CHAPTER 2

BACKGROUND AND RELATED WORK

2.1 Introduction

Terrain Exploration and Coverage is the problem of driving the footprint of a robot over all the points of a given terrain in an efficient manner. Robot terrain coverage shares many aspects with robot terrain exploration, but exploration usually requires only remote sensing of all the boundaries of the terrain, while coverage requires remote sensing of the entire terrain. Terrain exploration and coverage task is a time-consuming part in many approaches. Therefore, it is important in multi-robot approach to ensure that, additional robots improve the time taken for this task over a single robot approach. It can be assumed that the addition of robots certainly speeds up the exploration and coverage task. However, without coordination, all robots might follow the same path and require the same amount of time like single robot. Sometimes, interaction between robots may cause the task to take more time to complete, compared with a single robot. Therefore, the key idea in multi-robot terrain exploration and coverage is to allow the robots explore and cover different areas of the terrain simultaneously [Bruemmer et al., 2002; Cao et al., 1997; Winfield, 2000; Yamauchi, 1998; Zlot et al., 2002].

Robot architecture defines the configuration of the control system, and is described by using different structures and techniques. Different problems have different requirements, and the right choice of architecture provides
beneficial constraints on design and implementation of the desired problem. Basically, architectural structure refers to how a system is divided into subsystems, and how those subsystems act together. Complexity of a control system can be reduced by decomposing the complete architecture into small parts. Multi-Robot architecture has to do with the use of multiple robots, assisting each other to perform a task that is either too difficult or impossible for one robot to perform alone [Burgard et al., 2006; Carpin & Parker, 2002].

The research in multi-robots naturally extends from research on single robot systems, but is a distinct topic in its own right, due to the complexity of multiple robots interaction.

Almost all the work in multi-robotics began, after the introduction of the new robotics paradigm, based on behavior-based control [Oh et al., 2001]. Prior to this, research had been concentrating on single robot systems with a variety of sensors [Yamauchi et al., 1998]. Multi-robot systems are more complex than single robot systems because the robots may have to navigate in a largely unknown, unpredictable and dynamic terrain. Multi-robot system keeps up the promise of improved performance and fault tolerance for large-scale problems [Arai et al., 2002; Luo et al., 2003]. Communication is an important aspect in multi-robot systems. Generally speaking, more frequent communication provides more cooperation between robots, but requires more time for task completion. On the other hand, less frequent communication slightly speeds up the time for task completion, but produces more problems among robots. This chapter presents a review of researches, relevant to this thesis in the area of terrain exploration, coverage, cellular decomposition, and behavior-based multi-robot architectures.

2.2 Cellular Decomposition

This is a fine grid-based representation of a terrain, introduced by Elfes and Moravac [Elfes & Moravec, 1985]. Cellular decomposition breaks down
the terrain into small cells, so that doing coverage in each cell is easy. The cells are square shaped and equal in size. Exact cellular decompositions [Choset et al., 2000] represent the free terrain space by dividing it into non-overlapping grid cells such that adjacent grid cells share a common boundary, and the union of all the cells covers the entire free terrain. To achieve completeness, most of the single robot coverage planners use cellular decomposition of the terrain. Covering each cell is simple, and provably complete coverage results from ensuring that the robot visits every cell. The cellular decomposition can be established either off-line, which requires knowledge of the environment [Jager & Nebel, 2002], or on-line using on-board sensors [Kleiner et al., 2006].

Choset & Pignon [1997] proposes a method which decomposes the region to be covered into smaller pieces. This method is known as exact cellular decomposition. These pieces are then easily covered by a local coverage strategy, and the collection of pieces is connected by solving a traveling salesman problem. Rekleitis et al. [1997] uses two robots in online closely-coupled coordination, using a visibility graph-like decomposition, similar to exact cellular decomposition. The algorithm uses the robots as beacons to eliminate odometry errors. In another work, Rekleitis et al. [2004] extends the Boustrophedon approach to a multi-robot version. In this algorithm also, robots operate under the restriction that communication between two robots is available only when they are within the line of sight of each other. All the algorithms are not robust to failures and could stop if one of the key robots fails.

There are on-line algorithms using ant-type robots that use a cellular decomposition technique presented in [Batalin & Sukhatme, 2005; Koenig & Liu, 2001; Svennebring & Koenig, 2004]. The area is divided into a grid of square cells on which the exploring ant-type robots leave traces of their
passage, similar to real ants leaving pheromone. Robots tend to move to the least visited cell or in the least recently visited direction.

In the proposed schemes, cellular decomposition technique is used to decompose the terrain as shown in Figure 2.1. This work is based on the previous single robot coverage methods that guarantee complete coverage [Rekleitis et al., 2004; Rekleitis et al., 2005]. White grid cells are unoccupied and grey grid cells are partially or fully occupied by things or walls and are discarded from robot movable area. In this technique, it is assumed that once the robot entering a cell means that it has covered the cell. When robot visits each cell in the terrain, coverage is complete. The decomposition will be done in such a way that if one robot starts from a cell anywhere in the terrain then it can gain access to all the grid cells in the terrain by moving through adjacent cells. The complete coverage is attained by ensuring the robot visiting each cell in the decomposition.

![Figure 2.1: Cellular Decomposition of a Terrain](image)

2.3 The Exploration Problem

"Exploration is an activity to identify the locations of obstacles, objects, and free spaces by sensing the unknown environment".
Exploration is typically used in unknown terrain, where the robots have no idea before about the terrain. The problem of exploration of an unknown terrain has been extensively studied in the past, by using single robot systems with a variety of sensors and also using team of robots [Zelinsky et al., 1993]. The multi-robot version of exploration and coverage is proposed by many researchers [Burgard et al., 2000; Dias et al., 2004; Dias et al., 2005; Ferranti et al., 2007; Senthilkumar & Bharadwaj, 2008a; Senthilkumar & Bharadwaj, 2008b]. Even though there are many similarities in the approaches, they differ fundamentally in the use of live sensor data and stored data [Agmon et al., 2006; Agmon et al., 2008; Yamauchi, 1997; Yamauchi et al., 1998]. Multi-robot exploration extends the research from single robot systems, but it is also a topic in itself, because of the complexity of dealing with multiple robots and new constraints such as uncertainties of a real environment. Another researcher discusses the problem by using distributed robots with sensors in the wireless ad hoc network domain [Winfield, 2000]. He uses a random-walk algorithm to scatter the robot network into the environment to support communication. Most of the work in multi-robot exploration deals with map-building of two-dimensional environments in a decentralized manner and use either range sensors or camera vision [Parker, 2000].

A multi-robot market-based coordination scheme by dividing the task using an auction algorithm, for exploration in an indoor terrain was studied by Zlot and Stentz [2006]. This algorithm takes the advantage of a central unit to do the auction, but it is possible to do auctions locally by each robot. This algorithm is totally distributed and which minimizes traveling costs and maximizes information gain. A centralized exploration and mapping algorithm that uses maximum likelihood to find maps maximally consistent with the sensor data is described by Simmons et al. [2000a]. Burgard et al. [2006] also presents a centralized algorithm using explicit collaboration with
limited communication range. This algorithm trades off travel cost and information gain in order to distribute the robots in the environment.

2.4 The Coverage Problem

"Coverage is the problem of driving the footprint of a robot over all the points of a given terrain in an efficient manner".

The terrain coverage problem is also defined as the construction of a complete map of the terrain by sweeping the surface that finds a single spanning tree. The generated tree can be a composition of multiple single trees that allow the robot(s) to achieve the global task. The coverage problem can be divided into two types: static and dynamic.

Static coverage is the problem of using robot(s) in a static configuration, such that every point in the environment is under the robots’ sensor area at every instant of time. This type of coverage is difficult if the environment is unknown.

Dynamic coverage explores and hence it covers the environment with constant motion and neither settles to a particular configuration nor to a particular pattern of traversal. Exploration is viewed as the initial phase of dynamic coverage in the robotics literature.

The challenge of multi-robot terrain coverage has received considerable attention in the literature, because coverage task is easily incorporate in various domains [Batalin & Sukhatme, 2002a; Rekleitis et al., 2008; Schwager et al., 2006; Senthilkumar & Bharadwaj, 2008c]. And several solutions have been proposed in the literature including single and multi-robot solutions [Choset, 2001; Williams & Burdick, 2006]. In coverage, robots are
typically equipped with some kind of effectors or low-range inspection device and perform cellular decomposition of the environment that is used to plan coverage trajectories. Actuator and sensor errors have not been considered much in the terrain coverage literature, though they seem to play an important role on the performance in many applications. Next paragraphs present a brief overview of a variety of techniques used to solve the robot coverage problem.

Choset [2001] provides a survey of terrain coverage algorithms and distinguishes them as on-line and off-line. The survey further discusses between approximate cellular decomposition, and exact decomposition. An approach that uses simple sonar range sensors to detect critical points was introduced in [Acar & Choset, 2000]. In this approach, a robot can simultaneously cover an unknown terrain and ensure complete coverage while looking for critical points. The continuous covering problem has been studied by Arkin et al. [2000] and discussed a strictly off-line approach.

Gabriely and Rimon [2001; 2002; 2003] have worked on a coverage approach for a single robot that provides optimal paths in a grid-like representation of the terrain. Their algorithm, called Spanning Tree Covering (STC), subdivides the work-area into disjoint cells and then makes a spanning tree. It is a polynomial time coverage algorithm. They developed three versions of the STC algorithm. The first version is off-line, where the robot has perfect prior knowledge of its environment. The off-line STC algorithm computes an optimal covering path in linear time $O(N)$, where $N$ is the number of cells comprising the area. The second version of STC is on-line, where the robot uses its on-board sensors to detect obstacles and construct a spanning tree of the environment while covering the work-area. The on-line STC algorithm completes an optimal covering path in time $O(N)$, and requires $O(N)$ memory for its implementation. The third version of STC is ant-like type. In this version, too, the robot has no prior knowledge of the environment,
but it may leave pheromone-like markers during the coverage process. This ant-like STC algorithm runs in time $O(N)$, and requires only $O(1)$ memory. In this algorithm, cell that are partially covered by obstacles or walls of the terrain are not considered for coverage.

There is another algorithm called as Backtracking Spiral Algorithm (BSA) which is proposed in [Gonzalez & Alarcon, 2003]. One of the main innovations of this algorithm is the use of spiral filling paths instead of zig-zag like paths, which make it robust regardless of the robot initial orientation. BSA uses a small set of rules, which allows it to work properly with low sensorial and computational demands. The first ring of a spiral path is accomplished nearby the obstacles, thus BSA indirectly contributes to improve the performance of a localization system based on the use of the obstacles in the environment as beacons. BSA is an on-line algorithm and it builds incrementally a very simple grid map in order to control the spiral filling procedure and the backtracking mechanism. Therefore, BSA can be used by any terrain acquisition or map building system to visit the whole accessible area and construct a complete model of the environment. Backtracking procedure in the algorithm ensures that the whole surface is covered. They prove that the spiral paths are more efficient and robust to obstacle orientation.

Wagner et al. [1999] proposed discrete coverage algorithms for multiple robots using pheromone-like markers to guide the coverage process. This algorithm also used the markers to do the coordination between several robots called ant-robots. In the implementation, the environment is presented as a graph of nodes, where node is a cell in the grid-world.

Hazon and Kaminka [2008] proposed a variety of algorithms for covering a grid-like terrain using a spanning tree. In that work, they considered the environment to be covered, could be either known or unknown
and the issues of redundancy and robustness to robot coverage were also considered. In Hazon et al.’s [2006] work, each robot constructs parts of a minimal spanning tree of the environment, assuming global communication and localization. Spanning trees are shared such that there is no redundancy occurs if none of the robots fail. Hazon presents a multi-robot coverage algorithm that builds upon the provably complete and optimal STC algorithm for covering grid like environments from [Gabriely & Rimon, 2001]. The algorithm considers optimal spanning trees of known environments that are divided among the robots based upon their initial position on the spanning tree. One of the algorithms proposed by Hazon & Kaminka [2005] focuses on the robustness of the solution, so that even if only one robot remains in operation, it will be able to carry on and complete the exploration task. In a recent paper by Hazon & Kaminka [2008], STC was generalized to Multi-Robot Spanning Tree Coverage (MSTC), a polynomial-time multi-robot coverage heuristic. This is an offline algorithm for multi-robot coverage. MSTC first computes the same spanning tree as STC, and considers the tour that circumnavigates the spanning tree. Each robot follows the tour segment clockwise ahead of it. To improve the cover time, the longest segment is divided evenly between the two adjacent robots.

Zheng et al. [2005] proposed an offline multi-robot coverage algorithm. Their solution is based on dividing the given spanning tree into k sub-trees, where there might exist path overlapping between the robots. But their solution is not robust and also, they pointed out that, different choices of trees may result in different coverage time. McLurkin & Smith [2006] did an experiment on multi-robots, where 108 Swarmbot1 robots were dispersed in an indoor environment by relying on local communication. The fully scalable and distributed algorithm can maintain the connectivity of the swarm, and allows the swarm-robots to navigate by using information about the network topology.
Butler et al. [2001] introduced a sensor-based multi-robot complete coverage algorithm. This covers unknown rectilinear environments using a square shaped robot with contact sensing. This algorithm performs an on-line decomposition on the terrain, where rectangular shaped cells are formed in such a way that the terrain could be covered completely by doing back-and-forth motions parallel to one of the walls of the terrain. They have also extended their work to multiple robots and unrestricted communication is assumed among the robots in their work. There is a supervisor robot actually performs an operation in such a way that coverage can continue under the direction of the single robot algorithm, but coverage is done by multiple robots. Butler presents his algorithm requiring only bumper sensors and hence the implementation is limited to rectangular environments.

Kurabayashi et al. [1995] proposed an off-line, centralized multi-robot coverage algorithm for sweeping a known area with the ability to plan for dynamic objects. In this algorithm, first the complete path is planned off-line, and the area is divided among robots. Secondly, the location of the dynamic object is examined, and the optimal path is estimated. They used a technique to smartly identify relocation of dynamic objects to improve the efficiency of the algorithm. This algorithm uses exact cellular technique. They did not provide any guarantee for robustness. Luo and Yang [2002] used neural network to represent the environment in their coverage algorithm. Each neuron corresponds to a cell in the occupancy grid, and the activity in each neuron represents the cell, which is occupied, unknown, or covered. They have demonstrated their approach by using multiple robots covering an environment in simulation only.

Solanas and Garcia [2004] addressed the coverage problem by first exploring an unknown space and then dividing the area into a number of cells equal to the number of robots. This is an unsupervised clustering algorithm
that partitions the unknown space into as many cells as possible. The allocation of regions to various robots is based on bids that are estimates of information gain traded-off against traveling costs to that region. Kong et al. [2006] provides an improved algorithm for multi-robot coverage with unbounded communication, where the algorithm is demonstrated to be robust to failures.

2.5 Robot Architectures

In the robot literature, the robot architecture is categorized into three types such as hierarchical (or deliberative), reactive (or behavioral), and a hybrid of these two models. These three models are based on three basic tasks namely, sensing, planning and acting. The hierarchical model shown in Figure 2.2 is the oldest model used in robot implementations. First the robot senses and constructs a global map of the environment, then plans all the actions, needed to reach the target and finally acts according to the plan. This chain is repeated until the task is completed. Hence, this model is naturally slow.

![Deliberative Architecture](image)

Figure 2.2: Deliberative Architecture

The second model is called reactive architecture and it was introduced to interact with environmental changes. The architecture model is shown in Figure 2.3, in which software modules are known as "behaviours" and they run simultaneously. This model contains two tasks: sensing and acting, and this will be repeated until the task is completed. This behavior-based architecture is rooted in biological inspirations of ants, bees, and birds. Hence, we can say that, this behavior-based model influences strongly in much of the multiple mobile robots researches. Behaviours are a direct mapping of sensor input into actuator output and are easy to implement too. Work in this area has
demonstrated the ability for multi-robot teams to gather, scatter, forage, and follow trails.

![Figure 2.3: Reactive Architecture](image)

The third model is a hybrid of reactive and hierarchical models, which is shown in Figure 2.4. In this model, long term control planning is carried out by the hierarchical layer and the basic functions are done by the reactive layer. The hybrid architecture provides for an abstraction of navigational tasks, for example path planning can be done by the hierarchical layer and at the same time obstacle avoidance and moving can be done by the reactive layer of the model.

![Figure 2.4: Hybrid Architecture](image)

### 2.6 Multi-Robot Approaches

The field of multi-robot was introduced in the late 1980's when researchers started investigating issues in multiple mobile robot systems [Liu & Wu, 2001]. Before the multi-robot system was introduced, most of the researches had focused on either single robot systems or distributed problem solving without robotic components. In recent years several Multi-robot approaches have been proposed [Ichikawa & Hara, 1999]. These approaches are usually classified as centralized vs. decentralized architecture; swarm vs. intentional cooperation; explicit vs. implicit communication; and
homogeneous vs. heterogeneous robots. The definitions of these types are not presented as a means of categorizing the approaches, but are used to introduce the general ideas of approaches, and used by many researchers, in the field. Several different combinations of these approaches are presented in the following sections.

2.6.1 Centralized vs. Decentralized Approaches

In recent past, several Multi-robot approaches for exploration purposes have been proposed and usually classified as centralized and decentralized. Centralized approaches are suitable for a small number of robots only, and they are not fault tolerant since the entire system will fail when the central entity fails, or in case of disconnections. Decentralized approaches are flexible, scalable and robust, but frequently achieve this at the cost of suboptimal solutions, compared to those of centralized approaches [Carpin & Parker, 2002].

All multi-robot approaches have a tendency to focus on providing a specific type of capability to the distributed robot team. Capabilities include task distribution [Pagello et al., 2003; Stone & Veloso, 1999], swarm control, fault tolerance [Parker, 1998], and so forth. Distributed systems such as chemical plants or nuclear reactors deal with the real world, but operate under well-known constraints. Centralized approaches assume global communication while distributed approaches assumes local communication. It is difficult to compare the performance of multi-robot architectures due to the differences in the hardware or the experimental scenario of the architectures. A review of relevant centralized and decentralized approaches is presented in the following subsections.

Centralized Approaches: these approaches try to scatter or distribute the robots throughout the environment to achieve efficiency [Burgard et al, 2006].
Distribution of robots is suitable for small teams when global communication is available. In real environments, it is important to consider signal limitations, because robots might not be able to communicate with each other, if they scatter beyond their communication range. Thrun [2000] designed an algorithm using a Monte Carlo localizer with maximum likelihood function on grid maps. The maximum likelihood function determines the best alignment of laser scanned data. In this approach a team leader merges the maps and shares it with the rest of the team.

Burgard et. al. [2002] presented a frontier-based exploration by allowing a robot to calculate the tradeoff between the cost and the utility of reaching a frontier cell. In this approach the central system computes the frontier cells based on a utility value for each location. The cost is determined by how much effort it puts to reach a frontier cell while the utility value is determined by the expected travel cost to reach the frontier cell. The algorithm is implemented using two or three robots in a large simulated area. When one robot is assigned to some location, the information gain of the location cells is decreased. Based on this approach Solanas and Garcia [2004] proposed a clustering technique that allows faster partial coverage of the environment.

A market-based approach was proposed by Dias and Stentz [2001]. In this approach each robot set up a set of goals in areas where little information is known and robot produces a tour containing several goals. The tours are auctioned and the robots submit bids, based on the distance that they have to travel to cover the tour. Each robot sends its information about recently explored area to a central agent and the central agent merges the areas and creates a global map. Simmons et al. [2000b] implemented an algorithm to merge maps and coordinate the robots using a central unit. Each robot creates a local map, and sends the information to a central unit which develops the map by combining data from the robots. For example every tenth map is sent...
to a remote computer and used for global map-building and path-planning tasks. The bids created by each robot are sent to and processed by the central unit which assigns tasks to each robot. The global map is used to decide where the robots should explore in the next time step. A team of three robots were used for experiments.

In all these centralized approaches, robots frequently send grid maps information to a central unit, and strongly depend on the communication system and having a single point of failure too. Furthermore, as the number of robots in a team increases the communication bandwidth too becomes a bottleneck for this kind of system.

**Decentralized Approaches:** A great deal of research in decentralized approach in robotics has focused on the development of architectures, task planning capabilities, and control. In this type of approach communication between robots is necessary to achieve coordination and each robot is responsible for distributing its information. Within the robot network, information is distributed to the point where all the robots share a common model of the environment. Decentralized systems are flexible and robust, but frequently achieve considerably sub optimal solutions, compared to those of centralized systems [Burgard et al., 2006].

Asama [1991] use a reactive control architecture called ACTRESS robotic system, which is a decentralized multi-robot system, designed for the maintenance tasks in nuclear power plants. ACTRESS is shown in simulation with the cooperative task of two mobile robots pushing boxes to the sides of a room. The architecture has multi functions, and series of rules are applied for communicating in an attempt to agree movement. Their architecture deals with multiple tasks that must be accomplished in real-time, in applications that consist of a large number of tasks, relative to the number of available robots.
Although this is an expensive and bulky approach, it allows a large amount of information, such as global maps, to be passed on and processed. They used two types of planner, namely, priority-based task allocation planner and motion planner. Illustration is done in simulation.

Yamauchi [1998] proposed an approach, in which robots exchange their grid maps and continuously update their own map by merging the map received, with their local maps. This approach uses frontier-based exploration to direct the robots to the areas that are likely to provide the most new information about the terrain. Multiple robots move to the same frontiers, which is a highly inefficient part of this approach. An algorithm developed by Arkin and Diaz [2002], maintains line of sight communications between robots while searching for a hazard with different apriori knowledge of the environment with few obstacles. One robot serves for the rest of the robots and the exploration area is limited by the total communication range.

Konolige et al. [2004] proposed an approach where the robots initially explore the environment by themselves. When two or more robots come within the range of each other, their maps are merged. The leader coordinates all the robots, builds a complete map that represents the data collected by all the robots, and broadcasts the map often to all the robots in the range. The robot keeps exploring the space in an autonomous fashion when it loses its communication with other. The author remarks that maps are built less efficiently when robots cannot communicate with each other. Another algorithm similar to Arkin and Diaz’s algorithm [2002] is introduced by Antonelli et al. [2005]. They used idealized RF communication model and the robots move simultaneously, and an expected signal is predicted for close locations only. This work investigates the implementation of a wireless mobile ad-hoc network to guarantee that an autonomously driven mobile vehicle remains connected to a limited-coverage base antenna during its motion. The
environments contain only a few randomly placed obstacles and the explored area is limited by the total communication range.

2.6.2 Swarm vs. Intentional Cooperation

It is often difficult to distinguish between swarm and intentional cooperation approaches because many examples of applications have characteristics from both categories. The swarm-type approach deals with a large number of lower-level robots that are typically unaware of each other’s actions and there is only global cooperation [Parker, 1998]. On the other hand, each robot’s interaction with the environment or other robots is purposeful in intentional cooperation, and there is local and global cooperation. Also, in intentional cooperation approaches, there are often numerous goals to achieve in a logical or optimal order.

Hutin et. al. [1998] take an intentional cooperation approach in two simulated experiments using homogeneous robots. Each experiment, such as exploration, mapping and leader following, uses five agents. A combination of both swarm and intentional cooperation is used by Halme [1993] in a simulated stone-collecting experiment. In this experiment, they use two different types of robots namely, work units which collect stones using implicit cooperation and support units, which carry energy to the work units using intentional cooperation.

Mataric et al. [1995] defines the cooperation as explicit and implicit cooperation. Explicit cooperation occurs as a result of one agent performing actions to benefit another agent’s goals. In contrast, implicit cooperation occurs as a result of self motivations that help an agent to achieve its own goals. So swarm-type approaches take advantage of implicit cooperation, whereas intentional cooperation approaches require explicit cooperation.
2.6.3 Explicit vs. Implicit Communication

There are two distinct types of communication that are commonly used, in many multi-robot systems and are usually referred to as explicit and implicit [Parker, 2000].

"Explicit communication occurs with the transmissions, while implicit communication occurs through an awareness of the side-effects of other actions."

As an example, consider two robots unloading a truck, one robot is taking boxes off the truck and the other is stacking them in a storage location [Parker, 1998]. If the stacking robot running out of boxes to stack, it might realize, through implicit communication, that either the unloading robot is having trouble or that the truck is empty. Also stacking robot could distinguish between these two possibilities if a blinking light was triggered when the truck became empty. This would be explicit communication.

Explicit communication was used by Mataric et al. [1995] by equipping two box-pushing robots with radio communication. They exchanged "my turn, your turn" messages along with each robot's sensory data and are able to successfully push a box towards a moving infrared-emitting source. In another work, Dadios and Maravillas [2002] use implicit communication in a team of two soccer-playing robots. They use fuzzy logic and an overhead camera for navigation. The overhead camera is used by both robots for implicit communication, but the robots are not allowed to communicate explicitly with each other. In the simulation the opposing team is represented by static obstacle and the robots are able to effectively pass and shoot the ball into a goal. An explicit communication technique is used by Asama [1991], by putting laptops, equipped with wireless modems, on each mobile robot. This approach allows a large amount of information, such as
global maps, to be passed and processed. But it is a quite expensive implementation.

Previous research work in distributed approaches for exploration and coverage can be also classified as insensitive and sensitive communication approaches. Sensitive communication approach is that where the robot tries to be in communication range with at least one robot, during the whole process. Insensitive communication architecture allows robot to explore the space on its own. Latimer et al. [2002] and Reikleitis et al. [2004] proposed sensitive communication approach where robots methodically sweep the unknown environment by advancing in close line formations.

2.6.4 Homogeneous vs. Heterogeneous Robot Types

Research in multi-robot can be further categorized according to the type of robots that are used, as homogeneous or heterogeneous.

"The identical robots running identical code, is called homogeneous robot. Whereas, the heterogeneous robots, which may run different codes and often have different sensing and/or manipulation capabilities".

Swarm-type approach uses homogeneous robots, because it allows faster multi-robot design, whereas intentional coordination approach uses heterogeneous robots [Howard et al., 2006], because heterogeneous robots allow specialized tasks. Homogeneous robots are used in many intentional cooperation approaches too. Dadios & Maravillas [2002] and Mataric et. al. [1995] proposed an intentional cooperation approach with two homogeneous robots. Dadios applies on soccer-playing robots performing a pass-shoot task while Mataric applies on two six-legged box-pushing robots. Mataric uses a group of twenty homogeneous robots in a swarm approach to show the benefits of increased awareness during a homing behavior. She gives results
from three experimental cases, namely, ignorant coexistence, informed coexistence, and intelligent coexistence.

2.7 Behavior-Based Robotics

Behavior is a biologically motivated term, defined as regularity in the interaction of a robot with its environment and the behavior need not be complex. Behavior-based robotics is the foundation for much of the multi-robot work over the past fifteen years [Mataric, 1997; Pagello et al., 1999]. Examples of simple behaviors include wandering, obstacle avoidance, aggregation, dispersion and following. Wandering is a behavior that keeps a robot in motion. Avoidance behavior keeps robots from hitting obstacles and other robots. Aggregation and dispersion enable robots to form a group or to spread out. The following is a behavior that is useful to avoid collisions. Many researchers in the field of multi-robot have attempted to mimic the behaviors, observed in the societies in robots [Parker, 2000].

After Rodney Brooks [1986] proposed a new method known as subsumption architecture, for controlling autonomous mobile robots, behavior-based robotics became popular. Presently, Behavior-based strategies follow the subsumption approach and generally fall between planner-based and purely reactive approaches [Choset & Pignon, 1997]. And also, behavior-based approaches are distributed and decentralized. The architecture introduced by Brooks allows for increasing levels of competence that run concurrently and whose results are joined to form a single action. Each level of competence forms a layer of the control system. In multi-robot systems, the simple and complex behaviors of each robot are combined to form a group behavior that is both new and desirable.

Mataric divides the combination of behavior-based robotics into two categories: temporal combination and direct combination and performs many
experiments on these combinations [Mataric et al., 1995]. The temporal combination approach executes only one behavior at a time and switches between them in a more reactive manner. And the direct combination approach executes multiple behaviors concurrently in a strict subsumption style. The temporal combination approach is not a strict reactive, since identical sensor readings produce different results, depending on the current and recent behavior.

The ALLIANCE architecture based on a strict subsumption-style approach has been developed by Parker [1998]. He introduced two mathematically-modeled parameters called impatience and acquiescence and these parameters work together to solve hardware malfunction problems. This architecture is non task-specific and designed for maximum fault tolerance. Kube et. al. [1993] propose an Adaptive Logic Network (ALN) approach with a set of four behaviors for behavior arbitration. They pointed out that the subsumption architecture only works well when the interaction and complexity of behaviors are low. Their swarm approach is tested on a simulated group of box-pushing robots in which the robots are manually guided through the task in order to train the ALN’s. The four behaviors in this approach are: find, avoid, slow and goal. The find behavior keeps the robot in motion; avoid behavior keeps it from colliding with other robots; slow behavior allows the robot to push the box without causing damage to it; and goal behavior keeps the robot moving the box towards a goal.

2.8 Ant-type Robots

"Ant-type robots are robots with limited sensing and computational abilities".
Recently most of the multi-robot coverage algorithms are for robots, that interact and plan only locally [Eliyahu et al., 2008; Hahnel et al., 2004], called ant-type robots [Kobe & Bonabeau, 2000; Koenig et al., 2001; Wagner & Bruckstein, 2001; Wagner et al., 2008]. The ant-type robots do not even need to communicate with each other except via the marks. It is sufficient for them to have a limited sensing and computational capabilities and also they need not know or learn the map of the terrain. They only have to leave marks on the terrain, sense marks at their neighboring cells, and change the marks of their current cell. These robots have the advantages that they are easy to design, program and cheap to build.

The ant's energy equalizing ability was used by Halme [1993], and their approach allows two robots to equalize their energies, when they meet. In the simulation of stone collecting task, this ability allows more efficient use of energy and results in less failure robots. Kube and Bonabeau [2000] study ant robots that cooperate to move large body which is mimicked in a group of box-pushing robots. The ants systematically rearrange their positions until the body can be lifted and transported back to the bed.

Ant-type robot is also used in several methods, proposed by many researchers for coverage problem [Ge & Fua, 2005; Menezes et al., 2007]. These robots cannot use conventional planning or coordination methods due to their limited sensing and computational capabilities which limit their planning capabilities even for simple planning tasks such as path planning or coverage of terrain. Thus, they might not be able to cover terrain as efficiently as robots with more powerful sensing and computational capabilities. On the other hand, groups of ant robots can take advantage of both their fault tolerance and their parallelism. Since, this problem is NP complete, it becomes necessary to consider heuristics for solving it. The experiments show that real-time search
methods robustly cover terrain even if the ant robots are moved without realizing that, some ant robots fail, and some marks get destroyed.

Wagner et al. [2000] proposes many multi-robot ant-based algorithms, which use approximate cellular decomposition. The algorithms involve no direct communications, and use simulated pheromones for communications. Some of these algorithms solve only the discrete coverage problem whereas, some others offer complete, robust coverage. Kube and Bonabeau [2000] have also looked at ant-like movements for robot exploration where a leader is elected and the remaining robots follow the leader along the arbitrary path.

Svennebring and Koenig [2004] present a feasibility study on an ant-type multi-robot coverage algorithm where robots leave traces on the environment. These traces can be used by other robots for implicit collaboration. Experiments are being performed with real ant-robots and large-scale simulations. The algorithm did not provide guarantee for completeness, but shows robustness.

Mobile robots have been built to leave short-lived markings in the terrain such as heat traces, or alcohol traces. These results demonstrate that terrain coverage with real-time search methods is an interesting alternative to more conventional terrain coverage methods. Other researchers have also been inspired by ants but they study very different methods in the context of “ant colony optimization” to solve discrete optimization problems [Di Caro & Dorigo, 1998]. Koenig and Liu [2001] studied a simple means for coordinating team of simple agents called ant robots. A simulation study was presented which demonstrated that terrain coverage with real-time search methods was an interesting alternative to more conventional terrain coverage methods. Their results also expose some trade-offs between the different real-time search methods.
Batalin and Sukhatme [2002b; 2003a; 2003b; 2005; 2007] have done plenty of work using radio beacons to guide the navigation of robots and assist them in the coverage of an unknown terrain. They spread the robots in the environment and communication is limited to visual contact only. But the algorithm did not guarantee a complete coverage of terrain. In this work, it was aimed at achieving optimal coverage area, and did not prove any formal statement regarding optimality of coverage time. Chaimowicz et al. [2005] present a distributed control approach for multi-robots based on implicit functions for deploying a system in the environment. Four robots were used to validate the approach and the controller is convergent. Latimer et al. [2002] engaged an algorithm based on the single robot cellular decomposition approach, spreading the robots throughout the environment and allow parallel coverage with finer granularity. In this approach many robots did repeat coverage, because this approach only allowed communication between robots, in physical contact with each other.