A survey on pattern formation of autonomous mobile robots: asynchrony, obliviousness and visibility

To cite this article: Yukiko Yamauchi 2013 J. Phys.: Conf. Ser. 473 012016

View the article online for updates and enhancements.

Related content
- Sharp thresholds and the partition function
  Amin Coja-Oghlan and Daniel Reichman
- Randomness in algorithm design
  Shuji Kijima
- The physics of distributed information systems
  Erik Aurell
A survey on pattern formation of autonomous mobile robots: asynchrony, obliviousness and visibility

Yukiko Yamauchi

Department of Informatics, Graduate School of ISEE, Kyushu University,
744 Motooka, Nishi-ku, Fukuoka, Japan.
E-mail: yamauchi@inf.kyushu-u.ac.jp

Abstract. A robot system consists of autonomous mobile robots each of which repeats Look-Compute-Move cycles, where the robot observes the positions of other robots (Look phase), computes the track to the next location (Compute phase), and moves along the track (Move phase). In this survey, we focus on self-organization of mobile robots, especially their power of forming patterns. The formation power of a robot system is the class of patterns that the robots can form, and existing results show that the robot system’s formation power is determined by their asynchrony, obliviousness, and visibility. We briefly survey existing results, with impossibilities and pattern formation algorithms. Finally, we present several open problems related to the pattern formation problem of mobile robots.

1. Introduction

Motion planning of mobile robots has been studies in many areas such as sensing, exploration, tracking, boundary patrolling, vehicle routing, and so on. As mobile robots become inexpensive and large-scale robot systems become widely used, distributed control of robot systems has attracted increasing attention. In this paper, we consider a robot system consisting of autonomous mobile robots with very weak capabilities, that is, each robot observes the locations of other robots, computes a track to the next location, and move along the track. There is no centralized control, the robots cannot explicitly exchange messages, and they may stop on the track before reaching the destinations.

One of the most important properties of such robot system is its autonomy. Because initial deployment of robots can be arbitrary, states of robots may not be initialized, and the robots may not be activated simultaneously, an initial configuration can be arbitrary. It is expected that the robots autonomously adapt to a given initial configuration and start their task. A system is self-stabilizing if it autonomously recovers correct behavior starting from any arbitrary initial configuration [3]. Self-stabilization also guarantees fault-tolerance for any finite number of transient faults, because the configuration after the last fault is considered as an arbitrary initial configuration.

The pattern formation problem is to form a given pattern from a given initial configuration. See Figure 1, which is an example of circle formation of 24 robots. A robot system is self-organizing if the robots autonomously form a given pattern. Since Yamashita and Suzuki [11] shed light to the theoretical aspect of distributed control of robot systems, many papers have revealed self-organization and self-stabilization of robot systems. In this survey, we focus on the pattern formation problem of mobile robot systems.
We model a robot by a point in a two dimensional Euclidean space. Each robot repeats “Look-Compute-Move” cycle, where it observes the locations of other robots (Look phase), computes a track to the next location using a given algorithm (Compute phase), and move along the track (Move phase). The robots are anonymous, that is, they have no identifiers and they cannot recognize each other. They are oblivious, that is, they do not remember the past cycles and they compute with the observation of the current cycle. They observe and move with its local coordinate system. They neither know the global coordinate system nor agree on the local coordinate system. Finally, their Look-Compute-Move cycles are neither instantaneous nor synchronized. This model is called asynchronous (ASYNC). Note that any look phase is instantaneous, in the sense that it returns the positions of robots at a time. Two stronger models are the semi-synchronous (SSYNC) model and the fully-synchronous (FSYNC) model. In the SSYNC model, Look-Compute-Move cycles are instantaneous, i.e., the time to execute a cycle is negligible, and in the FSYNC model, all robots execute the i-th cycle simultaneously. The essential difference between the ASYNC and the SSYNC models is that a robot may be observed while moving in the ASYNC model, while such a situation never occurs in the SSYNC model.

Suzuki and Yamashita first revealed the pattern formation power of non-oblivious SSYNC and FSYNC robots [11]. They characterized pattern $F$ formable from a given initial configuration $I$: $I$ is formable from $F$ if the geometric symmetricity of $I$ divides that of $F$. Here, geometric symmetricity is the number of rotations centered at the smallest enclosing circle of $I$ ($F$, respectively) that produces $I$ ($F$, respectively) itself. Later, Yamashita and Suzuki showed that oblivious robots in the SSYNC model have the same formation power as non-oblivious robots in the FSYNC model except the gathering of two robots [12]. The gathering problem is to gather all robots at some point, which is a special case of the pattern formation problem. They also pointed out that any oblivious pattern formation algorithm is self-stabilizing which motivates many researches on oblivious robots. Flocchini, Prencipe, Santoro, and Widmayer introduce the ASYNC model [5]. The main difficulty in the ASYNC model is that the robots may observe other robots while they are moving, and never recognize it. Fujinaga, Ono, Kijima, and Yamashita introduced the embedded pattern formation problem where robots can “see” the target pattern as if is drawn on the ground [7]. They proposed an algorithm in ASYNC model that outputs an optimal matching between the initial configuration and the embedded target pattern in the sense that the sum of Euclid distance of the matching is minimized. Hence, even when the robots move asynchronously, the matching never changes. Based on the embedded pattern formation, Fujinaga, Yamauchi, Kijima, and Yamashita showed that oblivious robots in the ASYNC model have the same formation power as non-oblivious robots in the FSYNC model [8]. They showed that robots can agree on the embedding and keep it during the formation even in the ASYNC model.

1 Their algorithm considers target patterns without multiplicity, that is, no two robots occupy the same point in the target pattern. The algorithms in [11, 12] allows multiplicities in the target pattern.
We assume that during a move phase, the origin of multiplicity detection, unless explained explicitly, does not know the number of robots on the point. In the following, we assume robots without two robots. A robot without multiplicity detection only recognizes a point as occupied, but of \(G\) system dimensional Euclidean space. Each robot \(r\) let Robot System:

However, we assume all local coordinate systems are right-handed.

\[ P(t) = \{p_1(t), p_2(t), \ldots, p_n(t)\} \] is a configuration of robots at time \(t\). The robots initially occupy distinct locations, i.e., \(|P(0)| = n\). The robots do not agree on the coordinate system, and each robot \(r_i\) has its local coordinate system \(Z_i(t)\) such that the origin of \(Z_i(t)\) is the position of \(r_i\) at time \(t\) (i.e., \(0 = Z_i(t)[p_i(t)]\))^2. However, we assume all local coordinate systems are right-handed.

We denote the distance (in \(Z_0\)) between two points \(p, q\) by \(\text{dist}(p, q)\). Unlimited visibility provides each robot \(r_i\) with the positions of all robots irrespective of their distance from \(r_i\), while limited visibility provides \(r_i\) with the positions of robots in distance \(V\) (in \(Z_0\)) from \(r_i\). Each robot does not know the value of \(V\). Let \(R_0(t)\) be the set of robots visible from \(r_i\) at time \(t\), and

\[ P_i(t) = \{p_j | r_j \in R_i(t)\} \]

In a look phase, \(r_i\) obtains \(S_i(t) = Z_i(t)[P_i(t)]\) at some time \(t\) in the look phase. We call \(S_i(t)\) the local view of \(r_i\) at time \(t\). Because the visibility range \(V\) is common to all robots, \(r_i \in S_j(t)\) means \(r_j \in S_i(t)\). We define an undirected graph \(G(t) = (R, E(t))\) for \(P(t)\), called mutual visibility graph at time \(t\) where \((r_i, r_j) \in E(t)\) if and only if \(r_i \in S_j(t)\). We assume that \(G(0)\) is connected. In the following, we use \(P(t)\) and \(G(t)\) \((S_i(t)\) and the subgraph of \(G(t)\) induced by \(R_i(t)\), respectively) interchangeably.

Multiplicity detection allows robots to determine whether a point is occupied by more than two robots. A robot without multiplicity detection only recognizes a point as occupied, but does not know the number of robots on the point. In the following, we assume robots without multiplicity detection, unless explained explicitly.

We assume that during a move phase, the origin of \(Z_i(t)\) is fixed to the point of \(r_i\) in the look phase, and does not change.
the radius of an arbitrary center, and any pair of points co-center $c$.

We can also define $Z$ as the set of all patterns of $n$ points. For any $P, P' \in \mathcal{P}_n$, $P$ is similar to $P'$, if there exists $Z \in \mathbb{T}$ such that $Z[P] = P'$, denoted by $P \sim P'$.

We say that a robot $\psi$ forms pattern $F$ in $\mathcal{P}_n$ from initial configuration $I$, if for any execution $P(0)(= I), P(1), \ldots$, there exists a time instant $t$ such that $P(t') \sim F$ for all $t' > t$.

Pattern Formation: A target pattern $F$ is given to every robot $r_i$ as a set of points $Z_0(F) = \{Z_0(p) : p \in F\}$. (Remember that $r_i$ does not have access to the global coordinate system $Z_0$.) We assume that $|Z_0| = n$ and $|F| \leq n$. In the following, as long as it is clear from the context, we identify $p$ with $Z_0[p]$, and write, e.g., “$F$ is given to $r_i$”, instead of “$Z_0[F]$ is given to $r_i$”.

Let $\mathbb{T}$ be the set of all coordinate systems, which can be identified with the set of all transformations consisting of transformations, rotations, and uniform scalings. Let $\mathcal{P}_n$ be the set of all patterns of $n$ points. For any $P, P' \in \mathcal{P}_n$, $P$ is similar to $P'$, if there exists $Z \in \mathbb{T}$ such that $Z[P] = P'$, denoted by $P \sim P'$.

We say that a robot $\psi$ forms pattern $F$ in $\mathcal{P}_n$ from initial configuration $I$, if for any execution $P(0)(= I), P(1), \ldots$, there exists a time instant $t$ such that $P(t') \sim F$ for all $t' > t$.

For any $P \in \mathcal{P}_n$, let $C(P)$ be the smallest enclosing circle of $P$, and $c(P)$ be the center of $C(P)$. Formally, the symmetry $\rho(P)$ of $P$ is defined by

$$\rho(P) = \begin{cases} 1 & \text{if } c(P) \in P, \\ \frac{1}{|\{Z \in \mathbb{T} : P = Z[P]\}|} & \text{otherwise.} \end{cases}$$

We can also define $\rho(P)$ in the following way [11]: $P$ can be divided into regular $k$-gons with co-center $c(P)$, and $\rho(P)$ is the maximum of such $k$. Here, any point is a regular 1-gon with an arbitrary center, and any pair of points $\{p, q\}$ is a regular 2-gon with its center $(p + q/2)$.

A point on the circumference of $C(P)$ is said to be “on circle $C(P)$” and “the interior of $C(P)$” does not include the circumference. We denote the interior of $C(P)$ by $\text{Interior}(P)$, and the radius of $C(P)$ by $r(P)$.

**Figure 2.** Symmetry and decomposition: Black dots represent the points in $P$. 

An algorithm is a function, say $\psi$, that returns a curve to the next location in the Euclidean space, and each robot moves along the curve. A robot is oblivious if the Compute phase only depends on the Look phase of the cycle, otherwise non-oblivious. A move phase is rigid if $r_i$ moves along $\psi$ and to the endpoint of $\psi$. On the other hand, a non-rigid move phase may finish while $r_i$ is still on the way to the next location. However, we assume that each robot moves at least $\delta$ (in $Z_0$), or reaches the next location if the length of the curve is shorter than $\delta$. We use non-rigid moves without any explicit explanation when it is clear from the context.

An execution is a sequence of configurations, $P(0), P(1), P(2), \ldots$. The execution is not uniquely determined even when it starts from a fixed initial configuration $I$. Rather, there are many possible executions depending on the distance that the robots move, the activation schedule of robots in SSYNC and ASYNC model, and the length of the move phase in ASYNC model.
3. Gathering and convergence
We start with a special case with $|F| = 1$, which is called the gathering problem, that is, the problem is to make the robots form a point. It is also called point formation problem. One weaker version of the gathering problem is the convergence problem that requires the robots to move arbitrarily close to some point. We say a robot system solves the gathering problem (the convergence problem, respectively) if the robots can form a point (converge to a point, respectively) in any execution from a given initial configuration.

3.1. Asynchrony and obliviousness
Clearly, gathering is easy in the FSYNC model because the robots are activated simultaneously and can agree the gathering point based on the common observation (in their own coordinate system). For example, Algorithm 3.1 gathers the robots at the center of gravity of the current positions of robots.

Algorithm 3.1 Gathering algorithm for oblivious robots in the FSYNC model [12]
For robots $r_i$, move to the center of gravity of the current position of robots.

However, Algorithm 3.1 cannot guarantee gathering in the SSYNC model, because semi-synchrony do not always activate all robots. Now, we consider a simplest case of two robots ($n = 2$). Consider an approach that makes a robot to move the position of the other robot. This approach does not work because in the SSYNC model, the two robots may forever swaps their positions. Though we need more theoretical discussion, this is the intuition for the impossibility of gathering of $n = 2$ robots in the SSYNC and ASYNC models. We note that Algorithm 3.1 guarantees convergence in the ASYNC and SSYNC models. Now, we have the following three theorems.

Theorem 1 [12] For the gathering problem and the convergence problem of oblivious robots, we have the following:

(i) The gathering problem is solvable in the FSYNC model.

(ii) The convergence problem is solvable in the SSYNC model.

(iii) When $n = 2$, the gathering problem is unsolvable in the SSYNC model.

Even when $n = 2$, non-oblivious robots with fixed origins of local coordinate systems can solve the gathering problem in the SSYNC model. Suzuki and Yamashita proposed a solution (Algorithm 3.2) that makes the two robots show their initial positions and gather at the midpoint of their initial positions [12].

Algorithm 3.2 Gathering for $n = 2$ non-oblivious robots [12]
For robot $r_i$,

(i) At the first activation, rotate $Z_i$ so that the other robot $r_j$ is on the positive $y$-axis, and move slightly along the positive $x$ axis. (See Figure 3 (i).)

(ii) After the first activation, move to the same direction as the first activation until $r_j$’s position changes twice.

(iii) Then, $r_i$ and $r_j$ can recognize their initial positions, and move to the midpoint.

Depending on the timing of the first observations of the two robots, we have the following two cases:
Case 1: The two robots observe their positions simultaneously. They move in parallel and each robot can recognize the initial position of the other robot by using its initial position.

Case 2: The two robots observe their positions at different time. Let \( r_i \) observe first, and \( r_j \) observe \( r_i \)'s position after \( r_i \) moves (See Figure 3 (ii)). In this case, \( r_j \)'s trajectory intersects \( r_i \)'s negative \( x \)-axis. Thus, both robots can recognize the case when they observed each other. Additionally, \( r_i \) observed initial positions of \( r_i \) and \( r_j \), and \( r_j \) can recognize \( r_i \)'s initial position by the intersection of its own initial position and \( r_i \)'s positive \( y \)-axis.

Consequently, the two robots can agree on the rendezvous point.

When \( n > 2 \) gathering is solvable for ASYNC (thus, SSYNC) robots if the robots have the multiplicity detection or an agreement on the coordinate system. Assume, for contradiction, that there exists an algorithm \( \mathcal{A} \) that solves the gathering problem for oblivious SSYNC robots without multiplicity detection and any agreement on the coordinate system. Let \( R \) be divided into \( R_1 = r_i \) and \( R_2 = R \setminus \{r_i\} \), where all robots in \( R_2 \) has the same coordinate system, which is different from \( r_i \)'s coordinate system. Consider an execution of \( \mathcal{A} \), say \( E \), where gathering of robots \( R \) terminates at time \( t \) and the robots gather at point \( p \). Thus, at time \( t - 1 \), the robots in \( R \) occupy different positions. We have the following two cases:

Case 1: When \( r_i \) is not on \( p \). Then, consider an execution \( E' \) that is same as \( E \) from time 0 to \( t - 1 \), and at time \( t \), only \( r_i \) is not activated. Thus, at time \( t + 1 \), the robots in \( R_2 \) occupy \( p \), and \( R_1 \) is not on \( p \). We denote this configuration by \( C_1 \). It’s the same configuration as the gathering for \( n = 2 \) SSYNC robots. From Theorem 1, algorithm \( \mathcal{A} \) cannot solve the gathering problem.

Case 2: Otherwise. Thus, there exists some \( r_j \) in \( R_2 \) that is not on \( p \) at time \( t - 1 \). Then, consider an execution \( E'' \) that is same as \( E \) from time 0 to \( t - 1 \), and at time \( t \), only \( R'_1 = \{r_j\} \) is not activated. Thus, at time \( t + 1 \), the robots in \( R'_2 = R_1 \setminus \{r_j\} \cup \{r_i\} \) occupies \( p \) and \( R'_1 \) is not at \( p \). We denote this configuration as \( C_2 \). It is easily shown that from configuration \( C_2 \), there exists an execution that either reaches \( C_1 \) configuration or keeps forming \( C_2 \) configuration, which means gathering is impossible.

Consequently, we have the following theorem.

**Theorem 2** [10] Oblivious robots without multiplicity detection and any agreement on the coordinate system cannot solve the gathering problem in the SSYNC model.

Cielievak, Flochini, Prencipe, and Santoro proposed a gathering algorithm for \( n > 2 \) oblivious SSYNC robots with multiplicity detection [2]. The strategy is as follows; first the robots make a single “dense” point with multiple robots, and all the other robots gathers at the point. However, during the gathering, the robots should avoid collisions that make dense points.
Theorem 3 [2] When \( n > 2 \), oblivious robots with multiplicity detection can solve the gathering problem in the ASYNC (thus, SSYNC) model.

Clearly, if the robots have a common coordinate system, they can gather at, for example, the position of the lowest rightmost robot. Motivated by this fact Izumi et al. [9] introduced unreliable compasses and characterized the solvable cases in terms of the types of compasses and their maximum deviation angles.

3.2. Visibility

When the robots have only limited visibility, gathering and convergence can be used as a very important first step for the pattern formation problem. Once they get close enough so that each robot can observe all the other robots, they can execute existing pattern formation algorithms for unlimited visibility robots\(^3\).

Ando, Oasa, Suzuki, and Yamashita proposed a point convergence algorithm for oblivious ASYNC robots [1]. Their algorithm is based on the following simple idea: (i) robots in a connected visibility graph “get closer” and (ii) robots that are mutually visible at time \( t \) remain so for all \( t' > t \). Intuitively, (i) is achieved by making each robot \( r_i \) move toward the center of the smallest enclosing circle of its local view. To predict the moving distance of other robots, their algorithm checks a common point of circles with radius \( V/2 \) and centered at the midpoint between \( r_i \) and \( r_j \) for all \( r_j \) in its local view. Then, (ii) is achieved by stopping \( r_i \) at the furthest common point on the way to the center of the smallest enclosing circle. By the above two properties, the algorithm ensures that the convex hull of the positions of robots shrinks and eventually robots converge to a point.

Flocchini, Prencipe, Santoro, and Widmayer investigated the gathering problem in ASYNC model with common compass [4]. In their algorithm, each robot moves to the bottom and right most robot in its local view. Eventually, the robots gather at the unique bottom and right most robot.

4. Pattern formation

In this section, we briefly overview existing pattern formation algorithms for \( n \geq 3 \) [11, 12, 7, 13]. Suzuki and Yamashita [11] first showed that the symmetry of a given initial configuration \( I \) and the symmetry of a target pattern \( F \) determines whether \( F \) is formable from \( I \). Specifically, \( F \) is formable from \( I \) if and only if \( \rho(I) \) divides \( \rho(F) \). Even though oblivious robots cannot remember the symmetry of the initial configuration, they showed that oblivious robots have the same pattern formation power as non-oblivious robots in the SSYNC and FSYNC models. In other words, during the pattern formation, robots encodes \( \rho(I) \) to their global positions. Later, the result is extended to the ASYNC model [12] by using the embedded pattern formation algorithm by Fujinaga, Ono, Kijima, and Yamashita [7]. All these results assume unlimited visibility of robots. When the visibility of the robots are limited, the robots can neither recognize \( \rho(I) \) nor encode \( \rho(I) \) to their global positions. Yamauchi and Yamashita showed this impossibility for oblivious robots with limited visibility [13], and they showed non-oblivious robots still have the same formation power in the SSYNC and FSYNC models.

4.1. Obliviousness

Suzuki and Yamashita [11] first characterized the class of patterns formable by non-oblivious robots in the SSYNC and FSYNC models. They pointed out that geometric symmetricities of initial configuration \( I = \{p_1(0), p_2(0), \ldots, p_n(0)\} \) and target pattern \( F = \{f_1, f_2, \ldots, f_n\} \) determine whether \( F \) is formable from \( I \). From definition, \( I \) (\( F \), respectively) can be partitioned

\(^3\) For example, the pattern formation algorithm in [8] does not move any robot to the outside of the smallest enclosing circle of the initial configuration during execution.
into regular $\rho(I)$-gons ($\rho(F)$-gons, respectively) all with center $c(I)$ ($c(F)$, respectively). Let $n/\rho(I) = k_I$ and $n/\rho(F) = k_F$. An important property is that all robots observe $I$ even in its local coordinate system can agree on an order of $I_i$’s, such that the distance of the points in $I_i$ from $c(I)$ is no greater than the distance of the points in $I_i$, and that each robots is conscious of the group $I_i$ it belongs to. The same property holds for $F$. Hence, when $\rho(I)$ divides $\rho(F)$, the robots can recognize the correspondence between $I_1, I_2, \ldots, I_{k_I}$ and $F_1, F_2, \ldots, F_{k_F}$ such that in the order of $I_0, I_1, \ldots$, robots occupy the empty $F_j$ with the smallest $j$ ($1 \leq j \leq k_F$). Then, the robots in $I_0$ is considered as (part of) $F_0$, and other robots move in the order of $I_1, I_2, \ldots$ to occupy the nearest position in the corresponding $\rho(F)$-gon. This is an overview of the pattern formation algorithm for non-oblivious robots in the SSYNC and FSYNC models in [11].

The authors also pointed out that when $\rho(I)$ does not divide $\rho(F)$, it is impossible to agree this ordering [11]. This is the source of impossibility of pattern formation. For example, consider pattern formation of $F$ from $I$ where $\rho(I)$ does not divide $\rho(F)$ and $|I_1| > |F_1| = \rho(F)$. The robots in $I_1$ first check the matching between $F_1$ by selecting $\rho(F)$ robots from $I_1$. Remember that the robots are anonymous. The robots in $I_1$ can use the activation schedule, and observation to agree the selection of $\rho(F)$ robots. However, in the worst case, they are activated at the same time and obtain the same observation by the symmetric local coordinate system. Hence, robots can never agree on the selection, and it is impossible to break the symmetricity.

**Theorem 4** [11] Let $I, F \in \mathcal{P}_n$ for any $n \geq 3$. Then $F$ is formable by non-oblivious robots from an initial configuration $I$ in the FSYNC and SSYNC model if and only if $\rho(I)$ divides $\rho(F)$.

Later, Yamashita and Suzuki investigated the formation power of oblivious robots, and showed that oblivious robots have the same formation power except the gathering problem for two robots [12]. Now, we present the sketch of the algorithm in [12]. The robots in $I_{k_I}$ first move slightly outside $C(I)$ if $I_{k_I-1}$ is on $C(I)$. This first step is necessary to keep the symmetricity smaller than $\rho(F)$ during an execution, because when $\rho(I) = \rho(F)$, and $\rho(F)$ contains a regular $k$-gon ($k > \rho(F)$) in its interior, the symmetricity becomes larger than $\rho(F)$. Oblivious robots have no way to reduce the symmetricity thereafter. After this perturbation, the robots in $I_{k_I-1}$ stays there until other robots occupies their target positions in $F$ so that the smallest enclosing circle remains unchanged.

Then, each robot $r_i$, moves to the closest position of the corresponding $\rho(F)$-gon according to the above order by using the following coordinate system $Z^*_r$: the origin is $c(I)$, the positive x-axis is the half line $c(F)p_i(0)$, and the unit distance is $r(P)$. When $r_i$ and $r_j$ are in the same $I_{k_I}$, then their track is symmetric without intersection. Finally, the robots in $I_{k_I}$ move to their destinations with carefully keeping the smallest enclosing circle. During the execution, the symmetricity may decrease, however, the robots recognize the new symmetricity and move according to the new ordering.

**Theorem 5** [12] Let $I, F \in \mathcal{P}_n$ for any $n \geq 3$. Then $F$ is formable by oblivious robots from an initial configuration $I$ in the FSYNC and SSYNC model if and only if $\rho(I)$ divides $\rho(F)$.

### 4.2. Asynchrony

The crucial difference between the ASYNC model and the SSYNC model is that the robots may observe other robots while some of them are moving. Hence, we cannot use the algorithm for the SSYNC model.

Fujinaga, Ono, Kijima and Yamashita proposed the embedded pattern formation problem, that is, the target pattern is embedded on the ground, and the robots autonomously move to form the pattern [7]. The robots should avoid forming a dense point because once two or more robots occupy a position, they may move completely the same thereafter. Hence, the embedded pattern formation problem is to obtain a matching between the initial configuration $I$ and the embedded
target pattern $F$. In [7], they defined and showed the existence of a *clockwise matching* (CWM) $f_{CWM} : I \rightarrow F$ that is (i) a perfect matching between $I$ and $F$, (ii) an optimum matching in the sense that the sum of the Euclidean distance between $p_i(0)$ and $f_{CWM}(p_i(0))$ for all $r_i$ is minimized. The optimality guarantees that any straight movement of $r_i$ to $f_{CWM}(p_i(0))$.

**Theorem 6** [7] Let $I, F \in \mathcal{P}_n$ for any $n \geq 3$ without multiplicity. Then an embedded pattern $F$ is formable from $I$ in the ASYNC (thus, SSYNC and FSYNC) model.

Fujinaga, Yamauchi, Kijima, and Yamashita proposed a pattern formation algorithm based on the embedded pattern formation [8]. Their algorithm first forms an agreement on a coordinate system and embeds the target pattern $F$, and moves the robots according to the CWM algorithm in [7]. Given a pattern $P$, let $Z(P, q)$ be the coordinate system defined by the smallest enclosing circle and position $q \in P$ as shown in Figure 4. An important property is that, when $\rho(I)$ divides $\rho(F)$, $Z(I, r_i)[F] = Z(I, r_j)[F]$ for any $r_i, r_j \in I_0$. In [8], the proposed a pattern formation algorithm that embeds $F$ by using $Z(I, r_i)$ ($r_i \in I_0$). The difficulty is that during the formation, no robot cannot go inside $C(I_0)$. The authors used a mapping from the two dimensional Euclidean space to a cylinder defined by the distance from $c(I)$ and the radius from the x-axis. The CWM is computed on the cylinder, and the corresponding track in the two dimensional Euclidean space is a curve in the $\text{Interior}(I) \setminus \{C(I_0) \cup \text{Interior}(I_0)\}$. Hence, the robots moving along the curve never go inside $C(I_0)$.

**Theorem 7** Let $I, F \in \mathcal{P}_n$ for any $n \geq 3$ without multiplicity. Then $F$ is formable from an initial configuration $I$ by oblivious robots in the ASYNC model if and only if $\rho(I)$ divides $\rho(F)$.

### 4.3. Visibility

Robots with limited visibility can observe other robots within distance $V$, and they are not aware of the global configuration, such as their smallest enclosing circle, and the symmetricity of their positions.

A necessary and sufficient condition for the pattern formation with unlimited visibility is based on the “global” symmetricity [11]. The pattern formation algorithms [11, 12, 8] also use the positions of robots as the indicator of the formation process while robots are oblivious.

Yamauchi and Yamashita showed that robots with limited visibility cannot form target pattern $F$ from initial configuration $I$ even when $\rho(I)$ divides $\rho(F)$ in the FSYNC model [13]. The counter example is a visibility graph with low connectivity, where local observation leads the robots to increase the global symmetricity. They further showed that non-oblivious robots can regain the symmetricity of the initial configuration by recording all local observations and all outputs of compute phases.
The algorithm first executes the convergence algorithm [1], and after robots get closer enough, executes existing a pattern formation algorithm for robots with unlimited visibility. Because algorithm [1] guarantees that the output track is a straight line, the robots can compute the rotation angle between any two consecutive outputs. Hence, it is clear that when all moves are rigid, each robot can compute its initial position. The authors showed that the robots can draw its initial position after they get closer enough so that the robots obtains initial positions all other robots. When the moves are non-rigid, they further showed that the robots can regain $\rho(I)$ in the FSYNC model because the local history of observations and output traces break the symmetry.

**Theorem 8** [13] Let $I, F \in \mathcal{P}_n$ for any $n \geq 3$ without multiplicity.

(i) There exist infinitely many patterns $F$ that is not formable from $I$ by non-oblivious robots with limited visibility in the FSYNC model, even when $\rho(I)$ divides $\rho(F)$.

(ii) $F$ is formable by non-oblivious robots with limited visibility from an initial configuration $I$ in the FSYNC model if and only if $\rho(I)$ divides $\rho(F)$.

(iii) $F$ is formable by non-oblivious robots with limited visibility and rigid moves from an initial configuration $I$ in the SSYNC model if and only if $\rho(I)$ divides $\rho(F)$.

5. Concluding Remark and Future Directions

We surveyed existing results on pattern formation problems by mobile robots. The pattern formation power of mobile robots is determined by their asynchrony, obliviousness, and visibility. One of the future directions is to reveal formation power of robots with limited visibility. For example, the pattern formation problem for robots with limited visibility in the ASYNC model has not been solved. Another direction is to develop a randomized solution for the problems that have been shown to be deterministically unsolvable. Such a randomized and localized mobile robot system model would be related to wireless sensor networks, vehicular networks, chemical reactions, and so on. Recently, mobile robot system model founds new applications in molecular robotics where system size (i.e., the number of robots) is in the Avogadro number scale and robots with very small capabilities cooperate by interacting with each other randomly. We are interested in whether and how the size of the system has effect on their computational power.

Acknowledgment

This work is supported in part by JSPS Grant-in-Aid for Scientific Research on Innovative Areas (No.24104519), and JSPS Grant-in-Aid for Scientific Research (No. 23700019).

References