The 2010 IEEE/RSJ International Conference on
Intelligent Robots and Systems
October 18-22, 2010, Taipei, Taiwan

A Cooperative Approach for Multi-Robot Area Exploration

Jing Yuan, Yalou Huang, Tong Tao and Fengchi Sun

***Abstract*—A cooperation approach with consideration of communication limit is proposed for multi-robot area exploration, in which all the robots select local destinations satisfying the constraints on communication range and reach their destinations at the same time to communicate and fuse their map information. Firstly, the robots compute the frontier between the explored region and the unexplored one. The robots choose the optimal frontier points, which maximize information gain, minimize navigation cost and satisfy communication limit as their local destinations. Then the problem of global exploration in unknown environment is converted into that of multi-stage trajectory planning in local known environment. Collision-free, synchronous and separate trajectories are planned for all the robots to realize the limited communication at their destinations. In such a way, efficient and distributed exploration can be achieved. Simulation results are presented to show the effectiveness of our method.**

I. Introduction

Efficient exploration of unknown environments is a fundamental problem in mobile robotics. As autonomous exploration and map building becomes increasingly successful on single robot, more challenge is to extend these techniques to multi-robot cooperation. Multi-robot teams have the potential to accomplish the exploration task faster than a single robot. However, compared to the problems occurring in single robot exploration, the extension to multiple robots poses two new issues, including (1) coordination of multiple robots and (2) dealing with the constraints on communication range.

For multi-robot coordination, efficiency increasing is one of the key reasons for distribution multiple robots in environment instead of single robot. The more robots that detect an unknown environment, the more important the coordination among them becomes. Yamauchi demonstrated that frontier-based single-robot exploration can be used to build occupancy grids that represent the spatial structure of the environments [1]. And the above strategy has also been extended to deal with multi-robot cooperative area exploration by Yamauchi[2], in which each robot moves to the closest frontier in the current map. However, there is no coordination component, which chooses different frontier cells for individual robots. Frontier-based exploration has been adopted by in many literatures [3, 4] to assign multiple robots to different exploration frontiers, which guide the robots to detect non-overlapping areas of the environment with consideration of the exploration cost and expected visibility range. In [5], a decision-theoretic approach to multi-robot exploration in structured environment was presented, in which the global uncertainty about the robot's relative locations is considered and each robot is assigned to a local destination by calculating a trade-off between the expected utility and the expected cost. In [6], heuristic exploration was proposed based on the concepts of entropy and Yamauchi's frontier. Entropy is used to quantify the information gain obtained during the exploration process, guiding the robots move toward the areas where information is less certain. Zlot presented a new approach for coordination using market- based approach [7]. The market architecture seeks to maximize benefit (information gained) while minimize cost (in terms of the collective travel distance). In Zlot's market economy strategy, the multi-robot system does not rely on perfect communication, it can still be carried out if some of the colony members lose communications.

For multi-robot team with constraints on communication range, coordination under limited communication situation must be taken into account. In [8], another frontier-based, distributed bidding model for coordination of multiple robots was proposed. The limited communication is considered in two ways. Firstly, robots are guided to stay close to each other by considering the distances between robots. Additionally, a new coding mechanism is developed for map representation. New coding mechanism reduced the exchanged data volume. Rendezvous strategy was introduced to deal with the limited communication, such as [5], [9] and [10]. In [11], the multi-robot routing problem under communication constraints was addressed. The robot team forms a connected mobile ad-hoc network and the connectivity remains intact (even through relaying) during the entire mission.

In this paper, the problem of multi-robot cooperative exploration with limited communication is addressed. Multiple robots select the candidate local destinations from the frontiers, and calculate the optimal ones as their destinations, satisfying the constraints on communication range. Then effective motion planning scheme is designed such that all the robots can reach their individual destinations at the same time to communicate and share information with each other. In such a way, the problem of exploration in global unknown environment can be converted into that of multi-stage motion planning in local known environment. Therefore, many existing methods for motion planning in known environment can be used.

The remainder of this paper is organized as follows: the

Manuscript received January 24, 2010. This work was supported in part by NSF of China under Grant 60805031, NSF of Tianjin under Grant 10JCYBJC07600 and the Ph.D. Programs Foundation of Ministry of Education of China under Grant 200800551015.

J. Yuan is with the Department of Automation, Nankai University, 300071 Tianjin, China, nkyuanjing@gmail.com

Y. Huang, T. Tao and F. Sun are with the College of Software, Nankai University, 300071 Tianjin, China, {Huangyl, taotong2006, fengchisun} @nankai.edu.cn

problem of multi-robot cooperative area exploration is described in section II and the calculation of local destinations for all the robots is presented in section III. In section IV, the trajectory planning is dealt with. Simulations are made to verify our method in section V.

II. PROBLEM STATEMENT

Given 2-D unknown environment W in presence of unknown obstacles. For simplicity, we restrict our analysis to circular robots and obstacles. This is not a severe assumption since general polygons can be represented by a number of circles. A circular obstacle can be denoted with its center and radius, i.e. $OBS_j = (x_j, y_j, r_j)\ (j = 1, 2\cdots)$, where (x_j, y_j) and r_j represent the center and the radius respectively and they are unknown.

Given a robot team consisting of M differential-driven mobile robots $R_i = (x_{r_i}, y_{r_i}, \theta_{r_i})\ (i = 1, 2\cdots M)$. Assume that the on-board sensor of the robot R_i can cover a circular region denoted as Sen_i, with center (x_{r_i}, y_{r_i}) and radius S_i. The maximum communication range of R_i is C_i, i.e. the accessible communication region is covered by a circle denoted as Com_i, with center (x_{r_i}, y_{r_i}) and radius C_i. In general, C_i is greater than S_i.

Given that the initial poses of the robots are known and the initial relative locations among them satisfy the constraints on communication range, i.e. at the initial locations, each of them can communicate with the others. Then, the problem of multi-robot cooperative exploration can be stated as: the robot team composed of M homogeneous members detect the unknown environment W with their on-board sensors together, such that the free space can be covered by robot sensors, i.e. $W - \bigcup_j O_j \subseteq \bigcup_{i,k} Sen_i(k)$, simultaneously the robots can detect the space occupied by obstacles. Where $O_j = \{(x, y) \mid (x - x_j)^2 + (y - y_j)^2 \le r_j^2\}$ denotes the region occupied by the obstacle OBS_j, and k denotes the discrete time step.

III. LOCAL DESTINATIONS GENERATION

We start by describing evidence grid [12] as the spatial representation. After an evidence grid has been constructed, each cell in the grid is classified as:

Open: the cell is free;

Unknown: the cell has not been detected by the sensor;

Occupied: the cell has been occupied by the obstacle.

As defined in [1], *frontiers* are regions on the boundary between open space and unknown space. When a robot moves to a frontier, it can see into unexplored space and add the new information to its map. As a result, the mapped territory expands, pushing back the boundary between the known and the unknown. By moving to successive frontiers, the robot can constantly increase its knowledge of the world. In the frontier-based exploration, when the robot reaches its destinations, that location is added to the list of previously visited frontiers. The robot performs a 360-degree sensor sweep and adds the new detected cells to the explored region. Then the robot detects frontiers present in the updated grid and attempts to move to the nearest accessible, unvisited frontier. This strategy has been extended to multi-robot exploration in [2]. However, if each robot only navigates to its nearest frontier cell without coordination with the others, some other factors, such as information gain and communication constraint can not be dealt with. In our paper, optimal frontiers will be selected for robot team by evaluating information gain and navigation cost, simultaneously considering the constraint on communication range.

A. Candidate destination cells

The multi-robot team calculates the set of frontier cells, denoted as F, and then select M candidate destination cells from F such that evaluation for these cells will be optimal. The evaluating indexes include predictive information gain $G_{i,f}$ and predictive navigation cost $L_{i,f}$. $G_{i,f}$ is determined with the area of the new region that would be detected by the robot R_i if R_i were located at a frontier cell $f \in F$ in current map. $L_{i,f}$ is defined as the predictive length of the path from the current position of R_i to f and $L_{i,f}$ can be calculated with Bug2 algorithm [13] [14].

The robots successively explore the unknown space, thus the number of the frontier cells could increase quickly. The computation complexity to evaluate all of the frontier cells will become unacceptable. While, each frontier cell usually has the similar characteristics with its neighbors, i.e. the evaluating indexes of the adjacent frontier cells are approximately identical. Hence it's a better way to partition the frontier cells into different groups and evaluate each group as a whole.

Although there are a lot of methods that can be applied to partition the frontier cells, the clustering algorithm can provide a natural and flexible way. By means of the clustering algorithm, the frontier cells between explored and unexplored regions will be grouped into several clusters in terms of their internal properties. The cell nearest to the geometric center of the frontier cells in each cluster is selected as a candidate destination cell.

K-means is one of the most popular clustering algorithms. It will produce the clustering result which minimizes the distance metric within classes. Although it has been proved that k-means clustering will always terminate, it does not necessarily find the most optimal solution, because the number of groups and the initial centers can not be determined easily beforehand. If the number of clusters can't be given ahead or the initial values are not proper, the conventional k-means clustering algorithm may cause incorrect result. To solve these problems, the subtractive clustering algorithm [15] will be introduced to estimate the initial centers and the number of groups. In subtractive clustering, a few data points are selected as the potential centers according to the density measurement, which can provide initial values for conventional k-means clustering algorithm, although the actual centers are not always at the data points, in most cases, it can provide a good approximation.

To make the adjacent frontier cells fall into the same group, the two dimension Cartesian coordinates that represent the position of the cell center should be taken into account as the first two properties. $G_{i,f}$ and $L_{i,f}$ are selected as the third and fourth one. Then the clustering feature can be chosen as $P(k)=[x_f(k), y_f(k), G_{i,f}(k), L_{i,f}(k)]^T$, where $[x_f(k), y_f(k)]^T$ is the position of $f(k)$ at current time step k. Assume that the robot is located at $(x_{r_i}(k), y_{r_i}(k))$, then the predictive information gain is calculated as:

$$G_{i,f}(k)=\frac{\min\{Dis(f(k),OBS_{known}),S_i\}}{S_i}\cdot[Sen_{i,f}(k)\cap Region_{unknown}(k)] \quad (1)$$

Where $Dis(f, OBS_{known})$ represents the Euclidean distance from $f(k)$ to the detected obstacles; $Sen_{i,f}(k)$ is the region that could be detected by R_i in current map $Map(k)$ if R_i were put onto $f(k)$; $Region_{unknown}(k)$ denotes the unexplored region and $\min\{\cdot\}/S_i$ is the weight. If R_i is put onto $f(k)$, the less distance from the obstacles within $Sen_{i,f}(k)$, the less weight of the cell will be.

The robot R_i implements the subtractive clustering algorithm such that m_i clusters, of which the geometric centers are $c_j (j=1,2\cdots m_i)$, can be generated. We define each cluster center cen_j as the frontier cell that is nearest to c_j. Then the set of candidate local destination cells for R_i can be denoted as $D_{i\_candidate}=\{cen_j, j=1,2\cdots m_i\}$ $(i=1,2\cdots M)$. Fig. 1 shows a clustering result of the above algorithm, in which thirteen candidate cells are generated.

B. Calculation of destination cells

After the sets of candidate destination cells have been generated for all the robots, each robot will integrate its evaluating indexes of each candidate destination cell according to (2).

$$J_{i,j}=w_1\cdot G_{i,cen_j}/G_{i,\max}+w_2\cdot L_{i,\min}/L_{i,cen_j} \quad (2)$$

Where w_1 and w_2 are weights, $G_{i,\max}=\max\{G_{i,cen_j}, j=1,2\cdots m_i\}$, $L_{i,\min}=\min\{L_{i,cen_j}, j=1,2\cdots m_i\}$.

Then the local destinations of multiple robots are determined as the optimal cells satisfy:

$$\{D_1, D_2,\cdots D_M\}=\underset{D_{i\_candidate}(i=1,2\cdots M)}{\arg\max}\sum_{i=1}^{M}J_i \quad (3)$$

Subject to the constraints:

$$\begin{cases} J_i\in\{J_{i,j}, j=1,2\cdots m_i\} \\ \|D_p-D_q\|\le\min\{C_p,C_q\} \quad (\forall p,q=1,2\cdots M)\end{cases} \quad (4)$$

In (4), $\|D_p-D_q\|\le\min\{C_p,C_q\}$ indicates that local destinations of all the robots should meet the constraints on communication range.

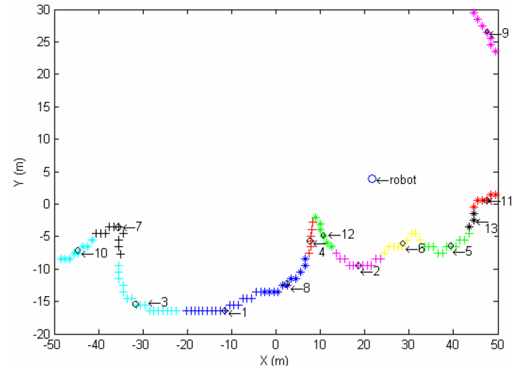

Fig.1. Generation of candidate destination cells

Here, for efficiency, we calculate $D_1, D_2,\cdots D_M$ in turn with greedy strategy. For robot R_1, D_1 can be decided as:

$$D_1=\underset{j_1=1,\cdots m_1}{\arg\max}J_{1,j_1} \quad (5)$$

And for R_i $(i=2,\cdots M)$,

$$D_i=\underset{j_i=1,\cdots m_i}{\arg\max}J_{i,j_i} \quad (6)$$

Subject to

$$\|D_i-D_q\|\le\min\{C_i,C_q\} \quad (q=1,2\cdots i-1) \quad (7)$$

IV. TRAJECTORY PLANNING

Since the selected destinations of the robot team lie on the frontiers between the explored and unexplored regions, at time step k, the region enveloped by frontiers is known, we denote it as $W_{known}(k)$, and the region outside frontiers is unknown, denoted as $W_{unknown}(k)=W-W_{known}(k)$. In such a way, the navigation of robot R_i from its current position to its local destination can be treated as a problem of trajectory planning in known environment $W_{known}(k)$. In this section, the task of trajectory planning is to generate M collision-free and separate trajectories, in order that the multiple robots can move along these trajectories and reach their individual destinations at the same time to communicate with each other.

We first enlarge the obstacles by the radius of the robot and reduce the robot to a particle. We divide the process of trajectory planning into multiple stages denoted as $PH^j=[t_{j-1},t_j]$ $(j=1,2\cdots;\ t_0=0)$. Each stage corresponds to a process of trajectory planning between sequential two destinations D_i^{j-1} and D_i^j. Given that $k=t_{j-1}$ (the initial time for the stage PH^j) is the current time, all the robots are located at D_i^{j-1} $(i=1,2\cdots M)$ which satisfy the constraints on communication range. At this time, the destinations D_i^j for stage PH^j are generated using the scheme introduced in section III for all the robots. Then, we discuss the generation of motion trajectories from D_i^{j-1} to D_i^j as follows.

Step 1. The robot of which the predictive navigation cost $L_{i,f}$ is largest is selected and labeled as R_l. General search algorithms such as A* and Dijkstra can be used to find a shortest (approximately shortest) collision-free path connecting D_l^{j-1} and D_l^j for R_l. This path will be smoothened by attempting to connect the initial cell with the other cells in turn until collision-free segment can not be generated. Then the endpoint of the last segment obtained is chosen as the new initial cell and the above process will be repeated until D_l^j. We denote the smoothened path as $Path_l^j :< P_i^j, i=1,2\cdots n >$, composed of cells P_i^j. The length of $Path_l^j$ is $Len(Path_l^j)=\sum_{i=1}^{n-1}\left|P_i^j P_{i+1}{}^j\right|$. For illustration, in Fig. 2 a collision-free path $A \to B \to C \to D \to E$, consisting of four segments can be smoothened by a shorter two-segment path, i.e. $A \to D \to E$, represented by $Path_l^j$: $< P_1^j, P_2^j, P_3^j >$.

Step 2. We will convert $Path_l^j$ into a motion trajectory for robot R_l. Given that the expected average velocity along $Path_l^j$ is $V(Path_l^j)$ and the expected motion steps along $Path_l^j$ can be calculated as:

$$T(Path_l^j)=\left\lceil \frac{Len(Path_l^j)}{V(Path_l^j)\cdot T_c} \right\rceil \tag{8}$$

Where $\lceil\cdot\rceil$ denotes the ceiling function, T_c is the control period of the robot. For each segment $< P_i^j P_{i+1}{}^j >$ on $Path_l^j$, R_l follows it with constant velocity and the actual motion steps can be obtained as:

$$T_{l,i}{}^j=\left\lfloor \frac{\left|P_i^j P_{i+1}{}^j\right|}{V(Path_l^j)\cdot T_c} \right\rfloor \tag{9}$$

Where $\lfloor\cdot\rfloor$ denotes the floor function. The actual motion velocity is $V_{l,i}{}^j=\left|P_i^j P_{i+1}{}^j\right|/(T_{l,i}{}^j\cdot T_c)$. As a result, the actual motion steps along the path $Path_l^j$ are $T_l^j=\sum_{i=1}^{n-1}T_{l,i}{}^j \approx T(Path_l^j)$ and $t_j=t_{j-1}+T_l^j$. Until now, the path $Path_l^j$ of R_l from D_l^{j-1} to D_l^j has been converted into its motion trajectory $traj_l^j=\{< V_{l,i}{}^j, T_{l,i}{}^j >, i=1,2\cdots n-1\}$. Then the robot R_l will tell its motion trajectory between D_l^{j-1} and D_l^j to the other robots $R_i\,(\forall i \neq l)$ by communication.

Step 3. In this step, we will generate motion trajectories for the other robots $R_i\,(\forall i \neq l)$. These robots should avoid collision between each other, simultaneously with R_l and OBS_{known}. However, computation complexity of the motion planning for $M-1$ robots will increase with M and the number of obstacles. In this paper, the motion trajectories of the other robots will be planned in turn after $traj_l^j$ has been available. Here, the method of velocity obstacles proposed in [16] is introduced to deal with this problem. We first take into account robot R_1 (if $l=1$, R_2 will be considered). R_1 treats R_l as a dynamic obstacle and has to avoid R_l and the observed static obstacles OBS_{known}. According to [16], the velocity obstacle of R_1 can be represented by VO_{R_l}, as shown with the shaded region in Fig.3.

At current time step $k=t_{j-1}$, R_1 can calculate the velocity obstacle VO_{R_l} at any time within PH^j because R_1 has known the motion velocity of R_l along each path segment $< P_i^j P_{i+1}{}^j >$. For static obstacles OBS_{known}, the velocity obstacles can also be decided using the same strategy [16]. Then, the trajectory planning for R_1 can be divided into two sub-processes including avoiding and tracking.

1) *Avoiding*. To disperse the robot team, R_1 will evade from R_l until R_l can observe D_l^j. And then, R_1 will shift its motion state from avoiding to tracking. At current time $k=t_{j-1}$, R_1 calculates its motion velocity during the avoiding sub-process within PH^j as follows.

$$V_1^j(t\,|\,t_{j-1})=\underset{V_1(t)\in U_{feasible}}{\arg\max}\,(w_3\cdot dis_R/dis_{R,\max}+w_4\cdot dis_{D,\min}/dis_D)$$
$$(t\in PH^j, D_l^j \notin Sen_l(t)) \tag{10}$$

Where, the two items in (10) mean that R_1 should move far away from R_l to disperse relative locations between these two robots, simultaneously tend to its destination cell D_1^j. $dis_R=\left|R_1(t\,|\,t_{j-1})\,R_l(t\,|\,t_{j-1})\right|$ and $dis_D=\left|R_1(t\,|\,t_{j-1})\,D_1^j\right|$ represent the predictive Euclidean distances from R_1 to R_l and from R_1 to D_1^j respectively if R_1 executes the velocity V_1^j. $U_{feasible}=U_1-(U_1\cap VO_{R_l})-U_{outfrontier}$. $U_1=\{V_1 | V_{1,\min}\le V_1 \le V_{1,\max}\}$, $U_{outfrontier}=\{V_1 | V_1\in U_1, (x_{r_1}, y_{r_1})\in W_{unknown}(t_{j-1})\}$. $dis_{R,\max}$ and $dis_{D,\min}$ represent the maximum value of dis_R and the minimum value of dis_D when $V_1(t)\in U_{feasible}$. Here, we restrict $V_1(t)\notin U_{outfrontier}$ because at current time $k=t_{j-1}$ the motion trajectory has to be planned only in the known region $W_{known}(t_{j-1})$. The solution to (10) can be obtained with the searching algorithm given in [16].

2) *Tracking*. If R_l moves along $traj_l^j$, the position at which R_l can observe D_l^j for the first time will be denoted as P, as illustrated in Fig.4. We translate the motion trajectory of R_l between P and D_l^j to the destination of R_1 such that a virtual trajectory can be created. Then we introduce a virtual robot R_l' and synchronize R_l' and R_l to track their individual trajectories. Thus, tracking to R_l will be equivalent to colliding with R_l' at D_1^j. The motion velocity of R_1 during tracking sub-process is planned as $V_1^j(t|t_{j-1})\in(VO_{R_l'}\cap\overline{U}_{outfrontier})$ $(t\in PH^j, D_l^j\in Sen_l(t))$, where $\overline{U}_{outfrontier}$ is the complementary set of $U_{outfrontier}$.

After R_1 has finished trajectory planning, it will communicate with the other robots R_i $(\forall i\neq 1,l)$ and share its motion trajectory. Then, regarding R_l and R_m $(m=1,2\cdots$

$i-1; m \neq l)$ as dynamic obstacles, the other robots R_i $(\forall i \neq 1, l)$ plan their trajectories in turn using scheme same as R_1. R_i $(\forall i \neq 1, l)$ should evade R_l and R_m $(m = 1, 2 \cdots i-1; m \neq l)$, simultaneously tend to D_i^j. When R_i tracks R_l, R_i should choose its motion trajectory without collision with R_m and static obstacles.

In the following, the complete algorithm of multi-robot cooperative exploration is given.

(i) $j = 1$, $k = t_0 = 0$. The initial poses of the robot team R_i $(i = 1, 2 \cdots M)$ are given as D_i^0 which satisfy the constraints on communication range, i.e. all the robots can communicate between each other. Each robot makes an observation.

(ii) $k = t_{j-1}$ $(j = 2, 3 \cdots)$, the robot R_i $(i = 1, 2 \cdots M)$ standing at D_i^{j-1} determines its local destination D_i^j of the stage PH^j. Then the robot R_l whose predictive navigation cost is largest will be selected. The shortest path $Path_l^j$ from D_l^{j-1} to D_l^j will be planned for R_l and converted into its motion trajectory $traj_l^j$. Finally, $traj_l^j$ will be transmitted to the other robots by communication.

(iii) The motion trajectory $traj_i^j$ during the stage PH^j will be planned in turn for the other robots R_i $(i \neq l)$.

a) $i = 1$;

b) R_l predicts that if it moves along $traj_l^j$, when $D_l^j \notin Sen_l(t)$ $(t \in PH^j)$, R_i will carry out the avoiding sub-process, go to c); when $D_l^j \in Sen_l(t)$ $(t \in PH^j)$ it will carry out the tracking sub-process, go to d);

c) In the avoiding sub-process, R_i $(i \neq l)$ will plan motion trajectory such that it can tend towards D_i^j, simultaneously can evade far from the robots R_m $(m = 1, \cdots i-1, l)$ that have finished trajectory planning;

d) In the tracking sub-process, R_i $(i \neq l)$ will plan motion trajectory such that it can track R_l;

e) $R_i (i \neq l)$ transmits its trajectory $traj_i^j$ by communication to the other robots R_m $(m = i+1, \cdots M, m \neq l)$;

f) If all the robots have finished trajectory planning, go to (iv); otherwise $i = i+1$, go to b);

(iv) All the robots $R_1, \cdots R_M$ execute the motion in the stage PH^j, in which $R_1, \cdots R_M$ will start at the same time to move along their individual trajectories until they reach the local destinations D_i^j $(i = 1, 2 \cdots M)$ simultaneously.

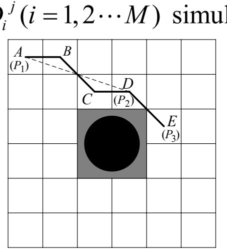

Fig.2 Path smoothing

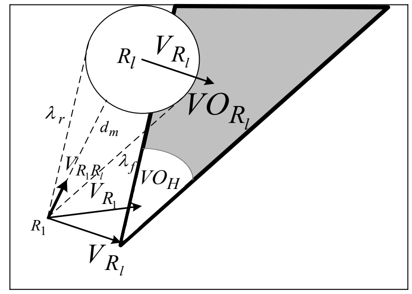

Fig.3 Velocity obstacle

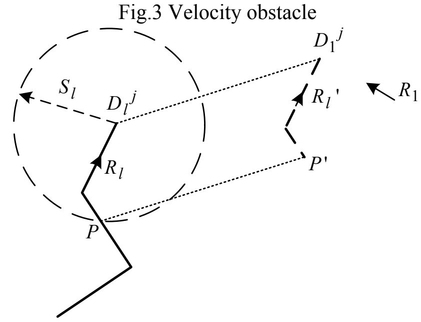

Fig.4 Tracking phase

(v) $j = j+1$, go to (ii) until the whole environment has been explored or the known region has been large enough.

V. SIMULATION

Simulations are conducted in a workspace with size of $100m \times 60m$, in which there exist nine circular obstacles with radii $2m$ or $3m$. Three robots are coordinated to explore the environment. The radius of each robot is $0.5m$, $S_1 = S_2 = S_3 = 8m$ and $C_1 = C_2 = C_3 = 16m$. The maximum linear velocity of each robot is $3m/s$, $V(Path_l^j) = 2m/s$. $T_c = 1s$, resolution of the evidence grid is $1m$. Here, Dijkstra algorithm is applied to search the shortest path. The algorithm will terminate when more than 98% of the whole environment has been detected. After 137 steps, the robot team has accomplished the exploration task. The simulation results are given in Fig.5. To display the influence on exploration under different communication ranges, two kinds of other communication range including $C_1 = C_2 = C_3 = 8m$ and $C_1 = C_2 = C_3 = 24m$ are selected. And the simulation results under three different communication ranges are shown in Table 1, from which we can find that exploration efficiency will increase as C_i because larger communication range allows more dispersancy among the robots.

To testify the utility of evaluation function given as Eq. (2), we compare with the strategy of nearest-frontier selection proposed in [2], in which each robot selects the nearest frontier cell as the next local destination, i.e. the Euclidean distance is calculated as evaluation function. When the local destinations of all the robots are decided, trajectory planning scheme proposed in section IV will be also introduced to guide the robot team to reach their local destinations simultaneously. The simulation results are given in Table 1 and the explored environment when $C_1 = C_2 = C_3 = 16m$ is shown in Fig. 6. Due to the communication constraint, the robots have to distribute at their local destinations within a relatively small region. In nearest-frontier strategy, each robot will select the nearest frontier cell such that all the robots could select very adjacent frontier cells as their next destinations. Moreover, each robot calculates local destination without considering its information gain. As a result, the total information gain acquired by the robot team will be less in each step and more exploration steps have to be spent.

VI. CONCLUSION

In this paper, multi-robot cooperative exploration is investigated under the constraints on communication range. In view of the information gain, navigation cost and distance adjacency, the frontier cells between the known and unknown regions are grouped into different clusters, from which the optimal frontier cells satisfying communication constraints are selected as local destinations. Then the exploration in unknown environment can be converted into a problem of multi-stage trajectory planning in known environment. Avoiding and tracking are introduced to realize disperse exploration and synchronous rendezvous for multi-robot team such that communication can be achieved at the local destinations.

In future work, we will deal with the cooperative exploration with motion and measurement uncertainty, i.e. multi-robot SLAM. The scheme proposed in this paper will be combined with the existing SLAM techniques to solve multi-robot SLAM effectively.

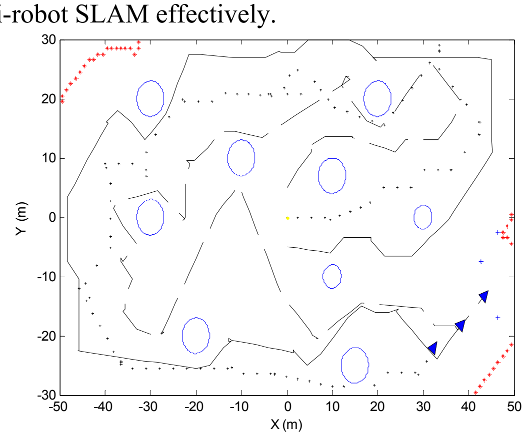

Fig.5 Simulation result of our strategy

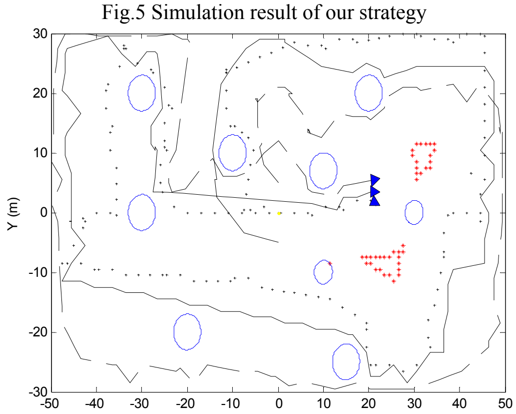

Fig.6 Simulation result of nearest-frontier strategy

Table 1 Simulation results under different communication ranges

C_i	strategy	Exploration steps	Area explored solely by R_1	Area explored solely by R_2	Area explored solely by R_3
8	Our strategy	201	12.36%	6.98%	11.19%
	Nearest frontier	265	10.06%	4.95%	10.34%
16	Our strategy	137	19.51%	9.73%	22.06%
	Nearest frontier	169	16.07%	6.35%	18.21%
24	Our strategy	98	26.27%	11.90%	23.83%
	Nearest frontier	123	21.77%	8.61%	24.12%

REFERENCES

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

[2] B. Yamauchi, "Frontier-based exploration using multiple robots", in *Proc. the Second International Conference on Autonomous Agents*, Minneapolis, 1998, pp. 47-53.

[3] W. Burgard, M. Moors, D. Fox, R. Simmons and S. Thrun, "Collaborative multi-robot exploration", in *Proc. IEEE Int. Conf. on Robotics and Automation*, San Francisco, 2000, pp. 476-481.

[4] W. Burgard, M. Moors, C. Stachniss and F. Schneider, "Coordinated multi-robot exploration", *IEEE Trans. Robotics*, vol. 21, pp.376-386, 2005.

[5] J. Ko, B. Stewart, D. Fox, K. Konolige and B. Limketkai, "A practical, decision-theoretic approach to multi-robot mapping and exploration", in *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, Las Vegas, 2003, pp. 3232-3238.

[6] M. Baglietto, M. Paolucci, L. Scardovi and R. Zoppoli, "Information-based multi-agent exploration", in *Proc. IEEE Int. Workshop on Robot Motion and Control*, Bukowy, 2002, pp. 173-179.

[7] R. M. Zlot, A. Stentz, M. B. Dias and S. Thayer, "Multi-robot exploration controlled by a market economy", in *Proc. IEEE Int. Conf. on Robotics and Automation*, Washington, 2002, pp. 3016-3023.

[8] W. Sheng, Q. Yang, S. Ci, and N. Xi, "Distributed multi-robot coordination algorithm for area exploration and coverage", in *Proc. IEEE Int. Conf. on Robotics and Automation Workshop on the State-of-the-Art of Mobile Robot Area Coverage,* New Orleans, 2004.

[9] N. Roy and G. Dudek, "Collaborative robot exploration and rendezvous: algorithms, performance bounds and observations", *Autonomous Robots*, vol. 11, pp. 117-136, 2001.

[10] X. S Zhou and S. I. Roumeliotis, "Mulit-robot SLAM with unknown initial correspondence: the robot rendezvous case", in *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems*, Beijing, 2006, pp. 1785-1792.

[11] A. R. Mosteo, L. Montano and M. G. Lagoudakis, "Multi-robot routing under limited communication range", in *Proc. IEEE International Conference on Robotics and Automation*, Pasadena, 2008, pp.1531 -1536.

[12] A. Elfes, "Sonar-based real-world mapping and navigation", *IEEE Journal of Robotics and Automation*, vol. 3, pp. 249-265, 1987.

[13] V. Lumelsky and A. Stepanov, "Path planning strategies for a point mobile automaton moving amidst unknown obstacles of arbitrary shape", *Algorithmica*, vol. 3, pp. 403-430, 1987.

[14] V. Lumelsky and K. Harinarayan, "Decentralized motion planning for multiple mobile robots: the cocktail party model", *Autonomous Robots*, vol. 4, pp. 121-135, 1997.

[15] S. Chiu, "Fuzzy model identification based on cluster estimation", *Journal of Intelligent and Fuzzy System*, vol. 2, pp. 68-72, 1994.

[16] P. Fiorini and Z. Shiller, "Motion planning in dynamic environments using velocity obstacles", *International Journal of Robotics Research*, vol. 17, pp. 760-772, 1998.