14

Distributed Control of Multi-Robot Deployment Motion

Yu Zhou
State University of New York at Stony Brook
The United States

1. Introduction

A multi-robot system is a collection of mobile robots, each of which is equipped with onboard processors, sensors and actuators and is capable of independent operation and individual autonomous behaviours, collaborating with one another through wireless communications or other forms of interactions to fulfil global goals of the system. The mobile robots bring mobility, sensing capability and processing capability to the system; while a communication network is established among the robots to support data delivery and facilitate collaboration.

Multi-robot systems have higher flexibility, efficiency and reliability than single robots: a team of collaborative robots can accomplish a single task much faster, execute tasks beyond the limits of single robots, perform a complex task with multiple specialized simple robots rather than a super robot, and provide distributed, parallel mobile sensing and processing; a group of robots with heterogeneous capabilities can be organized to handle different tasks; the fusion of information from multiple mobile sensors helps to reduce sensing uncertainty and improve estimation accuracy; and the system function is less influenced by the failure of any individual robot.

Multi-robot systems have numerous applications, from regular civilian tasks, such as surveillance and environment monitoring, to emergency handling, such as disaster rescue and risky material removal, from scientific activities, such as space and deep sea exploration, to military operations, such as de-mining and battle field support, and to largescale agricultural and construction activities. Many applications require a multi-robot system to rapidly deploy into a target environment to provide sensor coverage and execute tasks while maintaining communication connections, and promptly adapt to the changes in the system, environment and task. This imposes significant requirements and challenges on the deployment control of the involved multi-robot systems.

Multi-robot deployment has become a fundamental research topic in the field of multi-robot systems. Both centralized and distributed schemes have been proposed in the literature. In general, centralized control depends on a leading robot to collect the state information of all the member robots, tasks and environment and to determine the appropriate motion of each individual robot. It helps to achieve globally optimal deployment and can be very effective in stable environments. However, centralized processing imposes high computational complexity on the leading robot and makes the multi-robot system vulnerable to the failure

Source: Motion Control, Book edited by: Federico Casolo,
ISBN 978-953-7619-55-8, pp. 580, January 2010, INTECH, Croatia, downloaded from SCIYO.COM

of the leader. Moreover, real-time centralized control of multiple robots requires very high communication throughput which is difficult to achieve with the current wireless communication technology. As a result, centralized control has difficulties in adapting to dynamic environments and scaling to large multi-robot systems. Alternatively, distributed control allows each member robot to determine its motion according to the states of itself, its local environment and its interactions with nearby robots and other objects. Distributed processing largely reduces the computational and communication complexities. As a result, distributed control is highly scalable to large multi-robot systems and adaptive to unknown and dynamic environments and changes in multi-robot systems. With properly designed distributed control laws, the desired global goal of a multi-robot system can be achieved as the combined outcome of the self-deployment motion of individual robots.

Recognizing its advantages, we focus our discussions in this chapter on distributed control of multi-robot deployment motion, with the objective to form and maintain sensor coverage and communication connections in a target environment. Section 2 will provide a review of some representative existing distributed multi-robot deployment control schemes. Although distributed multi-robot deployment has received a substantial amount of attention, there has not been enough effort made to address their implementation on realistic robot systems, in particular to explicitly take the kinematic and dynamic constraints into account when determining the deployment motion of individual robots. This disconnection between the control algorithm and physical implementation may degrade the operational effectiveness and robustness of these multi-robot deployment schemes. We will introduce a novel distributed multi-robot deployment control algorithm in Section 3, which takes into account the limited ranges of robot sensing and communication, and naturally incorporates the nonholonomic constraint arising among wheeled robots into individual robots’ equation of deployment motion. Simulation results will be reported in Section 4, which proves the effectiveness of the proposed scheme. Section 5 will summarize the proposed scheme and discuss future work.

2. Review of distributed multi-robot deployment schemes

Due to its distributiveness, adaptability and scalability, distributed multi-robot deployment control has attracted a substantial amount of research effort. Here we review some related works on this topic.

One major category of distributed multi-robot deployment control schemes are based on artificial potential or force fields. Parker developed a two-level approach to deploy a homogeneous multi-robot system into an uncluttered environment to observe multiple moving targets (Parker, 1999; Parker, 2002). The low-level control is described in terms of force fields attractive for nearby targets and repulsive for nearby robots. The high-level control is described in terms of the probability of target existence and the probability of a target not being observed by other robots. The summation of the force vectors weighted by the high-level information yields the desired instantaneous location of the robot. The robot’s speed and steering commands, which are the functions of the angle between the robot’s current orientation and the direction of the desired location, are computed to move the robot in the direction of the desired location. Reif and Wang proposed a “social potential field” method for deploying very large scale multi-robot systems containing hundreds even thousands of mobile robots (Reif & Wang, 1999). Inverse-power force laws between pairs of robots or robot groups were defined, incorporating both attraction and repulsion, to reflect the social relations among robots, e.g. staying close or apart. An individual robot's motion is controlled by the resultant artificial force imposed by other robots and other components of the system. The resulting system displays social behaviors such as clustering, guarding, escorting, patrolling and so on. Howard et al. presented an algorithm for deploying a mobile sensor network in an unknown environment from a compact initial configuration, based on an artificial potential field in which each node is repelled by both obstacles and other nodes (Howard et al., 2002). Poduri and Suktame presented a deployment algorithm for mobile sensor networks to maximize the collective sensor coverage while constraining the degree of the network nodes so that each node maintains a number of connected neighbors, where the interaction between nodes is governed by the repulsive forces among nodes to improve their coverage and the attractive forces to prevent the nodes from losing connectivity (Poduri & Suktame, 2004). Popa et al. proposed a potential field framework to control the behavior of the mobile sensor nodes by combining navigation, attracted by goals and repulsed by obstacles and other nodes, and communication, attracted by maximum communication capacity and avoiding exceeding communication range (Popa et al., 2004). Fan et al. presented a potential field method to ensure the communication among the robots belonging to a formation by adding to each robot one attractive communication force generated by topologically nearby robots (Fan et al., 2005). Ji and Egerstedt presented a collection of graph-based control laws for controlling multi-agent rendezvous and formation while maintaining communication connections, based on weighted graph Laplacians and the edge-tension function (Ji & Egerstedt, 2007).

Closely related, Lam and Liu presented an algorithm for deploying mobile sensor networks such that the network graph approximates the layout of an isometric grid, under the force field defined by the difference between current and ideal local configurations (Lam & Liu, 2006). Jenkin and Dudek presented a distributed method to deploy multiple mobile robots to provide sensor coverage of a target robot (Jenkin & Dudek, 2000). It is formulated as a global energy minimization task over the entire collective in which each robot broadcasts its current position in the target-based coordinate system and moves in the gradient descent direction of its local estimate of the global energy. Butler and Rus presented two event-driven schemes to deploy mobile sensors toward the distribution of the sensed events (Butler & Rus, 2003). In one method, the sensors do not maintain any history of the events, and the robot position is determined by the positions of events like a potential field. In the other method, event history is maintained as a cumulative distribution of events by the sensors for more informed decisions about where to go at each step. With the intention to reduce the communication complexity, Tan presented a distributed self-deployment algorithm for multi-robot systems by combining potential field method with the Delaunay triangulation, which defines the potential field for each robot based on only the one-hop neighbors defined by the Delaunay triangulation (Tan, 2005).

Other than potential/force field methods, Cortes et al. defined the coverage problem as a locational optimization problem, and showed that the optimal coverage is provided by the centroidal Voronoi partitions where each sensor is located at the centroid of its Voronoi cell (Cortes et al., 2004). A gradient decent algorithm is presented to lead the sensor locations converge to the centroidal Voronoi configurations. A similar centroidal Voronoi diagram-based deployment was presented in (Tan et al., 2004). Jiang presented a slightly different method based on the r-limited Voronoi partition (Jiang, 2006). Schwager et al. proposed an adaptive, decentralized controller to drive a network of robots to the estimated centroids of their Voronoi regions while improving sensory distribution over time (Schwager et al., 2007). For this category of methods, local minimum is a potential problem. That is, the robots may be stuck at some Voronoi centroids determined by local configuration and cannot achieve the desired configuration.

Diffusion-based multi-robot deployment schemes were also proposed. Winfield presented a distributed method that deploys a group of mobile robots into a physically bounded region by random diffusion (Winfield, 2000). Kerr et al. presented two physics-based approaches for multi-robot dynamic search through a bounded region while avoiding multiple large obstacles, one based on artificial forces, and the other based on the kinetic theory of gases (Kerr et al., 2005). By mimicking gas flow, the agents will be able to distribute themselves throughout the volume and navigate around the obstacles. Along the same line, Pac et al. proposed a deployment method of mobile sensor networks in unknown environments based on fluid dynamics, by modeling the sensor network as a fluid body and each sensor node as a fluid element (Pac et al., 2006). These methods are designed for continuous sweeping-like coverage, but not suitable for converging multi-robot deployment.

Besides, Bishop presented a method which distributes the functional capability of a swarm of robots to a number of objectives (Bishop, 2007). His method is based on the definition of the capability function of each robot. The primary task (functional coverage) controller is defined based on the definition of the swarm-level objective function. The secondary task (e.g. obstacle avoidance, maintaining of line of sight) is carried out in the null space of the primary task. The potential problems with this method include local minima of the secondary functions and possible incompatibility of the secondary task with the null space of the primary task.

In addition, Jung and Sukhatme addressed the problem of tracking multiple targets using a network of communicating robots and stationary sensors (Jung & Sukhatme, 2002). Their region-based approach controls robot deployment at two levels. They divided a bounded environment into topologically simple convex regions. A coarse deployment controller distributes robots across regions based on the urgency estimates for each region. A target-following controller attempts to maximize the number of tracked targets within a region.

Existing works on distributed multi-robot deployment mostly focus on general schemes. There has not been sufficient attention paid to their implementation on realistic robot systems, e.g. most of existing methods assume reliable information broadcasting among robots to facilitate self-deployment control and multi-robot coordination, which is in fact communication intensive and has reduced robustness in large multi-robot systems. In particular, various kinematic and dynamic constraints must be taken into account in order to determine physically-realizable deployment motion of individual robots. However, there is a lack of a natural framework to incorporate them into deployment control. A very limited number of works have considered kinematic and dynamic constraints explicitly. In general, the dynamic constraints of maximum velocity and maximum acceleration are accommodated by enforcing the computed above-limit acceleration and velocity into the desired ranges (Howard et al., 2002; Jiang, 2006), and the nonholonomic kinematic constraints are ignored by assuming that the robots have holonomic drive mechanisms, i.e. they can move equally well in any direction (Howard et al., 2002; Bishop, 2007). This disconnection between the control algorithm and physical implementation may cause the computed deployment motion unrealizable, and therefore degrades the effectiveness and robustness of the deployment of realistic multi-robot systems.

As an important step towards solving this problem, we propose a novel distributed multirobot deployment control algorithm for deploying a team of mobile robots to establish sensor coverage while maintaining communication connections over a target environment. It takes into account the limited ranges of robot sensing and communication, and in particular naturally incorporates the nonholonomic constraint which arises among wheeled robots. Each member robot self-deploys based on the state of its neighborhood and approaches the desired neighborhood configuration. The resulting local coverage in the neighborhoods of all the robots altogether forms a global coverage of the multi-robot system over the targeted environment.

3. Proposed distributed control algorithm for multi-robot deployment

Targeting to develop a robust distributed multi-robot deployment control scheme which can be reliably implemented in realistic multi-robot systems, we have recently initiated an alternative scheme for distributed multi-robot deployment (Zhou & Tan, 2008; Zhou, 2008). In the following sections, we will provide a extended and detailed description of our original distributed multi-robot deployment algorithm for establishing sensor coverage while maintaining communication connections over targeted environments, based on a lumped dynamics model of involved robots, accommodating the limited robot sensing and communication ranges, and incorporating the nonholonomic kinematic constraint which arises in wheeled robots.

3.1 Objective and assumptions

The objective of the proposed distributed multi-robot deployment algorithm is to deploy a multi-robot system into a targeted environment to form and maintain reliable sensor coverage and communication connections.

To achieve the goal, it is required that

1. The team of robots, each of which has limited independent sensing capability, must collaborate with one another to form and maintain sufficient sensor coverage at the multi-robot system level.
2. The team of robots must maintain globally networked communications at the system level at any time in order to guarantee the information delivery and data sharing in the multi-robot system.

In order to deploy the involved multi-robot system to form and maintain reliable sensor coverage and communication connections in the targeted environment, we propose an overall control strategy as guiding the multi-robot system to approach a desired global deployment configuration which is defined as that each member robot maintains desired distances with nearby objects, including other robots, obstacles and the boundary of the environment.

To facilitate the discussion, we assume that (Fig.1)

1. A group of N mobile robots (nonholonomic or holonomic) are to be deployed into a 2D environment in order to provide sensor coverage while maintaining communication connections, where R_i denotes the ith mobile robot.
2. The underlying environment is a general 2D region which contains objects other than mobile robots, known in general as obstacles (stationary or moving) where O_i denotes the ith obstacle, and can be confined with a boundary denoted by B.

3. Each robot R_i has limited sensing and communication capabilities, represented approximately by a limited circular sensing range with a radius of rs_i and a limited circular communication range with a radius of rc_i respectively. Though the robot sensing range and can be determined by off-line sensor calibration, multiple factors can affect the robot communication range, e.g. the obstacles and humidity. Therefore, varying rc_i is admitted.
4. Each robot R_i updates and maintains a record of its pose (position and orientation) with respect to a global reference frame defined in the environment. The robot self-localization can be accomplished using either relative localization techniques, e.g. odometry and inertial navigation, which is based on the integration of incremental motion, or absolute localization techniques, e.g. GPS, which is based on the measurement of external references, or a fusion of both (Borenstein et al., 1997).

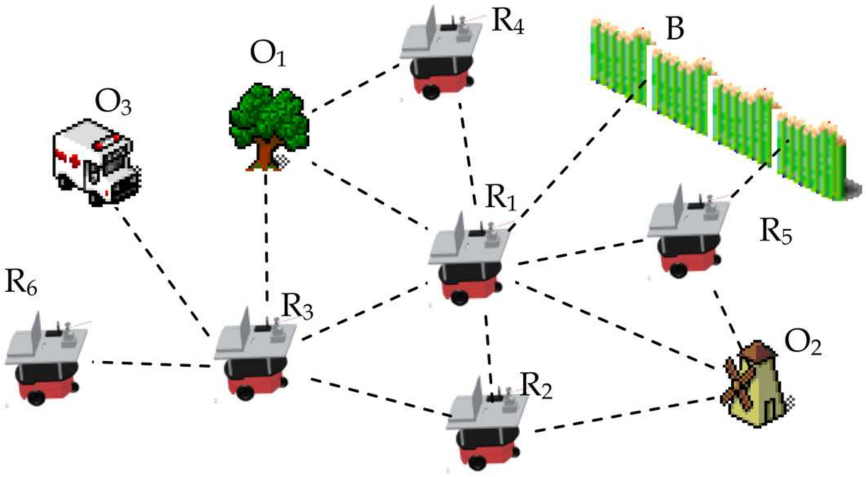

Fig. 1. Multi-robot deployment in a general 2D environment

3.2 Desired deployment configuration

In our scheme, a desired global deployment configuration is defined as such that each member robot maintains desired distances with nearby objects, including other robots, obstacles and the boundary of the environment.

Considering the limited ranges of robot sensing and communication, we define a desired distance a_{ij} between robots R_i and R_j as a designated distance over which a sufficiently large but reliable in-between sensor coverage can be established and reliable in-between wireless communication can be maintained. Since R_i and R_j have circular sensing ranges with radii of rs_i and rs_j respectively, in order to achieve sensor coverage, one should choose $a_{ij} \leq rs_i + rs_j$. Meanwhile, in order to achieve communication coverage, one should choose $a_{ij} \leq \min(rc_i, rc_j)$ such that R_i and R_j are inside each other's communication range. As a result, in order to achieve both the sensor and communication coverage, one should choose $a_{ij} \leq \min(rs_i + rs_j, \min(rc_i, rc_j))$. In practice, the robot sensing range is relatively stable, and can be determined through off-line sensor calibration. The determination of the robot communication range is more complicated, because it is easily affected by various environmental factors in real time, such as surrounding objects and atmospheric conditions. However, a communication calibration and a conservative estimation are helpful. Moreover, the communication range is usually much longer than the sensing range, which means that an estimate of the desired robot-to-robot distance based on the sensing ranges of the involved robots is often reliable.

Besides the desired distance between a pair of robots, one can also specify the desired distance between a robot and an obstacle and the desired distance between a robot and the boundary. In fact, when detecting an obstacle (either stationary or moving), the robot should keep a distance from it to avoid collision while probably maintaining an observation of it. Similarly, in a confined environment, e.g. surrounded by a wall, when detecting the boundary (either stationary or evolving), the robot should also keep a distance from it to avoid collision while maintaining a sufficient sensor coverage of it. Denoting the desired distance between a robot R_i and an obstacle O_j inside its sensing range as b_{ij}, denoting the desired distance between R_i and the boundary B inside its sensing range as c_{ij}, denoting the safety margin of R_i as s_i and noticing $s_i \leq r_i$ in general, one should choose $b_{ij} \in [s_i, r_i]$ and $c_{ij} \in [s_i, r_i]$.

In order to achieve the distributed control of the multi-robot deployment, we decompose the desired global deployment configuration into the desired local deployment configuration around each robot. That is, each robot only needs to approach and maintain desired distances with nearby objects, including other robots, obstacles and the boundary of the environment, in its neighborhood. It results in largely reduced computational and communication complexities. The combined effect of approaching desired local deployment configuration in the neighborhood of each member robot will lead to the desired global deployment configuration at the system level.

To unify the representations, we denote the jth object, which can be another robot, an obstacle or the boundary, in the neighborhood of R_i as T_{ij}, and the desired distance between R_i and T_{ij} as d_{ij} which can be a_{ij}, b_{ij} or c_{ij}, as defined above, corresponding to the actual type of object. In general, a more conservative choice of d_{ij} tends to result in a more reliable but smaller coverage.

3.3 Concept of neighborhood

In order to reduce the computational and communication complexities and achieve distributed control of multi-robot deployment, we propose that each member robot determines its desired motion based on only the state information of other robots and objects in its neighborhood with the intention to approach the desired local deployment configuration in its neighborhood. Therefore, defining the neighborhood for each robot is of high importance to our distributed multi-robot deployment control algorithm. The proposed scheme applies to the following two different definitions of neighborhood:

1. Physical neighborhood: The physical neighborhood of a robot R_i is defined by the robot sensing and communication ranges. If another robot R_j is inside the communication range of R_i, R_i can obtain the state of R_j through communication and retrieve the geometric relationship between R_i and R_j. If R_j is inside the sensing range of R_i, R_i may even sense the state of R_j directly, if appropriate onboard sensors are available. In these cases, we consider that R_j belongs to the neighborhood of R_i. If R_j is outside the sensing and communication ranges of R_i, R_i cannot obtain the state of R_j or retrieve the geometric relationship between R_i and R_j from either direct sensing or communication between them. In these case, we consider that R_j does not belong to the neighborhood of R_i. The concept of physical neighborhood provides a complete count of those physically nearby robots. However, intensive communications among multiple robots may arise in a dense robot gathering, such as the initial stage of the multi-robot deployment process.

2. Topological neighborhood: The topological neighborhood of R_i is defined as the set of its one-hop neighbors on a topological graph representation of the multi-robot system, such as the Delaunay triangulation and the Gabriel graph (Tan, 2005; Sander et al., 2002; Preparata & Shamos, 1985). Since, at any time, there are only a very limited number of one-hop topological neighbors, both the computational and communication complexities are relatively low. However, if the topological relationship among the multiple robots changes, in order to redefine the global topology, multi-hop communications are often necessary and may become intensive. Moreover, depending on the specific type of underlying graph, such as the Delaunay triangulation, we notice that the one-hop neighbors of a boundary node may include those at physically long distances.

Besides robots, an obstacle or the boundary is considered belonging to the neighborhood of a robot R_i only when it is inside the sensing range of R_i.

3.4 Mathematical formulation

The deployment motion of each member robot is governed by its equation of deployment motion. According to the Hamilton's principle (Goldstein, 1980), the optimal deployment motion of a mobile robot R_i during the time period $[t_1, t_2]$ should minimize the total action of R_i during this period, i.e.

$$\mathbf{q}_i(t) = \arg\min_{\mathbf{q}'_i(t)} \int_{t_1}^{t_2} L_i(\mathbf{q}'_i(t))dt \, , \tag{1}$$

where $\mathbf{q}_i(t)$ and $\mathbf{q}'_i(t)$ denote respectively the optimal trajectory and a candidate trajectory of R_i in its configuration space which is spanned by the set of variables uniquely defining the state of R_i, and L_i denotes the Lagrangian of R_i which is defined based on the states of R_i and its neighborhood and will guide R_i to approach the desired neighborhood configuration. In principle, using the method of the variational calculus, one can obtain the following Lagrange's equation for R_i

$$\frac{d}{dt}\left(\frac{\partial L_i}{\partial \dot{\mathbf{q}}_i}\right) - \frac{\partial L_i}{\partial \mathbf{q}_i} = \mathbf{0} \, , \tag{2}$$

which is the equation governing the deployment motion of R_i. Following (2), R_i will self-deploy. The combined effect of the self-deployment motion of all the member robots will lead to the desired global deployment.

Strictly speaking, the dynamics, represented by (2), of a realistic mobile robot can be substantially complicated, with $\mathbf{q}_i$ composed of various motion parameters for the wheels, links and body of the robot. In practice, commercial robot systems mostly provide a transparent lower-level control for the motion of the components, such as the wheels, and users only need to define the motion parameters at the robot level, such as the position, orientation and speed of the whole robot. This is equivalent to an upper-level control of the robot motion, which is based on a lumped model of the robot. Following this practice, we further our discussion and derivation of the distributed multi-robot deployment algorithm with a lumped robot model.

A lumped model for a mobile robot R_i moving in a 2D environment can be defined as following:

1. The position of the robot is represented by the coordinates of its center of mass, (x_i, y_i), in the global frame;
2. The orientation of the robot is represented by the angle θ_i between the longitudinal direction of the robot and the x axis of the global frame;
3. The robot is considered having a point mass m_i at its center of mass;
4. The robot is considered having a moment of inertia I_i about the vertical axis passing through its center of mass.

Based on this 2D lumped robot model, we have

$$\mathbf{q}_i = [x_i \quad y_i \quad \theta_i]^T. \tag{3}$$

We also define the Lagrangian of R_i in the 2D environment as

$$L_i = T_i - U_i, \tag{4}$$

where T_i denotes the kinetic energy of R_i

$$T_i = \frac{1}{2} m_i (\dot{x}_i^2 + \dot{y}_i^2) + \frac{1}{2} I_i \dot{\theta}_i^2, \tag{5}$$

and U_i denotes an artificial potential energy which drives R_i.

The artificial potential energy U_i is defined to move R_i towards its desired neighborhood configuration, based on the definitions of the desired distances between R_i and nearby objects

$$\begin{aligned} U_i = & \frac{1}{2} \sum_{j=1}^{n_i} w_{ij} (\sqrt{(x_{ij} - x_i)^2 + (y_{ij} - y_i)^2} - d_{ij})^2 \\ & + \chi \frac{1}{2} (\mathrm{atan2}(\frac{1}{n_i} \sum_{j=1}^{n_i} w_{ij} x_{ij} - x_i, \frac{1}{n_i} \sum_{j=1}^{n_i} w_{ij} y_{ij} - y_i) - \theta_i)^2 \end{aligned}. \tag{6}$$

Here, n_i denotes the number of other objects in the neighborhood of R_i. With (x_{ij}, y_{ij}) denoting the position of the jth object T_{ij} (in fact the position of the nearest point on T_{ij} relative to R_i) in the neighborhood of R_i, which can be another robot, an obstacle or the boundary, $\sqrt{(x_{ij} - x_i)^2 + (y_{ij} - y_i)^2}$ gives the actual instantaneous distance between R_i and T_{ij}. With d_{ij} denoting the desired distance between R_i and T_{ij}, the term $\frac{1}{2}(\sqrt{(x_{ij} - x_i)^2 + (y_{ij} - y_i)^2} - d_{ij})^2$ defines a potential energy component based on the difference between the desired and actual distances between R_i and T_{ij}, which generates an actuating force to drive R_i towards the desired distance between R_i and T_{ij}. If the actual distance is shorter than d_{ij}, it tends to push R_i away from T_{ij}; if the actual distance is longer than d_{ij}, it tends to pull R_i towards T_{ij}. In addition, w_{ij} is a coefficient weighting the effect among the objects in the neighborhood of R_i. A larger w_{ij} means a bigger influence of T_{ij} on R_i. Meanwhile, $(\frac{1}{n_i} \sum_{j=1}^{n_i} w_{ij} x_{ij}, \frac{1}{n_i} \sum_{j=1}^{n_i} w_{ij} y_{ij})$ defines C_i—the center of mass of the neighborhood of R_i, and $\mathrm{atan2}(\frac{1}{n_i} \sum_{j=1}^{n_i} w_{ij} x_{ij} - x_i, \frac{1}{n_i} \sum_{j=1}^{n_i} w_{ij} y_{ij} - y_i)$ gives the direction angle of the vector pointing from R_i to C_i (where atan2(x, y) gives the arc tangent of y/x, taking into account which quadrant the point (x, y) is in). Therefore, the term $\frac{1}{2}(\text{atan2}(\frac{1}{n_i}\sum_{j=1}^{n_i} w_{ij}x_{ij} - x_i, \frac{1}{n_i}\sum_{j=1}^{n_i} w_{ij}y_{ij} - y_i) - \theta_i)^2$ defines a potential energy component based on the difference between the current orientation of R_i and the direction of C_i relative to R_i, which generates an actuating force to turn R_i towards C_i. This will help to drive R_i towards C_i, and therefore establish a balance in the local deployment. In addition, χ is a coefficient making the translational and rotational terms in (6) compatible. Altogether, (6) defines an artificial potential energy for a member robot according to the difference between the actual and desired configurations in the neighborhood of the robot, which generates the actuating force to drive the robot towards the desired local coverage distances.

Substituting (3)-(6), which define the lumped dynamics model of a mobile robot moving in a 2D environment, into (2), we obtain the basic equation of deployment motion for R_i

$$m_i\ddot{x}_i = -\frac{\partial U_i}{\partial x_i}, \quad m_i\ddot{y}_i = -\frac{\partial U_i}{\partial y_i}, \quad I_i\ddot{\theta}_i = -\frac{\partial U_i}{\partial \theta_i}, \tag{7}$$

Where

$$\frac{\partial U_i}{\partial x_i} = \sum_{j=1}^{n_i} -\frac{w_{ij}(x_{ij}-x_i)(\sqrt{(x_{ij}-x_i)^2+(y_{ij}-y_i)^2}-d_{ij})}{\sqrt{(x_{ij}-x_i)^2+(y_{ij}-y_i)^2}} + \frac{\chi(\frac{1}{n_i}\sum_{j=1}^{n_i} w_{ij}y_{ij} - y_i)(\text{atan2}(\frac{1}{n_i}\sum_{j=1}^{n_i} w_{ij}x_{ij} - x_i, \frac{1}{n_i}\sum_{j=1}^{n_i} w_{ij}y_{ij} - y_i) - \theta_i)}{(\frac{1}{n_i}\sum_{j=1}^{n_i} w_{ij}x_{ij} - x_i)^2 + (\frac{1}{n_i}\sum_{j=1}^{n_i} w_{ij}y_{ij} - y_i)^2},$$

$$\frac{\partial U_i}{\partial y_i} = \sum_{j=1}^{n_i} -\frac{w_{ij}(y_{ij}-y_i)(\sqrt{(x_{ij}-x_i)^2+(y_{ij}-y_i)^2}-d_{ij})}{\sqrt{(x_{ij}-x_i)^2+(y_{ij}-y_i)^2}} - \frac{\chi(\frac{1}{n_i}\sum_{j=1}^{n_i} w_{ij}x_{ij} - x_i)(\text{atan2}(\frac{1}{n_i}\sum_{j=1}^{n_i} w_{ij}x_{ij} - x_i, \frac{1}{n_i}\sum_{j=1}^{n_i} w_{ij}y_{ij} - y_i) - \theta_i)}{(\frac{1}{n_i}\sum_{j=1}^{n_i} w_{ij}x_{ij} - x_i)^2 + (\frac{1}{n_i}\sum_{j=1}^{n_i} w_{ij}y_{ij} - y_i)^2},$$

$$\frac{\partial U_i}{\partial \theta_i} = -\chi(\text{atan2}(\frac{1}{n_i}\sum_{j=1}^{n_i} w_{ij}x_{ij} - x_i, \frac{1}{n_i}\sum_{j=1}^{n_i} w_{ij}y_{ij} - y_i) - \theta_i)\cdot$$

Equation (7) does not include any kinematic constraint. However, in practice, a wheeled robot is under the nonholonomic constraint. That is, at any time it can only have a non-zero speed in its longitudinal direction (i.e. along its orientation) while its side speed is zero. For the lumped robot model, the nonholonomic constraint is defined as

$$\dot{x}_i \sin\theta_i - \dot{y}_i \cos\theta_i = 0\cdot \tag{8}$$

Then, instead of using (2), the optimal deployment motion of a wheeled robot R$_i$ during the time period [t$_1$, t$_2$] should minimize the total action of R$_i$ during this period and satisfy the nonholonomic constraint, i.e.

$$[x_i(t), y_i(t), \theta_i(t)] = \arg \min_{x_i(t), y_i(t), \theta_i(t)} \int_{t_1}^{t_2} L_i dt \text{ subject to } \dot{x}_i \sin\theta_i - \dot{y}_i \cos\theta_i = 0 \cdot \tag{9}$$

Using the method of the variational calculus (Goldstein, 1980), we can obtain the following Lagrange's equation for R$_i$

$$\begin{aligned} & m_i \ddot{x}_i - \lambda_i \sin\theta_i = -\frac{\partial U_i}{\partial x_i}, \quad m_i \ddot{y}_i + \lambda_i \cos\theta_i = -\frac{\partial U_i}{\partial y_i}, \\ & I_i \ddot{\theta}_i = -\frac{\partial U_i}{\partial \theta_i}, \quad \dot{x}_i \sin\theta_i - \dot{y}_i \cos\theta_i = 0 \end{aligned} \tag{10}$$

where λ_i denotes the Lagrange undetermined multiplier.

Furthermore, it is important for each member robot to converge towards the desired neighborhood configuration. In order to stabilize the deployment motion of each robot around its equilibrium position, a virtual Rayleigh's dissipation function is adopted to provide the necessary damping mechanism

$$F_i = \frac{1}{2}(k_{xi}\dot{x}_i^2 + k_{yi}\dot{y}_i^2 + k_{\theta i}\dot{\theta}_i^2), \tag{11}$$

where k$_{xi}$, k$_{yi}$ and k$_{\theta i}$ are the viscous damping coefficients associated with the linear and angular velocities of R$_i$ respectively. F$_i$ defines the damping force for each velocity component as

$$\frac{\partial F_i}{\partial \dot{x}_i} = k_{xi}\dot{x}_i, \quad \frac{\partial F_i}{\partial \dot{y}_i} = k_{yi}\dot{y}_i, \quad \frac{\partial F_i}{\partial \dot{\theta}_i} = k_{\theta i}\dot{\theta}_i \cdot \tag{12}$$

Incorporating (12) into (7), we obtain the equation of deployment motion for a holonomic mobile robot with dissipation as

$$m_i \ddot{x}_i = -\frac{\partial U_i}{\partial x_i} - k_{xi}\dot{x}_i, \quad m_i \ddot{y}_i = -\frac{\partial U_i}{\partial y_i} - k_{yi}\dot{y}_i, \quad I_i \ddot{\theta}_i = -\frac{\partial U_i}{\partial \theta_i} - k_{\theta i}\dot{\theta}_i, \cdot \tag{13}$$

Incorporating (12) into (10), we obtain the equation of deployment motion for a nonholonomic mobile robot with dissipation as

$$\begin{aligned} & m_i \ddot{x}_i - \lambda_i \sin\theta_i = -\frac{\partial U_i}{\partial x_i} - k_{xi}\dot{x}_i, \quad m_i \ddot{y}_i + \lambda_i \cos\theta_i = -\frac{\partial U_i}{\partial y_i} - k_{yi}\dot{y}_i, \\ & I_i \ddot{\theta}_i = -\frac{\partial U_i}{\partial \theta_i} - k_{\theta i}\dot{\theta}_i, \quad \dot{x}_i \sin\theta_i - \dot{y}_i \cos\theta_i = 0 \end{aligned} \tag{14}$$

Equations (13) and (14) are the final equations governing the self-deployment motion of a member robot to approach the desired local coverage. In practice, each member robot can online calculate its desired instantaneous acceleration for the deployment motion by substituting its current pose (position and orientation) and the positions of other objects in its neighborhood into (13) (holonomic) or (14) (nonholonomic), and command the lower-level controller to move the robot at the resulting acceleration. By moving each mobile robot in the way defined by its equation of deployment motion, eventually the resulting local coverage in the neighborhoods of all the robots altogether forms a global coverage of the multi-robot system to the targeted environment.

4. Simulation results

The effectiveness of the proposed distributed multi-robot deployment control algorithm has been verified by simulations programmed in Matlab.

4.1 Settings

In the following reported simulations, we assume that 120 mobile robots, each with unit mass and unit moment of inertia, are deployed into a 2D environment. The deployments of both holonomic and nonholonomic mobile robots in both open and corridor environments based on both physical and topological neighborhood consideration have been simulated respectively. We assume that the initially the group of robots are uniformly distributed in a small square region of $x_i \in [-1, 1]$ and $y_i \in [-1, 1]$ with uniformly distributed orientation $\theta_i \in [-\pi, \pi]$. We set the desired distance between a pair of nearby robots as $d_{ij}=5$, the viscous damping coefficients $k_{xi}=k_{yi}=k_{\theta i}=2$, and the relevant weighting coefficients $\chi=w_{ij}=1$. We also discretize the time into a equally-divided sequence $\{t_0, t_1, t_2, \ldots\}$ with the common interval $\Delta t=0.2$. At each time t_k, we calculate the desired instantaneous acceleration of each robot R_i from its equation of deployment motion (13) (if R_i is holonomic) or (14) (if R_i is nonholonomic). The velocity and the pose of R_i at t_k are updated iteratively by numerical integration.

4.2 Deployment in an open environment

Considering 120 holonomic mobile robots being deployed in an open 2D environment, at first we define the self-deployment motion of each mobile robot based on the state of its physical neighborhood. A representative deployment of the robots at t=30 is shown as Fig.2. Then we define the self-deployment motion of each mobile robot based on the state of its topological neighborhood which is generated using the Delaunay triangulation. A representative deployment of the robots at t=30 is shown as Fig.3.

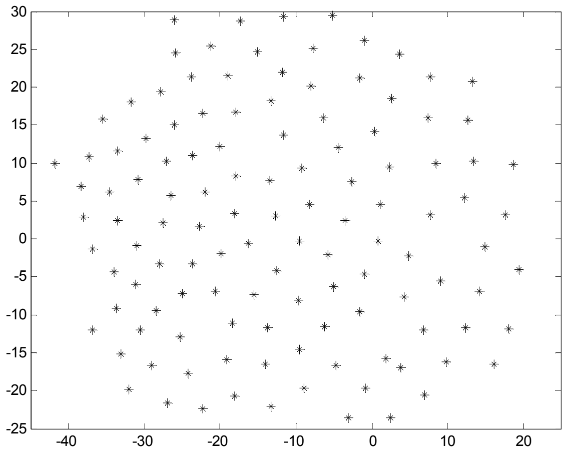

Fig. 2. Deployment of 120 holonomic mobile robots in an open 2D environment based on the state of the physical neighborhood

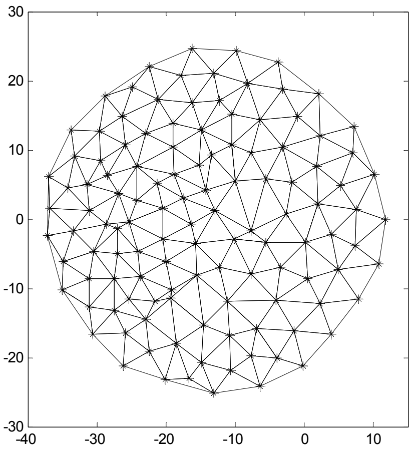

Fig. 3. Deployment of 120 holonomic mobile robots in an open 2D environment based on the state of the topological neighborhood

Considering 120 nonholonomic mobile robots being deployed in an open 2D environment, at first we define the self-deployment motion of each mobile robot based on the state of its physical neighborhood. A representative deployment of the robots at t=30 is shown as Fig.4. Then we define the self-deployment motion of each mobile robot based on the state of its topological neighborhood which is generated using the Delaunay triangulation. A representative deployment of the robots at t=30 is shown as Fig.5.

Figures 2-5 indicates that using the proposed distributed multi-robot deployment control algorithm, both holonomic and nonholonomic mobile robots spread out effectively from their initial gathering to cover the environment, based on the local driving forces defined on either the physical neighborhood or the topological neighborhood. However, while a holonomic mobile robot is capable of moving in any direction at any time (Fig.6), a nonholonomic mobile robot, under the nonholonomic constraint, has to move along its longitudinal direction at any time (Fig.7). For the convenience of display, we show the representative deployment paths of both holonomic (Fig.6) and nonholonomic (Fig.7) mobile robots generated from 12 robots instead of 120 robots, with the arrow head representing the robot orientation. In particular, in the nonholonomic case, backward movement along the longitudinal direction, which satisfies the nonholonomic constraint, is allowed as long as it moves a robot towards the desired neighborhood configuration.

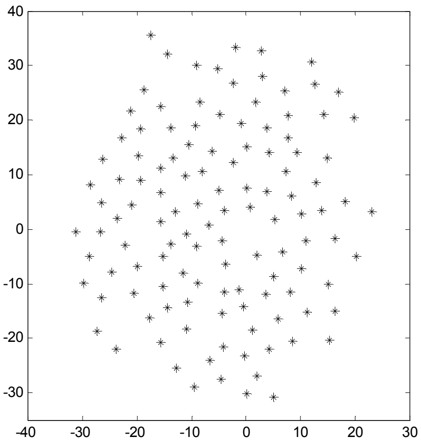

Fig. 4. Deployment of 120 nonholonomic mobile robots in an open 2D environment based on the state of the physical neighborhood

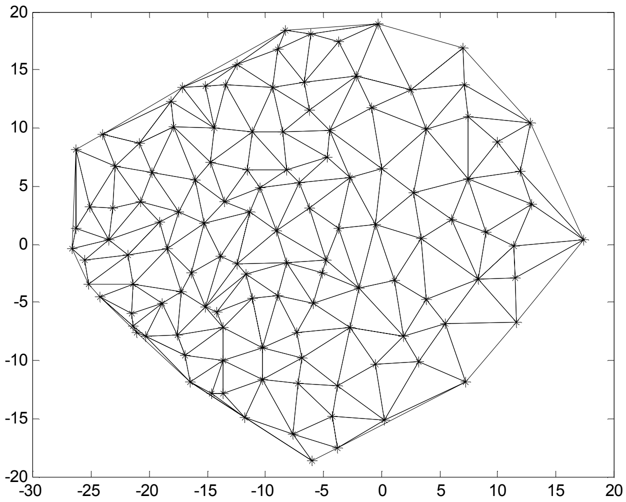

Fig. 5. Deployment of 120 nonholonomic mobile robots in an open 2D environment based on the state of the topological neighborhood

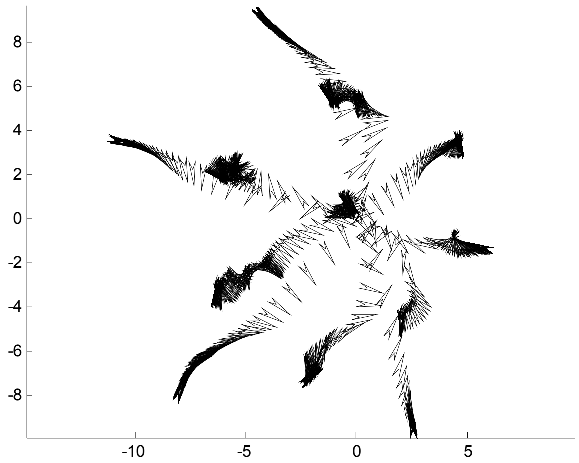

Fig. 6. Deployment paths for 12 holonomic mobile robots

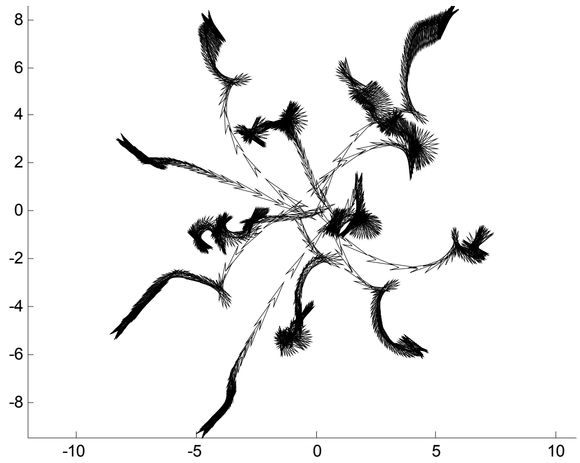

Fig. 7. Deployment paths for 12 nonholonomic mobile robots

4.3 Deployment in a corridor environment

Considering 120 holonomic mobile robots being deployed into a corridor environment with straight walls at y=±15, at first we define the self-deployment motion of each mobile robot based on the state of its physical neighborhood. A representative deployment of the robots at t=30 is shown as Fig.8. Then we define the self-deployment motion of each mobile robot based on the state of its topological neighborhood which is generated using the Delaunay triangulation. A representative deployment of the robots at t=30 is shown as Fig.9.

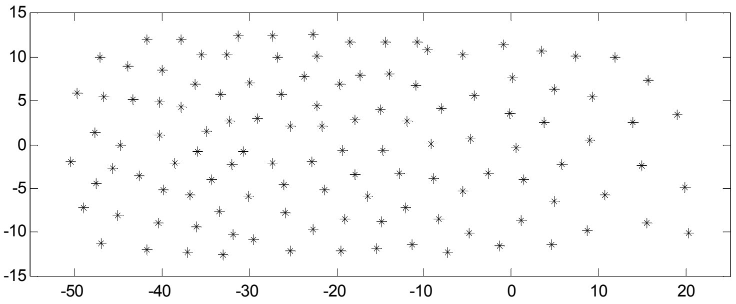

Fig. 8. Deployment of 120 holonomic mobile robots in a corridor environment based on the state of the physical neighborhood

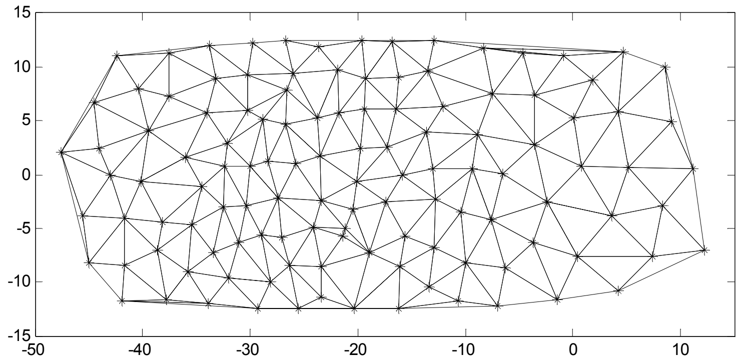

Fig. 9. Deployment of 120 holonomic mobile robots in a corridor environment based on the state of the topological neighborhood

Considering 120 nonholonomic mobile robots being deployed into the same corridor environment, at first we define the self-deployment motion of each mobile robot based on the state of its physical neighborhood. A representative deployment of the robots at t=30 is shown as Fig.10. Then we define the self-deployment motion of each mobile robot based on the state of its topological neighborhood which is generated using the Delaunay triangulation. A representative deployment of the robots at t=30 is shown as Fig.11.

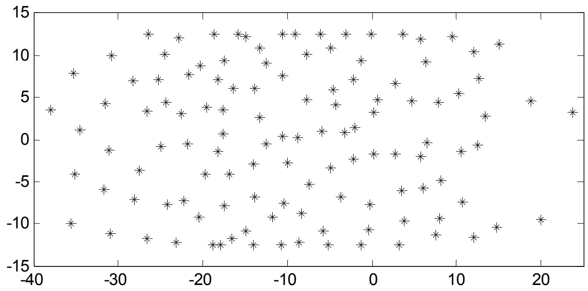

Fig. 10. Deployment of 120 nonholonomic mobile robots in a corridor environment based on the state of the physical neighborhood

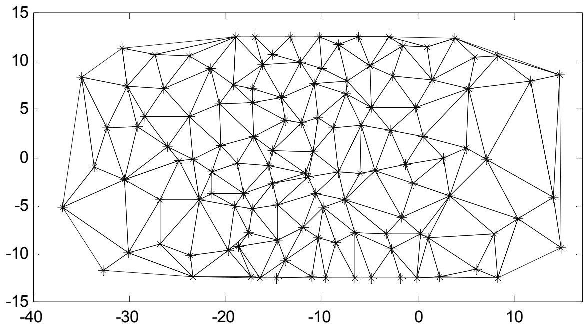

Fig. 11. Deployment of 120 nonholonomic mobile robots in a corridor environment based on the state of the topological neighborhood

Figures 8-11 indicates that using the proposed distributed multi-robot deployment control algorithm, both holonomic and nonholonomic mobile robots spread out effectively from their initial gathering to cover the corridor environment, based on the local driving forces defined on either the geometric neighborhood or the topological neighborhood.

5. Conclusions, discussions and future work

This chapter addresses an important research topic in the field of multi-robot systems, the deployment problem, and introduces a novel distributed multi-robot deployment control algorithm for spreading a team of mobile robots into a targeted environment to form sensor and communication coverage. Each member robot self-deploys according to its equation of deployment motion. The driving force for each mobile robot is defined according to the difference between the actual and desired configurations in the neighborhood of the robot. The Rayleigh's dissipation function is adopted to provide the necessary damping mechanism which maintains the stability of the deployment motion for each robot. Derived from the Hamilton's principle using the method of the variational calculus, the equation of deployment motion naturally incorporates the nonholonomic constraint arising in wheeled robots. Since the equation of deployment motion for each robot depends on only the robot's own kinematic state and its detectable positional relationship with nearby objects, the proposed scheme decentralizes the multi-robot deployment problem into the motion control of individual robots. The combined outcome of the local deployment motion of individual robots leads to the desired global coverage. Simulation results show that the proposed approach can effectively guide the deployment of multi-robot systems.

When deployed into a static environment with fixed obstacles and boundary, the multi-robot system may converge to static sensor coverage. When deployed into a dynamic environment with moving obstacles and evolving boundary, the multi-robot system can change its configuration adaptively. Moreover, the resulting coverage can be a partial instead of complete coverage to the targeted environment, when the environment is larger than the maximum static coverage area of the multi-robot system. After the initial coverage is formed, the multi-robot system can move to provide a mobile coverage of the whole environment. In fact, to take the full advantage of the mobility of the multi-robot system and reduce the operation cost, instead of using a large number of mobile robots to form a static coverage network for a large environment, it is often more efficient to send fewer mobile robots to provide a mobile coverage to the environment. The technique governing multi-robot mobile coverage will be discussed in our future work.

Since the proposed multi-robot deployment control scheme is derived based on a lumped dynamics model of mobile robots and incorporates the nonholonomic kinematic constraint, it helps to result in physically realizable deployment motion in realistic robot systems. However, in order to further improve the implementation robustness of the proposed algorithm to guarantee the physical realizability of the resulting deployment motion, more kinematic and dynamic constraints, such as maximum velocity and acceleration, will be incorporated, and more detailed dynamics models of realistic mobile robots will be studied. Experimental research with physical multi-robot systems will be conducted to verify the results of our algorithm research.

Moreover, we will investigate the convergence property of the whole multi-robot system towards the global optimal coverage as the result of the collection of local coverage. In fact, it is convenient to prove that, if a member robot R_i has a static neighborhood, i.e. other objects in R_i neighborhood are static, the self-deployment motion of R_i, defined by (13) and (14), converges asymptotically to a fixed equilibrium location. We consider the total energy function of each robot R_i as the Lyapunov function candidate

$$E_i = T_i + U_i \cdot \tag{15}$$

From both (13) and (14), we obtain

$$\dot{E}_i = -2F_i \tag{16}$$

for both holonomic and nonholonomic mobile robots. Since $k_{xi}>0$, $k_{yi}>0$ and $k_{\theta i}>0$, E_i tends to decrease, and hence the self-deployment motion of R_i is Lyapunov asymptotically stable. However, in practice, all member robots are moving. As indicated by the definition of the artificial potential energy on R_i (6), the motion of R_i is coupled with that of other objects in its neighborhood. The changing positions of other objects in R_i neighborhood result in the changing equilibrium position of R_i. Therefore, at each moment, R_i moves towards its new equilibrium position. However, this change will slow down as the robots spread out, and R_i deployment motion will converge, as we observe from the simulations. As a future work, we will study the analytical relationship between the global convergence at the multi-robot system level and the local convergence at the individual robot level.

The influence of the dynamic change in the multi-robot topological structure on the operational efficiency of deployment will also be investigated. Though the concept of topological neighborhood reduces the computational and communication complexities of the multi-robot deployment problem by coupling the deployment motion of each member robot with that of only its one-hop topological neighbors defined by Delaunay Triangulation, once the system topology changes, however, intensive communication and computation are required to update the Delaunay Triangulation of the whole multi-robot system. We will seek to maintain the global topologic optimality using local topologic adjustment instead of global reorganization.

6. Acknowledgements

Special thanks are given to Prof. Jindong Tan from the Department of Electrical and Computer Engineering, Michigan Technological University and Prof. Xin Wang from the Department of Electrical and Computer Engineering, State University of New York at Stony Brook for their suggestions to this work.

7. References

Bishop, B. E. (2007). On the use of capability functions for cooperative objective coverage in robot swarms. *2007 IEEE International Conference on Robotics and Automation.*

Borenstein, J.; Everett, H. R.; Feng, L. & Wehe, D. (1997). Mobile robot positioning: sensors and techniques. *Journal of Robotic Systems,* Vol.14, No.4, 231-249.

Butler, Z. & Rus, D. (2003). Event-based motion control for mobile sensor networks, *IEEE Pervasive Computing,* Vol.2, No.4, 34–43.

Cortes, J.; Martinez, S.; Karatas, T. & Bullo, F. (2004). Coverage control for mobile sensing networks, *IEEE Transactions on Robotics and Automation,* Vol.20, No.2, 243–255.

Fan, W. H.; Liu, Y. H.; Wang, F. & Cai, X. P. (2005). Multi-robot formation control using potential field for mobile ad-hoc networks. *2005 IEEE International Conference on Robotics and Biomimetics.*

Goldstein, H. (1980). *Classical Mechanics, 2nd edition,* Addison-Wesley, Reading, MA.

Howard, A.; Mataric, M. J.; Sukhatme, G. S. (2002). Mobile sensor network deployment using potential fields: a distributed, scalable solution to the area coverage problem. *Proceedings of the 6th International Conference on Distributed Autonomous Robotic Systems*.

Jiang, Q. (2006). An improved algorithm for coordination control multi-agent system based on r-limited Voronoi partitions. *Proceedings of the 2006 IEEE International Conference on Automation Science and Engineering.*

Jenkin, M. & Dudek, G. (2000). The paparazzi problem. Proceedings of the 2000 IEEE/RSJ International Conference on Intelligent Robots and Systems.

Ji, M. & Egerstedt, M. (2007). Distributed coordination control of multiagent systems while preserving connectedness. *IEEE Transactions on Robotics,* Vol.23, No.4, 693-703.

Jung, B. & Sukhatme, G. S. (2002). Tracking targets using multiple robots: the effect of environment occlusion. *Autonomous Robots,* Vol.13, 191-205.

Kerr, W.; Spears, D.; Spears, W. & Thayer, D. (2005). Two formal gas models for multi-agent sweeping and obstacle avoidance. *Lecture Notes in Artificial Intelligence*, Vol.3228, 111-130.

Lam, M. & Liu, Y. (2006). ISOGRID: an efficient algorithm for coverage enhancement in mobile sensor networks. *Proceedings of the 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems.*

Pac, M. R.; Erkmen, A. M. & Erkmen, I. (2006). Scalable self-deployment of mobile sensor networks: a fluid dynamics approach. *Proceedings of the 2006 IEEE/RSJ International Conference on Intelligent Robotics and Systems.*

Parker, L. (1999). Cooperative robotics for multi-target observation. *Intelligent Automation and Soft Computing,* Vol.5, No.1, 5-19.

Parker, L. (2002). Distributed algorithms for multi-robot observation of multiple moving targets. *Autonomous Robots,* Vol.12, 231-255.

Preparata, F. & Shamos, M. I. (1985). *Computational Geometry: An Introduction,* Springer-Verlag, New York.

Poduri, S.; Sukhatme, G. S. (2004). Constrained coverage for mobile sensor networks. *Proceedings of 2004 IEEE International Conference on Robotics and Automation.*

Popa, D. O.; H. E. Stephanou; Helm, C. & Sanderson, A. C. (2004). Robotic deployment of sensor networks using potential fields. *Proceedings of the 2004 IEEE International Conference on Robotics & Automation.*

Reif, J. H. & Wang, H. (1999). Social potential fields: a distributed behavioral control for autonomous robots. *Robotics and Autonomous Systems,* Vol.27, 171-194.

Sander, P. V.; Peleshchuk, D. & Grosz, B. J. (2002). A scalable, distributed algorithm for efficient task allocation. *Proceedings of the first international joint conference on Autonomous agents and multiagent systems.*

Schwager, M.; Slotine, J. & Rus, D. (2007). Decentralized, adaptive control for coverage with networked robots. *2007 IEEE International Conference on Robotics and Automation.*

Tan, J.; Xi, N.; Sheng, W. & Xiao, J. (2004). Modeling multiple robot systems for area coverage and cooperation. *Proceedings of the 2004 IEEE International Conference on Robotics & Automation.*

Tan, J. (2005). A scalable graph model and coordination algorithms for multi-robot systems. *Proceedings of 2005 IEEE/ASME International Conference on Advanced Intelligent Mechatronics.*

Winfield, A. (2000). Distributed sensing and data collection via broken ad hoc wireless connected networks of mobile robots, In: *Distributed Autonomous Robotic Systems 4*, Parker, L.; Bekey, G. & Barhen, J., Springer-Verlag.

Zhou, Y. & Tan, J. (2008). Deployment of multi-robot systems under the nonholonomic constraint. *IEEE/ASME International Conference on Advanced Intelligent Mechatronics.*

Zhou, Y. (2008). A distributed self-deployment algorithm suitable for multiple nonholonomic mobile robots. *Proceedings of the ASME 2008 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference.*

Motion Control

Edited by Federico Casolo

ISBN 978-953-7619-55-8
Hard cover, 590 pages
Publisher InTech
Published online 01, January, 2010
Published in print edition January, 2010

The book reveals many different aspects of motion control and a wide multiplicity of approaches to the problem as well. Despite the number of examples, however, this volume is not meant to be exhaustive: it intends to offer some original insights for all researchers who will hopefully make their experience available for a forthcoming publication on the subject.

Yu Zhou (2010). Distributed Control of Multi-Robot Deployment Motion, Motion Control, Federico Casolo (Ed.), ISBN: 978-953-7619-55-8, InTech, Available from: http://www.intechopen.com/books/motion-control/distributed-control-of-multi-robot-deployment-motion

INTECH
open science | open minds

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821