Cooperative Robot Exploration Strategy Using an Efficient Backtracking Method for Multiple Robots

Jinho Kim
Department of Mechanical Engineering,
University of Maryland, Baltimore County,
Baltimore, MD 21250
e-mail: umbcjhkim@umbc.edu

Stephanie Bonadies[1]
Department of Mechanical Engineering,
University of Maryland, Baltimore County,
Baltimore, MD 21250
e-mail: stephanie.bonadies@ngc.com

Charles D. Eggleton
Department of Mechanical Engineering,
University of Maryland, Baltimore County,
Baltimore, MD 21250
e-mail: eggleton@umbc.edu

S. Andrew Gadsden[2]
Fellow ASME
College of Engineering and Physical Sciences,
University of Guelph,
Guelph, ON N1G 2W1, Canada
e-mail: gadsden@uoguelph.ca

This paper presents a cooperative robot exploration (CRE) strategy that is based on the sensor-based random tree (SRT) star method. The CRE strategy is utilized for a team of mobile robots equipped with range finding sensors. Existing backtracking techniques for frontier-based (FB) exploration involve moving back thorough the previous position where the robot has passed before. However, in some cases, the robot generates inefficient detours to move back to the position that contains frontier areas. In an effort to improve upon movement and energy efficiencies, this paper proposes the use of a hub node that has a frontier arc; thereby, the robots backtrack more directly to hub nodes by using the objective function. Furthermore, each robot cooperatively explores the workspace utilizing the data structure from the entire team of robots, which consists of configuration data and frontier data. Comparative simulations of the proposed algorithm and the existing SRT-star algorithm are implemented and described. The experiment is presented to demonstrate the application of the proposed strategy in real-time. Utilizing the proposed algorithm and exploration strategy, the results indicate that a team of robots can work more efficiently by reducing the distance of exploration and the number of node visited. [DOI: 10.1115/1.4041332]

1 Introduction

In recent years, many researchers have focused on mobile robot motion planning [1–4]. Various strategies of planning robot motion assume that mobile robots already have access to a map of their operating location and can navigate using this information. For example, potential field [5,6], Voronoi diagram [7], A^* algorithm [8] are map-based path planning methods for robots navigation. These algorithms are not, however, applicable for planning robot motion in an unknown environment such as would be the case during search and rescue missions in dangerous buildings, disaster assessment, reconnaissance.

The missions stated above are considered exploration which is defined as the act of moving through an unknown environment while building a map that can be used for subsequent navigation [9]. Most of the existing exploration techniques fall under the class *frontier-based (FB) exploration* proposed in Ref. [10]. In this framework, a robot enters an unknown indoor environment and explores and maps cluttered rooms using only a laser scanner or sonar sensors [11–13]. Based on this technique, many sensor-based exploration methods have been developed. Among them, the sensor-based random tree (SRT) method [14] is one of the most effective exploration methods. This method makes a data structure using the randomized generation of position that is referred to as the SRT. This SRT can be considered as a sensor-based version of the rapidly exploring random tree proposed in Ref. [15]. Exploration strategies for a single robot have been developed in Refs. [16–19].

After single robot exploration strategies were successfully developed, researchers turned their focus to exploration with multiple robots [20–23] because of its advantages over the single robot case [20]. First, the exploration mission in a multiple robots setting is generally completed faster when compared with a single robot (i.e., the "sum of the whole"). Second, multiple robots can localize themselves with increasing map accuracy and quality if they share information [21].

In this paper, we consider the problem of cooperative robot exploration (CRE) of an unknown environment. This paper is based on the concept of the FB strategy using the SRT, which is called as FB-SRT. In Ref. [23], a team of robots cooperatively explored an unknown environment in the form of a graph, called sensor-based random graph. The sensor-based random graph method gave satisfactory results with a team of mobile robots. Nonetheless, it has the possibility that each robot can take a long detour route to backtrack in order to continue exploring, which decreases the efficiency of the exploration. In order to minimize the disadvantage, we propose an algorithm that allows a robot to backtrack along the most direct route to a position that has a frontier arc, rather than using an indirect detour route. This is the main contribution of the work described in this paper. Furthermore, we devise a hub node, which is the target point with at least one frontier arc to perform efficient backtracking, and also cooperatively explore other robots' SRT.

2 Problem Setting

In this section, we state some assumptions and explain robot kinematics.

2.1 Assumptions. First, we state the basic assumptions used in this paper.

(1) The workspace $\mathcal{W}$ is an n-dimensional Euclidean space R^n, where $n = 2$ or 3. In this paper, we consider only case $n = 2$.
(2) The shape of the robot is circular and it is free to move in any direction.
(3) The robot knows its configuration q containing its position and heading angle.
(4) The robot has sensory equipment that provides the robot information of the surrounding area within sensor range R_s. This information is called a local safe region and denoted by $\mathcal{S}$.
(5) From $\mathcal{S}$, the robot calculates the frontier and saves it in the data structure called frontier data (FD) denoted by $\mathcal{F}$.

[1]Present address: Northrop Grumman Corporation, Baltimore, MD 21240.
[2]Corresponding author.

Contributed by the Mechanisms and Robotics Committee of ASME for publication in the Journal of Mechanisms and Robotics. Manuscript received September 23, 2017; final manuscript received August 22, 2018; published online October 1, 2018. Assoc. Editor: Xilun Ding.

(6) Each robot can share its q and $\mathcal{F}$ within a communication range R_c at any time. In this paper, we assume R_c is large enough to cover all the $\mathcal{W}$.

The proposed algorithm operates under these assumptions to alleviate the complexity of the problem. We use a robot with a laser range scanner as sensory equipment to perceive obstacles. The plane of a laser range scanner is a subset of R^2, so we assume that obstacles are not below the laser plane. In addition, the robot dynamics plane is a subset of R^2. Therefore, we consider just R^2 as a workspace. Assumption 2 implies that the robot uses a turn-and-go scheme that allows the robot to move in any direction. This is presented in the next subsection. In assumption 4, the FB-SRT strategy is divided into two categories according to the shape of $\mathcal{S}$. We utilize the FB-SRT-Star method, which has a star-shaped $\mathcal{S}$, because it is more efficient than the other FB-SRT methods [14]. Assumption 5 is proposed for multirobot exploration with frontier method using SRT presented in Sec. 3. With these assumptions, our ultimate objective is explained in the paper: cooperative exploration of unknown environments using multiple robots. A more specific explanation will be given in the rest of the paper with simulations and experimental results.

2.2 Robot Kinematics. A brief explanation of the robot kinematics of this scheme is presented in this subsection.

A kth robot $(k = 1, 2, \ldots, N)$ with radius of r has a position $(x_k,\, y_k) \in \mathcal{W}$ and a heading angle, θ_k, as depicted in Fig. 1. Each robot has two wheels and its own laser range scanners to measure the surrounding of each robot. A robot uses the turn-and-go scheme to move in any direction; the formulation of the robot kinematics is as follows:

$$\dot{x}_k = V_c \cos\theta_k \tag{1}$$

$$\dot{y}_k = V_c \sin\theta_k \tag{2}$$

$$\dot{\theta}_k = \omega_c \tag{3}$$

where x_k and y_k are positions of kth robot, and θ_k is a heading angle of kth robot. In Eqs. (1)–(3), the velocity and angular velocity of robot, V_c and ω_c, can be written as

$$V_c = \frac{V_r + V_l}{2} \tag{4}$$

$$\omega_c = \frac{V_r - V_l}{2r} \tag{5}$$

where V_l and V_r are left and right wheel speeds, respectively.

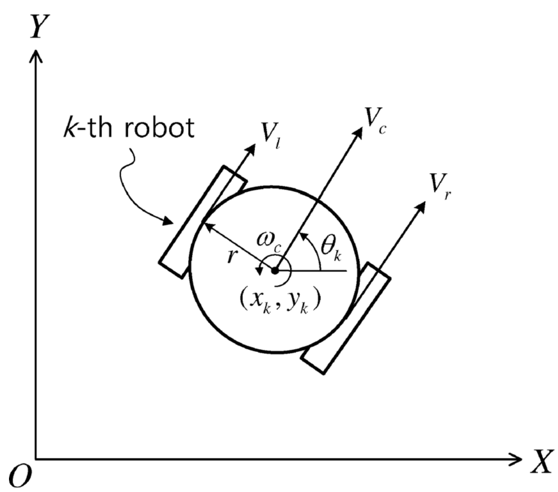

Fig. 1 The kth robot located at $(x_k,\, y_k)$

3 Frontier-Based Strategies for Sensor-Based Exploration

In the FB-SRT strategy, the workspace is divided into the explored and unexplored regions as depicted in Fig. 2. Since there are no obstacles in the explored region, a robot can move everywhere in this region, and it is called the *safe region*. The explored region is also divided into three parts. If a boundary exists between the obstacles and explored region, it is referred to as an *obstacle arc*. The boundary between the explored region and an unexplored region is referred to as a *frontier arc*. However, in the case when a configuration covers the frontier arc of another configuration, this frontier arc turns into a *free arc*.

Algorithm 1 FB_SRT_EXPLORATION [14]

Require: $q_{\text{init}}, K_{\max}, I_{\max}, \alpha, d_{\min}$
$q_{\text{curr}} = q_{\text{init}}$
for $k = 1$ to $K_{\max}$ **do**
  $\mathcal{S}(q_{\text{curr}}) \Leftarrow \mathit{OBSTACLE\_SCAN}(q_{\text{curr}})$
  $\mathcal{F}(q_{\text{curr}}) \Leftarrow \mathit{FRONTIER\_SCAN}(q_{\text{curr}}, \mathcal{S}(q_{\text{curr}}))$
  $\mathcal{T} \Leftarrow \mathrm{SAVE\_}\mathcal{T}(q_{\text{curr}}, \mathcal{S}(q_{\text{curr}}), \mathcal{F}(q_{\text{curr}}))$
  $i = 1$
repeat
  $\theta_{\text{rand}} \Leftarrow \mathit{RAND\_DIR}$
  $l \Leftarrow \mathit{RANGE}(\mathcal{S}(q_{\text{curr}}), \theta_{\text{rand}})$
  $q_{\text{cand}} \Leftarrow \mathit{Q\_CAND}(q, \theta_{\text{rand}}, \alpha \cdot l)$
  $i = i + 1$
until $(\mathit{CHECK}(q_{\text{cand}}, d_{\min}, \mathcal{T})$ or $i = I_{\max})$
if $(\mathit{CHECK}(q_{\text{cand}}, d_{\min}, \mathcal{T})$ ***then***
  $\mathit{MOVE\_TO}(q_{\text{cand}})$
  $q_{\text{curr}} \Leftarrow q_{\text{cand}}$
else
    $\mathit{MOVE\_TO}(q_{\text{cand,parent}})$
    $q_{\text{curr}} \Leftarrow q_{\text{cand,parent}}$
  end if
end for
return q_{curr}

A pseudocode description of the FB-SRT algorithm is shown in algorithm 1. To perform this algorithm, the initial position of robot q_{init}, maximum iteration number $K_{\max}$, the number of sector $I_{\max}$ and step movement constant α, and minimum step movement $d_{\min}$ are required.

In the first step, sensory equipment scans the robot's surroundings. Next, the algorithm collects the $\mathcal{S}$ at the current configuration and $\mathcal{F}$ using the *OBSTACLE_SCAN* and *FRONTIER_SCAN* functions, respectively. A random direction, θ_{rand}, is then generated and the radius l of $\mathcal{S}$ along θ_{rand} is computed. Finally, this algorithm generates the next target position, q_{cand}, by taking a step movement, α, multiplied by l along θ_{rand}. When q_{cand} is generated in the previous step, a robot moves to q_{cand} and this position is updated to q_{curr}. However, if there is no next target position generated, a robot moves to a point $q_{\text{cand,parent}}$, which has a frontier arc through q_{prev}.

The FB-SRT strategy is divided into two categories according to the shape of $\mathcal{S}$: FB-SRT-Ball and FB-SRT-Star. The FB-SRT-Star

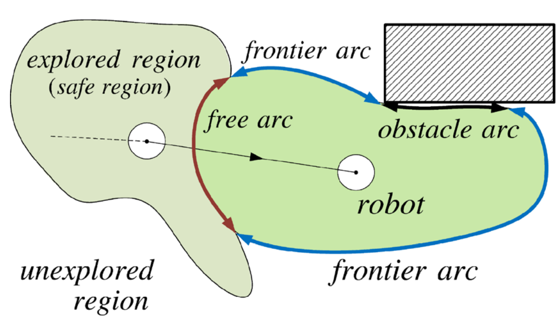

Fig. 2 The division of workspace in FB-SRT strategy

method can allow for more efficient movements than the FB-SRT-ball method [14]. Because the FB-SRT-star considers a unique movement radius for each direction depending on the surrounding environment, the FB-SRT-ball limits the sensor range in all directions in the same R_s. As such, we will consider only the FB-SRT-star method in this paper.

In the FB-SRT-star method, $\mathcal{S}$ is a star-shaped region, which consists of several cone-shaped sectors as shown in Fig. 3. Each cone can be defined by three points when frontiers exist. The *mid–point*, can be defined as a point, which is placed along the middle axis of the cone at a distance of the full scanning range. The other points, *right–point* and *left–point*, can be defined as frontier points when there is a long gap between adjacent sectors. In Fig. 4, the thick outer line of $\mathcal{S}$ represents frontier arcs while the remaining portions represent either free arcs or obstacle arcs. Furthermore, Fig. 4 depicts an example of the FB-SRT-star exploration. In this figure, a robot at position 2 has frontier arcs that are thick outer lines.

4 Frontier-Based-Sensor-Based Random Tree-Star for Cooperative Robot Exploration

In this section, the CRE strategy is proposed and explained. The proposed CRE strategy is based on the concept of the FB-SRT-star method that was presented in Sec. 3. Here, we modify the FB-SRT-star method to not only apply it to a team of robots, but also to generate an efficient backtracking path.

In the CRE strategy, the data tree of each q consists of configuration data, $\mathcal{Q}\_data$, and frontier data, $\mathcal{F}\_data$. Mathematically, $\mathcal{F}\_data$ of q_{ij} can be written as follows:

$$\mathcal{F}_{q_{ij}}(k) = \begin{cases} -1 & \text{if} \quad \text{obstacle arc} \\ 0 & \text{if} \quad \text{free arc} \\ 1 & \text{if} \quad \text{frontier arc} \end{cases} \tag{6}$$

where i is the robot number, j is the node number, and k is the number of the frontier sector. By assumption 6, robots can share the corresponding SRT data with other robots. Thereby each robot reconstructs their own SRT trees by considering SRT data provided by other robots.

Figure 5 shows an example of $\mathcal{F}\_data$ revision in the CRE strategy. In this example, as Robot1 moves to q_{12} from q_{11}, $\mathcal{F}\_data$ of q_{12} can be depicted as thick outer lines (red). However, Robot2 is located around Robot1 at q_{12} such that the frontier arcs of q_{12} overlap the frontier arcs of Robot2. The proposed CRE strategy revises $\mathcal{F}\_data$ combining information from Robot1 and Robot2,

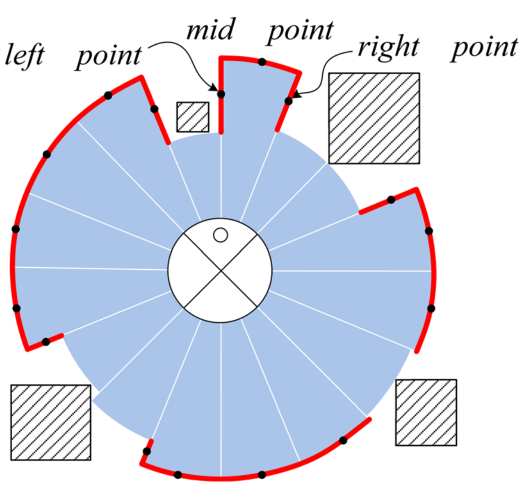

Fig. 3 The definition of frontier arcs with mid, left, right-point in FB-SRT-star

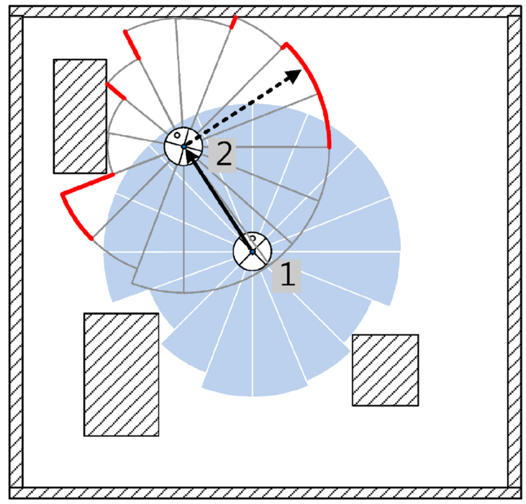

Fig. 4 An example of FB-SRT-star exploration. The robot is moving from 1 to 2, and the thick outer lines represent frontier arcs at 2.

and both robots will move to the frontier direction under the CRE strategy. For an efficient backtracking, we can define q_{ij} as a hub node, $q_{\text{hub}} \in \mathcal{H}$, when maximum of the set of $\mathcal{F}_{q_{ij}}$ is 1. Here, $\mathcal{H}$ is a set of hub nodes. Furthermore, the CRE strategy enables a team of robots to cooperatively explore a configuration by sharing other robots' data trees when its own $\mathcal{F}\_data$ no longer contains frontier arcs. In the following subsections, we present a pseudocode of the CRE algorithm first, and then explain the transfer and backtracking algorithms with the hub node.

4.1 Cooperative Robot Exploration Algorithm.

The pseudocode of the CRE algorithm is shown in algorithm 2. To perform this algorithm, we need $\mathcal{Q}\_data$, $\mathcal{F}\_data$, current position of each robot q_{curr}, and the number of robots N_{robot}.

Algorithm 2 COOPERATIVE_ROBOT_EXPLORATION

Require: $\mathcal{Q}\_data, \mathcal{F}\_data, q_{\text{curr}}, N_{\text{robot}}$
for $i = 1$ to N_{robot} **do**
  $q_{ij} = q_{\text{curr}}(i)$
  FB_SRT_EXPLORATION
  if $\max(\mathcal{F}_{q_{ij}}) \neq 1$ **then**
    $\mathcal{H}_i \Leftarrow FIND\_HUB(i)$
  if $\mathcal{H}_i \neq \phi$ **then**
    for $n = length(\mathcal{H}_i)$ to 1 **do**
      $q_{\text{target}}(i) \Leftarrow BACKTRACK(\mathcal{H}_i(n))$
      break;
    end for
  else
    $\mathcal{H}_m \Leftarrow FIND\_HUB\_MAX(i)$
   if $\mathcal{H}_m \neq \phi$ **then**
     $TRANSFER(\mathcal{H}_m)$
   else
     $q_{\text{target}}(i) \Leftarrow HOMING(q_init)$
   end if
  end if
 end if
end for

In the first step, each robot performs the FB-SRT exploration using $\mathcal{Q}\_data$ and $\mathcal{F}\_data$. If there are no frontier arcs at q_{curr} during an exploration, the algorithm finds the nearest q_{hub} in its own $\mathcal{F}\_data$ using the *FIND_HUB* function, and the robot move to q_{hub} using the *BACKTRACK* function. If there is no q_{hub} in its own $\mathcal{F}\_data$, the algorithm determines which q_{hub} has most frontier arcs in $\mathcal{F}\_datas$ of other robots using the *FIND_HUB_MAX* function. Then, the robot moves to q_{hub} using the *TRANSFER* function. When none of the robots have q_{hub}, robots move to their

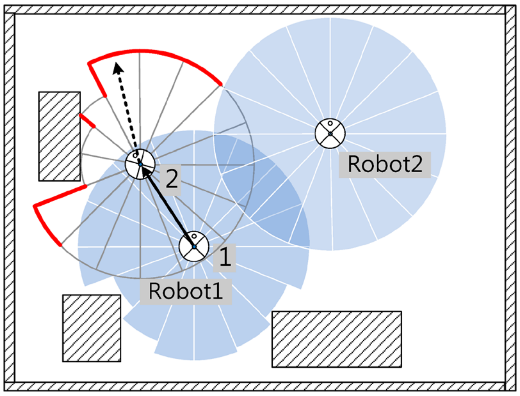

Fig. 5 An example of the CRE exploration. Robot1 is moving from 1 to 2, and thick outer lines represent frontier arcs of Robot1 at 2.

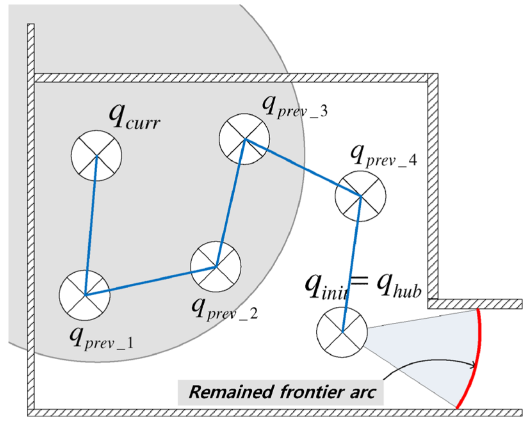

Fig. 6 An example of exploration using the FB-SRT-star method

initial positions using the *HOMING* function. The nonexistence of q_{hub} indicates that the entire workspace has been explored.

4.2 Transfer Algorithm. When the ith robot has no q_{hub} in its own data tree, $max\{\mathcal{F}_{q_{ij}}\} \neq 1$ for all j, the algorithm determines a q_{hub} in another tree which has the most hub nodes. Then, the q_{hub} of $\mathcal{H}_m$ becomes an input of the *TRANSFER* algorithm as shown in algorithm 3. At first, the ith robot moves to its initial position, q_{i1}, using the *BACKTRACK* function. When the robot reaches q_{i1}, it moves to $q_{m1,}$ which is the initial position of the mth robot. The robot then moves to the nearest q_{hub} of $\mathcal{H}_m$ using the *BACKTRACK* function. Since the robot expands its own data tree transferring from its own tree to another robots' tree using the *TRANSFER* algorithm, a team of robots can explore an unknown environment cooperatively.

Algorithm 3 TRANSFER

Require: $\mathcal{H}_m, i$
$q_{ij} = q_{\text{curr}}(i)$
$flag = 0$
if $\mathcal{H}_m \neq \phi$ **then**
  if $flag = 0$ **then**
    $q_{\text{target}}(i) \Leftarrow BACKTRACK(q_{i1})$
  if $q_{ij} = q_{i1}$ **then**
    $q_{\text{target}}(i) \Leftarrow BACKTRACK(q_{m1})$
    $flag = 1$
  end if
else
  for $n = 1$ to $length(\mathcal{H}_m)$ **do**
    $q_{\text{target}}(i) \Leftarrow BACKTRACK(\mathcal{H}_m(n))$
    break;
  end for
end if
end if

4.3 Backtracking Algorithm. Unlike the backtracking of the FB-SRT strategy, the proposed backtracking algorithm can direct a robot to q_{hub} more efficiently using only the position and frontier arc information of each node.

Figure 6 shows an example of exploration in which a robot explores from q_{init} to q_{curr} through the blue line. In this example, q_{init} has a frontier arc and can be designated as q_{hub}. To establish the most efficient path, a candidate set, $q_{\text{back\_cand}}$, of next backtracking positions, q_{back}, can be defined as written in the following equation:

$$q_{\text{back\_cand}} = \{q_{\text{back}} \mid d(q_{\text{curr}}, q_{\text{back}}) \leq d_c,\ q_{\text{back}} \in (q_{\text{curr}}, q_{\text{hub}})\} \quad (7)$$

where $d(\cdot,\cdot)$ is the Euclidean distance in two dimensions, and d_c is a range parameter for the next movement position. Note that d_c should be set to a value obtained by doubling R_s plus the margin to maximize the performance of the proposed method.

Next, the distances, d_{c_i}, between q_{curr} and $q_{\text{back\_cand}}(i)$, and d_{h_i}, between q_{hub} and $q_{\text{back\_cand}}(i)$, are calculated as shown in Fig. 7. Elements of $q_{\text{back\_cand}}$ are then sorted in increasing order of the following objective function's value:

$$J(q_{\text{back\_cand}}(i)) = c_1\, d_{c\_i} + c_2\, d_{h\_i} \quad (8)$$

where c_1 and c_2 are constant parameters. In this example, $q_{\text{back\_cand}}$ can be written as $q_{\text{back\_cand}} = [q_{\text{prev\_2}}, q_{\text{prev\_1}}, q_{\text{prev\_3}}]$.

Next, the algorithm validates the frontier directions of q_{curr} and each of the candidates of q_{back}. In the final step, the algorithm designates $q_{\text{prev\_2}}$ as q_{target} when the robot confirms that the frontiers of q_{curr} and $q_{\text{prev\_2}}$ are unobstructed using $\mathcal{F}\_data$. If the line connection of the two points is obstructed, the next shortest path of q_{back} is examined and the process is iterated until an unobstructed path to q_{target} is identified. This process is repeated until the robot reaches q_{hub}, while other robots continue exploring unexplored regions.

5 Simulations

In this section, simulation results are described to confirm the efficiency of the proposed CRE strategy for a team of mobile robots by comparing with the FB-SRT strategy. Each robot in this simulation is equipped with a 360 deg laser range scanner with a maximum scanning range of 20 units and $I_{\max}$ of 18 (20 deg interval). Simulations were performed in two types of environments

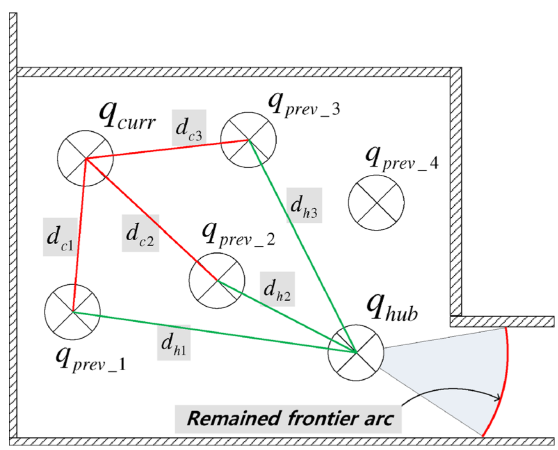

Fig. 7 Designation of $q_{\text{back\_cand}}$ and computing distances, d_{c_i}, between q_{curr} and $q_{\text{back\_cand}}(i)$, and d_{h_i}, between q_{hub} and $q_{\text{back\_cand}}(i)$

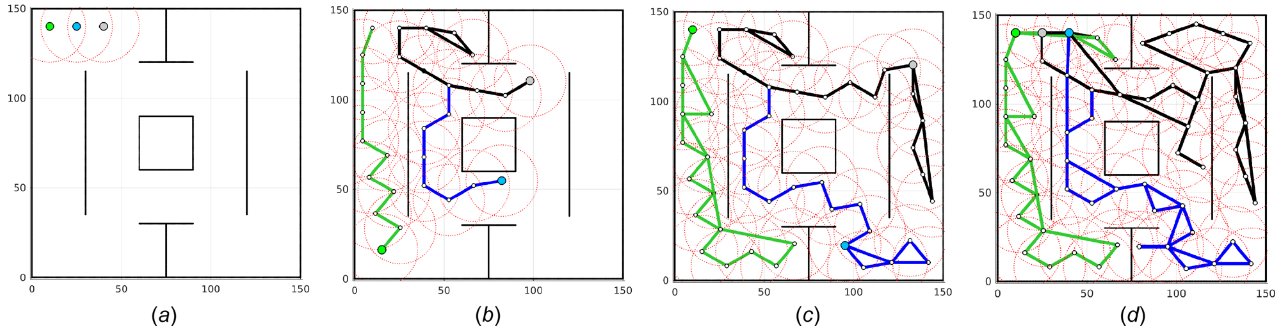

Fig. 8 Case 1: progress of the CRE with three robots in the square environment. The thin lines represent walls, the three large circles are robots, and the dotted circles around the robots are the sensor range areas, and the small white circles are nodes that the robot already passed by. (Iteration: (*a*) 0, (*b*) 10, (*c*) 20, (*d*) 44).

with a varying number of robots, ranging from one to four. The first environment is a square region with walls and a square-shaped pillar in the center of the space. The second is an office-like environment, which consists of a hallway and three rooms that can be accessed through only one entrance. Simulations were performed five times in each environment; then the performance index is evaluated in terms of the *number of visited nodes per robot* and the *traveled distance per robot*.

Figures 8 and 9 show the progress of the CRE with three robots. The thin lines represent walls, the three large circles are robots, and the dotted circles around each robot depict the sensor range areas. In addition, the small white circles are nodes that the robot has already visited. Thick lines in the map represent trajectories of each robot. The exploration of the workspace has been successfully completed when the sensor range areas cover the entire workspace. Simulations end when all the robots return to their initial positions, as seen in the last frame of Figs. 8 and 9.

5.1 Case 1: Maze-Like Square Environment. In case 1, the workspace is an square environment with some walls and a square-shaped pillar in the center. Its size is 150×150 as shown in Fig. 8.

The resulting number of visited nodes per robot and distance traveled per robot for the CRE strategy and the FB-SRT strategy are compared in Fig. 10. Both the number of visited nodes and distance traveled per robot is lower for the CRE strategy for all number of robots considered, demonstrating improved efficiency. As the number of robots increases, the number of visited nodes and the distance traveled per robot decreases. As the numerical simulation results show, when the CRE strategy is used, the number of visited nodes per robot decreases by 19.2–41.4% compared to utilizing the FB-SRT strategy. In addition, the distance traveled per robot decreases by 8.1–18.8% when using the CRE strategy compared to the FB-SRT strategy.

5.2 Case 2: Office-Like Environment. The workspace is an office environment with three rooms. In this case, the size of the simulation environment is 180×100 as shown in Fig. 9.

The results of this simulation compare the use of the CRE strategy to the FB-SRT strategy results in Fig. 11. As the number of robots increases, the number of visited nodes and the distance traveled per robot decreases. In addition, the distance traveled per robot when using the CRE strategy is always less than the distance traveled per robot when utilizing the FB-SRT strategy. The CRE strategy results in a 32.3–43.4% decrease in the number of visited nodes per robot, and 10.4–19.0% decrease in the distance traveled per robot decreases compared with the simulations using the FB-SRT strategy. In both environments, the CRE strategy demonstrates better efficiency when compared with the FB-SRT strategy.

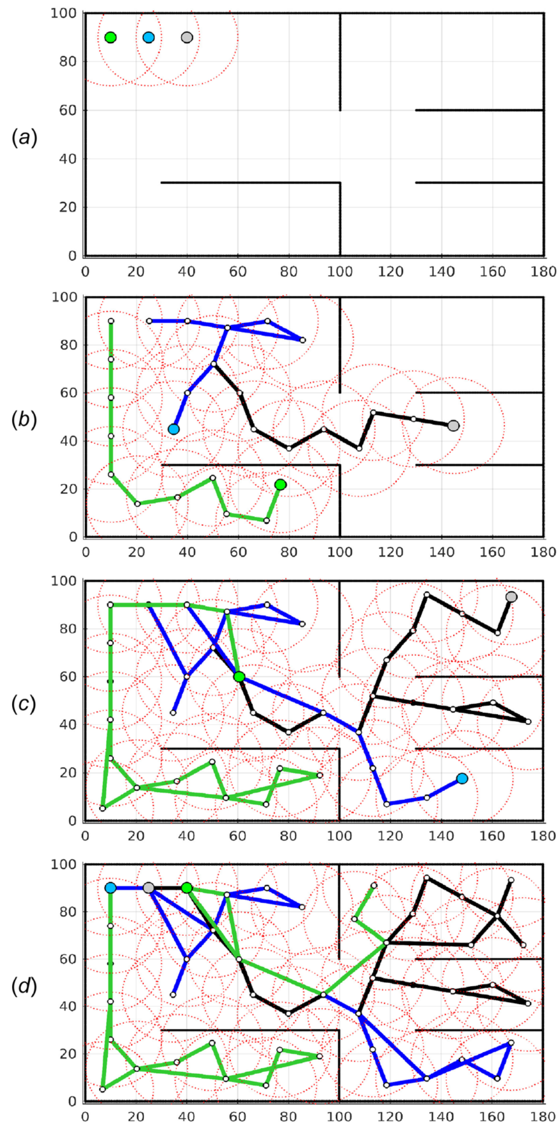

Fig. 9 Case 2: progress of the CRE with three robots in the office environment. The thin lines represent walls, the three large circles are robots, and the dotted circles around the robots depict the sensor range areas, and the small white circles represent nodes that the robot already passed by. (Iteration: (*a*) 0, (*b*) 10, (*c*) 20, (*d*) 31).

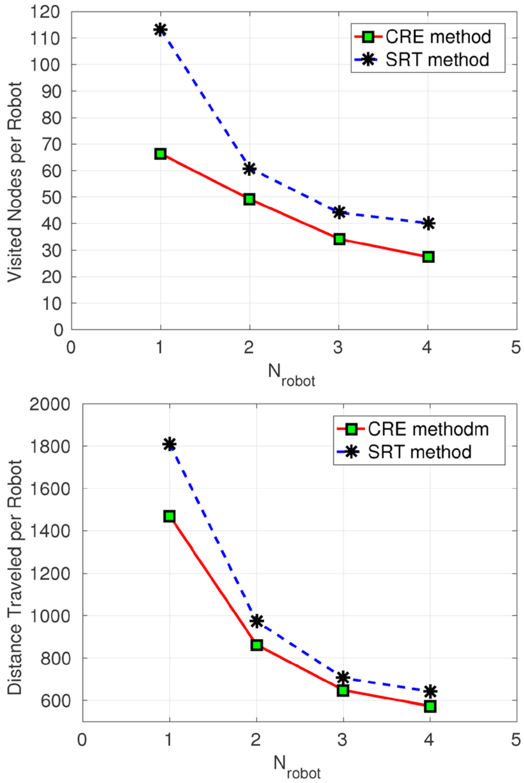

Fig. 10 Case 1: The number of visited nodes per robot (above) and the distance traveled per robot (below). Squares and asterisks are results of CRE algorithm and SRT algorithm, respectively.

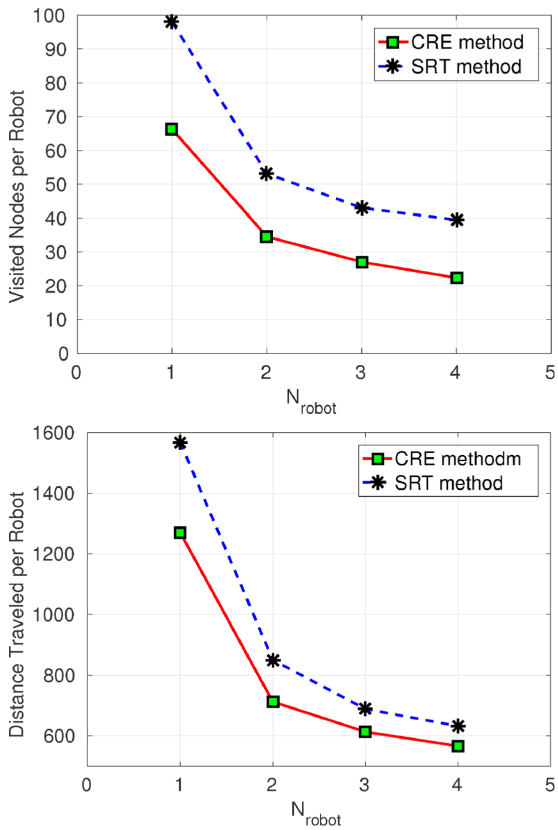

Fig. 11 Case 2: the number of visited nodes per robot (above) and distance traveled per robot (below). Squares and asterisks are results of CRE algorithm and SRT algorithm, respectively.

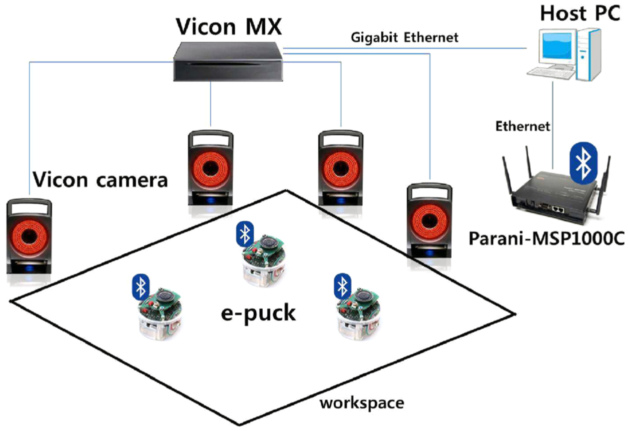

Fig. 12 Experiment setting

Table 1 Experimental parameters

Parameter	Value
$\mathcal{W}$ size	1.80 m × 1.08 m
$I_{\max}$	18
R_s	0.2 m
α	0.8
q_{init}	$q_{11} = [0.1\text{ m}, 0.9\text{ m}, -\pi/2]$ $q_{21} = [0.25\text{ m}, 0.9\text{ m}, -\pi/2]$ $q_{31} = [0.4\text{ m}, 0.9\text{ m}, -\pi/2]$

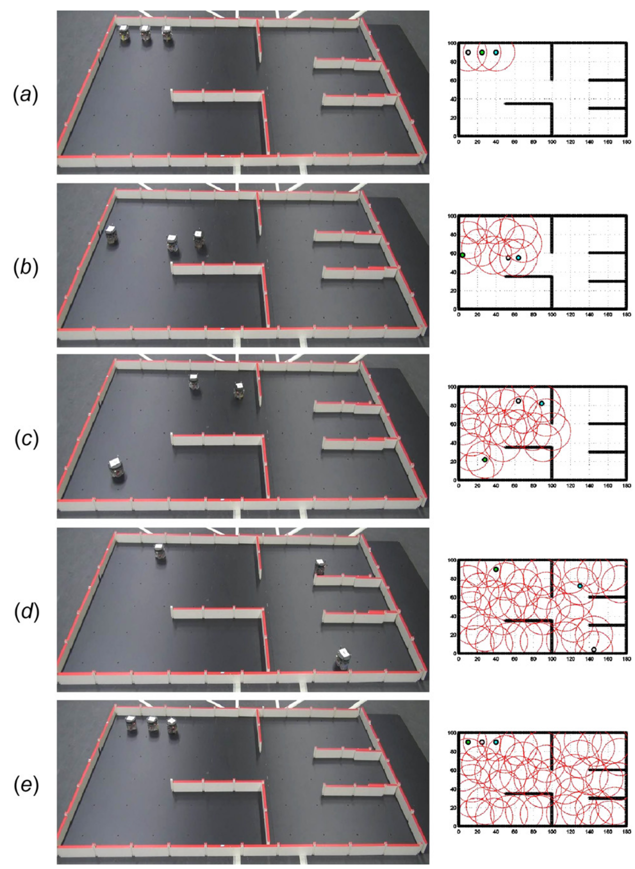

Fig. 13 The experiment of CRE with three e-puck robots ((*a*) 0 s, (*b*) 11 s, (*c*) 26 s, (*d*) 79 s, (*e*) 121 s)

Table 2 Numerical results of the experiment

	Robot 1	Robot 2	Robot 3	Mean	Total
The number of visited nodes	28	24	28	26.67	80
Traveled distance (m)	5.85	4.91	5.54	5.43	16.30
Exploration time (s)	119	79	121	106.33	319

6 Experiments

The proposed CRE strategy has been experimentally performed with the hardware system in this section. The experimental environment was composed as shown in Fig. 12, and the experimental parameters used for the experiment are shown in Table 1.

Figure 13 shows the progress of the exploration using the CRE strategy, and robot positions in the explored area are plotted on the right of each frame. In the line plots, the colored circles, and red circles represent the robots and sensor range areas around the robots, respectively.

The overall performance of the CRE strategy in this experiment is summarized in Table 2. In this table, we present the number of visited nodes, distance traveled, and exploration time for each robot. We also present the mean value and total value of each measurement. These data represent the successful application of the CRE strategy in real time with an efficient backtracking method for a team of robots.

7 Conclusion

In this paper, we proposed a CRE strategy that is based on the FB-SRT-star method to explore an unknown environment efficiently using multiple robots. We defined hub nodes and proposed an efficient backtracking method. Numerical simulations were performed to identify the efficiency of the proposed algorithm. The simulation results demonstrate that the number of visited nodes and the distance traveled per robot was reduced using the CRE strategy. This indicated that the proposed algorithm improves the efficiency of the existing FB-SRT strategy for multi-robot exploration. We also performed experiments and applied the proposed strategy in real time. The proposed strategy can be implemented effectively and efficiently by multiple robots for the exploration of unknown environments.

Acknowledgment

The authors would like to thank all ICSL members (Seoul National University, South Korea) for their support and assistance with obtaining the experimental results.

References

[1] Palmieri, L., and Arras, K. O., 2014, "A Novel Rrt Extend Function for Efficient and Smooth Mobile Robot Motion Planning," IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Chicago, IL, Sept. 14–18, pp. 205–211.

[2] Bennet, D. J., and McInnes, C. R., 2010, "Distributed Control of Multi-Robot Systems Using Bifurcating Potential Fields," Rob. Auton. Syst., **58**(3), pp. 256–264.

[3] Heidari, F., and Fotouhi, R., 2015, "A Human-Inspired Method for Point-to-Point and Path-Following Navigation of Mobile Robots," ASME J. Mech. Rob., **7**(4), p. 041025.

[4] Menon, M. S., Ravi, V., and Ghosal, A., 2017, "Trajectory Planning and Obstacle Avoidance for Hyper-Redundant Serial Robots," ASME J. Mech. Rob., **9**(4), p. 041010.

[5] Latombe, J.-C., 2012, *Robot Motion Planning*, Kluwer Academic Publishers, Norwell, MA.

[6] Koren, Y., and Borenstein, J., 1991, "Potential Field Methods and Their Inherent Limitations for Mobile Robot Navigation," IEEE International Conference on Robotics and Automation (ICRA), Sacramento, CA, Apr. 9–11, pp. 1398–1404.

[7] Choset, H., and Burdick, J., 2000, "Sensor-Based Exploration: The Hierarchical Generalized Voronoi Graph," Int. J. Rob. Res., **19**(2), pp. 96–125.

[8] Hart, P. E., Nilsson, N. J., and Raphael, B., 1968, "A Formal Basis for the Heuristic Determination of Minimum Cost Paths," IEEE Trans. Syst. Sci. Cybern., **4**(2), pp. 100–107.

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

[10] Yamauchi, B., 1997, "A Frontier-Based Approach for Autonomous Exploration," IEEE International Symposium on Computational Intelligence in Robotics and Automation (CIRA), Monterey, CA, July 10–11, pp. 146–151.

[11] Yamauchi, B., Schultz, A., and Adams, W., 1998, "Mobile Robot Exploration and Map-Building With Continuous Localization," IEEE International Conference on Robotics and Automation (ICRA), Leuven, Belgium, May 20, pp. 3715–3720.

[12] González-Banos, H. H., and Latombe, J.-C., 2002, "Navigation Strategies for Exploring Indoor Environments," Int. J. Rob. Res., **21**(10), pp. 829–848.

[13] Basilico, N., and Amigoni, F., 2011, "Exploration Strategies Based on Multi-Criteria Decision Making for Searching Environments in Rescue Operations," Auton. Rob., **31**(4), p. 401.

[14] Oriolo, G., Vendittelli, M., Freda, L., and Troso, G., 2004, "The Srt Method: Randomized Strategies for Exploration," IEEE International Conference on Robotics and Automation (ICRA), New Orleans, LA, Apr. 26–May. 1, pp. 4688–4694.

[15] LaValle, S. M., and Kuffner, J. J., Jr, 2000, "Rapidly-Exploring Random Trees: Progress and Prospects," New Directions in Algorithmic and Computational Robotics, A K Peters, Natick, MA, pp. 293–308.

[16] Albers, S., Kursawe, K., and Schuierer, S., 2002, "Exploring Unknown Environments With Obstacles," Algorithmica, **32**(1), pp. 123–143.

[17] Lee, D., and Recce, M., 1997, "Quantitative Evaluation of the Exploration Strategies of a Mobile Robot," Int. J. Rob. Res., **16**(4), pp. 413–447.

[18] Lumelsky, V. J., Mukhopadhyay, S., and Sun, K., 1990, "Dynamic Path Planning in Sensor-Based Terrain Acquisition," IEEE Trans. Rob. Autom., **6**(4), pp. 462–472.

[19] Kim, J., Seong, K. J., and Kim, H. J., 2012, "An Efficient Backtracking Strategy for Frontier Method in Sensor-Based Random Tree," 12th International Conference on Control, Automation and Systems (ICCAS), JeJu Island, South Korea, Oct. 17–21, pp. 970–974.

[20] Cao, Y. U., Fukunaga, A. S., and Kahng, A., 1997, "Cooperative Mobile Robotics: Antecedents and Directions," Auton. Rob., **4**(1), pp. 7–27.

[21] Fox, D., Burgard, W., Kruppa, H., and Thrun, S., 1999, "Collaborative Multi-Robot Localization," *Mustererkennung 1999*, Springer, Berlin, Germany, pp. 15–26.

[22] Chang, H. J., Lee, C. G., Hu, Y. C., and Lu, Y.-H., 2007, "Multi-Robot Slam With Topological/Metric Maps," IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), San Diego, CA, Oct. 29–Nov.2 pp. 1467–1472.

[23] Franchi, A., Freda, L., Oriolo, G., and Vendittelli, M., 2009, "The Sensor-Based Random Graph Method for Cooperative Robot Exploration," IEEE/ASME Trans. Mechatronics, **14**(2), pp. 163–175.