Proceedings of the 2006 IEEE/RSJ
International Conference on Intelligent Robots and Systems
October 9 - 15, 2006, Beijing, China

A Randomized Method for Integrated Exploration

Luigi Freda Francesco Loiudice Giuseppe Oriolo
Dipartimento di Informatica e Sistemistica
Università di Roma "La Sapienza"
Via Eudossiana 18, 00184 Roma, Italy
freda@dis.uniroma1.it, fra.esc@gmail.com, oriolo@dis.uniroma1.it

***Abstract*— We present an integrated exploration strategy for mobile robots. 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. A continuous localization procedure based on natural features of the safe region is integrated in the scheme. Both the information gain and the localization potential are taken into account when evaluating candidate configurations for exploration. Simulations and experiments on the MagellanPro robot show the performance of the proposed technique.**

I. Introduction

A fundamental requisite for autonomous navigation is the knowledge of an environment model, usually known as a *map*. Exploration is the basic task by which a robot puts together such a model through its sensory system for subsequent use. At each step, in order to be able to merge sensor data consistently in the current map, the robot position must be estimated; to this end, a concurrent localization process is performed. Hence, an efficient exploration strategy should take into account different, possibly conflicting criteria when selecting the next action: its cost (in terms of time or energy), the expected information gain and the associated localization potential.

Many exploration strategies choose the next action on the basis of only a subset of the above criteria. For example, the techniques proposed in [1], [2], [3], [4] can be classified as frontier-based, as they take into account — although in different forms — the information gain associated to each action and the final cost of the exploration. On the other hand, active localization techniques, such as [5], aim at increasing the localization quality along the chosen path without considering the associated cost or information gain. Indeed, most SLAM algorithms (e.g, see [6] and the references therein) do not deal with the planning task.

Recently, various researchers have emphasized the importance of integrating mapping, localization and planning. A common approach is the association of a utility function to each of these processes, in order to evaluate the contribution of candidate next actions towards the fulfillment of the corresponding task. Some sort of minimization of a mixed criterion (the total utility) combining the individual utility functions is then used to select the next action, possibly over a finite number of choices. An optimal strategy should maximize the expected utility over the whole exploration path, but the complexity of the problem, together with the lack of a priori information, suggests a more effective greedy approach, where the evaluation of the utility of an action is based on a single-step look-ahead. As a consequence, the selection of actions in integrated exploration strategies is inherently local. However, in principle, more intelligent decisions become possible as more information is gathered about the environment.

The paradigm of *integrated exploration* was first explicitly stated in [7]: the proposed method, based on an information-theoretic approach [8], associates to each possible robot destination an estimate of the map information gain as well as of the position certainty achievable at that point. These estimates are respectively used as mapping and localization utility functions; moreover, a navigation utility is added to the total function to make shorter paths more appealing. A similar approach is used in [9], where local decisions are taken in order to improve the estimates of the robot and landmark locations. In [10], another decision-theoretic approach is proposed, in which each action is selected in order to reduce the posterior uncertainty about the robot trajectory and resulting map. The adopted total utility function combines the expected change of entropy of the chosen Rao-Blackwellized particle filter and the cost of reaching the possible destination. Other related techniques include [11], where SLAM is performed using a parameterized class of spiral trajectories and maximizing a utility function which combines the navigation cost and the uncertainty of the adopted EKF filter, and [12], where the localization quality is indirectly taken into account by requiring a minimum overlap between the new expected observation and the available map.

This paper builds upon the randomized exploration method of [4] to present a frontier-based strategy for integrated exploration, in which the optimization of information gain and navigation cost are automatically taken into account by the local randomized strategy which proposes candidate destinations. In our view, this avoids the problematic definition of mixed criteria typical of other approaches, where the frontier is globally defined and an explicit penalty must be put on traveled distance in order to avoid erratic behaviors. Our approach includes a feature-based continuous localization scheme relying on a simple but effective extraction of characteristic points from the current range reading profile. The new destination is then selected so as to guarantee a minimum localization potential, which is related to the number of natural features of the map that will still be visible from there.

Another interesting aspect of our method is the use of very compact data structures. In particular, the map is built in the form of a Sensor-based Random Tree (SRT), in which

1-4244-0259-X/06/$20.00 ©2006 IEEE

```
SRT_INTEGRATED_EXPLORATION(q_init, K_max, ℓ_min, I_loc)
1   for k=1 to K_max                                              % at each step:
2     q̂ ← ODOMETRY;                                               % estimate the configuration from sensor data
3     S ← LSR;                                                    % estimate the Local Safe Region from sensor data
4     (q_curr, S, T) ← LOCALIZE(q̂,S,T);   % localize the robot comparing the LSR with the map, and update the map
5     F ← FRONTIER(q_curr,S,T);                                   % compute the local frontier
6     i ← 0;
7     if F ≠ ∅                                                    % if there exists at least a frontier arc:
8       do
9         φ_rand ← RANDOM_DIR(F);                                 % choose a random direction φ_rand toward a frontier arc
10        q_cand ← DISPLACE(q_curr, φ_rand);                      % generate q_cand by taking a feasible step along φ_rand
11        if i < I_loc
12          ℓ ← LOCALIZABILITY(q_cand,T);                         % compute the localization potential of q_cand
13        else
14          ℓ ← ℓ_min + 1;
15        i ← i + 1;
16      while ℓ > ℓ_min;                                          % until q_cand is acceptable for localization
17      MOVE_TO(q_cand);                                          % go there...
18      q_curr ← q_cand;                                          % ...and repeat the cycle
19    else
20      MOVE_TO(q_curr.parent);                                   % if there are no frontier arcs, backtrack
21      q_curr ← q_curr.parent;
22  return T;
```

Fig. 1. A description in pseudocode of the SRT-based integrated exploration algorithm.

each node contains a configuration assumed by the robot, the sensory data perceived from that location and the associated features. The localization and planning procedures are very simple and designed so as to require no additional elaboration of the map.

The paper is organized as follows. In the next section, we give an overview of our integrated exploration method. Sections III, IV and V discuss in detail the structure of the mapping, localization and planning processes, respectively. Simulation and experimental results supporting the proposed method are presented in Sect. VI.

II. SRT-BASED INTEGRATED EXPLORATION

In this section, we give a general outline of our integrated exploration method.

We shall take the following assumptions: *(i)* the robot moves in a planar workspace, i.e., $\mathbb{R}^2$ or a (connected) subset of $\mathbb{R}^2$; *(ii)* its configuration is described by $q = (x, y, \theta)$, with x, y the Cartesian coordinates of a representative point on the robot and θ its orientation, all measured w.r.t. a fixed world frame; *(iii)* an odometric estimate $\hat{q}$ of the robot configuration (to be corrected by localization) is continuously available; *(iv)* at each configuration, the sensory system provides the Local Safe Region (LSR), an estimate of the free space surrounding the robot, in the form of a star-shaped[1] subset $\mathcal{S}$ of $\mathbb{R}^2$.

The core of the method is the construction of a data structure called Sensor-based Random Tree (SRT). The SRT is stored in a tree called $\mathcal{T}$. Each node of $\mathcal{T}$ initially consists of an odometric estimate $\hat{q}$ of a configuration assumed by the robot, together with a description of the associated Local Safe Region $\mathcal{S}$. A continuous localization process, based on the comparison between $\mathcal{S}$ and $\mathcal{T}$, corrects $\hat{q}$ as well as $\mathcal{S}$ so as to achieve global consistency. The tree is incrementally built by extending the structure towards unexplored but localizable areas in such a way that the new configuration (and the path reaching it) is contained in $\mathcal{S}$.

The algorithm implementing the SRT method is shown in Fig. 1. While the detailed structure of each procedure will be discussed in the following sections, we highlight below the basic steps of the algorithm.

At each iteration of the algorithm, the ODOMETRY and LSR procedures are first executed: proprioceptive and exteroceptive sensor data are gathered and elaborated to compute $\hat{q}$ and the Local Safe Region $\mathcal{S}$. Relevant features of $\mathcal{S}$ are also identified for subsequent use.

A feature-based localization step is performed at this point. In particular, an association between features of $\mathcal{S}$ and features of previous LSRs (contained in $\mathcal{T}$) is performed, in order to identify which of the current features have already been observed from other locations. The odometric estimate $\hat{q}$ is then corrected to $q_{\rm curr}$ by 'realigning' the Local Safe Region $\mathcal{S}$ in such a way that associated features match as much as possible. The tree $\mathcal{T}$ is then updated by adding $q_{\rm curr}$ and the associated $\mathcal{S}$, and processed so as to recover a consistent representation of the environment. For example, when the robot completes a loop around an obstacle, a global consistent

[1]This means that $\mathcal{S}$ is homeomorphic to the closed unit ball and that the line segment connecting the robot representative point (x, y) to any point of $\mathcal{S}$ is completely contained in $\mathcal{S}$.

alignment of all the involved LSRs and their features is performed.

The next step is the computation of the local frontier $\mathcal{F}$, i.e., the portion of the boundary of the Local Safe Region $\mathcal{S}$ leading to unexplored areas. In general, $\mathcal{F}$ will be a collection of disjoint arcs. At this point, the procedure RANDOM_DIR selects first one of these arcs on the basis of the expected information gain, and then a direction of exploration ϕ_{rand} within the chosen arc. A candidate configuration q_{cand} is then generated by taking a step from q_{curr} in the direction of ϕ_{rand}. The stepsize is chosen as a fixed fraction of the extension of $\mathcal{S}$ in that particular direction; due to the star-shape of $\mathcal{S}$, q_{cand} will certainly be collision-free. If no frontier arc exists, the robot backtracks to the parent node of q_{curr} and the exploration cycle starts again.

Before moving to q_{cand}, the robot performs a validation step based on the LOCALIZABILITY function. This estimates the amount of information that q_{cand} can provide to the localization step, essentially based on the number of already observed features that will be visible from the new configuration. If this localization potential is sufficiently large, or if a maximum number of candidate configurations have been rejected, the robot moves to q_{cand} and the cycle is repeated.

When all the frontier arcs have been visited, the robot is forced to backtrack to the tree root (the starting configuration), thus realizing an automatic *homing* mechanism. Note the following points, which will be further clarified in the rest of the paper:

- The above method is a general paradigm. Its performance in practice will depend on the accuracy of localization and LSR reconstruction, and hence on the available sensory system.
- The Sensor-based Random Tree $\mathcal{T}$ plays the role of a *global map* in our approach, and hence it is often called so in the following. Note, however, that $\mathcal{T}$ is simply a collection of configurations with the associated Local Safe Regions (and features), made consistent by the localization process. No explicit 'fusion' of these regions is performed, because our method does not need it.
- In comparison with other frontier-based exploration methods (e.g., see [1]), the local frontier $\mathcal{F}$ is a subset of the boundary of the current Local Safe Region. This means that the robot will perform *depth-first* exploration as much as possible. In other approaches, the frontier is globally defined and an explicit penalty must be put on traveled distance in order to avoid erratic behaviors.

III. Construction of the Local Safe Region

The LSR procedure in Fig. 1 builds a star-shaped estimate $\mathcal{S}$ of the Local Safe Region currently surrounding the robot. In itself, $\mathcal{S}$ is merely an array of consecutive range readings obtained along different directions — it is therefore an *implicit* representation of the free space. Clearly, a laser range finder will produce a much more accurate LSR than, e.g., a ring of ultrasonic sensors. In any case, it should be considered that any range finder can measure distances up to a certain limit; if the range reading is equal to this maximum value, it represents a conservative estimate of the extension of free space in that direction.

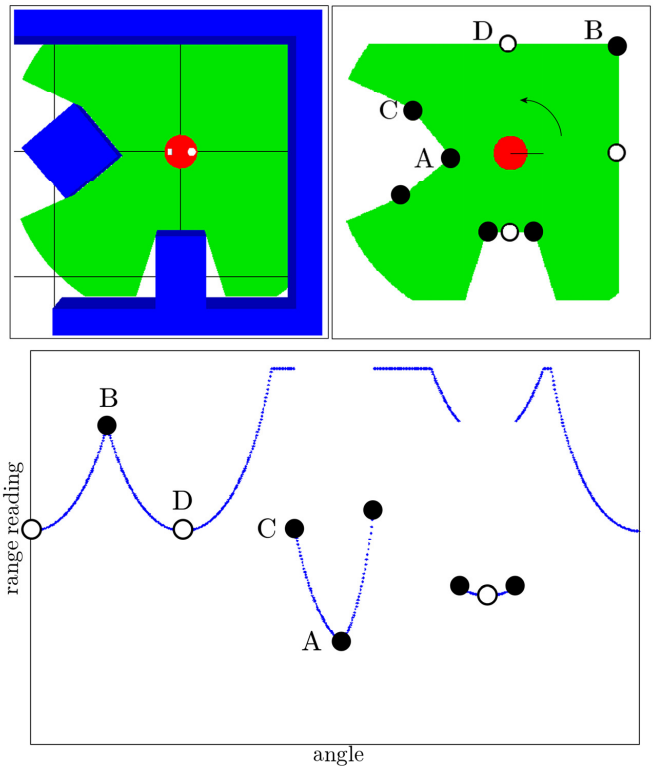

Fig. 2. The construction of a LSR and the identification of the features using a laser range finder. Above left: the LSR at the current robot configuration. Above right: the features include fixed features (black dots) and moving features (white dots). Below: the range reading as a function of the angle (measured counterclockwise with respect to the robot heading).

For localization purposes, it is necessary to extract natural features from $\mathcal{S}$. In particular, the LSR procedure identifies two kinds of features (see Fig. 2):

- *Fixed* features, corresponding to vertices of the Local Safe Region (indicated by a black dot in Fig. 2). These are identified from the range reading profile as either non-differentiable local minima/maxima (e.g., A and B) or jump discontinuities (such as C). In principle (i.e., for an ideal sensor), the position of these features does not depend on the observation point.
- *Moving* features, corresponding to locally minimum/maximum distance points along the boundary of Local Safe Region that are not vertices (indicated by a white dot in Fig. 2). These are identified from the range reading profile as differentiable local minima/maxima (e.g., D). As the robot observation point changes, the position of these features will obviously change as well.

Note that, in the case of a jump discontinuity, the feature must be placed in correspondence of the *smaller* range reading (see Fig. 2). Also, for obvious reasons, no feature is placed on those arcs of the LSR boundary which result from the maximum measurable range constraint.

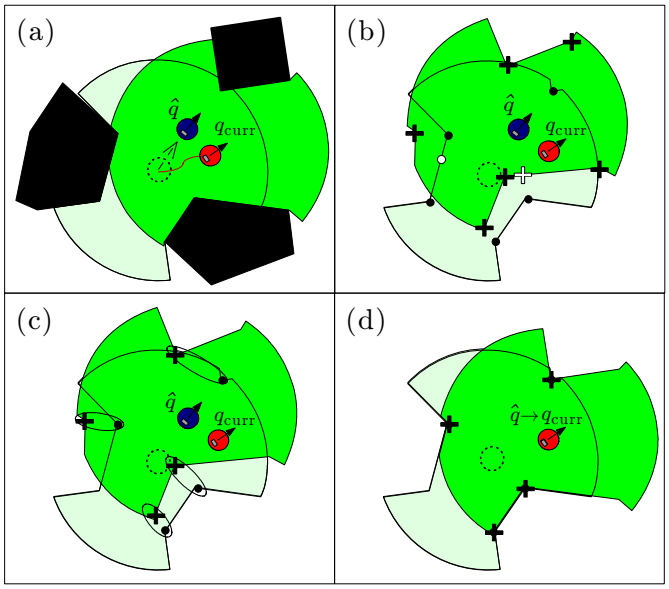

Fig. 3. The local correction step. From the previous configuration $q_{\rm ref}$ (shown in dotted line), the robot has traveled to the current configuration $q_{\rm curr}$, but the odometric estimate is $\hat{q}$ (a). The current LSR $\mathcal{S}$, which was in fact gathered at $q_{\rm curr}$, is spatially placed according to $\hat{q}$; fixed/moving features of $\mathcal{S}$ are shown by black/white crosses, while fixed/moving features of $\mathcal{S}_{\rm ref}$ are shown by black/white dots (b). Feature association between the two LSRs is then performed on the basis of the type and relative distance (c). The configuration error is then corrected in two phases (first angular and then translational) so as to maximize feature consistency (d).

At end of the process, the features extracted from $\mathcal{S}$ are collected in the set

$$\mathcal{P} = \{p_i, i = 1, \ldots, N\}$$

in which p_i is a polar representation of the i-th feature w.r.t. the frame attached to the robot.

IV. Localization

Assume that we have computed the odometric estimate $\hat{q}$, built the Local Safe Region $\mathcal{S}$ and extracted its relevant features. Denote by $q_{\rm ref}$ the last configuration assumed by the robot on its path before the current one, and by $\mathcal{S}_{\rm ref}$ the corresponding Local Safe Region. These data, stored in the global map $\mathcal{T}$, will be used as *reference* for the localization process, which is accomplished in two steps:

1) *local correction* modifies the odometric estimate $\hat{q}$ so as to recover feature consistency between $\mathcal{S}$ and $\mathcal{S}_{\rm ref}$;
2) *global correction* performs a global consistent alignment of the LSRs when loops are detected, i.e., features of $\mathcal{S}$ can be associated with features of $\mathcal{T}$ that do not belong to $\mathcal{S}_{\rm ref}$.

Both steps are discussed in some detail in the following.

A. Local correction

The local correction step, which involves two LSRs, i.e., $\mathcal{S}$ and $\mathcal{S}_{\rm ref}$, is achieved in three substeps: feature association, angular error correction and translational error correction.

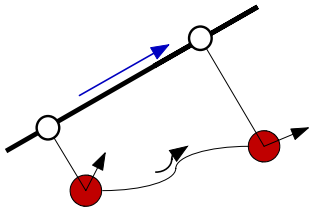

Fig. 4. Before attempting an association with $\mathcal{S}$, moving features of $\mathcal{S}_{\rm ref}$ are displaced according to the measured robot motion.

Refer to Fig. 3 for illustration. From the previous configuration $q_{\rm ref}$ (shown in dotted line), the robot has traveled to the current configuration $q_{\rm curr}$; the current odometric estimate is $\hat{q}$ (see Fig. 3a). The current LSR $\mathcal{S}$, which was in fact gathered at $q_{\rm curr}$, is spatially placed according to $\hat{q}$ (see Fig. 3b, where fixed/moving features of $\mathcal{S}$ are shown in by black/white crosses, while fixed/moving features of $\mathcal{S}_{\rm ref}$ are shown by black/white dots). Feature association between the two LSRs is then performed on the basis of the type and relative distance. In particular, to each feature of $\mathcal{S}$, stored in the $\mathcal{P}$ set, we associate the closest feature of the same type (i.e., fixed/moving) of $\mathcal{S}_{\rm ref}$ (see Fig. 3c). It should be noted that moving features of $\mathcal{S}_{\rm ref}$ are displaced according to the measured robot motion before attempting an association (see Fig. 4). Also, two features are not associated if their relative distance exceeds a certain threshold d_m.

The results of the association process are collected in a set $\mathcal{A}$, whose $N_a < N$ elements are pairs of associated features in the two LSRs. This set is used to correct the configuration error in two phases (first angular and then translational) as follows.

Let $q_{\rm curr} = (x_{\rm curr}, y_{\rm curr}, \theta_{\rm curr})$ be the current robot configuration and $\hat{q} = (\hat{x}, \hat{y}, \hat{\theta})$ its odometric estimate, with

$$\begin{aligned} x_{\rm curr} &= \hat{x} + e_x \\ y_{\rm curr} &= \hat{y} + e_y \\ \theta_{\rm curr} &= \hat{\theta} + e_\theta \end{aligned}$$

where e_x, e_y and e_θ are the estimation errors.

The angular correction e_θ is computed as follows. From the association set $\mathcal{A}$, we extract all the features of $\mathcal{S}_{\rm ref}$, join all the possible pairs of such features by line segments (obtaining thus a *complete feature graph*), and compute the vector of the angular coefficients of these segments. Repeating this operation for the corresponding features of $\mathcal{S}$ will clearly give a different vector due to the presence of an estimation error. The idea is then to correct the orientation estimate $\hat{\theta}$ in such a way that the norm of the difference between the two vectors is minimized in a least-squares sense.

In particular, given a generic LSR with its N features, let (ρ_k, ψ_k) be the polar coordinates of the k-th feature point p_k w.r.t. the frame attached to the robot configuration (x, y, θ) from which the LSR was gathered. Consider now a generic pair of features (p_i, p_j) belonging to the LSR, and build the line segment joining them. Its angular coefficient is obtained

as

$$\alpha_{ij} = \arctan \frac{\rho_j \sin(\psi_j + \theta) - \rho_i \sin(\psi_i + \theta)}{\rho_j \cos(\psi_j + \theta) - \rho_i \cos(\psi_i + \theta)} \qquad (1)$$

Assume now that the above computation is carried out for $q_{\rm ref}$ and $\mathcal{S}_{\rm ref}$, whose data are stored in the global map $\mathcal{T}$, using only the N_a features included in the association set, and denote the result by $\alpha_{ij}^{\rm ref}$. To compute the corresponding value $\alpha_{ij}^{\rm curr}$ for $\mathcal{S}$, it is necessary to use the current coordinates of the corresponding features and $\theta_{\rm curr} = \hat{\theta} + e_\theta$ in place of θ in eq. (1). As a consequence, $\alpha_{ij}^{\rm curr}$ for $\mathcal{S}$ will depend on e_θ only (and not on e_x, e_y), so that we may identify an optimal correction e_θ^* as

$$e_\theta^* = \arg\min_{e_\theta} \sum_{i=1}^{N_a} \sum_{j=i+1}^{N_a} c_{ij} \left(\alpha_{ij}^{\rm ref} - \alpha_{ij}^{\rm curr}\right)^2$$

with the weight c_{ij} depending on the length of the segment (the orientation of shorter segments is more sensitive to displacements of the endpoints) as well as the reliability of the feature pair (for example, pairs of fixed features are assigned a higher weight).

After the angular correction e_θ^* has been obtained, the translational corrections e_x and e_y are computed, using again the complete feature graph generated above. In particular, denote by $M_{ij}^{\rm ref} = (x_{ij}^{\rm ref}, y_{ij}^{\rm ref})$ the midpoint of the generic feature segment in $\mathcal{S}_{\rm ref}$, and by $M_{ij}^{\rm curr} = (x_{ij}^{\rm curr}, y_{ij}^{\rm curr})$ the corresponding midpoint in $\mathcal{S}$. Clearly, for a given e_θ, coordinates $x_{ij}^{\rm curr}$ and $y_{ij}^{\rm curr}$ depend only on e_x and e_y, respectively. Therefore, optimal estimates of these translational corrections are obtained as

$$e_x^* = \arg\min_{e_x} \sum_{i=1}^{N_a} \sum_{j=i+1}^{N_a} c_{ij} \left(x_{ij}^{\rm ref} - x_{ij}^{\rm curr}\right)^2$$

$$e_y^* = \arg\min_{e_y} \sum_{i=1}^{N_a} \sum_{j=i+1}^{N_a} c_{ij} \left(y_{ij}^{\rm ref} - y_{ij}^{\rm curr}\right)^2$$

Note the following points.

- While in principle the angular correction could be performed using a single segment (i.e., a single pair of features), the use of the complete feature graph corresponds to the incorporation of redundant information which allows to robustify the correction procedure in the presence of conflicting data in the two LRSs, i.e., features which cannot be matched via a rototranslation of $\hat{q}$.
- While the angular coefficient $\alpha_{ij}^{\rm curr}$ does not depend on e_x and e_y, the coordinates of $M_{ij}^{\rm curr}$ depend on e_θ as a matter of fact. The decoupled minimization entailed by the above procedure is only possible because it is assumed that e_θ^* has been computed in advance on the basis of the α_{ij}'s only. In other words, we have first computed a rotation of the estimated robot configuration so as to minimize the angular mismatch of the complete feature graph, and then a translation (at a fixed orientation) so as to minimize the linear mismatch.
- The above remark indicates that, in principle, the correction of the configuration should have been computed by simultaneous minimization of both criteria. We found, however, that the results were essentially the same of the decoupled minimization, at the expense of an remarkable increase in computational load. Note, in fact, that the three minimization problems which provide e_x^*, e_y^*, e_θ^* are one-dimensional and therefore efficiently solved by numeric line search procedures.

At the end of the local correction process, the positions of the associated features in $\mathcal{S}_{\rm ref}$ and $\mathcal{S}$ should in principle coincide, because the error associated to the odometric estimate has been corrected. In practice, however, a residual mismatch will invariably persist; this is due to the non-ideal behavior of any sensor, which makes it impossible to exactly match the features from different robot configurations. Therefore, to enforce a consistency between $\mathcal{S}$ and $\mathcal{S}_{\rm ref}$ in $\mathcal{T}$ — which represents the map built by the robot — the feature positions are suitably averaged and the LSRs themselves are 'trimmed' accordingly (details are omitted here for compactness, see [13]).

B. Global correction

Even with the local correction procedure, the robot configuration estimate may become imprecise and even diverge in the long run. As mentioned above, this is partly due to the uncertainty intrinsic to any sensing process. Another typical reason may be an insufficient number of associated features between $\mathcal{S}$ and $\mathcal{S}_{\rm ref}$. Clearly, a number of factors play a role here, including the characteristics of the environment (such as the number of features and the possibility of occlusions), the average distance between two consecutive configurations, and the sensor range limit.

One obvious way to improve the performance of localization is to exploit, if possible, information in the global map that is not contained in $\mathcal{S}_{\rm ref}$. Such process, called *global correction*, will increase the possibility of feature association as well as correct slow drifts such as those occuring when circumnavigating a large obstacle region (large loops).

The global correction step is executed whenever $\mathcal{S}$ contains features that can be associated (according to the procedure explained above) to features in the global map that are *not* contained in $\mathcal{S}_{\rm ref}$. For example, this happens when a large loop is completed around an obstacle, so that $\mathcal{S}$ overlaps with an earlier LSR which was detected and built at the start of the loop. If this is the case, the previously described angular and translation error correction is performed between the two LSRs (or any number of overlapping LSRs) in order to improve the current estimate $q_{\rm curr}$. Also, mutual consistency among the LSRs is recovered by feature averaging and LSR trimming, as explained at the end of Sect. IV-A.

Clearly, the updated information deriving from a global registration step should be back-propagated in order to preserve consistency of the global map $\mathcal{T}$. While a detailed description of this process is not given here for lack of space, we mention that it is based on a series of local registrations along a virtual path that backtracks from $\mathcal{S}$ up to the LSR (or the LSRs)

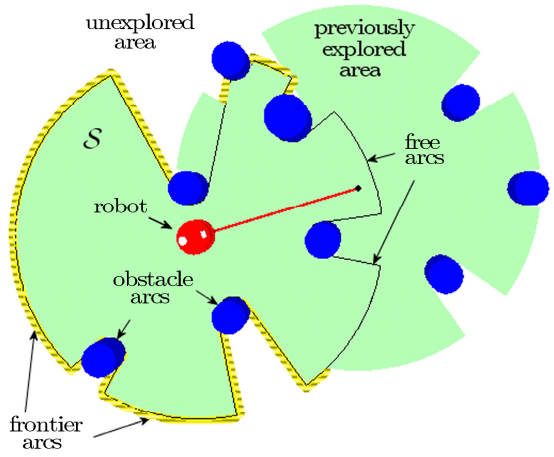

Fig. 5. The definition of frontier, free and obstacle arcs in the Local Safe Region $\mathcal{S}$.

involved in the global registration process. If a branching subtree is met on the backtracking path, the information is propagated in that direction as well (see [13]).

Another approach to global registration can be devised based on the work in [14]. This technique would require however to maintain all the original perceptions in memory, in order to be able to perform a simultaneous global correction which involves all the LSR's in $\mathcal{T}$.

V. Planning

In the integrated exploration approach, robot motion is incrementally planned in order to increase both the information gain and the possibility of localization (which is obviously related to the information gain in the long term). In our method, this is realized by a two-step procedure. First, a candidate configuration $q_{\rm cand}$ is generated according to a frontier-based strategy; then, its localization potential is computed and used to validate $q_{\rm cand}$.

A. Frontier-based generation of $q_{\rm cand}$

The central idea of *frontier-based* exploration is that, in order to maximize the information gain, the robot should approach the boundary between explored areas and unknown territory (called the *frontier*). Indeed, many exploration techniques belong to this family, e.g., see [1], [3], [7], [12]. Our SRT-based integrated exploration method adapts the same idea to a randomized setting by biasing the generation of configuration towards the frontier. However, differently from the above methods, we use a *local* frontier, defined as the intersection between the global frontier and the current LSR $\mathcal{S}$. This reduces the computational load associated with the identification of boundaries and, at the same time, enforces a depth-first exploration behavior that avoids erratic motions of the robot.

In order to identify unexplored directions in $\mathcal{S}$, its boundary is divided in *obstacle*, *free* and *frontier* arcs (see Fig. 5). Loosely speaking, an obstacle arc is located along the directions where an obstacle has been sensed, a free arc falls within other LSRs, while a frontier arc leads to an unexplored area. It is easy to devise an arc classification procedure (called FRONTIER in the pseudocode of Fig. 1) for the boundary of $\mathcal{S}$ directly based on the range reading profile [4].

If $\mathcal{S}$ has a frontier in the above sense, a random configuration $q_{\rm cand}$ is generated as follows (if no frontier arc exists, the robot backtracks). The RANDOM_DIR function selects an exploration direction $\phi_{\rm rand}$ by first choosing a frontier arc, with a probability proportional to its length, and then generating $\phi_{\rm rand}$ according to a normal distribution with mean value $\phi_{\rm m}$ (the orientation of the frontier arc bisectrix) and standard deviation $\gamma/6$ (γ being the central angle of the frontier arc). A candidate new configuration $q_{\rm cand}$ is then computed by the function DISPLACE along the selected direction. The stepsize used by DISPLACE is typically a fraction (say, 3/4) of the range reading in the direction of $\phi_{\rm rand}$.

B. Validation based on localizability

Once a candidate new configuration $q_{\rm cand}$ has been generated, it is evaluated under the viewpoint of *localizability*, i.e., an estimate of the possibility it offers of localizing the robot. In view of the feature-based approach described in Sect. IV, it is natural to relate this possibility to the number of features of $\mathcal{T}$ (i.e., of previous LSRs) that will be observable from $q_{\rm cand}$, and hence result in associated pairs. In particular, denoting this number by n_l, the LOCALIZABILITY procedure of Fig. 1 computes the localization potential as

$$\ell(q_{\rm cand}) = \sum_{i=1}^{n_l} w_i,$$

in which the weight w_i represents the 'reliability' of the i-th feature. For example, it is reasonable to assign a higher weight to fixed features and a lower weight to moving features, which are intrinsically more uncertain (typical values used in our experiments in the two cases are 1 and 0.5).

If the localizability of $q_{\rm cand}$ is above a minimum threshold[2] $\ell_{\rm min}$, the configuration is validated and the robot moves to $q_{\rm cand}$, where the exploration cycle is repeated. Otherwise, the previously selected frontier arc is marked as 'free', and a new configuration is generated by RANDOM_DIR and DISPLACE. If the number of candidate configurations that have not been validated reaches a maximum value $I_{\rm loc}$, the localizability criterion is discarded.

VI. Simulations and Experiments

In this section we present simulation and experimental results obtained using the presented SRT-based integrated exploration method.

A. Simulations

We have simulated the proposed method in Move3D [15], a software platform developed at LAAS-CNRS and dedicated

[2]Clearly, $\ell_{\rm min}$ should depend on the average number of features visible in the specific scene under consideration. This allows the exploration to proceed even in environments that are poor under this viewpoint.

to motion planning[3]. The simulated robot is a MagellanPro with an onboard 360° laser range finder: angular and linear resolutions of the simulated sensor are respectively 1° and 0.5 cm, and the maximum range is set at 2 m. In order to include an odometry drift, the robot configuration at each step is perturbed by a gaussian noise with a standard deviation chosen as 3% of the nominal displacement.

The simulated environment is depicted in Fig. 6, and contains a corridor (approximately 12 m long) and two loops. The final resulting map is shown in Fig. 7, which also reports for comparison the map obtained using SRT-based exploration without localization (odometric estimates only). The robot has successfully explored the whole environment; the final localization error at the homing position is less than 1 cm for an exploration path longer than 100 m.

Figure 8 shows two snapshots of the map building and localization progress. On the left, the currently perceived LSR (in darker tone) is placed according to the odometric configuration estimate. On the right, the LSR is realigned to the map after the localization process.

B. Experiments

The SRT integrated exploration strategy has 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. The robot carries an onboard SICK LMS 200 laser range finder with a 1° angular resolution (see Fig. 9). During the experiments, the built-in dead reckoning system is used to retrieve an odometric estimation of the current configuration.

Each Local Safe Region is built merging three different laser scans of 180° acquired at the same place with orientations spaced at 120° increments. In order to construct a consistent 360° LSR, each 180° scan is merged with the others by means of a point-based matching algorithm [16], [17].

The results of a typical experiment results are shown in Figs. 10 and 11. Here, MagellanPro must explore a loop-shaped environment whose perimeter is approximately 40 m. In particular, Figure 10 shows the final resulting map compared with the map obtained using SRT-based exploration without localization (odometric estimates only). The exploration process, which starts from the configuration in the lower-left quadrant, is successful. The total traveled distance (including homing) is approximately 18 m, with a final positioning error of less than 3 cm. It should be mentioned that, when SRT-based exploration without localization was used, the robot was not able to backtrack to the home position because, due to the odometric drift, it collided with an obstacle (note that no obstacle avoidance module is included in our system).

Figure 11 shows two snapshots of the map building and localization progress. On the left, the currently perceived LSR (in darker tone) is placed according to the odometric configuration estimate. On the right, the LSR is satisfactorily realigned to the map after the localization process.

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

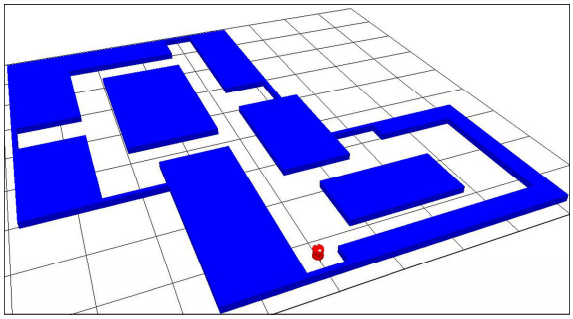

Fig. 6. The simulated environment.

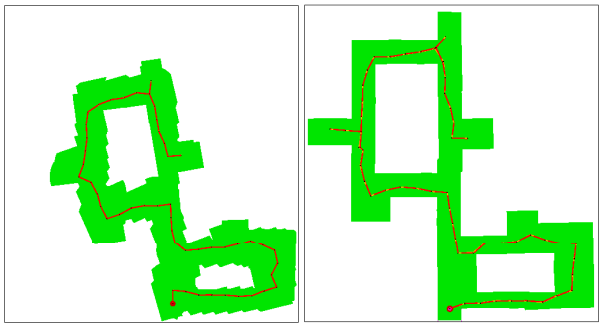

Fig. 7. Simulation: Final maps. Left: SRT-based exploration without localization (odometric estimates only). Right: SRT-based integrated exploration.

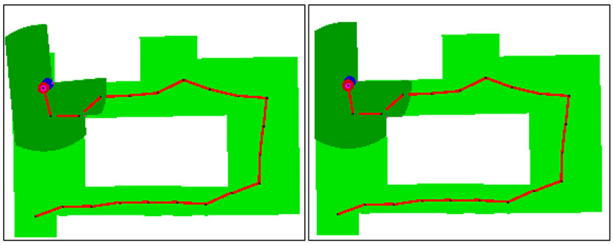

Fig. 8. Simulation: Two snapshots of the map building and localization progress. Left: the currently perceived LSR (in darker tone) is placed according to the odometric configuration estimate. Right: the LSR is realigned to the map after the localization process.

VII. Conclusions

We have presented a novel strategy for integrated exploration.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. A continuous localization procedure based on natural features is included. Both the information gain and the localization potential are taken into account in evaluating candidate configurations for exploration. Simulations and experiments on the MagellanPro robot have been presented to show the performance of the proposed technique.

In our view, the presented method presents two main advantages w.r.t. other techniques:

- The information gain, navigation cost and localization potential of each candidate action are automatically taken

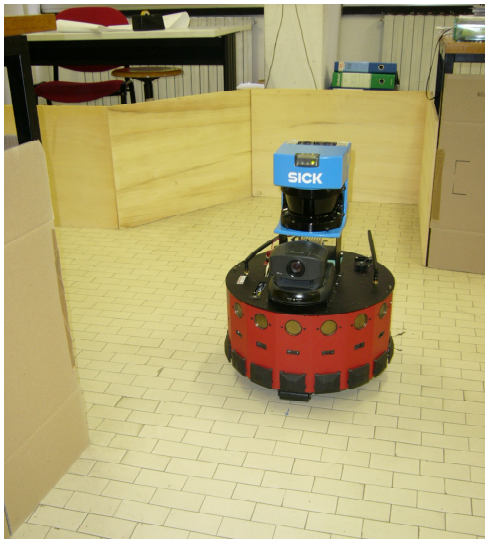

Fig. 9. The robot MagellanPro in the experimental environment.

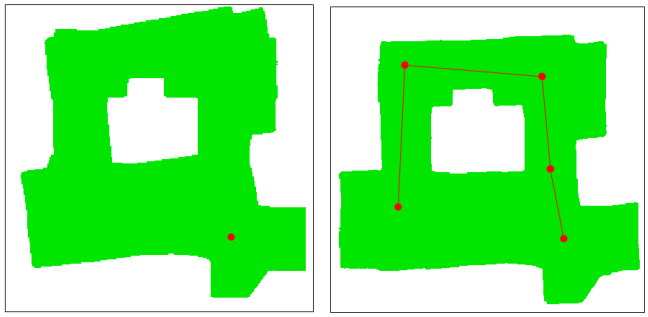

Fig. 10. Experiment: Final maps. Left: SRT-based exploration without localization (odometric estimates only). Right: SRT-based integrated exploration.

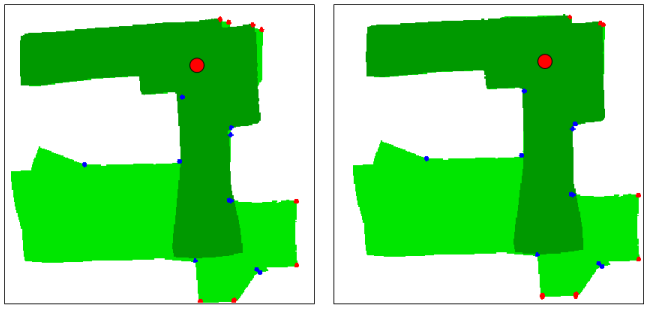

Fig. 11. Experiment: Two snapshots of the map building and localization progress. Left: the currently perceived LSR (in darker tone) is placed according to the odometric configuration estimate. Right: the LSR is realigned to the map after the localization process.

ito account by our local planner, thus avoiding the problematic definition of mixed criteria typical of other methods.

- The use of a very compact data structure (the Sensor-based Random Tree) allows to design simple localization and planning procedures which do not require additional elaboration of the map.

We are currently working in order to improve the localization process. The local correction can be achieved more robustly in two steps: first the feature-based matching algorithm described in Sect. IV-A is applied; then, an iterative point-based algorithm (e.g., the IDC algorithm in [16]), which does not depend on feature extraction, is used to refine the first correction. At the same time, following the approach in [14], a network of pose relations can be built which stores all the pose estimates and the relative covariances coming from the history of applied local corrections. When a loop closure is detected, a global alignment of the LSRs should be performed minimizing an energy function associated to the currently built network of pose relations.

REFERENCES

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

[2] 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.

[3] 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.

[4] L. Freda and G. Oriolo, "Frontier-based probabilistic strategies for sensor-based exploration," in *IEEE Int. Conf. on Robotics and Automation*, 2005, pp. 3892–3898.

[5] L. Kaelbling, A. Cassandra, and J. Kurien, "Acting under uncertainty: Discrete bayesian models for mobile robot navigation," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 1996.

[6] D. Hähnel, W. Burgard, D. Fox, and S. Thrun, "An efficient fastslam algorithm for generating maps of large-scale cyclic environments from raw laser range measurements," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2003.

[7] 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.

[8] F. Bourgault, A. Makarenko, S. Williams, B. Grocholsky, and H. Durrant-Whyte, "Information based adaptive robotic exploration," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2002.

[9] H. Feder, J. Leonard, and C. Smith, "Adaptive mobile robot navigation and mapping," *Int. J. Robotics Research*, vol. 18, no. 7, pp. 650–668, 1999.

[10] C. Stachniss, G. Grisetti, and W. Burgard, "Information gain-based exploration using rao-blackwellized particle filters," in *IEEE Int. Conf. on Robotics and Automation*, 2005.

[11] R. Sim, G. Dudek, and N. Roy, "Online control policy optimization for minimizing map uncertainty during exploration," in *IEEE Int. Conf. on Robotics and Automation*, 2004.

[12] 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.

[13] L. Freda, F. Loiudice, and G. Oriolo, "A randomized method for integrated exploration," Tech. Rep., Università di Roma "La Sapienza", 2006.

[14] F. Lu and E. Milios, "Globally consistent range scan alignment for environment mapping," *Autonomous Robots*, vol. 4, pp. 333–349, 1997.

[15] 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.

[16] F. Lu and E. Milios, "Robot pose estimation in unknown environments by matching 2d range scans," in *CVPR94*, 1994, pp. 935–938.

[17] A. Diosi and L. Kleeman, "Laser scan matching in polar coordinates with application to slam," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2005, pp. 1439–1444.