also highlighted

maps completely before moving on to others which is different to the other greedy approaches. However, the robot spends so much time navigating through regions with low gain of information resulting in a low rate of map's covering. The others expand the map faster but ignores the cost to dislocate the robot, as presented in Fig. 15b and c. In this way, a lot of gaps with low information are left to be explored later and the robot tries to head to the areas with the highest information gain. Despite they have a faster expansion during the first minutes of the exploration, they start to get slow when new frontiers are not anymore detected. In this moment, the robots needs to move around places that have already been explored to fulfill the small gaps. Finally, Fig. 15c presents the map with the methodology results demonstrating that our approach tries to reach the areas with highest information gain although it does not leave small gaps around the robot behind.

After 30 minutes of exploration, the map gets closer to its final configuration. In this moment, it was detected three small frontiers as it can be seen on Fig. 16. Normally, the latest frontiers represent small gaps or regions that were complicated to be reached by the RRT method. For example, the frontier with number 1 was not easy to reach when the robot was not close and when it was there were others with higher estimated information gain to be explored.

The frontiers represented by the numbers 2 and 3 illustrate corners of a room with low information gain, as it can

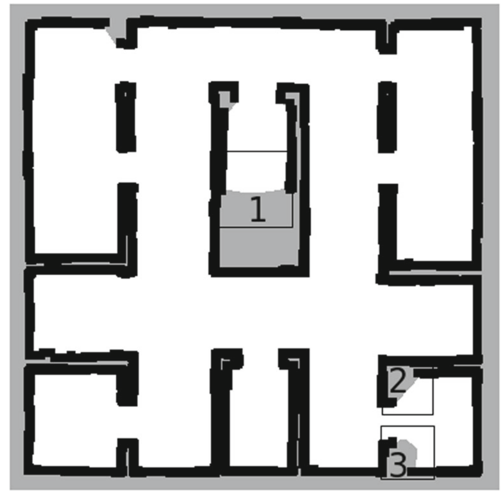


Fig. 16 Detected frontiers and prediction zones after 30 minutes of exploration

be seen on Fig. 17. These regions are common to not bring considered information for the map building. Different from the others, Frontier 1 resembles a large room or corridor. Then, comparing to the other options, it would be the best choice because of the higher estimated information gain.

Nonetheless, the robot generates a tree which identify a route to a desired destination. At this moment, it was created a small path into Frontier 3, illustrated in Fig. 18. The tree is composed only by two nodes, the root and the one close

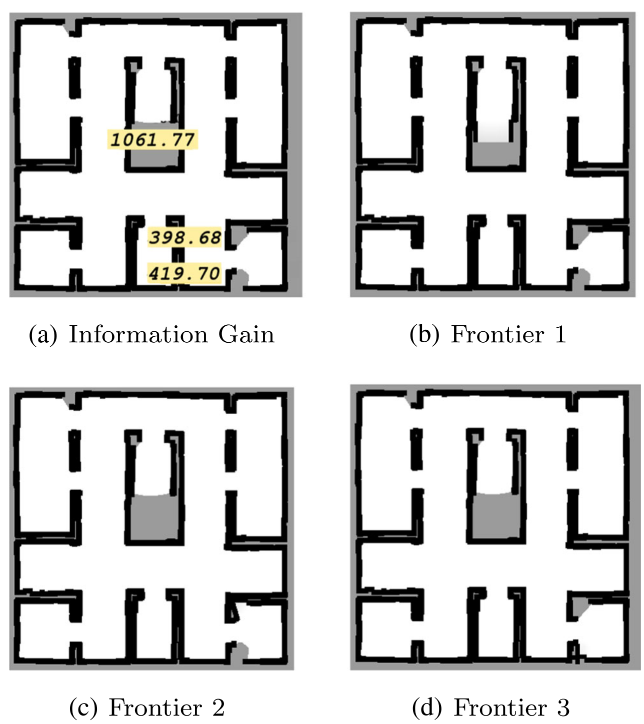


Fig. 17 **a** The values related to each estimated information gain on current map. Predictions for **b** Frontier 1, **c** Frontier 2, and **d** Frontier 3

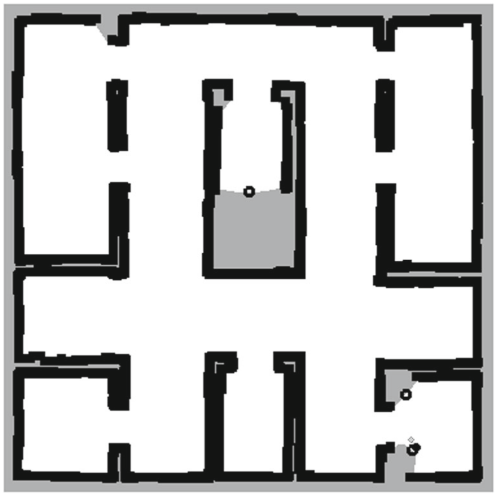


Fig. 18 Tree generated to create an exploration route. The final route will be the only path between the robot and Frontier 3

to the frontier. Because of the proximity from the robot to those frontiers, the robot decides to explore them first and then the frontier marked with number 1. Finally, when it reach and explore the last one, the exploration task will be completed.

4.3 Quantitative analysis

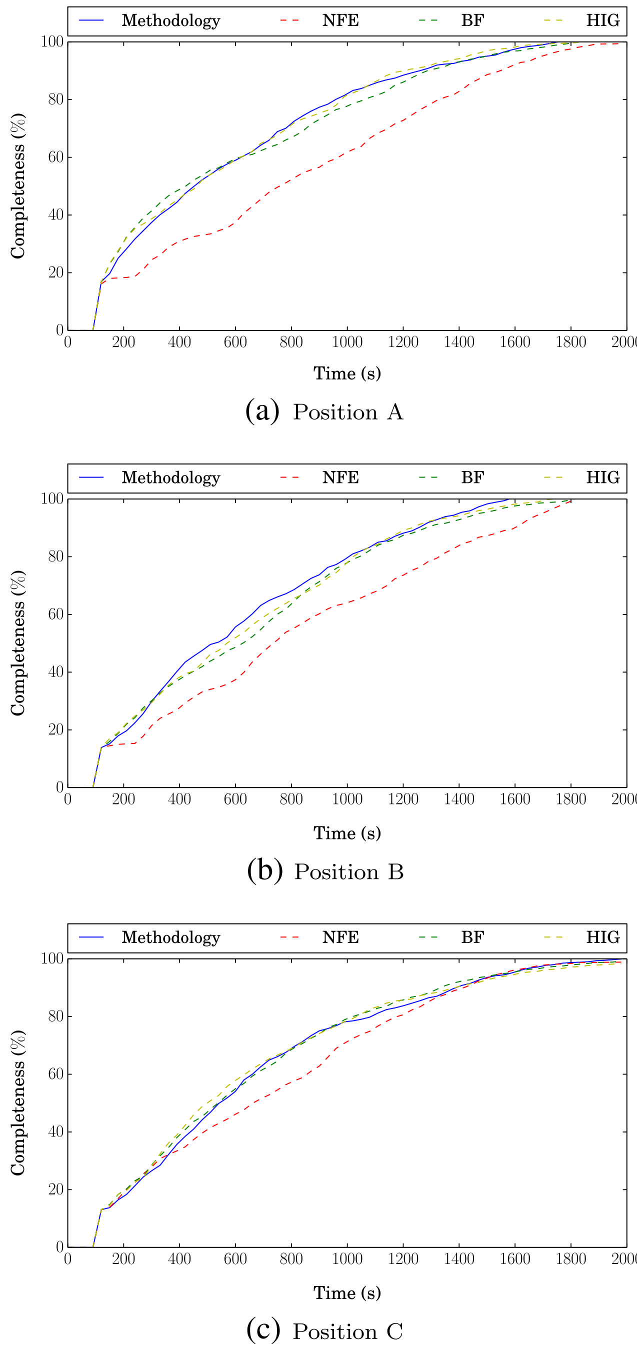
This section compares the solutions obtained by the previous discussed approaches and our methodology. It shows different evolution features for each technique applied. For each technique, 45 experiments were executed, 15 for each initial position shown on Fig. 7. Overall, each experiment has a commitment to explore autonomously the environment until it gets completely covered.

The graphics present the maps' evolution for each strategy. We used only the mean of the values to plot the results. As the dispersion level was low, we decided not to use the ranges of the standard deviation on graphics for better visualization.

4.3.1 Office

Figure 19 presents the percentage of the map covered against exploration time for the office environment considering each proposed initial position. As it can be noticed, the NFE strategy presents a slow evolution when compared to all other approaches. The main reason for that behavior is because the NFE algorithm always search for frontiers that are nearby the robot even when it does not yield much information. Hence, the robot takes too long to expand the map.

The results obtained for position B were similar to position A, however, the last initial position obtained results a little bit different. When the robot is initialized on position C, there is basically only one direction to be explored, different from positions A and B. Basically, all the experiments execute the same exploration behavior during the first three minutes. Then, different directions appears resulting on different maps expansions. The NFE continues exploring the nearest frontiers without considering the estimated

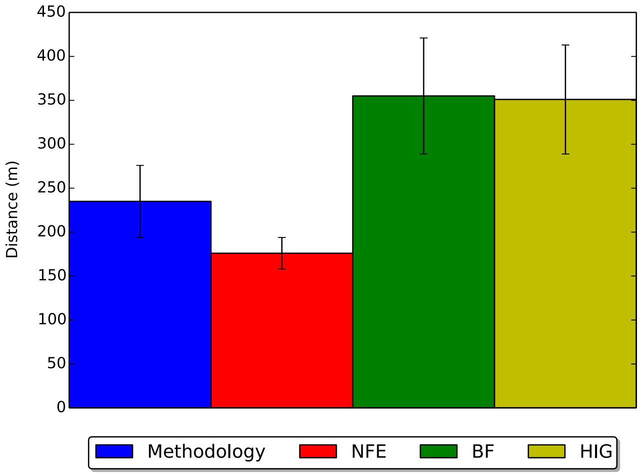


(a) Position A

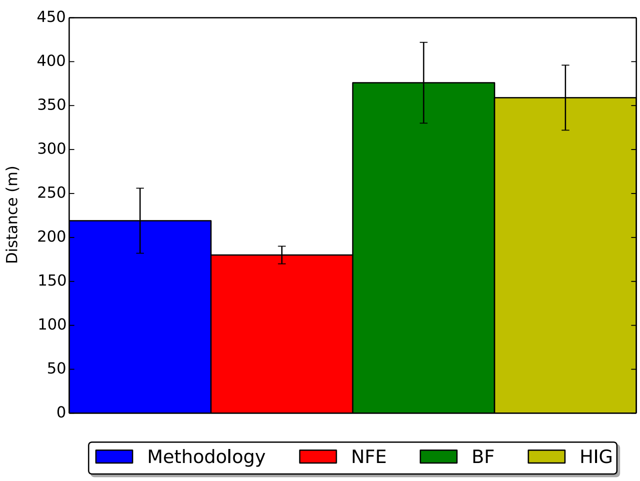
(b) Position B

(c) Position C

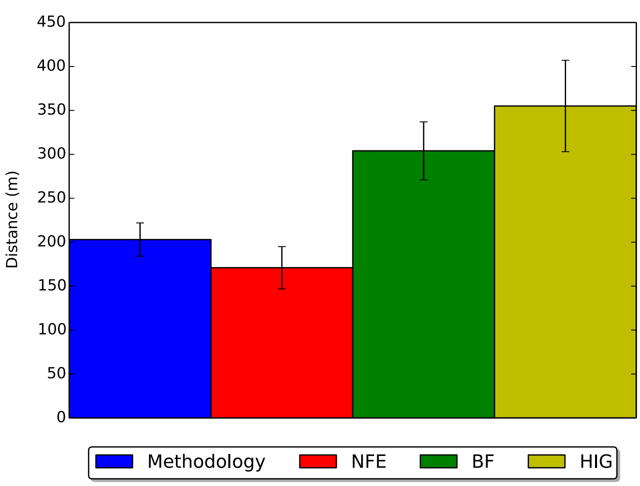
Fig. 19 Percentage of the map covered against exploration time for office environment considering different initial positions. The solid blue line represents the proposed methodology, the dashed red line the NFE approach and dashed green and yellow the other greedy approaches



(a) Position A



(b) Position B

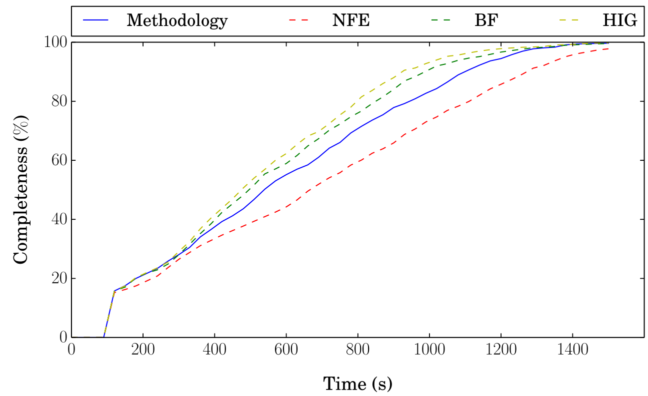


(c) Position C

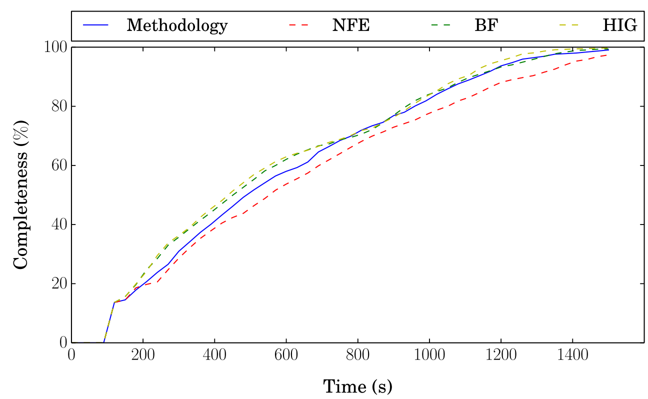
Fig. 20 Distance traveled to complete the exploration mission for office environment considering different initial positions

information gain. Consequently, the result becomes similar to position A and B.

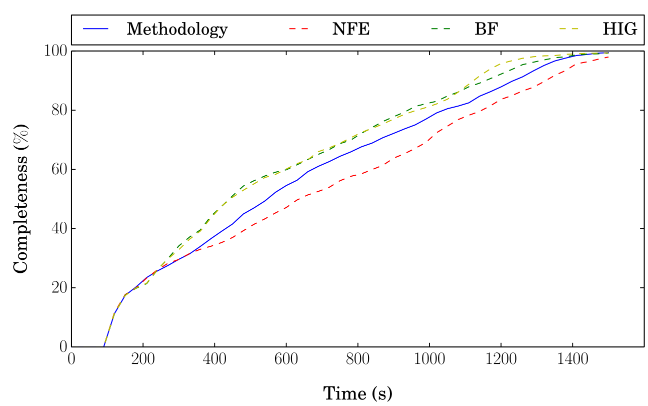
Based only on the map expansion, our method was quite similar to the other greedy strategies (BF and HIG). On the other hand, when it is analyzed the total cost to dislocate the robot through the environment, we notice that our technique



(a) Position A

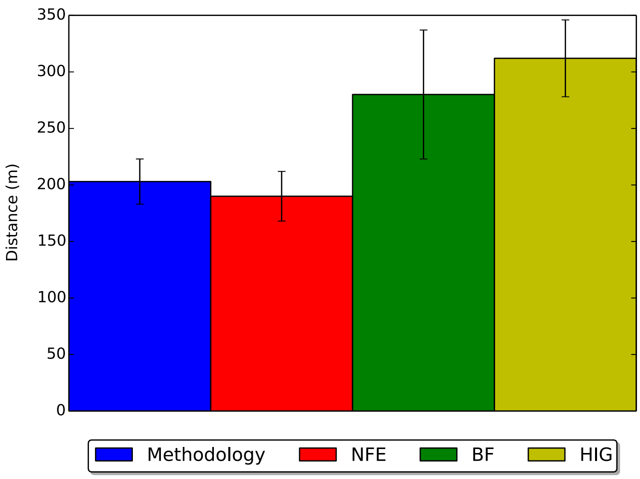


(b) Position B

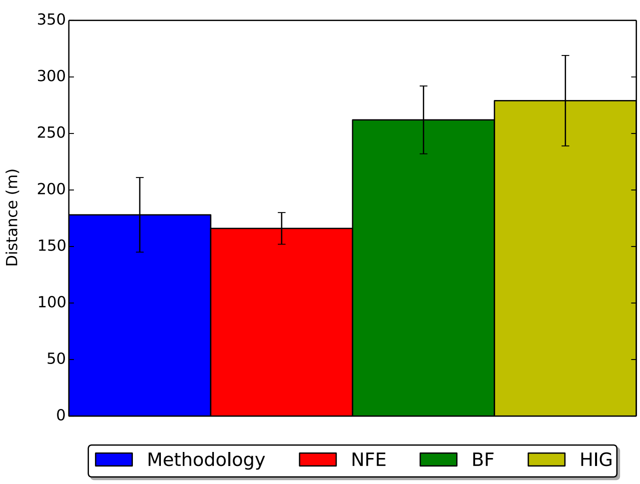


(c) Position C

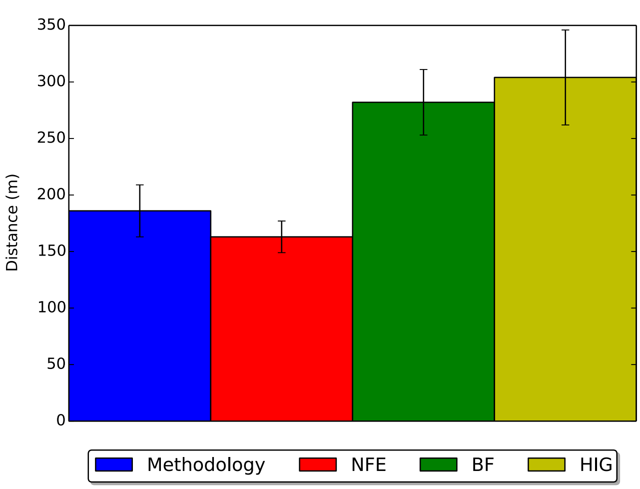
Fig. 21 Percentage of the map covered against exploration time for cave environment considering different initial positions. The solid blue line represents the proposed methodology, the dashed red line the NFE approach and dashed green and yellow the other greedy approaches



(a) Position A



(b) Position B



(c) Position C

Fig. 22 Distance traveled to complete the exploration mission for cave environment considering different initial positions

spent way less energy to explore and build the map, as it can be seen in Fig. 20. The NFE hardly navigates through areas already explored and the others do it more often. Our approach represents the balance between the NFE and the other greedy strategies.

4.3.2 Cave

Figure 21 presents the percentage of the map covered against exploration time for the office environment considering each proposed initial position.

The NFE approach obtained results slightly better on cave environment but still inferior than our methodology and the other greedy strategies. The main reason for this effect is because this new environment is characterized by large open areas. Consequently, when the robot explores frontiers around itself it still yields a significant information gain. Besides, this environment does not have much corners with low information gain to be explored. Corners are common inside more clustered environments.

Our methodology tries to reach the highest estimated information gain areas without leaving gaps with low information gain to be explored later. As the areas are bigger, these gap are easier reached by our technique. Hence, our methodology results were similar to the other greedy approaches.

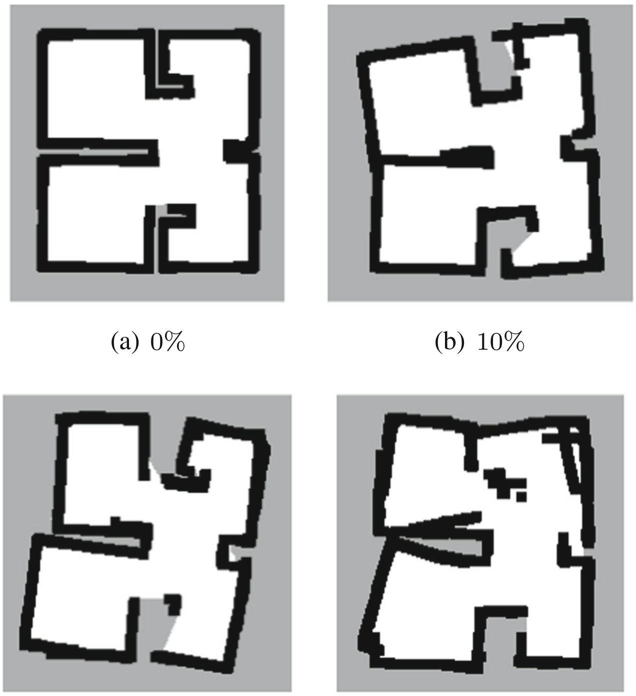
Nonetheless, we obtained similar outcomes when we analyzed the total cost to explore the environment completely, shown in Fig. 22. In this scenario, our approach got results slightly closer to the NFE. It shows that our approach explored more areas around the robot than moved to reach the highest estimated information gain regions. Our methodology tries to indirectly balance the information gain criteria with the distance cost criteria during the tree expansion for each path generated.

4.4 Noise Influence Evaluation

The algorithm used to solve the SLAM problem depends on the data collected by the sensors during the exploration in order to build a map and find the robot's location. If the sensors used on the robot are not precise, i.e., with too much noise, the map can be formed with a low accuracy state.

This section about noise influence was introduced in order to better evaluate and understand the effects of the estimated map on the robot's decision about where to go next.

There are in the literature algorithms that succesfully solve this problem by fusing the sensors' data or using other information (e.g. visual features) [29–31]. However, even new SLAM approaches do not garantee a precise map in all kind of scenarios. Figure 23 presents an example of different maps that have been built considering an odometrical information with 0%, 10% and 20% of linear and



(a) 0% (b) 10%

(c) 20% (d) 30%

Fig. 23 Maps after exploring the same environment with different error rates

angular errors. As it can be noticed, it is possible to see a deformation on maps even when the error rate is not too high.

A map with a bad quality can harm the methodology in different ways. First, frontiers that do not really exist can be inserted on map. In this way, the robot will eventually select a false frontier in order to complete the task. As a consequence it will not be able to reach the area and will increase the total cost of navigation. Second, some obstacles can be designed over free areas creating fake blockages on map. Consequently, the adapted RRT will not be able

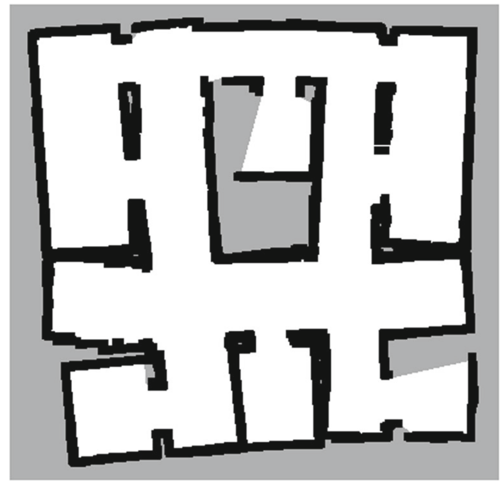


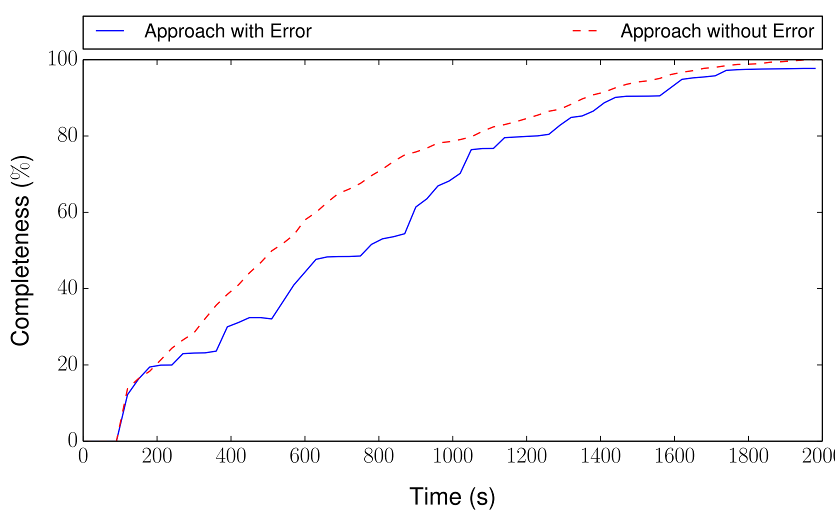
Fig. 25 Map of the Office scenario after 2000 seconds of exploration and considering 10% of error

to expand the tree for some regions of the map. Besides, with the sensors' noise, straight walls can be mapped in curves. Our methodology works properly when the walls are mapped in straight lines or in a slight curvature. Otherwise, the predictions might be generated incorrectly. Hence, in its current state, the proposed methodology depends of the proper functioning of the module responsible to solve the SLAM problem.

Figure 24 illustrates the noise influence during an exploration in the Office environment. It was considered only a 10% error on the odometry during the experiments to not distort much the map. Most of the experiments were capable to complete the task, however, the maps generated were not precise compared to the real state, e.g., Fig. 25. Due to the aforementioned problems, the execution of the exploration missions were slower then the average already presented before.

Regarding the total distance traveled to transverse the environment, the experiments showed a higher cost then the

Fig. 24 Comparison between an experiment with and without error on the odometry



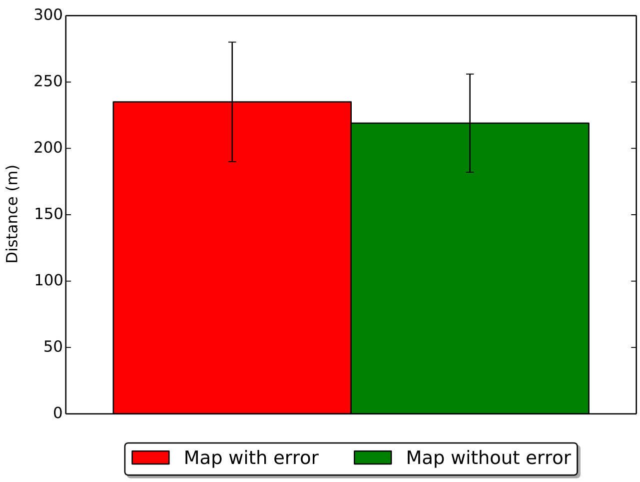


Fig. 26 Distances traveled to explore the Office scenario between the approaches with and without error on map

average distances obtained before, with total accurate maps. It is important to point that some experiments were interrupted after 2500 seconds of exploration. This is due to some of them were not capable to complete the task, as consequence of false frontiers and unreachable areas. Figure 26 compares both outcomes. Even though it shows only a slightly increase, it is enough to notice that the errors on map affects directly the total cost to explore the environments.

5 Conclusion and Future Work

In this paper, we proposed a novel solution for autonomous exploration tasks in situation where the time to create a map is paramount. The strategy combines prediction techniques with an information-theoretic approach in order to enhance rapid environmental coverage. In our approach, we exploit some common indoor characteristics to make predictions possible. This assumption allows the robot to estimate the potential information gain for each candidate frontier and make better decisions about where to go next. Then we proposed an adaption on the RRT technique to select and generate the path to best destinations to be explored. The methodology provides a good balance among prioritizing candidate frontiers with highly expected information gain and candidate frontiers that are closer to the current position of the robot.

The combination of the decision-maker and navigation processes brought to our methodology a reduction of the exploration cost. When the robot was using a greedy algorithm to select destinations to areas with the highest estimated information gain, it resulted in constantly revisiting regions already explored in order to reach the goals. In this way, the map could be rapdly expanded but with a high cost to dislocate the robot throught the environment. Our new contribution made possible to expand the map more efficiently.

In simulation we evaluated the performance of our algorithm using two main metrics: completeness by time and distance traveled. The results of the proposed strategy illustrated a better performance compared to the Nearest-Frontier Exploration strategy and other greedy approach. As a result, with the proposed approach it becomes possible to obtain a more complete map in a shorter amount of time.

The experiments show that our approach may fail if the map generated by SLAM algorithm is not accurate. The reasons for that are because the walls start to create unreal curves and areas with no obstacles becomes free or it turns free with real obstacles within. In this way, it is important to use good sensors with an SLAM algorithm that brings accurate maps.

The work has several future directions that can be pursued in order to improve the proposed methodology, which includes the extension of the proposed methodology to multi-robot ensembles. With a bigger number of agents exploring different regions by the same time, it will become faster to cover the environment completely. Besides, we intend to create a module that adapts the walls propagation based on the local areas, making the it possible even when it is not a straight wall. In the same way, we expect to develop an adaptive prediction zone that would analyze the old robot's observations to set the best shape and size. Consequently, it might better select the destination and faster reduce the map's uncertainty.

Acknowledgements This work was developed with support of Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq), Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) and Fundação de Amparo à Pesquisa do Estado de Minas Gerais (FAPEMIG).

References

1. Yamauchi, B.: A frontier-based approach for autonomous exploration. In: IEEE International Symposium on Computational Intelligence in Robotics and Automation, 1997. CIRA'97., Proceedings., 1997, pp. 146–151 (1997)
2. Jadidi, M.G., Miró, J.V., Valencia, R., Andrade-Cetto, J.: Exploration on continuous gaussian process frontier maps. In: 2014 IEEE International Conference on Robotics and Automation (ICRA), pp. 6077–6082 (2014)
3. Vallvé, J., Andrade-Cetto, J.: Active pose SLAM with RRT*. In: 2015 IEEE International Conference on Robotics and Automation (ICRA), pp. 2167–2173 (2015)
4. Amigoni, F., Caglioti, V.: An information-based exploration strategy for environment mapping with mobile robots. Robot. Auton. Syst. **58**(5), 684–699 (2010)
5. Shannon, C.E.: A mathematical theory of communication. Bell Syst. Tech. J. **27**(3), 379–423 (1948)
6. Cover, T.M., Thomas, J.A.: Elements of Information theory, 2nd edn. Wiley (2006)

7. Pimentel, J.M., Macharet, D.G., Campos, M.F.M.: Information-theoretic frontier selection for environment exploration. In: 13th Latin American Robotics Symposium and 4th Brazilian Symposium on Robotics (LARS-SBR) (2016)
8. Lavalle, S.M.: Rapidly-exploring random trees: a new tool for path planning. Computer Science Department, Iowa State University, Tech. Rep. (1998)
9. Freda, L., Oriolo, G.: Frontier-based probabilistic strategies for sensor-based exploration. In: Proceedings of the IEEE International Conference on Robotics and Automation, pp. 3881–3887 (2005)
10. Schmidt, D., Luksch, T., Wettach, J., Berns, K.: Autonomous behavior-based exploration of office environments. In: ICINCO-RA, pp. 235–240 (2006)
11. Colares, R.G., Chaimowicz, L.: Information window: uma nova abordagem para exploração de fronteiras. Anais do XX Congresso Brasileiro de automática (2014)
12. Jadidi, M.G., Miro, J.V., Dissanayake, G.: Mutual information-based exploration on continuous occupancy maps. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2015, pp. 6086–6092 (2015)
13. González-Baños, H.H., Latombe, J.: Navigation strategies for exploring indoor environments. I. J. Robotic Res. **21**(10-11), 829–848 (2002)
14. Surmann, H., Nüchter, A., Hertzberg, J.: An autonomous mobile robot with a 3D laser range finder for 3D exploration and digitalization of indoor environments. Robot. Auton. Syst. **45**(3–4), 181–198 (2003)
15. Charrow, B., Liu, S., Kumar, V., Michael, N.: Information-theoretic mapping using Cauchy-Schwarz quadratic mutual information. In: 2015 IEEE International Conference on Robotics and Automation (ICRA), pp. 4791–4798 (2015)
16. Chang, H.J., Lee, C.S.G., Lu, Y.H., Hu, Y.C.: P-SLAM: simultaneous localization and mapping with environmental-structure prediction. IEEE Trans. Robot. **23**(2), 281–293 (2007)
17. Ström, D.P., Nenci, F., Stachniss, C.: Predictive exploration considering previously mapped environments. In: IEEE International Conference on Robotics and Automation (ICRA), pp. 2761–2766 (2015)
18. Oßwald, S., Bennewitz, M., Burgard, W., Stachniss, C.: Speeding-up robot exploration by exploiting background information. IEEE Robotics and Automation Letters **1**(2), 716–723 (2016)
19. Stachniss, C., Martínez Mozos, Ó., Burgard, W.: Efficient exploration of unknown indoor environments using a team of mobile robots. Ann. Math. Artif. Intell. **52**(2), 205–227 (2009)
20. Wurm, K.M., Stachniss, C., Burgard, W.: Coordinated multi-robot exploration using a segmentation of the environment. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 1160–1165 (2008)
21. Juliá, M., Gil, A., Reinoso, O.: A comparison of path planning strategies for autonomous exploration and mapping of unknown environments. Auton. Robot. **33**(4), 427–444 (2012)
22. Moravec, H., Elfes, A.: High resolution maps from wide angle sonar. In: International Conference on Robotics and Automation, pp. 116–121 (1985)
23. Grisetti, G., Stachniss, C., Burgard, W.: Improved techniques for grid mapping with rao-blackwellized particle filters. IEEE Trans. Robot. **23**(1), 34–46 (2007)
24. Carrillo, H., Dames, P., Kumar, V., Castellanos, J.A.: Autonomous robotic exploration using occupancy grid maps and graph SLAM based on Shannon and Rényi entropy. In: 2015 IEEE International Conference on Robotics and Automation (ICRA), pp. 487–494 (2015)
25. Jarvis, R.A.: Collision-free trajectory planning using distance transforms. In: Proceedings of the National Conference and Exhibition on Robotics (1984)
26. Hough, P.V.C.: Method and means for recognizing complex patterns, US Patent 3,069,654. [online]. Available: https://www.google.com/patents/US3069654 (1962)
27. Grisetti, G., Stachniss, C., Burgard, W.: Improved techniques for grid mapping with Rao-Blackwellized particle filters. IEEE Trans. Robot. **23**(1), 34–46 (2007)
28. Howard, A., Roy, N.: The robotics data set repository (radish). [Online]. Available: http://radish.sourceforge.net/ (2003)
29. Liang, X., Chen, H., Li, Y., Liu, Y.: Visual laser-slam in large-scale indoor environments. In: 2016 IEEE International Conference on Robotics and Biomimetics (ROBIO), pp. 19–24 (2016)
30. Kadir, H.A., Arshad, M.R.: Improved simultaneous localization and mapping (slam) algorithms for aerial vehicle under dynamic sea surface environment. In: 2016 IEEE International Conference on Underwater System Technology: Theory and Applications (USYS), pp. 61–66 (2016)
31. Kumar, S.R., Ramkumar, K., Srinivasan, S.: Map spread factor based confidence weighted average technique for adaptive slam with unknown sensor model and noise covariance. In: 2016 International Conference on Robotics: Current Trends and Future Challenges (RCTFC), pp. 1–6 (2016)

Jhielson M. Pimentel is a Research Engineer at the Institute for Systems and Robotics of IST, University of Lisbon, Portugal. He received his M.Sc. in Computer Science from Universidade Federal de Minas Gerais, Brazil in 2016. During his Master's program, he was with the Computer Vision and Robotics Laboratory (VeRLab). Currently, he is with the Laboratory for Robotics and Engineering Systems (LARSys), and his main research interests are in mobile robotics, path planning, multi-robots systems and decision making.

Mário S. Alvim is an Assistant Professor at the Department of Computer Science (DCC) of Universidade Federal de Minas Gerais (UFMG), Brazil. He received his M.Sc. degree in Computer Science from the same university in 2008, and his Ph.D. in Computer Science from École Polytechnique, France, in 2011. In 2012 and 2013 he was a post-doc research fellow at the Mathematics Department of the University of Pennsylvania. He is with the Laboratory of Information Security, Cryptography, Privacy, and Transparency (Inscrypt), and his main research interests are in formal methods for security, focusing in particular on quantitative information flow and privacy.

Mario F. M. Campos Ph.D., is a Professor of Computer Vision and Robotics in the Department of Computer Science at the Federal University of Minas Gerais (UFMG), Belo Horizonte, Brazil. He holds B.S. degrees in Engineering, and M.S. in Computer Science, all from UFMG, and a Ph.D. in Computer and Information Science from the University of Pennsylvania, USA. His research interests include cooperative robotics, robot vision, sensor information processing, in which areas he has published over 250 papers in both qualified journals and conferences. He has supervised over 60 thesis and dissertations, and his main contributions are in haptics, multirobot cooperation, aerial robotics and robot vision. He is the founder of the Computer Vision and Robotics Lab – VeRLab, UFMG, Brazil. He has close collaboration with researchers from US and Europe sponsored by Brazilian and foreign agencies, and has been the PI and participated in several R&D projects with industry partners such as Petrobras, Vale, Engetron, LG Electronics, ABC Bull, and CEMIG. He has been a Distinguished Lecturer in the IEEE Robotics and Automation Society, and is currently the Vice-Provost for Administration of the Universidade Federal de Minas Gerais, Brazil.

Douglas G. Macharet is currently an Assistant Professor at the Department of Computer Science (DCC) at the Federal University of Minas Gerais (UFMG). He received M.Sc. and D.Sc. degrees in Computer Science from the same university in 2009 and 2013, respectively. He is with the Computer Vision and Robotics Lab (VeRLab), and he has published papers in major conferences and journals such as ICRA, IROS, GECCO, Computers & Operations Research, and Journal of Intelligent & Robotic Systems. His main research interests include mobile robotics, focusing on localization, mapping, path planning for nonholonomic robots, multi-robots systems and human-robot interaction.