COLLABORATIVE UAV EXPLORATION OF HOSTILE ENVIRONMENTS

Linus J. Luotsinen*, Avelino J. Gonzalez, and Ladislau Bölöni
Department of Electrical and Computer Engineering
University of Central Florida
Orlando, FL, 32816

ABSTRACT

Unmanned Aerial Vehicles are frequently used for the exploration of a hostile environment. UAVs can be lost or significantly damaged during the exploration process. Although employing multiple UAVs can increase the chance of success, their efficiency depends on the collaboration strategies used. We present a cooperative exploration strategy for UAVs controlled by autonomous agents. The agents are sharing information, coordinate their short-term goals and path choices, while each agent uses state of the art algorithms for its individual path planning and obstacle avoidance. The overall goals are to minimize the exploration time, avoid damage by sharing information about threats, and be robust to the failures of individual UAVs. Extensive simulation results prove the validity of the approach and provide ways to determine the optimal number of UAVs for different exploration tasks.

1 INTRODUCTION

Unmanned Aerial Vehicles are considered the cutting edge of modern flight and aviation technology. Future unmanned combat robot systems will most likely incorporate autonomy and collaboration to further improve the machine-human interface (B. Koetting, 2003; C.M. Shoemaker and J.A. Bornstein, 1998).

In this paper we introduce and evaluate a collaborative exploration strategy for UAVs in hostile environments. This strategy is intended to be used for rapid exploration of large areas in e.g. war zones, contaminated zones or in areas where landmarks are to be avoided. Each agent utilizes a path-finder based on the work by Stentz (A. Stentz, 2003), tailored for probability based Occupancy Grid Maps (OGMs) (W. Burgard, M. Moors and F. Schneider, 2002), and Context Based Reasoning (CxBR) (F.G. Gonzalez, G. Patric and A.J. Gonzalez, 2000) to safely navigate through known zones. The choice of OGMs as internal map representation is based on its uncertainty management, merging- and search effectiveness. OGM consists of nodes with probabilities describing three classes. These classes are known, unknown and occupied. In addition to this the algorithm uses a frontier based collaboration scheme (B. Yamauchi, 1998) to delegate and distribute new waypoints within the agents.

2 PROBLEM STATEMENT

We consider a set of autonomous UAVs, controlled by intelligent agents, which are exploring a hostile environment in a collaborative manner. Their task consists of searching for safe paths, sharing information about known regions and finding unexplored regions within maps. The problem of collaboratively exploring hostile environments using autonomous UAV units can be divided into several sub problems:

1. Agent behaviour and decision making
2. Map representation
3. Collaboration

The agents controlling the UAVs are implemented using a common *agent framework*. The agent framework provides inter-agent communication, behavior modeling and team modeling capabilities for each individual agent. Another common component is the *map* used for storing agent sensory data acquired from the environment. The map needs to store information about known and unknown regions. Also, the organization of the map must allow efficient implementation of the path planning algorithms. An important component of our approach is the collaboration between agents during exploration. This collaboration includes mutual updating of the maps to allow for agents to know the total exploration progress and to determine when exploration is complete. The current destinations of the UAVs are distributed to avoid multiple explorations.

3 AGENT FRAMEWORK

The Context Based Reasoning framework, or CxBR, is primarily used to model tactical agent behavior. The CxBR paradigm is a simple and easily understood modeling technique that can be used to concisely represent knowledge and behavior for intelligent agents. The main concept is that contextual information influence the agent in its various decisions. Contextual information is represented as an extraction of key features from each situation in the environment where the most important

Report Documentation Page					*Form Approved* *OMB No. 0704-0188*
Public reporting burden for the collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to a penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.					
1. REPORT DATE **00 DEC 2004**		2. REPORT TYPE **N/A**			3. DATES COVERED **-**
4. TITLE AND SUBTITLE **Collaborative Uav Exploration Of Hostile Environments**					5a. CONTRACT NUMBER
					5b. GRANT NUMBER
					5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)					5d. PROJECT NUMBER
					5e. TASK NUMBER
					5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **Department of Electrical and Computer Engineering University of Central Florida Orlando, FL, 32816**					8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)					10. SPONSOR/MONITOR'S ACRONYM(S)
					11. SPONSOR/MONITOR'S REPORT NUMBER(S)
12. DISTRIBUTION/AVAILABILITY STATEMENT **Approved for public release, distribution unlimited**					
13. SUPPLEMENTARY NOTES **See also ADM001736, Proceedings for the Army Science Conference (24th) Held on 29 November - 2 December 2005 in Orlando, Florida. , The original document contains color images.**					
14. ABSTRACT					
15. SUBJECT TERMS					
16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**	**UU**	**6**	

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std Z39-18

features are based on the background and experience of the specific agent (F.G. Gonzalez, G. Patric and A.J. Gonzalez, 2000).

CxBR is based on the ideas that any recognized situation inherently defines a finite set of actions which address the current situation and that the current situation then can be used to simplify the identification of future situations by focusing on those that are likely to happen (A. J. Gonzalez and R.H. Ahlers, 1998). The CxBR paradigm of knowledge and tactical behavior representation is split between the following major components:

- Agent
- Mission context
- Major contexts
- Sub contexts
- Inference engine

The agent component is used as a base for a CxBR agent and it contains valuable information and capabilities, e.g. localization and velocity, about each individual agent in a system. The mission context component is used to describe the agent's overall objective and detect when it has been reached. Hence, each agent is assigned with a mission. A mission context is built upon a set of major contexts and their sub contexts. A major context contains transition rules and sub-contexts that may be activated during the agent's life cycle. The last component, the inference engine, is a general purpose component that shall be used when applying rule-based knowledge to agents. By using the inference engine and the fact base, new knowledge may be derived using either forward-chaining or backward-chaining. Figure 1 illustrates the relationships within CxBR and its components.

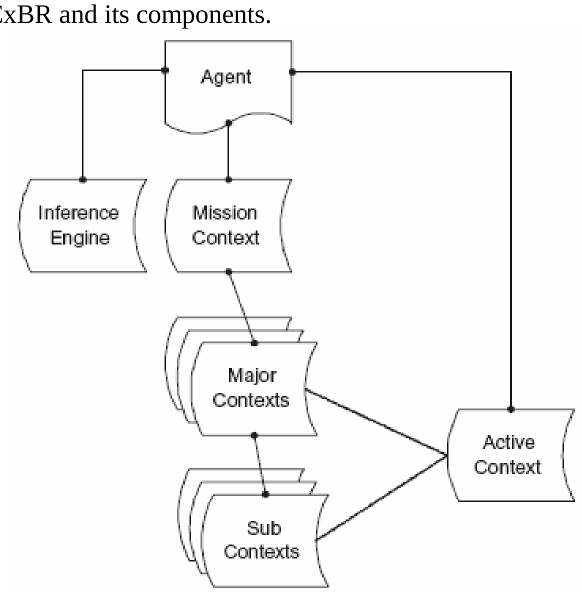

Figure 1. The CxBR framework and its components. Each agent is assigned a mission. Each mission has number of major and sub contexts from where only one can be active at any time. The inference engine provides rule based knowledge to each agent.

4 OCCUPANCY GRID MAPS

The main data structure for any exploration algorithm is the map. The map serves as a data store for unknown regions and known regions in the environment. In addition, for this study, the map must store information about possible threats and obstacles.

We chose to use the Occupancy Grid Map data structure to represent our map information. OGM is a probability based grid map where each cell in the grid represents a probability value of occupancy. We modeled our occupancy grid maps so that probabilities of unknown cells have values of 0.5; known spaces are either higher or lower than 0.5 depending on whether the cell is occupied or not occupied respectively. We define occupied cells to be threats and obstacles.

4.1 Merging

Merging multiple OGMs can easily be performed by applying the following equations to *n* maps (W. Burgard, M. Moors, D. Fox, R. Simmons and S. Thrun, 2000):

$$P(occ_{x,y}) = \frac{odds_{x,y}}{1 + odds_{x,y}} \tag{1}$$

where

$$odds_{x,y} = \prod_{i=1}^{n} odds^{i}_{x,y} \tag{2}$$

and

$$odds^{i}_{x,y} = \frac{P(occ^{i}_{x,y})}{1 - P(occ^{i}_{x,y})} \tag{3}$$

4.2 Convolving

Avoiding obstacles in path-finder problems for known environments is necessary so that an agent can have a certain space for error and mistake. E.g. for robots in indoor environments one would want to calculate a path that will be able to carry the robot through narrow spaces without being trapped. The following equations (Burgard, 2000) can be used as a fast and reliable solution to the problem for occupancy grid maps.

$$P(occ_{x_i,y}) = \frac{1}{4} * P(occ_{x_{i-1},y}) + \frac{1}{2} * P(occ_{x_i,y}) + \frac{1}{4} * P(occ_{x_{i+1},y}) \tag{4}$$

$$P(occ_{x_0,y}) = \frac{2}{3} * P(occ_{x_0,y}) + \frac{1}{3} * P(occ_{x_1,y}) \tag{5}$$

$$P(occ_{x_{n-1},y}) = \frac{1}{3} * P(occ_{x_{n-2},y}) + \frac{2}{3} * P(occ_{x_{n-1},y}) \quad (6)$$

These equations should be applied, in the case of a two dimensional map, to both rows and columns. The first equation should be used for the general case where the cells in the map are located inside the map borders. The second equation should be used when the cells reside within the very first row or column and the third equation should be used when the cells reside on the very last row or column of the map.

5 PATH FINDING

Path-finding is the process of generating or planning a path for a movable robot or any type of moveable agent in an environment. Although the path-finding problem is one of the most studied ones in classic artificial intelligence, it remains one of the most difficult ones. In general, uninformed search algorithms can solve only the most trivial toy problems. The algorithms used in practice are a combination of search and heuristics. The algorithm used by our system is a version of the heuristic A* adapted from (A. Stentz, 2003).

As mentioned before, occupancy grid maps provide good management of uncertainty. Occupancy grid maps can also be merged and integrated with other occupancy maps using equation described in section 4.1. Also, we have seen that occupancy maps can be convolved, by applying equations described in section 4.2, so that obstacles and threats in the environment are avoided for safe planned paths.

To apply A* we need to determine the metric ***g*** used to measure the cost of a path. In an occupancy grid map the ***g*** values are represented by the sum of probabilities in the map along the chosen path. This situation is presented in Figure 2, where we can see the initial start and goal points as well as obstacles/threats and movable spaces. The next step in the application of a A* style-search is to choose an admissible heuristic function ***h***. The heuristic function is normally the cost function for the simplified problem. This function will always underestimate the true cost of the optimal path. In our case, we choose the distance function assuming there are no obstacles in the map, which is essentially the Manhattan distance function.

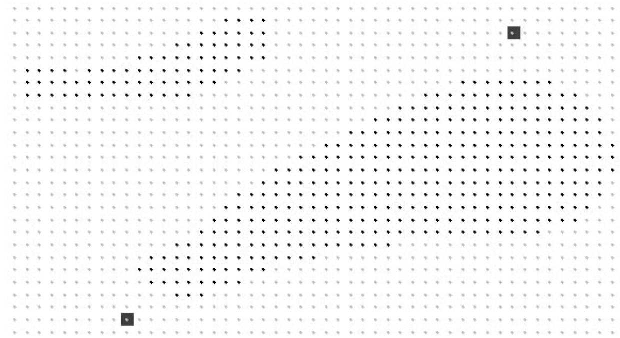

Figure 2. Initial path finding problem. The black squares represent start point (upper right) and goal point (lower left). The black dots depict obstacles or threats and the light-gray dots represent know and movable terrain within the OGM.

Applying the A* influenced path-finding algorithm to the start and goal points in Figure 2 results in a collision free path depicted by Figure 3.

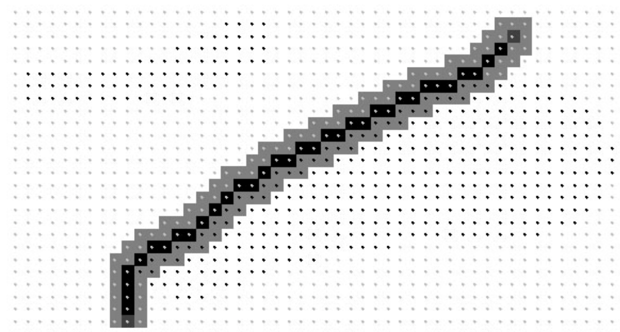

Figure 3. Collision free path. The black squares show the generated path and gray squares shows additional nodes explored by the search algorithm.

As Figure 3 depicts, there is no collision in the generated path. However, having an agent's path planned that close to an obstacle/threat may incur unnecessary risk of collision due to real-world constraints, erroneous positioning or sensory noise. To solve this problem a convolving factor is introduced. The convolving factor simply depicts how many convolutions that should be performed, by applying the equations in section 4.2, on the occupancy probabilities. As we can see in Figure 4, convolving the map produced a much safer path by the path finding algorithm. Figure 4 was produced with a convolve factor of 3.

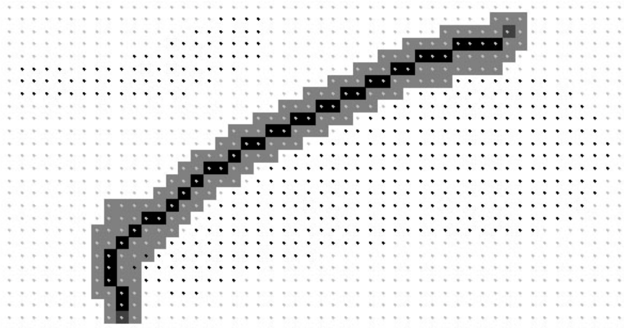

Figure 4. Safe and collision free path using convolving factor set to 3. The path is generated with a distance to the threat.

6 EXPLORATION STRATEGY

Each agent has its own OGM map which is updated as more information becomes available. The agents communicate to update their maps with information collected by other UAVs, and to efficiently allocate the exploration subtasks. The goal of the agents is to explore unknown terrain while avoiding threats. The unknown terrain to be explored is chosen from the set of *frontier points*, defined as a point that separates unknown terrain from known terrain. A high level overview of the strategy deployed by each agent is as follows:

1) Let ***F*** be a list of available frontier points
    a. If ***F*** is empty then exploration is completed
2) Let ***AF*** be a list of allocated frontier points
    a. For each ***P*** in ***AF***
        i. Remove ***P*** from ***F***
        ii. Filter vicinity frontiers of ***P*** from ***F***
    b. Select a frontier point ***A*** from ***F***
    c. Inform team members that ***A*** now is allocated
    d. Plan path using ***A***
3) Navigate through generated path
    a. When ***A*** is reached start over at 1)

It is up to each individual agent to filter and choose waypoints from 2aii) and 2b) respectively. If filtered vicinity points in 2aii) are not chosen with care the algorithm will most likely suffer from frontier starvation. This is a state where the no frontier points are available due to overcrowding.

6.1 Frontier starvation

Frontier starvation occurs in the algorithm when there are too many agents trying to search the environment at the same time. The system gets over-crowded because there is a limited amount of frontier points available at any instant of time. Since the number of frontiers available for the agents is dynamically changing, agents may not be able to find a free frontier point. Frontier starvation generally occurs when agents are launched too close to each other in time. The impact of frontier starvation on the algorithm is that the overall mapping efficiency is reduced. The number of agents useable to search an environment is mainly limited by the frontier starvation problem.

6.2 Frontier selection

In order to achieve the most efficient exploration, the frontier point selection of the agents needs to be coordinated. For example, it is inefficient for two UAVs to explore the same region (or even regions very close to each other). The frontier selection can be seen as a resource allocation problem, and there are several algorithms which can be applied.

Due to the real-time nature of the problem, one-show heuristic approaches are preferable to complex optimization algorithms. In the following we investigate two algorithms (a) greedy selection and (b) lowest cost selection.

Greedy selection of frontier points always chooses the closest frontier point to the current location of the UAV in terms of physical distance. This approach is easy to implement and very fast (its computational complexity is O(n), where n is the number of available frontier points). The drawback of this algorithm is that the physical distance might not be the best predictor of the cost to reach the frontier point. Relatively close points may be expensive to reach, if they are separated by obstacles from the current location.

The *lowest cost selection* algorithm is similar to the greedy algorithm, but instead of the physical distance it is using the actual cost of the path to the given frontier point. This approach avoids the drawbacks of the greedy algorithm. The complexity of this algorithm is O(n)*O(pathfinder), where O(pathfinder) is the complexity of the path finder algorithm, in our case, A*. As the complexity of A* is exponential at worst case and relatively large even for average case, the lowest cost selection can be a significant problem.

In the case where there simply are too many frontier points to find costs for, one can select a set of frontier points that are closest to the current agent location and find the costs for these. By limiting the cost value generation to these frontier points only one can increase the performance significantly. However, this hybrid approach cannot guarantee that the best path is always chosen.

6.3 Frontier allocation

Once an agent have selected a frontier point to visit it is important to allocate this frontier point and its vicinity points as the agent's resource. This is done to decrease the amount of overlapping environmental mappings performed by the agents. This implicitly forces agents to collaborate. In this section we will discuss one type of resource allocation which may be performed on multiple agent environmental mapping application domains.

The sensor based approach, introduced here, relies on the functionality, or maximum coverage, of the sensor. If such value is available it can assist in the estimation of expected visibility by allocating the frontier points in its range. It is this value that decides the behaviour of the team. We might choose to use a value higher or lower than the estimate, thus controlling the deployment of the UAVs. A value higher than the estimate will force the agents in the team to explore the area in a more scattered way.

The main advantage of using this allocation algorithm is that it requires very little communication between the UAVs. Basically a team of homogeneous agents, with equal sensor range, only need to provide each other with its current frontier point selection, as the agents can estimate each others resource allocation. Hence, each agent in the system will have a list of already allocated frontier points. The filtering of available frontier points in the local map can then be performed onboard each agent in the system based on the already known sensor visibility range and the list of allocated frontier points.

7 SIMULATION RESULTS

We have simulated the algorithms presented in this paper in the context of a realistic mission scenario. In this scenario a set of scout UAVs depart from the UAV base. The mission is to find enemy landmarks, such as buildings or SAM sites, as fast as possible. The main difficulty in this scenario is that the agents must cooperate and coordinate their actions for a more efficient exploration. In this type of scenario each agent must maintain its own path-finding algorithm so that planning of navigation is performed in a secure and reliable way. The agent can distribute the locations of hostile units among each other so that danger points can be avoided when planning for routes. This scenario ends once the agents have covered the whole map and when all the landmarks have been identified.

We have simulated the algorithm on a in-house developed framework influenced by (G.E. Smid K.C. Cheok, G. Gerhart and G Hudas, 2002). We used a realistic generated terrain, with 5 SAM sites distributed in the environment. A snapshot of the simulation environment, presenting the terrain, the location of the UAVs and SAMs and the currently active communication links is presented in Figure 6.

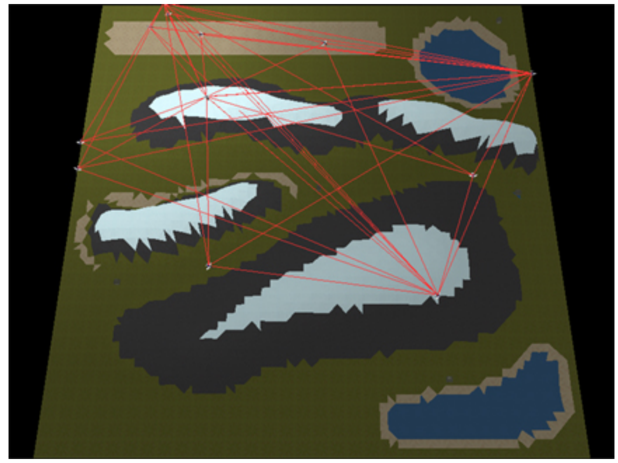

Figure 5. A birds-eye view of the environment. The lines indicate active agent communication.

Extensive experiments were performed in which teams of 1, 2, 4, 6 and 8 UAVs explored the environment under the control of the agents. The diagram in Figure 7 presents the results from 100 simulated runs for every configuration. We observe that the exploration time is initially decreasing with the number of UAVs deployed, but after reaching an optimum (in our case, 6 UAVs) the time required to finish the exploration is actually increasing, due to phenomena such as frontier starvation and additional communication needs.

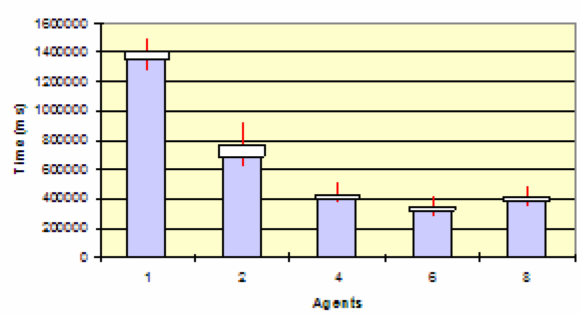

Figure 6. The relationship between the exploration time and number of UAVs. The diagram shows the mean, minimum, maximum and the 95% confidence interval.

8 CONCLUSIONS

UAV units and a brief description of their many modern day implementations have been introduced in this paper. One of the most interesting tasks for UAV implementations is autonomous mapping of environments in team collaboration and coordination domains. To efficiently search for landmarks or simply map an environment in a system consisting of multiple UAV units it is beneficial to use flexible agent frameworks that can handle contextual information and react to this information accordingly. The CxBR paradigm was proposed, implemented and utilized as a base agent framework for UAV agents mainly because of its simplicity and expressive design features.

The environmental exploration problem was proposed to consist of the choice of internal agent map representation, path finding, obstacle avoidance, information merging and expected visibility for resource allocation. A probabilistic approach that makes use of occupancy grid map representations was presented to solve the problem of map representation choice as well as information merging. A path finder, tailored for occupancy grid maps, was developed not only to solve the problem of planning a collision free path but also to find a safe path by using convolving factors. A simple expected visibility solution, based on sensor capability, was proposed to determine each agent's allocated resources. An algorithm that incorporates all of the exploration problems was proposed and implemented. The algorithm was tested in a simulation environment and the results were statistically evaluated. The algorithm used in a multi-agent system, with 4 agents, produced over a 300% speed-up improvement compared to a single agent system.

REFERENCES

W. Burgard, M. Moors and F. Schneider. Collaborative Exploration of Unknown Environments with Teams of Mobile Robots. In dagstuhl. Springer Verlag, 2002.

W. Burgard, M. Moors, D. Fox, R. Simmons and S. Thrun. Collaborative Multi-Robot Exploration. In proceedings of the IEEE International Conference on Robotics and Automation (IRCA), 2000.

F.G. Gonzalez, G. Patric and A.J. Gonzalez. Autonomous Automobile Behaviour Through Context-Based Reasoning. In proceedings of the Thirteenth International Florida Artificial Intelligence Research Symposium Conference (FLAIRS), 2000, p2-6.

A. J. Gonzalez and R.H. Ahlers. Context-Based Representation of Intelligent Behaviour in Training Simulations. In Naval Air Warfare Center Training Systems Division Conference, 1998.

B. Koetting. Advanced Robotic Planning: Breaking the Barriers between Man and Machine. In 23rd Army Science Conference, 2003, NP-11.

C.M. Shoemaker and J.A. Bornstein. The Demo III UGV program: A Testbed for Autonomous Navigation Research. In proceedings of the IEEE International Symposium on Intelligent Control, 1998, p644-651.

G.E. Smid K.C. Cheok, G. Gerhart and G Hudas. High level architecture for collaborative robotics simulation environment. In proceedings of the ISCA 11th International Conference. Intelligent Systems: Emerging Technologies, 2002, p161-166.

A. Stentz. Constrained Dynamic Route Planning for Unmanned Ground Vehicles. In 23rd Army Science Conference, 2003, NO-04.

B. Yamauchi. Frontier-Based Exploration Using Multiple Robots. Proceedings of the Second International Conference on Autonomous Agents (Agents '98), 1998, p47-53.