Proceedings of the 2005 IEEE
International Conference on Robotics and Automation
Barcelona, Spain, April 2005

Frontier-Based Probabilistic Strategies for Sensor-Based Exploration

Luigi Freda Giuseppe Oriolo
Dipartimento di Informatica e Sistemistica
Università di Roma "La Sapienza"
Via Eudossiana 18, 00184 Roma, Italy
{*freda,oriolo*}*@dis.uniroma1.it*

Abstract— **We present a frontier-based modification of the SRT (Sensor-based Random Tree) method, a previously proposed probabilistic strategy for sensor-based exploration of unknown environments by a mobile robot. The idea is to improve the efficiency of the method by biasing the randomized generation of configurations towards unexplored areas. Effective implementations of this strategy are proposed for SRT-Ball and SRT-Star, two instances of the general SRT method corresponding to different perception attitudes and sensing equipments. Comparative simulations are presented to show the benefits of the proposed technique.**

Index Terms— **Sensor-based exploration, probabilistic strategies.**

I. Introduction

One of the most challenging problems in robotics is the exploration of unknown environments. Depending on the application, this may be aimed at different goals, such as building a navigation map or locating a given object. Many exploration strategies fall into the class of *frontier-based* exploration, e.g., see [1–4]: based on the idea that the robot should approach the boundary between explored areas and unknown territory, exploration proceeds by maximizing the expected utility of robot motions. A different approach is to use a random selection mechanism (*random walk*), whose advantages are simplicity and the fact that any sequence of actions will be executed eventually [5]. Clearly, pure random selection may prove inefficient.

Our approach derives from randomized motion planning techniques [6]: these can be considered as goal-oriented exploration strategies based on random walk which achieve high efficiency by adding heuristics to the basic scheme. In particular, in [7] we have introduced a method based on the random generation of configurations within the Local Safe Region detected by the sensors. A data structure called Sensor-based Random Tree (SRT) is incrementally built, which represents a roadmap of the explored area with an associated Safe Region. The SRT can be considered as a sensor-based version of the RRT proposed in [8].

Depending on the shape of the Local Safe Region, the general method results in different exploration strategies. In SRT-Ball, the safe region is estimated as the maximum collision-free circle surrounding the robot, a conservative choice particularly suited to noisy or wide-angle rangefinders. In SRT-Star the perception technique is more confident, and the Local Safe Region is a star-shaped set.

In this paper, we present a frontier-based modification of the SRT method. The idea is to increase the exploration efficiency by biasing the randomized generation of configurations towards unexplored areas. The difference with the methods in [1–4] stands in the fact that our method does not use a global map for identifying the frontier of the explored region, and is still probabilistic in nature. The advantages of this choice are *(i)* simplicity and *(ii)* the fact that any sequence of actions will be executed eventually. The latter property opens the road to *completeness*: in principle, a solution will be found whenever one exists.

After recalling the foundations of the SRT method, we present the basic idea and work out frontier-based versions of both SRT-Ball and SRT-Star, whose improved performance is shown by comparative simulations. Since our interest is to assess the potential of randomized exploration techniques, the exploration method is presented under the assumption of exogenous perfect localization. The integration of a localization module into the exploration process is currently in progress (see the concluding section).

II. The SRT exploration method

In this section, we recall the basics of SRT [7] as a general exploration method (i.e., independently of the chosen perception strategy), and summarize two particular instances of SRT: the first, conservative and 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.

SRT is developed under the following assumptions[1]: *(i)* the workspace is planar, i.e., $I\!R^2$ or a (connected) subset of $I\!R^2$; *(ii)* the robot is holonomic and disk-shaped; *(iii)* the robot always knows its configuration q; *(iv)* at each q, the sensory system provides an estimate $\mathcal{S}(q)$ of the surrounding free space in the form of a *star-shaped* [9] subset of $I\!R^2$ called Local Safe Region.

The core of the method is the construction of a data structure called Sensor-based Random Tree (SRT). Each node of the SRT consists of a collision-free configuration q which the robot has visited, together with a description of the Local Safe Region $\mathcal{S}(q)$ as perceived through the sensory system. The tree is incrementally built by extending

[1] See [7] for comments on the possibility of relaxing these as well as for more details on the method.

```
BUILD_SRT ($q_{\rm init}$,$K_{\max}$,$I_{\max}$,$\alpha$,$d_{\min}$)
  $q_{\rm curr}$ = $q_{\rm init}$;
  for $k$=1 to $K_{\max}$                                   % at the current configuration:
   $\mathcal{S}(q_{\rm curr})$ ← PERCEPTION($q_{\rm curr}$);                    % compute $\mathcal{S}$ from sensor data
   ADD($\mathcal{T}$,($q_{\rm curr}$, $\mathcal{S}(q_{\rm curr})$));                       % update the tree
   $i$ ← 0;
   loop
     $\theta_{\rm rand}$ ← RANDOM_DIR;                         % generate a random direction $\theta_{\rm rand}$
     $r$ ← RAY($\mathcal{S}(q_{\rm curr})$,$\theta_{\rm rand}$);              % compute the radius $r$ of $\mathcal{S}$ along $\theta_{\rm rand}$
     $q_{\rm cand}$ ← DISPLACE($q_{\rm curr}$, $\theta_{\rm rand}$,$\alpha{\cdot}r$);   % generate $q_{\rm cand}$ by taking a step of length $\alpha{\cdot}r$ along $\theta_{\rm rand}$
     $i$ ← $i+1$;
   until (VALID($q_{\rm cand}$,$d_{\min}$,$\mathcal{T}$) or $i = I_{\max}$)
   if VALID($q_{\rm cand}$,$d_{\min}$,$\mathcal{T}$)                                  % validate $q_{\rm cand}$
     MOVE_TO($q_{\rm cand}$);                                    % if $q_{\rm cand}$ is valid, go there
     $q_{\rm curr}$ ← $q_{\rm cand}$;                                     % ...and repeat the cycle
   else
     MOVE_TO($q_{\rm curr}$.parent);               % after $I_{\max}$ unsuccessful trials, backtrack
     $q_{\rm curr}$ ← $q_{\rm curr}$.parent;
  return $\mathcal{T}$;
```

Fig. 1. A pseudocode description of the SRT algorithm

the structure towards randomly selected directions in such a way that the new configuration (and the path reaching it) is contained in the $\mathcal{S}(q)$.

The algorithm implementing the SRT method is shown in Fig. 1. Note the following points.

- Letting $\alpha \leq 1$ guarantees that both $q_{\rm cand}$ and the path reaching it lie in $\mathcal{S}$; thus, no collision check is needed. Smaller values of α will increase the safety margin.
- The validation step performed by the VALID function is illustrated in Fig. 2: $q_{\rm cand}$ must *(i)* be further than a given $d_{\min}$ from $q_{\rm curr}$, and *(ii)* not fall in the Local Safe Region of any other node belonging to $\mathcal{T}$.
- A succession of failures in finding exploration directions, typical when the free space has been completely explored, forces the robot to backtrack to the root.
- The length of the SRT edges varies depending on the radius r in the direction $\theta_{\rm rand}$. The robot will take longer steps in open areas and smaller steps in cluttered regions. As an exploration method, SRT is depth-first, due to its sensor-based nature. The introduction of backtracking was natural in view of this fact.

The SRT method is a general paradigm. The shape of the Local Safe Region $\mathcal{S}$ must reflect the sensor characteristics and the adopted perception technique; in turn, the exploration strategy will be affected by the shape of $\mathcal{S}$.

A particular instance of the general SRT method, called SRT-Ball, is obtained by defining $\mathcal{S}$ as the ball (a special case of star-shaped region) whose radius r is the minimum range reading, see Fig. 3. Note that r may be the distance to the closest obstacle or, in wide open areas, the maximum range of the available sensors. In SRT-Ball, the function RAY($\mathcal{S}$,$\theta_{\rm rand}$) simply returns the same value r for any direction $\theta_{\rm rand}$, and the Safe Region is built as the union of balls of different size. SRT-Ball embodies a *conservative* approach to perception and, hence, to exploration.

Another instance of the method, called SRT-Star, takes full advantage of the directionality of sensor rings. In this version, $\mathcal{S}$ is defined as a star-shaped region given by the union of different 'cones', with a different radius in each cone (see Fig. 4). The i-th cone radius is the minimum between 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 $\theta_{\rm rand}$.

SRT-Star shows a more pronounced depth-first search attitude with respect to SRT-Ball, whose tree typically expands more in width. The estimate of the free space built by SRT-Star is more accurate from the very start, because the variable shape of $\mathcal{S}$ allows a finer reconstruction of the obstacle region boundary. Moreover, the total traveled distance and the final number of nodes in the tree are much smaller with SRT-Star than with SRT-Ball. See [7] for a detailed comparison. Movie clips of other simulations of SRT, as well as experiments on the MagellanPro robot, are available at the web page http://www.dis.uniroma1.it/˜labrob/research/SRT.html.

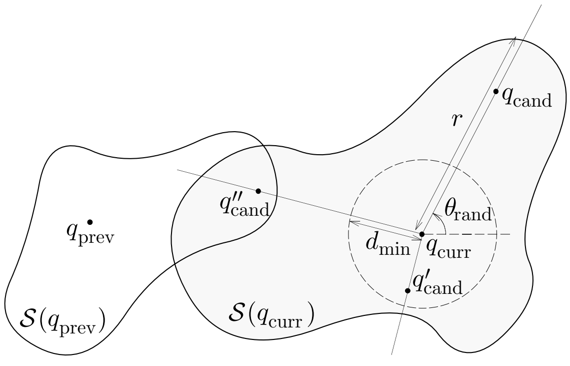

Fig. 2. Validation of candidate configurations in SRT: $q_{\rm cand}$ would be validated, while $q'_{\rm cand}$ and $q''_{\rm cand}$ would not.

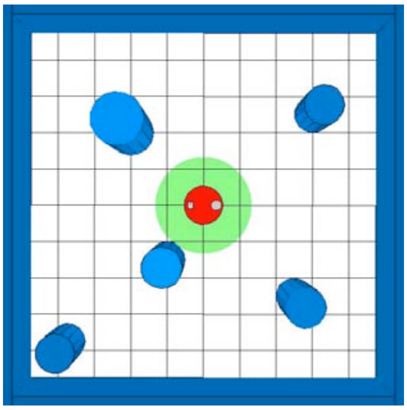

Fig. 3. The Local Safe Region $\mathcal{S}$ according to the SRT-Ball perception strategy. The robot is the circular body located at the center of the scene.

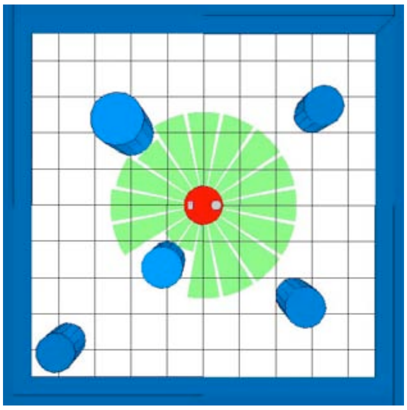

Fig. 4. The Local Safe Region $\mathcal{S}$ according to the SRT-Star perception strategy. Note how the extension of $\mathcal{S}$ in some cones is reduced due to the sensor limited measurable range.

III. A FRONTIER-BASED VERSION OF SRT

Differently from frontier-based approaches, the SRT method does not distinguish between obstacles and unexplored areas; in fact, the boundary of the Local Safe Region contains obstacle points as well as free points (see Figs. 3–4). This means that during the exploration the robot may approach areas which are occluded. When the obstacle is at sufficiently close range, candidate configurations which would push the robot closer to the obstacle will not be validated, because they will fall within the minimal distance $d_{\min}$ from the current point; thus, the robot will move away from the obstacle boundary, only with some 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 rangefinders from closer distance, and *(ii)* the smaller size[2] of the Local Safe Region.

In large environments, however, the 'meticulousity' of SRT may result in inefficient exploration. To alleviate this drawback, we propose to modify the basic strategy in a frontier-based sense, at least locally. The idea is to divide the boundary of each ball in *obstacle*, *free* and *frontier* arcs, as in Fig. 5: the first are located along the directions where an obstacle has been sensed, the second are contained within other Local Safe Regions (and therefore lie in the free explored workspace), while the third identify unexplored directions. Generation of random directions (i.e., procedure RANDOM_DIR) is then biased towards the frontier arcs.

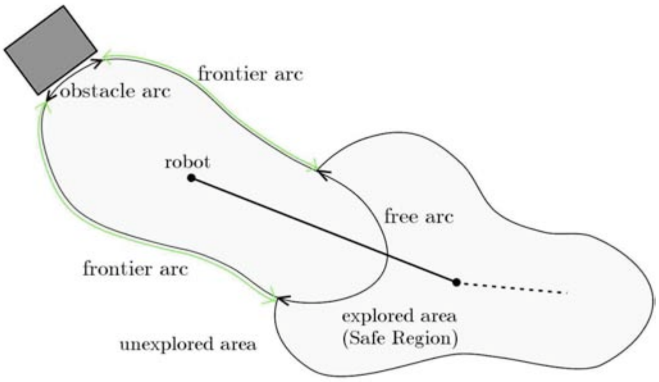

Fig. 5. The definition of *obstacle*, *free* and *frontier* arcs

A. Frontier identification in FB_SRT-Ball

In SRT-Ball, a procedure for identifying frontier arcs is the following (Fig. 6). One sample point is placed along the axis of each rangefinder at a distance r (the radius of the ball). Each point is classified as *obstacle* if the corresponding range measure is r, as *free* if it falls in the ball of another node belonging to the tree, and as *frontier* otherwise. The resulting classification is propagated to the *elementary arc* represented by the point, i.e., the arc resulting from the intersection of the ball with the corresponding sensor cone. The final frontier, free, and obstacle arcs are obtained by joining contiguous elementary boundary portions that have been classified as such. In the following, the frontier-based version of SRT-Ball is called FB_SRT-Ball.

B. Frontier identification in FB_SRT-Star

In SRT-Star, the Local Safe Region $\mathcal{S}$ is given by the union of the sensor cones representing the free range measured by the sensors. Therefore, in this case, the boundary of the star-shaped region is constituted of arcs and portions of segments belonging to the cones. To represent efficiently these parts, three sample points can be associated to the i-th cone as described in Fig. 7 (left). The first point, called $mid\_point[i]$, is placed along the middle axis of the i-th cone at a distance $radius[i]$, equal to the reading of the i-th range finder. The other points, $right\_point[i]$ and $left\_point[i]$, are placed on the lateral segments of the sensor cone: each of these points exists and is associated to the i-th cone if and only if the reading of the contiguous cone (on the corresponding side) is smaller than $radius[i]$. If $radius[i]$ is smaller than $radius[i-1]$ ($radius[i+1]$) the right (left) lateral segment of the i-th cone falls inside the Local Safe Region, and consequently $right\_point[i]$ ($left\_point[i]$) is not considered.

Once this construction is completed, $mid\_point[i]$ represents the arc of the i-th cone and is classified as *obstacle* if

[2] Note that the global Safe Region built by SRT is the result of a topological *erosion* of the actual free space, in which the LSR acts as a variable-size *structuring element*.

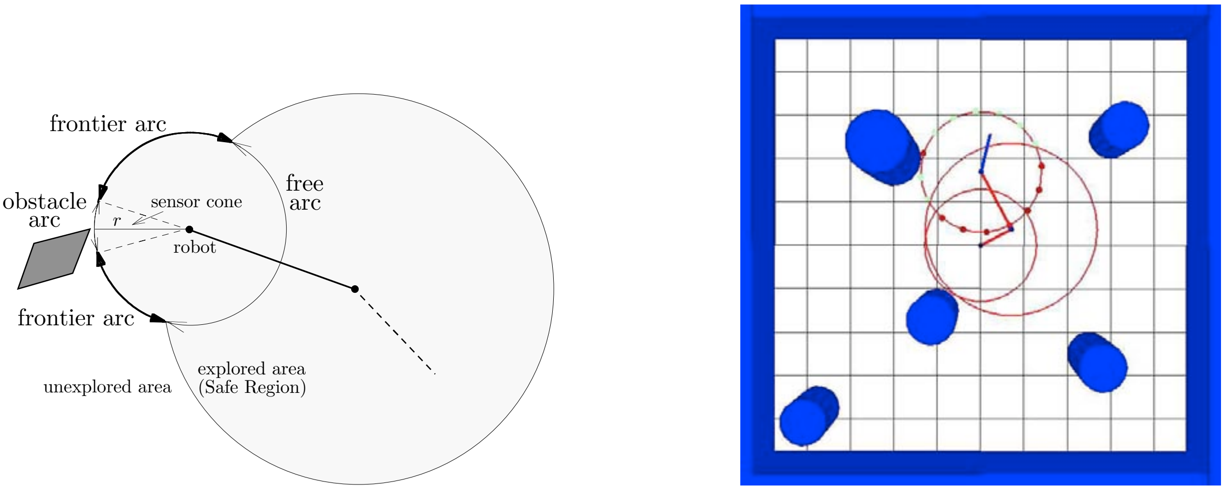

Fig. 6. *Left*: The definition of *obstacle*, *free* and *frontier* arcs in FB_SRT-Ball. *Right*: The computation of these arcs in practice: the light (green) dots represent frontier points, while the dark (red) dots represent either free points or obstacle points (only one in this case). Also shown (in blue) is the randomly generated direction θ_{rand}, which is close to the bisectrix of a frontier arc as a result of the probabilistic generation mechanism.

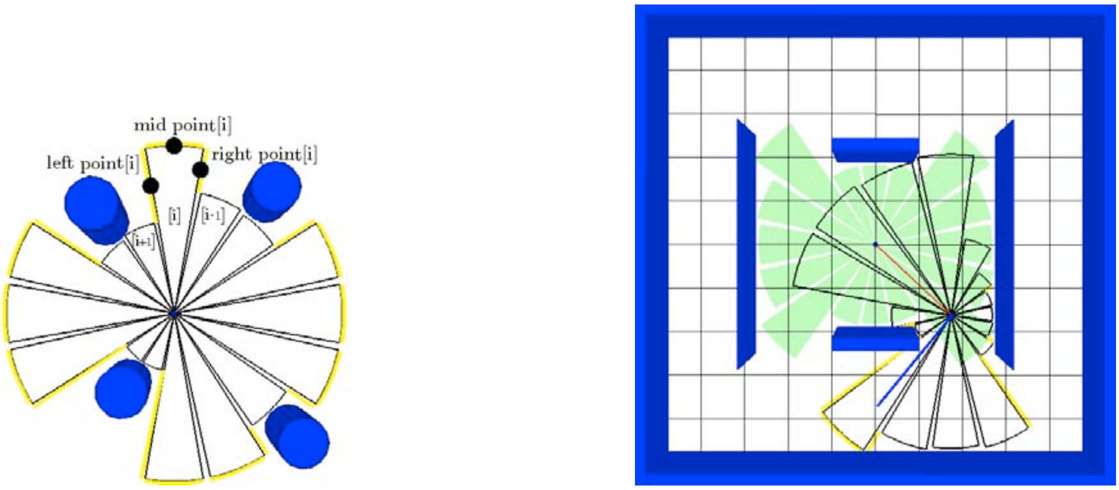

Fig. 7. *Left*: The computation of *frontier* arcs in FB_SRT-Star through association of three sample points in the cone. The light (yellow) portions of the boundary represent frontier arcs, while the black portions represent represent either free arcs or obstacle arcs. *Right*: Generation of a random direction θ_{rand} close to the bisectrix of a frontier arc as a result of the probabilistic generation mechanism.

the corresponding range measure $radius[i]$ is smaller than the maximum sensor reading, as *free* if it falls in the Local Safe Region of another node belonging to the tree, and as *frontier* otherwise (Fig. 7). The lateral point $right\_point[i]$ ($left\_point[i]$), if it exists, represents the portion of the right (left) segment of the i-th cone protruding outside the contiguous cone, and is classified as *free* if it falls in another Local Safe Region, and as *frontier* otherwise. Again, this classification is propagated to the elementary arc or segment represented by the point, and elementary arcs with the same classification are grouped. The frontier-based version of SRT-Ball is called FB_SRT-Star.

C. Biased random generation of configurations

Node generation is realized in two steps using the same mechanism for both FB_SRT-Ball and FB_SRT-Star. Once the above classification of the Local Safe Region boundary has been completed, one of the frontier arcs is randomly selected, with a probability proportional to its length (if no frontier arc exists, the robot backtracks). The selected arc is represented by its central angle γ and the orientation θ_m of its bisectrix[3]. An exploration direction θ_{rand} is then generated according to a normal distribution with mean value θ_m and standard deviation $\sigma = \gamma/6$. A candidate new configuration q_{cand} is then determined and validated following the general SRT method.

IV. Simulations

To compare the performance of the two instances of SRT and FB_SRT, we report the results of simulations realized in Move3D [10], a software platform developed at LAAS-CNRS and dedicated to motion planning[4]. The algorithm parameters are $I_{\max} = 32$, $\alpha = 1$, $d_{\min} = 0.07$ m. The

[3] With FB_SRT-Star, if a frontier arc is composed only by a lateral segment of the i-th cone, γ and θ_m are selected as the central angle and the bisectrix of the i-th cone, even if $mid\_point[i]$ is not classified as frontier. This choice still pushes the robot towards the frontier portion detected by the sensors.

[4] Move3D is at the origin of the product KineoWorks currently marketed by the company Kineo CAM (www.kineocam.com).

simulated robot is the MagellanPro, equipped with 16 ultrasonic rangefinders. The robot diameter is 0.4 m.

A. Simulations with FB_SRT-Ball

Results from the first simulation, in which the environment to be explored is quite small (16 m^2), are shown in Fig. 8. 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. Both simulations have been stopped after the same number of iterations ($K_{\max} = 200$): note how FB_SRT-Ball is more efficient in exploring the environment, but also slightly less accurate in reconstructing the free space. In Table I we give results for five realizations of the exploration process, in terms of the distance traveled by the robot and of the percentage of the actual free space covered by the final Safe Region (*filling*).

The second simulation (Fig. 9 and Table II) takes place in a larger environment (256 m^2) and confirms the results of the first; here, $K_{\max} = 800$. Note in particular how FB_SRT-Ball achieves a (roughly) double average filling percentage with respect to SRT-Ball, at the price of a modest increase in the traveled distance.

B. Simulations with FB_SRT-Star

FB_SRT-Star was simulated in the same two environments used for FB_SRT-Ball. In both cases, the number of iterations reported is much smaller than before ($K_{\max} = 40$ for the first simulation and $K_{\max} = 200$ for the second) due to the better efficiency of SRT-Star and FB_SRT-Star.

The more pronounced depth-first nature of FB_SRT-Star pushes the robot to explore faster and more effectively the environment with respect to SRT-Star (Fig. 10 and Fig. 11). Note how the edges of the tree built by FB_SRT-Star (which correspond to the robot path segments) stay closer to the center of the passages between walls; this is partly due to the gaussian random generation, which privileges directions (and hence displacements) close to the bisectrix of the selected frontier arc. Again, SRT-Star is more accurate in reconstructing the obstacle boundary, whereas FB_SRT-Star leaves an area as soon as it has been mostly covered. For example, the last frames of Fig. 11 show that with SRT-Star the robot insists in exploring the bottom-right corner of the environment, while with FB_SRT-Star it has already covered most of the free area.

Again, the frontier-based method achieve much better filling in the same number of iterations (Tables III–IV).

V. Conclusions and Current Work

We have presented a frontier-based modification of the SRT method, a recent probabilistic strategy for sensor-based exploration of unknown environments by a mobile robot. The idea was to improve the efficiency of the method by biasing the randomized generation of configurations towards unexplored areas. Implementations of this strategy have been described for SRT-Ball and SRT-Star, two instances of the general SRT method corresponding to different perception attitudes. Comparative simulations have confirmed the benefits of the proposed technique.

To conclude, we mention some of the work currently in progress or recently completed:

- Preliminary experimental validation of the frontier-based strategies on the MagellanPro robot has confirmed their increased efficiency with respect to the original SRT-Ball and SRT-Star, whose implementation on the same robot was discussed in [7]. In addition, we are currently preparing an experimental comparison between our strategies and the deterministic exploration technique introduced in [1], in order to better assess the pro/cons of our method in practice.
- It is well-known that sensor-based map building and navigation requires localization. In the spirit of [3], we are developing an *integrated exploration* method based on SRT. In particular, incremental localization is performed as the exploration progresses with the aid of natural features directly extracted from the Local Safe Area description. In turn, the 'localization potential' is kept into account when validating a candidate configuration.
- A third instance of SRT, called SRT-Spot, is being devised for robots equipped with rotating laser rangefinders or other similar sensors with high angular resolution. In this version, the Local Safe Region $\mathcal{S}$ directly built from the sensor readings (a very 'dense' star) is transformed in a more compact, algebraically defined representation by clustering and interpolation. This post-processing step increases both the space and time efficiency of the algorithm.

References

[1] B. Yamauchi, "A frontier-based approach for autonomous exploration," in *1997 IEEE Int. Conf. on Robotics and Automation*, pp. 146–151, 1997.

[2] B. Yamauchi, A. Schultz, and W. Adams, "Mobile robot exploration and map-building with continuous localization," in *1998 IEEE Int. Conf. on Robotics and Automation*, pp. 3715–3720, 1998.

[3] A. Makarenko, S. B. Williams, F. Bourgault, and H. F. Durrant-Whyte, "An experiment in integrated exploration," in *2002 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, vol. 1, pp. 534–539, 2002.

[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] S. Thrun, "Exploration in active learning," in *The Handbook of Brain Theory and Neural Networks,*, M. A. Arbib, Ed., pp. 381–384. MIT Press, 1995.

[6] 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.

[7] G. Oriolo, M. Vendittelli, L. Freda, and L. Troso "The SRT Method: Randomized strategies for exploration," in *2004 IEEE Int. Conf. on Robotics and Automation*, pp. 4688-4694, 2004.

[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., 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*, pp. 25–30, 2001.

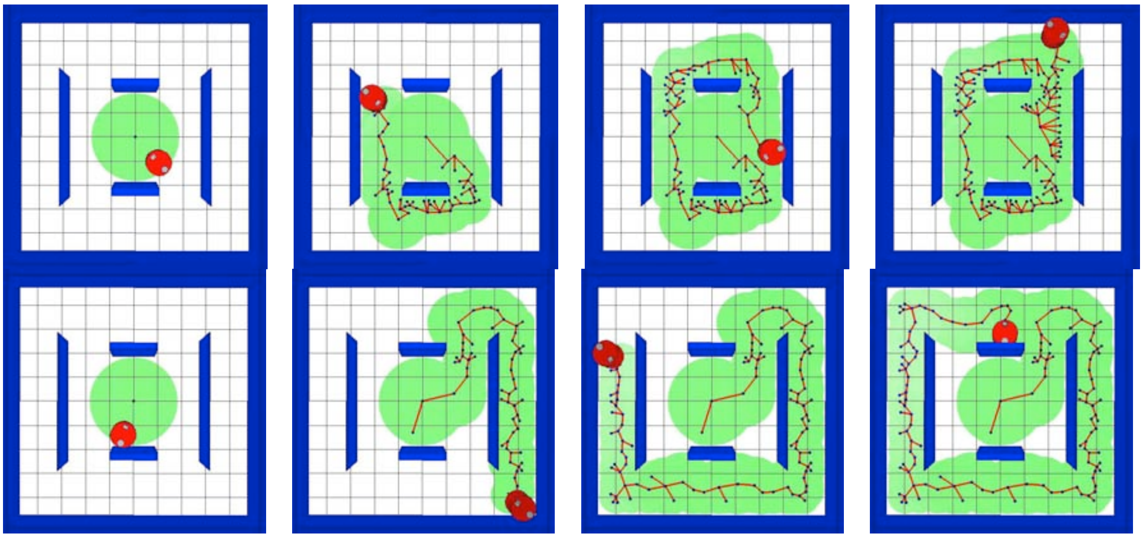

Fig. 8. Small environment: Exploration with SRT-Ball (above) and FB_SRT-Ball (below)

SRT-Ball	traveled distance (m)	30.28	29.16	30.10	31.84	31.84	ave. 30.57
	filling	52.31%	46.95%	51.23%	47.68%	49.57%	ave. 49.54%
FB_SRT-Ball	traveled distance (m)	32.50	28.16	30.62	29.82	30.72	ave. 30.36
	filling	63.55%	81.15%	65.15%	63.96%	72.49%	ave. 69.26%

TABLE I

RESULTS IN THE SMALL ENVIRONMENT (200 ITERATIONS)

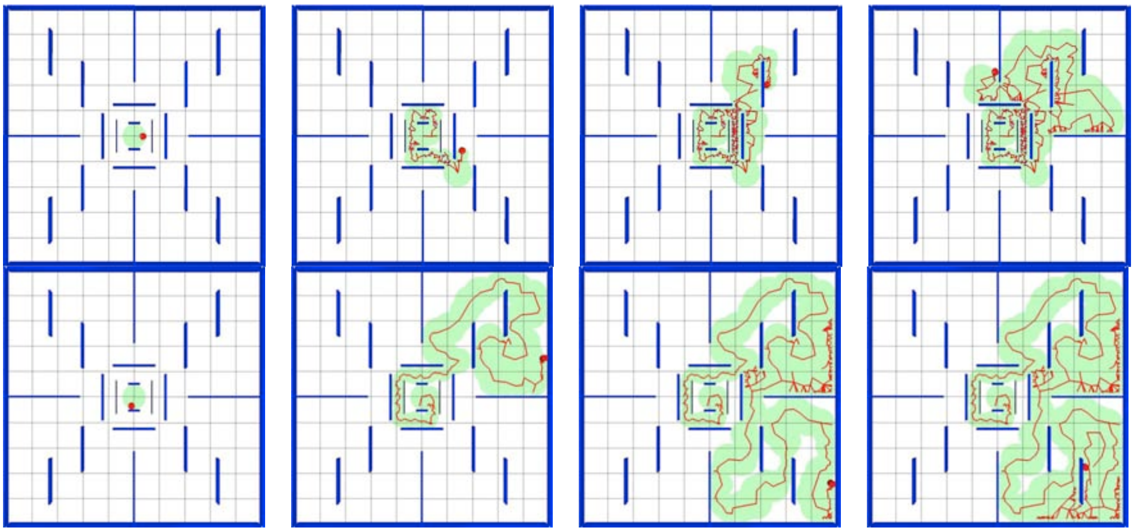

Fig. 9. Large environment: Exploration with SRT-Ball (above) and FB_SRT-Ball (below)

SRT-Ball	traveled distance (m)	197.74	210.26	218.04	188.10	236.84	ave. 210.20
	filling	26.99%	26.26%	26.50%	32.87%	26.74%	ave. 27.87%
FB_SRT-Ball	traveled distance (m)	222.40	232.46	237.46	232.56	243.62	ave. 233.70
	filling	48.33%	52.31%	42.57%	40.17%	67.68%	ave. 50.21%

TABLE II

RESULTS IN THE LARGE ENVIRONMENT (800 ITERATIONS)

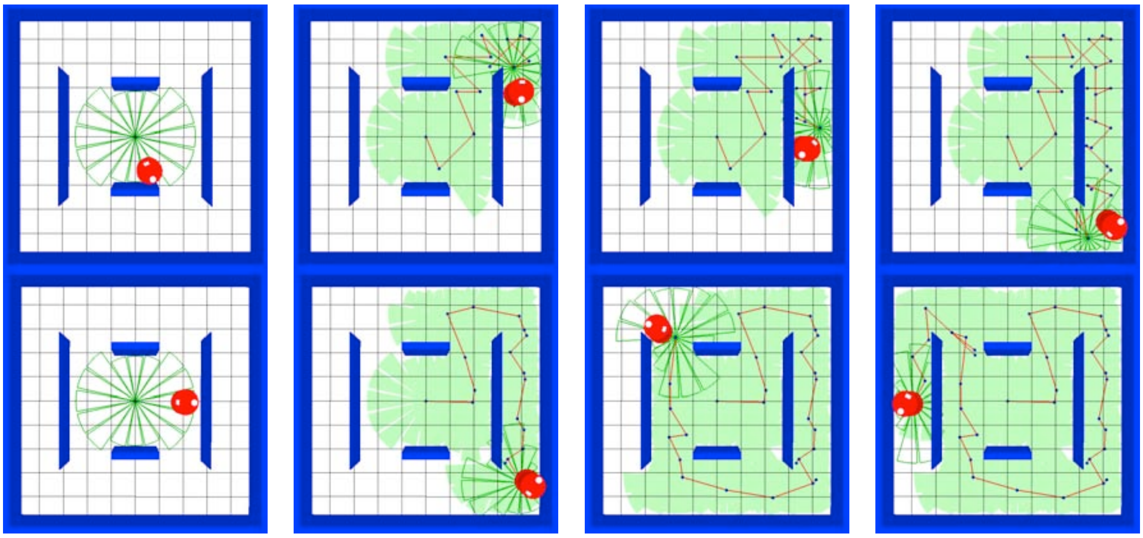

Fig. 10. Small environment: Exploration with SRT-Star (above) and FB_SRT-Star (below)

SRT-Star	traveled distance (m)	19.29	19.62	17.82	17.01	17.49	ave. 18.24
	filling	70.23%	64.76%	75.12%	74.23%	72.18%	ave. 71.30%
FB_SRT-Star	traveled distance (m)	16.30	13.75	15.04	15.98	16.85	ave. 15.58
	filling	86.51%	86.76%	88.95%	89.34%	86.43%	ave. 87.59%

TABLE III

RESULTS IN THE SMALL ENVIRONMENT (40 ITERATIONS)

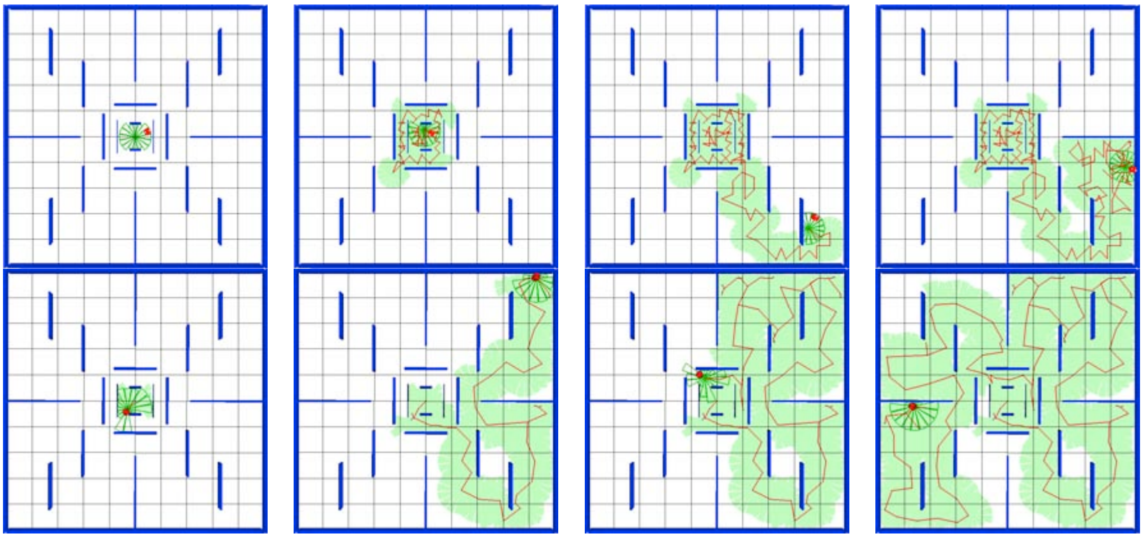

Fig. 11. Large environment: Exploration with SRT-Star (above) and FB_SRT-Star (below)

SRT-Star	traveled distance (m)	192.27	206.31	184.25	211.23	188.39	ave. 194.49
	filling	37.90%	39.16%	28.72%	36.24%	35.83%	ave. 35.48%
FB_SRT-Star	traveled distance (m)	267.91	278.32	284.76	292.77	265.93	ave. 277.93
	filling	77.19%	68.74%	69.48%	76.14%	73.33%	ave. 72.97%

TABLE IV

RESULTS IN THE LARGE ENVIRONMENT (200 ITERATIONS)