Ad Hoc Networks xxx (2012) xxx–xxx

Contents lists available at SciVerse ScienceDirect

Ad Hoc Networks

journal homepage: www.elsevier.com/locate/adhoc

The effects of exploration strategies and communication models on the performance of cooperative exploration

Gurkan Tuna [a,*], Kayhan Gulez [b], V. Cagri Gungor [c]

[a] Department of Computer Programming, Trakya University, Edirne, Turkey
[b] Department of Control and Automation Engineering, Yildiz Technical University, Istanbul 34349, Turkey
[c] Department of Computer Engineering, Bahcesehir University, Istanbul 34353, Turkey

ARTICLE INFO

Article history:
Received 1 November 2011
Received in revised form 3 July 2012
Accepted 31 July 2012
Available online xxxx

Keywords:
Cooperative exploration strategies
Communication models
Processing time requirements
Exploration performances
NetBeans IDE

ABSTRACT

Exploration with mobile robots is utilized in a wide range of applications including search and rescue missions, planetary exploration, homeland security, surveillance, and reconnaissance. Cooperative exploration offers the potential of exploring an unknown zone more quickly and robustly than single-robot case. However, coordinating multiple robots is a challenging task due to heterogeneous processing and communication requirements, and the complexities of exploration algorithms. This paper presents a comparison of different cooperative exploration strategies, such as frontier-based exploration, market-driven exploration, and role-based exploration, based on their exploration performances and processing time requirements. To show the effect of CPU power on the processing time of the exploration algorithms, two notebooks and a netbook with different specifications have been extensively used. Comparative simulation results of our own application developed in Java show that the processing time requirements are consistent with the computational complexities of the exploration strategies. The results we obtained are consistent with the CPU power tests of independent organizations, and show that higher processing power reduces processing time accordingly.

1. Introduction

With the recent advances in robotics, digital electronics, and wireless communications, mobile robots are currently being used in many mission-critical applications including search and rescue missions in disaster zones, inspection of hazardous areas, planetary exploration, homeland security, surveillance, and reconnaissance [25]. Although these scenarios clearly differ from each other in some aspects, they share many common requirements including full or partial autonomy, gathering information in an unknown zone and mapping, communication, information relay, and coordination. In many of the scenarios where mobile robots can be used, there is no priori map. Hence, exploration of an unknown zone is the first step of all these envisaged scenarios. Exploration is a complicated work with two main goals. Gathering information in the zone is the first goal. And the second goal is to ensure that the target zone has been completely covered.

In recent years, cooperative exploration has received considerable attention. Different strategies have been proposed and discussed based on their effectiveness, exploration time requirements, etc. Specifically, mapping is of critical importance during cooperative exploration. Mapping of an unknown zone autonomously by mobile robots and simultaneous localization are parts of Simultaneous Localization and Mapping (SLAM) problem. Basically, SLAM is the process of building a map of an unknown zone from a sequence of noisy landmark measurements obtained from sensors [1]. While exploration with multiple robots brings several complexities, it offers numerous advantages [19,30–32]. The major advantages of multi-robot exploration systems can be outlined as follows:

* Corresponding author. Tel.: +90 284 2240283.
E-mail addresses: gurkantuna@trakya.edu.tr (G. Tuna), gulez@yildiz.edu.tr (K. Gulez), cagri.gungor@bahcesehir.edu.tr (V.C. Gungor).

1570-8705/$ - see front matter © 2012 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.adhoc.2012.07.011

- Multi-robot systems can explore more area in less time than a single-robot based system.
- Mutual observation by several robots can improve localization.
- With a proper implementation and coordination strategy, multi-robot systems can be more robust than single-robot based systems due to redundancy in case of hardware failures.
- If heterogeneous robots are preferred, certain tasks can be assigned to different robots depending on requirements.
- In multi-robot systems, each robot can exchange its sensor data with other robots. In this way, robots can sense the environment from multiple viewpoints. If properly implemented, this can lead to maps with higher accuracies.
- During a mission, multiple robots may collaborate to manipulate objects or help each other overcome obstacles.

Although cooperative exploration offers exploring an unknown zone more quickly and robustly than single-robot case, in addition to communication a strategy is required to coordinate multiple robots [26]. For cooperative exploration, special attention needs to be given to *map merging*, building a consistent model of an unknown zone with sensor data collected from different agents. For map merging, two strategies exist. In the centralized map merging strategy, each robot sends its observations to a central agent which is responsible for map building [2,3]. The central agent builds a common global map. In the distributed map merging strategy, each robot independently builds a local submap. Local submaps are fused into global map periodically [4–6]. Both map merging strategies require a stable communication model. Though exploration strategies have been extensively investigated, the effects of communication models on the performance of robotic exploration have been neglected. Only connectivity related issues have been taken into account while designing strategies for robotic exploration.

In this study, we explain different cooperative exploration strategies and compare them based on their exploration performances and processing time requirements. Simulation studies were performed to compare three different exploration strategies: frontier-based exploration, market-driven exploration, and role-based exploration. Different from existing studies, this paper addresses the relation between exploration performance and CPU processing power. Hence, the tests were performed on systems with CPUs having different processing powers. To show the effect of communication models on the cooperative exploration, we used three different communication models: Static Circle, Line of Sight and Propagation Model. We also included two more simulations to show the effect of the number of robots on the time to cover the whole zone. To perform simulations, we developed our own simulator. And we are planning to make the source codes of our application publicly available, which can help the research community further improve the performance of multi-robot exploration strategies and models.

The paper is organized as follows. Related works are explained in Section 2. Exploration strategies and communication models used in our applications are explained in Section 3. The simulation studies with cooperative exploration strategies application are explained in Section 4. Future research directions are given in Section 5. And, conclusions of the paper are given in Section 6.

2. Related work

Autonomous exploration itself is a highly sophisticated work. Cooperative exploration of an unknown zone requires gathering information from each robot and coordinating the robots to explore the zone completely. The performance of the mission depends mainly on the path length, total distance travelled by a group of robots to cover an unknown zone. Hence, a strategy is required to coordinate multiple robots during the mission in order to improve efficiency [27]. Cooperative exploration strategies can be broadly classified into two categories: model-based and behavior-based [8]. Model-based exploration strategies guarantee the exploration of an unknown zone but high computational power is required to implement these strategies. Different from model-based exploration strategies, during a behavior-based exploration mission, each robot decides an action quickly. This strategy cannot guarantee the exploration of an unknown zone completely.

Although various exploration strategies have been proposed, their common requirement is communication. Early exploration strategies were based on the principle of keeping the robots within communication range [9]. Some authors propose market-driven exploration strategies, in which an exploration task is divided into subtasks and robots place bids on these subtasks [19]. In these strategies, goal point selection is an import decision and aims to select unexplored regions for mobile robots to visit. Here, bids are based on specific values such as travelling cost to a particular location and expected information gain. Though market-driven exploration approach does not rely on perfect communication, and is still functional with zero communication, when communication strength is factored into these bids, robots avoid going beyond communication range [12,19].

Goal point selection strategies are Random, Greedy exploration and Space division by quad-tree. Random goal point selection is the simplest strategy, in which goal points are chosen randomly. If the area that surrounds the goal point has already been visited, it is discarded. This strategy has been effectively used in practice [19]. In the greedy exploration strategy, a goal point in the closest unexplored region is selected as a candidate and this strategy is effective for single robot exploration [19].

Frontier-based exploration is an exploration strategy with a key idea of gaining the most new information by moving to the boundary between open space and uncharted territory. In this strategy, robots share perceptual information while maintaining separate global maps. Even if robots are dispersed to the maximum extent which their communication ranges allow, unexplored spaces may remain. A solution to deal with the limited communication during multiple robot exploration is rendezvous points in which shared knowledge is communicated [16,17]. Clustering approach was proposed by some authors, in which groups of robots stay close together while exploring the zone [13–15]. Another strategy using the same principle is role-based exploration. In this strategy, robots are categorized into two groups: explorers and relays [12]. In this strategy, while explorers explore an unknown zone by using frontier-based exploration strategy and communicate their findings to a relay in rendezvous points, relays maintain the connection between the base station responsible for the mission and the explorers. Role-based exploration strategy offers a solution to connectivity related issues in large environments at the expense of additional robots responsible for messaging. Though most exploration strategies are successful in maintaining connectivity during an exploration mission, their performances are limited due to the constraint of having to keep the robots within communication range [12,28].

3. Exploration strategies and communication models

There are different strategies in the literature proposed for robotic exploration. In this paper, we specifically investigate and compare the performances of frontier-based exploration, role-based exploration, and market-driven exploration strategies.

- *Frontier-based exploration*: In the frontier-based exploration, a group of mobile robots are directed to regions on the boundary between unexplored space and the space known to be open [23,24]. In this strategy, each robot builds and maintains its global map. During the mission, each robot shares perceptual information with other robots. Each robot decides itself about where to explore [23]. This strategy has the advantage of likely keeping robots in communication range limits by preventing them from going beyond communication range during exploration. In real implementations of cooperative exploration strategies, full communication among robots is impractical in reality [11,16,18].
- *Role-based exploration*: In this strategy, robots are categorized into two groups: relays and explorers [12].
  - *Relays*: In role-based exploration, relays maintain the connection between explorers and a base station responsible for controlling the mission. If relays discover information about the zone while relaying, this information is added to the exploration team knowledge.
  - *Explorers*: In role-based exploration, explorers explore an unknown zone and communicate their findings to a relay in previously agreed rendezvous points. Explorers use frontier-based exploration strategy.
- *Market-driven exploration*: To address the challenges of multi-robot exploration, some authors propose exploration strategies based on the principle of dividing an exploration task into subtasks [19]. This strategy is called market-driven exploration. In these strategies, robots place bids on the subtasks and when communication strength is factored into the bids, robots avoid going beyond communication range.

We have some assumptions which are essentially common in most robotic exploration strategies. In our application, regardless of the exploration strategy chosen, each robot is assumed to have a range bearing sensor such as a laser range finder and an ultrasonic range finder. Robots have no priori maps and prior knowledge of the zone. A control center, which is responsible for controlling the overall mission and coordinating robots, is located at the entry point. Robots start exploring in the same location and each robot knows the starting point of the others. During the exploration, robots only know what they have observed and what other robots communicate to them. Robots can communicate with each other when they are within communication range of one another. Communication range is specified by the communication model. Regardless of strategy and communication model, the flow of each step of robot movement and observation is explained as follows:

Robot movement and observation.

```
for each robot do
next_location=relocate(robot);
if location_check(next_location)==true then
move(robot,next_location);
observation_data=simulateobservation
(robot,next_location,rangenoise,anglenoise);
sendobservation(robot,observation_data);
end if
end
```

The flow of each step of communication algorithm is explained as follows:

Communication.

```
for each robot do
for each crobot do
if crobot!=robot and rangecheck(robot,
crobot)==true then
communicatedata(robot,crobot);
end if
end
end
```

As a rule of thumb, some set of measurable performance values are required to evaluate and compare different exploration strategies. Four common metrics of robotic exploration algorithms are as follows:

1. Maximizing the total area explored.
2. Maximizing the global map created at the control center.
3. Minimizing the total area of recurrent visits.
4. Keeping robots in the range of the control center as much as possible.

Please cite this article in press as: G. Tuna et al., The effects of exploration strategies and communication models on the performance of cooperative exploration, Ad Hoc Netw. (2012), http://dx.doi.org/10.1016/j.adhoc.2012.07.011

We only use the first metric in our application since our main goal is to compare the times required by each exploration strategy and compare the processing times of these strategies. In addition to the exploration strategies, we implemented three different communication models into our application. The details of these models are as follows:

- *Static circle communication model*: Any robot X within a set communication radius of robot Y may communicate with that robot, regardless of the obstacles in the circle. Communication range is defined by the wireless network hardware specifications.
- *Line of sight communication model*: Any two robots may communicate if there are no obstacles on the direct line of sight between them. In this model called Line of Sight (LOS), we use the propagation characteristic of high-frequency radio. Any obstruction between the transmitting antenna and the receiving antenna blocks the signal at high frequencies and in lower levels of the atmosphere.

Assuming both the transmitting antenna and the receiving antenna have height h above the earth's surface as shown in Fig. 1, maximum line-of-sight distance is calculated by the following equation:

$$d_{\mathrm{LOS}} = 2\sqrt{2hR + h^2} \cong 2\sqrt{2Rh} \quad (1)$$

where R is the earth's radius which is 6.38×10^6 m.

- *Propagation model*: This is a realistic model in which thickness of obstacles in the environment is taken into account and communication range is calculated according to the model of Bahl and Padmanabhan [22]. This model uses a standard path loss model with a wall attenuation factor, and is a common model in widely used simulation environments including the popular Usarsim [29]. The model includes an attenuation factor called Wall Attenuation Factor (WAF) for building floors and it considers the effects of obstacles like walls between the transmitter and the receiver. Note that the attenuation value depends on the material and thickness of the wall [36]. WAF model is described by the following equation:

$$P(d) = P(d_o) - 10n\log\left(\frac{d}{d_o}\right) - \begin{cases} nW * \mathrm{WAF} & nW < C \\ C * \mathrm{WAF} & nW \geqslant C \end{cases} \quad (2)$$

where n indicates the rate at which the path loss increases with distance, $P(d_o)$ represents the signal power in dBm at some reference distance d_o, and d represents the separation distance from the transmitter to the receiver. C is the maximum number of obstructions up to which the attenuation factor makes a difference, nW is the number of obstructions between the transmitter and the receiver, and WAF is the wall attenuation factor. Generally, n and WAF are derived empirically. The WAF depends on building layout and construction material and WAF values of 3.1–5 are common for indoor office environments [22,35,37]. The value of $P(d_o)$ can either be obtained from the wireless network hardware specifications or derived empirically [22].

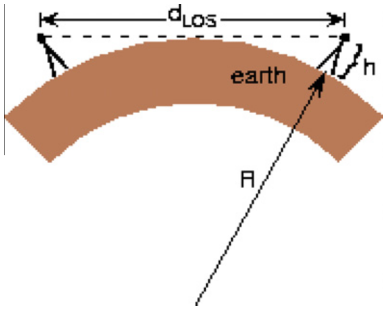


Fig. 1. LOS communication model.

Fig. 2. First environment to be explored.

4. Simulation studies

Simulations were performed by using the application, *Cooperative Exploration Strategies*, developed with NetBeans IDE 6.9.1 in Java. NetBeans IDE is a free, open-source Integrated Development Environment (IDE) and in this platform, different programming languages can be used [21]. Source codes of MRESim [12] simulator were used to develop our own Java based simulator. MRESim allows the simulation of two exploration strategies: frontier-based exploration and role-based exploration. Different from MRESim, our application allows integrating motion noise and observation noise to measurements. The application allows loading custom maps, setting the number of robots and their properties, defining the sensor measurement range of each robot, applying measurement and sensor noise and selecting different exploration strategies and communication models. The application also allows log analyzing. In this way the user can see relative performances of each strategy. For example, after 60 s, how many percent of the map has been explored by the exploration strategy?

The application runs in a single thread. In each cycle of an exploration, robots make a decision on where they want to move next. Then, they are provided with range data, and messages are passed between any two robots within range of one another. All relevant data is logged, and the user interface of the application is updated. The main limitation related to the performance of the application is that real

Fig. 3. Second environment to be explored.

robots run multi-thread processes and do not always move the same distance in each time segment.

Environments are flat two dimensional binary Portable Network Graphics (PNG) files, with each cell being either free or an obstacle. In every step of a robot, the simulator provides it with sensor data at its new location. It acts the same as real laser scanners, and provides the robot with an array of 181 measurements (180° field of view and 1° resolution). Then, the robot detects obstacles where two points are sufficiently close to each other and below the sensor's range limit. And finally the robot maintains this data in an occupancy grid map. Whenever robots are within communication range of each other, they exchange their occupancy grids. All robots use a simple A* path planning algorithm [10].

In the simulation studies, two notebooks and a netbook with different specifications were used to show the effect of CPU power on the processing time of exploration algorithms. Here, the processing time refers to the total time required to explore a previously unknown environment. Table 1 shows the hardware specifications of the notebooks and the netbook used during tests. Windows 7 Home Basic operating system (OS) runs on System 1 and System 2, and Windows 7 Starter OS runs on System 3. Table 2 shows the CPU power rankings of these three systems and these tests were performed by independent organizations. In CPU Benchmark tests, higher grade means higher performance. The data in this table is used to check the consistency of the simulation results with the CPU power rankings.

The exploration strategies used during the simulations are frontier-based exploration, role-based exploration, and market-driven exploration with Random goal point selection. The strategies were tested with three different communication models: Static Circle, Line of Sight and Propagation Model. Three different exploration strategies were compared based on their exploration strategies and processing time requirements. They were simulated for the same environment. The environments to be explored are shown in Figs. 2, 3 and 5. They are 800×600 pixels PNG images. A set of parameters are used to represent the environment in the real world. For instance, 800×600 pixel map can represent an 80×60 m area or an 8×6 m area. If we use an 800×600 pixel map to represent an 80×60 m area, then each cell represents 10 cm $\times$ 10 cm. Though the set of parameters are hypothetical, it is useful for performance evaluations. White regions represent the areas to be explored. The parameters related to the first simulation are as follows:

- Number of robots: 4.
- Number of explorers: 2.
- Measurement range: 30 m.
- Communication range: 100 m.
- Range noise of sensors: 0.05 m.
- Angle noise of sensors: 0.5°.

The application specific parameters of propagation model-based communication were defined empirically

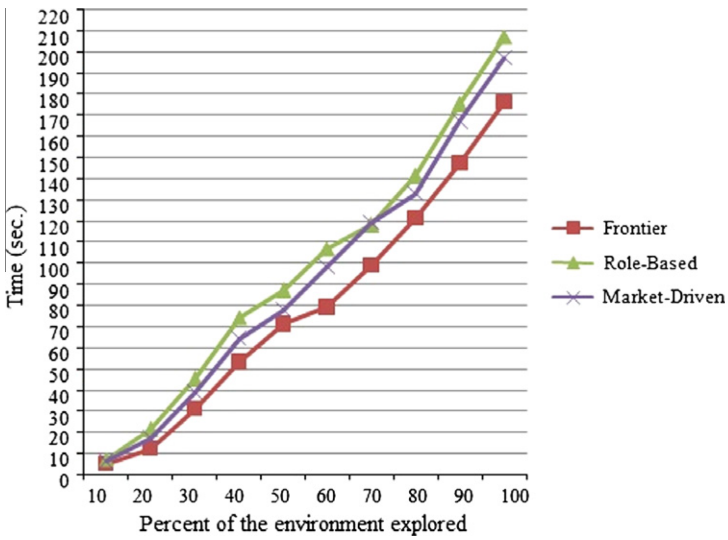


Fig. 4. Exploration rates.

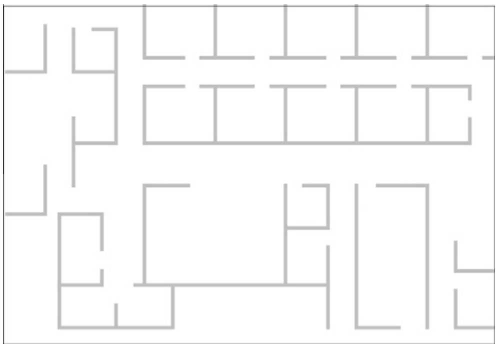


Fig. 5. Third environment to be explored.

based on [22]. The parameters related to propagation model are as follows:

- Reference signal: 50.
- Reference distance d_o: 1 m.
- WAF: 20 dBm.
- Maximum number of walls C: 4.
- Path loss factor: 1.
- Cutoff value: −92 dBm.

For the first environment in Fig. 2 and the second environment in Fig. 3, each exploration strategy was run multiple times for each communication model. The averages of these simulations are shown in Tables 3 and 4. Fig. 4 shows the exploration rates of the strategies (using propagation model as the communication model) performed on the System 1. Considering Tables 1 and 2, System 1 is the fastest system and System 3 is the slowest one. Considering the processing times of Frontier Exploration strategy with static circle communication model, we can conclude that the processing time requirement of the exploration strategy in a system is consistent with processing power of the system. The same case exists for Role Based Exploration and Market Driven Exploration strategies with different communication models.

However, considering the independent performance test results in Table 2 and the simulation results in Table 3, we can conclude that the increase rate in CPU powers is not in parallel with the decrease rate in processing times. System 1's CPU is Intel Core I5 460M and 3DMark06 CPU test result is 2923. System 2's CPU is Intel Core2 Duo T5450 and 3DMark06 CPU test result is 1381. Processing time of Frontier Exploration strategy with static circle communication model on System 1 is 195 s, and processing time of Frontier Exploration strategy with static circle communication model on System 2 is 382 s. Since the memory usages of all three systems are below 100% during the tests and low graphics performance is enough for the simulations, we can conclude that CPU processing powers of the systems are not used efficiently. This is understandable for most software development platforms. Independent organizations use different type of applications utilizing 100% of CPU's processing power to benchmark CPUs but this is not the aim of development platforms.

We performed another two sets of simulations to show the number of robots on the time requirement of exploration using the environment in Fig. 5. Each exploration strategy and communication model pair was run multiple times. The results are shown in Tables 5 and 6. The values in these tables are the averages of the simulations conducted for each pair. The parameters related to the second simulation are as follows:

- Number of robots: 4–8.
- Number of explorers: 2–4.
- Measurement range: 100 m.
- Communication range: 100 m.
- Range noise of sensors: 0.05 m.
- Angle noise of sensors: 0.5°.

The parameters related to propagation model are the same as the ones in the first simulation.

In addition, another 20 sets of simulations with the environment shown in Fig. 2 were conducted on each system for Static Circle (SC), Line of Sight (LOS) and Propagation Model (PM) communication models to calculate CPU and memory usages of each strategy. The CPU and memory monitoring tool used is JConsole. CPU and memory usage data related to the Frontier Exploration (FE), Role Based Exploration (RBE) and Market Driven Exploration (MDE) were collected with a polling interval of 5 s and logged. The averages of 20 simulation sets were calculated using

Table 1
Specifications of the notebooks/netbooks used to perform simulations.

System ID	CPU	Main memory	Graphics card and memory
1	Intel Core I5 460 M	3 GB	ATI HD5650 – 1 GB
2	Intel Core2 Duo T5450	2 GB	ATI HD2600 – 256 MB
3	Intel Atom N455	2 GB	Intel GMA3150 – 256 MB (shared)

Table 2
CPU frequencies and performance comparisons [20].

CPU	CPU frequency (MHz)	Test 1 – 3DMark06 CPU	Test 2 – Cinebench R10 single	Test 3 – Cinebench R10 multi
Intel Core I5 460M	2530–2800	2923	3096	7022
Intel Core2 Duo T5450	1660	1381	1504	2827
Intel Atom N455	1660	477	547	856

Table 3
Processing times of different exploration strategies (first environment).

Strategy and communication model	System 1 (s)	System 2 (s)	System 3 (s)
Frontier Exploration – communication model: static circle	195	382	891
Frontier Exploration – communication model: line of sight	189	369	849
Frontier Exploration – communication model: propagation model	176	355	778
Role Based Exploration – communication model: static circle	251	483	1163
Role Based Exploration – communication model: line of sight	306	552	1271
Role Based Exploration – communication model: propagation model	207	391	959
Market Driven Exploration – communication model: static circle	234	421	1024
Market Driven Exploration – communication model: line of sight	229	407	981
Market Driven Exploration – communication model: propagation model	197	386	903

Table 4
Processing times of different exploration strategies (second environment).

Strategy and communication model	System 1 (s)	System 2 (s)	System 3 (s)
Frontier Exploration – communication model: static circle	201	397	919
Frontier Exploration – communication model: line of sight	199	386	865
Frontier Exploration – communication model: propagation model	183	362	801
Role Based Exploration – communication model: static circle	262	490	1186
Role Based Exploration – communication model: line of sight	313	561	1290
Role Based Exploration – communication model: propagation model	212	396	979
Market Driven Exploration – communication model: static circle	241	429	1041
Market Driven Exploration – communication model: line of sight	240	420	998
Market Driven Exploration – communication model: propagation model	209	395	927

Table 5
Processing times of different exploration strategies (third environment, using two explorers).

Strategy and communication model	System 1 (s)	System 2 (s)	System 3 (s)
Frontier Exploration – communication model: static circle	213	397	1126
Frontier Exploration – communication model: line of sight	159	380	1105
Frontier Exploration – communication model: propagation model	141	354	994
Role Based Exploration – communication model: static circle	292	508	1480
Role Based Exploration – communication model: line of sight	247	531	1484
Role Based Exploration – communication model: propagation model	205	498	1311
Market Driven Exploration – communication model: static circle	237	493	1288
Market Driven Exploration – communication model: line of sight	211	394	1173
Market Driven Exploration – communication model: propagation model	203	367	1094

Table 6
Processing times of different exploration strategies (third environment, using four explorers).

Strategy and communication model	System 1 (s)	System 2 (s)	System 3 (s)
Frontier Exploration – communication model: static circle	150	305	923
Frontier Exploration – communication model: line of sight	114	294	892
Frontier Exploration – communication model: propagation model	109	253	857
Role Based Exploration – communication model: static circle	220	421	1017
Role Based Exploration – communication model: line of sight	189	443	1003
Role Based Exploration – communication model: propagation model	153	397	913
Market Driven Exploration – communication model: static circle	170	374	945
Market Driven Exploration – communication model: line of sight	154	302	856
Market Driven Exploration – communication model: propagation model	145	263	837

the logs. The results can be seen in Figs. 6 and 7. As it is well-known, there is no consistent relationship between CPU and memory usage, since the CPU usage depends on how much processing is required by the application, and the memory usage depends on how much space the application needs to hold while it's running. Considering the results of the simulations, we can conclude that RBE and MDE are more processor intensive compared to FE due to the inherent mechanisms of RBE and MDE such as role assignment and bidding on subtasks.

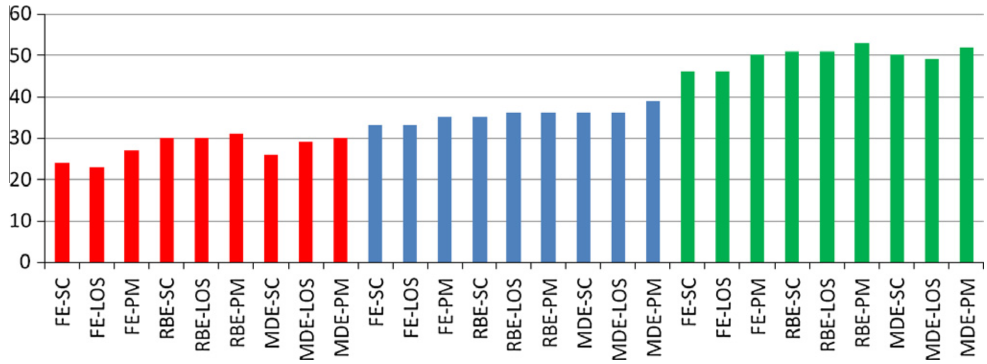


Fig. 6. Comparative average CPU usages (%). Red bars represent the values obtained on System 1, blue bars represent the values obtained on System 2, and green bars represent the values obtained on System 3. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

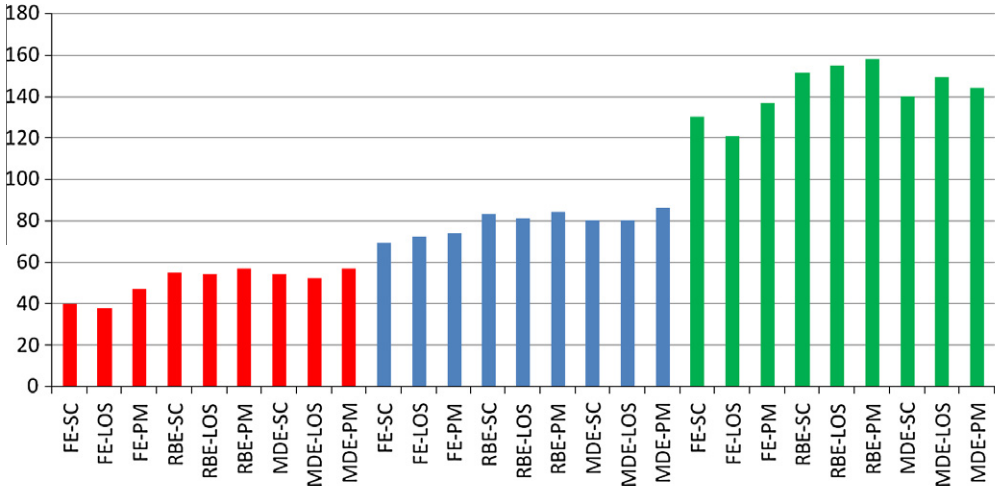


Fig. 7. Comparative average main memory usages (MB). Red bars represent the values obtained on System 1, blue bars represent the values obtained on System 2, and green bars represent the values obtained on System 3. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Overall, the performance metric of our application is the total time spent for the full exploration since the aim of our simulations is to show the total processing requirements of each individual exploration strategy. Fig. 4 shows that Frontier Exploration is the fastest strategy. Role-based exploration is the slowest strategy. Similar results were obtained by Hoog et al. [12] and they proved that in most cases frontier-based exploration is faster than role-based exploration due to the performance criterions used by role-based exploration. However, for a scenario where the main goal is both to explore as much of the environment as possible and to maintain the robot team in a communication network as tight as possible, role-based exploration can be preferred to Frontier Exploration [12]. Market-driven exploration is slower than frontier-based exploration since the algorithm used in market-driven exploration is based on the distance based costs to minimize distance traveled. If time-based costs are used it may explore faster. Though frontier-based exploration may seem the best strategy for multiple robot exploration, the amount of coordination in this strategy is limited. Frontier-based exploration may also cause a large amount of repeated coverage due to lack of communication, in addition to the limitation of not taking the full advantage of the number of robots available [19].

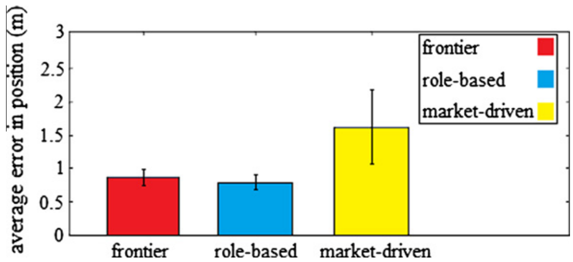


Fig. 8. Average error in position.

From our experiments, we see that realistic communication models such as propagation model can be used

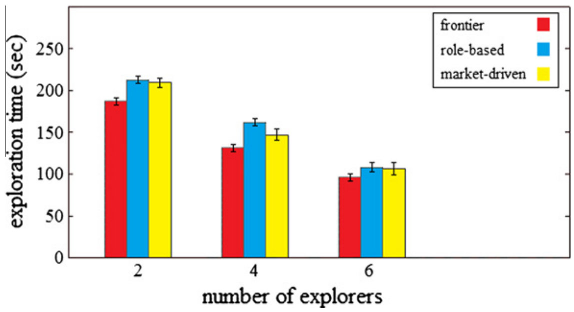


Fig. 9. Average time needed by robots to explore the environment.

effectively while designing cooperative exploration strategies. We believe that keeping in mind the conditions found in real world environments is an important design criterion. We can also conclude that increasing the number of robots reduces the total time required for the exploration mission at the expense of higher CPU load and memory usage. This conclusion can be easily seen by checking the System 3 columns of Tables 5 and 6. Since the processing power of Atom CPU's is limited, increasing the number of robots creates additional processing load on the CPU, which results in reducing the total processing time slightly.

4.1. Quantitative analysis

To evaluate the performance of the exploration strategies quantitatively, we performed a series of simulations on System 1 by using the environment shown in Fig. 3. We performed 30 simulations with each exploration strategy. We preferred propagation model for communication since it is more realistic than the other communication models in this study. To compare the average errors in the poses of robots in each strategy, after the end of each simulation, we measured the average error in relative distances. To calculate the average error, we compared the real ground truth values with the estimated trajectory of each robot. Finally, we calculated the average of 30 simulations. The results are shown in Fig. 8. In this figure, the error bars indicate 95% confidence intervals. As can be seen from the figure, the average position error in a role-based exploration mission is less than the other strategies. This is understandable because in this strategy, it is aimed to cover as much of the environment as possible in addition to strictly maintaining the communication between the robots.

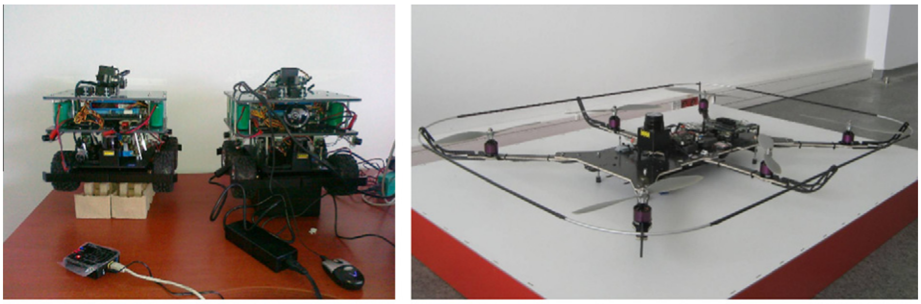
To compare the average time needed by robots to explore a given environment, we performed a series of 90 simulations for each exploration strategy. We preferred propagation model for communication in these simulations. Each exploration strategy was conducted 30 times for three different cases: 2 explorers, 4 explorers, and 6 explorers. In each comparison of the three strategies, the robots were started at the same location. Then, we evaluated the average number of time steps the robots needed to complete the mission. The results of these simulations are shown in Fig. 9. In this figure, the error bars indicate 95% confidence intervals. As can be seen from the figure, the Frontier Exploration requires less time than the other strategies.

5. Future research directions

To exploit the full performance of the cooperative exploration strategies, each robot computes its own path individually given its local map. In contrast to practical implementations of cooperative exploration where robots generally run multi-thread processes, the current version of the simulator runs locally in a single thread and cannot exploit the full performance of these strategies since it places all processing load on a single system. In the next version of the simulator and in our field tests, we are going to implement the distributed processing approach with heterogeneous robots. We have two four-wheeled (4WD) robots and an unmanned aerial vehicle (UAV) in our lab environment as shown in Fig. 10. They all run Robot Operating System (ROS) on Ubuntu. ROS provides device drivers, hardware abstraction layers, message-passing functionalities, package management capabilities, libraries and tools to help developers create robot application [33].

In order to quantify the results of our field tests, we are going to use a simple metric defined in [34]. The metric used to calculate the quality of exploration is inversely proportional to the combined distance traveled by each robot, and directly proportional to the total area covered. The metric is calculated by using the following equation:

$$Q = \frac{A}{\sum_{i=1}^{n} d_i} \tag{3}$$



(a) (b)

Fig. 10. (a) 4WD Autonomous robots. (b) Hexarotor helicopter.

Please cite this article in press as: G. Tuna et al., The effects of exploration strategies and communication models on the performance of cooperative exploration, Ad Hoc Netw. (2012), http://dx.doi.org/10.1016/j.adhoc.2012.07.011

Where d_i is the distance traveled by the robot i, A is the total area covered, and n is the total number of robots. We are also planning to integrate this metric into the next version of the simulator.

In [18], a new approach to model bandwidth-constrained relay node placement is proposed. It is modeled as a new variation of the Steiner Minimum Tree Problem with Minimum Number of Steiner Points and bounded edge length (SMT–MSP). In the next version of the simulator, we are going to integrate this exploration strategy into our comparisons.

In the current version, the simulator tries to maximize the total area explored. In the next version of the simulator, we are planning to integrate common performance goals, such as maximizing the global map created at the control center, minimizing the total area of recurrent visits and keeping robots in the range of the control center as much as possible.

6. Conclusion

In this study, different exploration strategies for cooperative exploration are explained and the strategies are compared based on their exploration performances and processing time requirements. We developed our own application to evaluate the performances of different exploration strategies. The tests were performed on systems with CPUs having different processing powers.

Based on extensive simulation experiments, we observed that there exists a strong correlation between the processing time requirements and the computational complexities of exploration strategies. Another correlation exists between the processing time requirements and the CPU processing power. Considering the processing power increase of CPUs in recent years, we expect that it will be relatively easier to implement the exploration strategies with high computational complexities. The simulation experiments also allowed us to compare the effect of communication model on the performance of multi-robot exploration strategies. The results show that the propagation model improves the performance of multi-robot exploration and reduces the time required to complete an exploration mission. The simulations also show that increasing the number of robots reduces the total time required for exploration though it increases CPU and memory usage.

While our simulation results provide valuable insights into the advantages of cooperative exploration strategies and their processing requirements, these are only first steps. Future work includes implementing cooperative exploration strategies on autonomous robot platforms, i.e., Corobot [7].

Acknowledgement

This research has been supported by Yildiz Technical University Scientific Research Projects Coordination Department. Project Number: 2010-04-02-ODAP01.

References

[1] G. Dissanayake, P. Newman, S. Clark, H.F. Durrant-Whyte, M. Csorba, A solution to the simultaneous localization and map building (SLAM) problem, IEEE Transactions on Robotics and Automation 17 (3) (2001) 229–241.
[2] J.W. Fenwick, P.M. Newman, J.J. Leonard, Cooperative concurrent mapping and localization, in: Proceedings of the 2002 IEEE International Conference on Robotics and Automation, 2002, pp. 1810–1817.
[3] S.B. Williams, G. Dissanayake, H. Durrant-Whyte, Towards multi-vehicle simultaneous localisation and mapping, in: Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), 2002, pp. 2743–2748.
[4] T. Tao, Y. Huang, J. Yuan, F. Sun, Multi-robot cooperative map building in unknown environment considering estimation uncertainty, in: Proceedings of the 2008 Chinese Control and Decision Conference (CCDC 2008), 2008. pp. 2896–2901.
[5] S.B. Williams, Efficient Solutions to Autonomous Mapping and Navigation Problems, Ph.D. Dissertation, University of Sydney, 2001.
[6] D. Rodriguez-Losada, F. Matia, A. Jimenez, Local maps fusion for real time multirobot indoor simultaneous localization and mapping, in: Proceedings of the 2004 IEEE International Conference on Robotics and Automation, vol. 2, 2004, pp. 1308–1313.
[7] http://robotics.coroware.com/.
[8] C. Trevai, N. Fujii, J. Ota, T. Arai, Multiple mobile robot exploration and patrol strategy using a self-organizing planner based on a reaction–diffusion equation on a graph, IEEE Transactions on Robotics and Automation 20 (1) (2008).
[9] R. Arkin, J. Diaz, Line-of-sight constrained exploration for reactive multiagent robotic teams, in: Proceedings of the 7th International Workshop on Advanced Motion Control, 2002, pp. 455–461.
[10] N.J. Nilsson, Artificial Intelligence: A New Synthesis, Morgan Kaufman Publishers, 1998.
[11] W. Sheng, Q. Yang, J. Tan, N. Xi, Distributed multi-robot coordination in area exploration, Robotics and Autonomous Systems 54 (12) (2006) 945–955.
[12] J. De Hoog, S. Cameron, A. Visser, Role-based autonomous multi-robot exploration, in: Proceedings of the International Conference on Advanced Cognitive Technologies and Applications (COGNITIVE), 2009, pp. 482–487.
[13] W. Burgard, M. Moors, C. Stachniss, F. Schneider, Coordinated multi-robot exploration, IEEE Transactions on Robotics 21 (3) (2005) 376–378.
[14] D. Fox, J. Ko, K. Konolige, B. Limketkai, D. Schulz, B. Stewart, Distributed multirobot exploration and mapping, Proceedings of the IEEE 94 (7) (2006) 1325–1339.
[15] M.N. Rooker, A. Birk, Multi-robot exploration under the constraints of wireless networking, Control Engineering Practice 15 (4) (2007) 435–445.
[16] J. Yuan, Y. Huang, T. Tao, F. Sun, A cooperative approach for multi-robot area exploration, in: Proceedings of the 2010 IEEE International Conference on Intelligent Robots and Systems, 2010, pp. 1390–1395.
[17] N. Roy, G. Dudek, Collaborative robot exploration and rendezvous algorithms performance bounds and observations, Journal of Autonomous Robots 11 (2) (2001) 117–136.
[18] Y. Pei, M.W. Mutka, N. Xi, Coordinated multi-robot real-time exploration with connectivity and bandwidth awareness, in: Proceedings of the 2010 IEEE International Conference on Robotics and Automation, 2010, pp. 5460–5465.
[19] R. Zlot, A. Stentz, M.B. Dias, S. Thayer, Multi-robot exploration controlled by a market economy, in: Proceedings of the International Conference on Robotics and Automation (ICRA'02), vol. 3, 2002, pp. 3016–3023.
[20] http://www.notebookcheck.net/Mobile-Processors-Benchmarklist.2436.0.html.
[21] http://netbeans.org/.
[22] P. Bahl, V.N. Padmanabhan, RADAR: an in-building RF-based user location and tracking system, in: Proceedings of the INFOCOM 2000. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies, 2000, pp. 775–784.
[23] B. Yamauchi, Frontier-based exploration using multiple robots, in: Proceedings of the Second International Conference on Autonomous Agents, 1998, pp. 47–53.
[24] K.M. Wurm, C. Stachniss, W. Burgard, Coordinated multi-robot exploration using a segmentation of the environment, in: Proceedings of the 2008 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2008, pp. 1160–1165.

[25] A. Marjovi, J.G. Nunes, L. Marques, A. Almeida, Multi-robot exploration and fire searching, in: Proceedings of the 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2009, pp. 1929–1934.

[26] A.D. Haumann, K.D. Listmann, V. Willert, DisCoverage: a new paradigm for multi-robot exploration, in: Proceedings of the 2010 IEEE International Conference on Robotics and Automation, 2010, pp. 929–934.

[27] W. Wu, F. Zhang, A switching strategy for robust cooperative exploration, in: Proceedings of the 49th IEEE Conference on Decision and Control, 2010, pp. 5493–5498.

[28] J. Yuan, Y. Huang, T. Tao, F. Sun, A cooperative approach for multi-robot area exploration, in: Proceedings of the 2010 IEEE/RSJ Intelligent Robots and Systems, 2010, pp. 1390–1395.

[29] S. Carpin, M. Lewis, J. Wang, S. Balakirsky, C. Scrapper, Usarsim: a robot simulator for research and education, in: Proceedings of the 2007 IEEE International Conference on Robotics and Automation, April 2007, pp.1400–1405.

[30] Y.U. Cao, A.S. Fukunaga, A. Kahng, Cooperative mobile robotics: antecedents and directions, Autonomous Robots 4 (1) (1997) 7–27.

[31] I. Rekleitis, G. Dudek, E. Milios, Multi-robot exploration of an unknown environment, efficiently reducing the odometry error, in: Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI), 1997, pp. 1340–1345.

[32] R. Grabowski, L.E. Navarro-Serment, C.J.J. Paredis, P.K. Khosla, Heterogeneous teams of modular robots for mapping and exploration, Autonomous Robots 8 (3) (2000) 293–308.

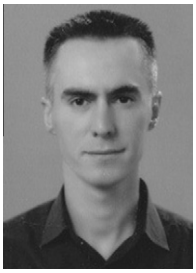
[33] http://www.ros.org/wiki/.

[34] R. Zlot, A. Stentz, M.B. Dias, M. Bernardine, S. Thayer, Market-driven multi-robot exploration, Robotics Institute Paper 156 (2002). <http://repository.cmu.edu/robotics/156>.

[35] M. Lot, I. Forkel, A multi-wall-and-floor model for indoor radio propagation, in: Proceedings of the IEEE Vehicular Technology Conference, vol. 1, 2001, pp. 464–468.

[36] A.R. Sandeep, Y. Shreyas, Shivam Seth, Rajat Agarwal, G. Sadashivappa, Wireless network visualization and indoor empirical propagation model for a campus WI-FI network, World Academy of Science, Engineering and Technology 42 (2008) 730–734.

[37] S. Zvanovec, P. Pechac, M. Klepala, Wireless LAN networks design: site survey or propagation modeling?, Radioengineering 12 (4) (2003) 42–49



Gurkan Tuna received his B.S. degree in computer engineering from Yildiz Technical University, Istanbul, Turkey, in 1998 and his M.S. degree in computer engineering from Trakya University, Edirne, Turkey, in 2009. He received his Ph.D. degree in control and automation engineering from Yildiz Technical University, Istanbul, Turkey, in 2012. His current research interests include mobile robotics, multi-robot simultaneous localization and mapping, cooperative exploration strategies, clustering, wireless ad hoc and sensor networks, and IP networks.

Kayhan Gulez received B.S., M.S., and Ph.D degrees in Electrical Engineering from Yildiz Technical University, Istanbul, Turkey, in 1992, 1995, and 1999 respectively. He joined and worked in the Department of Electrical Engineering at Yildiz Technical University between 1997 and 2010. He worked as a Research Associate in a JSPS project and other short-term projects in Keio University and in Tokyo Metropolitan Institute of Technology between 1999 and 2002. Currently, he is working for the Department of Control and Automation Engineering as Associate Professor at Yildiz Technical University, since 2010. His major research interests are Electrical Vehicle and Unmanned Air Vehicle Applications, Neural Networks on Control Systems, EMC and EMI Control Methods, Active, Passive and EMI Filter Design Methods and Applications for EMI Noise and Harmonic Problems on which he has over 200 scientific papers and technical reports in various journals and conference proceedings. He has also nine science grand awards between 1998 and 2009 and The Superior Success Science Award in 2009 in Yildiz Technical University in Istanbul, and best paper awards from SCI'2001, M&S'2001 and TOK 2009 conferences.

V. Cagri Gungor received his B.S. and M.S. degrees in electrical and electronics engineering from Middle East Technical University, Ankara, Turkey, in 2001 and 2003, respectively. He received his Ph.D. degree in electrical and computer engineering from the Broadband and Wireless Networking Laboratory, Georgia Institute of Technology, Atlanta, GA, USA, in 2007. Currently, he is an Assistant Professor and Co-Director of the Computer Networks and Mobile Communications Lab at the Department of Computer Engineering, Bahcesehir University, Istanbul, Turkey. Before joining to Bahcesehir University, he was working at Eaton Corporation, Innovation Center, WI, USA as a ProjectLeader. His current research interests are in smart grid communications, next-generation wireless networks, wireless ad hoc and sensor networks, cognitive radio networks, and IP networks. He has authored several papers in refereed journals and international conference proceedings, and has been serving as a reviewer and program committee member to numerous journals and conferences in these areas. He is also the co-recipient of the IEEE Transactions on Industrial Informatics 2012 Best Paper Award, IEEE ISCN 2006 Best Paper Award, the European Union FP7 Marie Curie IRG Award in 2009, and the San-Tez Project Awards issued by Alcatel-Lucent and the Turkish Ministry of Science, Industry and Technology in 2010. He is also the Principal Investigator of the Smart Grid Communications R&D project funded by Turk Telekom and European Union.

Please cite this article in press as: G. Tuna et al., The effects of exploration strategies and communication models on the performance of cooperative exploration, Ad Hoc Netw. (2012), http://dx.doi.org/10.1016/j.adhoc.2012.07.011