International Journal of Latest Research In Engineering and Computing (IJLREC)
Volume 1, Issue 1 : Page No.12-20, September-October 2013
www.ijlrec.com

A Study of Behavior-based and Role-based Autonomous Multi-Robot Exploration and Coverage Systems

K. S. SENTHILKUMAR
Department of Mathematics and Computer Science
Papua New Guinea University of Technology
Lae 411, Morobe Province
PAPUA NEW GUINEA

Abstract: The autonomous multi-robot exploration and coverage is a well-addressed problem in the field of robotics. Exploration and coverage is the task of guiding robots in such a way that they cover the environment in an efficient and effective manner. The Behaviour-based system and Role-based system for multi-robot control is popular in autonomous multi-robot exploration, because they are robust to the dynamic interactions inherent in any multi-robot systems. In this paper, the existing Multi-Robot exploration and coverage systems under the characterization of behaviour-based and role-based are identified and their strengths and shortcomings discussed. Existing solutions to this problem differ primarily by the type of coordination that exists between the robots. The level of coordination depends on the type of communication the robots are expected to share.

Key words: autonomous mobile multi-robot, behaviour-based exploration, role-based exploration, navigation behaviour, path-planning behaviour.

INTRODUCTION

Autonomous Mobile Robot (AMR) is a machine that can interact with the environment, perceiving the state of its properties through its sensor and changing its state. The way sensors and actuators are linked depends on the control architecture responsible for the behaviour of the robot under the environmental conditions. One of the fundamental challenges in AMR is to explore the unknown environment efficiently and effectively. The efficiency and effectiveness of the exploration are typically measured by the map coverage, map accuracy and exploration time. Exploration and Coverage of terrain during motion is important in many application domains, such as floor cleaning, lawn mowing, demining, harvesting, and search and rescue robots, etc. In such applications usually one needs to cover the terrain only once. The main reason for using the Multi-Robot System (MRS) is that they are a convenient solution in terms of costs, performance, efficiency, reliability, and reduced human exposure.

Arkin (1998) classified the current AMR control design into four groups. First category is Deliberative systems associated to the intelligence, the main goal is to develop the brain of the robot, leaving out the processing related to sensors and actuators (think hard and then act). In this type the robot uses all of the sensory information and internally stored knowledge to reason the action to take next. The deliberative system consists of three steps that need to be performed in sequence: Sensing, Planning and Acting. The sensing module, responsible for perception, has to obtain a highly reliable representation of the current state of the system based on readings from sensors. After extracting from the readings the current state of the environment, the planning module finds the sequence of actions that will bring the robot to the destination or the goal. Hence, the acting module is responsible for transforming the sequence of actions to signals that the actuators can understand to move towards the goal.

The second category is Reactive systems, which reacts to the environment becomes the main goal instead of thinking (don't think, just react). This is a popular approach to mobile robot control. But in more complex types of tasks where in-build memory and learning are required, reactive control is not sufficient. The reactive system has no planning module, so it neither has to create nor to maintain the environment model. It is simply based on the direct readings of the sensors to act as quickly as possible. As a consequence, the main loop of this control paradigm is much faster than that of the deliberative one. The control usually consists of a collection of rules that do not require much thought.

The third category is Hybrid systems, which is the combination of the first two systems attempts to overcome the drawbacks of deliberative systems adding the capability of interacting with the environment (in parallel as well as independently think and act). But the interaction of deliberative and reactive components needs an intermediary, whose design is typically the greatest challenge of this system. The main idea of hybrid systems is to combine the better of these two systems, taking the intelligence of deliberative systems and enabling them to react upon dynamic environmental changes (Matari´c, M. J. 2009).

And finally the fourth is behaviour-based systems, which are based on reactive control, but robots with more complex capabilities are developed. The behaviour approach to autonomous robot control is based on the observations of animal behavior. The most common application uses of the simple local control rules of various biological societies, particularly ants, bees, and birds, to the development of similar behaviours in cooperative robot systems. Many works have demonstrated the ability for multi-robot teams to flock, disperse, aggregate, forage, and follow trails. Behavior-based system has been a popular paradigm of choice in the control of multi-robot systems and was already studied in (Jones and Maja J Mataric, 2005). This control system represents a robust and effective way to control individual as well as multiple robots.

The distinction between deliberative and reactive control, hybrid and behavior-based control is often a matter of degree, based on the amount of computation performed and the response time of the system to relevant changes in the world. Rather than to try to summarize the research articles in this paper into a taxonomy of autonomous multi-robot systems, I organize it by the principal topics that have generated significant levels of study, to the extent possible in a limited space. The principle topic areas of Multi-Robot Systems that I have identified are:

Behaviour-based System

Role-based System

Aspects of articles as they apply to each of these key research areas and their strengths and shortcomings are also discussed in this paper.

BEHAVIOUR-BASED SYSTEMS

After Brooks (1986) (one of the founders of modern robotics) invented his *Subsumption Architecture,* the behavior-based robot control is established as one of the most powerful systems for controlling complex robot systems. It has served as an effective methodology for multi-robot control in a large number of multi-robot problem domains. This behavior-based paradigm has had a strong influence in much of the cooperative mobile robotics research. Instead of top-down decomposition, he proposed a bottom-up methodology for building robots. Agents with sensing and acting capabilities should be developed and behaviours should be integrated incrementally. Behaviour-based systems are distributed parallel control architectures in which each basic processing unit produces behaviour on the robot. Thus, a behavior-based system lacks centralized control. This control architecture is closer to reactive control and it is often confused with it. In other word, behaviour-based systems are biologically inspired systems developed bottom up, like reactive systems. The difference lies in the processing units and the level of representation allowed in the system too. Instead of precompiled rules, the basic processing units of behaviour-based systems are named behaviours. Intelligent behaviour should emerge from the interaction between the agent and the environment. Behaviour-based systems are best suited for the systems in environments with heavy dynamic changes. But effective behaviour selection is the key challenge in this system, as it determines which behaviour controls the robot at a given time. The following are the characteristics of behavior-based system:

- Behaviour has only one goal.
- Behaviour receives inputs and produce outputs in a continuous manner.
- One behaviour is able to communicate with other behaviours, receiving inputs from other modules as well as from sensors.
- Behaviours are more complex than rules

When the coordination of behaviors is concerned, there are two schemes available. The first one is command arbitration scheme is the process of selecting one action among all possible candidates actions, so called competitive method. In robotics, controlling individual behaviors is called *arbitration* and is accomplished by an *arbitration mechanism,* which decides how the various behavioral outputs will combine to produce the response. This method has three types, such as priority, action-selection, vote-based (Sorbello R., and et al., 2004). Priority means response of behavior with highest level of subsumption, action-selection uses response of behavior with highest activation level and vote-based response the behaviour with most votes. Secondly the command fusion or cooperative scheme is the process of combining all the candidate outputs in a single output action. This type often implemented as a neural network. An interesting technique, Learning Momentum (LM) for multi-robots, was demonstrated by Lee and Arkin (Lee J. and R. Arkin, 2003). The main idea of LM is behavior's weight's modification. Weights are adjusted by a gradient descent method during the learning process and then the behavior manager is fusing behaviors accordingly to them. All of the behaviors have different goals, and thus the technique learns to select between competing goals.

NAVIGATION BEHAVIOUR

This is one of the important behaviour in AMR exploration. Navigation behaviour could be classified into two main groups. They are local navigation strategy and path finding strategy.

3.1 Local Navigation Strategy

This is local control mechanism that allows the agent to choose actions based only on its current sensory input. These strategies rely only on the characteristics that are within the agent's perceptual range. Four types of strategies that fall in this group are:

- Integrate or store the direction and distance of the movement to maintain the association between the source and the target positions is so called Path integration.
- Without any purpose or objective move around in the environment safely and avoiding all types of obstacles is so called Search or wander.
- Agent sets its self-centered relationship with respect to the elements surrounding the target to find the goal so called goal orientation.
- Move towards a target that produces a certain stimulus so called taxis. It is certainly involved in most navigation behaviours but does not necessarily include the essential capabilities of locomotion and goal recognition.

3.2 Path-finding Strategy

This is responsible for motivating the agent to goals outside the agent's perceptual range which require recognition of different areas and relations among them. But they depend on local strategies. There are three types of path-finding strategies and are perception-triggered response, topological navigation and terrain exploration.

3.2.1 Perception-triggered Response

This strategy connects two locations via a local navigation mechanism. A location is defined as a particular perception or a landmark and used to build paths. The paths or the sequences are independent of each other and direct the agent towards a unique goal. But this strategy cannot be used for path planning since knowledge is limited to the next action to perform. Linking perceptions to a unique action is called PTR and limits the navigation system to always use the same sequence of locations. These locations are combined to define the route.

3.2.2 Topological Navigation

This strategy requires abilities like path integration and planning. Topological maps represent the connectivity of places located in the environment and facilitate fast planning. It is essential to know the relationship among the connected locations in the environment to plan a path between the robot's location and the goal; hence this strategy is not suitable to generate paths in unknown environment (Szabo, R., 2004). Daniel Schmidt and et al. (2006) introduced a new hierarchical behavior-based control system which enables a mobile robot to derive a topological map of indoor environments completely autonomously. This approach has been tested using simulation and in real environment on robot Marvin. They assumed that the walls are arranged in right angled, rooms are described by using rectangles, and walls are fixed. The main challenge was the coordination of the behavioral network in order to guide the robot autonomously through priori unknown environments using laser scanners. One of the negative sides of this approach is that the environments with other shapes will be approximated by rectangles which may lead to inaccurate maps of the rooms. These restrictions have been made to keep the representation as abstract as possible and to make it more robust against dynamic changes. A topology matrix was used to show the connections between the respective room maps for navigation tasks. Targeting a hierarchical structure that is easy to upgrade a control system has been implemented with different grades of abstraction. The lowest level is composed of basic behaviors including safety aspects and simple movements while the exploration behavior and additional components are implemented on higher levels. The exploration behaviour consists of three concurrent behaviors so called local exploration, path tracker and global observer. Local Exploration sub behaviour allows the exploration of the current room. Path Tracker behaviour used for navigation task drives the robot to the closest point of interest by using a topological graph. And the Global Observer behaviour surveys the local exploration and if for a period of time no new information has been retrieved it triggers the path tracking behavior.

3.2.3 Terrain Exploration

A robot is able to find novel paths over unknown terrain, since the embedding of the current location into the common frame of reference allows the robot to infer its spatial relation to the known places. Burgard (Burgard and et al., 2005) worked on the problem of exploring unknown environments with teams of robots. An algorithm was developed for coordinating a group of robots for exploration.

Steven (Steven Damera and et al.) addresses on dispersion and exploration problems by focusing on methods for developing reusable behaviors to control multi-robots in an unknown environment. They presented algorithm and validate them experimentally using a Player-Stage simulation environment. The experiment was conducted in an unknown environment with complex connectivity and populated with obstacles; the robots operate independently with limited communications and sensing capabilities with no central control. This approach uses basic behaviors to control the motions of each robot. They also presented two different distributed algorithms that build a network of robots connected by communication links while they disperse in the environment. They do not assume the robots know where they are, since for small robots localization is very hard. The robots were dispersed in an environment which includes walls and obstacles and each robot remains within communications range of at least one other robot. They were interested in working in very large environments where requiring a higher degree of connectivity might require using too many robots. No possibility of single point of failure in this algorithm since there is any global control. The authors didn't want to put a heavy burden on sensors to either recognize other robots or calculate odometry accurately. And also they like to allow some flexibility since more robots could be introduced after the initial deployment and failures are not catastrophic. Because of the above two reasons they preferred that the robots do not need to know how many other robots are operating in the same environment, where those robots are located, and where those robots have been. After experiments, they have found three basic behaviors that are the most effective for robot dispersion such as random walk, find opening and comparison. The Random Walk behavior is used to move the robot on a slightly curved path by turning a small random amount per step and when the robot detects an obstacle, it stops, turns by a larger random amount. The find Openings behavior uses sensors to locate openings such as doorways or halls. The comparison behavior is able to recognize another robot that is nearby and move away from it. They have developed two more behaviors such as random range control and backtrack-range control to move the robot outside the communications range of all other robots. The Random Range control algorithm moves the robot randomly until communications is reestablished while the backtrack-range control behaviour attempts to move the robot back to previous positions.

In this paper, they presented a Hybrid approach for single and multi-robot efficient autonomous exploration (Jesus S. and et al.,). There are multiple behaviors in this approach named as obstacle avoidance, avoid past locate open area, disperse and explore-emergent. The Obstacle avoidance behaviour considers three particular conditions to maintain the robot's integrity. The Avoid Past behaviour is for gathering the newest locations. This kind of explorative behavior is used for avoiding local minima when navigating towards a goal. The Locate Open Area is for locating the largest open area in which the robot's width fits. They focused on gathering new locations by enhancing reactive solutions with a local spatial information memory. It consists of a wandering rate that represents the frequency at which the robot must locate the open area, which is basically considered as the biggest surface without obstacles being perceived by the laser scanner. The Disperse behavior is inspired by the work of Mataric´ activates just in the case two or more robots get into a predefined comfort zone. In a situation for m robots near in a pool of n robots, they call for simple conditional checks so as to derive an appropriate dispersion action serve as the coordination mechanism for efficiently spreading the robots as well as for avoiding teammate's interference. The explore-emergent behavior fuses the outputs of the triggered behaviors with different strategies according to the current state. At initial state generally the robots are deployed together, so that the initial state comes to be Dispersing. During this state, the Disperse and Avoid Obstacles behaviors take control of the outputs. They tested using 3 robots starting inside the predefined comfort zone considered only *Disperse* and *SafeWander* behaviours. The results showed this combination may be appropriate in cases where it is preferable to get an initial rough model of the environment and then focus on improving potentially interesting areas with more specific detail.

PATH PLANNING BEHAVIOUR

The years of research have resulted in a great number of path planning methods. The purpose of the path planning behaviour is to determine an optimal path between the robot's current location and that of the goal. When robot is in a completely unknown environment, it has to be able to gain a certain abstract internal representation of its world without any user interaction. This knowledge enables the robot to know how to get from its actual place in one place to a target position in another place as a prerequisite for tasks such as transportation. In this context, the combination of a behavior-based motion control system and an abstract topological map based on geometric representations of the environment seems promising. It has been used to develop completely autonomous exploration strategies for deriving topological representations of common indoor environments in the application areas such as cleaning robots, factory floor robots, etc. These traditional methods are generally based on three approaches to path planning (Latomb, J., 1991, Timothy Arney, 2007), such as Potential field approach, roadway approach (Visibility graph, Voronoi diagram, free space method) and cellular decomposition (grid method).

4.1 Roadway based approach

This approach tries to identify sets of routes in the configuration of free space of the environment. Voronoi graph has been used for topological map building, navigation, and place detection [Beeson P. and et al., 2005, Howie Choset, and et al., 2000, Berg J. and et al., 2005). When applied to a map, points in the walls are added to the set S in order for standard Voronoi graph calculation algorithms to work. These methods reduce the problem to that of a graph search by fitting a graph to the space. In order for the algorithm based on a given roadmap to be complete, it must follow certain topographical properties.

Rekleitis (Rekleitis I., and et al., 1997) uses two robots in on-line settings, using a visibility graph-like decomposition. The algorithm uses the robots as beacons to eliminate odometry errors, but does not address catastrophic failures. Howlett (Howlett, J.K., and et al., 2004) discusses use of Voronoi roadmap methods for practical unmanned helicopter operation. In a more recent article, Rekleitis (Rekleitis I., and et al., 2004) extends the Boustrophedon approach to a multi-robot version. Their algorithm also operates under the restriction that communication between two robots is available only when they are within line of sight of each other, but has many points of failure, i.e., it could stop functioning if one of the key robots fails. Khalid and et al. introduced a real time algorithm based on D* lite for multi-robot path planning in dynamic environment [Khalid Al-Mutib, and et al., 2012). Their method uses a grid based road map to define the problem. They assumed that single centralized obstacle map is available for all the robots. This algorithm was implemented and tested in simulation only. Static or dynamic obstacle can be added during execution to the obstacle map. Initially each robot marks obstacles over the occupancy grid on a central server, and then the algorithm chooses a priority order for the robots based on a heuristic criterion to plan a path for each robot.

4.2 Cellular Decomposition Based Approach

Grid-based method decomposes the environment in regions named as cells. After determining the free cells and the adjacency of the cells, cell decomposition searches for routes that link the starting and the goal cells (Szabo, R. 2004, Senthilkumar and et al, 2012). In cell decomposition approach, the path is actually a sequence of adjacent cells which connect the initial cell, where the robot currently resides, and the cell of the goal. While it is straightforward to determine which cells, the robot and goal lie in. The challenge is to determine the sequence of cells which connect them. Grid-based maps are considerably easier to learn, because they facilitate accurate localisation, and are easy to maintain. Ferranti proposed an algorithm [Ferranti, E., and et al., 2007) which divides the area into regular square grid cells, and the robot explores the environment marking the traced cells during its movements. This algorithm is not efficient in terms of the number of needed steps to explore the whole environment as it has no stopping criterion, and the robot will keep working until its battery becomes empty or the user terminates the algorithm.

4.3 Potential Field Based Approach

Potential field methods are based on the idea of assigning a potential function to the free space, and simulating the robot as a particle reacting to forces due to the potential field. A potential field which has the properties of a navigation function makes a complete path planner. Obstacles generate repulsive forces whereas the free space and the target position generate attractive forces. Within that configuration the robot reaches the goal by moving on the negative of the gradient. There are two classes of potential fields known to satisfy properties of a navigation function: those based on harmonic functions and those based on solving the optimal distance-to-go function.

Gu Fang (Gu Fang, and et al., 2004) proposed a method for multi-robot exploration that combines the behaviour-based approach and the optimization strategy. They used the social potential-fields algorithm to obtain the coarse direction of the robot movement and a grid-based map is used to represent the environment. It is assumed that the robots are equipped with sensors with a fixed finite sensing range and the locations of robots are known to each other. There is a computer which controls the system which accepts control commands at a fixed frequency. They introduced three robots to explore the same environment. The behaviour based method is used to obtain the direction of each robot’s movement that will avoid obstacles and inter-robot collisions as well as will move towards the unexplored regions. And the optimisation technique is used to fine tune the moving angle and speed of each robot, such that the coverage of the unknown environment for the movement is maximized. Three forces are considered for each robot. One is the repulsive force introduced by obstacles that are near the robot. Another repulsive force is introduced by nearby robot(s). The other force is the attractive force created by the unexplored areas. The equation of the combined force F_{total} generated using the potential-fields can be expressed as:

$$F_{total} = \sum_{i=1}^{n} F_{obstacle.i} + \sum_{j=1}^{m} F_{robot.j} + \sum_{k=1}^{p} F_{frontier.k}$$

Generated by obstacle *i*, robot *j*, and frontier *k*, respectively, to the robot, *n*, *m*, and *p* are the total number of obstacles, robots, and frontiers, respectively, in the range to be considered. The robot movement will be in the direction of F_{total}. When the potential-fields based method is used, robots may potentially be trapped in a local minimum when the total force equals to zero. Therefore, an optimization method is introduced to fine tune the movement of the robot to achieve a maximum exploration. It is stated that an optimization method is used to find the moving speeds of the robot as well as the small angle deviations from the direction of F_{total} for a certain number of time steps, such that the total coverage of the unknown area in the considered time period is maximized.

Miguel et al. (Miguel Juli´a, and et al., 2010) presents an approach to the integrated exploration problem for multi-robot. Their technique is based on a combination of several basic behaviours that model a potential field. They have demonstrated experimentally that a simple coordination consisting in the weighted summation of the potential generated by each behaviour is enough when setting the appropriate width for each behaviour. The weight for each behaviour k_i represents its relative importance. The problem of local minima in potential field based techniques is also considered. In this sense, a strategy of detection and escape from local minima is used. As a novelty, this technique considers returning to previously explored areas when the localization uncertainty is high. As a result, the accuracy obtained in the construction of the maps is higher than with other classical exploration techniques. The potential fields in this approach are generated by simple discrete Gaussian functions. A positive sign of Gaussian function means a repulsive behaviour, whereas negative amplitude means an attractive behaviour. Thereby, using this simple potential field approach, six basic behaviours have been defined, such as *goto-frontier, goto-unexplored cells, avoid obstacle, avoid other robots, goto-precise pose and path following*. This approach proposes an arbitration scheme that decides what behaviours are enabled at each moment. They consider three possible states that are related to different sets of active behaviours, which are associated to different tasks. They are the exploration of the unknown environment, the preservation of a good localization of the robots, and the prevention of getting blocked by a local minimum. The detection of local minima is made by analyzing the potential field in the neighbor-hood of the robot. In the case of detecting a local minimum, the robot determines its nearest frontier (or past precise pose) and the shortest path to arrive to it was achieved by using the Dijkstra's Algorithm. This method has been tested in simulation. Experiments were performed for each scenario varying the number of robots in the team from 1 to a group of 4 robots. However, the robots always start the exploration in near locations. The authors showed that the exploration time is reduced notably with the number of robots. The experimental statistics show that the robots employ a great part of the time exploring and little time in active localization actions.

Howard (Howard, A., and et al., 2002) uses a traditional approach of potential fields in robotics and enhances it for the purposes of deployment of mobile sensor networks. Robots are subjected to virtual repulsive forces from other robots and an internal viscous force that ensures equilibrium is eventually reached. Obstacles are likewise treated as repulsive forces, allowing robots to deploy around obstacles while maintaining an equilibrium state with other robots. The approach is an entirely local approach, and does not require global positioning or localization techniques.

ROLE-BASED SYSTEMS

In Role-based exploration, each robot in the team is assigned one of two possible roles: *explorer* and *relay*. When environments have severe communication limitations, we can choose role-based exploration technique. Most of the time roles are assigned prior to the beginning of the exploration. It is essential that *Relay* and *Explorer* agree on a specific location for rendezvous (meeting point), so that they can find one another. The rendezvous location is thus a specific place in the map, chosen by the *explorer* and communicated to the *relay*. The difficulties are compounded in dynamic environments, where paths previously believed to be free can suddenly become blocked. Using role-based exploration, some robots continuously explore the environment while others ferry information back and forth to a central command centre. This type of approach can be applied for the real world problem of robotic such as search-and-rescue, underwater or planetary exploration. Role-Based Exploration demonstrates several advantages over other methods, particularly as communication becomes less reliable. New information obtained by the team is brought to a single location quickly and in regular intervals, team members share information well and often, and the full team effort can be easily monitored and controlled.

1. *Explorer* has the task of exploring the extreme reaches of the environment. To communicate their findings, it returns periodically to previously agreed rendezvous points where it passes its knowledge about the terrain to a *relay*.

2. Relay acts as mobile link between *explorer* and the Base Station, ferrying information back and forth. The primary purpose of a *relay* is to communicate *explorer's* findings up the communication chain, and to communicate control commands from the base station down the communication chain.

Relay and *Explorer* most of the time share the same map when in range. Therefore, the *explorer* can predict the *relay's* path, and determine how long it will take the *relay* to return to the base station, turn around, and make its way to the next rendezvous point. Sometimes robots need to swap their roles and exchange places within the team to solve the problem of inefficient motion in dead-ends and loops. At any point in time the *explorer* can check whether the *relay* is close to reach rendezvous, and whether the *explorer* itself should stop exploring and make its way to rendezvous as well. In Role-Based Exploration, there is a great deal of uncertainty. At the lowest level, sensor readings always carry a degree of uncertainty with them and at a higher level, teammates' locations and status are not always known. This method cannot be used when the situation need quick exploration of the full environment with the highest priority or full connectivity of the team is required at all times.

Julian de Hoog, and et al. introduced a role-based exploration approach by using a novel way of calculating rendezvous points for robots to meet and share information (Julian de Hoog, S. and et al., 2009). They wanted to use role-based exploration, and the determination of optimal rendezvous points. Calculation of optimal rendezvous points is crucial and can significantly speed up the exploration effort to coordinate efficient meetings between *explorers* and *relays*. Subsequent rendezvous point is calculated by the *explorer* while it is in communication range of the *relay*, and uses thinning on the free space in the map. Now there are a list of potential rendezvous points, which is the best one? They have examined a number of different utilities and combinations such as estimated communication range at the rendezvous point, closeness to nearest frontiers, and path cost. They expected the *relay* to follow the *explorer*, however, it turned out that the most important consideration is the *explorer's* next choice of frontier. More specifically, in their implementation they selected a rendezvous point by considering only a small number of points near the *explorer's* next frontier of choice and choosing the one having highest *neighbourTraversal* value. In case multiple points have equal *neighbourTraversal* values, they selected the one with the best estimated communication range. They have developed their own JAVA-based simulation environment to implement this exploration approach and compare it with other available approaches. They have considered two cases such as *relay* cannot reach rendezvous, and *explorer* cannot reach rendezvous. In the first case, the *relay* finds that it cannot reach rendezvous. It re-computes the next best rendezvous point, reaches this point, and waits, hoping that the *explorer* will find it. In the second case, *explorer* has had his return path to the rendezvous point blocked. In this case, the *relay* can still reach the originally agreed rendezvous point, and waits there. They claimed that the novel rendezvous point calculation method proposed in this paper leads to significantly more efficient exploration than the previous proposed algorithms on rendezvous point calculation. These rendezvous points could be used for replanning when unexpected changes in the environment occur. Extensions of such a method can be found in (Julian de Hoog, S. and et al., 2010), in which a new rendezvous point selection procedure and dynamic team hierarchies are adopted. However, existing cooperative methods that allow robots to go beyond the communication range typically assume that perfect sensor data and localization are available.

One more behavior-based system for sensor placement in unknown environments was proposed in (Jacob H. and Z. Butler). A set of behaviors, or tasks, are used to control the exploration and sensor placement as well as when and how the groups of modules should merge together or split apart to more efficiently achieve their goals. The system that they have developed to control blobs is based off of the ALLIANCE architecture for multi-robot teams. ALLIANCE is a distributed behavior-based architecture, designed to respond well to failure of subtasks or whole robots, and noisy sensors and communication. The implementation of this system takes much from the ALLIANCE architecture. It has a three layered architecture. The lowest layer receives all incoming messages from other blobs. It forwards them unchanged to the higher layers, but some are used to keep track of the status of other blobs and the environment up to date. A general A* path finding algorithm finds the optimal waypoints to that location. In this system each task also implements a number of other functions, such as responding to sensor and feedback input, which affect its internal state. These functions can be responses to specific actions in a lower level, like a blob reaching a destination, splitting or merging, but also include external inputs like receiving a message from another blob, or switching to a different task. This proposed system learned that splitting would not be profitable and eventually kept itself together as a single blob. They also found that the tradeoff between optimality of sensing coverage and efficiency of placement could be effectively managed with a simple threshold parameter.

Another algorithm was introduced based on a population that samples the possible moves of all robots and a utility to select the best one in each time (Martijn N. Rooker, Andreas Birk, 2007). A combination of odometry and sensing is used for rough localization. The odometry is used to determine translational movements of the robot. Exploration of an unknown environment is performed while all the robots in the group maintain communication with each other and with a base station. The utility value will make sure that communication is maintained and also responsible for attracting the robots toward the frontier. To accomplish the recovery from deadlock situations with a role-based approach, it is necessary to define the different roles that are necessary to recover from a deadlock situation. The robots can detect the deadlocks by a lack of progress in the exploration as well as in their movements. Then, they select one robot as a meeting point and the others move there. In doing so, they only traverse the already known and mapped regions of the environment. The moment a deadlock situation is detected, a robot in the group is selected to become the meeting point and it is assigned this role. The selected robot stops its exploring behavior and will remain stationary at its current location. The other robots in the group also change their behavior and are assigned the wanderer role. The robots that are assigned this role are moving toward the meeting point instead of the frontier. The robots can use the joined map and the Manhattan distance transform to find the shortest path for moving toward the meeting point. The robots keep on moving toward the meeting point until all the robots are within a certain distance of the meeting point and within communication range. As soon as this is the case, all robots change back into the role of explorer and the normal exploration process is resumed. Two different versions are tested for the meeting point strategy. They are the random meeting point strategy and the nearest to the frontier meeting point strategy. Roles are used as a remedy for this.

CONCLUSION

In a typical multi-robot system, behaviors are embedded in the control architecture and are intended as building blocks for achieving higher-level goals. In this system, the simple and complex behaviors of each robot are combined to form a group behavior that is both new and desirable. This system provides a tight blend between sensing and action and does not rely on the acquisition of such world models. As such it is a very effective system in the dynamic and unstructured environments in which multiple robots naturally operate. At the same time Role-Based Exploration can be used when, availability of communication is ubiquitous throughout the environment and also the team is composed of robots that have difficulty turning on the spot, or cannot retrace their paths due to environmental factors. Reactive algorithms are important in dealing with uncertainty, and run very quickly since no elaborate couplings are involved. The aim of this paper is to address the most recent developments of MRS by classifying the proposed approaches in terms of a number of features concerning the system organization and specifically focusing on behaviour-based and role-based systems.

REFERENCES

1. Arkin, R. (1998). Behavior Based Robotics. MIT Press.
2. Beeson P., N.K. Jong, and B. Kuipers. (2005). Towards autonomous topological place detection using the extended Voronoi graph, In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA).
3. Berg, J. and Overmars M. H. (2005). Road-map based Motion Planning in Dynamic Environments, in IEEE Transaction on Robotics, vol. 21(5).
4. Burgard W., M. Moors, C. Stachniss, and F. Schneider. (2005). Coordinated multi-robot exploration," Robotics, IEEE Transactions on [see also Robotics and Automation, IEEE Transactions on], vol. 21(3), pp. 376-386.
5. Daniel Schmidt, Tobias Luksch, Jens Wettach and Karsten Berns. (2006). Autonomous Behavior-based Exploration of Office Environments, ICINCO 2006 - Robotics and Automation.
6. Ferranti, E., N. Trigoni, and M. Levene. (2007). Brick&Mortar: an on-line multi-agent exploration algorithm, In Proceedings of IEEE International Conference on Robotics and Automation, pp. 761−767.
7. Gu Fang, Gamini Dissanayake, and Haye Lau. (2004). A Behaviour-Based Optimisation Strategy for Multi-Robot Exploration, In Robotics, Automation and Mechatronics, 2004 IEEE Conference on, volume 2, pages 875 – 879 vol.2.
8. Hoog J., S. Cameron, and A. Visser. (2009). Role-based Autonomous Multi-Robot Exploration, Proc. Of Computation World: Future Computing, Service Computation, Cognitive, Content, Patterns, pp. 482-487.
9. Hoog J., S. Cameron, and A. Visser. (2010). Selection of Rendezvous Points for Multi-Robot Exploration in Dynamic Environments, Proc. Intl. Conf. on Autonomous Agents and Multi-Agent System.
10. Howard, A., Mataric, M. J., and Sukhatme, G. S. (2002). Mobile sensor network deployment using potential fields: A distributed, scalable solution to the area coverage problem.
11. Howie Choset, SeanWalker, Kunnayut Eiamsa-Ard, and Joel Burdick. (2000). Sensor-based exploration: Incremental construction of the hierarchical generalized Voronoi graph. The International Journal of Robotics Research, vol.19 (2), pp.126-148.
12. Howlett, J.K., Schulein, G.,Mansur, M.H. (2004) A practical approach to obstacle field route planning for unmanned rotorcraft, In proceedings of the American Helicopter Society 60th Annual Forum.
13. Jacob H. and Z. Butler. Behavior-Based Control of Self-Reconfigurable Robot Teams for Sensor Placement.
14. Jesus S. Cepeda, Luiz Chaimowicz, Rogelio Soto, J. L. Gordillo, Eden A. Alanıs-Reyes, L. C. Carrillo-Arce, A Behavior-Based Strategy for Single and Multi-Robot Autonomous Exploration.
15. Jones and Maja J Mataric(2005). Behavior-Based Coordination in Multi-Robot Systems, Autonomous Mobile Robots: Sensing, Control, Decision-Making, and Applications.
16. Julian de Hoog, S. Cameron, and A. Visser. (2009). Role-based Autonomous Multi-robot Exploration, proceeding of the International Conference on Advanced Cognitive Technologies and Applications (COGNITIVE), pp. 482-487.
17. Julian de Hoog, S. Cameron and A. Visser,(2010). Selection of Rendezvous Points for Multi-Robot Exploration in Dynamic Environments. Proc. of International Conference on Autonomous Agents and Multi-AgentSystems.
18. Khalid Al-Mutib, AlSulaiman M, Emaduddin, H., and Mattar, E. (2012) D* Lite Based Real-time Multi-agent Path planning in Dynamic Environments, IJERA, vol 2(2), pp. 1414-1419.
19. Latomb, J. (1991). Robot Motion Planning, Kluwer Academic Publishers, Boston.
20. Lee J. and R. Arkin.(2003). Adaptive multi-robot behavior via learning momentum. In *IEEE Conference on Intelligent Robot Systems (IROS-03)*.
21. Martijn N. Rooker, Andreas Birk. (2007). Multi-robot exploration under the constraints of wireless networking, Control Engineering Practice 15, pp. 435–445.
22. Matari´c, M. J. (2009), The Robotics Primer. MIT Press.
23. Miguel Juli´a, Oscar Reinoso, Arturo Gil, M´onica Ballesta and Luis Pay´a. (2010). Behaviour Based Multi-Robot Integrated Exploration, International Journal of Innovative Computing, Information and Control, pp. 1-20.
24. Rekleitis I., Dudek, G., and Milios, E. (1997). Multi-robot exploration of an unknown environment, efficiently reducing the odometry error, proceeding of the International Joint Conference in Artificial Intelligence *(IJCAI)*, vol. 2, pp. 1340-1345.

25. Rekleitis I., V. Lee-Shue, A. P. New and Choset, H. (2004). Limited communication, multi-robot team based coverage, proceeding of the IEEE International Conference on Robotics and Automation, pp. 3462-3468.

26. Senthilkumar, K. S., Bharadwaj, K. K. (2012). Multi-Robot Exploration and Terrain Coverage in an Unknown Environment, Robotics and Autonomous Systems, ELSEVIER, (ISSN: 0921-8890), Vol. 60(1), pp. 123-132.

27. Sorbello R., A. Chella, and R. C. Arkin. (2004). Metaphor of Politics: A Mechanism of Coalition Formation, presented at Workshop on Forming and Maintaining Coalitions and Teams in Adaptive Multiagent Systems.

28. Steven Damera, Luke Ludwiga, Monica Anderson LaPointa, Maria Ginia, Nikolaos Papanikolopoulosa, and John Budenske, Dispersion and exploration algorithms for robots in unknown environments,

29. Szabo, R. (2004). Topological Navigation of Simulated Robots using Occupancy Grid, International Journal of Advanced Robotic Systems, Vol. 1(3), pp. 201 - 206.

30. Timothy Arney. (2007). An Efficient Solution to Autonomous Path Planning by Approximate Cell Decomposition, IEEE ICIAFS07.

K.S. Senthilkumar received his B.Sc. and M.Sc. Degrees in Computer Science from the University of Peradeniya, SriLanka in 1998 and 2003, respectively. Then, he received his M.Tech. Degree in 2006 and Ph.D. Degree in 2011, from Jawaharlal Nehru University, New Delhi, India. He did research on Multiple Robot Terrain exploration and coverage for his Ph.D. Degree in Computer Science. He is a Senior lecturer in the Department of Mathematics & Computer Science, PNG University of Technology, Papua New Guinea. His research area covers Robotics, Artificial Intelligence and Soft Computing. He has published several scientific papers in international conference proceedings and reputed international Journals.