Chapter

Extending the Limits of the Random Exploration Graph for Efficient Autonomous Exploration in Unknown Environments

Alfredo Toriz Palacios and Abraham Sánchez López

Abstract

The autonomous construction of environment maps using mobile robots is a fundamental problem of robotics; this is because virtually all tasks performed by robots need a representation of the working environment to operate. Although many works have addressed this problem known as SLAM, it still remains open; since most of the solutions do not consider a planner that allows the robot to explore autonomously the working environment or the works that consider it, they have developed slow algorithms that do not guarantee a total coverage of the environment or an optimal development of the exploration, which may result in maps of poor quality or definitely not usable given this lack of information. Thus, this work presents a new exploration method based on the random exploration graph (REG), which, unlike its predecessor, defines a systematic analysis of the next positions to be explored eliminating randomness in decision-making and thus minimizing the amount of movements that the robot must make to reach them and the time required to achieve total coverage of the environment. Additionally, a series of tests carried out on the proposed method are presented, and the results obtained in classical variables such as time and distance allow to validate the efficiency of our approach.

Keywords: SLAM, integrated exploration, path planning, unknown environments, random exploration graph

1. Introduction

Path planning is a well-known topic in the area of robotics whose main objective is to determine the best way for a robot to navigate autonomously in a work environment. Although many areas of robotics have benefited from research in this field, one of the most recent is its application to the problem of autonomous construction of environment maps, also known as integrated exploration or active SLAM, where the basic principle of operation consists of a mobile robot that must move through an unknown environment while constructing an environment map of it.

In this context, one of the first contributions can be traced to the work of Feder et al. [1], in which the authors describe an adaptive trajectory planning technique applied to the SLAM problem, where through the minimization of the inverse of the error covariance as an objective function, the next position is determined where the robot must move with the intention of maximizing the information obtained, while it simultaneously localizes and constructs the environment map.

Commonly, the development of SLAM path planners requires dynamic and agile algorithms that can be adapted to new operating conditions in environments when obstacles are detected; with this in mind, many proposals have been developed by various researchers [2–10], being one of the most popular the sensor-based random tree (SRT) presented by Oriolo, Freda, and Franchi in [11]. This method is based on the random generation of robot configurations within a local security area detected by the robot's sensors, from which a compact tree-type data structure is constructed, which represents the road map of the area explored. In this structure, the leaves represent a previously reached robot position and their respective representation of the environment segment detected by the onboard sensors in that position called local safety region (LSR).

The SRT method randomly selects free borders detected at the current position of the robot where he can continue the exploration task; in case it is not possible to find one, the robot will automatically go to its parent node to look for new areas with exploration possibility. The process ends when the backspace behavior leads the robot to the root of the tree.

However, despite the popularity of the SRT scheme, it has certain problems that should be considered. The first of them lies in the ignorance of the state of the structure that is being built, where it is not possible to know if the nodes of the structure left behind contain more areas available for exploration, and therefore the total coverage of the environment cannot be guaranteed. The second problem depends on the first one, since not knowing which areas of the environment you remain unexplored, it is necessary for the robot to go back to parent nodes to find out if it is possible to continue exploring, which causes the structure to be traveled twice, and consequently the exploration time is very high.

From the above, a new method based on the SRT is developed by Franchi and others [4] for the multirobot case known as the sensor-based random graph (SRG). This method transforms the tree structure generated by the SRT method into an exploration network when the robot finds a safe way to travel between two nodes. In this method, a probability proportional to the length of the arc of the free edges that are in the node in which the robot is located is used to determine which will be the next position to explore; in addition, the way to verify the structure to establish the way to revisit zones already explored to continue the exploration is carried out by means of the generation of a tree of minimum expansion with all the adjacent nodes of the network, choosing that of the adjacent node with the greater weight with respect to the length of the free limits of the frontiers.

The SRG method presents similar problems to those of the SRT method: although the data structure is transformed into an exploration graph, the structure is not fully exploited to make exploration more efficient, because the way of revisiting nodes to verify unexplored areas creates a tree structure, which generates a discontinuous path that forces the robot to go through the parent nodes, ignoring the versatility of the graph. In fact, if the number of adjacent nodes and the number of nodes that conforms the environment are too large, the time to complete the exploration is increased. Also, like the SRT method, the robot decides the next position to explore without considering that the randomness of the selection causes too many orientation changes, which directly affects the odometric system.

More recently, Toriz et al. [12] presented a new approach known as the random exploration graph (REG), which optimizes map coverage in the exploration process. This method is based on the working principle of the SRT method and adapts it to build an exploration graph structure. Although this method has a probabilistic nature that can cause an excess of movements in the robot to complete its task and increase the exploration time, one of its main advantages is the accumulation of knowledge acquired through the concept of border control, which stores information about areas that the robot left behind in the exploration process and that needs to be revisited to complete the exploration. This feature, plus the generated graph structure, allows an optimal return to unexplored areas to complete the exploration.

As it can be observed, the methods presented here maintain a random character to define the next position to explore, the problems found in these algorithms are the excess of time required to complete the task, and in some cases the uncertainty on the total coverage of the exploration area, which can have repercussions on partially constructed or low-quality maps, the reason why an integrated exploration strategy created from these methods would not be viable.

Thus, this work presents an approach to the problem of path planning of unknown environments based on the basic principles of the REG method; however, unlike this method, our proposal eliminates the randomness of the choice of the next frontiers to explore and, instead, relies on an analysis of the best frontier whose choice criterion is based on minimizing the amount of movements the robot has to make to reach it and maximizing the amount of information from the environment that will be acquired.

2. Extended random exploration graph

The exploration strategy presented in this research is a modified version of the REG algorithm, where the main difference lies in the way in which the robot will plan the exploration trajectory by performing a deterministic analysis of the next position to be explored; the algorithm is shown in **Figure 1**.

```
EXTENDED_REG_EXPLORATION

Node_Ag = 0
q_curr = q_init
L_Nodes_Ex = Null
S_curr ← PERCEPTION(q_curr)
for k=1 to k_max
    S_curr ← CURR_INTERSECTION(S_curr, q_curr, L_Nodes_Ex)
    F ← FRONTIERS(S_curr)
    if  F ≠ Null
        Node_Ag= Node_Ag+1
        (F_det, θ_det) ← FRONT_DET(F)
        F ← REMOVE(F , F_det)
        q_dest ← DISPLACE(q_curr , θ_det, α , r)
        MOVE_TO(q_curr, q_dest)
        S_dest ← PERCEPTION(q_dest )
        F_det ← VERIFICATION( F_det , S_dest)
        F ← F ∪ F_det
        if  F ≠ Null
            L_Nodes_Ex = L_Nodes_Ex ∪ Node_Ag
        end
        G ← ADD(G, Node_Ag , q_curr , S, F)
        q_curr = q_dest
        S_curr = S_dest
    else
        (P, Ind_Node) ← FIND_PATH(q_curr, L_Nodes_Ex)
        for i = 1 to length(P)
            MOVE_TO( q_curr , P(i) )
            q_curr ← P(i)
        end
        L_Nodes_Ex ← REMOVE(L_Nodes_Ex , Ind_Node)
        end
end
Return ( G )
```

Figure 1.
Extended REG algorithm.

The initial node considered in the algorithm will be the starting and finishing node, and as in the rest of the exploration structure, it will contain a position reached by the robot (in this case it will be the initial q_{init} position), as well as a representation of the environment surrounding it known as the local security region (LSR) where the robot will be able to navigate without the risk of colliding with any obstacle. With this node created, the cycle controls the exploration process.

Next, in each iteration k of the algorithm, the frontiers of the nodes adjacent to the current node are evaluated with the intention of verifying which free frontier segments with possibility of exploration of these are of the current LSR. The nodes that present positive intersections in this evaluation will be updated eliminating the free frontier segments of both the neighboring node and the current node, with the intention of not considering these frontiers in a possible return of the robot to continue with the exploration. In addition, the verification of intersections between nodes is used to modify the structure of the exploration graph by adding edges between nonadjacent nodes as long as there are safe roads to travel between them (see **Figure 2**). The described analysis is performed by the function CURR_INTERSECTION.

After the analysis of the frontiers of neighboring nodes covered by the new LSR and the modification of the exploration structure with new edges, the next step is to identify the remaining free frontiers F of the current position, which is performed by the FRONTIERS function. For each of the frontiers found, if they exist, an approximation point will be determined, which will serve to prioritize the free frontiers,

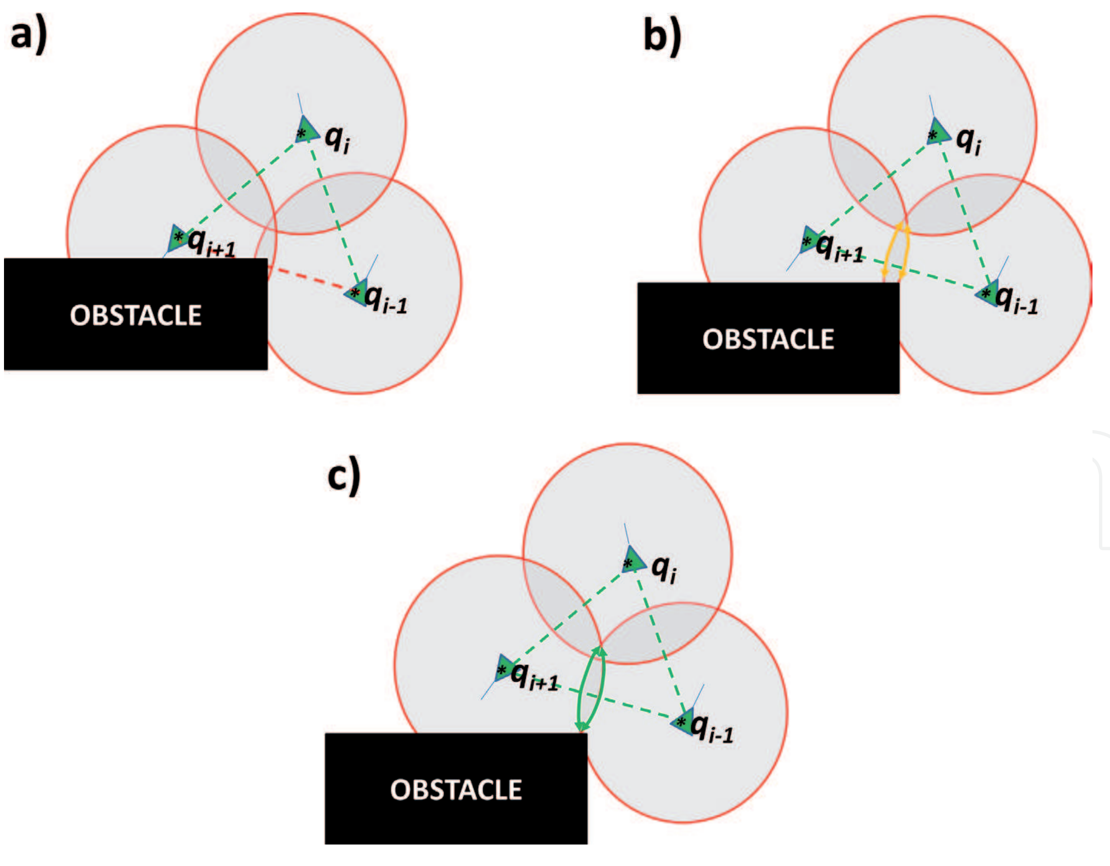

Figure 2.
Modification of the exploration graph structure through the insertion of new edges between nonadjacent nodes. (a) The insertion of the edge between q_{i-1} and q_{i+1} nodes is not possible (dotted red line) since there is no safe path between the nodes. (b) The insertion of the edge between the q_{i-1} and q_{i+1} nodes is not possible (dotted red line) since, although there is a collision-free path between the nodes, the intersection of the LSRs does not have enough space to navigate safely between the nodes. (c) The insertion of the edge between the q_{i-1} and q_{i+1} nodes is possible (dotted green line) since there is a collision-free path and enough space at the intersection of the LSRs to navigate safely.

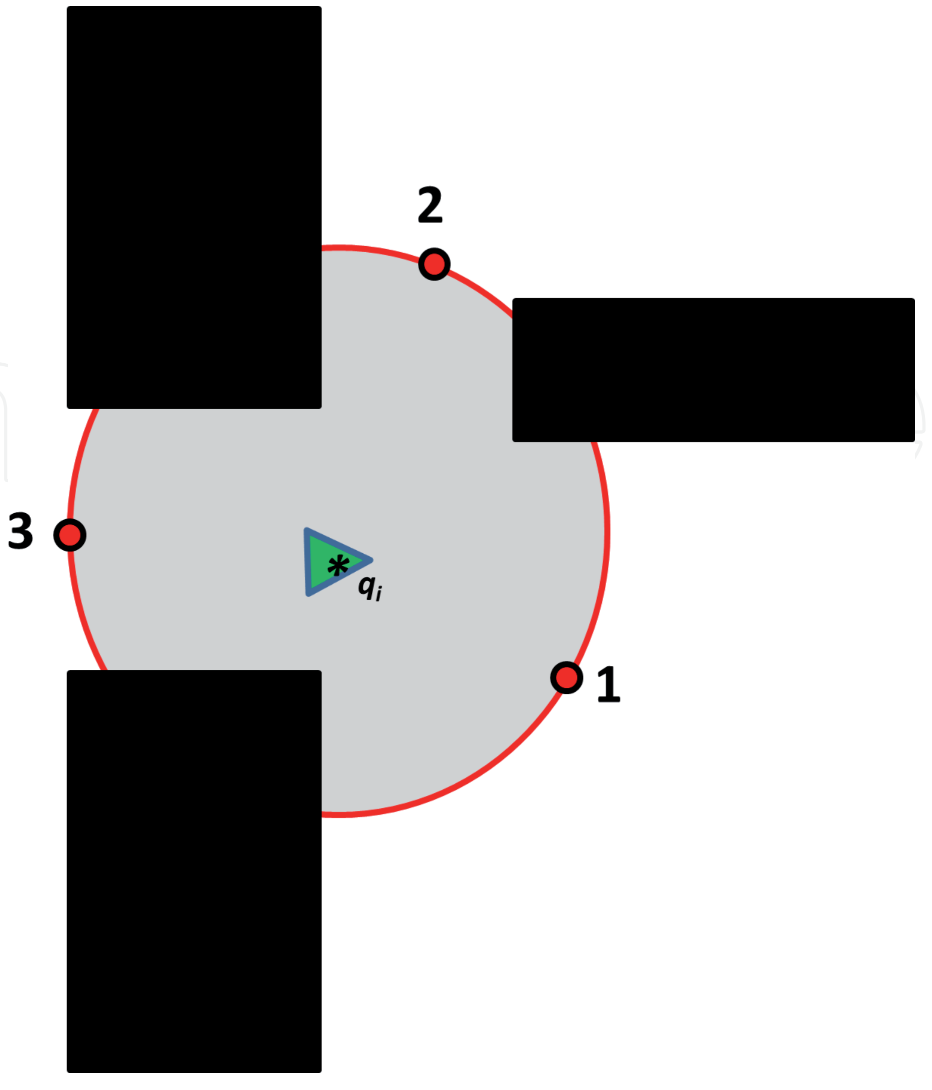

Figure 3.
Hierarchy of frontiers.

ranking them according to the effort required to reach them (FRONT_DET function) and choosing a new frontier to explore which has the highest hierarchy.

The approximation point is defined as the midpoint of the arc segment formed by the frontiers, if they can be covered in their entirety by the threshold defined by the LSR area (see **Figure 3**).

In the case that the criterion of choice of approximation point is not met, it will be redefined by taking the midpoint of the arc length proportional to the area that can be covered by the LSR, taken from the initial end of the border. With this new point chosen to continue the exploration, the frontier or segment of it, as the case may be, is removed from the group of free borders found by the REMOVE function (see **Figure 4**).

With the new frontier to explore chosen and the approximation point of it defined, the DISPLACE function will obtain the new q_{dest} position to visit to continue the exploration. This is done by taking a step of dimension $\alpha * r$ in the direction of the border approximation point, where the parameter α represents the defined radius of the LSR and the value $r < 1$ will guarantee that the new position will remain within it. Once the q_{dest} position is obtained, the MOVE_TO function will plan the path and take the robot to this new position.

In the q_{dest} position, the robot will calculate the surrounding space S_{dest} of this position (PERCEPTION function) and the VERIFIES function will determine precisely which is the real portion of the free frontier that was covered by this new LSR. In case the chosen frontier has not been fully covered, the remaining portion

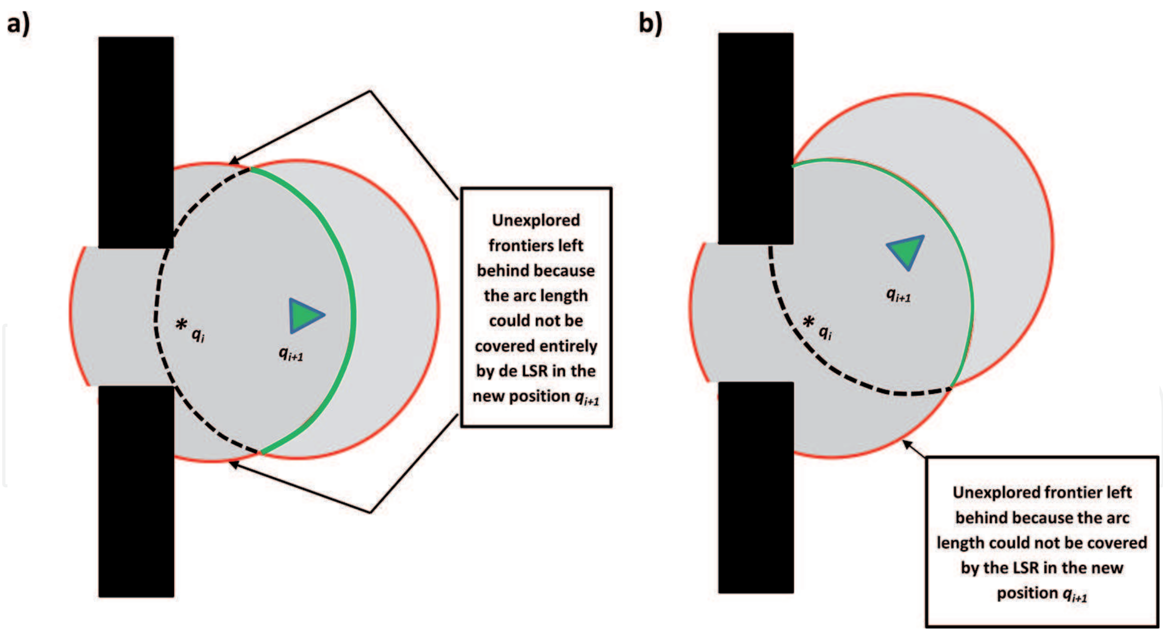

Figure 4.
Criterion of approximation to the new frontier to explore. (a) The robot makes an incorrect approach to the new frontier, since the new LSR of the chosen position leaves two free frontiers in opposite directions. (b) Correct choice of the next position to explore, since, although the new RSL is unable to cover the border completely, no more than one free border is left.

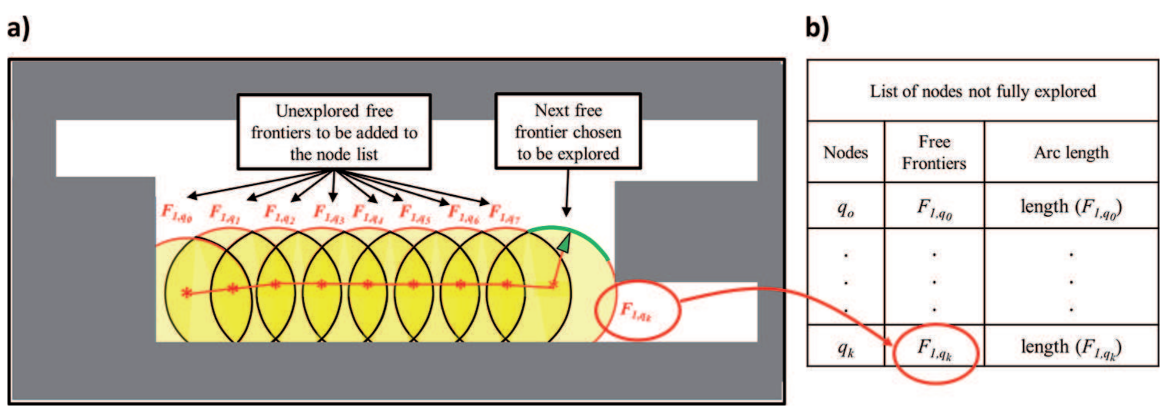

Figure 5.
Frontier control. (a) Environment almost explored, where the arcs F_i and q_j represent unexplored free frontiers. (b) List of nodes not fully explored (frontiers control).

will be added to the list of free frontiers F of the previous node. Thus, if the F list of the node is not empty, its header will be added to the list of nodes with exploration possibility, also known as frontier control (see **Figure 5**).

After verification and validation of the structure with the new node, the ADD function will attach it to the exploration graph, and the objects on the map being constructed will be extended with the new information collected. At this point, the destination information (q_{dest} and S_{dest}) obtained at the previous point will become the current node information (q_{curr} and S_{curr}), and the described process will start again.

When the robot fails to find a new position to explore in the current node, i.e., there are no more free frontiers, one of the nodes contained in the frontier control will be chosen to continue the exploration, where the choice of it will be determined by the A* search algorithm in bidirectional way, where a path will extend from the current node to the frontier control nodes and from the nodes in the frontier control to the current position, ending when some path P is found (see **Figure 6**). At this moment the index of the node on which the trajectory was found will be removed from frontier control. This task is carried out by the FIND_PATH function.

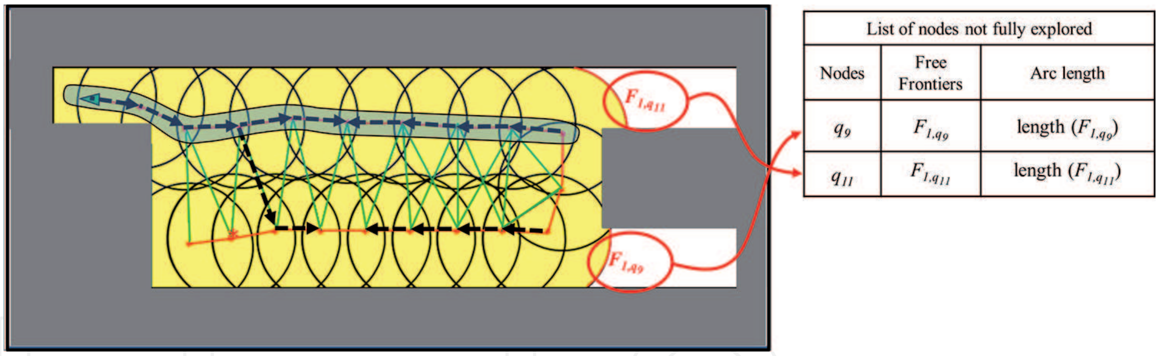

Figure 6.
Bidirectional application of the A algorithm from the current position of the robot to the nodes with possibility of exploration, stored in the frontiers control.*

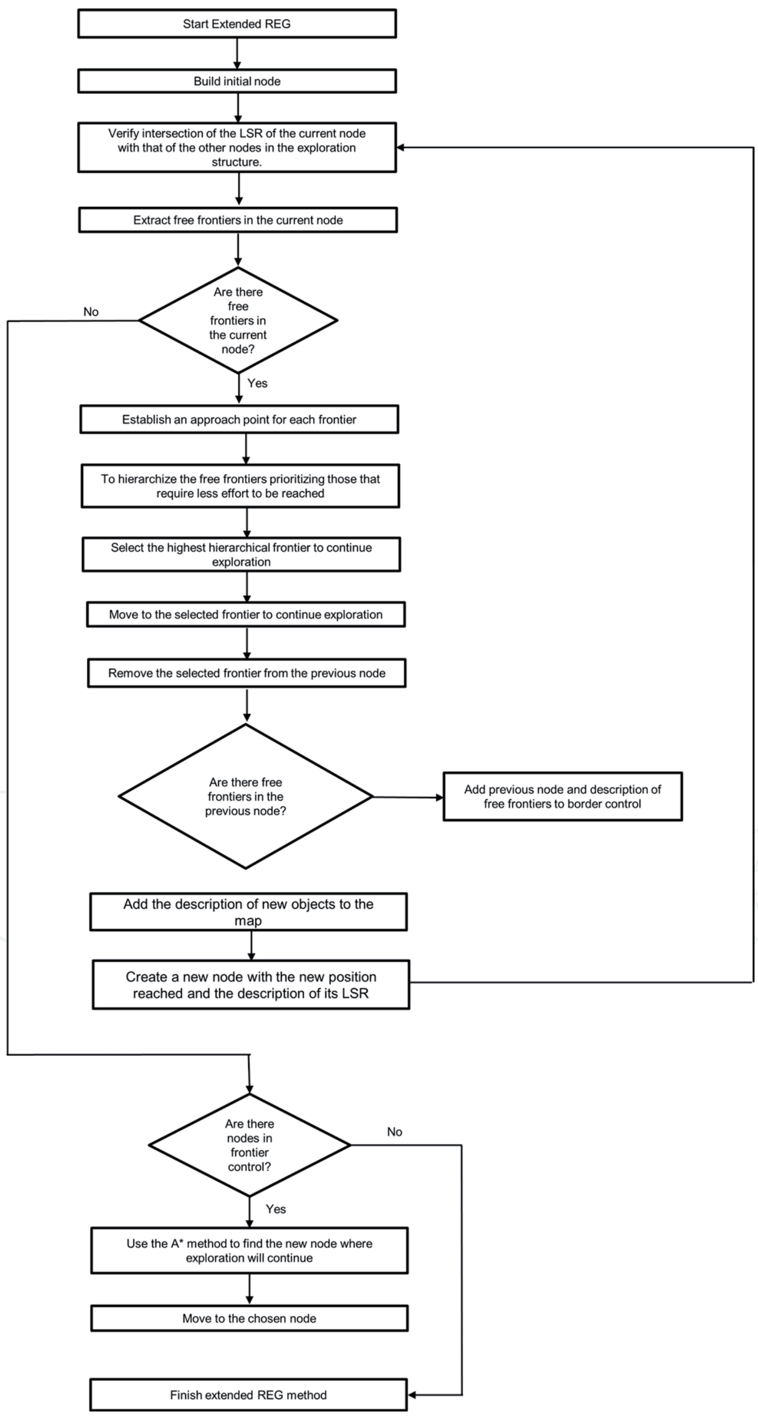

Figure 7.
Flowchart of the extended REG method.

The MOVE_TO function will then use the path P obtained in the previous step to take the robot to the node from where scanning will continue. In this way, the method will continue executing the described process, until there are no more free frontiers in the current node, and the frontier control list is empty; at this point, the robot will look for a path to return to the initial node from where the exploration process starts. **Figure 7** shows the flow diagram of the extended REG algorithm.

3. Experimental results

Numerous experiments were carried out with the intention of validating the accuracy and consistency of the proposal made in this investigation; in addition, typical quantitative variables used in the field of exploration methods were analyzed, such as exploration time, distance traveled, and total environmental coverage, which were compared with data obtained by other methods such as SRT [11], SRG [6], and REG [12], which allows us to explain the efficiency of our method.

With respect to the integrated exploration paradigm, our exploration approach was designed to operate under the general concept of any SLAM method; however, for the tests performed, it was determined to use the method presented in [13] given the integral way of exploiting data from the work environment.

The tests were conducted using simulated information from a pioneer P3DX differential robot, which was equipped with a Hokuyo URG-04LX range sensor with a maximum detection range of 4 meters, an angular resolution of 0.360°, and a scanning angle of 240°. The environment used for the experiments is a modified version of the corridors of the Montpellier Computer, Robotics and Micro-electronics laboratory (LIRMM) (see **Figure 8**).

Figure 9 shows the exploration structure generated by the extended REG method after its application in the LIRMM environment; in it, the edges represent routes that the robot can navigate without the risk of colliding with obstacles in the environment.

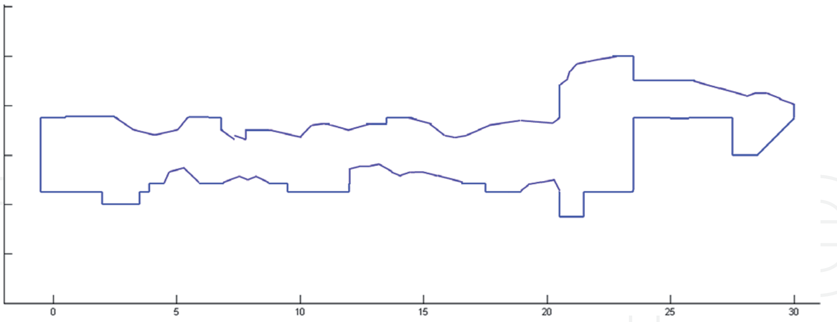

Figure 8.
The LIRMM environment.

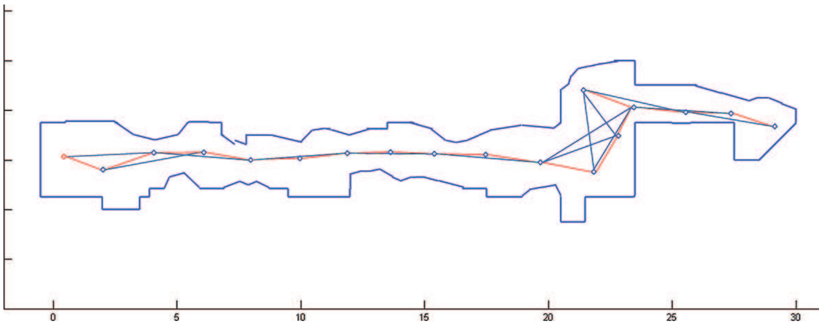

Figure 9.
Generated graph structure by the proposed exploration approach.

Tables 1 and **2** show the comparative results of the time and distance variables traveled by the robot using the SRT, SRG, REG, and Extended REG exploration methods; the results were obtained on the basis of 30 tests. In these tables, it is easy

LIRMM environment Exploration time (seconds)				
Test	Extended REG Method	REG Method	SRG Method	SRT Method
1	278.0889	315.7859	415.1445	563.1102
2	278.7979	326.7662	545.4457	687.7065
3	258.2204	359.3230	522.7711	441.5671
4	242.5923	318.7591	560.2500	688.0134
5	252.1377	402.3106	556.7573	519.7771
6	246.4645	334.3762	490.4706	502.1979
7	251.1057	294.8047	573.6789	584.2922
8	269.7181	336.8587	590.7163	666.9585
9	276.3011	380.5149	464.2052	650.8572
10	243.0794	310.7739	619.7049	503.9038
11	244.2341	416.5971	533.4251	665.0674
12	275.0934	364.7327	529.7478	577.1821
13	284.9026	374.9523	442.7496	569.8460
14	251.8885	294.3133	608.8775	453.4542
15	258.2455	337.2547	535.1509	475.1350
16	262.9808	309.5245	470.4473	626.2440
17	278.7238	340.1444	526.6756	636.6569
18	246.9795	378.1428	596.8990	515.8445
19	244.9627	327.7210	449.4658	683.2569
20	276.8152	384.1076	401.0013	606.8604
21	267.0117	305.4459	485.9701	505.5880
22	261.6335	358.1534	566.8682	420.1538
23	245.5880	330.3814	487.1188	570.1567
24	260.7357	406.2692	405.7856	426.6971
25	272.2311	406.2317	522.1408	475.3501
26	242.6921	369.3099	441.1361	573.6439
27	243.2405	329.8577	517.6172	532.1449
28	279.0752	383.5061	539.0328	609.7719
29	271.6649	302.1935	496.7664	675.1354
30	276.7029	417.5154	499.6830	672.9646
Average time	261.3969	350.5543	513.1901	569.3179
Standard Deviation	14.2286	37.9059	59.3018	84.9176

Table 1.
Time needed for the Extended REG, REG, SRT, and SRG exploration methods to explore the LIRMM environment on the basis of 30 tests

LIRMM environment Distance traveled by the robot (meters)				
Test	Extended REG Method	REG Method	SRG Method	SRT Method
1	94.2953	110.8750	176.5043	202.1030
2	97.0948	122.0881	171.5603	185.0873
3	93.5696	128.1114	148.7670	122.2305
4	92.8974	120.4839	154.3054	115.6293
5	95.2164	111.0441	99.9813	172.9572
6	96.4587	97.0754	161.0658	129.4353
7	89.4794	100.6069	122.4132	185.6566
8	97.8138	130.6982	135.4928	143.5254
9	88.1791	126.4050	112.2472	206.2968
10	90.1664	112.0377	137.2371	190.1586
11	88.9507	121.3078	120.6996	126.0833
12	96.3502	97.5515	128.1775	189.8909
13	98.2066	90.2293	176.2596	157.3321
14	89.9009	127.5927	119.5021	180.6779
15	91.3684	104.0339	174.5081	197.0456
16	92.2725	113.2718	133.4387	119.8085
17	93.6515	96.5836	165.1104	123.7504
18	88.6493	92.1664	164.9564	204.3196
19	97.8853	107.8139	127.0498	115.9971
20	98.4636	125.8068	158.0025	169.8498
21	97.9201	105.5806	127.9679	136.3876
22	95.8657	101.2719	152.5624	198.1987
23	96.7244	121.5712	153.0089	115.7825
24	93.0438	114.6411	117.7339	207.7524
25	96.3796	125.3939	103.3812	192.1662
26	88.3230	91.8653	127.6628	165.6248
27	96.4622	91.8655	160.8764	153.5368
28	92.7974	124.6801	154.4797	117.7001
29	89.3724	124.8849	106.7780	131.7636
30	95.9194	103.8811	104.1588	130.3949
Average distance	93.7893	111.3806	139.8630	159.5714
Standard Deviation	3.4125	12.9830	23.8407	33.4651

Table 2.
Distance traveled for the Extended REG, REG, SRT, and SRG exploration methods to cover the LIRMM environment on the basis of 30 tests

to observe that the Extended REG requires approximately 25% less time than the best average time of the other three methods, and about 16% in the best average distance was reported by the other three methods. In addition, it is possible to observe that

LIRMM environment Environment coverage (%)				
Test	**Extended REG Method**	**REG Method**	**SRG Method**	**SRT Method**
1	100%	100%	92.3592%	94.4365%
2	100%	100%	99.6659%	87.2937%
3	100%	100%	93.7106%	98.9930%
4	100%	100%	96.3776%	91.7059%
5	100%	100%	95.5086%	85.5518%
6	100%	100%	98.7745%	91.7032%
7	100%	100%	96.1068%	97.6451%
8	100%	100%	95.9189%	91.6015%
9	100%	100%	91.7190%	85.7556%
10	100%	100%	92.7108%	92.6386%
11	100%	100%	98.2033%	88.0735%
12	100%	100%	94.7589%	89.2148%
13	100%	100%	98.4398%	90.7828%
14	100%	100%	93.6729%	89.3881%
15	100%	100%	93.6892%	88.6274%
16	100%	100%	91.2083%	89.2020%
17	100%	100%	98.6001%	91.2047%
18	100%	100%	96.9909%	95.9539%
19	100%	100%	99.5792%	95.5916%
20	100%	100%	97.9685%	97.0562%
21	100%	100%	94.5170%	87.2147%
22	100%	100%	94.0025%	93.1977%
23	100%	100%	99.9579%	97.3336%
24	100%	100%	91.0922%	84.6677%
25	100%	100%	92.2726%	92.4071%
26	100%	100%	98.1241%	91.1557%
27	100%	100%	94.6410%	86.5926%
28	100%	100%	96.5149%	88.0764%
29	100%	100%	98.8809%	92.4875%
30	100%	100%	96.8469%	96.2568%
Average coverage	100%	100%	95.7604%	91.3937%
Standard Deviation	0%	0%	3%	4%

Table 3.
Surface covered of the LIRMM environment for the Extended REG, REG, SRT, and SRG exploration methods on the basis of 30 tests

the standard deviation in both variables is very low compared to the other methods due to the deterministic way of choosing the next position to explore, which allows sustaining the affirmation that the method will always obtain the same results.

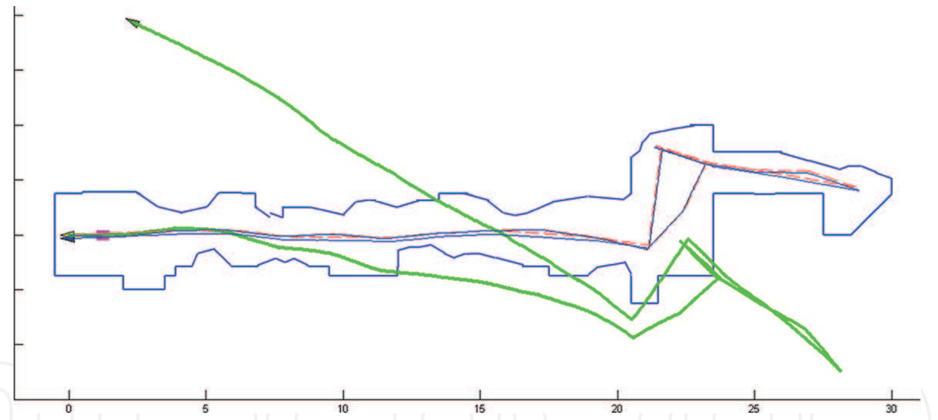

Figure 10.
Consistency test of the Extended REG method applied to the SLAM problem.

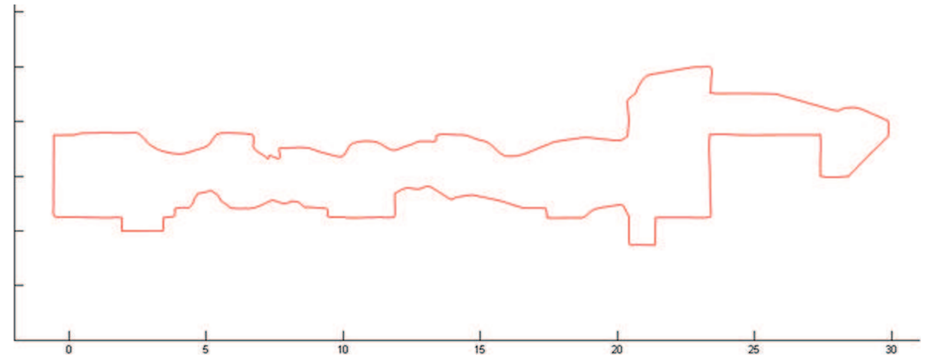

Figure 11.
Map obtained with the Extended REG method applied to the SLAM problem.

Moreover, since our proposal is based on the REG algorithm, one of the main benefits contained in the extension presented in this paper is the guarantee with a high degree of confidence that the environment will be fully covered in most cases, because it is possible to have a constant knowledge of the state of unexplored areas of the environment thanks to frontier control. Thus, to evaluate the coverage of the environment by the exploration method, this was divided into grids, which served to determine which of them had been explored (**Table 3**).

Finally, the algorithm of path planning for unknown environments presented in this article was developed with the intention of being integrated to SLAM algorithms to obtain an integral tool for the construction of autonomous maps. Although the Extended REG method could be used as a control module with any SLAM algorithm, for the tests performed, it was decided to use the method developed by Pedraza et al. [13] given the similarity of approaches when applying the methods in unstructured environments. The tests and results obtained are shown in **Figures 10** and **11**.

4. Conclusions

In this work, a strategy was presented for the problem of exploration of environments for SLAM; the approach presented is based on the REG algorithm introduced in [12], which builds a graph-like data structure that integrally exploits the experience acquired during the exploration process to perform this task efficiently. The main contribution of the exploration proposal made in this article is the use of a simplified criterion to find the next position to explore based on the hierarchy of free borders detected in an instant of time, which allows the elimination of unnecessary movements of the robot, increasing its efficiency. The main advantage of this choice criterion is that the robot will travel short distances to the position closest to being explored, reducing the amount of time needed to reach them, which can be verified in the results of the tests performed to the method.

Also the Extended REG method is designed to be integrated in the context of a SLAM method, which facilitates the construction of environment maps simplifying the task of planning paths in unknown environments, which allows giving true autonomy to the robot responsible for obtaining the environment map eliminating the dependence on decision-making by a human operator. Finally, a series of simulations of the proposed integrated exploration strategy were carried out, which allowed us to validate our approach.

Author details

Alfredo Toriz Palacios[1]* and Abraham Sánchez López[2]

1 Popular Autonomous University of the State of Puebla, Puebla, Mexico

2 Autonomous University of Puebla, Puebla, Mexico

*Address all correspondence to: alfredo.toriz@upaep.mx

References

[1] Feder H, Leonard J, Smith C. Adaptive mobile robot navigation and mapping. International Journal of Robotics Research. 1999;**18**(7):650-668. DOI: 10.1177/02783649922066484

[2] Campos FM, Marques M, Carreira F, Calado JMF. A complete frontier-based exploration method for Pose-SLAM. In: IEEE International Conference on Autonomous Robot Systems and Competitions. 2017. pp. 79-84. DOI: 10.1109/ICARSC.2017.7964056

[3] Fermin-Leon L, Neira J, Castellanos JA. TIGRE: Topological graph based robotic exploration. In: European Conference on Mobile Robots. 2017. pp. 1-6. DOI: 10.1109/ECMR.2017.8098718

[4] Meng Z, Sun H, Qin H, Chen Z, Zhou C, Ang MH. Intelligent robotic system for autonomous exploration and active SLAM in unknown environments. In: IEEE/SICE International Symposium on System Integrations. 2017. pp. 651-656. DOI: 10.1109/SII.2017.8279295

[5] Ristic B, Palmer J. Autonomous exploration and mapping with RFS occupancy-grid SLAM. In: Journal Entropy. 2018;**20**:456-464. DOI: 10.3390/e20060456

[6] Rodríguez-Arévalo ML, Neira J, Castellanos JA. On the importance of uncertainty representation in active SLAM. In: IEEE Transactions on Robotics. 2018;**34**:829-834. DOI: 10.1109/TRO.2018.2808902

[7] Saeedi S, Nardi L, Johns E, Bodin B, Kelly PH, Davison AJ. Application-oriented design space exploration for SLAM algorithms. In: IEEE International Conference on Robotics and Automation. 2017. pp. 5716-5723. DOI: 10.1109/ICRA.2017.7989673

[8] Sun X, Sun F, Wang B, Yin J, Sheng X, Xiao Q. Robotic autonomous exploration SLAM using combination of Kinect and laser scanner. In: IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems. 2017. pp. 632-637. DOI: 10.1109/MFI.2017.8170393

[9] Valencia R, Andrade-Cetto J. Path planning in belief space with pose SLAM. In: Mapping, Planning and Exploration with Pose SLAM. Vol. 53. Pittsburgh, United States of America: Springer; 2017. p. 87. DOI: 10.1007/978-3-319-60603-3_4

[10] Wang H, Zhang C, Song Y, Pang B. A perception-driven exploration hierarchical simultaneous localization and mapping for mobile robots adapted to search and rescue environments. In: Advances in Mechanical Engineering. 2018;**10**:1-18. DOI: 10.1177/1687814018765874

[11] Oriolo G, Vendittelli M, Freda L, Troso G. The SRT method: Randomized strategies for exploration. In: IEEE/RSJ International Conference on Intelligent Robots and Systems. 2004. pp. 4688-4694. DOI: 10.1109/ROBOT.2004.1302457

[12] Toriz A, Sánchez A, Bedolla JME. The random exploration graph for optimal exploration of unknown environments. In: International Journal of Advanced Robotic Systems. 2017;**1**: 1-11. DOI: 10.1177/1729881416687110

[13] Pedraza L, Dissanayake G, Miro JV, Rodriguez-Losada D, Matia F, BS-SLAM: Shaping the world. In: Robotics: Science and Systems. 2007. pp. 1-8. DOI: 10.15607/RSS.2007.III.022