

The SRT Method: Randomized strategies for exploration

Giuseppe Oriolo, Marilena Vendittelli, Luigi Freda, Giulio Troso
Dipartimento di Informatica e Sistemistica, Università di Roma “La Sapienza”
Via Eudossiana 18, 00184 Roma, Italy
Email: {oriolo,vendittelli,freda}@dis.uniroma1.it, giulio.troso@fastwebnet.it

Abstract— We present a method for sensor-based exploration of unknown environments by a mobile robot. The method is based on the randomized incremental generation of a data structure called Sensor-based Random Tree (SRT), which represents a roadmap of the explored area with an associated safe region. Different exploration strategies may be obtained by instantiating the general method with different perception techniques. Two such techniques are discussed: the first, conservative and particularly suited to noisy sensors, results in an exploration strategy called SRT-Ball. The second perception technique is more confident, and the corresponding strategy is called SRT-Star. The two strategies are critically compared by simulations as well as by experiments on the MagellanPro robot.

I. INTRODUCTION

The exploration of unknown environments can be considered to be the fundamental problem for mobile robots, as it involves all the basic capabilities of such systems, e.g., perception, planning, localization, and navigation. From a practical viewpoint, exploration is a central task in many applications, such as planetary missions, intervention in hostile areas and automatic map building.

A widely accepted definition of exploration is “the act of moving through an unknown environment while building a map that can be used for subsequent navigation” [1]. The performance of exploration strategies must then be assessed on the basis of the quality of the obtained map as well as of the time needed (or space traveled) in order to assemble it. Most of the existing techniques fall into the class of *frontier-based exploration*, pioneered by Yamauchi in the above cited paper. The rationale of this approach is that the robot must move towards the boundary (the *frontier*) between safe explored areas and unknown territory in order to maximize the information gain coming from new perceptions. For example, the methods proposed in [2–5] may be classified as frontier-based exploration, although they differ under other aspects (e.g., the representation of the environment, the inclusion of localization, the consideration of perception costs, the use of multiple robots).

It is also interesting to adopt a more general AI perspective, according to which exploration is “the process of selecting actions in active learning” [6]. In the active learning paradigm, training data are obtained as a result of the learner’s actions. In particular, the case of robot that

gathers information about the environment by moving and sensing is a case of *order-sensitive* active learning, because the data flow is the result of all past robot actions. The central problem of exploration is how to select the next action. Frontier-based exploration is obtained when the criterion is the maximization of the action’s (expected) utility. There exists however another possibility, i.e., use a random selection mechanism (also called a *random walk*). The advantages of this choices are (i) simplicity and (ii) the fact that any sequence of actions will be executed eventually. The latter property opens the road to *completeness*: a solution will be found whenever one exists. On the other hand, pure random action selection may be very inefficient.

Our approach to the problem derives from randomized (also called probabilistic) motion planning (RMP) techniques, which build roadmaps of the free configuration space by generating random samples and checking them for collision [7]. In RMP, the problem setting is that of *order-free* active learning: what is observed about the environment depends only on the last action (the random query), because the map of the environment is available in advance and the robot does move to gather data. Hence, randomized planners can be considered as goal-oriented exploration strategies based on random action selection. As already mentioned, the (probabilistic) completeness of these planners is inherent to their nature; in addition, very high efficiency can be achieved by adding heuristics to the basic random scheme. The RRT method of [8] is a typical example which served as an inspiration for our work.

In this paper, we describe an exploration method based on the random generation of robot configurations within the local safe area detected by the sensors. A data structure called Sensor-based Random Tree (SRT) is created, which represents a roadmap of the explored area with an associated Safe Region (SR). Each node of the SRT consists of a free configuration with the associated Local Safe Region (LSR) as reconstructed by the perception system; the SR is the union of all the LSRs.

The LSR is an estimate of the free space surrounding the robot at a given configuration; in general, its shape will depend on the sensor characteristics (for example, angular resolution) but may also reflect different attitudes towards perception. We will present two exploration strategies obtained by instantiating the general method

with different perception techniques. The first, where the LSR is a ball, realizes a conservative attitude, particularly suited to noisy or low-resolution sensors, and results in an exploration strategy called SRT-Ball. The second technique is confident, and the corresponding strategy is called SRT-Star; in this case, the LSR shape reminds of a star. The two strategies are compared by simulations as well as by experiments on the mobile robot MagellanPro.

Our exploration method will be presented under the assumption of perfect localization provided by some other module. While this may be the case (for example, a GPS system may be used in a planetary exploration mission), one cannot overlook the fact that such an assumption is often unreasonable in unknown environments. However, our investigation is still at a preliminary stage and our first interest was to assess the potential of randomized exploration techniques. The integration of a localization step into the exploration process is currently in progress.

II. EXPLORATION VIA THE SRT METHOD

In this section, we first clarify the working assumptions on the robot and the environment; then, we describe the proposed exploration method from a general viewpoint, i.e., independently from the particular perception strategy.

A. Working assumptions

A robot must explore a workspace, i.e., an environment with obstacles. We take the following assumptions:

- 1) The workspace is planar, i.e., either \mathbb{R}^2 or a (connected) subset of \mathbb{R}^2 .
- 2) The robot is a disk (or may be reasonably approximated by a disk) free to translate in any direction (*holonomic* or *free-flying* robot). Hence, the configuration space is a copy of the workspace with the obstacles grown so as to allow for the robot size [9].
- 3) The robot always knows its configuration q (see the last paragraph of the previous section).
- 4) The robot is equipped with a sensory system which provides at each q an estimate (possibly conservative) of the surrounding free space. This estimate, called Local Safe Region at q and denoted by \mathcal{S} , is a *star-shaped* subset of \mathbb{R}^2 , i.e., it is homeomorphic to the closed unit ball and the line segment connecting q to any point of \mathcal{S} is completely contained in \mathcal{S} [9].

While the techniques presented in this paper rely on such assumptions, the concluding section contains some comments about the possibility of relaxing them. The fact that the Local Safe Region is star-shaped (Assumption 4) is consistent¹ with the physics of the most common sensors, i.e., range finders, but also applies to more sophisticated perception techniques (e.g., vision). A concept similar to LSRs are the ‘safe areas’ of [4].

¹Simple geometry shows that this is true provided that the sensors are located on a circle having the same center of the robot (as in sonar rings) or at the center itself (as in a rotating laser range finder).

B. The SRT method

The exploration method proposed in this paper builds a data structure called Sensor-based Random Tree (SRT), which can be considered as a variation of the Rapidly-exploring Random Tree (RRT) proposed by LaValle in [8]. Like RRT, the SRT is a tree which represents a roadmap of the free configuration space of the robot. Each node of the SRT consists of a collision-free configuration q which the robot has reached, together with a description of the Local Safe Region \mathcal{S} surrounding q as perceived through the sensory system. The tree is incrementally built by extending the structure towards randomly selected directions in such a way that the new configuration (and the path reaching it) is contained in the Local Safe Region.

The algorithm implementing the SRT method can be described as follows.

```

BUILD_SRT ( $q_{\text{init}}, K_{\text{max}}, I_{\text{max}}, \alpha, d_{\text{min}}$ )
 $q_{\text{curr}} = q_{\text{init}}$ ;
for  $k=1$  to  $K_{\text{max}}$ 
 $\mathcal{S} \leftarrow \text{PERCEPTION}(q_{\text{curr}})$ ;
ADD( $\mathcal{T}, (q_{\text{curr}}, \mathcal{S})$ );
 $i \leftarrow 0$ ;
loop
 $\theta_{\text{rand}} \leftarrow \text{RANDOM\_DIR}$ ;
 $r \leftarrow \text{RAY}(\mathcal{S}, \theta_{\text{rand}})$ ;
 $q_{\text{cand}} \leftarrow \text{DISPLACE}(q_{\text{curr}}, \theta_{\text{rand}}, \alpha \cdot r)$ ;
 $i \leftarrow i + 1$ ;
until (VALID( $q_{\text{cand}}, d_{\text{min}}, \mathcal{T}$ ) or  $i = I_{\text{max}}$ )
if VALID( $q_{\text{cand}}, d_{\text{min}}, \mathcal{T}$ )
MOVE_TO( $q_{\text{cand}}$ );
 $q_{\text{curr}} \leftarrow q_{\text{cand}}$ ;
else
MOVE_TO( $q_{\text{curr}}.\text{parent}$ );
 $q_{\text{curr}} \leftarrow q_{\text{curr}}.\text{parent}$ ;
return  $\mathcal{T}$ ;

```

At each iteration k of the algorithm, a perception process takes place (i.e., sensor data are gathered and elaborated) to obtain a star-shaped region \mathcal{S} which estimates the free space surrounding the robot at the current configuration q_{curr} . A new node containing the configuration q_{curr} and the associated LSR \mathcal{S} is then added to the tree \mathcal{T} . The way \mathcal{S} is actually represented within the SRT data structure depends on the perception strategy: in general, an algebraic description of its boundary could be used, but we shall see in the next two sections that much lighter representations are possible in practice.

At this point, a direction of exploration θ_{rand} is randomly generated by the function RANDOM_DIR, and the function RAY is invoked to compute the ‘radius’ r of \mathcal{S} in the direction of θ_{rand} (see Figure 1). A candidate new configuration q_{cand} for the robot is determined by taking a step of length $\alpha \cdot r$ in the direction of θ_{rand} . The constant $\alpha < 1$ guarantees that q_{cand} is within the safe area \mathcal{S} and can be reached with a path contained in \mathcal{S} ; values of α close to 1 will increase the exploratory attitude of the algorithm, while smaller values will augment the safety margin.

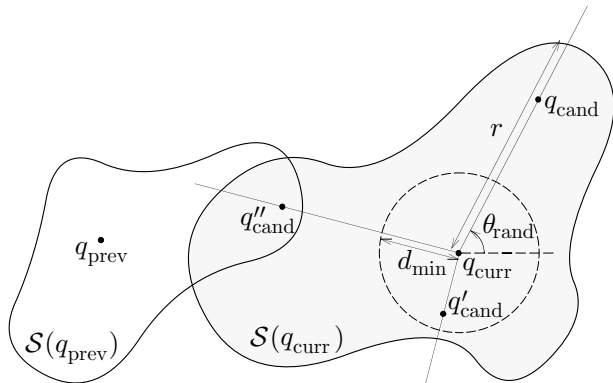


Fig. 1. The generation of candidate configurations in the SRT method. In the case shown, q_{cand} would be validated, while q'_{cand} and q''_{cand} would not: the first falls within a minimum distance d_{min} from q_{curr} , the second in the Local Safe Region of another node.

Once q_{cand} has been randomly generated within the safe area \mathcal{S} , it goes through a validation step performed by the boolean function VALID. As shown in Figure 1, q_{cand} must (i) be at a distance larger than a given d_{min} from q_{curr} , and (ii) not fall in the Local Safe Region of any other node belonging to \mathcal{T} . If validation is successful, the robot moves to q_{cand} and the cycle is repeated. Otherwise, the algorithm generates other random configurations from q_{curr} until one is validated or a maximum number I_{max} of trials is exceeded. In the latter case, the robot backtracks to the parent node of q_{curr} , where the exploration cycle starts again. A succession of failures in finding new exploration directions, typical when the free space has been completely explored, will force the robot to backtrack to the root — realizing therefore an automatic *homing* mechanism.

A comparison of the SRT method with the RRT planner of [8] suggests the following remarks.

- If compared with the RRT data structure, the SRT is a tree with edges of variable length, depending on the radius r of the LSR in the random direction θ_{rand} . Hence, during the exploration, the robot will take longer steps in regions scarcely populated by obstacles and smaller steps in cluttered regions. Also, no collision check is needed on the candidate configurations as they are always generated within the safe area.
- From the exploration viewpoint, the SRT method is inherently depth-first. In fact, the tree is expanded along directions originating from q_{curr} , the current robot position; this is in contrast with the breadth-first expansion typical of the pure RRT, which however does not apply to sensor-based exploration. The introduction of a backtracking mechanism was obviously a consequence of the depth-first nature of SRT.
- The SRT method retains some of the most important features of RRT, such as the fact of being particularly suited for high-dimensional configuration spaces and easily modifiable to account for the presence of both holonomic and nonholonomic constraints.

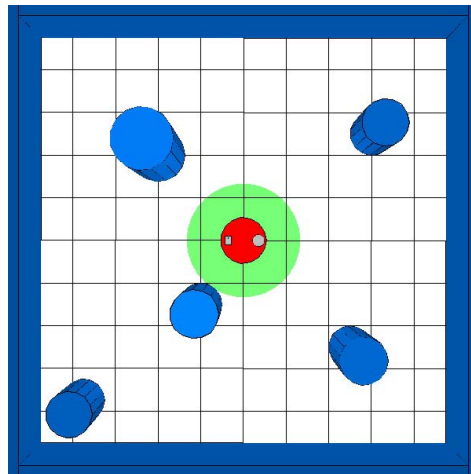


Fig. 2. The Local Safe Region \mathcal{S} according to the SRT-Ball perception strategy. The robot is the circular body located at the center of the scene.

III. CONSERVATIVE EXPLORATION: SRT-BALL

As already mentioned, the shape of the Local Safe Region \mathcal{S} must reflect the sensor characteristics as well as the adopted perception technique. In turn, the exploration strategy is strongly affected by the shape of \mathcal{S} . In this section, we consider a particular instance of the general SRT method, called SRT-Ball, which embodies a conservative approach to perception and, hence, to exploration.

In SRT-Ball, \mathcal{S} is defined as a ball (a very special case of star-shaped region) whose radius r is determined by the distance η of the closest obstacle. In particular, the radius of the ball is computed as the minimum between η and the maximum measurable range with the available sensors (see Figure 2). The function $\text{RAY}(\mathcal{S}, \theta_{\text{rand}})$ simply returns the same value r for any direction θ_{rand} , and the Safe Region is built as the union of balls of different size.

To illustrate the behavior of SRT-Ball, we report the results of two simulations realized within Move3D², under the assumption of perfect sensing. The algorithm parameters are $K_{\text{max}} = 1000$, $I_{\text{max}} = 50$, $\alpha = 0.8$, $d_{\text{min}} = 0.07$ m.

Results from the first simulation are shown in Figure 3. In the first frame, the robot starts from the center of the scene; each successive frame shows the tree built up to that point (remember that each node corresponds to a configuration assumed by the robot) together with the associated Safe Region. When the robot has reached the end of the corridor, in the lower right corner of the scene, no valid candidate configuration can be generated; the robot then backtracks along the tree and, where possible, expands the latter by adding new configurations. As a consequence, the estimation of the Safe Region becomes increasingly accurate. In the end, the robot goes back to the root node where the exploration stops. Due to the

²Move3D [10] is a powerful software platform developed at LAAS-CNRS which provides many motion planning primitives and a graphical environment for scene description. More details can be found at <http://www.laas.fr/~nic/Move3D/>.

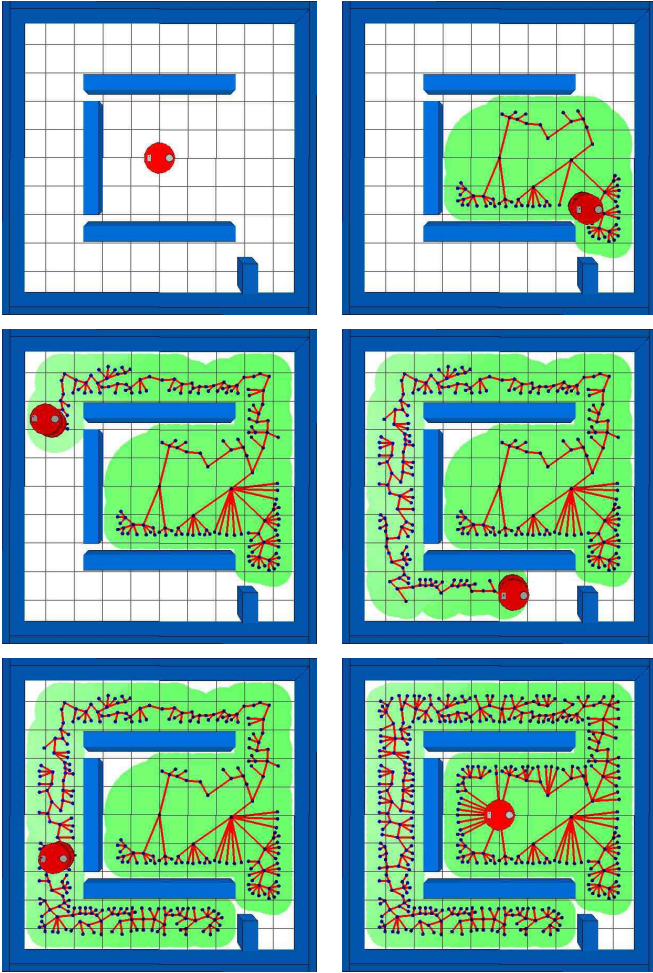


Fig. 3. Simulation 1: The exploration process with SRT-Ball (frames are ordered from left to right and from top to bottom).

fixed circular shape of \mathcal{S} , the boundary of the Safe Region is a coarse approximation of the obstacles in the first four frames. Accuracy however improves during backtracking, as can be appreciated from the last two frames.

The second simulation (Figure 4) shows how the conservative nature of SRT-Ball may preclude the robot from completing exploration tasks which require the negotiation of narrow passages. As the robot approaches the small door, the perceived LSR gradually reduces due to the lateral walls: the SRT-Ball perception strategy cannot ‘see’ through the opening. When the radius r of \mathcal{S} becomes smaller than d_{\min} , no candidate configuration can be validated and the robot backtracks to the home configuration, where the exploration terminates.

IV. CONFIDENT EXPLORATION: SRT-STAR

While the conservative perception of SRT-Ball ignores the directional information provided by most sensory systems, SRT-Star can exploit it. Below, we consider the case of a ring of range finders placed along the robot profile.

In SRT-Star, \mathcal{S} is a star-shaped region given by the union of different ‘cones’, with a different radius in each

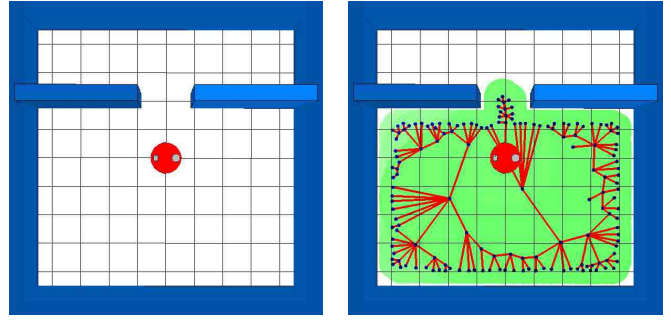


Fig. 4. Simulation 2: The exploration process with SRT-Ball. Only the initial and final frames are shown.

cone (see Figure 5). The i -th cone radius is the minimum between η_i (the distance to the closest obstacle within the cone) and the maximum measurable range with the available sensors. Hence, to compute r the function RAY must first identify the cone corresponding to θ_{rand} .

To allow a performance comparison between the two exploration strategies, we have run the same simulations of the previous section with SRT-Star, under the assumption that a ring of 16 range finders is available. The same parameter values have been used, except for $I_{\max} = 16$.

Results obtained for the first simulation are shown in Figure 6. As can be noted by comparing the second frame with the corresponding frame of Figure 3, SRT-Star has a more pronounced depth-first exploration attitude with respect to SRT-Ball, whose tree is more expanded in width. Still, the estimate of the free space built by SRT-Star is more accurate right from the start because the variable shape of \mathcal{S} allows a finer reconstruction of the obstacle region boundary. Table I summarizes the performance of SRT-Ball and SRT-Star in this simulation. The total traveled distance and the final number of nodes in the tree are much smaller with SRT-Star than with

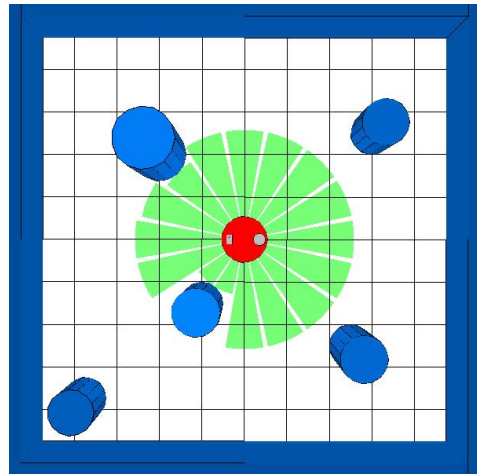


Fig. 5. The Local Safe Region \mathcal{S} according to the SRT-Star perception strategy. Note how the extension of \mathcal{S} in some cones is reduced due to the sensor limited measurable range.

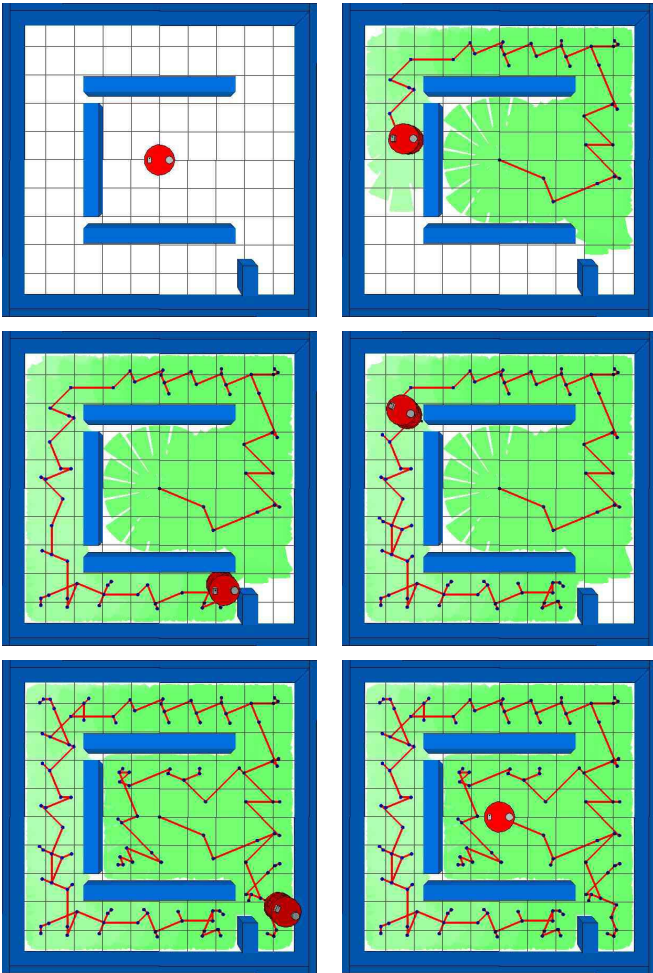


Fig. 6. Simulation 1: The exploration process with SRT-Star (frames are ordered from left to right and from top to bottom).

1st simulation	SRT-Ball	SRT-Star
traveled distance (m)	143.4	63.8
number of nodes	419	125
filling	92%	98%

TABLE I

SRT-Ball. We have also computed the percentage of the free space that is covered by the final Safe Region (*filling*).

The superiority of SRT-Star is even more evident from the results of the second simulation in Figure 7. Now, the robot is able to cross the narrow passage; in fact, the radius of \mathcal{S} is not reduced in all directions as the robot approaches the small opening, and candidate configurations across the door are generated and validated. A comparative summary of the results is given in Table II: of course, SRT-Star gives a much better filling since the area beyond the small passage is explored. Moreover, this is obtained by traveling a smaller distance and reaching less nodes (and hence, performing less perceptions) with respect to SRT-Ball.

Simulations clips are available at the web page <http://www.dis.uniroma1.it/~labrob/research/SRT.html>.

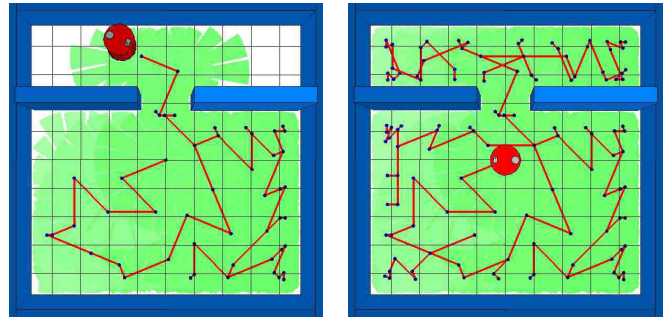


Fig. 7. Simulation 2: The exploration process with SRT-Star. An intermediate and the final frame are shown.

2nd simulation	SRT-Ball	SRT-Star
traveled distance (m)	87.6	74.4
number of nodes	155	120
filling	72%	99%

TABLE II

V. SRT VS. FRONTIER-BASED METHODS

If compared with the frontier-based approach, we note that the SRT method does not make a distinction between obstacles and unexplored areas; in fact, the boundary of the Local Safe Region may indifferently describe the sensor range limit or the profile of an object (see Figure 5). This means that during the exploration the robot may approach areas which appear to be occluded. However, when the obstacle is at close range, candidate configurations which would push the robot closer to the occlusion will not be validated because they fall within the minimal distance d_{\min} from the current point; thus, the robot will promptly move away from the obstacle boundary without a significant loss of efficiency. On the other hand, a more accurate estimate of the Safe Region will typically be obtained, due to (i) the increased precision of wide-angle range finders from closer distance, and (ii) the smaller size³ of the Local Safe Region. The above is valid for the general SRT method, and thus for SRT-Ball and SRT-Star as well.

Another difference between the SRT method and other techniques is the way in which the environment is represented. The free space estimate built during the exploration is simply the union of the LSR associated to the nodes of the tree. Relatively simple post-processing operations would however allow to compute a global description of the Safe Region, more useful for navigation tasks.

VI. EXPERIMENTS WITH SRT

The SRT-Ball and SRT-Star exploration strategies have been implemented on the MagellanPro robot available in our lab. MagellanPro is a two-wheeled differential-drive robot with a caster wheel added for stability; its shape is circular with a diameter of 40 cm. Its sensory system

³The global Safe Region may be considered as the result of a topological *erosion* of the actual free space, in which the LSR acts as a variable-size structuring element.

includes a ring of 16 ultrasonic range finders (range 0.2 to 4 m), a ring of 16 infrared range finders (range 0.07 to 1 m) and another ring of 16 bumper sensors; besides, an on-board pan-tilt camera is available.

MagellanPro is a nonholonomic robot, and thus violates the free-flying assumption of Sect. II-A. In these preliminary experiments, we have simply solved this problem by using a rotation/translation/rotation maneuver to implement the MOVE_TO function of the SRT method. Also, we have used the built-in dead reckoning system for localization; while this is clearly a limitation, the homing error at the end of the exploration was quite acceptable (see the details below).

The perception step for both SRT-Ball and SRT-Star is realized via the ultrasonic and infrared sensors: in each sensor cone (having a 22.5° angle) a very basic sensor fusion between the two available range readings is performed, so that a single range reading is obtained for each cone. While this directly provides the Local Safe Region \mathcal{S} for SRT-Star, an additional minimization over all the cones is required for SRT-Ball. The parameters for both algorithms have been chosen as follows: $K_{\max} = 500$, $I_{\max} = 50$, $\alpha = 3/4$ and $d_{\min} = 10$ cm.

The first experiment refers to MagellanPro exploring a very simple T-shaped environment using SRT-Star. Figure 8 shows some snapshots of the exploration process: the left image of each frame was taken by a hand-held camera, while the right image shows the tree built by the robot together with the safe area reconstructed up to that point; the actual location of the obstacles is superimposed for the sake of clarity. The environment is successfully explored and the resulting SRT (which includes 28 nodes) provides a satisfactory roadmap of the free space. In particular, the total traveled distance is 19.8 m (consider that the perimeter of the environment is 14.2 m). Thanks to the backtracking mechanism, MagellanPro returns to the starting position once the exploration is completed; the homing error is 0.07 m.

Note that at some configurations the Local Safe Region extends beyond the actual free space. This happens when the angle of incidence between the ultrasonic beam and the obstacle surface is below a given threshold (false reflection). A small value of α is used by SRT-Star to avoid that, in these situations, the candidate configuration q_{cand} is placed within an obstacle. In any case, MagellanPro continuously checks its range finders while moving; if an unexpected obstacle is detected at close distance (typically, by the infrared sensors) the robot stops and performs a new perception step. We emphasize that it is possible to detect and correct such erroneous range measures by relatively simple elaborations (e.g., see [11]); here, we have purposely chosen to avoid any post-processing of the measures in order to assess the performance of the exploration method in the presence of noisy data.

In the second experiment, MagellanPro was deployed into a rectangular ‘room’ with two obstacles protruding

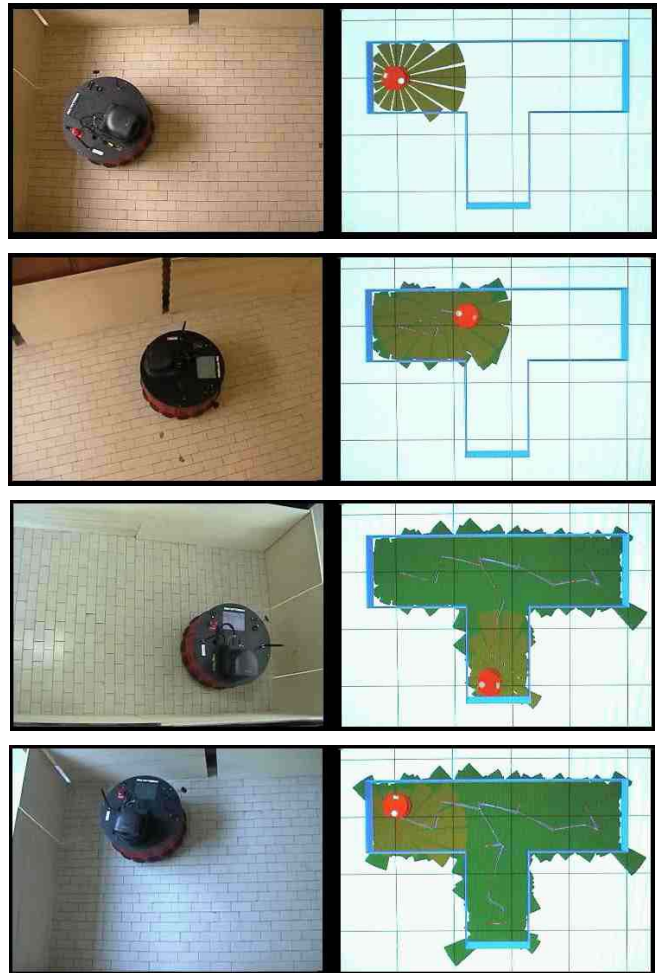


Fig. 8. Experiment 1: The exploration process with SRT-Star.

from the upper and lower wall (see Figure 9). The perimeter of the open space boundary is now 17.5 m. The results obtained using SRT-Ball and SRT-Star are summarized in Table III for comparison. As expected, the exploration path produced by SRT-Star is much shorter (and therefore the odometric homing error is reduced); moreover, the SRT-Star tree contains approximately 1/6 of the nodes of the SRT-Ball tree. On the other hand, the conservative perception attitude typical of SRT-Ball drastically reduces the inclusion of false reflections in the estimated safe area.

Experiments clips are available at the web page <http://www.dis.uniroma1.it/~labrob/research/SRT.html>.

VII. CONCLUSIONS AND FUTURE WORK

We have described a general method for sensor-based exploration of unknown environments by a mobile robot. The method proceeds by building a data structure called Sensor-based Random Tree (SRT) through random generation of configurations. The SRT represents a roadmap of the explored area with an associated Safe Region, an estimate of the free space as perceived by the robot during the exploration. Different strategies are then obtained by instantiating the general method with different perception

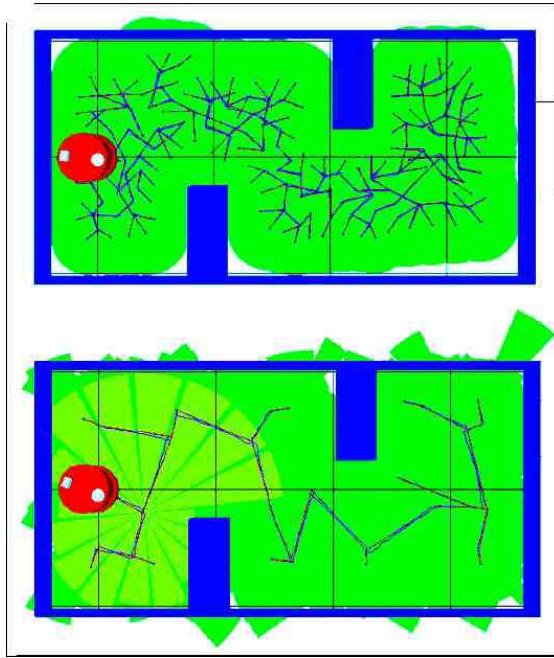


Fig. 9. Second experiment: the final appearance of the tree and the associated Safe Region obtained with SRT-Ball (above) and SRT-Star (below).

2nd experiment	SRT-Ball	SRT-Star
traveled distance (m)	74.6	21.8
homing error (m)	0.21	0.08
number of nodes	190	30

TABLE III

techniques. In particular, two such techniques have been presented: SRT-Ball and SRT-Star. The first is more conservative and particularly appropriate for noisy sensors, while SRT-Star is more confident. The two strategies have been critically compared by simulations as well as by experiments performed on the mobile robot MagellanPro.

Below, we briefly mention some of the extensions of the current work which will be the subject of future work.

- Nonholonomic mobile robots (such as the MagellanPro) do not satisfy the free-flying assumption of Sect. II-A. While we have coped with this difficulty by using rotation/translation/rotation maneuvers, more general nonholonomic local planners can be used to connect q_{curr} to a validated q_{cand} ; clearly, the planner must be appropriately designed so as to guarantee that the resulting path is completely contained in \mathcal{S} . Also, it would be interesting to modify the construction of the Local Safe Region \mathcal{S} so as to account for the nonholonomic obstacle distance defined in [12].
- In principle, the SRT method can be used for sensor-based exploration of high-dimensional configuration spaces; this may be, for example, the case of a mobile manipulator. The main issue would become the generation of a *configuration-space* Local Safe Region from

the available sensor measures. Clearly, such region cannot be expected to be star-shaped, so it would be necessary to reduce it appropriately. The same happens with single-body mobile robots if the range finders are not arranged on a ring.

- Experimental results of SRT-Star in the presence of false reflections indicate that, in order to achieve robust performance and accurate estimation of the free space, the definition of the Local Safe Region \mathcal{S} must incorporate a model of the available sensors including uncertainty and/or faults. For example, the safe area definition of [4] includes a simple mechanism for taking into account incidence angle constraints.
- As mentioned in the introduction, the integration of a localization step into the exploration process based on a SLAM technique is currently under way.
- One of the benefits of randomized search techniques is that they are generally amenable to parallelization. In our case, this means that the SRT method lends itself quite directly to an extension to multirobot exploration. However, to guarantee that the promise for increased efficiency is fulfilled, some degree of coordination between the robots must be introduced [5].

REFERENCES

- [1] B. Yamauchi, "A frontier-based approach for autonomous exploration," in *IEEE Int. Conf. on Robotics and Automation*, 1997, pp. 146–151.
- [2] B. Yamauchi, A. Schultz, and W. Adams, "Mobile robot exploration and map-building with continuous localization," in *IEEE Int. Conf. on Robotics and Automation*, 1998, pp. 3715–3720.
- [3] A. Makarenko, S. B. Williams, F. Bourgault, and H. F. Durrant-Whyte, "An experiment in integrated exploration," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2002, vol. 1, pp. 534–539.
- [4] H. H. Gonzalez-Banos and J.-C. Latombe, "Navigation strategies for exploring indoor environments," *Int. J. Robotics Research*, vol. 21, no. 10, pp. 829–848, 2002.
- [5] W. Burgard, M. Moors, D. Fox, R. Simmons, and S. Thrun, "Collaborative multi-robot exploration," in *IEEE Int. Conf. on Robotics and Automation*, 2000, vol. 1, pp. 476–481.
- [6] S. Thrun, "Exploration in active learning," in *Handbook of Brain and Cognitive Science*, M. Arbib, Ed. MIT Press, 1995.
- [7] J. Barraquand, L. Kavraki, J.-C. Latombe, T.Y. Li, R. Motwani, and P. Raghavan, "A random sampling scheme for path planning," in *Robotics Research: The Seventh International Symposium*, G. Giralt and G. Hirzinger, Eds., pp. 249–264. Springer-Verlag, London, 1996.
- [8] S. M. LaValle and J. J. Kuffner, "Rapidly-exploring random trees: Progress and prospects," in *Algorithmic and Computational Robotics: New Directions*, B. R. Donald, K. M. Lynch, and D. Rus, Eds., chapter 10, pp. 293–308. A K Peters, Wellesley, MA, 2001.
- [9] J.-C. Latombe, *Robot Motion Planning*, Kluwer Academic Publishers, Norwell, MA, 1991.
- [10] T. Simeon, J.-P. Laumond, and F. Lamiraux, "Move3D: A generic platform for path planning," in *4th Int. Symp. on Assembly and Task Planning*, 2001, pp. 25–30.
- [11] G. Oriolo, G. Ulivi, and M. Vendittelli, "Real-time map building and navigation for autonomous robots in unknown environments," *IEEE Trans. on Systems, Man and Cybernetics, Part B*, vol. 28, no. 3, pp. 316–333, 1998.
- [12] M. Vendittelli, J.-P. Laumond, and C. Nissoux, "Obstacle distance for car-like robots," *IEEE Transactions on Robotics and Automation*, vol. 15, no. 4, pp. 678–691, 1999.