

# Multi-Robot Exploration and Fire Searching

Ali Marjovi, João Gonçalo Nunes, Lino Marques, Aníbal de Almeida  
Institute of Systems and Robotics  
University of Coimbra  
Portugal

Email: {ali, jgnunes, lino, adealmeida}@isr.uc.pt

**Abstract**—Exploration of an unknown environment is a fundamental concern in mobile robotics. This paper presents an approach for cooperative multi-robot exploration, fire searching and mapping in an unknown environment. The proposed approach aims to minimize the overall exploration time, making it possible to localize fire sources in an efficient way. In order to achieve this goal, the robots should cooperate in an effective way, so they can individually and simultaneously explore different areas of the environment while they identify fire sources. The proposed approach employs a decentralized frontier based exploration method which evaluates the cost-gain ratio to navigate to target way-points. The target way-points are obtained by an A\* search variant algorithm. The potential field method is used to control the robots motion while avoiding obstacles. When a robot detects a fire, it estimates the flame's position by triangulation. The communication between the robots is done in a decentralized control way where they share the necessary data to generate the map of the environment and to perform cooperative actions in a behavioral decision making way. This paper presents simulation and experimental results of the proposed exploration and fire search method and concludes with a discussion of the obtained results and future improvements.<sup>1</sup>

## I. INTRODUCTION

Searching operations inside buildings, caves, tunnels and mines are sometimes extremely dangerous activities. The use of autonomous robots to perform such tasks in complex environments will reduce the risk of these missions. Exploration of an unknown environment is a fundamental issue in mobile robotics. As autonomous exploration and map building becomes increasingly robust with a single robot, the next stage is to extend these techniques to teams of robots. Using multiple robot systems may potentially provide several advantages over single robot systems namely speed, accuracy, and fault tolerance [1], [15], [3] and [5]. Nowadays, swarm based exploration and mapping where the robots can be smoothly added or removed to the operation is an area with increasing interests to the robotics community [1]. Cooperation, map merging, decision making, dealing with uncertainty in localization and reasoning, task sharing and navigation are the most significant research topics in multi-robot exploration.

This study is a part of a European project named Guardians<sup>2</sup>. The Guardians are a swarm of autonomous robots applied to navigate and search an urban environment. The project's central example is search and rescue in an industrial

warehouse in smoke, as proposed by the Fire and Rescue Service of South Yorkshire. The job is time consuming and dangerous; toxins may be released and humans senses can be severely impaired. They get disoriented and may get lost. The robots warn for toxic chemicals, provide and maintain mobile communication links, infer localization information and assist in searching. Map exploration and fire source detection are the topics in this paper.

The problem of coordination and control of multiple robots for mapping and exploration has been almost addressed through several research approaches. Most approaches rely upon centralized control to direct each vehicle in the swarm. This centralized approach has been popular in the robotics community, because it allows near optimal behaviors in well understood environments. However its performance decreases in new unidentified environments. Yamauchi [6] proposed a distributed method for multi-robot exploration, yielding a robust solution even with the loss of one or more vehicles in the swarm. A key aspect of this approach involves sharing map information among the robotic agents so they execute their own exploration strategy, independently of all other agents. While this technique effectively decentralizes control, exchange of map information is not enough to prevent inefficient cooperative behaviors. This approach also required known starting positions and failed to provide a robust mechanism for map merging [8].

Simultaneous localization and mapping (SLAM) has been the topic of much interest because it provides an autonomous vehicle with the ability to discern and represent its location in a feature rich environment. But if there is a localization system and robots know their relative positions, SLAM techniques are not required.

There are some proposed methods for exploration based on cooperation between agents. Several researchers have suggested stigmergy methods [7] and [8]. Stigmergy is a mechanism of spontaneous, indirect coordination between agents or actions, where a trace left in the environment by an action stimulates the performance of subsequent actions, by the same or by different agents. Nevertheless, this method is mainly useful when there are a lot of small robots working together.

Most of the existing approaches to coordinate multi-robot exploration assume that all agents know their locations in a shared (partial) map of the environment. Effective coordination can be achieved by extracting exploration frontiers from the partial map and assigning robots to frontiers based on a global measure of performance [1], [15], [3] and [9]. Frontiers

<sup>1</sup>This work was partially supported by project GUARDIANS contract FP6-IST-045269 and also Portuguese Foundation for Science and Technology contract SFRH/BD/45740/2008.

<sup>2</sup><http://www.guardians-project.eu>

are the borders of the partial map, between explored free space and unexplored areas [15]. These borders thus represent locations that are reachable from within the partial map and provide opportunities for exploring unknown terrain, thereby allowing the robots to greedily maximize information gain [10]. Compared to the problems occurring in single robot exploration, the extension to multiple robots poses several new challenges, including:

**Coordination and cooperation:** Since there are several robots working in the same area, they must have some kind of cooperation with each other in order to prevent collisions and sharing the tasks. If the robots know their relative locations and share a map of the area which they have explored so far, effective cooperation can be achieved by guiding the robots into different non-overlapping areas [15], [3], [11]. The idea is that at a given time each robot should be dedicated to exploring one and only one frontier.

**Integration of information collected by different robots into a single map:** The main goal of exploration is to build a general map representing the environment. The robots should integrate all the data into a single map. Map merging and map sharing are two big challenges in this field that has been addressed in several studies [2].

**Dealing with limited communication:** Communication protocol between robots and the amount of data which should be transferred in the media are the other noticeable challenges.

**Uncertainty in localization and sensing:** The effect of sensor errors (“noise”) and errors in sensing the gradient of a “resource profile” (e.g., a nutrient profile) should be considered. Several researchers have illustrate that the agents can forage in noisy environments more efficiently as a group than individually [12], [13]. Sermanet et al. presented a mapping and planning system that accurately represents range and category uncertainties, and accumulates the evidence from multiple frames in a principled way [4].

**Decision making, reasoning, task sharing and navigation:** Decision making for each robot in an unknown environment is a very complex problem. Since nobody knows what is after the frontier of an unexplored area, there is no unique optimum algorithm that is completely reliable. In each situation a robot should make a decision to progress exploring task based on partial existing map and also the other robots’ positions and objectives.

In terms of decision making algorithms, a lot of studies for multi-robot exploration do not address unknown environments. Moreover, most of the research in this field is based on centralized control of the agents. More significant approaches for multi-robot exploration have been presented in [14] and [6]. In both techniques the robots share a common map which is built during the exploration. Singh and Fujimura [14] presented a decentralized online approach for heterogeneous robots. Most of the time, the robots work independently. When a robot finds a situation that is difficult to resolve by itself, it will send the problem to another robot which is likely to be able to resolve the situation. The candidate robot is chosen by trading off the number of areas to be explored, the size of the

robot and the straight-line distance between the robot and the target region. This technique generates a grid geometric map; therefore the accuracy of the of map depends on the grid size. Also all the robots need to have a huge memory to keep the entire map. In the approach of Yamauchi [6] the robots move to the closest frontier which is the closest unknown area around the robot according to the current map. However, there is no coordination component which chooses different frontiers for the individual robots.

Our approach, in contrast, is specifically designed to coordinate the robots so that they do not choose the same frontier. Furthermore it generates a topological map which needs much less memory capacity. As a result, this method needs significantly less time to accomplish the task. The objective in this research is to generate the map of an unknown environment and also localize all the fire sources in the area. Actually the final goal is to create a fire risk map of an unknown environment with multiple robots. We do not address the problems of risk map in this paper.

During the exploration process if there is a fire source, robots should report it. In terms of fire source detection, authors have addressed the issue in previous papers [16], [17], [18] and [19]. The last achievement of that research is kheNose. The kheNose is a device developed by the authors to sense olfactory information through the use of gas sensors, anemometers, a temperature and humidity sensor [20]. In the current study the last version of kheNose has been used to detect the fire sources. Collision avoidance between the robots during the exploration is a considerable issue that has not been addressed pragmatically in the previous studies. In this study, we propose a new practical method for multi-robot unknown environment exploration with fire source detection which takes “collision avoidance” and “task sharing” into consideration. This method has been tested in the real world and also in simulation. The effect of complexity of the environment and the numbers of robots which are participating in the explorations are the main parameters that have been studied with this method in this paper.

## II. THE PROPOSED METHOD

This section explains concept of proposed multi-robot cooperation technique. This method is illustrated in schematic diagram Fig.1 in a globally point of view. As it’s shown in the diagram, the method includes three main tasks: navigation and exploration, decision making, and fire source detection. Following parts of this section describes some of the parts of the schematic diagram (Fig.1).

### A. Motion control

The robots must be able to navigate through the paths without colliding into obstacles. The method used for obstacle avoidance is based on virtual force fields. In this method the obstacles detected by the robot apply repulsive forces keeping it from hitting the walls, while the objective target applies an attractive force to the robot guiding it to the objective direction. The summation of the repulsive and attractive forces gives the

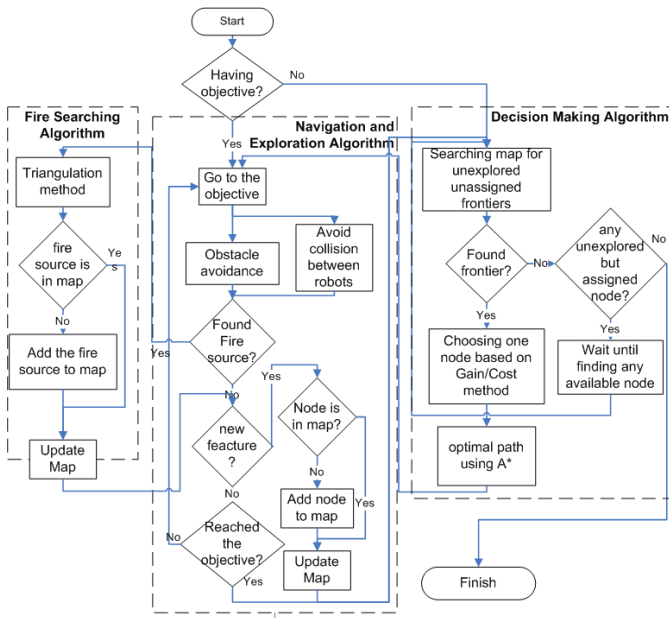


Fig. 1. Whole system's schematic

actual direction that the robot should navigate and the exact speed and turn rate it should take.

### B. Decision making

The main goal of the exploration process is to cover the whole environment in the minimum possible time. Therefore, it is essential for the robots to share their tasks and individually reach the objectives through optimal paths. In an unknown environment the immediate goals are the frontiers. Frontiers are the borders of the explored area with the unexplored area, where robots can expect to gain the knowledge of the environment. Most of the time when the robots are exploring the area, there are several unexplored regions; this poses a problem of how to assign specific frontiers to the individual robots. We want to avoid sending several robots to the same frontier which may result in collision concerns. The other issue is that we do not want to have a central station for moderating the robots. To address these problems the proposed method is based on a decision-theoretic exploration strategy. The decision making algorithm is shown in Algorithm.1.

The frontier is selected based on the cost of reaching it and the utility it can provide to the exploration. The cost is calculated through the A\* method, which simultaneously determines the optimal path to reach the frontier and its distance. Therefore the cost is proportional to the distance that the robot has to travel to reach the frontier.

$$cost = dist(A_{i=0,n}^*[(X_R, Y_R), (X_{f_i}, Y_{f_i})])$$

where:  $(X_{f_i}, Y_{f_i})$  : position of the frontier  $i$   
 $(X_R, Y_R)$  : position of the robot, and  $n$  : number of frontiers.

The utility depends on the number of the robots and their proximity to the frontier; it means that if there are several frontiers at similar distances, the robot will go to the one that has higher utility. This procedure will make the robots disperse and explore the environment in a more efficient way.

### Algorithm 1: Map exploration algorithm - Decision making method

```

1 Receive_map_from_server()
2 while there is at least one unexplored link in the map do
3   Follow the potential field algorithm until getting a different feature
   in the environment
4   Receive_map_from_server()
5   if the new node exists in the map then
6     Update the map's data with new information.
7     Send_map_to_the_server()
8   else
9     Add_new_node_to_map()
10    Send_map_to_the_server()
11  if the current node has any unexplored link then
12    Calculate direction gain of taking each unexplored link, based
    on the position of the other robots and get the higher gain
    direction to follow
13  else
14    Determine the best not-assigned unexplored frontier,
    based on their gain / cost.
15    Assign the frontier to the robot.
16    Calculate the best path to take, based on the A*
    algorithm and get that direction to follow.
17 End of algorithm

```

$$utility = \sum_{i=1|i \neq R}^m dist[(X_{f_k}, Y_{f_k}), (X_{r_i}, Y_{r_i})]$$

where:  $(X_{r_i}, Y_{r_i})$  : position of the robot  $i$   
 $m$  : number of the robots.

The cooperation between the robots permits the exchange of data, allowing the task sharing and consequently an efficient distributed exploration. During the exploration there is only one global shared map in the system. This map is in a server that sends and receives the map to the robots whenever they request for it, which is when the robots acquire new information of the environment. Within this map, besides having some information regarding the kind of nodes and their position, it also has the data describing the location of the robots and their frontier target, as can be seen in Fig.2. Through this data, a robot can see which frontiers are unexplored, their position and if any robot has targeted them as its objective, thus allowing a distributed efficient exploration (see Algorithm.1 and diagram Fig.1). The cooperation between multiple robots is more complex than to give different frontiers to each one individually. In the case of dealing with multiple robots in one environment, collision between robots is a very important aspect. As it is stated earlier, this research has already addressed the collision avoidance of walls with the implementation of virtual force fields, but the collision between the robots has not been addressed yet. For instance, two robots might be in a narrow corridor with different directions and they may want to pass but cannot because they are facing each other or they may even treat each other as a dead end. It is necessary to avoid this type of problems. Therefore we have implemented some rules of engagement to prevent exactly these kind of situations. If the two robots are going against each other in a corridor, it's because the task sharing has not been done previously in the most efficient way. These rules of engagement are actually an algorithm shown in

**Algorithm 2:** Collision avoidance between the robots (and increasing efficiency for collaborating multi-robot exploration)

```

1 Calculate a confined circular area around the robot
2 if any other robot is inside the circle then
3   Determine if the robots are in a collision pattern
4   if there is a possibility of being in a collision pattern then
5     //Follow the rules of engagement:
6     if they are in a direct collision path then
7       reevaluate their goals.
8     else
9       if they are both currently exploring frontiers OR
10      they are both moving inside explored area then
11         give priority to the one which has lowest ID.
12       else
13         Give priority to the one that is exploring a
14         frontier.
15 else
16   Continue exploration algorithm.
17 End of algorithm

```

Algorithm.2.

### C. Map structure and map merging

As it is mentioned, the map is shared between the robots and it includes necessary information for the cooperation between the robots. When a robot asks for the map, it merges the received map with the one it is constructing through the exploration. When the robots are mapping the environment they are constructing the map and verifying if the current node they have acquired isn't already in the map, thus assuring the coherency of the map and making the merging process simple, where most of the times it is only needed to add new acquired nodes to the global shared map. In the Fig.2 it can be seen an example of the map as well as a small description of its content (see Algorithm.1 and diagram Fig.1).

Number of explored nodes	Topological Map +					Connection to other nodes:					
Index of the node	Start and End position					- 1 means it is unexplored; - 2 means it has no connections; - And the other numbers specify to which node it is connected.					
0	72	144	28	151	28	1	-1	-1	-1	0	0
1	24	151	28	183	27	2	0	-2	-2	0	0
2	71	183	27	187	52	-2	1	3	-2	2	0
3	32	187	52	187	86	-2	-2	4	2	1	0
4	72	187	86	225	98	5	-2	-1	3	0	0
Type of the node						Contains the ID of the robot currently occupying the nodes, therefore giving the position of the robots					

Fig. 2. Example of topological map data

### D. Fire source detection

During exploration and navigation, the robots are simultaneously acquiring information from the environment (see diagram Fig.1). All the robots are equipped with a board developed by this research group which integrated temperature and chemical sensors named kheNose (Fig.3). They use an eight element thermopile array sensor to measure the absolute temperature as well as the ambient temperature on the robot to be able to distinguish the heat values. When the data received from the sensor informs that there is an evidence of existence of a fire source in the environment, the robot identifies it as a heat source clue and proceeds to make a sequence of movements based on triangulation method to acquire more

accurate values in order to be certain that it isn't a random value.

KheNose is composed by six transducer interface modules (TIM): An eCO, three thermal anemometers, and two eNostrils. The eCO and the anemometers are single channel IEEE1451.4 compliant boards and the eNostrils are double channel boards. All the functions related with the transducers, namely signal conditioning, data acquisition and processing and calibration management are performed by the kheNose board. The calibration data for each sensing module is stored in a local EEPROM located in the module. A Microchip dsPIC33F controller acquires all the analog and digital information from the sensors, processes that data and sends it to the Khepera III extension board. This extension board supports several communication protocols, like I2C, used to physically connect the kheNose to the Khepera III [20]. KheNose uses an 8 element thermopile array sensor to measure the absolute temperature as well as the ambient temperature on the robot to be able to distinguish the difference between the heat values (shown in Fig.3). When the temperature data received from the sensor informs that there is an increase in temperature, the robot identifies it as a heat source clue and proceeds to make a sequence of movements to acquire more accurate values to be certain it isn't a random value.



Fig. 3. Khepera III and kheNose with sensing modules

## III. EXPERIMENTS

As a testing plan, we built several maze-like environments which are combinations of corridors, corners, crossings, T-junctions and dead-ends (one of them is shown in Fig.4). Three Khepera III robots are used for testing the algorithm.

Robots should know their positions in the environment to be able to explore and navigate. Localization in multi-robot systems (such as SLAM problem) is not among the main considerations of this research, so a network camera is mounted on the top of the environment and an image processing computer program is able to track and locate each robot. Each Robot has two colored labels on the top that can be seen by the camera. The camera is connected to the network and an image processing program tracks the robots' position and provides the absolute pose of each robot via wireless network. Image processing program is an object tracking application developed by the authors. Recognizing the center of each colored label and also calculating the line crossing from these two centers, the orientation of the robot can be computed. The program is written in C++. Each Khepera can

request its absolute pose from this program through wireless network. Fig.4 shows a screen shot of this program.

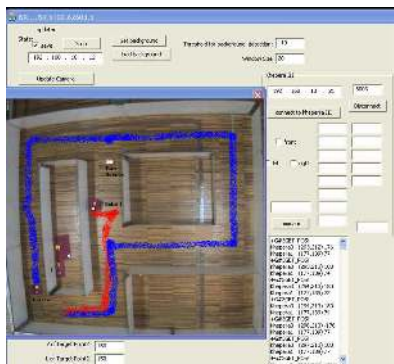


Fig. 4. Real maze experiment, Visual positioning application screenshot

In terms of feature extraction, based on values measured by sonar and infrared sensors, the robot recognizes the features and should take an action and modify the shared map; it will save this data in the map structure as a new node, and will also update the data related to the previous feature. For each feature, the robot saves the data in the topological map, including start position and end position of that node and some other information (Fig.2). Each Khepera III has several infrared sensors and also sonar sensors. Robots are using these sensors for feature extraction and navigation.

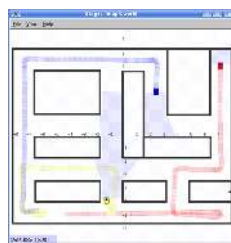


Fig. 5. Maze No. 1, 34 nodes, 3 robots exploring, player/stage screenshot

Different maze structures were tested. The system has been tested with different start positions for the robots. There is a small candle acting like a heat source in the environment which robots try to localize it. All of the robots are equipped with kheNose boards for heat source detection.

The algorithm has been tested in the real world and also in a simulation world. For optimizing the exploration algorithm and measuring its performance, we used the Player/Stage simulator. In the real world, there are a lot of constraints that do not let to test the proposed method very easily. It is not effortless to build various test plans with different scales for testing the method. The other constraint in the real world testing plan is the visual positioning system that makes a restriction to work in a fixed constant area since the area must be in sight of the camera. For these reasons, the system (except for the heat source detection) is developed and evaluated in the simulation before real world experiments. Since there is no reliable simulator for fire and smoke in Player/Stage, the whole system has been tested in the real world.

Fig.4 shows two robots exploring a small maze and finding a fire source. Both robots started from the same point but not at the same time. We intentionally ran one of the robots a few seconds after the first one. The red footprint shows the first robot's path and the blue footprint is related to the second robot. As it is shown, the first robot has found the fire source. For an example of the coordination algorithm, when the second robot reached the junction it figured out that the path in the front is already explored and it chose the right way.

Since there is no accepted standard benchmark, measuring the performance of a behavioral based multi-robot unknown area exploration algorithm is a very difficult job. One of the possible ways to do that is to compare the proposed method with optimal possible method. But the issue is that there is no optimal method for exploring an unknown world. But there is an optimal solution for minimizing the traveling path if the world (maze) is completely known before exploration.

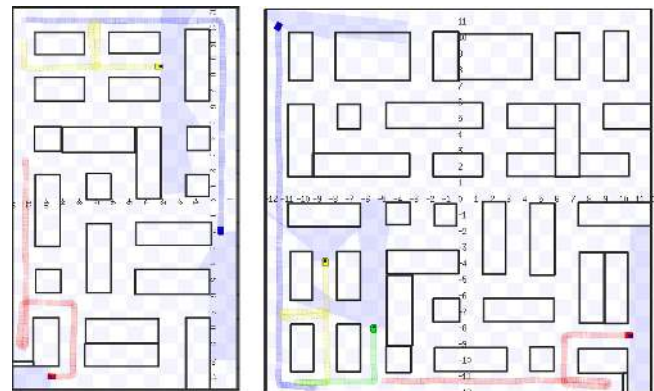


Fig. 6. Maze No.2(left), 82 nodes, 3 robots exploring. Maze No.3 (right), 135 nodes, simulation, 4 robots exploring

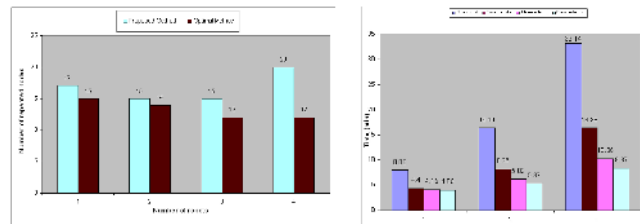


Fig. 7. Left: Comparing the results of the proposed method with optimal method, Right: Test of various numbers of robots against complexity of the environment, 1: simple maze(Fig.5), 2: maze Fig.6, 3: complex maze(Fig.6)

the algorithm has been tested with different number of robots in a specific maze. The model of that maze is also given to the optimal method and then we have compared the results of the proposed algorithm with the optimal method. Since the optimal method has the world's model but the proposed method is exploring the unknown world, it is obvious that the results of proposed method is always worse than optimal method but it can be a good criteria for evaluating the method.

Number of repeated nodes during the travel can be a good parameter for measuring the performance of the method. A



repeated node is a node that robots pass for more than one time. Fig.7 (left) is showing the number of nodes that have been repeated more than once in the optimal method as well as in the proposed algorithm for the maze shown in Fig.5. A good conclusion from the graph in Fig.7 (left) is that there is a trade-off between the number of robots and the size of the world. This graph is showing that the proposed approach is acceptably comparable with optimal method.

Another parameter for evaluation of the method is the exploration time. The proposed method has been tested with different number of robots in different mazes. The environment shown in Fig.4 that is a 3.5 x 4 meters maze is tested by one, two and three Khepera robots separately. one robot could explore the environment in 412 seconds. This environment has been explored by two robots in 254 seconds. The exploration time for the same maze with three Kheperas was 212 seconds. Each result is the average of five similar tests. Different tests with constant conditions had similar results with about 10 percent variance. In all the tests the maximum speed of the Kheperas are kept constant.

In simulation, the mazes shown in Fig.5, Fig.6 and Fig.6 have been tested separately with one, two, three and four robots and the results are shown in Fig.7(right). The graph shows the average of five times tests for each data. the variance was again about ten percent. It is obvious that with more robots, exploration time will be improved. Another conclusion from the graph is that having more number of robots is more advantageous in a complex maze than in a simple maze. this also proves that cooperation algorithm in this approach is efficiently functional.

The real experiments test are very similar to simulations. The biggest difference is that in the real experiment the fire source detection was also tested and the robots could also locate the fire sources during the exploration. However in real world test, the mazes are more simple and we are dealing with uncertainty in data received by the sonar sensors.

The performance of fire source detection has been addressed in previous studies by the group [18], [20]. The robots are able to realize and localize the fire source in the environment.

#### IV. CONCLUSIONS AND FUTURE WORKS

The proposed method for multi-robot unknown environment exploration has been implemented and experimented in the test plans. The robots are able to cooperate and create a shared topological map of the unknown environment. Cooperation between the robots is done by sharing information in the shared map. The algorithm has been tested against several different configurations in Player/Stage simulation program. The final working algorithm is merged with fire source detection algorithm and has been tested in the real world. The effect of the number of the robots on exploration in different type of environment has been discussed. The results show the efficiency and reliability of this method.

Visual localization system provides many constraints for testing, it is being replaced by a none-centralized localization system that integrates odometry with gyroscopes and

accelerometers as well as an inbuilt compass. The system works only in a maze like environment; it should work in a real environment with less restrictions. For this purpose, feature extraction should be more developed, range sensors should be more accurate and the topological map should contain some geometric data. The current algorithm is working based on a graph. However in a real environment it is very difficult to model the area with a graph. Maybe a mixture of topological and geometric maps will be useful in this case.

#### REFERENCES

- [1] D. Fox, J. Ko, K. Konolige, B. Limketkai and B. Stewart, "Distributed multi-robot exploration and mapping", *Proceeding of the IEEE, Special Issue on Multi-Robot Systems*, 94(7), 2006.
- [2] M. Pfingsthorn and A. Birk, "Efficiently communicating map updates with the pose graph", *Proceeding of the IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS*, P.2519-2524, 2008.
- [3] W. Burgard, M. Moors, C. Stachniss and F. Schneider, "Coordinated multi-robot exploration", *Proceeding of the IEEE, Robot.*, Vol. 21, No. 3, P. 376386, Jun. 2005.
- [4] P. Sermanet, R. Hadsell, M. Scoffier, U. Muller and Y LeCun, "Mapping and planning under uncertainty in mobile robots with long-range perception", *Proceeding of the IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS*, P.2525-2530, 2008.
- [5] G. Dedeoglu and G.S. Sukhatme, "Landmark-based matching algorithm for cooperative mapping by autonomous robots", *Proceeding of 5th Int. Symp. Distributed Autonomous Robotic Systems (DARS)*, 2000.
- [6] B. Yamauchi, "Frontier-based exploration using multiple robots", *Proceeding of Second Int. Conf. Autonomous Agents*, 1998.
- [7] D. Scheidt, J. Stipes and T. Neighoff, "Cooperating Unmanned Vehicles", *Proceeding of Networking, Sensing and Control, IEEE*, 2005
- [8] J. Stipes, R. Hawthorne, D. Scheidt and D. Pacifico, "Cooperative Localization and Mapping", *Proceeding of the IEEE, Networking, Sensing and Control*, 2006
- [9] S. Koenig, C. Tovey and W. Halliburton, "Greedy mapping of terrain", *Proceeding of the IEEE, ICRA*, 2001
- [10] R. Zlot, A. Stentz, M. Bernardino Dias and S. Thayer, "Multi-robot exploration controlled by a market economy", *Proceeding of the IEEE, ICRA*, 2002
- [11] M.W.M. Dissanayake, P. Newman, S. Clark, H.F. Durrant-Whyte and M. Csorba, "A solution to the simultaneous localization and map building (SLAM) problem", *Proceeding of the IEEE, Robot. Automat.*, Vol. 17, No. 3, Jun. 2001
- [12] Y. Liu and K.M. Passino, "Stable Social Foraging Swarms in a Noisy Environment", *IEEE Transactions on Automatic Control*, Vol. 49, No. 1, Jan. 2004
- [13] Y. Liu and K.M. Passino, "Biomimicry of social foraging behavior for distributed optimization: Models, Principles and emergent behaviors", *Theory Application*, Vol. 115, P. 603-628, Dec. 2002.
- [14] K. Singh and K. Fujimura, "Map making by cooperating mobile robots", *Proceeding of the IEEE, ICRA*, 1993
- [15] W. Burgard, M. Moors, D. Fox, R. Simmons and S. Thrun, "Collaborative multi-robot exploration", *Proceeding of the IEEE, ICRA*, 2000.
- [16] L. Marques, N. Almeida and A.T. Almeida, "Olfactory sensory system for odour-plume tracking and localization", *Proceedings of IEEE, Sensors*, Vol. 1 2003.
- [17] L. Marques and A.T. Almeida, "Electronic nose-based odour source localization", *6th International Workshop on Advanced Motion Control*, 2000.
- [18] L. Marques, U. Nunes and A.T. Almeida, "Particle swarm-based olfactory guided search", *Springer journal of Autonomous Robots*, Vol. 20, No. 3, pages 277-287, 2006
- [19] L. Marques, U. Nunes and A.T. Almeida, "Olfaction-based mobile robot navigation", *Elsevier journal of Thin Solid Films*, Vol. 418, No. 1, pages 51-58, 2002
- [20] J. Pascoal, P. Sousa and L. Marques, "KheNose - A smart transducer for gas sensing", *11th International conference on Climbing and walking robots and the support technologies for mobile machines, CLAWAR*, 2008