

# Robotic Exploration: New Heuristic Backtracking Algorithm, Performance Evaluation and Complexity Metric

Regular Paper

Haitham El-Hussieny<sup>1,3\*</sup>, Samy F.M. Assal<sup>1,4</sup> and Mohamed Abdellatif<sup>2</sup>

<sup>1</sup> Mechatronics and Robotics Engineering Department School of Innovative Design Engineering, Egypt-Japan University of Science and Technology, Egypt

<sup>2</sup> Mechanical Engineering Department, Faculty of Engineering and Technology, Future University, Egypt

<sup>3</sup> Electrical Engineering Department, Faculty of Engineering, Benha University, Egypt

<sup>4</sup> Department of Production Engineering and Mechanical Design, Faculty of Engineering, Tanta University, Egypt (on leave)

\*Corresponding author(s) E-mail: haitham.elhussieny@ejust.edu.eg

Received 1 April 2014; Accepted 27 October 2014

DOI: 10.5772/60043

© 2015 The Author(s). Licensee InTech. This is an open access article distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/3.0>), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

## Abstract

Mobile robots have been used to explore novel environments and build useful maps for navigation. Although sensor-based random tree techniques have been used extensively for exploration, they are not efficient for time-critical applications since the robot may visit the same place more than once during backtracking. In this paper, a novel, simple yet effective heuristic backtracking algorithm is proposed to reduce the exploration time and distance travelled. The new algorithm is based on the selection of the most informative node to approach during backtracking. A new environmental complexity metric is developed to evaluate the exploration complexity of different structured environments and thus enable a fair comparison between exploration techniques. An evaluation index is also developed to encapsulate the total performance of an exploration technique in a single number for the comparison of techniques. The developed backtracking algorithm is tested through computer simulations for several structured environments to verify its effectiveness using the developed complexity metric and the evaluation index. The

results confirmed significant performance improvement using the proposed algorithm. The new evaluation index is also shown to be representative of the performance and to facilitate comparisons.

**Keywords** Robot exploration, sensor-based random tree technique, backtracking, complexity metric, evaluation index

## 1. Introduction

There is increasing need for autonomous robots in hazardous environments, such as disaster sites and nuclear plants, as well as in inaccessible areas such as volcanoes and for space missions. Exploration is essential for robots that are required to move autonomously in novel environments. Therefore, developing efficient strategies for exploration is both interesting and important. Map information is important for path planning and task execution since the availability of a map increases the speed with which the robot can reach areas of interest in the environment. It is

important to define the goal and evaluation criteria to judge the exploration performance. Intuitively, the objective of exploration is to gain the maximum amount of accurate information about the environment - represented by the explored space completeness - in the shortest time and over the minimum distance travelled.

### *1.1 Robot exploration algorithms and backtracking techniques*

Exploration is usually made using a greedy strategy that plans one step ahead by determining the next-best locations - called 'frontiers' - which maximize the acquired information. One of the most popular frontier-based exploration techniques was developed in [1], in which the frontiers are defined as the boundaries between free and unexplored areas. Approaching those frontiers enables the acquisition of more information about the unknown environment. Frontier-based exploration [1] is common to almost all exploration techniques and, depending upon the frontier selection mechanism, the existing techniques can be broadly classified into three categories, namely: optimal-frontier, behaviour-based and randomized motion techniques.

In optimal-frontier exploration, the next frontier is selected based upon a cost function. In [1], this function was selected to be the shortest distance required to reach each of them. The cost function may take other forms. For example, in [2], two criteria were considered in the evaluation, namely: the travelling cost required to reach a frontier and the expected information gain when performing a sensing action at that frontier. The cost function was controlled by three parameters: the distance cost, the expected utility and localizability (the latter of which is defined by the suitability of a frontier to enhance the robot localization when reaching that frontier, as described in [3]).

Optimal frontier strategies have two common problems. First, due to continuous map updates, the currently approached region might become fully explored during navigation before reaching the destination frontier. In this case, the robot will start to explore the next unknown region. This problem occurs when using sensors with a wide perceptual range. Second, the optimal frontier can lie outside the room being explored. This situation may cause the robot to explore the same room twice, with unnecessary long distances. Repetitive re-checking of the frontier during navigation and segmentation of the partially built map were suggested in [4] to solve the first and the second problem respectively. The segmentation process separates rooms from each other causing the robot to favour visiting frontiers lying inside the currently explored room. However, although such a solution works well in office-like environments, open environments cannot be reasonably segmented, which may impair performance.

The exploration process could be decomposed into smaller, simple, reactive tasks, which leads to the use of behaviour-

based exploration approaches [5-11]. The exploration task is divided into simple simultaneous actions. Such actions or behaviours involve repulsive and attractive forces, such as avoiding obstacles and reaching targets, respectively. For instance, in [6] a behaviour-based approach combined three reactive behaviours for exploration, namely: "reach frontiers", "avoid obstacles" and "avoid other robots". Another example in which a simple wall-following behaviour as a reactive model led exploration was proposed in [7]. In [8], repulsive behaviour from previously visited areas was used for exploration. In [9, 10], a complex behaviour architecture was proposed in which a combination of several weighted behaviours were fused together for efficient exploration. These behaviours can be generally formulated as: "go to frontier", "go to unexplored areas", "avoid other robots" and "avoid obstacles". However, reactive systems do not perform well in large and complex environments [11]. In such environments, the forces combining those behaviours could compensate each other in a certain region, causing local minima that trap the robot at a certain point. The local minima problem is common in behaviour-based techniques, not only for exploration but also for normal navigation. This requires such systems to have a local minima detection and recovery mechanism to avoid this problem.

In randomized motion planning [12-14], robots are directed to acquire more information through random steps. Randomized increments of a data structure called 'sensor-based random trees' (SRTs) were generated in [13]. This tree represents the roadmap of the explored area with an associated safe region that describes obstacle-free regions around the robot depending upon its sensor aspects. The nodes of this tree are the explored locations. This basic SRT strategy was later modified to enhance the process. For instance, in frontier-based SRT (FB-SRT) [14], the random selection of target points was biased towards local frontier arcs in the current safe region. This improves efficiency in terms of shorter travelled paths and greater area coverage. Furthermore, in [15] and [16] a sensor-based exploration tree (SET) was constructed for depth-first search exploration. The target configurations were selected to maximize the estimated information utility in forward mode only by calculating the expected utility along the local frontier boundary. This helps the robot to filter out any useless actions that might be executed. In these strategies (SRT, FB-SRT and SET), if there are no more areas to explore, the robot goes back through the previous nodes to find new regions and explore again. This backtrack strategy causes long exploration distances and times, especially in places with wide open spaces.

Several approaches were proposed to solve this backtracking problem. For instance, bridges were added in [17] to the exploration tree so that the robot could plan quick paths without looping all the previous nodes. A bridge was added between any two adjacent nodes with a common safe region between them and separated by a distance greater than double the sensor range. The drawback of this improvement is the possibility that other nodes might be

worth reaching despite having no shared safe region with the currently visited node. Ignoring such a node will reduce the area covered by the robot. In addition, in [18] a short path was planned between the current node without additional information and the initial node assuming that no information will be gathered from the nodes between them. This hypothesis works well in corridor environments rather than wide spaces and office-like environments.

### 1.2 Performance metrics for robotic exploration

Although robots are designed bearing in mind their working environment, there is no efficient way to compare robot performances in different places. One way to do this is to run a series of simulations of robots working in different structures. However, this is time-consuming and requires much effort. Several complexity metrics (CMs) had been introduced [19-22], namely: a space syntax, entropy, and obstacle distribution and compression. Those metrics are suitable for path planning rather than exploration. The space syntax method [19], which is concerned with the connectivity of environmental features rather than distances, uses a labour-intensive axial map to measure environmental complexity. Thus, it is not suitable for exploration. Entropy is regarded as a measure of uncertainty in the working environment [21, 22]. The more free spaces there are, the more decisions the robot should make to reach its target and the more time will be taken. Hence, entropy is a measure of the environmental complexity if the robot wishes to reach a particular goal. For exploration, there is no inaccurate decision, since any given decision will have benefits for exploration. Additionally, the sensor details are not considered in the entropy computations. Therefore, entropy is not an adequate measure of environmental complexity from the exploration point of view. Since the distribution of obstacles in an environment affects the navigation task's complexity, the environment is identified by a unique factor called 'the compression factor' [22]. This factor measures the repeatability of patterns of obstacles in a certain environment, and hence the environment's complexity. In other words, if the obstacles have repeated patterns and - in turn - can be easily described, then this environment has a low CM, and vice versa. Even though it considers the sensor properties, it is computationally expensive and not suitable for measuring the environmental complexity for the exploring robot. Therefore, there is no complexity measure in the published literature suitable from the exploration point of view, to the best of our knowledge.

Although sensor-based exploration enjoys simplicity and completeness, it can take long time and involve a greater distance, mainly due to its backtracking strategy [18]. Suppose that there are no frontier areas during exploration, then the robot must go back to previous locations until there exists a frontier area. In some situations, the robot can travel a significant distance until it reaches a position which has frontier regions. In this paper, which extends our preliminary work in [30], a novel, simple yet effective backtracking technique based on a heuristic algorithm is

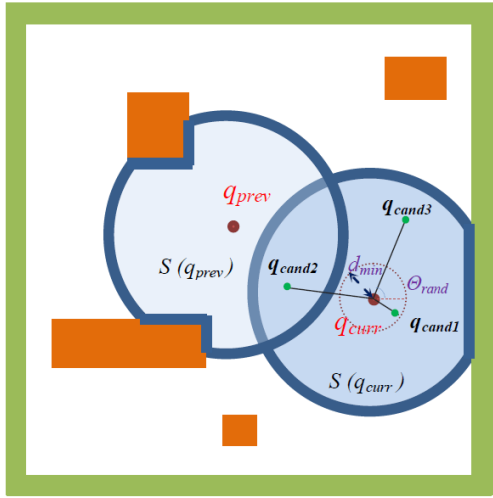
proposed. This new algorithm is based on the selection of the most informative node to approach directly rather than travelling across all previous nodes in order. The most informative node is determined using the ray-casting algorithm. The proposed technique can be regarded as a combination of the optimal frontier and randomized motion planning strategies. Random exploration planning is applied in forward mode, while the optimal node is approached in backward mode. The new technique is suitable for time-critical applications, such as rescue applications. Another contribution of this paper is in devising a new evaluation index *EI* for exploration that is useful for comparing different techniques. The *EI* encapsulates the exploration efforts, distance and time, the percentage of the area covered, completeness and the number of nodes in a single number to avoid any trade-off among those metrics. An efficient environmental CM is also proposed to evaluate the degree of complexity of an environment and to compare techniques working in different structured environments.

This paper is organized as follows: In Section 2, the basics of the basic SRT exploration strategy are briefly outlined. In Section 3, the new exploration technique using the proposed heuristic backtracking algorithm is described. The proposed environmental CM and *EI* are introduced in Section 4. Simulations running on different exploration scenarios and a comparison with the basic sensor-based technique are presented in Section 5. Finally, conclusions are drawn in Section 6.

## 2. Sensor-based random tree exploration

The SRT exploration technique [13] is based on a random selection of robot configurations  $q=[x \ y \ \theta]^T$  inside the local safe region (LSR), where  $x$  and  $y$  represent the position of the robot and  $\theta$  represents the robot orientation with respect to the local coordinate frame. The LSR represents the free space around the robot in the current configuration  $q_{curr}$ , where its shape depends upon the sensor characteristics as described in [23]. A road map of the visited configurations - with the associated LSR - is represented by an incremental data structure called 'SRT'. Each node in the tree  $T$  represents the explored location. Pseudo-code of the SRT technique is shown in Algorithm 1. This algorithm is repeated  $K_{max}$  times, which is the maximum number of nodes assumed to be found in the environment. The Algorithm starts at  $q_{curr}$  acquiring the sensor measurements through the **PERCEPTION** function. The LSR, denoted by  $S(q_{curr})$ , and  $q_{curr}$ , are added to  $T$ . Next, a random angle  $\theta_{rand}$  is generated to select the direction of the path that the robot will travel. The length of this path is the radius  $r$  of the LSR in that direction,  $\theta_{rand}$ , which is obtained by the **RAY** function. According to this random direction and radius  $r$ , a candidate configuration  $q_{cand}$  is obtained inside  $S$  through the function **DISPLACE**. Afterwards,  $q_{cand}$  is tested to validate two conditions. Firstly, it must be at a distance greater than a given distance

$d_{min}$  from  $q_{curr}$ , where  $d_{min}$  is a given threshold. Secondly, it should not belong to the LSR of any other node in the exploration tree  $T$ , as shown in Fig. 1. This search for a valid candidate node is done by the **VALID** function and is repeated until the valid node or the maximum number of searches,  $I_{max}$ , has been reached. Note that setting the SRT radius multiplier constant  $\alpha \leq 1$  guarantees that  $q_{cand}$  is within the safe region, and hence collision-avoidance is not needed. If there is no configuration satisfying these conditions, backtracking or the homing step will start, letting the robot travel along previous nodes to find a new unexplored region.



**Figure 1.** Validation of different candidate configurations in SRT:  $q_{cand3}$  is accepted, while  $q_{cand1}$  and  $q_{cand2}$  are not

#### Algorithm 1: A pseudo-code for the basic SRT algorithm **Build\_SRT**

**Input:**  $(q_{init}, K_{max}, I_{max}, \alpha, d_{min})$

**Output:** Roadmap tree  $T$

```

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

```

### 3. A new heuristic backtracking algorithm

The following are the explicit assumptions for the new exploration approach:

1. Robot localization is provided by a separate module.
2. The exploration environment is planar, i.e.,  $R^2$ , due to the nature of the planar range sensor used.
3. The exploration environment is static, i.e., it consists of unchanging surroundings in which the robot explores.
4. The robot is holonomic, i.e., it can turn in any direction.

In SRT, when there are no more valid configurations to reach, the robot traverses the parent node of  $q_{curr}$ , searching for new candidate locations to explore. This backtrack step takes time, which is not justified. In the proposed approach, forward mode exploration is performed in a similar fashion to the basic SRT method as shown in Algorithm 2. While in backtracking mode, when there are no valid configurations to reach, the tree data structure that has been built is tested in reverse order; starting from the current node, and the parent nodes,  $q_{test}$ , are checked for whether they can provide more information. A gain  $G(q_{test})$  is calculated by the function **GET\_GAIN**, which is based on the ray-casting algorithm as described in Section 3.2. This gain, measured in terms of free cells, is a measure of how a given node is worth visiting for further exploration. If the estimated gain of the tested node is more than a given threshold  $G_{thresh}$ , the node will be selected as a valuable node. In other words, this node can be considered to be the most informative node. Next, the shortest path will be planned to reach this selected node using  $A^*$  through the **APPROACH**( $q_{test}$ ) function. If no more valuable nodes are found, the current node will be identified as a home node, at which no further moves are to be made and exploration is considered complete. This is the difference between the proposed algorithm and the basic SRT algorithm. The homing process in the proposed algorithm does not require moving the robot to its starting node as in basic SRT. Instead, in the proposed approach, the homing process may be at any node whenever there are no more valuable nodes to visit.

The basic idea for enhancing backtracking in SRT is shown schematically in Fig. 2. In the basic SRT strategy, as shown in Fig. 2(a), when there are no new areas the robot backtracks to all the previous nodes until exiting the currently explored room. While using the proposed approach, the robot searches for the most informative node, as shown in Fig. 2(b), to which the starting node is assumed as a new starting node. A shortest path - with the help of the built LSRs - is then planned to approach this node of interest, saving both distance and time.

#### 3.1 Map building

A spatial representation for the unknown environment is required for two tasks: environment monitoring (such as in



Algorithm 2: A pseudo-code for the proposed heuristic backtracking algorithm.

---

**Enhanced\_SRT**

---

**Input:** ( $q_{init}, K_{max}, I_{max}, \alpha, d_{min}, G_{thresh}$ )

**Output:** most informative node  $q_{test}$ , roadmap tree  $T$ .

---

```

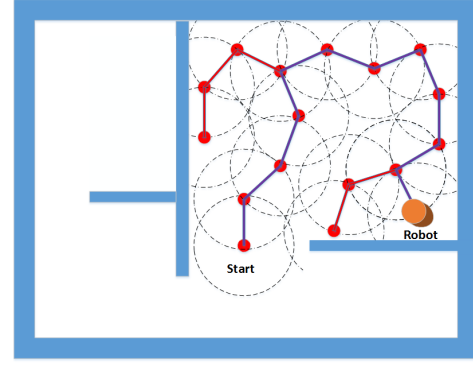
 $q_{curr} = q_{init};$ 
for  $k = 1 \rightarrow K_{max}$ 
     $S(q_{curr}) \leftarrow \text{PERCEPTION}(q_{curr});$ 
     $\text{ADD}(T, (q_{curr}, S(q_{curr})));$ 
     $i \leftarrow 0;$ 
    repeat
         $\theta_{rand} \leftarrow \text{RANDOM\_DIR};$ 
         $r \leftarrow \text{RAY}(S(q_{curr}), \theta_{rand});$ 
         $q_{cand} \leftarrow \text{DISPLACE}(q_{curr}, \theta_{rand}, \alpha r);$ 
         $i \leftarrow i + 1;$ 
    until ( $\text{VALID}(q_{cand}, d_{min}, T)$  or  $i = I_{max}$ )
    if  $\text{VALID}(q_{cand}, d_{min}, T)$ 
         $\text{MOVE\_TO}(q_{cand});$ 
         $q_{curr} \leftarrow q_{cand};$ 

% Modifications done to the backtrack:
else
    % prepare the parent node to be tested
     $q_{curr} \leftarrow q_{curr}.\text{parent};$ 
    repeat
         $q_{test} = q_{curr};$ 
    % calculate the information gain for  $q_{test}$ 
     $G = \text{GET\_GAIN}(q_{test});$ 
     $q_{curr} \leftarrow q_{curr}.\text{parent};$ 
    % exit if the tested node is valuable
    % or the configurations tree is empty
    until ( $G \geq G_{thresh}$  or  $q_{curr}.\text{parent} = \text{NULL}$ )
    % if the tested node is valid
    if ( $G \geq G_{thresh}$ )
        % plan a shortest path to reach
         $\text{APPROACH}(q_{test});$ 
         $q_{curr} \leftarrow q_{test};$ 
    else
        % stop at the current node for homing
         $q_{curr} \leftarrow q_{curr};$ 
    end
end
loop
return  $T;$ 

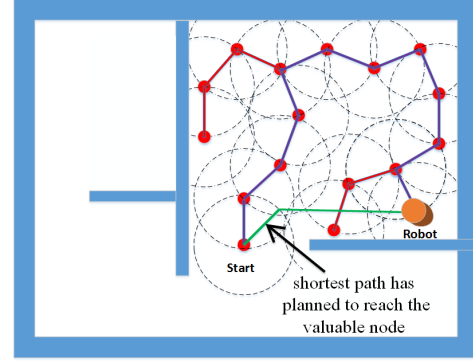
```

---

surveillance applications) and helping the robot to navigate the environment in backtracking mode. In this paper, the occupancy grid-based map is used for this representation. The environment is divided into small grids, each containing a value representing the probability of being occupied by obstacles. It is necessary to know for each cell whether it is unknown, free or an obstacle. Initially, the map is assumed to be unknown. Given a range scan and the robot pose, the occupancy grids within the sensor range are updated as follows. Firstly, scan readings are converted into Cartesian coordinates of the occupancy grid map, creating a polygon of points. Secondly, this polygon is identified as the LSR and filled using the flood-fill algorithm. Thirdly, obstacle cells are identified by the sensor readings that are lower than the maximum sensor range  $R_{max}$ . The process is shown in Fig. 3.

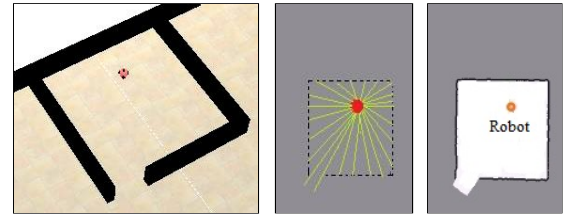


a)



b)

**Figure 2.** A sketch showing a robot exploring a room using: a) the basic SRT strategy, and b) the proposed approach, where a shortest path, in green, is planned to the most informative node



**Figure 3.** Building an occupancy grid map using scanner range data

### 3.2 The most informative node

In the proposed approach, when there is no valid configuration to reach, the robot selects the most informative node among the previous nodes. This informative node is expected to have information gain (in terms of free cells) greater than the threshold  $G_{thresh}$ , which depends upon the maximum range of the sensor. In other words, a large sensor range means more regions to explore and, hence, a large threshold is required. This threshold is selected heuristically to achieve a compromise between exploration completeness and the total distance travelled. The higher the threshold, the shorter the exploration time and the lower the level of exploration completeness. In fact, the estimation of the information gain that could be obtained at any point is difficult. The actual gain is hard to predict, as it varies according to the structure of the corresponding

region. In [11], this gain was calculated by counting the number of unknown cells lying in a particular region surrounded by the maximum sensor range. This method did not guarantee a correct estimation, since some unreachable and unknown regions could be counted. In [4], the information gain was approximated as the relative difference between the current map entropy and the expected entropy after the simulated robot step at the candidate location. This approach requires scanning all the cells in the global map. In our algorithm, a simple heuristic ray-casting [24] method is applied to estimate how valuable a certain node  $q_{test}$  will be. During the ray-casting process, as illustrated in Algorithm 3, the number of configurations traversed by the rays which can contribute to the exploration process is recorded, and the sum of all the valid configurations traversed by the rays is used as a measure of how much information gain can theoretically be obtained from a particular node. This suits the laser scanner sensor used, where the number of scan rays and the angle between them depend upon the characteristics of the actual sensor used. Note, that sensor noise is not considered when estimating the information gain of each node, since the noise can affect the estimation and should be modelled. Valid configurations are tested through the **VALID** function to meet the conditions mentioned above in Section 2. This is shown in Fig. 4, where the tested node  $q_2$  on the left is estimated to have more information gain  $G \geq G_{thresh}$  than the node  $q_2$  on the right.

Algorithm 3: Ray-Casting algorithm to estimate the most informative node.

---

**GET\_GAIN**

---

**Input:**  $q_{curr}$   
**Output:** Node gain  $G$ .

---

% starting from angle 0 to the field of view FOV  
**for**  $\theta_{ray} = 0 \rightarrow FOV$   
 % cover the total maximum range  
**for**  $r = 0 \rightarrow R_{max}$   
 % convert to Cartesian co-ordinate  
 $q.x = q_{test}.x + r * \cos(\theta_{ray});$   
 $q.y = q_{test}.y + r * \sin(\theta_{ray});$   
 % testing for validity  
**if** **VALID**( $q, d_{min}, T$ )  
 %increment gain by 1  
 $G \leftarrow G + 1;$   
**loop**  
**loop**  
**return**  $G;$

---

After identifying the most informative node, a shortest path is planned using the  $A^*$  algorithm [25] to reach it. This saves both exploration distance and time as compared to visiting all the previous nodes. The dimensions of the robot are taken into account while planning the path by eroding the partially built map with a disk structure element. Unknown and obstacle cells are avoided during robot navigation.

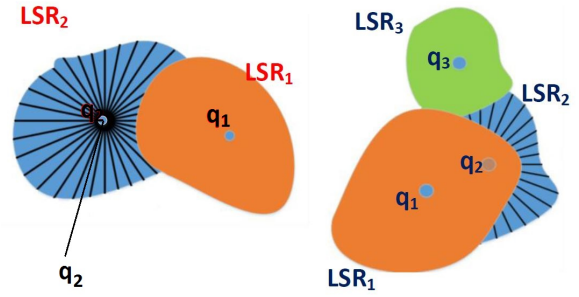


Figure 4. Estimation of information gain at different configurations by the use of a ray-casting technique

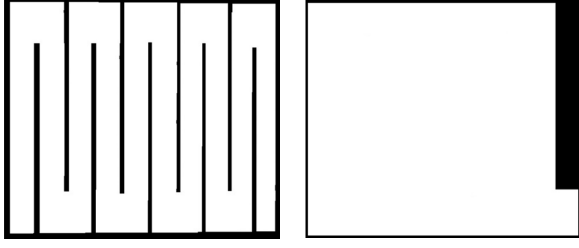
## 4. The proposed complexity metric and evaluation index

### 4.1 Environmental complexity metric

Robots are tested in different environments with different degrees of complexity, which is problematic for comparing the performance of different exploration strategies. Therefore, it is useful to develop a metric that can quantify environmental complexity. This will help to reduce the effect of the environmental complexity on the performance comparison of the different exploration strategies. The available complexity measures, such as space syntax, entropy and obstacle compression, are measures related to path planning rather than robot exploration. From the path-planning perspective, free areas mean more choices for the robot to reach its target, and hence a more complex environment. On the other hand, from the exploration perspective, free areas mean more information to be acquired about the unknown environment, and hence imply a less complex environment. As such, in this paper a novel environmental CM for the exploration process is proposed, as follows.

Intuitively, obstacles in the environment could help or hinder the exploration process. The exploration process will be efficient if the sensor range spans the entire environment. The effect of the obstacle density and distribution is illustrated in Fig. 5, where two environments - namely, a mazy environment and an open space environment - have the same number of obstacle cells but a different distribution. From the exploration point of view, the mazy environment is more complex than the open space environment. The complexity measurement could be simplified by calculating the difference between the actual number of nodes required to cover a certain environment,  $N_{act}$ , and the estimated number of nodes,  $N_{est}$ , required to cover the abstract free area without considering the obstacle distribution, as follows:

$$CM = 1 - \frac{N_{est}}{N_{act}} \quad (1)$$



**Figure 5.** Effect of obstacle distribution over complexity spaces - two environments with the same number of obstacle cells with different distributions. The first mazy environment is more complex than the open space environment.

where  $CM \in [0,1]$  is the CM, ranging from zero to one, whereby zero means an absence of scattered obstacles and that greater benefits will be acquired from the sensor range, while higher values mean that the sensor range is not used effectively to cover the environment, such as with the maze-like environment. The estimated number of nodes,  $N_{est}$ , can be calculated as the number of free cells in the environment divided by the area covered by the inner rectangle of the sensor field of view (FOV), as follows:

$$N_{est} = \frac{FreeSpaceArea}{2 * R_{max}^2} \quad (2)$$

where  $R_{max}$  is the sensor's perceptual range.

On the other hand, calculating the actual number of nodes required to fully cover the structured environment while considering the obstacles' density and distribution,  $N_{act}$ , which was addressed in [26], is not easy. Here, a modification of the art gallery algorithm [27], a well-known algo-

rithm for visibility in computational geometry, will be used. The reason for this is that the basic algorithm does not consider the sensor details, such as the maximum range and the FOV. Furthermore, it assumes a line of sight sensor, which is not practical (besides being computationally expensive). Thus, only an approximate solution that depends upon the density and distribution of obstacles in the environment can be obtained. Therefore, it is proposed to find the minimum actual number of nodes required to cover the entire space based on the basic art gallery problem. Basically, it randomly samples the environment to construct a relatively large set  $S_{sam}$  of covering nodes. Sensor aspects, such as the maximum range and the FOV, are considered by using the ray-casting **GET\_GAIN** algorithm. Algorithm 4 is structured as follows:

**Algorithm 4:** Greedy Randomized Art-Gallery Algorithm.

**Input:** 2D model of the environment's map,  $U_{thresh}$ ,  $n$

**Output:**  $N_{act}$  Actual number of nodes required to cover the entire environment.

---

$N_{act} = 0$ ;

**repeat**

    generate  $n$  random samples,  $s_i$ ;

    % where samples should be taken from the area of

    % the map not yet covered by previous nodes

**for**  $i = 1 \rightarrow n$

$G(s_i) = \mathbf{GET\_GAIN}(s_i)$ ;

**loop**

        select sample  $s_i^*$  with max.  $G$ ;

**PLOT**( $s_i^*$ )

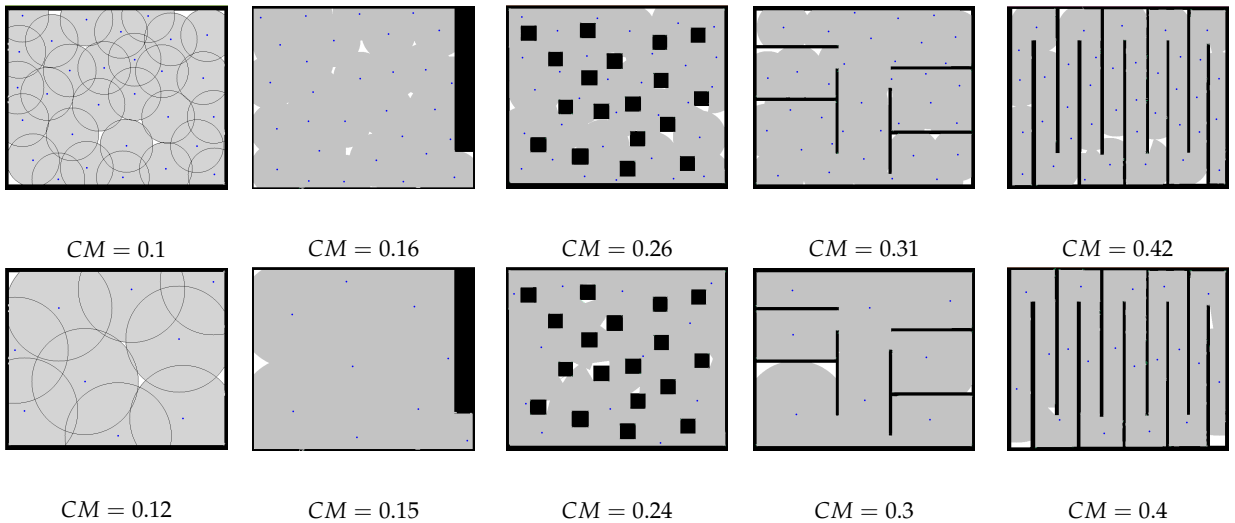
$N_{act} \leftarrow N_{act} + 1$ ;

        compute % of Coverage  $U$ ;

**until**  $U \leq U_{thresh}$

**return**  $N_{act}$ ;

---



**Figure 6.** The CM values, obtained over a single run, for different environments with  $n=100$  random samples,  $FOV=360^\circ$ , a coverage threshold.  $U_{thresh}=99\%$ . and with different sensor ranges. Top:  $R_{max}=2m$ . Down:  $R_{max}=4m$ .

After acquiring the 2D model of the environment, the algorithm computes the estimated number of nodes,  $N_{est}$ , required to cover the entire environment using (2). Afterwards,  $n$  samples will be generated randomly to be distributed over the environment. To eliminate the randomness effect over the output of the algorithm, the number of random samples is selected to be large, for example, 100 samples for the  $R_{max}=2m$ . The gain  $G(s_i)$  of each sample,  $s_i$ , will be computed by the use of the ray-casting algorithm, where sensor aspects will be considered. The sample  $s_i$  with maximum gain will be selected, and the area covered by this sample will be plotted over the map by the **PLOT** function. This is made to exclude the area from the sample generation in successive iterations. Next, the actual number of nodes,  $N_{act}$ , will be incremented by one. This algorithm will be repeated until reaching a percentage of threshold coverage  $U_{thresh}$ . Afterwards, the actual number of required nodes,  $N_{act}$ , will be returned. However, this algorithm is not guaranteed to give an optimal solution for the required number of covering nodes, but it is practical since the robot cannot be accurately positioned at the optimal nodes. Near-optimal solutions can be achieved by increasing the number of randomly generated samples. This will give a chance to fairly distribute the nodes over the entire space at the expense of the computing time. Figure 6 shows the CM values,  $CM$ , for different environments:  $n=100$  random samples,  $FOV=360^\circ$ , the coverage termination criterion is limited to  $U_{thresh}=99\%$  and with different sensor ranges (on the top row,  $R_{max}=2m$  and on the bottom,  $R_{max}=4m$ ). It can be noted that those metric values are obtained over one run with a large  $n$  to eliminate the randomness effect over the output of the algorithm. Additionally, they are nearly independent of the sensor parameters, since  $N_{est}$  and  $N_{act}$  are varied in the same manner for different sensor parameters.

#### 4.2 Performance evaluation index

The performance of an exploration strategy is usually measured by four metrics, namely: the distance travelled,  $D$ , the exploration time,  $T$ , the number of nodes created in the tree,  $N_{nodes}$ , and the completeness,  $C$ . The distance travelled is defined as the total distance travelled by a robot after returning back to the home position, while the exploration time is defined as the time taken by a robot to complete the exploration process. Completeness is the percentage of total area covered after the homing step. However, there is always a trade-off among these metrics by which the performance of an exploration strategy can be evaluated. In order to avoid this trade-off, a single  $EI$  is proposed here. This  $EI$  encapsulates all the metrics in just one number. Intuitively, the proposed index can be formulated based on its relationship with the mentioned metrics.  $EI$  is proposed to be directly proportional to the completeness,  $C$ , and inversely proportional to the normalized exploration time,  $\bar{T}$ , the normalized distance travelled,

$\bar{D}$ , and the normalized number of nodes in the tree structure,  $\bar{N}$ . This proposed index is useful for comparing the performance of different SRT-like algorithms. The larger the values of this index, the better the performance of a given strategy. This proposed index can be formulated as follows:

$$EI = \frac{w_c C}{w_t \bar{T} * w_d \bar{D} * w_n \bar{N}} \quad (3)$$

where  $w_c$ ,  $w_t$ ,  $w_d$  and  $w_n$  are the proportional weights which are added to measure the importance of each normalized metric for the  $EI$ . In the proposed approach, each of those weights is set to unity since the four metrics are equally important. The completeness is defined as:

$$C = \frac{\text{Known Cells}}{\text{Map Area}} * 100\% \quad (4)$$

The normalized number of nodes is made here with respect to the near-optimal number of actual nodes,  $N_{act}$ , calculated from the greedy randomized art-gallery algorithm, and so  $\bar{N}$  is defined as:

$$\bar{N} = \frac{N_{nodes}}{N_{act}} \quad (5)$$

The distance travelled is normalized to the total distance required for a robot to explore the entire environment, which can be estimated using the actual number of nodes calculated. The distance between two nodes in the coverage problem based on graph theory [28] is equal to double the maximum sensor range, and so the total distance,  $D_{total}$ , required can be approximated to the following:

$$D_{total} = 2 * R_{max} * (N_{act} - 1) \quad (6)$$

As such, the normalized travel distance is given as:

$$\bar{D} = \frac{D}{D_{total}} = \frac{D}{2 * R_{max} * (N_{act} - 1)} \quad (7)$$

The normalized exploration time is given as:

$$\bar{T} = \frac{T}{2 * R_{max} * (N_{act} - 1)} * v \quad (8)$$

where  $v$  is the average speed of the robot, which is a fixed value during the exploration process. Note that the exploration time measured here is the total simulation time, including the computation time of the algorithm. This is



why the normalized value of the distance travelled differs from the normalized exploration time.

## 5. Simulation results

Several simulation scenarios have been implemented to validate the proposed exploration approach. Each scenario has been identified by its environmental CM and the exploration  $EI$ . The 3D mobile robot simulator Webots [29] was used in all our simulations. A three-wheel omnidirectional mobile robot was used. The robot has a diameter of 0.2 m and carries a  $360^\circ$  laser range finder with one-degree angular resolution. In the simulations, the parameters for the SRT algorithm were selected as follows:  $\alpha=0.9$ ,  $d_{min}=0.7m$ ,  $G_{thresh}=100$  cells and  $R_{max}=2m$ . The performance of the developed approach was compared with the basic SRT approach through several metrics, representing the effort paid (the distance travelled,  $D$ , the exploration time,  $T$ ), the coverage gained (the completeness,  $C$ ) and the number of nodes created in the tree,  $N_{nodes}$ , as well as the exploration  $EI$ . Two environmental scenarios are presented here, namely an office-like environment and a maze-like environment.

### 5.1 The office-like environment

The proposed exploration approach with heuristic backtracking and the basic SRT strategy are implemented in the office-like environment shown in Fig. 7. The environmental CM for this environment is calculated using (1) as 0.31. Detailed simulation steps at different simulation times with the associated roadmap are shown in Fig. 8. The nodes created and the paths travelled are shown in green and blue, respectively, while the shortest paths planned by the heuristic approach are shown in red. In a basic SRT strategy, red edges mean that the robot has backtracked along them to return to its home position, which does not appear in the proposed backtracking approach as the robot is not required to go back over all previous nodes (it is just required to plan a short path to the most informative nodes). A comparison between the proposed approach and the basic SRT approach is given in Table 1 in terms of the mentioned metrics at different sensor ranges, namely 1 m and 2 m. Due to the random behaviour of SRT, values are averaged over five simulation runs with different initial robot positions. The standard deviations are also shown for the four metrics.

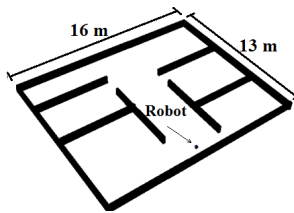


Figure 7. Office-like environment

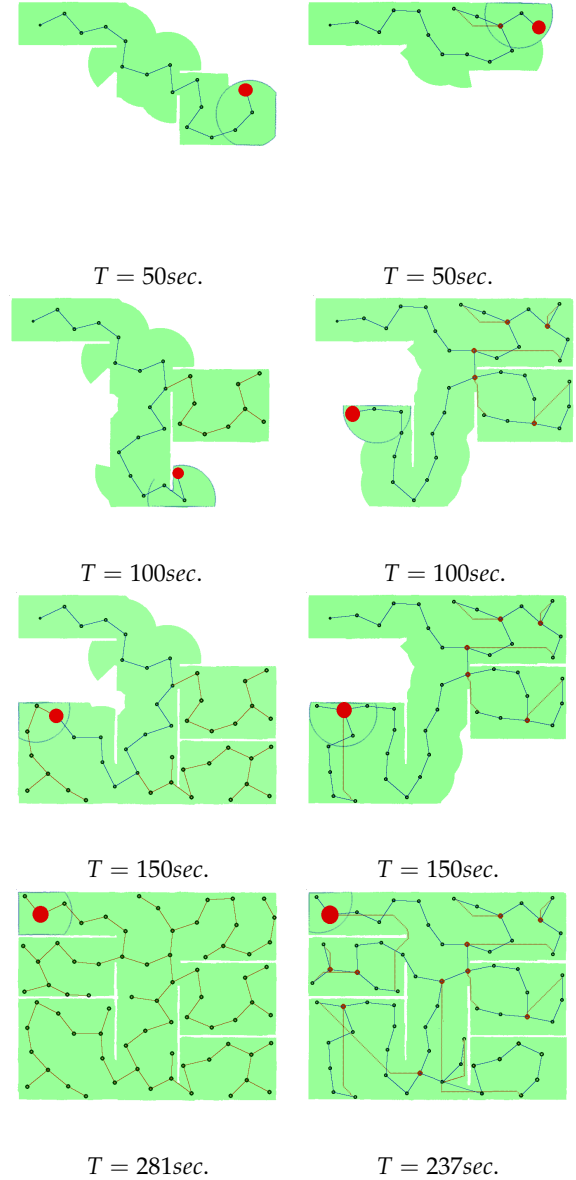


Figure 8. Simulation steps at different simulation times  $T$  with the associated roadmap. Left: the basic SRT approach. Right: the proposed approach.

It can be noted from Table 1 that there is a significant decrease in the total path length and the total exploration time for the proposed approach compared to the basic SRT approach. Furthermore, it can be observed that a significant reduction in the exploration distance appears clearly in the scenario with a smaller sensor range, which can be attributed to the high number of exploration edges required to fill the entire open space. Additionally, the proposed approach provides nearly complete coverage, which is comparable to that of the original SRT approach, while exerting less exploration effort. Furthermore, the standard deviations of the four metrics are shown to be small values, which proves the high reliability of the results. The new  $EI$  is also shown to be representative of the exploration performance avoiding trade-off among the metrics. The proposed approach showed higher evaluation indices

compared to those of the basic SRT at different perceptual ranges. The average speed of the robot is set to  $10\text{ m/s}$  and each of the proportional weights measuring the contribution of each metric to the index is taken to be one. In addition, it is worth noting that the complexity of exploration process is reduced through the decreased number of tree nodes.

Strategy	$T$ (sec.)	$C$ (%)	$D$ (m)	$N_{nodes}$	$EI$
Perceptual Range $R_{max} = 1\text{ m}$					
Basic SRT	$536 \pm 6$	$97.2 \pm 1$	$336.5 \pm 5$	$193 \pm 3$	7.3
Proposed Approach	$360 \pm 4$	$97.1 \pm 1$	$217.7 \pm 4$	$182 \pm 6$	18.2
% of Benefit	32.8	-0.1	35.3	5.7	145
Perceptual Range $R_{max} = 2\text{ m}$					
Basic SRT	$281 \pm 3$	$99 \pm 1$	$180 \pm 2$	$52 \pm 3$	16.12
Proposed Approach	$237 \pm 3$	$98.5 \pm 1$	$146.95 \pm 3$	$50 \pm 2$	46.22
% of Benefit	15.7	-0.5	18.4	3.8	186

**Table 1.** Simulation results of the office-like environment

Here, it is imperative to discuss the advantages of the proposed backtracking approach over the approaches presented in [17] and [18]. The backtracking approach in [17] is based on constructing a bridge between two adjacent nodes with a common safe region between them. This approach is helpful in corridor-like environments, which have a low probability of having in-between nodes with unexplored regions. Moreover, in [18] the backtracking approach is based on constructing a short cut between the initial and the most recently visited nodes without taking into account the possibility of there being a valuable node between them. This demonstrates the advantage of the proposed backtracking approach over these two approaches, since the proposed heuristic backtracking is based on approaching the most informative node which has a high probability of having unexplored regions in office-like or even in open-space environments, as illustrated in the results.

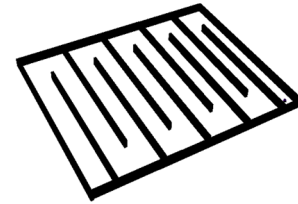
### 5.2 The maze-like environment

Similarly, simulations of the proposed exploration approach with heuristic backtracking and the basic SRT strategy were conducted in a maze-like environment, as shown in Fig. 9. The environmental CM for this environment is calculated using (1) as 0.42. The simulation results of the basic SRT are presented in Fig. 10(a) while those of the proposed SRT are shown in Fig. 10(b), in which the red lines represent the shortest paths followed whenever there are no valuable nodes to travel to. Table 2 summarizes the results obtained with a 2 m perceptual range averaged over five simulation runs with different initial robot positions.

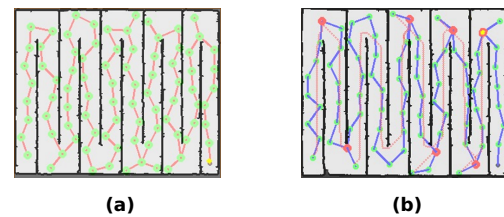
Similarly, the significant benefit in the total path length and the total exploration time are shown for the proposed approach compared with those of the basic SRT approach. In addition, the proposed approach showed a higher  $EI$  compared to that of the basic SRT.

Strategy	$T$ (sec.)	$C$ (%)	$D$ (m)	$N_{nodes}$	$EI$
Perceptual Range $R_{max} = 2\text{ m}$					
Basic SRT	$347 \pm 3$	$99 \pm 1$	$276.71 \pm 3$	$87 \pm 2$	12.17
Proposed Approach	$291 \pm 2$	$98.5 \pm 1$	$219.55 \pm 2$	$85 \pm 2$	18.56
% of Benefit	16	-0.5	20.65	2.3	50

**Table 2.** Simulation results of the maze-like scenario



**Figure 9.** Maze-like environment



**Figure 10.** Simulation results for the maze-like scenario: (a) the basic SRT approach, (b) the proposed approach

## 6. Conclusions

A novel heuristic backtracking algorithm has been developed for sensor-based random tree exploration to reduce the exploration time and the distance travelled so as to cope with time-critical applications. The new approach is based on the selection of the most informative node to approach rather than backtracking across all unnecessary explored areas. The enhancement of SRT exploration using the developed backtracking algorithm has been confirmed by conducting several simulations in different exploration scenarios. A new evaluation index has been devised to encapsulate the exploration metrics, namely the exploration time, the distance travelled, the coverage, and the number of nodes in a single number, avoiding trade-off among these metrics. This index has been shown to be representative of the exploration performance to be used for exploration comparison, especially in SRT-like algorithms. A step towards finding a unique complexity metric has also been developed,

representing the environment's structural complexity from the exploration point of view. This metric reflects the difference between the ideal number of visits required to fully cover the environment and the actual number of visits required. It has been shown that this complexity metric is dependent upon the disparity among obstacles in the environment. Although the estimated number of nodes ultimately depends upon the sensor properties, such as range, field of view and angular resolution, the complexity metric is nearly independent of these properties; since these properties will be reflected in both the actual and the estimated number of nodes required to cover the environment, the complexity metric will therefore not change. In the future, we will explore the potential of the evaluation index in diverse scenarios as a step towards generalizing it for the measurement of exploration performance and comparing between techniques.

## 7. . References

- [1] B. Yamauchi, A frontier-based approach for autonomous exploration, *Int. Symp. on Computational Intelligence in Robotics and Automation, CIRA*, (1997) 146-151.
- [2] H. H. Gonzalez-Banos and J.-C. Latombe, Navigation strategies for exploring indoor environments, *The Int. J. of Robotics Research*, 21 (2002) 829-848.
- [3] T. Tao, Y. Huang, F. Sun, and T. Wang, Motion planning for SLAM based on frontier exploration, *Int. Conf. on Mechatronics and Automation, ICMA*, (2007) 2120-2125.
- [4] D. Holz, N. Basilico, F. Amigoni, and S. Behnke, Evaluating the efficiency of frontier-based exploration strategies, *6th German Conf. on Robotics (ROBOTIK)*, (2010) 1-8.
- [5] R. C. Arkin, *Behavior-based robotics*, 1998.
- [6] H. Lau, Behavioral approach for multi-robot exploration, *Australasian Conf. on Robotics and Automation*, (2003).
- [7] D. Schmidt, T. Luksch, J. Wettach, and K. Berns, Autonomous behavior-based exploration of office environments, *Int. Conf. on Informatics in Control, Automation and Robotics, ICAR*, (2006) 235-240.
- [8] T. Balch, Avoiding the past: a simple but effective strategy for reactive navigation, *Int. Conf. on Robotics and Automation, ICRA*, (1993) 678-685.
- [9] M. Juliá, O. Reinoso, A. Gil, M. Ballesta, and L. Payá, A hybrid solution to the multi-robot integrated exploration problem, *Engineering Applications of Artificial Intelligence*, 23, 4, (2010) 473-486.
- [10] M. Abdellatif, Behavior fusion for visually guided service robots. *I-Tech*, Vienna, Austria (2008) 1-12.
- [11] M. Juliá, A. Gil, and O. Reinoso, A comparison of path planning strategies for autonomous exploration and mapping of unknown environments, *Autonomous Robots*, 33, 4 (2012) 427-444.
- [12] J. Barraquand, L. Kavraki, R. Motwani, J.-C. Latombe, T.-Y. Li, and P. Raghavan, A random sampling scheme for path planning, *Robotics Research. Springer*, (2000) 249-264.
- [13] G. Oriolo, M. Vendittelli, L. Freda, and G. Troso, The SRT method: Randomized strategies for exploration, *Int. Conf. on Robotics and Automation*, 5. (2004) 4688-4694.
- [14] L. Freda and G. Oriolo, Frontier-based probabilistic strategies for sensor-based exploration, *Int. Conf. on Robotics and Automation, ICRA*, (2005) 3881-3887.
- [15] L. Freda, G. Oriolo, and F. Vecchioli, Sensor-based exploration for general robotic systems, *Int. Conf. on Intelligent Robots and Systems, IROS*, (2008) 2157-2164.
- [16] L. Freda, G. Oriolo, and F. Vecchioli, An exploration method for general robotic systems equipped with multiple sensors. *Int. Conf. on Intelligent Robots and Systems, IROS*, (2009) 5076-5082.
- [17] A. Franchi, L. Freda, G. Oriolo, and M. Vendittelli, The sensor-based random graph method for cooperative robot exploration, *IEEE/ASME Transactions on Mechatronics*, 14, 2, (2009) 163-175.
- [18] J. Kim, K. J. Seong, and H. J. Kim, An efficient backtracking strategy for frontier method in sensor-based random tree, *Int. Conf. on Control, Automation and Systems, ICCAS*, (2012) 970-974.
- [19] S. Read, The grain of space in time: the spatial/functional inheritance of Amsterdam centre, *Urban Design Int.*, 5, 3-4, (2000) 209-220.
- [20] D. A. Shell and M. J. Mataric, Human motion-based environment complexity measures for robotics, *Int. Conf. on Intelligent Robots and Systems, IROS*, 3 (2003) 2559-2564.
- [21] A. Rnyi, On measures of entropy and information, *Fourth Berkeley Symp. on Mathematical Statistics and Probability*, (1961) 547-561.
- [22] G. T. Anderson and G. Yang, A proposed measure of environmental complexity for robotic applications, *Int. Conf. on Systems, Man and Cybernetics, SMC*, (2007) 2461-2466.
- [23] M. Vendittelli, L. Freda, and G. Oriolo. The SRT method. <http://www.dis.uniroma1.it/labrob/arch/SRT.html>. Accessed Jan 2013.
- [24] S. D. Roth, Ray casting for modeling solids, *Computer graphics and image processing*, 18, 2, (1982) 109-144.
- [25] T. Ersson and X. Hu, Path planning and navigation of mobile robots in unknown environments, *Int. Conf. on Intelligent Robots and Systems, IROS*, 2 (2001) 858-864.

- [26] M. Cardei and J. Wu, Energy-efficient coverage problems in wireless ad hoc sensor networks, *Computer Communications*, 29, 4, (2006) 413-420.
- [27] H. González-Báños, A randomized art-gallery algorithm for sensor placement, *The 17th. Annual Symposium on Computational Geometry*. ACM, (2001) 232-240.
- [28] D. B. West et al., *Introduction to graph theory*, 2 (2001).
- [29] Webots website, "<http://www.cyberbotics.com>," Commercial Mobile Robot Simulation Software. Accessed Jan 2013.
- [30] H. El-Hussieny, S. F. M. Assal, and M. Abdellatif, Improved sensor-based mobile robot exploration of novel environments, *6th. Int. Conf. on Intelligent Computing and Information Systems, ICICIS*, (2013) 43-49.