A Cooperative Exploration Strategy with Efficient Backtracking for Mobile Robots

Jinho Kim
Dept. of Mechanical Engineering
Univ. of Maryland, Baltimore County
Baltimore, MD 21250, USA
umbcjhkim@umbc.edu

Stephanie Bonadies
Northrop Grumman Corporation
Baltimore, MD 21240, USA
stephanie.bonadies@gmail.com

Andrew Lee
School of Engineering
University of Guelph
Guelph, Ontario, Canada, N1G 2W1
alee20@umbc.edu

S. Andrew Gadsden
School of Engineering
University of Guelph
Guelph, Ontario, Canada, N1G 2W1
gadsden@uoguelph.ca

***Abstract*—This paper proposes a cooperative robot exploration (CRE) strategy which is based on the sensor-based random tree (SRT) method. The proposed CRE strategy is for exploring unknown environments with a team of mobile robots equipped with range finder sensors. An existing backtracking technique for frontier-based exploration involves moving back through inefficient routes. To enhance the efficiency of the backtracking algorithm, a hub node is defined and the most direct backtracking route is generated using its frontier data. Numerical simulations demonstrate that the proposed strategy enables exploration of unknown environments by robots more efficiently than other common methods.**

***Index Terms*—frontier-based exploration, sensor-based random tree (SRT), cooperative robot exploration (CRE), efficient backtracking**

I. Introduction

Path planning is one of the interesting challenges in intelligent control of autonomous mobile robots. In the early era, almost path planning strategies assumed that mobile robots already know a map of their operating area and navigate using this information. For example, potential field [1], Voronoi daigram [2], A* algorithm [3] are map-based path planning methods for robots navigation.

However, these methods are not suitable for exploration in unknown environments, such as search and rescue missions in dangerous buildings, reconnaissance, and surveillance tasks. To perform the exploration in unknown environments without any prior information, most of the existing exploration method are based on the frontier-based exploration [4]. In this framework, a robot is located in an unknown environment and scans its surrounding using laser range scanners or sonar sensors. After obtaining scanning data, the robot extends its map by moving to the frontier, the boundary between explored space and unexplored space. To develop this method, numerous sensor-based exploration methods have been proposed [5]–[9]. In [5], [6], the Sensor-based Random Tree (SRT) method is proposed as one of the most effective exploration methods. This method constructs a data structure (SRT) which consists of a collision-free configuration which the robot has already explored, and the Local Safe Region (LSR) utilizing the sensory system. To improve efficiency of exploration using the SRT method, two types of shape for LSR are proposed; a ball shape and a star shape.

Due to the fact that the multiple robots exploration has advantages over the single robot case [10], [11], some research groups extended their strategies to cooperative robot exploration (CRE) based on frontier-based exploration strategy [12]–[14]. In [12], the extended version of frontier-based exploration is proposed for multiple robots based on [4]. In this approach, each robot can share its perceptual data except its own global map, and decides where to explore independently. The proposed approach enables robots to explore more effectively sharing their data, but it is also robust to the loss of individual robots at the same time. Developing [6], [14] proposed a decentralized cooperative exploration strategy with a sensor-based random graph (SRG), which is a data structure of the explored area with the associate LSR. As robots explores and builds the SRG, bridges are created by the SRG manager to enhance the connectivity of the SRG. In those works, however, a robot cannot move back directly to a position which has a frontier when a robot reaches a position which has no frontier. Instead, each robot will take a long detour route to backtrack to continue exploring, decreasing the efficiency of exploration when using those strategies.

In this paper, we propose an backracking algorithm for a team of robots based on the previous research [15]. This allows a team of robots to backtrack along the most direct route to continue exploring or return to their initial points, rather than using an indirect detour rout. With the proposed backtracking algorithm, the CRE strategy enables a team of robots to explore unknown environments efficiently. This is the objective of our work which is described in this paper.

This paper is organized as follows. In section II, the assumptions are listed and the robot kinematics model is introduced. Then, SRT method which is the basic concept of this paper is presented in section III. In section IV, the proposed CRE strategy is presented with the efficient backtracking algorithm. Finally, the results of numerical simulations and conclusion are presented in section V and VI.

II. Problem Setting

In order to apply the proposed CRE strategy, we need to clarify characters of the workspace and robot used in this paper. In this section, we first state the some assumptions. Then, we describe robot kinematics.

A. Assumptions

The following assumptions are used to develop the proposed CRE strategy.

1) The workspace $\mathbb{W}$ is an n-dimensional Euclidean space $\mathbb{R}^n$, where n = 2 or 3. In this paper, we consider only case n = 2.
2) The shape of robot is circular and it is free to move in any direction. See robot kinematics in section II-B.
3) The robot knows its configuration q containing its position and heading angle.
4) The robot has sensory equipment, and this provides the robot information of the surrounding area within sensor range R_s. This information is called a $Local\ Safe\ Region$ (LSR) and denoted by $\mathcal{S}$.
5) From the data of $\mathcal{S}$, the robot can calculate the frontier and then saves it in the data structure called Frontier Data (FD). It is denoted by $\mathcal{F}$.
6) Each robot can share its q and $\mathcal{F}$ within a communication range R_c in real time. In this paper, we consider R_c is large enough to cover all the $\mathbb{W}$.

The algorithm presented in this paper is perfomed under these assumptions to simplify the problem. A robot is equipped with a laser range scanner as sensory equipment to perceive obstacles, and the plane of a laser range scanner is a subset of $\mathbb{R}^2$. In this paper, we consider obstacles are not below the laser plane. Furthermore, a robot dynamics plane is also a subset of $\mathbb{R}^2$. Hence, $\mathbb{R}^2$ is considered as a workspace in this paper. The assumption 2 implies that the robot moves in any direction using a turn-and-go scheme. In assumption 4, the SRT method, the basic concept of the proposed algorithm, is divided into two categories according to the shape of $\mathcal{S}$. In this paper, the SRT-Star method is used, because this method is more efficient than the SRT-ball method [5]. Assumption 5 is taken for the proposed CRE strategy based on frontier method using SRT presented in section III. Under these assumptions, the CRE of unknown environments for a team of robots, is explained in this paper. A more specific explanation will be presented in the following sections.

B. Robot kinematics

In order to make a robot move in any direction, the turn-and-go scheme is applied to the robot. A brief explanation of the robot kinematics is presented in this subsection.

A k-th robot (k = 1, 2, $\cdots$, n) with radius of r has a position (x_k, y_k) $\in \mathbb{W}$ and a heading angle, θ_k, as shown in Fig. 1.

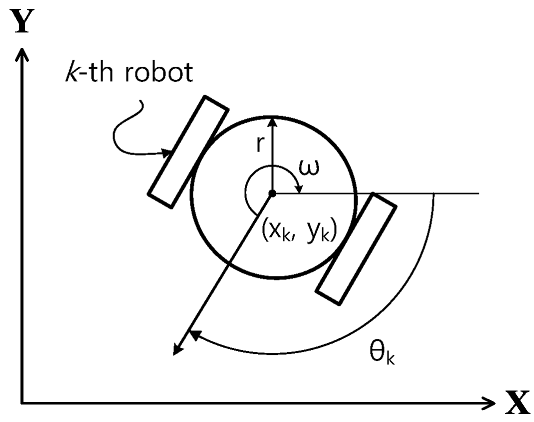

Fig. 1. Robot kinematics.

Each robot has two wheels and is equipped with a laser range scanners to measure its surrounding. 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, y_k) is a position of k-th robot, and θ_k is a heading angle of k-th robot. In 1-3, the velocity and angular velocity of k-th robot, V_c and ω_c, can be written as

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

$$\omega_c = \frac{V_r + V_l}{D}, \tag{5}$$

where D is robot diameter and V_l, V_r is left and right wheel speed, respectively.

III. Sensor-based Random Tree Exploration

In this section, we introduce the SRT method to understand the basic concept used in this paper.

The workspace used in the SRT method is divided into the explored and unexplored regions as depicted in Fig. 2. Since obstacles are not located in the explored region, a robot can move everywhere in this region, thus we call it the $safe\ region$. The explored region can also be divided into three categories. If there is boundary between obstacles and explored region, it is referred to as an $obstacle\ arc$. If there is boundary between the explored region and an unexplored region, it is referred to as a $frontier\ arc$. However, when some configuration covers the frontier arc of another configuration, this frontier arc turns into a $free\ arc$.

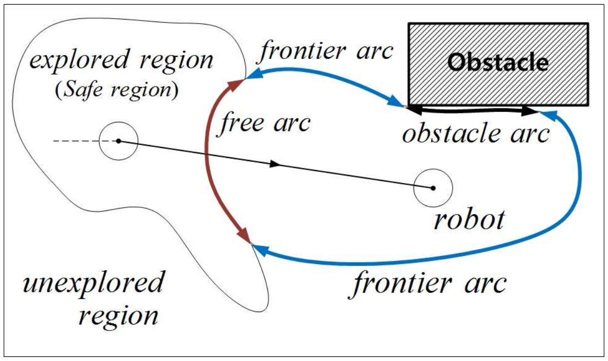

Fig. 2. The categorized workspace in SRT method.

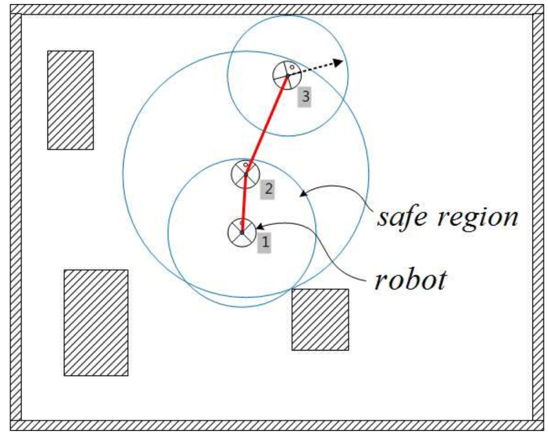

Fig. 3. The shape of Local Safe Region $\mathcal{S}$ using the SRT-Ball method.

A. SRT method

A pseudocode description of the SRT method is shown in Algorithm 1. As inputs 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 surroundings of a robot. Then, the algorithm collects the free space data $\mathcal{S}$ from the surrounding area, and frontier, $\mathcal{F}$, through the $OBSTACLE\_SCAN$ and $FRONTIER\_SCAN$ functions, respectively. And these data are stored in the SRT data structure. From obtained $\mathcal{F}$, the algorithm generates a set of heading angle candidates, θ_{cand}, to find the next target position. Then, a random direction, θ_{rand}, is randomly selected and the radius r of $\mathcal{S}$ along θ_{rand} is computed. Finally, the algorithm obtained the next target position, q_{cand}, by taking a step movement, α, multiplied by r along θ_{rand}. When the next target position 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 $\mathcal{F}$, a robot moves to a point $q_{cand,parent}$ which has a frontier arc through q_{prev}. In this algorithm, α must be less than or equal to one. A smaller value of α results in a larger safety margin.

Algorithm 1 SRT_EXPLORATION

```
Require : q_init, K_max, I_max, α, d_min
q_curr = q_init
for k = 1 to K_max do
    S(q_curr) ⇐ OBSTACLE_SCAN(q_curr)
    F(q_curr) ⇐ FRONTIER_SCAN(q_curr, S(q_curr))
    T ⇐ SAVE_T(q_curr, S(q_curr), F(q_curr))
    θ_cand(q_curr) ⇐ THETA_CAND(F(q_curr))
    i = 1
    repeat
        θ_rand ⇐ RAND_DIR
        r ⇐ RANGE(S(q_curr), θ_rand)
        q_cand ⇐ Q_CAND(q, θ_rand, α · r)
        i = i + 1
    until (CHECK(q_cand, d_min, T) or i = I_max)
    if (CHECK(q_cand, d_min, T) then
        MOVE_TO(q_cand)
        q_curr ⇐ q_cand
    else
        MOVE_TO(q_cand,parent)
        q_curr ⇐ q_cand,parent
    end if
end for
return q_curr
```

B. Shape of LSR: SRT-Ball and SRT-Star

The SRT method is divided into two categories according to the shape of $\mathcal{S}$. First, the shape of $\mathcal{S}$ can be defined as a ball whose radius is r as depicted in Fig. 3. This is the SRT-Ball method. The r of the SRT-Ball method is determined by the distance between the robot and the closest obstacle. Since the $\mathcal{S}$ of the SRT-Ball method has the same r in all directions despite the possibility that there is a greater distance in the direction of frontier, it could cause a decrease in efficiency of exploration. On the other hand, the SRT-Star method has a unique radius of $\mathcal{S}$ for each direction depending on the surrounding environment. Using the SRT-Star method, the

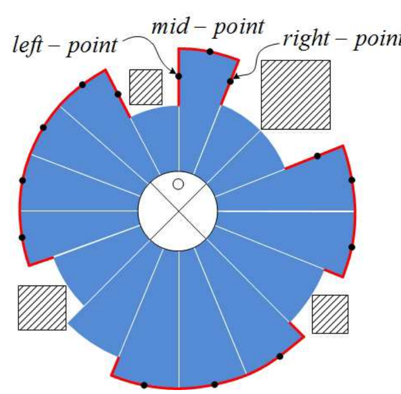

Fig. 4. The definition of frontier arcs with mid, $left$, $right$ $point$ in the SRT-Star method.

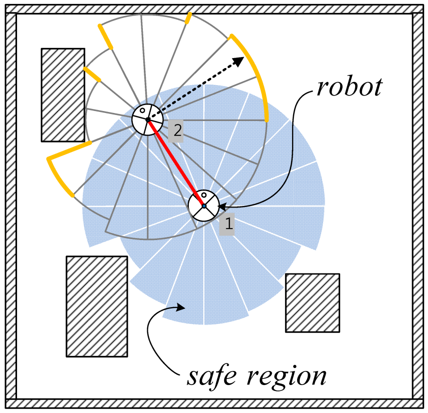

Fig. 5. An example of the SRT-Star method exploration.

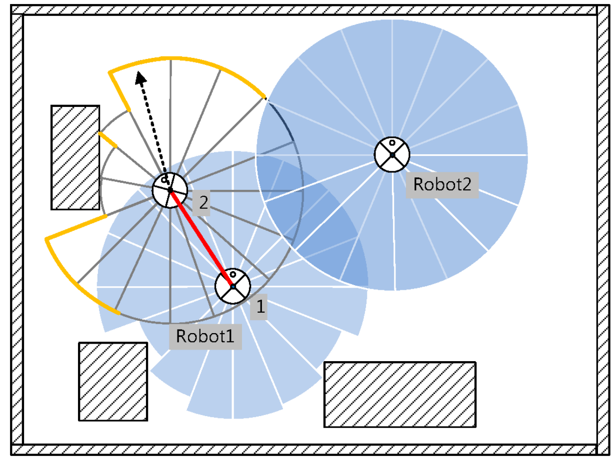

Fig. 6. An example of the CRE exploration.

robot can move a greater distance in a particular direction. Based on this added flexibility, the SRT-Star method can allow for more efficient movements than the SRT-Ball method [5]. As such, we will consider only the SRT-Star method in this paper.

To understand the SRT-Star method, it is necessary to explain its shape of LSR. $\mathcal{S}$ of the SRT-Star method is a star-shaped region which consists of several cone-shaped regions as shown in Fig. 4. The SRT-Star method divides $\mathcal{S}$ into cone-shaped sectors in which three points can be defined. The $mid\text{–}point$, 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, $right\text{–}point$ and $left\text{–}point$, can be defined as frontier points when there is a long gap between adjacent sectors. In Fig. 4, the thick outer line (red) of $\mathcal{S}$ represents frontier arcs while the rest of portions represent either free arcs or obstacle arcs. An example of the SRT-Star exploration is illustrated in Fig. 5. In this figure, a robot at position 2 has frontier arcs which are colored yellow (thick outer lines).

IV. SRT-Star method for CRE

The CRE strategy is based on the concept of the SRT-Star method which is presented in the previous section. In order to apply the SRT-Star method to a team of robots, it is necessary to revise the SRT-Star method. At first, we assumed that robots can share their SRT data with each other as shown in II. So each robot rebuild their own SRT data considering others' SRT data. In the CRE strategy, the data tree of each q consist of configuration data, $\mathcal{Q}$, and frontier data, $\mathcal{F}$. In Fig. 6, an example of $\mathcal{F}$ in the CRE strategy is represented. As the *Robot1* moves to q_2 from q_1, $\mathcal{F}$ of q_2 is shown as thick outer lines (yellow). The *Robot2*, however, is close to the *Robot1* at point 2, such that the frontier arcs of q_2 is overlapped by the frontier arcs of the *Robot2*. Hence, the CRE algorithm updates their $\mathcal{F}$ combining information from *Robot1* and *Robot2*, and each robot will move to the frontier direction using the CRE method.

Mathematically, $\mathcal{F}$ of q_{ij} can be written as follows;

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

where i is the robot number, j is the node number and k is the frontier sector number. To perform an efficient backtracking algorithm, we define q_{ij} as a hub node q_{hub} when maximum of the set of $\mathcal{F}_{q_{ij}}$ is 1. We present a pseudocode of the CRE algorithm first, then explain the backtracking algorithm.

A. CRE algorithm

The pseudocode of the CRE algorithm is described in Algorithm 2. To execute the CRE algorithm, it requires a

Algorithm 2 CRE

Require : $\mathcal{Q}, \mathcal{F}, q_{curr}, N_{robot}$
for i = 1 to N_{robot} **do**
  $q_{ij} = q_{curr}(i)$
  $MODIFIED\_SRT\_EXPLORATION$
  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**
        $q_{target}(i) \Leftarrow BACKTRACK(\mathcal{H}(i))$
        break;
      end for
    else
      $\mathcal{H}(m) \Leftarrow FIND\_HUB\_MAX(i)$
      if $\mathcal{H}(m) \neq \phi$ **then**
        $q_{target}(i) \Leftarrow TRANSFER(\mathcal{H}(m))$
      else
        $q_{target}(i) \Leftarrow HOMING(q_{init})$
      end if
    end if
  end if
end for

Algorithm 3 TRANSFER

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

configuration data tree, $\mathcal{Q}$, a frontier data tree, $\mathcal{F}$, a current position of robots, q_{curr}, and the number of robots, N_{robot}, as inputs. In the first step, $robot(i)$ performs the modified $SRT\_EXPLORATION$ function using the $\mathcal{Q}(q_{ij})$ and $\mathcal{F}(q_{ij})$. In order to perform an efficient backtracking algorithm, the backtracking part is eliminated from the $SRT\_EXPLORATION$ function, and it is used in the CRE algorithm. During the exploration, if $\mathcal{F}$ at current position, q_{ij}, is empty, it tries to find the nearest hub node, $\mathcal{H}(i)$, of its own $\mathcal{Q}$ through the $FIND\_HUB$ function. Then, the robot moves to $\mathcal{H}(i)$ using the $BACKTRACK$ function. If there is no $\mathcal{H}$ in its own $\mathcal{Q}$, it will determine $\mathcal{H}(m)$ from the another robot's $\mathcal{Q}$ which has most using the $FIND\_HUB\_MAX$ function, and the robot will move to $\mathcal{H}(m)$ using the $TRANSFER$ function and $BACKTRACK$ function. When none of the robots have $\mathcal{H}$, it indicates that the entire workspace has been explored, such that the robots move back to their initial positions using the $HOMING$ function. This is the CRE algorithm based on the SRT algorithm.

B. TRANSFER Algorithm

The pseudocoed of $TRANSFER$ algorithm is shown in Algorithm 3. When a robot has no q_{hub} in its $\mathcal{H}$, it determines a q_{hub} in another robot's $\mathcal{H}$ which has the most hub nodes. Then, this q_{hub} of the m-th tree, $\mathcal{H}(m)$, is an input of the $TRANSFER$ algorithm. Also, the $TRANSFER$ algorithm requires a robot number, i. If $\mathcal{H}(m)$ is not an empty set, the i-th robot will move to the its initial position, q_{i1}, using the $BACKTRACK$ function at first. Then, it moves to q_{m1}, which is an initial point of the m-th tree, when a robot reaches q_{i1}. Finally, a robot moves to the nearest q_{hub} of $\mathcal{H}(m)$ through the $BACKTRACK$ function. Since a robot explores transferring from its own data tree to another robot's data tree using the $TRANSFER$ algorithm, the proposed strategy enable a team of robots to explore an unknown environment cooperatively.

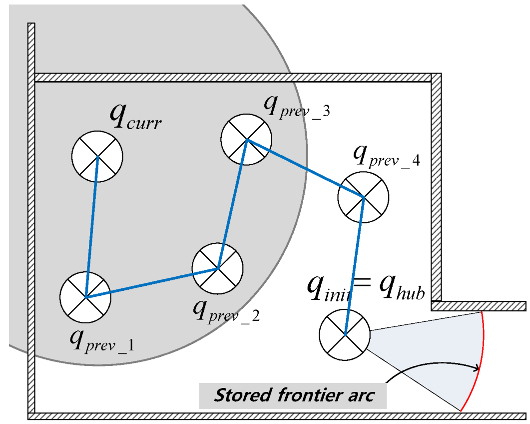

Fig. 7. The exploration of FB-SRT-Star method.

C. BACKTRACKING algorithm

When there are no frontier arcs at q_{curr} during the exploration under the SRT method, the robot backtracks to the q_{hub} along the previous positions. However, as the robot backtracks through its previous positions, it may not generate the shortest route between q_{curr} and q_{hub}. In order to solve this problem, we propose a backtracking algorithm in this section. Unlike the backtracking of the SRT method, the proposed backtracking algorithm enable a robot to reach q_{hub} more efficiently with $\mathcal{Q}$ and $\mathcal{F}$.

Imagine that a robot explores from q_{init} to q_{curr} following the line depicted in Fig. 7. In this example, q_{init} has a frontier arc and can be determined as q_{hub}. Since there is no frontier arc at q_{curr}, the robot moves back to q_{prev_1} and q_{prev_2}, q_{prev_3}, q_{prev_4}, and arrive at q_{hub} finally. Then, the robot continues exploring in the direction of the remaining frontier arc. This is clearly an inefficient backtracking route in this example. Therefore, we remedy this shortcoming of backtracking, such that a robot can move back almost directly

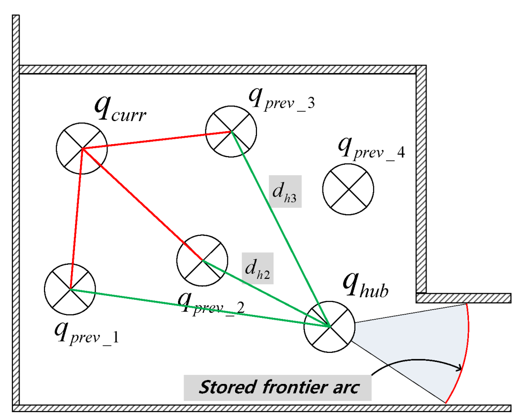

Fig. 8. Designation of q_{back_cand} and calculate the distance d_{hi}, between q_{hub} and $q_{back\_cand}(i)$).

to q_{hub}.

In order to generate the most efficient path, previous nodes, q_{prev_i} $(i = 1, 2...j)$, between q_{hub} and q_{curr}, are selected as a candidates of q_{back}. Only q_{prev_i} within a certain range, d_c, from q_{curr} can be included to the candidate set, q_{back_cand}. In this example, q_{prev_1}, q_{prev_2}, and q_{prev_3} can be elements of q_{back_cand} at q_{curr}.

Next, the distance, d_{hi}, between q_{hub} and $q_{back\_cand}(i)$ is calculated as presented in Fig. 8. Then, the elements of q_{back_cand} are sorted in order of lowest d_{hi}, and the frontier directions of q_{curr} and each of the candidates of q_{back} are validated. In this example, q_{back_cand} can be written as q_{back_cand} = $[q_{prev\_2}, q_{prev\_1}, q_{prev\_3}]$. In this case, the backtracking algorithm choose q_{prev_2} for q_{back} because a robot confirms that the frontiers of q_{curr} and q_{prev_2} are free to each others directions through their $\mathcal{F}$. If the arc between these two positions is obstructed, then calculations of the backtracking algorithm will iterate to find the next q_{back_cand} until q_{back} is determined.

Finally, all of the previous steps are iterated until the robot reaches q_{hub}, and then the robot will continue exploring unexplored regions. By defining q_{hub}, the backtracking algorithm generates an efficient path to the team of robots using $\mathcal{Q}$ and $\mathcal{F}$ of each node.

V. Simulations

We present the simulation results to confirm the efficiency of the CRE algorithm including the proposed algorithms in this paper. Each robot carries a 360° laser range scanner with a maximum scanning range of 20 units. The number of sectors, I_{max}, is set to 18 (20 degree interval) for the simulations.

Explorations were performed with a varying number of robots, ranging from one to four. The performance index is evaluated in terms of the *number of visited nodes per robot* and the *distance of movement per robot*.

In the simulation, the workspace consists of a hallway and three rooms which can be entered through only one entrance. The size of the workspace is 100 by 180, as shown in Fig. 9. This figure depicts the progress of the CRE strategy with three robots. The black lines are walls, the large colored circles are robots and the red dotted circles around the robots are the sensor range areas. The small black dots are positions that the robot has already passed by and these positions can be referred to as nodes. The colored lines represent the trajectory of the robots. When the sensor range areas cover the entire workspace, it indicates that the exploration of the workspace has been successfully completed. Simulations have been stopped if all the robots return to their initial position, indicating that there are no unexplored regions remaining, as seen in the last frame of Fig. 9.

The resulting number of visited nodes per robot and distance of movement per robot demonstrate how the CRE strategy is more efficient than the SRT method, as compared in Fig. 10. The distance of movement per robot of the CRE strategy is always smaller than that of SRT method simulation. As the numerical simulation results show, when the CRE strategy is performed, the number of visited nodes per robot decreases by 32.3%-43.4% compared to the cases that utilize the SRT method. Moreover, the distance of movement per robot decreases by 10.4%-18.9% when using the CRE strategy compared to the SRT method.

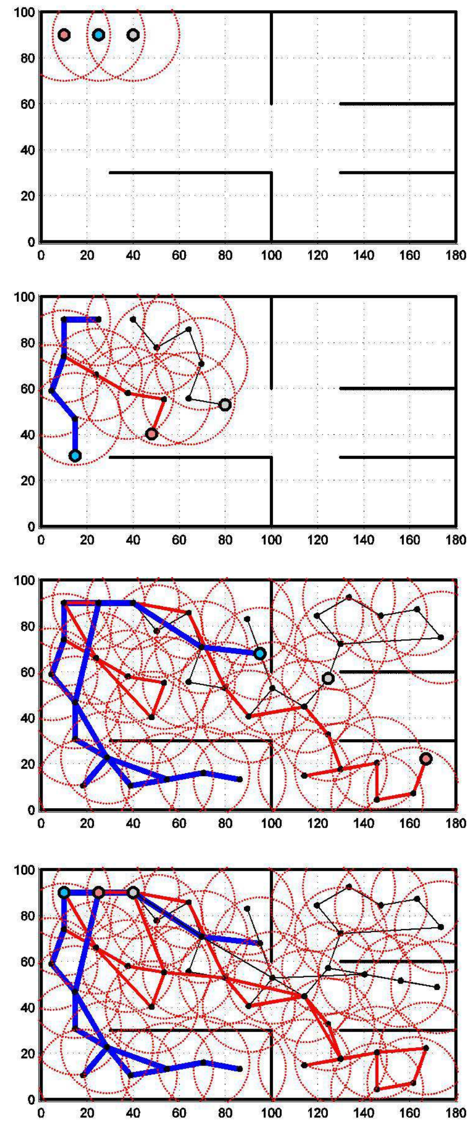

Fig. 9. Progress of the cooperative robot exploration. The black lines are walls, the big colored circles are robots and the red dotted circles around the robots is the sensor range areas, the small black dots are positions that the robot already passed by.

VI. Conclusion and Future Work

This paper describes a cooperative robot exploration (CRE) strategy based on the SRT algorithm. In order to explore an unknown environment efficiently using multiple robots, hub nodes are defined and an efficient backtracking method was proposed for the CRE strategy. Numerical simulations were performed to validate the efficiency of the proposed strategy. The simulation results demonstrate that the number of visited

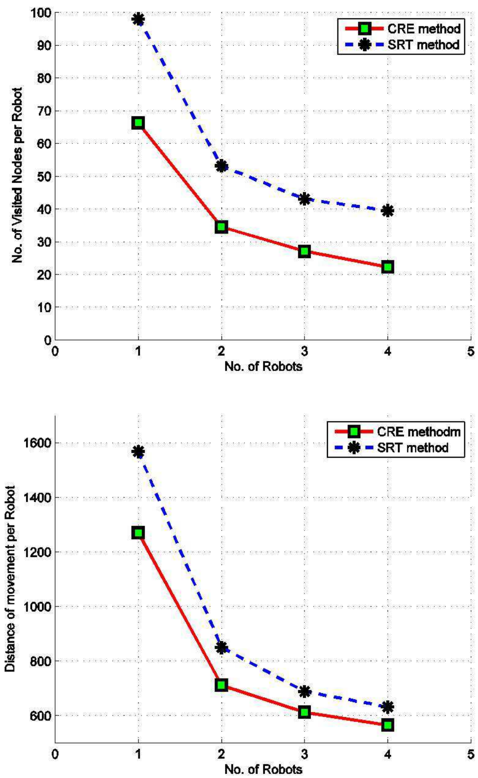

Fig. 10. The number of visited nodes per robot (above) and distance of movement per robot (below). Squares and asterisks are results of CRE algorithm and STR algorithm, respectively.

nodes and the distance of movement per robot was reduced. This indicates that the proposed strategy is more efficient. In the future, we will apply the CRE strategy to experimental systems. Furthermore, we will develop the CRE strategy for a team of UAVs to explore unknown environments more efficiently in terms of time and energy.

References

[1] H. Rezaee and F. Abdollahi, "A decentralized cooperative control scheme with obstacle avoidance for a team of mobile robots," *IEEE Transactions on Industrial Electronics*, vol. 61, no. 1, pp. 347–354, 2014.

[2] H. Choset and J. Burdick, "Sensor-based exploration: The hierarchical generalized voronoi graph," *The International Journal of Robotics Research*, vol. 19, no. 2, pp. 96–125, 2000.

[3] J.-C. Latombe, *Robot motion planning*. Springer Science & Business Media, 2012, vol. 124.

[4] B. Yamauchi, "A frontier-based approach for autonomous exploration," in *Computational Intelligence in Robotics and Automation, 1997. CIRA'97., Proceedings., 1997 IEEE International Symposium on*. IEEE, 1997, pp. 146–151.

[5] G. Oriolo, M. Vendittelli, L. Freda, and G. Troso, "The srt method: Randomized strategies for exploration," in *Robotics and Automation, 2004. Proceedings. ICRA'04. 2004 IEEE International Conference on*, vol. 5. IEEE, 2004, pp. 4688–4694.

[6] L. Freda and G. Oriolo, "Frontier-based probabilistic strategies for sensor-based exploration," in *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on*. IEEE, 2005, pp. 3881–3887.

[7] D. Holz, N. Basilico, F. Amigoni, and S. Behnke, "Evaluating the efficiency of frontier-based exploration strategies," in *Robotics (ISR), 2010 41st International Symposium on and 2010 6th German Conference on Robotics (ROBOTIK)*. VDE, 2010, pp. 1–8.

[8] S. Albers, K. Kursawe, and S. Schuierer, "Exploring unknown environments with obstacles," *Algorithmica*, vol. 32, no. 1, pp. 123–143, 2002.

[9] A. AlDahak, L. Seneviratne, and J. Dias, "Frontier-based exploration for unknown environments using incremental triangulation," in *Safety, Security, and Rescue Robotics (SSRR), 2013 IEEE International Symposium on*. IEEE, 2013, pp. 1–6.

[10] G. Dudek, M. R. Jenkin, E. Milios, and D. Wilkes, "A taxonomy for multi-agent robotics," *Autonomous Robots*, vol. 3, no. 4, pp. 375–397, 1996.

[11] Y. U. Cao, A. S. Fukunaga, and A. Kahng, "Cooperative mobile robotics: Antecedents and directions," *Autonomous robots*, vol. 4, no. 1, pp. 7–27, 1997.

[12] B. Yamauchi, "Frontier-based exploration using multiple robots," in *Proceedings of the second international conference on Autonomous agents*. ACM, 1998, pp. 47–53.

[13] W. Burgard, M. Moors, D. Fox, R. Simmons, and S. Thrun, "Collaborative multi-robot exploration," in *Robotics and Automation, 2000. Proceedings. ICRA'00. IEEE International Conference on*, vol. 1. IEEE, 2000, pp. 476–481.

[14] A. Franchi, L. Freda, G. Oriolo, and M. Vendittelli, "The sensor-based random graph method for cooperative robot exploration," *IEEE/ASME Transactions on Mechatronics*, vol. 14, no. 2, pp. 163–175, 2009.

[15] J. Kim, K. J. Seong, and H. J. Kim, "An efficient backtracking strategy for frontier method in sensor-based random tree," in *Control, Automation and Systems (ICCAS), 2012 12th International Conference on*. IEEE, 2012, pp. 970–974.