The 2009 IEEE/RSJ International Conference on
Intelligent Robots and Systems
October 11-15, 2009 St. Louis, USA

An Exploration Method for General Robotic Systems Equipped with Multiple Sensors

Luigi Freda, Giuseppe Oriolo and Francesco Vecchioli

Abstract— **This paper presents a novel method for sensor-based exploration of unknown environments by a general robotic system equipped with multiple sensors. The method is based on the incremental generation of a configuration-space data structure called Sensor-based Exploration Tree (SET). The expansion of the SET is driven by information at the world level, where the perception process takes place. In particular, the frontiers of the explored region efficiently guide the search for informative view configurations. Different exploration strategies may be obtained by instantiating the general SET method with different sampling techniques. Two such strategies are presented and compared by simulations in non-trivial 2D and 3D worlds. A completeness analysis of SET is given in the paper.**

I. INTRODUCTION

This paper presents a novel exploration method by which a general robotic system equipped with multiple sensors can explore an unknown environment. The method is suitable for generic robotic systems (such as fixed or mobile manipulators, wheeled or legged mobile robots, flying robots), equipped with any number of range finders.

In a *sensor-based exploration*, the robot is required to 'cover' the largest possible part of the world with sensory perceptions. A considerable amount of literature addresses this problem for single-body mobile robots equipped with one sensor, typically an omnidirectional laser range finder. In this context, *frontier-based strategies* [1]–[5] are an interesting class of exploration algorithms. These are based on the idea that the robot should approach the boundary between explored and unexplored areas of the environments in order to maximize the expected utility of robot motions.

The problem of exploring an unknown world using a multi-body robotic system equipped with multiple sensors is more challenging. In fact, the sensing space (the world) and the planning space (the configuration space) are very different in nature: the former is a Euclidean space of dimension 2 or 3, while the latter is a manifold in general with dimension given by the number of configuration coordinates, typically 6 or more. While frontiers at the world level clearly retain their informative value, using this information to efficiently plan actions in configuration space is not straightforward.

In the literature, few works exist that address the sensor-based exploration problem for articulated structures, mainly for fixed-base manipulators equipped with a single sensor, e.g., see [6]–[9]. A related problem is 3D object reconstruction and inspection [10].

The SET (Sensor-based Exploration Tree) method, which was originally presented in [11] for single-sensor robotic systems, is a frontier-based exploration method. The basic idea is to guide the robot so as perform a depth-first exploration of the world, progressively sensing regions that are contiguous from the viewpoint of sensor location. In this process, frontiers are used to efficiently identify informative configurations. The information gathered about the free space is mapped to a configuration space roadmap which is incrementally expanded via a sampling-based procedure. The roadmap is used to select the next view configuration, which is added to the SET. In the exploration process, the robot alternates forwarding/backtracking motions on the SET, which essentially acts as an Ariadne's thread.

In this work, we present *(i)* an extension of the SET method to multi-sensor robotic systems *(ii)* a completeness analysis of the algorithm *(iii)* a SET implementation on non-trivial 2D and 3D worlds. In particular, we discuss how to identify which frontiers are relevant for guiding the perception of each sensor and how to assign priorities to the sensors during view planning.

The paper is organized as follows. The problem setting is given in Sect. II. A general exploration method is outlined in Sect. III and the SET method is presented in Sect. IV. Simulation results in different worlds are reported and discussed in Sect. VI. Some extensions of the present work are mentioned in the concluding section.

II. PROBLEM SETTING

The robot wakes up in a unknown world populated by obstacles. Its task is to perform an exploration, i.e. cover the largest possible part of the world with sensory perceptions [12].

A. Robot and World Models

The robot, denoted by $\mathcal{A}$, is a kinematic chain of r rigid bodies ($r \geq 1$) interconnected by elementary joints. This description includes: fixed-base manipulators, single-body and multiple-body mobile robots, flying robots, humanoids and mobile manipulators.

The *world* $\mathcal{W}$ is a compact connected subset of $I\!R^N$, with $N = 2, 3$. It represents the physical space in which the robot moves and acquires perceptions. $\mathcal{W}$ contains the static obstacles $\mathcal{O}_1, ..., \mathcal{O}_p$, each a compact connected subset of $\mathcal{W}$. One of these obstacles is the world boundary $\partial\mathcal{W}$ which is considered as a 'fence'. Denoting by $\mathcal{O} = \bigcup_{i=1}^{m} \mathcal{O}_i$ the *obstacle region*, the *free world* is $\mathcal{W}_{\text{free}} = \mathcal{W} \setminus \mathcal{O}$.

Luigi Freda, Giuseppe Oriolo and Francesco Vecchioli are with Dipartimento di Informatica e Sistemistica, Università di Roma "La Sapienza", Via Ariosto 25, 00185 Roma, Italy, {freda,oriolo}@dis.uniroma1.it, francesco_vecchioli@fastwebnet.it

This work has been funded by the European Commission's Sixth Framework Programme as part of the project PHRIENDS under grant no. 045359.

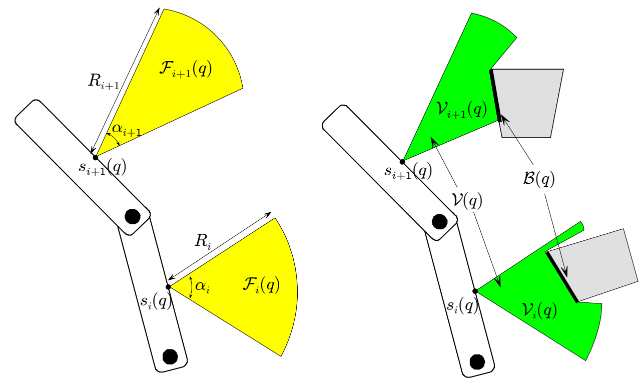

Fig. 1. *Left*: sensor centers $s_i(q)$ and $s_{i+1}(q)$, and the associated fields of view $\mathcal{F}_i(q)$ and $\mathcal{F}_{i+1}(q)$ when the robot is at configuration q. *Right*: The view $\mathcal{V}(q)$ and the visible obstacle boundary $\mathcal{B}(q)$.

The robot *configuration space* has dimension n and is denoted by $\mathcal{C}$, while q is a robot configuration. Let $\mathcal{A}(q)$ be the compact region of $\mathcal{W}$ occupied by the robot and its sensors at q. The *C-obstacle region* $\mathcal{CO}$ is the set of q such that $\mathcal{A}(q) \cap \mathcal{O} \neq \emptyset$. The *free configuration space* is $\mathcal{C}_{\text{free}} = \mathcal{C} \setminus \mathcal{CO}$.

B. Sensor Model

The robot is equipped with a system of m exteroceptive sensors, whose operation is formalized as follows.

Assuming[1] that the robot is at q, denote by $\mathcal{F}_i(q) \subset I\!R^N$ the compact region occupied by the i-th sensor *field of view*, which is *star-shaped* with respect to the sensor *center* $s_i(q) \in \mathcal{W}$. In $I\!R^2$, for instance, $\mathcal{F}_i(q)$ can be a circular sector with apex $s_i(q)$, opening angle α_i and radius R_i, where the latter is the perception range (see Fig. 1, left). The (total) *sensor field* at q is $\mathcal{F}(q) = \bigcup_{i=1}^m \mathcal{F}_i(q)$.

With the robot at q, a point $p \in \mathcal{W}$ is said to be *visible from the i-th sensor* if $p \in \mathcal{F}_i(q)$ and the open line segment joining p and $s_i(q)$ does not intersect $\partial\mathcal{O} \cup \partial\mathcal{A}(q)$. Denote by $\mathcal{V}_i(q)$ the points of $\mathcal{W}_{\text{free}}$ that are visible from the i-th sensor. At each configuration q, the robot sensory system returns (see Fig. 1, right):

- the *visible free region* (or *view*) $\mathcal{V}(q) = \bigcup_{i=1}^m \mathcal{V}_i(q)$;
- the *visible obstacle boundary* $\mathcal{B}(q) = \partial\mathcal{O} \cap \partial\mathcal{V}(q)$, i.e., all points of $\partial\mathcal{O}$ that are visible from at least one sensor.

The above sensor is an idealization of a 'continuous' range finder. For example, it may be a rotating laser range finder, which returns the distance to the nearest obstacle point along the directions (*rays*) contained in its field of view (with a certain resolution). Another sensory system which satisfies the above description is a stereoscopic camera.

C. Exploration task

The robot explores the world through a sequence of view-plan-move actions. Each configuration where a view is acquired is called a *view configuration*. Let q^0 be the initial robot configuration and $q^1, q^2, ..., q^k$ the sequence of view configurations up to the k-th exploration step. When the exploration starts, all the initial robot endogenous knowledge can be expressed as

$$\mathcal{E}^0 = \mathcal{A}(q^0) \cup \mathcal{V}(q^0), \quad (1)$$

where $\mathcal{A}(q^0)$ represents the free volume[2] that the robot body occupies (computed on the basis of proprioceptive sensors) and $\mathcal{V}(q^0)$ is the view at q^0 (provided by the exteroceptive sensors). At step $k \geq 1$, the *explored region* is

$$\mathcal{E}^k = \mathcal{E}^{k-1} \cup \mathcal{V}(q^k).$$

At each step k, $\mathcal{E}^k \subseteq \mathcal{W}_{\text{free}}$ is the current estimate of the free world. Since safe planning requires $\mathcal{A}(q^k) \subset \mathcal{E}^{k-1}$ for any k, we have

$$\mathcal{E}^k = \mathcal{A}(q^0) \cup \left(\bigcup_{i=0}^{k} \mathcal{V}(q^i) \right). \quad (2)$$

A point $p \in \mathcal{W}_{\text{free}}$ is defined *explored* at step k if it is contained in $\mathcal{E}^k$ and *unexplored* otherwise. A configuration q is *safe* at step k if $\mathcal{A}(q) \subset \text{cl}(\mathcal{E}^k)$, where $\text{cl}(\cdot)$ indicates the set closure operation (configurations that bring the robot in contact with obstacles are allowed). The *safe region* $\mathcal{S}^k \subseteq \mathcal{C}_{\text{free}}$ collects the configurations that are safe at step k, and represents a configuration space image of $\mathcal{E}^k$. A path in $\mathcal{C}$ is *safe* at step k if it is completely contained in $\mathcal{S}^k$.

The goal of the exploration is to expand $\mathcal{E}^k$ as much as possible as k increases [12].

III. EXPLORATION METHODS

Assume that the robot can associate an information gain $I(q, k)$ to any (safe) q at step k. This is an estimate of the world information which can be discovered at the current step by acquiring a view from q.

Consider the k-th exploration step, which starts with the robot at q^k. Let $\mathcal{Q}^k \subset \mathcal{S}^k$ be the *informative safe region*, i.e. the set of configurations which *(i)* have non-zero information gain, and *(ii)* can be reached[3] from q^k through a path that is safe at step k. A general exploration method (Fig. 2) searches for the next view configuration in $\mathcal{Q}^k \cap \mathcal{D}(q^k, k)$, where $\mathcal{D}(q^k, k) \subseteq \mathcal{C}$ is a compact *admissible set* around q^k at step k, whose size determines the scope of the search. For example, if $\mathcal{D}(q^k, k) = \mathcal{C}$, a global search is performed, whereas the search is local if $\mathcal{D}(q^k, k)$ is a neighborhood of q^k.

If $\mathcal{Q}^k \cap \mathcal{D}(q^k, k)$ is not empty, q^{k+1} is selected in it according to some criterion (e.g., information gain maximization). The robot then moves to q^{k+1} to acquire a new view (*forwarding*). Otherwise, the robot returns to a previously visited q^b ($b < k$) such that $\mathcal{Q}^k \cap \mathcal{D}(q^b, k)$ is not empty (*backtracking*). Given that the world is static, it is not necessary to acquire again a view from q_b. Hence, the actual number of views gathered so far may be less than k.

[1] The sensor placement is determined by the robot configuration q. Hence, for each sensor that is not rigidly attached to the robot (e.g., that can independently rotate around a certain axis, or is mounted on a pan-tilt platform), it is necessary to include the corresponding dof's in q.

[2] Often, $\mathcal{A}(q^0)$ in (1) is replaced by a larger free volume $\widetilde{\mathcal{A}}$ whose knowledge comes from an external source. This may be essential for planning safe motions in the early stages of an exploration.

[3] The reachability requirement accounts for possible kinematic constraints to which robot may be subject.

```
GENERAL EXPLORATION METHOD
if Q^k ∩ D(q^k, k) ≠ ∅                                   %forwarding%
   choose new q^{k+1} in Q^k ∩ D(q^k, k)
   move to q^{k+1} and acquire sensor view
else                                                     %backtracking%
   choose visited q^b (b < k) such that Q^k ∩ D(q^b, k) ≠ ∅
   move to q^b
```

Fig. 2. The k-th step of a general exploration method.

The exploration can be considered *completed* at step k if $\mathcal{Q}^k = \emptyset$, i.e., no informative configuration can be safely reached.

To specify an exploration method, one must define:

- an information gain;
- a forwarding selection strategy;
- an admissible set $\mathcal{D}(q^k, k)$;
- a backtracking selection strategy.

Due the complexity associated to the computation of $\mathcal{S}^k$, an efficient procedure to predict whether $\mathcal{Q}^k \cap \mathcal{D}(q^k, k)$ is non-empty or not would be useful.

IV. THE SET METHOD

In the SET method, the robot incrementally builds the *Sensor-based Exploration Tree* (SET) data structure. Each node of the SET represents a view configuration, while an arc between two nodes is a safe path joining them. A pseudocode description of the k-th step of SET is given in Fig. 3. A comparison with the general exploration method of Fig. 2 already suggests the specific choices that were made. These are detailed in the following.

A. Information Gain

At step k, the boundary of the explored region $\partial\mathcal{E}^k$ is the union of two *disjoint* sets:

- the *obstacle boundary* $\partial\mathcal{E}^k_{\text{obs}}$, i.e., the part of $\partial\mathcal{E}$ which consists of detected obstacle surfaces;
- the *free boundary* $\partial\mathcal{E}^k_{\text{free}}$, i.e., the complement of $\partial\mathcal{E}^k_{\text{obs}}$, which leads to potentially explorable areas.

One has $\partial\mathcal{E}^k_{\text{obs}} = \bigcup_{i=0}^{k} \mathcal{B}(q^i)$ and $\partial\mathcal{E}^k_{\text{free}} = \partial\mathcal{E}^k \setminus \partial\mathcal{E}^k_{\text{obs}}$.

Let $\mathcal{V}(q, k)$ be the *simulated view*, i.e., the view which would be acquired from q if the obstacle boundary were $\partial\mathcal{E}^k_{\text{obs}}$. The information gain $I(q, k)$ is defined as the measure of the set of unexplored points lying in $\mathcal{V}(q, k)$ [3], [7]. The SET method also makes use of the partial versions of these concepts, denoted respectively by $\mathcal{V}_j(q, k)$ and $I_j(q, k)$, which only consider the contribution of the j-th sensor. While $\mathcal{V}(q, k) = \bigcup_{j=1}^{m} \mathcal{V}_j(q, k)$, it is $I(q, k) \neq \sum_{j=1}^{m} I_j(q, k)$, since partial simulated views may overlap.

Finally, let $\mathcal{Q}^k_j = \{q \in \mathcal{Q}^k \,|\, I_j(q, k) \neq 0\}$ be the *partial informative safe region* of the j-th sensor. It is $\mathcal{Q}^k = \bigcup_{j=1}^{m} \mathcal{Q}^k_j$.

B. Forwarding Selection Strategy

If the condition of line 1 is met (see Sect. IV-D), q^{k+1} is selected in $\mathcal{D}(q^k, k) \cap \mathcal{Q}^k$ so as to maximize the utility function $U(q, k) = I(q, k)$ (line 2). A maximum certainly exists because $\mathcal{D}(q^k, k) \cap \mathcal{Q}^k$ is compact and $I(q, k)$ is continuous in q. In principle, the navigation cost from q^k to q^{k+1} could be included in U, to avoid erratic behaviors. However, our definition of $\mathcal{D}(q^k, k)$ together with the adopted search strategy (Sect. V-A) will give the same result.

```
SET METHOD
 1: if local free boundary L(q^k, k) is non-empty
 2:    (q^{k+1}, U^{k+1}) ← search configuration with
       maximum utility in D(q^k, k) ∩ Q^k
 3: if U^{k+1} > 0                                  %forwarding%
 4:    plan a safe path from q^k to q^{k+1}
 5:    move to q^{k+1} and acquire sensor view
 6:    update SET and world model
 7: else
 8:    if U^k is not empty                          %backtracking%
 9:       select the closest configuration q^b in U^k
10:       plan a path on SET leading to q^b
11:       move to q^b
12:    else
13:       homing
```

Fig. 3. The k-th step of the SET method.

C. Admissible Set

To simplify the notation, we assume below that all the sensors have the same perception range R. This does not imply any loss of generality.

Denote by $\mathcal{D}_{r,j}(q, k)$ the *partial admissible set (r,j)* around q at step k defined as the set of configurations w such that *(i)* the r-th sensor center $s_r(w)$ is contained in a ball $B(s_j(q), \rho)$ with radius $\rho \geq R$ and center $s_j(q)$, and *(ii)* $s_r(w)$ and $s_j(q)$ are *mutually visible* at step k, i.e., the open line segment $(s_r(w), s_j(q))$ does not intersect $\partial\mathcal{E}^k$. In this definition, the j-th sensor center $s_j(q)$ acts as a 'fixed pole' while the r-th sensor center $s_r(w)$ can 'move' in $B(s_j(q), \rho)$ as long as it remains visible from $s_j(q)$.

The *admissible set* $\mathcal{D}(q, k)$ around q at step k is defined as:

$$\mathcal{D}(q, k) = \bigcup_{r,j \in \{1,2,\ldots,m\}} \left(\mathcal{D}_{r,j}(q, k) \cap \mathcal{Q}^k_r \right). \tag{3}$$

Proposition 4.1: The admissible set (3) is such that:

$$\mathcal{Q}^k = \bigcup_{i=0}^{k} \mathcal{D}(q^i, k) \tag{4}$$

Proof: See [12]. ∎

D. Local Free Boundary

The SET method looks at a subset of the free boundary $\partial\mathcal{E}^k_{\text{free}}$ for predicting if $\mathcal{Q}^k \cap \mathcal{D}(q^k, k) = \emptyset$. This results in a significant computational saving, because $\partial\mathcal{E}^k_{\text{free}}$ has dimension $N - 1$ whereas $\mathcal{Q}^k \cap \mathcal{D}(q^k, k)$ has dimension n.

Let $\mathcal{L}_j(q, k)$ be the *partial local free boundary* of the j-th sensor around q at step k (Fig. 4), i.e., the set of points of the free boundary $\partial\mathcal{E}^k_{\text{free}}$ that *(i)* are contained in a ball $B(s_j(q), \rho + R)$ with center $s_j(q)$ and radius $\rho + R$, and *(ii)* can be connected to $s_j(q)$ through a world path contained in $\mathcal{E}^k \cap B(s_j(q), \rho + R)$. The parameter ρ of this definition is inherited from the partial admissible sets definition.

The *local free boundary* $\mathcal{L}(q, k)$ is defined as

$$\mathcal{L}(q, k) = \bigcup_{j=1}^{m} \mathcal{L}_j(q, k). \tag{5}$$

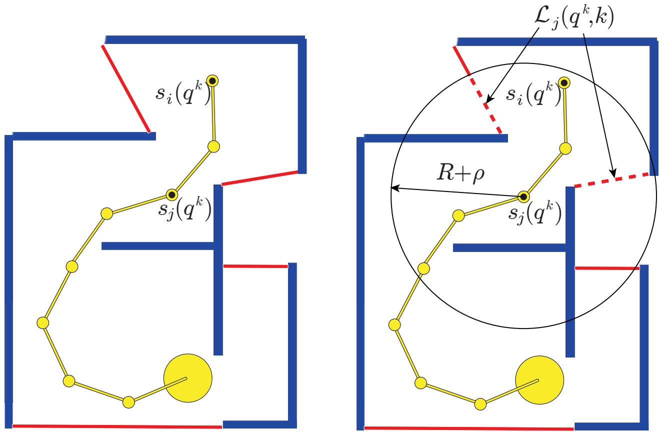

Fig. 4. A reconstructed world model at step k. *Left*: free boundary $\partial\mathcal{E}^k_{\text{free}}$ (red-thin) and obstacle boundary $\partial\mathcal{E}^k_{\text{obs}}$ (blue-thick). The boundary of $\mathcal{E}^k$ is $\partial\mathcal{E}^k = \partial\mathcal{E}^k_{\text{free}} \cup \partial\mathcal{E}^k_{\text{obs}}$. *Right*: the partial local free boundary $\mathcal{L}_j(q^k, k)$ consists in the two dashed segments.

$\mathcal{L}(q^k, k) \neq \emptyset$ is a necessary condition for $\mathcal{D}(q^k, k) \cap \mathcal{Q}^k$ to be non-empty, as shown by

Proposition 4.2: The following implication holds:

$$\mathcal{L}(q^k, k) = \emptyset \;\Rightarrow\; \mathcal{D}(q^k, k) \cap \mathcal{Q}^k = \emptyset \qquad (6)$$

Proof: See [12]. ■

If $\mathcal{L}(q^k, k) \neq \emptyset$ a search for a new view configuration is attempted[4] in $\mathcal{D}(q^k, k) \cap \mathcal{Q}^k$ (lines 1–2); otherwise no search is performed, U^{k+1} remains zero and the utility check (line 3) is negative.

E. Backtracking Selection Strategy

When the search of line 2 returns $U^{k+1} = 0$, the set $\mathcal{D}(q^k, k) \cap \mathcal{Q}^k$ is empty and a backtracking from q^k is attempted (line 7).

Let $l(k) \leq k$ be the last exploration step in which a view was acquired. Let $\mathcal{U}^k$ be the set of view configurations q^i such that *(i)* $\mathcal{L}(q^i, k) \neq \emptyset$, and *(ii)* no search has been performed in $\mathcal{D}(q^i, j) \cap \mathcal{Q}^j$ at a step $j > l(k)$ returning $U^{j+1} = 0$.

If $\mathcal{U}^k$ is not empty, the closest view configuration q^b in $\mathcal{U}^k$ is selected as destination (line 9); otherwise, exploration terminates and the robot follows a path on the SET leading back to q^0 (*homing*, line 13).

Proposition 4.3: The following implication holds:

$$\mathcal{U}^k = \emptyset \;\Rightarrow\; \bigcup_{i=0}^{k} \mathcal{D}(q^i, k) \cap \mathcal{Q}^k = \emptyset$$

Proof: See [12]. ■

F. Completeness

Proposition 4.4: Any SET exploration which ends at a finite step k is completed, in the sense that $\mathcal{Q}^k = \emptyset$.

Proof: See [12]. ■

[4]Even when $\mathcal{L}(q^k, k) \neq \emptyset$, it may happen that $\mathcal{D}(q^k, k) \cap \mathcal{Q}^k = \emptyset$. In general, this occurs when portions of the free boundary cannot be 'pushed-forward' by additional sensor views [12].

SEARCH STRATEGY IN $D(q^k, k)$

1: $q^{k+1} \leftarrow 0$, $U^{k+1} \leftarrow 0$, $J \leftarrow \{1, 2, ..., m\}$
2: **while** $U^{k+1} = 0$ and $J \neq \emptyset$
3:   $j \leftarrow$ select sensor $j \in J$ with $\mathcal{L}_j(q^k, k) \neq \emptyset$
4:   $R \leftarrow \{1, 2, ..., m\}$
5:   **while** $U^{k+1} = 0$ and $R \neq \emptyset$
6:     $r \leftarrow$ select by priority sensor $r \in R$
7:     $(q^{k+1}, U^{k+1}) \leftarrow$ search configuration with maximum utility in $\mathcal{D}_{r,j}(q^k, k) \cap \mathcal{Q}^k_r$
8:     **if** $U^{k+1} = 0$ $\;R \leftarrow R \setminus \{r\}$
9:   **end while**
10:   **if** $U^{k+1} = 0$ $\;J \leftarrow J \setminus \{j\}$
11: **end while**
12: **return** (q^{k+1}, U^{k+1})

Fig. 5. A pseudocode description of the search strategy in $D(q^k, k)$.

The above proposition only considers finite exploration sequences, because a compact free world may not be 'coverable' by a finite sequence of views. In such 'pathological' cases [12], maximizing $I(q, k)$ over $\mathcal{Q}^k$ results in an infinite sequence of view configurations q^i along which $I(q^i, k)$ tends to zero. Hence, $\mathcal{Q}^k$ never becomes empty.

V. IMPLEMENTATION

A. Search in the Admissible Set

In general, $D(q^k, k)$ (see Sect. IV-C) is a huge search space for the utility maximization problem (line 2, Fig. 3). In order to reduce the search complexity, an heuristic search algorithm can be worked out by relaxing the solution optimality requirement and exploiting the search space inherent decomposition (eq. (3)). This is described in Fig. 5.

Instead of searching for the optimal solution in $\mathcal{D}(q^k, k)$, the algorithm searches one of the *suboptimal solutions* which maximize the utility function in the partial sets $\mathcal{D}_{r,j}(q^k, k) \cap \mathcal{Q}^k_r$, for $r, j \in \{1, 2, ..., m\}$. In particular, the partial sets are visited and searched one by one until the first suboptimal solution is found. The visit order is heuristically designed.

A possible choice is detailed hereafter. For any fixed j, it can be shown [12] that: if $\mathcal{L}_j(q^k, k) = \emptyset$ one has $\mathcal{D}_{r,j}(q^k, k) \cap \mathcal{Q}^k_r = \emptyset$ for $r = 1, 2, ..., m$ and, thereby, the suboptimal solutions in these partial admissible sets do not exist. Otherwise, if $\mathcal{L}_j(q^k, k) \neq \emptyset$ a measure of the unexplored points lying in $B(s_j(q^k), \rho + R)$ can be used as an 'optimality indicator' of these suboptimal solutions. Our strategy selects a sensor j with a non-empty $\mathcal{L}_j(q^k, k)$ and with the highest 'optimality indicator' (line 3). Once j has been chosen, the sensor r is selected by priority (line 6). The highest priority is assigned to the index $r \in R$ which minimizes the distance $\|s_r(q^k) - s_j(q^k)\|$. Accordingly, $\mathcal{D}_{j,j}(q^k, k) \cap \mathcal{Q}^k_j$ is the first searched set. Note that it is certainly $q^k \in \mathcal{D}_{j,j}(q^k, k) \neq \emptyset$.

B. Search in Partial Admissible Sets

In the exploration process, SET incrementally updates a model of the configuration space for (i) searching new view configurations and (ii) performing planning operations. Since generic robotic systems typically have high-dimensional configuration spaces, a sampling based approach can be conveniently used to incrementally grow a roadmap which captures the connectivity of the current safe region.

In particular, let $\mathcal{G}^k$ be the roadmap built at step k in the safe region $\mathcal{S}^k$. In $\mathcal{G}^k$, a node represents a safe configuration at step k, while an arc between two nodes represents a local path that is safe at step k and connects the two configurations. Once $\mathcal{E}^k$ is computed merging $\mathcal{V}(q^k)$ with $\mathcal{E}^{k-1}$, the roadmap $\mathcal{G}^k$ is obtained expanding $\mathcal{G}^{k-1}$. During this expansion process, additional sampled configurations which are safe at step k are added to $\mathcal{G}^{k-1}$. In order to find these configurations a collision checking is performed in the reconstructed world model at step k: according to this model, $\mathcal{E}^k$ is the free world and $\partial\mathcal{E}^k$ is the obstacle boundary. In this framework, the SET built at step k represents the path actually traveled by the robot on the roadmap $\mathcal{G}^k$.

Two main instances of the SET method can be obtained depending on the strategy used for growing the roadmap and searching in the partial admissible sets (Fig. 5, line 7). SET with *Global Growth* (SET-GG), which incrementally performs a global expansion of the roadmap $\mathcal{G}^k$. SET with *Local Growth* (SET-LG), which privileges a local expansion of $\mathcal{G}^k$ around the current view configuration q^k.

1) SET-GG Search Strategy (see [12]): SET-GG incrementally expands $\mathcal{G}^k$ in the current safe region $\mathcal{S}^k$ using a sampling-based approach such as a multi-query PRM algorithm, or a single-query single-tree algorithm (e.g., RRT).

2) SET-LG Search Strategy (see [12]): SET-LG first performs a local search around q^k in the attempt to locally maximize the utility function, then, when no local informed configurations are found, it allows a global search (performing possible long jumps). In particular, in the local search: a single-query single-tree algorithm such as RRT or EST is locally expanded within a fixed radius δ from q^k. In the global search: a tree is expanded without performing collision checking (*lazy tree*) inside $\mathcal{D}_{r,j}(q^k, k)$ (see [12]).

Note that the local search mechanism automatically limits the navigation cost of the next robot motion and avoid erratic behaviors. A shortcoming of SET-LG is a non-uniform sampling of the free configuration space. In fact, local searches started at distinct view configurations may expand in overlapping C-space regions. This unwanted result can be almost avoided by suitably selecting the radius δ of the local expansions.

C. Path Planning

Once a new view configuration q^{k+1} has been selected, a safe path connecting q^k to q^{k+1} is computed by the path-planner. In the SET method, planning depends on the used search strategy. In SET-GG, a safe path is computed on the roadmap $\mathcal{G}^k$. In SET-LG, q^{k+1} is found either by a local search or by a global search (see [12]).

VI. Simulations

We present simulation results obtained implementing the presented SET method in Move3D [14]. The algorithms have been extensively tested in several scenes (both in 2D and 3D worlds) using various robots (both fixed-base and mobile manipulators). We report here the results obtained in the six cases of Fig. 6. Two groups of simulations were performed for each case: in the first group, a single range finder is mounted on the tip of the robot; in the second, two additional range finders are added and mounted on the last robot link (midway along the length of the link and close to the last robot joint). Each range finder has a perception range $R = 1$ m and an opening angle $\alpha = 60°$ (robot link lengths range from 0.3 m to 0.8 m). Its linear and angular resolution are respectively 0.01 m and 1°. In 2D cases, the sensors can rotate within a 120° planar cone; in 3D cases, the sensors can rotate within a 120° × 120° spatial cone. At the start of the exploration, a free box $\widetilde{\mathcal{A}}$ is assumed to be known from an external source. In particular, its volume is 200% of that of $\mathcal{A}(q^0)$ on the average.

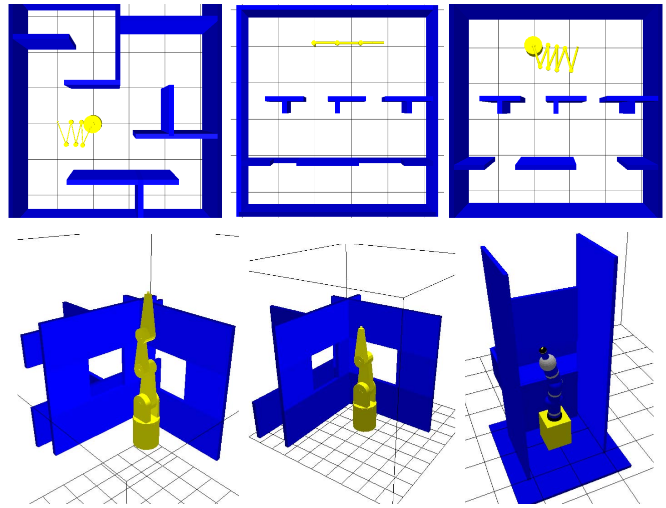

Fig. 6. *Top*: 2D cases (left to right): A6R, B3R*ff*, C8R. *Bottom*: 3D cases (left to right), D4R, E3R, F7R. In each world name, the first letter identifies the scene, while the number quantifies the robot revolute (R) joints; *ff* identifies a free-flying robot.

Gridmaps are used as world models (with a 0.1m grid resolution). Quadtrees/octrees are used to represent (and efficiently operate on) the free and obstacle boundaries. Information gain is computed via ray-casting procedures. At each step, the partial local free boundary $\mathcal{L}_j(q,k)$ is computed by expanding a numerical 'navigation' function from $s_j(q)$ within $\mathcal{E}^k \cap B(s_j(q), \rho + R)$: any cell in $\mathcal{E}^k \cap B(s_j(q), \rho+R)$ with a finite function value can be connected to $s_j(q)$ and is consequently inserted in $\mathcal{L}_j(q,k)$. Besides, $\mathcal{L}_j(q,k)$ is updated only when $\mathcal{V}(q^{k+1}) \cap B(s_j(q), \rho+R) \neq \emptyset$. Simulations were performed on a Intel Centrino Duo 2x1.8 GHz, 2GB RAM, running Fedora Core 8.

A. Sampling Methods

In SET-GG, the global roadmap $\mathcal{G}^k$ is incrementally expanded using PRM or RRT. In SET-LG, we found that RRT is more effective. In particular, RRT-Extend is used for local searches, while RRT-Connect is more suitable for the lazy tree expansions. In all these techniques, kd-trees are used to perform nearest neighbor searches, uniform random sampling is applied and path smoothing is performed.

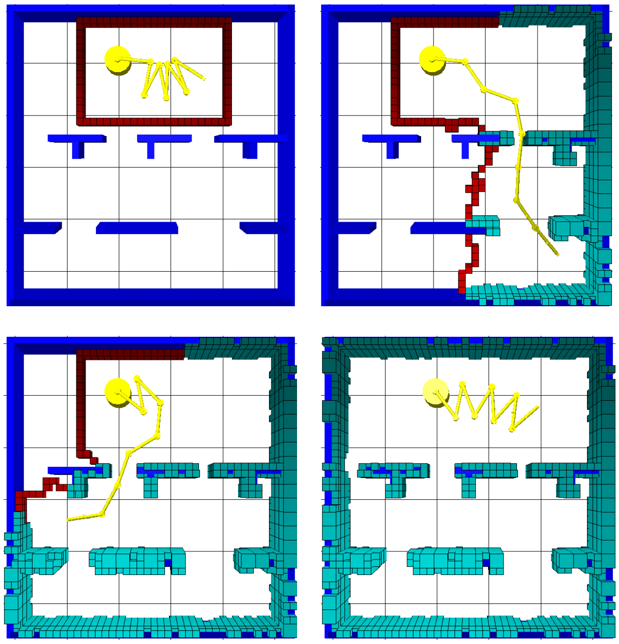

Fig. 7. An exploration progress in world C8R with three range finders.

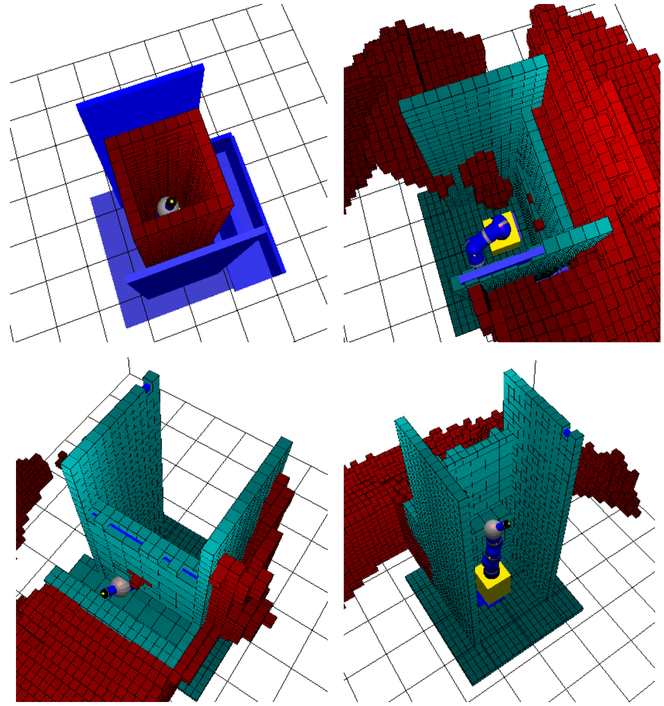

Fig. 8. An exploration progress in world F7R with three range finders.

Results with 1 range finder

World	*NV*	$\mathcal{W}_{\%}$	*NNGR*	*NCDC*
A6R (+1R)	45	100.00%	56938	446314
B3R*ff* (+1R)	48	100.00%	58565	495132
C8R (+1R)	54	100.00%	24662	360591
D4R (+2R)	82	100.00%	48502	323481
E3R (+2R)	79	100.00%	38734	291529
F7R (+2R)	81	100.00%	41679	232913

Results with 3 range finders

World	*NV*	$\mathcal{W}_{\%}$	*NNGR*	*NCDC*
A6R (+3×1R)	30	100.00%	69451	554523
B3Rff (+3×1R)	36	100.00%	86244	615137
C8R (+3×1R)	40	100.00%	52944	432305
D4R (+3×2R)	64	100.00%	77121	671163
E3R (+3×2R)	65	100.00%	84087	571219
F7R (+3×2R)	70	100.00%	91474	598014

TABLE I

RESULTS OBTAINED WITH SET-LG.

B. Parameter Choice

• *Admissible set*. The radius ρ in the partial admissible set definition was set to 1 m, i.e, equal to R.

• *RRT*. Each RRT expansion is performed for a maximum number of iterations $K_{\max}$. For both local and global searches, we used $K_{\max} \simeq 2000$, whereas for lazy expansions $K_{\max} \simeq 6000$.In SET-LG, a configuration space ball with radius δ is used in the local search. Typically, we set $\delta \simeq 0.1\,\delta_M$ where δ_M is the maximum estimated distance between two points in $\mathcal{C}$.

C. Performance Indexes

• *Number of Views* (*NV*). It is the total number of views acquired by the robot during an exploration.

• *World Coverage* ($\mathcal{W}_{\%}$). It represents the percentage of the free world included in the final explored region. This percentage is evaluated w.r.t. to an estimate of the free world which can be explored by the robot, i.e., the set of points $p \in \mathcal{W}_{\text{free}}$ such that $p \in \mathcal{V}(q)$ for some configuration $q \in \mathcal{C}_{\text{free}}$ which is reachable from q^0 through a safe path.

• *Number of Collision Detection Calls* (*NCDC*). It is the total number of collision detection calls performed during an exploration.

• *Number of Nodes of the Global Roadmap* (*NNGR*). It is the total number of nodes of the final roadmap $\mathcal{G}^k$.

D. Results

Two typical exploration processes obtained with SET-LG in cases C8R and F7R, both with three range finders, are shown in Figs. 7 and 8. The obstacles (in blue) are obviously unknown to the robot. They are incrementally reconstructed during the exploration as the obstacle boundary $\partial\mathcal{E}^k_{\text{obs}}$ (light-blue cells). In each frame, the free boundary $\partial\mathcal{E}^k_{\text{free}}$ (red cells) is also shown. Fig. 8 shows only the portions of the free boundary contained in the set of points $p \in \mathcal{W}$ such that $p \in \mathcal{A}(q) \cup \mathcal{V}(q)$ for some $q \in \mathcal{C}$. Note that, at the end of the exploration, the remaining free boundary can not be 'pushed-forward' by additional sensor views.

Clips of these two simulations are contained in the video attachment to the paper. Other simulations are available at the webpage http://www.dis.uniroma1.it/labrob/research/SET.html.

Table I compares the results obtained with SET-LG in the case of one range finder and three range finders. In view of the use of RRT, results are averaged over 20 simulation runs. Note that the world coverage is always 100%.

E. Comparison of SET-LG with SET-GG

An extensive simulation study has showed that SET-LG performs better than SET-GG. For lack of space, we do not report results obtained with SET-GG. In particular, we found that, for the same maximum number of iterations $K_{\max}$, the world coverage of SET-GG decreases by 10% on the average w.r.t. SET-LG, whereas exploration time increases by 40%.

A comparative analysis of the two methods can justify the results. At each step, the two main computational costs of the SET method are due to: *(i)* the expansion of the roadmap $\mathcal{G}^k$ *(ii)* the extraction of a subset of candidate configurations in $\mathcal{D}(q^k, k)$ from $\mathcal{G}^k$. In particular, at each step, SET-GG expands a global roadmap $\mathcal{G}^k$ which spreads uniformly over the whole safe region as k increases. Clearly, the number of nodes stored in $\mathcal{G}^k$ continuously grows. This causes a parallel, continuous increment of both the above computational costs. On the other hand, at each step, SET-LG mainly expands a new local tree $\mathcal{T}^k$ around the current view configuration q^k. Each of these trees, by construction, has a bounded number of nodes. Hence, with such mechanism, both the described computational costs are in principle bounded and held constant.

Another important advantage of the local growth performed by SET-LG is that it focuses the search process around the current view configuration q^k. This is convenient since, at least in the initial stages of the exploration, new informative configurations are likely to be contained in a neighbourhood of q^k. On the other hand, the global roadmap expansion results in the dispersion of new samples in uninformative configuration-space regions. Also, smaller traveled distance in $\mathcal{C}$ means less energy and exploration time.

VII. SET in the Presence of Uncertainty

If view sensing comes with uncertainty, a probabilistic world model (e.g. a probabilistic occupancy gridmap [15]) can be used to integrate collected sensor data. Here, a probability distribution associates each representative point in $\mathcal{W}$ with its probability of being in $\mathcal{O}$. Then, a point is classified as *free*, *occupied* or *unknown* comparing its occupancy probability with fixed probability ranges. In this context, SET definitions can be suitably modified. In particular, the explored region (obstacle boundary) is defined as the set of free (occupied) points, a point is unexplored if it is unknown, and the free boundary collects the set of unknown points lying 'close' to a free point. All the other definitions accordingly change and an entropy-based measure can be used in the information gain computation.

In a general probabilistic framework, the SET method (with the above modifications) can be thought of as a view planning module which can be suitably integrated with any localization module using a more general definition of utility function U in the spirit of an integrated exploration [2]. Correspondingly, motion planning should be also performed taking into account uncertainty [16].

VIII. Conclusion

We have presented a novel method for sensor-based exploration of unknown environments by a general robotic system equipped with multiple range finders. This extends the method originally presented in [11] for single-sensor robotic systems and comes along with a completeness analysis.

The method is based on the incremental generation of a data structure called Sensor-based Exploration Tree (SET). The generation of the next action is driven by information at the world level, where perception process takes place. In particular, the frontiers of the explored region are used to guide the search for informative view configurations. Various exploration strategies may be obtained by instantiating the general SET method with different sampling techniques. Two of these, SET-GG and SET-LG have been described and compared by simulations in non-trivial 2D and 3D worlds.

We are currently working to provide an accurate complexity analysis of the method, improve its completeness analysis and implement the SET method in presence of uncertainty both in sensing and control. Future work will address an experimental validation of SET on a real robotic system, and an extension of the method to a team of robotic systems equipped with multiple sensors along the lines of [5].

References

[1] 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*, 1998, pp. 3715–3720.

[2] 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*, 2002, pp. 534–539.

[3] H. Gonzalez-Banos and J. Latombe, "Robot navigation for automatic model construction using safe regions," in *Experimental Robotics VII*, ser. Lecture Notes in Control and Information Sciences, D. Russ and S. Singh, Eds., vol. 271. Springer, 2000, pp. 405–415.

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

[5] A. Franchi, L. Freda, G. Oriolo, and M. Vendittelli, "The sensor-based random graph method for cooperative robot exploration," *to appear in IEEE/AME Trans. on Mechatronics*.

[6] S. Hutchinson, R. Cromwell, and A. Kak, "Planning sensing strategies in a robot work cell with multi-sensor capabilities," in *1988 IEEE Int. Conf. on Robotics and Automation*, vol. 2, 1988, pp. 1068–1075.

[7] E. Kruse, R. Gutsche, and F. Wahl, "Efficient, iterative, sensor based 3-D map building using rating functions in configuration space," in *1996 IEEE Int. Conf. on Robotics and Automation*, 1996, pp. 1067–1072.

[8] P. Renton, M. Greenspan, H. ElMaraghy, and H. Zghal, "Plan-n-scan: A robotic system for collision-free autonomous exploration and workspace mapping," *Journal of Intelligent and Robotic Systems*, vol. 24, no. 28, pp. 207–234, 1999.

[9] P. Wang and K. Gupta, "A configuration space view of view planning," in *2006 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2006, pp. 1291–1297.

[10] W. R. Scott, G. Roth, and J.-F. Rivest, "View planning for automated three-dimensional object reconstruction and inspection," *ACM Computing Surveys*, vol. 35, pp. 64 – 96, 2003.

[11] L. Freda, G. Oriolo, and F. Vecchioli, "Sensor-based exploration for general robotic systems," in *2008 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2008.

[12] ——, "Exploration strategies for general robotic systems," *DIS Robotics Laboratory Working Paper*, 2009. [Online]. Available: http://www.dis.uniroma1.it/labrob/pub/papers/SET-WorkingPaper.pdf

[13] 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. Wellesley, MA: A K Peters, 2001, pp. 293–308.

[14] 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.

[15] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*. Cambridge, MA: MIT Press, 2005.

[16] A. Censi, D. Calisi, A. D. Luca, and G. Oriolo, "A bayesian framework for optimal motion planning with uncertainty," in *2008 IEEE Int. Conf. on Robotics and Automation*, 2008, pp. 1798 – 1805.