The 9th International Conference on Ubiquitous Robots adn Ambient Intelligence (URAI 2012)
Nov. 26-28, 2012 in Daejeon Convention Center (DCC), Daejeon, Korea

Adaptive Lattice Deployment of Robot Swarms Based on Local Triangular Interactions

Yasuhiro Nishimura, Geunho Lee, and Nak Young Chong

Japan Advanced Institute of Science and Technology, Nomi, Ishikawa 923-1292, Japan
(Tel : +81-761-51-1254; E-mail: {y.nishimura, geun-lee, nakyoung}@jaist.ac.jp)

Abstract - This paper addresses the adaptively latticed deployment problem for a swarm of autonomous mobile robots. As our decentralized solution, an adaptive triangle generation algorithm is proposed to allow individual robots to form different equilateral triangular configurations depending on their local distributions. Specifically, Delaunay triangulation is applied to examine a local distribution composed of triangles generated around each robot. From the computation of the local distribution, each robot determines an adequate side length and enables to form an equilateral triangle with the side length. In addition, two convergence conditions are considered according to the controlling way of the side length. By using the proposed algorithm, robot swarms can self-configure themselves while adapting to their distribution conditions. Through extensive simulations, we verify the effectiveness of the proposed algorithm.

Keywords - swarm robots, adaptive triangle generation, different equilateral triangles, Delaunay triangulation

1. Introduction

With lots of recent advances in robotic technologies, much attention has been paid to potential applications for swarms of mobile robots. Robotic swarms are expected to be applied to a wide variety of areas such as habitat or environmental monitoring, surveillance, exploration, and so on. In those applications, individual robots are assumed to be simple, cheap, and disposable. In order to enable robot swarms to perform the aforementioned applications, it is necessary to properly coordinate individual motions. Many decentralized coordination approaches [1]-[6] have been using simple local interactions based on the intuition observed from physical phenomena in nature, mainly employing some types of force balance between inter-individual interactions. These interactions result in lattice-type configurations that offer high level coverage and multiple redundant connections, but the interactions might over-constrain the swarm and frequently leaded to deadlocks.

Differently from the force-balanced interactions, a geometrically local interaction [7]-[9] was proposed to enable each robot in a swarm to generate an equilateral triangle configuration with an initially assigned side length. One self-configuration scheme based on the geometric interaction allowed robot swarms to establish the uniform link connectivity among adjacent robots after starting from arbitrary initial distributions. This makes it possible to take advantage of the redundancy provided by

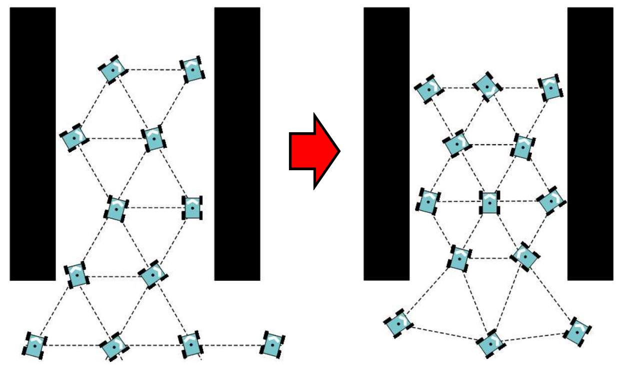
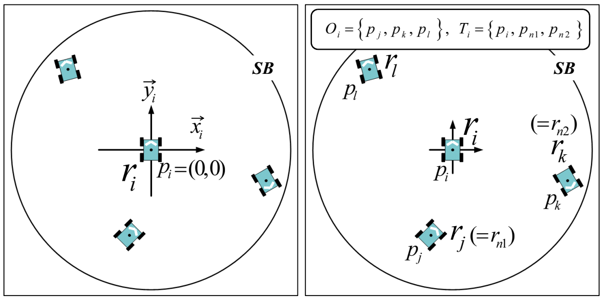


Fig. 1 Concept of potential applications by utilizing the proposed adaptive triangle generation algorithm

a fully connected network topology without the expense and complexity of networking processes. Despite several merits from the networking point of view, the previous works exposed the issue of adaptive and flexible deployments. Since the side length was defined beforehand, it was difficult for robots to generate their motions naturally according to environmental conditions and robot distributions.

To overcome the limitation, this paper addresses the adaptively latticed deployment problem for mobile robot swarms. Towards strong solution approach, it is assumed that robots do not have prior knowledge of their identification numbers, and do not share any common coordinate system nor the leader. Within a limited sensing range, they are able to locally interact by observing locations of other robots. Based on the minimal robot model, we present an adaptive triangle generation algorithm in a fully decentralized way. The proposed algorithm allows robots to self-configure their swarm network composed of equilateral triangle lattices with different side lengths. From the standpoint of the uniform self-configuration, a desired distance to neighboring robots is initially assigned for each robot. When executing the proposed algorithm towards forming the equilateral triangles, this paper presents two challenges of how to determine convergence into a uniform distance.

Next, necessities for the proposed solution are argued as follows. First, robots can self-adjust the side lengths of an target triangle according to the distribution of their adjacent robots, leading to enhancing adaptability of their behaviors. Secondly, by generating individual equilateral triangles, the swarm of robots can reach the overall equilibrium of the swarm. Thirdly, as illustrated in Fig. 1, forming the different regular triangles reduces the po-



(a) r_i's local coordinates and sensing boundary *SB*

(b) observation set O_i and triangular configuration $\mathbb{T}_i$

Fig. 2 Illustration of definition and notations

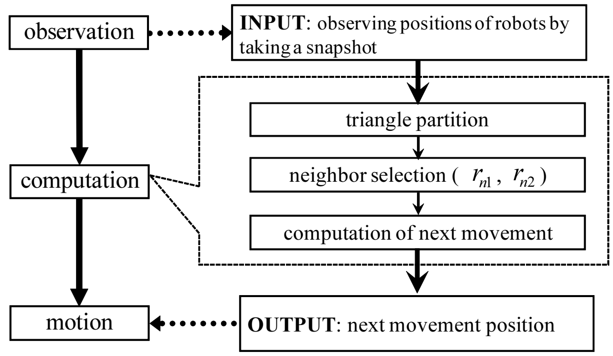


Fig. 3 Flow chart of the adaptive triangle generation algorithm

tential for a traffic jam (uniform navigation) happening in front of a narrow passageway as well as a disordered distribution or a divergence in a confined space. We describe our algorithm in detail, and perform extensive simulations to demonstrate that a swarm of robots can self-deploy themselves while generating equilateral triangles adapting to the distributions of neighboring robots in a scalable manner.

2. Problem Statement

2.1 Robot Model and Notations

In this paper, we consider a swarm of autonomous mobile robots represented as $\{r_1, r_2, \cdots, r_n\}$. It is assumed that an initial distribution of all robots is arbitrary and their positions are distinct. Robots have no leader and no identifiers. They do not share any common coordinate system. Due to a limited observation range, each robot can detect the positions of other robots only within its line-of-sight. Each of robots autonomously moves on a two-dimensional plane. In addition, robots are not allowed to communicate explicitly with other robots in order to obtain any information needed for their motion controls.

Next, let's consider a robot r_i with local coordinates $\vec{x}_i$ and $\vec{y}_i$. As illustrated in Fig. 2-(a), $\vec{y}_i$ defines the vertical axis of r_i's coordinate system as its heading direction. It is straightforward to determine the horizontal axis $\vec{x}_i$ by rotating 90 degrees counterclockwise. The position of r_i is denoted by p_i. Note that p_i is (0,0) with respect to r_i's local coordinates. The distance between p_i and p_j is defined as $dist(p_i, p_j)$. In particular, the desired inter-robot distance between r_i and r_j is denoted by d_u. Moreover, $ang(\vec{m}_i, \vec{n}_i)$ denote the angle between two arbitrary vectors $\vec{m}_i$ and $\vec{n}_i$.

As shown in Fig. 2-(b), r_i detects the positions p_j, p_k, and p_l of other robots located within its sensing boundary *SB*, yielding a set of the observed positions $O_i (= \{p_j, p_k, p_l\})$ with respect to its local coordinates. When r_i selects two robots r_{n1} and r_{n2} in its adjacent robots, namely O_i, located within its *SB*, r_{n1} and r_{n2} are defined as the neighbors of r_i, and their position set $\{p_{n1}, p_{n2}\}$ is denoted by N_i. Given p_i and N_i, a set of three distinct positions $\{p_i, p_{n1}, p_{n2}\}$ with respect to r_i is called triangular configuration $\mathbb{T}_i$, namely $\{p_i, p_{n1}, p_{n2}\}$. Specifically, we define an equilateral configuration, denoted by $\mathbb{E}_i$, as a configuration that all distance permutations between any two of p_i, p_{n1}, and p_{n2} of $\mathbb{T}_i$ are identical.

2.2 Problem Definition

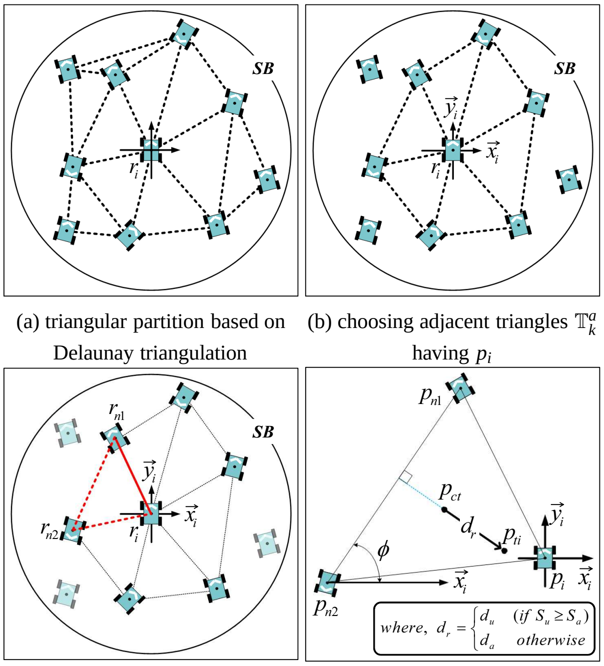
Using $\mathbb{T}_i$ and $\mathbb{E}_i$, we can formally define the *local interaction* as follows: Given $\mathbb{T}_i$, the local interaction allows r_i to maintain the same side length $dist(p_i, p_{n1})$ and $dist(p_i, p_{n2})$ to r_i's neighbors at each time (toward cooperatively forming $\mathbb{E}_i$). Based on the local interaction, we address the ADAPTIVELY LATTICED DEPLOYMENT problem as follows: *Given a swarm of n robots based on the aforementioned model, how to enable the swarm to geometrically deploy themselves into $\mathbb{E}_i$ while adapting to the distribution of the r_i's adjacent robots?*

Now, to solve the addressed problem, we propose the following decentralized approach: *adaptive triangle generation* algorithm. Specifically, by executing the algorithm at each time, geometric self-configuration of n robot swarms can be achieved, allowing larger numbers of robots to converge into $\mathbb{E}_i$ adapting to their local distribution. More details on these algorithms will be explained in the next section.

3. Algorithm Description

As illustrated in Fig. 3, the proposed adaptive triangle generation algorithm is executed for the input O_i with respect to r_i's local coordinate system to output r_i's next movement position at each time. Each robot performs the same algorithm, but acts independently and asynchronously from other robots. Thus, r_i can either be idle or execute an action. At each time, r_i computes its movement position based on O_i, and moves toward the computed position. Such a series of actions is repeated until r_i converges into its $\mathbb{E}_i$.

In Fig. 3, we present the schematic flow of the locally adaptive deployment algorithm composed of the following three functions: triangular partition, neighbor selection, and movement computation functions. Through the first triangular partition function, the triangular partition based on Delaunay triangulation is performed for the elements of O_i including p_i (see Fig. 4-(a)). After the partition generation, a relative adjacent ratio α_a is examined



(a) triangular partition based on Delaunay triangulation (b) choosing adjacent triangles $\mathbb{T}^a_k$ having p_i

(c) rotation of the upper motor (d) rotation of the base motor

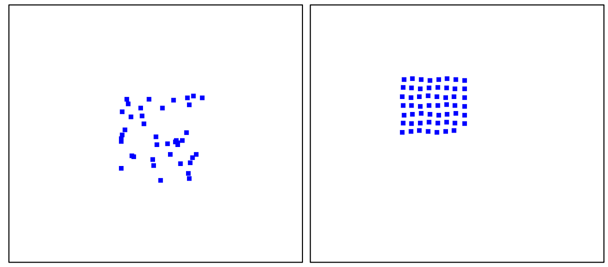
Fig. 4 Illustration of the adaptive triangle generation algorithm

depending on the current partitions around r_i. In detail, as shown Fig. 4-(b), r_i chooses its adjacent triangles $\mathbb{T}^a_{i,k}$ having p_i as one vertex of each triangle where k indicates a positive constant. Next, from the obtained $\mathbb{T}^a_{i,k}$, two kinds of computations are executed: 1) location set D_i of individual vertices included in $\mathbb{T}^a_{i,k}$ and 2) average size S_a for individual $\mathbb{T}^a_{i,k}$. Specifically, it is assumed that S_a is the area of an equilateral triangle with side length d_a. Under the assumption, the relative adjacent ratio α_a is defined as a proportional number between S_a and S_u that indicates the area of an equilateral triangle with d_u.

Secondly the neighbor selection function is to determine r_i's two neighbors in D_i. As illustrated in Fig. 4-(c), the first neighbor r_{n1} is selected as the one located the shortest distance away from r_i. The second neighbor r_{n2} is selected such that the length of the triangle's perimeter is minimized. Then, r_i forms $\mathbb{T}_i$ with N_i, and then checks whether $\mathbb{T}_i$ is greater than or equal to $\mathbb{E}_i$. If the condition is satisfied, r_i determines d_u to generate $\mathbb{E}_i$, resulting in S_u. Otherwise, d_a is determined.

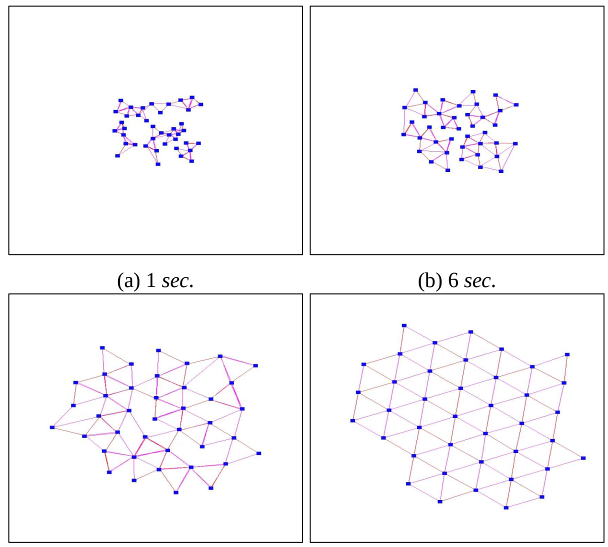
Thirdly, as presented in Fig. 4-(d), r_i finds the centroid p_{ct} of the triangle $\triangle p_i p_{n1} p_{n2}$ (= $\mathbb{T}_i$) with respect to its local coordinates, and measures the angle ϕ between the line $\overline{p_{n1}p_{n2}}$ connecting two neighbors and r_i's horizontal axis $\vec{x}_i$. Using p_{ct} and ϕ, r_i calculates the target point $p_{ti} = (p_{ti,x}, p_{ti,y})$ by the following equations: $(p_{ct,x} + d_r\cos(\phi+\pi/2)/\sqrt{3},\ p_{ct,y} + d_r\sin(\phi+\pi/2)/\sqrt{3})$ where d_r indicates either d_u or d_a. Then, r_i attempts to form an isosceles triangle with its two neighbors at each time. By repeatedly doing this, three robots configure into $\mathbb{E}_i$. Further details on the local interaction can be found in [8][9].

More importantly, two self-configurable options can be selected according to d_a. To begin, r_i forms a triangle lattice with d_a as the side length. Here, d_a varies while generating $\mathbb{E}_i$, and gradually reaches d_u, resulting in the same $\mathbb{E}_i$ with d_u for all robots. On the other hand, once $\mathbb{E}_i$ with the minimum d_a is generated, unless r_j is located shorter distance than the side length d_a, the generated triangle remains unchanged until other robots converge into their desired triangles. Thus target distribution with different equilateral triangles is obtained.



(a) (b)

Fig. 5 Initial distributions for a swarm of 40 robots in an open plane without geographical constraints



(a) 1 *sec.* (b) 6 *sec.*

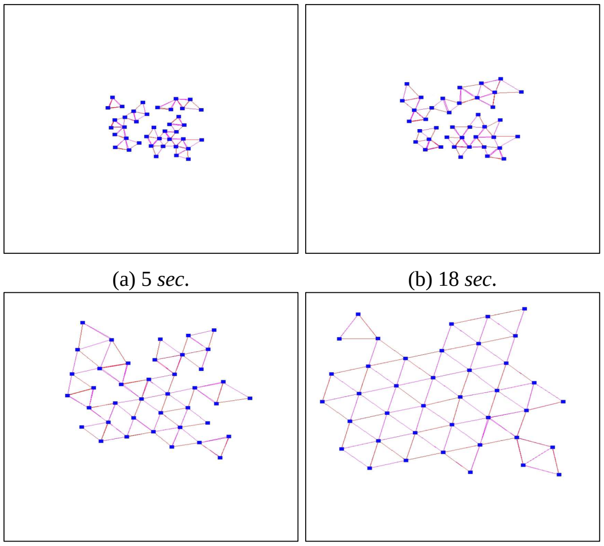
(c) 13 *sec.* (d) 23 *sec.*

Fig. 6 Simulation result of latticed deployment with d_u from the initial condition in Fig. 5-(a)

4. Simulation Results and Discussion

This section describes simulation results that examined the validity of our proposed adaptive triangle generation algorithm. It is supposed that a swarm of robots attempts to deploy themselves toward a desired distribution state composed of $\mathbb{E}_i$. As mentioned in Section 2, the coordinated deployment is achieved without using any leader, identifiers, global coordinate system, and explicit communication. We assumed that each robot possesses *SB*, 2.5 times longer than d_u where d_u is set to 25 unit in our simulator. Our algorithm terminates when individual robots converge into the distance $d_u \pm 1\%$ with their two neighbors.

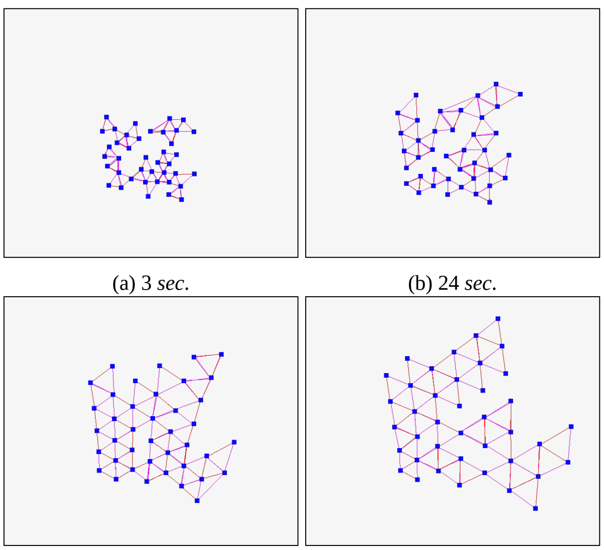
From her, we present simulations demonstrating the convergence property of the proposed algorithm. Fig. 5



(a) 5 *sec.* (b) 18 *sec.*

(c) 58 *sec.* (d) 87 *sec.*

Fig. 7 Simulation result of adaptively latticed deployment from the initial condition in Fig. 5-(a) while changing from d_a to d_u

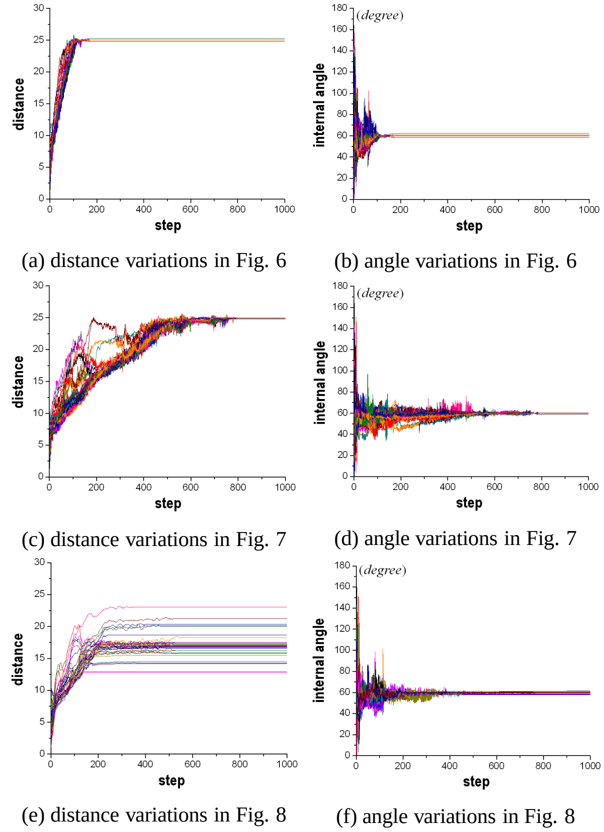


(a) 3 *sec.* (b) 24 *sec.*

(c) 33 *sec.* (d) 51 *sec.*

Fig. 8 Simulation result of adaptively latticed deployment according to d_a from the initial condition in Fig. 5-(a)

shows initial distribution conditions for a swarm of 40 robots. By using the initial distribution in Fig. 5-(a), comparative simulations were executed according to the following three conditions: (1) latticed deployment with d_u in Fig. 6, (2) adaptively latticed deployment while changing from d_a to d_u in Fig. 7, and (3) adaptively latticed deployment depending on d_a in Fig. 8. From the first simulation of Fig. 6, 40 robots could disperse themselves in an open area with a uniform spatial density based on d_u from a random configuration. By performing the proposed algorithm, the second simulation result in Fig. 7 showed, after individual robots determined d_a based on the distribution of their adjacent robots, they could finally generate the desired equilateral triangles with d_u while changing from d_a to d_u. Compared to the second simulation, the third simulation in Fig. 8 presented robots could form differently equilateral triangle configurations according to d_a. Next, Fig. 9 ploted the variation of both side lengths (d_a or d_u) and internal angle ($ang(\overrightarrow{p_i p_{n1}}, \overrightarrow{p_i p_{n2}})$) between $\overrightarrow{p_i p_{n1}}$ and $\overrightarrow{p_i p_{n2}}$ for all

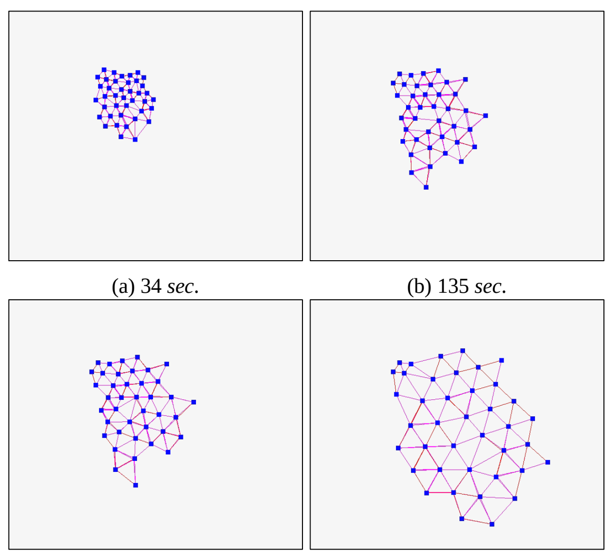


(a) distance variations in Fig. 6 (b) angle variations in Fig. 6

(c) distance variations in Fig. 7 (d) angle variations in Fig. 7

(e) distance variations in Fig. 8 (f) angle variations in Fig. 8

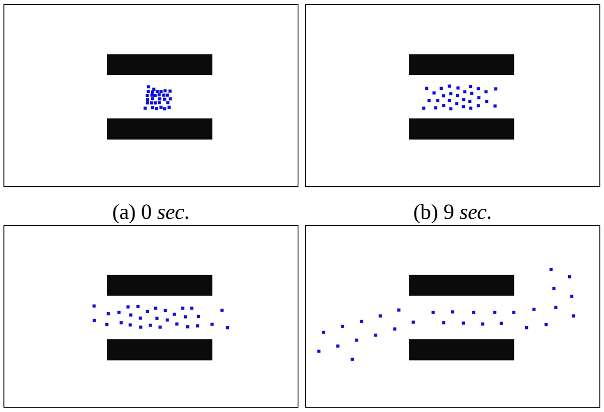
Fig. 9 Comparison results for the variations of distances d_u and d_a and internal angles during individual simulations in Figs. 6, 7, and 8



(a) 34 *sec.* (b) 135 *sec.*

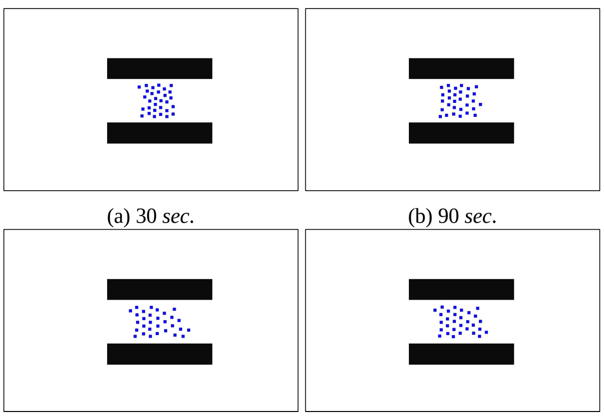
(c) 146 *sec.* (d) 177 *sec.*

Fig. 10 Simulation result of adaptively latticed deployment according to d_a from the different initial condition in Fig. 5-(b)



(a) 0 *sec*. (b) 9 *sec*. (c) 18 *sec*. (d) 55 *sec*.

Fig. 11 Simulation result of adaptively latticed deployment while changing from d_u to d_a in a constrained environment



(a) 30 *sec*. (b) 90 *sec*. (c) 190 *sec*. (d) 248 *sec*.

Fig. 12 Simulation result of adaptively latticed deployment according to d_a from the initial condition in Fig. 11-(a)

robots when performing individual simulations. From the analyzed results, it was also observed that the proposed algorithm allowed robots to converge into $\mathbb{E}_i$. Although robots converged into different d_a in Fig. 9-(e), the angle variations in Fig. 9-(f) could finally reach 60 degrees for their $ang(\overrightarrow{p_i p_{n1}}, \overrightarrow{p_i p_{n2}})$. Moreover, Fig. 10 showed simulation result of adaptively latticed deployment according to d_a from the initial condition in Fig. 5-(b). From this simulation, we confirmed that the adaptive triangular generation algorithm could converge into $\mathbb{E}_i$ adapting to the adjacent distributions of individual robots regardless of initial distribution conditions.

Finally, we examined the effect of the adaptively latticed deployment algorithm when robot swarms are deployed in an even plane bounded by geographical constraints. In these simulations as presented in Figs. 11 and 12, it was assumed that there are two objects with flat surfaces and robots are initially located between the objects. To adaptively coordinate the movements of the robots in the constrained area bounded by the objects, our previous scheme published in [7] was employed. Next, two simulation conditions for the proposed algorithm are set: (1) adaptively latticed deployment while changing from d_a to d_u in Fig. 11 and (2) adaptively latticed deployment depending on d_a in Fig. 12. As expected, the robots under the fist condition generated the desired configurations while changing their side lengths. Although the robots could conformed to the surface borders, due to the assigned d_u, their distribution stretched widely from the left side to the right. Compared to the result in Fig. 11, the simulation result in Fig. 12 seems that their distribution was condensed and adapted to the constrained environmental condition. Though the simulations were executed in the simple condition, from the simulation results, it was observed that the adaptive triangle generation algorithm was effective in coverage over an unknown area with a swarm of robots.

We believe that the proposed algorithm will work well under more complex conditions, but several issues remain to be solved. First of all, as considering any environmental constraints as potential applications mentioned in Section 1, it is required to develop an additional algorithm about how to coordinate the motions of robots under the geographic constraints. Moreover, we need to show a mathematical properties about convergence to $\mathbb{E}_i$ and analyzed results for various simulations. Next, from the practical point of view, we are planning to deal with recognition issues to distinguish between other robots and various objects quickly and accurately. Specifically, to provide the robots with reliable shape recognition capability for various environments, we are currently working on developing a more sophisticated proximity sensor prototypes [10][11].

5. Conclusion

This paper was devoted to developing a new coordinated adaptively latticed deployment approach for mobile robot swarms. As one preliminary study result, the adaptive triangle generation algorithm was presented. The proposed algorithm was built on the following assumptions: anonymity, disagreement on common coordinate systems, no pre-selected leader, and no direct communication. This algorithm was composed of three functions: triangular partition, neighbor selection, and movement computation functions. After each robot examines its local distributions by the triangular partition, it determines adaptive side length and neighbor selection. Next, each robot attempts to maintain an equilateral triangle configuration with its neighbors while moving toward the computed next movement position. Specifically, two convergence conditions were considered under the algorithm, allowing robots to reach final distributions composed of uniform or different equilateral triangles, respectively. We verified the effectiveness of the algorithm by using our in-house simulator, and the simulation results clearly demonstrated that robots could converge into equilateral triangles adapting to the adjacent distributions of robots. Our adaptive triangle generation algorithm for mobile robot swarms is expected to be applied to navigation applications such as examination and assessment of hazardous environments.

References

[1] B. Shucker, T.D. Murphey, and J.K. Bennett, “Convergence-preserving switching for topology-dependent decentralized systems”, *IEEE Trans. Robotics*, vol.24, no.6, pp.1405-1415, 2008

[2] Y.F. Zheng and W. Chen, “Mobile robot team forming for crystallization of protein”, *Autonomous Robots*, vol.23, no.1, pp.69-78, 2007

[3] D. Spears, W. Kerr, and W. Spears, “Physics-based robot swarms for coverage problems”, *Int. Jour. Intelligent Control and Systems*, vol.11, no.3, pp.124-140, 2006

[4] Y. Zou and K. Chakrabarty, “Sensor deployment and target localization based on virtual forces”, *Proc. IEEE Infocom Conf.*, pp.1293-1303, 2003

[5] B. Werger and M.J. Mataric, “From insect to internet: situated control for networked robot teams”, *Ann. Math. and Artificial Intelligence*, vol.31, no.1-4, pp.173-198, 2001

[6] J. Reif and H. Wang, “Social potential fields: a distributed behavioral control for autonomous robots”, *Robotics and Autonomous Systems*, vol.27, no.3, pp.171-194, 1999

[7] G. Lee and N.Y. Chong, “Self-configurable mobile robot swarms with hole repair capability”, *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems*,” pp.1403-1408, 2008

[8] G. Lee and N.Y. Chong, “A geometric approach to deploying robot swarms”, *Ann. Math. and Artificial Intelligence*,” vol.52, no.2-4, pp.257-280, 2009

[9] G. Lee and N.Y. Chong, “Adaptive flocking of robot swarms: algorithms and properties”, *IEICE Trans. Communications*, vol.E91-B, no.9, pp.2848-2855, 2008

[10] G. Lee, N. Noguchi, N. Kawasaki, and N.Y. Chong, “An integrated 2D and 3D location measurement system using spiral motion positioner”, *Proc. IEEE Int. Conf. Robotics and Automation*, pp.4422-4427, 2012

[11] G. Lee and N.Y. Chong, “Low-cost dual rotating infrared sensor for mobile robot swarm applications”, *IEEE Trans. Industrial Informatics*, vol.7, no.2, pp.277-286, 2011