SIAM J. COMPUT.
Vol. 28, No. 4, pp. 1347–1363
© 1999 Society for Industrial and Applied Mathematics

DISTRIBUTED ANONYMOUS MOBILE ROBOTS: FORMATION OF GEOMETRIC PATTERNS*

ICHIRO SUZUKI† AND MASAFUMI YAMASHITA‡

Abstract. Consider a system of multiple mobile robots in which each robot, at infinitely many unpredictable time instants, observes the positions of all the robots and moves to a new position determined by the given algorithm. The robots are anonymous in the sense that they all execute the same algorithm and they cannot be distinguished by their appearances. Initially they do not have a common x-y coordinate system. Such a system can be viewed as a distributed system of anonymous mobile processes in which the processes (i.e., robots) can "communicate" with each other only by means of their moves. In this paper we investigate a number of formation problems of geometric patterns in the plane by the robots. Specifically, we present algorithms for converging the robots to a single point and moving the robots to a single point in finite steps. We also characterize the class of geometric patterns that the robots can form in terms of their initial configuration. Some impossibility results are also presented.

Key words. distributed algorithms, anonymous robots, mobile robots, multiagent systems, formation of geometric patterns

AMS subject classification. 68Q99

PII. S009753979628292X

1. Introduction.

Suppose that a schoolteacher wants her 100 children in the playground to form a circle so that, for instance, they can play a game. She might draw a circle on the ground as a guideline or even give each child a specific position to move to. What if the teacher does not provide such assistance? Even without such assistance, the children may still be able to form a sufficiently good approximation of a circle if each of them moves adaptively based on the movement of other children and knowledge of the shape of a circle. If successful, this method can be called a *distributed* solution to the circle formation problem for children.

A similar distributed approach can be used for controlling a group of multiple mobile robots. The main idea is to let each robot execute a simple algorithm and plan its motion adaptively based on the observed movement of other robots, so that the robots as a group will achieve the given goal. The objective of this paper is to give a formal discussion on the power and limitations of the distributed control method in the context of the formation problems of geometric patterns in the plane.

The problem of forming an approximation of a circle having a given diameter by identical mobile robots was first discussed by Sugihara and Suzuki [13].[1] Assuming

*Received by the editors November 13, 1996; accepted for publication (in revised form) July 24, 1997; published electronically March 30, 1999. This work was supported in part by a Scientific Research Grant-in-Aid from the Ministry of Education, Science and Culture in Japan (0668032, 07243219, 076803604), the National Science Foundation under grant IRI-9307506, the Office of Naval Research under grant N00014-94-1-0284, and an endowed chair supported by Hitachi Ltd. at the Faculty of Engineering Science, Osaka University. An earlier version of some of the results contained in this paper appears as "Formation and agreement problems for anonymous mobile robots" in *Proc. 31st Annual Allerton Conference on Communication, Control, and Computing*, University of Illinois, Urbana, 1993, pp. 93–102.

http://www.siam.org/journals/sicomp/28-4/28292.html

†Department of Electrical Engineering and Computer Science, University of Wisconsin–Milwaukee, P.O. Box 784, Milwaukee, WI 53201 (suzuki@cs.uwm.edu).

‡Department of Electrical Engineering, Faculty of Engineering, Hiroshima University, Kagamiyama, Higashi-Hiroshima 739, Japan. Presently with Department of Computer Science and Communication Engineering, Kyushu University, Fukuoka, 812-8581, Japan (mak@csce.kyushu-u.ac.jp).

[1] In our terminology, the problem they consider is a convergence problem for a circle.

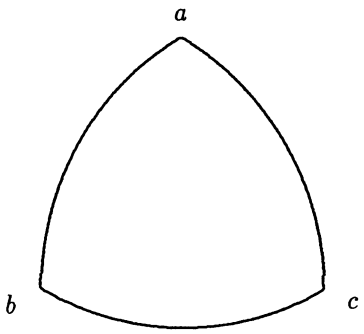

FIG. 1.1. *Reuleux's triangle.*

that the positions of the robots are the only information available, they proposed a simple heuristic distributed algorithm (to be executed independently by all robots), which, according to simulation results, sometimes brings the robots to a pattern reminiscent of a Reuleaux's triangle (Figure 1.1) rather than a circle. Tanaka [16] later improved their algorithm and demonstrated, using simulation, that his new algorithm avoids this problem and generates a better approximation of a circle. In essence, in his algorithm each robot simply adjusts its position regarding the midpoint of the positions of the nearest and farthest neighbors as the center of the circle to which the robots are converging, while moving away from its nearest neighbor if the distance to that midpoint is approximately equal to the given target radius. Figure 1.2 shows the behavior of 50 robots executing his algorithm starting from an initial distribution generated randomly. This extremely simple algorithm demonstrates the potential of the distributed method. The circle formation problem was also discussed recently by Debest [2] from the viewpoint of self-stabilization. A system is said to be *self-stabilizing* if it recovers from any finite number of transient errors [12], and thus self-stabilizing robot algorithms are robust against a finite number of sensor and control errors.

Formation problems of geometric patterns are closely related to certain agreement problems. Agreement on a common x-y coordinate system by the robots, for instance, can greatly reduce the complexity of motion coordination algorithms; e.g., convergence toward a single point can easily be solved by moving all the robots toward point $(0, 0)$ of the common coordinate system. However, such a simple solution is not possible if the robots have only their own local coordinate systems, whose origins may or may not agree. It is sometimes assumed in the literature, therefore, that either there exists a global coordinate system or that some navigation devices (e.g., a variety of potential functions [18], compasses [3], or beacons and lighthouses [4]) are available to compensate for the lack of such a system. Note here that the agreement problem on a common coordinate system can (partially) be reduced to certain formation problems: If the robots can form (i.e., gather at) a single point, then they can agree to use that point as the origin of the common coordinate system. Similarly, formation of a circle implies agreement on both the origin and the unit distance (i.e., the center and the radius of the circle). Formation of a symbol "$\top$" implies agreement on the origin, the unit distance, and the positive x-direction, i.e., agreement on a common x-y coordinate system.

Related work on the distributed robot control method includes the following. Wang and Beni [17] considered a cellular robotic system consisting of a large number

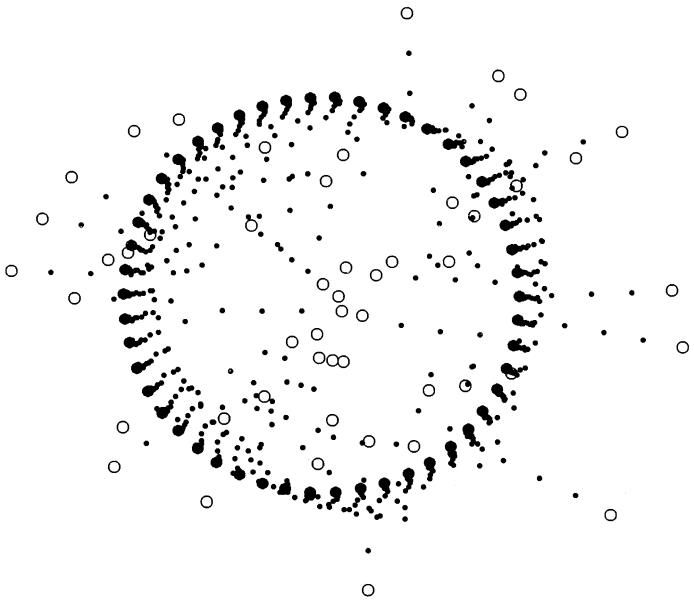

FIG. 1.2. *Hollow circles are the initial positions of* 50 *robots. Solid circles are their final positions after execution of Tanaka's algorithm. Small dots represent their intermediate positions.*

of robots that operate in a cellular space under distributed control. They discussed the problem of generating certain one- and two-dimensional cellular patterns using distributed control and showed how the technique can be applied to the design of sensor arrays and escape systems. Fukuda and Nakagawa [6] and Kawauchi, Inaba, and Fukuda [7] considered a dynamically reconfigurable robotic system called CEBOT, which consists of many simple cells that can detach and combine autonomously to change its overall shape, depending on the task and the environment. Kokaji [8] and Murata, Kurokawa, and Kokaji [10] designed self-reorganizing systems called Fractal Machine and Fractum, respectively, based on a similar idea (but unlike CEBOT, these systems consist of homogeneous units) and discussed dynamic reconfiguration based on a set of local rules. Fujimura [5] investigated how planning algorithms, knowledge about the environment, and action intervals of the robots affect the overall performance of two robots moving toward their respective goal positions while avoiding collision. Sugihara and Suzuki [13], [14], and Suzuki and Yamashita [15] considered formation and agreement problems for anonymous mobile robots in the plane. Work by others includes swarm intelligence [1] and collective behavior of multiple robots [9], [11].

The main emphasis of most of the work mentioned above has been on the development of heuristic algorithms for various problems, and rigorous proofs of the correctness of these algorithms have not been given. In contrast, as we stated earlier in this paper, we conduct a formal investigation on the power and limitations of the distributed control method.

We model a robot as a mobile processor with infinite memory and a sensor for detecting the positions of other robots[2] that repeatedly becomes *active* at infinitely

[2]We assume that the sensor cannot measure the velocity or acceleration of other robots and that other navigation devices such as compasses and beacons are not available.

many unpredictable time instants. (At other times it is *inactive*.) We assume that initially the robots do not have a common x-y coordinate system and that the local x-y coordinate systems of the robots may not agree on the location of the origin, the unit distance, or the direction of the positive x-axis. Each time a robot becomes active, using its sensor it observes the positions of all the robots in terms of its own local x-y coordinate system and moves to a new position determined by the given deterministic algorithm.[3] The algorithm is *oblivious* if the new position is determined only from the positions of the robots observed at that time instant. Otherwise, it is *nonoblivious*, and the new position may depend also on the observations made in the past. Note that oblivious algorithms are self-stabilizing by definition. To simplify the discussion and bring forth the fundamental issues of the problem, in this paper we assume that (1) the initial positions of the robots are all distinct, (2) the time it takes for a robot to move to its new position is negligibly small, and (3) a robot is a point (so two robots can occupy the same position simultaneously and never collide). The robots are anonymous in the sense that (1) they do not know their identifiers, (2) they all use the same algorithm for determining the next position, and (3) they cannot be distinguished by their appearances.

Let π be a predicate describing a geometric pattern, such as a point, a regular polygon, a line segment, etc. On the one hand, we say that an algorithm ψ solves the *convergence problem* for π if the robots' distribution converges to one that satisfies π, regardless of the number n of robots, their initial distribution, and the timing with which they become active. On the other hand, we say that ψ solves the *formation problem* for π if the robots eventually reach a distribution that satisfies π in a finite number of steps, regardless of n, their initial distribution, and the timing with which they become active. (See section 2 for formal definitions of these concepts.)

We begin with a simple problem of converging the robots toward a single point. (That is, this is the convergence problem for a predicate π that describes a point. Note that the process of convergence need not terminate in finite steps.) Note again that since the robots do not have a common x-y coordinate system, we cannot simply use an algorithm such as "move toward the origin $(0, 0)$." For this problem we give a simple oblivious algorithm.

We also consider the formation problem for a point, in which the robots must form (i.e., gather at) a single point in finite steps. We show that this problem can be solved by a nonoblivious algorithm for any $n \geq 2$ and by an oblivious algorithm for any $n \geq 3$, but it is not solvable by any oblivious algorithm for the case $n = 2$, where n is the total number of robots.

Finally, we characterize the class of geometric patterns for which the formation problem is solvable in our model. We do so by first examining the class of patterns that the robots can form, starting from a given initial configuration. Our main observation is that since the robots may happen to become active simultaneously all the time (i.e., their motions turn out to be synchronized) and (by definition) algorithms are required to solve the given problem regardless of the timing with which the robots become active, the robots may not be able to break the "symmetry" that exists in their initial distribution by executing an algorithm (which is deterministic by definition). Based on this and using techniques that have been developed for anonymous complete networks in [19], [20], we prove that the formation problem is solvable by an algorithm (in the sense defined above) only for two patterns: a point and a regular n-gon. The

[3] In this paper we do not consider nondeterministic algorithms that allow a robot to randomly select its next position from two or more candidates.

algorithm we present for the formation of a regular n-gon is nonoblivious. Whether an oblivious algorithm exists for this problem remains open.

We present necessary definitions and basic assumptions in section 2. Convergence and formation problems for a point are discussed in section 3. Section 4 gives a characterization of the class of geometric patterns that the robots can form in our model. Discussions and concluding remarks are presented in section 5.

2. Definitions and basic assumptions. We formalize the concepts described in section 1. Let $r_1, r_2, \ldots, r_n$ be the robots in a two-dimensional space. (The subscript i of r_i is used for convenience of explanation. The robots do not know their identifiers.) We denote by $Z_i = (o_i, d_i, u_i)$, $1 \leq i \leq n$, the local x-y coordinate system of r_i, where o_i, d_i, and u_i denote the position of the origin, direction of the positive x-axis, and size of the unit distance, respectively, under Z_i. It is possible that $Z_i \neq Z_j$ for some i and j, but the robots are assumed to have a common sense of orientation so that in each Z_i, the positive y-direction is 90 degrees counterclockwise from the positive x- direction. As we describe below, all robot positions that r_i observes and computes are given in terms of Z_i.

We assume discrete time $0, 1, 2, \ldots$ and let $p_i(t)$ be the position of r_i at time instant t, where $p_i(0)$ is the initial position of r_i. We assume that $p_1(0), p_1(0), \ldots, p_n(0)$ are all distinct. Define $P(t) = \{p_i(t) | 1 \leq i \leq n\}$ to be the multiset of the positions of the robots at time t. ($P(t)$ is a multiset, since we assume that two robots can occupy the same position simultaneously.) For any point p, we denote by $[p]_j$ the position of p given in terms of Z_j and define $[P(t)]_j = \{[p_i(t)]_j | 1 \leq i \leq n\}$. Thus $[P(t)]_j$ shows how r_j views the distribution $P(t)$ in terms of its own Z_j. Note that if $Z_j \neq Z_k$, then it is possible that $[P(t)]_j \neq [P(t)]_k$; i.e., r_j and r_k may observe distribution $P(t)$ differently. However, $[P(t)]_j = [P(t)]_k$ may hold even if $p_j(t) \neq p_k(t)$. In this case, r_j and r_k are located at different positions, but $P(t)$ looks identical to them.

At each time instant t, every robot r_i is either *active* or *inactive*. Without loss of generality we assume that at least one robot is active at every time instant. We use A_t to denote the set of active robots at t, and call the sequence $\mathcal{A} = A_0, A_1, \ldots$ an *activation schedule*. We assume that every robot becomes active at infinitely many time instants, but no additional assumptions are made on the timing with which the robots become active. Thus $\mathcal{A}$ need satisfy only the condition that every robot appears in infinitely many A_t's. Note that a special case is when every robot appears in A_t for every t; in this case we say that the robots are *synchronized*.

The algorithm that a robot uses is a function ψ such that, for any given sequence $(Q_1, p_1), (Q_2, p_2), \ldots, (Q_m, p_m)$ of pairs of a multiset Q_ℓ of points and a point $p_\ell \in Q_\ell$, $\psi((Q_1, p_1), (Q_2, p_2), \ldots, (Q_m, p_m))$ is a point such that the distance between p_m and $\psi((Q_1, p_1), (Q_2, p_2), \ldots, (Q_m, p_m))$ is at most 1. The position of a robot at $t \geq 1$ is determined by $P(0)$, $\mathcal{A}$, and ψ, as follows.

For any $t \geq 0$, if $r_i \notin A_t$ (r_i is inactive), then $p_i(t+1) = p_i(t)$; i.e., r_i does not move. If $r_i \in A_t$ (r_i is active), then let $0 \leq t_1 \leq t_2 \leq \cdots \leq t_m = t$ be the time instants when r_i has been active, and for each $1 \leq \ell \leq m$, let $Q_\ell = [P(t_\ell)]_i$ and $p_\ell = [p_i(t_\ell)]_i$ be the distribution that r_i observed and the position of r_i at t_ℓ, respectively. (Note that Q_ℓ and p_ℓ are given in terms of Z_i.) Then $p_i(t+1) = p$, where p is the point such that $[p]_i = \psi((Q_1, p_1), (Q_2, p_2), \ldots, (Q_m, p_m))$. That is, r_i moves to point $\psi((Q_1, p_1), (Q_2, p_2), \ldots, (Q_m, p_m))$ of Z_i. By the restriction on ψ stated above, the maximum distance that r_i can move in one step is the unit distance 1 of Z_i, which corresponds to some physical distance $\epsilon_i > 0$. Note that every robot is then capable of moving over distance at least $\epsilon = \min\{\epsilon_1, \epsilon_2, \ldots, \epsilon_n\} > 0$ in one step.

That is, r_i observes the distribution of the robots only when it is active, and its next position depends only on ψ and the distributions that r_i has observed so far. The p_ℓ in pair (Q_ℓ, p_ℓ) shows that r_i is always aware of its current position in Z_i. Algorithm ψ is said to be *oblivious* if $\psi((Q_1,p_1),(Q_2,p_2),\ldots,(Q_m,p_m)) = \psi((Q_m,p_m))$ for any $(Q_1,p_1),(Q_2,p_2),\ldots,(Q_m,p_m)$. In this case, the move of a robot depends only on the current configuration of the robots. Otherwise, ψ is *nonoblivious*. Note that the robots are anonymous in the following sense: (1) function ψ is common to all the robots, (2) the identifier i of robot r_i is not an argument of ψ, and (3) $[P(t)]_i$ contains only the positions of the robots (but not their identities).

Let π be a predicate over the set of multisets of points that is invariant under any rotation, translation, and uniform scaling. For example, π might be true iff the given points are on the circumference of a circle or on a line segment. For such π, we consider two types of problems: the *convergence problem* and the *formation problem*. An algorithm ψ is said to solve the convergence problem for π if, as t goes to infinity, $P(t)$ converges to a distribution that satisfies π, regardless of the number n of robots, initial distribution $P(0)$, and activation schedule $\mathcal{A}$. In contrast, in the formation problem the robots must reach a distribution satisfying π in finite steps and "halt." That is, an algorithm ψ is said to solve the formation problem for π if there exists some time instant t' such that $P(t')$ satisfies π and $p_i(t') = p_i(t'+1) = \cdots$ for all $1 \le i \le n$, regardless of n, $P(0)$, and $\mathcal{A}$. Since the robots have no knowledge of the underlying coordinate system, the robots can only converge to or form a pattern similar to the given goal pattern. The restriction on π stated above was introduced for this reason. All predicates discussed in the following sections satisfy this condition.

3. Convergence and formation problems for a point. Formally, the problem of converging the robots to a point is stated as the convergence problem for predicate π_{point}, where $\pi_{point}(p_1,\ldots,p_n) =$ true iff $p_i = p_j$ for any $1 \le i,j \le n$. We call this problem C-POINT. The corresponding formation problem for π_{point} is called F-POINT. Note that in F-POINT, all robots must occupy a single point in finite steps, whereas in C-POINT they need only converge to a single point. These are perhaps some of the simplest problems one could consider. Nevertheless, the discussions presented in this section can serve as an introduction to the technical results given in the rest of the paper. An algorithm that solves F-POINT also solves C-POINT.

For convenience, we present all algorithms by giving an informal description of the behavior of the robots executing it, instead of giving a formal definition of function ψ. Converting the informal description into a formal definition of ψ is straightforward.

It is easy to show that the following oblivious algorithm $\psi_{c-point(2)}$ solves C-POINT for the case $n = 2$.

ALGORITHM $\psi_{c-point(2)}$—OBLIVIOUS.

Each time r_i becomes active, it moves toward[4] the midpoint m of its current position and that of the other robot r_j. □

Suppose that we modify $\psi_{c-point(2)}$ so that each robot moves toward the position of the other robot. Then the two robots will continue to swap their positions if they are mutually reachable in one step and always become active simultaneously. (Recall that we assume robots never collide with each other.) Thus this modified algorithm does not solve C-POINT for $n = 2$.

Note that if exactly one robot becomes active at every time instant, then the

[4]Unless otherwise stated, "a robot moves toward point p" means that "a robot moves to the point p' closest to p that is reachable in one step from the current position." Of course, $p = p'$ if p is reachable in one step.

two robots executing $\psi_{c-point(2)}$ will never occupy the same point. Thus oblivious algorithm $\psi_{c-point(2)}$ does not solve F-POINT for $n = 2$. In fact, we have the following theorem.

THEOREM 3.1. *There is no oblivious algorithm for solving F-POINT for the case* $n = 2$.

Proof. Suppose that there is an oblivious algorithm ψ that solves F-POINT for two robots r_i and r_j. Note that since ψ is oblivious, the moves of the robots depend only on Z_i, Z_j and their current positions.

We first show that there exist distinct positions p and q of r_i and r_j, respectively, such that either (1) ψ moves r_i from p to q and r_j from q to q, or (2) ψ moves r_i from p to p and r_j from q to p. (That is, ψ moves exactly one robot to the position of the other if both robots become active simultaneously.) To see this, assume that such positions do not exist. Consider a scenario $\mathcal{S}$ in which r_i and r_j, located at distinct positions p and q, respectively, at time $t-1$ occupy the same position r at time t. Now we show that we can modify this scenario and obtain another scenario in which the robots never occupy the same position simultaneously. There are two cases.

Case 1. Both r_i and r_j are active at time $t-1$ in $\mathcal{S}$. By assumption, $r \neq p$ and $r \neq q$. Thus if exactly one robot, say, r_i, happens to be active at $t-1$, then at time t, r_i is located at r and r_j at q, where $r \neq q$.

Case 2. Exactly one robot is active at $t-1$ in $\mathcal{S}$. Suppose that r_i is active at $t-1$ but r_j is not. Then $r = q$. So if both robots happen to be active at $t-1$, then at time t, r_i is located at q and r_j at some point s, where by assumption $s \neq q$.

Using this argument repeatedly, we can construct an infinite sequence of moves in which the robots never occupy the same position simultaneously. (We can do so in such a way that each robot becomes active infinitely many times, since either of the robots can be chosen to be inactive in Case 1.) So ψ does not solve F-POINT. This is a contradiction.

Now consider an initial distribution $P(0) = \{p, q\}$ in which r_i and r_j are at p and q, respectively, and ψ moves r_i from p to q, and r_j from q to q; see Figure 3.1(a). (The case in which ψ moves r_j to the positions of r_i is similar.) Now, by modifying Z_i through translation and rotation, we can construct another configuration in which r_i observes distribution $P(0)$ the same way as r_j; i.e., $[P(0)]_i = [P(0)]_j$ and $[p]_i = [q]_j$; see Figure 3.1(b). Then ψ moves r_i and r_j in the same manner in the new configuration and, of course, ψ moves r_j in the same manner in both configurations (namely, from q to q). Therefore, in the new configuration ψ moves r_i from p to p and r_j from q to q. Then, since ψ is oblivious, both robots remain in their respective initial positions forever. Thus ψ does not solve F-POINT. This is a contradiction. □

However, F-POINT can be solved for two robots by the following *nonoblivious* algorithm $\psi_{f-point(2)}$.[5]

ALGORITHM $\psi_{f-point(2)}$—NONOBLIVIOUS.

When r_i becomes active for the first time, it translates and rotates its coordinate system[6] Z_i so that

1. r_i is at $(0, 0)$ of Z_i, and
2. the other robot r_j is on the positive y-axis of Z_i, say, at $(0, a)$ for some $a > 0$.

[5]If all robots are *known* to be active at every time instant (i.e., the robots are "synchronous"), then a simple oblivious algorithm that moves both robots toward the midpoint of their current positions solves F-POINT for two robots.

[6]Formally, r_i cannot modify Z_i in our framework, but the effect of such a transformation can easily be simulated within the framework.

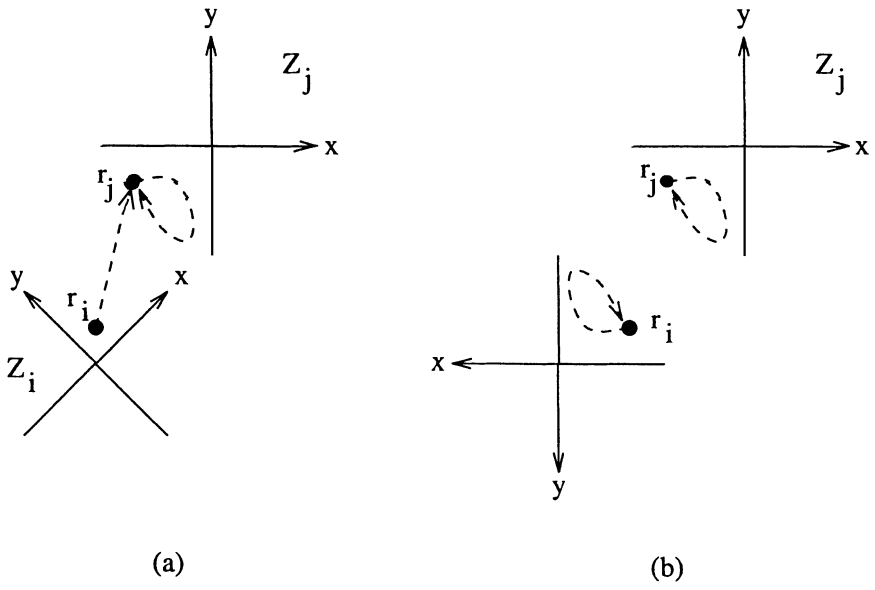

FIG. 3.1. (a) r_i *moves but* r_j *does not.* (b) *After modification of* Z_i.

Then it moves in the positive x direction of Z_i, over any nonzero distance. It then continues to move in the same direction each time it becomes active until it observes that the position of r_j has changed twice.

Now, r_i knows line ℓ that contains the first two distinct positions of r_j that r_i has observed. (Note that by symmetry ℓ is the x-axis of r_j's coordinate system Z_j.) Then using Lemma 3.2, r_i finds the initial position of r_j and moves to the midpoint of the initial positions of r_i and r_j. □

Lemma 3.2, which follows immediately from the description of $\psi_{f-point(2)}$, shows that robots r_i and r_j executing $\psi_{f-point(2)}$ eventually find out which of them became active first for the first time and what their initial distribution was.

LEMMA 3.2. *Let* t_i *and* t_j *be the time instants at which* r_i *and* r_j, *respectively, become active for the first time in* $\psi_{f-point(2)}$. *Then the following hold.*

1. *The trajectory of* r_i *and the trajectory of* r_j *are parallel iff* $t_i = t_j$. *In this case, each robot sees the other robot at its initial position at* $t_i (= t_j)$ (*Figure* 3.2(a)).
2. *The trajectory of* r_j *intersects the negative* x*-axis of* Z_i *iff* $t_i < t_j$. *In this case,* r_i *sees* r_j *at its initial position, and* r_i*'s initial position is the foot of the perpendicular drop from* r_j*'s initial position to the line containing the trajectory of* r_i (*Figure* 3.2(b)).
3. *The trajectory of* r_i *intersects the negative* x*-axis of* Z_j *iff* $t_j < t_i$. *In this case,* r_j *sees* r_i *at its initial position, and* r_j*'s initial position is the foot of the vertical drop from* r_i*'s initial position to the line containing the trajectory of* r_j.

THEOREM 3.3. *Algorithm* $\psi_{f-point(2)}$ *solves problem F-POINT for* $n = 2$.

Proof. A key observation is the following: When r_i observes that the position of r_j has changed twice, r_j must have already observed that r_i's position has changed at least once and thus r_j knows where the x-axis of Z_i is. Similarly, r_j will know that r_i knows where the x-axis of Z_j is. Then the correctness of $\psi_{f-point(2)}$ follows from Lemma 3.2. □

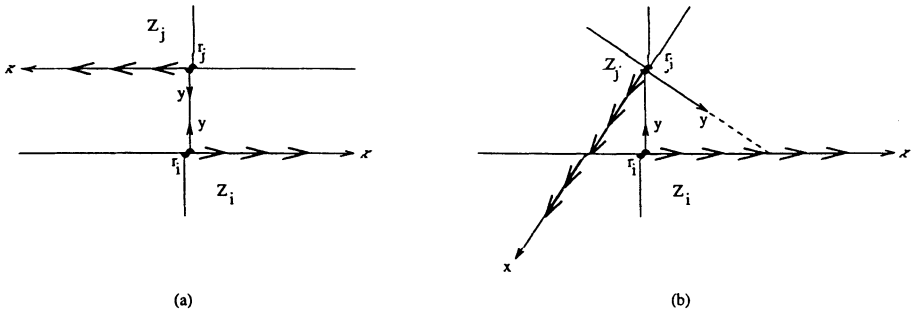

FIG. 3.2. *Illustration for* $\psi_{f-point(2)}$*;* (a) $t_i = t_j$, (b) $t_i < t_j$.

Finally, we have the following result on F-POINT and C-POINT for $n \geq 3$.

THEOREM 3.4. *There is an oblivious algorithm for solving F-POINT (and thus C-POINT) for* $n \geq 3$.

Proof. It suffices to give an oblivious algorithm $\psi_{f-point(n)}$ that solves F-POINT. The idea is the following. Starting from distinct initial positions, we move the robots in such a way that eventually there will be exactly one position, say, p, that two or more robots occupy. Once such a distribution is reached, all robots that are not located at p move toward p in such a way that no two robots will occupy the same position at any location other than p. Then all robots eventually occupy p, solving F-POINT.

Such a distribution can be obtained if each robot, each time it becomes active, determines which of the following cases applies and moves to a new position (or remains stationary) as specified. Since a robot's action is based only on the current robot distribution, this strategy can be implemented as an oblivious algorithm.

Case 1. $n = 3$; p_1, p_2, and p_3 denote the positions of the three robots.

1.1. If $n = 3$ and p_1, p_2, and p_3 are collinear with p_2 in the middle, then the robots at p_1 and p_3 move toward p_2 while the robot at p_2 remains stationary. Then eventually two robots occupy p_2.

1.2. If $n = 3$ and p_1, p_2, and p_3 form an isosceles triangle with $|\overline{p_1p_2}| = |\overline{p_1p_3}| \neq |\overline{p_2p_3}|$, then the robot at p_1 moves toward the foot of the perpendicular drop from its current position to $\overline{p_2p_3}$ in such a way that the robots do not form an equilateral triangle at any time, while the robots at p_2 and p_3 remain stationary. Then eventually the robots become collinear and the problem is reduced to part 1.1.

1.3. If $n = 3$ and the lengths of the three sides of triangle $p_1p_2p_3$ are all different, say, $|\overline{p_1p_2}| > |\overline{p_1p_3}| > |\overline{p_2p_3}|$, then the robot at p_3 moves toward the foot of the perpendicular drop from its current position to $\overline{p_1p_2}$ while the robots at p_1 and p_2 remain stationary. Then eventually the robots become collinear and the problem is reduced to part 1.1.

1.4. If $n = 3$ and p_1, p_2, and p_3 form an equilateral triangle, then every robot moves towards the center of the triangle. Since all robots can move up to at least a constant distance $\epsilon > 0$ in one step, if part 1.4 continues to hold then eventually either the robots meet at the center, or the triangle they form becomes no longer equilateral and the problem is reduced to part 1.2 or part 1.3.

Case 2. $n \geq 4$; C_t denotes the smallest enclosing circle of the robots at time t.

2.1. If $n \geq 4$ and there is exactly one robot r in the interior of C_t, then r moves toward the position of any one robot, say, r', on the circumference of C_t while all other robots remain stationary. Then eventually r and r' occupy the same position.

2.2. If $n \geq 4$ and there are two or more robots in the interior of C_t, then these robots move toward the center of C_t while all other robots remain stationary (so that the center of C_t remains unchanged). Then eventually at least two robots reach the center.

2.3. If $n \geq 4$ and there are no robots in the interior of C_t, then every robot moves toward the center of C_t. Since all robots can move up to at least a constant distance $\epsilon > 0$ in one step, if part 2.3 continues to hold, then eventually the radius of C_t becomes at most ϵ. Once this happens, then the next time some robot moves, say, at t', either (i) two or more robots occupy the center of C_t or (ii) there is exactly one robot r at the center of C_t, and therefore there is a robot that is not on $C_{t'}$ (and the problem is reduced to part 2.1 or part 2.2) since a cycle passing through r and a point on C_t intersects with C_t at most at two points. □

Suppose that for $1 \leq i \leq n$, robot r_i has (privately) chosen a directed line ℓ_i that passes through its initial position. Algorithm $\psi_{f-point(2)}$ uses a technique with which all robots can simultaneously "broadcast" the locations and directions of $\ell_1, \ell_2, \ldots, \ell_n$. The basic idea is that each robot r_i moves repeatedly along ℓ_i in the given direction until it observes that every r_j, $j \neq i$ has changed positions at least twice (i.e., until r_i sees r_j at three or more distinct positions). Then, as we explained in the proof of Theorem 3.3, every r_j, $j \neq i$ must have (become active and) seen r_i at two or more distinct positions along ℓ_i, and thus r_j can conclude that the ℓ_i that r_i has chosen passes through the first two distinct positions of r_i that r_j has observed and that ℓ_i is oriented in the direction from the first to the second positions of r_i that r_j has observed. Care must be taken so that r_i continues to move at least one more time (to any distinct position) after observing that every r_j has changed position at least twice, since at this moment some r_j might have observed r_i only at two distinct positions.

Another problem is that, since the robots are indistinguishable by their appearances, if $n > 2$, then r_j may not be able to determine how r_i has moved, given the robot distributions at two time instants. To cope with this, if $n > 2$, then we let each robot r_i memorize the distance $a_i > 0$ to its nearest neighbor when it becomes active for the first time and move at most distance $a_i/2^{k+1}$ in the kth move. Then each r_i will remain in the interior of the $a_i/2$-neighborhood of its initial position, and thus every robot can correctly determine which robot has moved to which position even after it has remained inactive for a long time.

4. Achievable geometric patterns. In this section we characterize the class of geometric patterns that the robots can form regardless of the activation schedule $\mathcal{A}$, starting from a fixed initial configuration. For simplicity of explanation we assume that each robot r_i is located at the origin of its coordinate system Z_i at time 0. Essentially the same result holds even without this assumption.

Whether or not a particular geometric pattern can be formed depends not only on the given initial positions of the robots but also on their local x-y coordinate systems. For example, suppose that, initially, four robots r_1, r_2, r_3, and r_4 form a square in counterclockwise order, where r_2 is at position $(1, 0)$ of Z_1, r_3 is at position $(1, 0)$

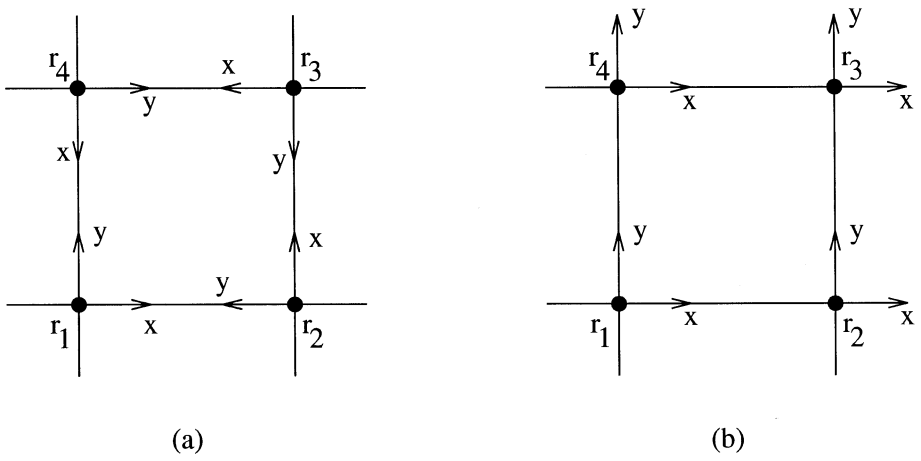

FIG. 4.1. *Two configurations of four robots that are* (a) *symmetric,* (b) *not symmetric.*

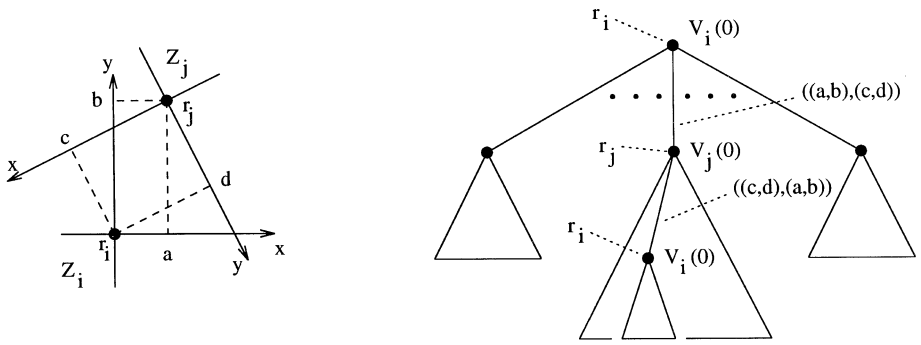

FIG. 4.2. r_i, r_j, *and views* $V_i(0)$ *and* $V_j(0)$.

of Z_2, and so on, as shown in Figure 4.1(a). Intuitively, the robots have the same "view," and thus, if they are synchronized, then they will never be able to break symmetry and form a pattern other than a square, but if the direction of the positive x-axis happens to be the same for all four robots, as shown in Figure. 4.1(b), then intuitively every robot has a unique "view," and hence the robots may be able break symmetry and form a pattern that is not a square. (In fact, the result given below shows that the robots can form *any* pattern for this case.) We now formalize this observation.

Following [19], [20], the *view* of robot r_i at time t, denoted $V_i(t)$, is defined recursively as a rooted infinite tree as follows. See Figure 4.2.

1. The root of $V_i(t)$ has $n-1$ subtrees, one for each robot r_j, $j \neq i$.
2. The edge from the root of $V_i(t)$ to the subtree corresponding to r_j is labeled $((a,b),(c,d))$, where (a,b) is the position of r_j in terms of Z_i and (c,d) is the position of r_i in terms of Z_j.
3. The subtree corresponding to r_j is the view $V_j(t)$ of r_j at time t.

Note that each vertex of $V_i(t)$ corresponds to a robot, but it is not labeled as such. Two views $V_i(t)$ and $V_j(t')$ are said to be *equivalent*, written $V_i(t) \equiv V_j(t')$, if they are isomorphic to each other, including the labels. A view is defined as an infinite tree

for convenience of discussion; the relevant information is contained in the subtree of height 2 from the root.

$V_i(0)$ is thus the view of r_i at time 0. Note that since the robots occupy distinct positions at time 0, the edges incident on the root of $V_i(0)$ have distinct labels. Since at time 0 the robots have no knowledge of other robots' local coordinate systems, at time 0 robot r_i does not know its view $V_i(0)$. Using the following algorithm, the robots can obtain sufficient information to construct their views at time 0.

ALGORITHM $\psi_{getview}$—NONOBLIVIOUS.

The robots first broadcast the x-axes of their respective local coordinate systems by moving in the respective positive x directions, return straight to their respective initial positions, broadcast the y-axes of their respective local coordinate systems by moving in the respective positive y directions, and finally return straight to their respective initial positions. Since different robots may start the second broadcast (of their local y-axes) at different time instants, every robot r_i broadcasting its y-axis must continue to move along its y-axis until it observes that every r_j, $j \neq i$ has changed positions at least twice along a line perpendicular to the first line that r_j broadcasted.

At this moment every robot r_i has discovered the initial distribution $P(0)$ (in terms of Z_i) as well as the direction of the positive x-axis of Z_j for every robot r_j. Then r_i measures the minimum distance d_i between any two robots in $P(0)$ in terms of Z_i and "announces" the value of d_i to all other robots by broadcasting the directed line through its initial position with direction $f(d_i)$ of Z_i, where for $x > 0$, $f(x) = (1 - 1/2^x) \times 360^\circ$ is a monotonically increasing function with range $(0^\circ, 360^\circ)$. Then, any robot observing the movement of r_i can determine the value of d_i (and hence the unit distance of Z_i) from its knowledge on the positive x direction of Z_i and direction $f(d_i)$ of Z_i. Finally, the robots return to their respective initial positions. □

When $\psi_{getview}$ is completed, each robot r_i can determine the positions of all other robots in terms of Z_j for any j. Using this information, r_i can construct its view $V_i(0)$.

Let m be the size of a largest subset of robots having an equivalent view at time 0. If $m = 1$, then every robot has a unique view, and thus once Algorithm $\psi_{getview}$ is executed the robots can be ordered using a suitable total ordering of the views. Then for any multiset F of n points, using a predetermined total ordering of the points in F, the ith robot in the ordering can compute the location of the ith point in F relative to some reference points (e.g., the positions of the first and second robots at time 0 if the first and second points of F are distinct) and move to that point. Therefore, if $m = 1$, the robots can form a pattern similar to F for arbitrary F.

Therefore, in the following, we consider the case $m \geq 2$. Lemmas 4.1, 4.2, 4.3, and 4.4 refer to a fixed initial configuration with $m \geq 2$.

LEMMA 4.1. *The robots can be partitioned into n/m groups of m robots each, such that two robots have an equivalent view iff they belong to the same group.*

Proof. The claim is trivial if $m = n$. Thus assume that $m < n$, and without loss of generality suppose that $V_1(0) \equiv V_2(0) \equiv \cdots \equiv V_m(0)$ but $V_1(0) \not\equiv V_{m+1}(0)$. That is, $r_1, r_2, \ldots, r_m$ have an equivalent view at time 0 but r_{m+1} does not. Let $((a, b), (c, d))$ be the label of the edge from the root of $V_1(0)$ to the vertex corresponding to r_{m+1}. Since $V_1(0) \equiv V_2(0) \equiv \cdots \equiv V_m(0)$ for each ℓ, $1 \leq \ell \leq m$, there exists an edge with label $((a, b), (c, d))$ from the root of $V_\ell(0)$ to a vertex corresponding to some robot r_{i_ℓ}, where $r_{i_1} = r_{m+1}$. Now we show that the robots $r_{i_1}, r_{i_2}, \ldots, r_{i_m}$ are all distinct. Note that by symmetry there is an edge with label $((c, d), (a, b))$ from the

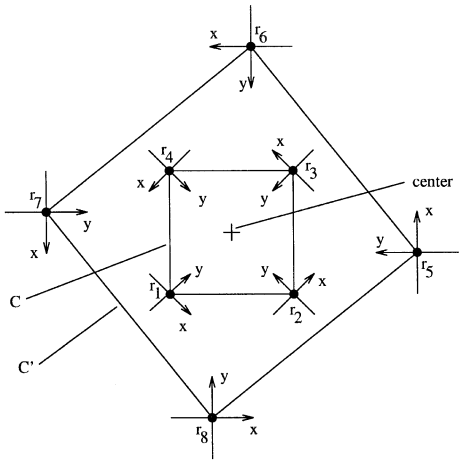

FIG. 4.3. *Illustration for the proof of Lemma* 4.2*, for the case* $m = 4$.

root of $V_{i_\ell}(0)$, leading to a vertex that corresponds to robot r_ℓ. Thus if $r_{i_1} = r_{i_2}$, for instance, then we have $r_1 = r_2$, a contradiction. Thus $r_{i_1}, r_{i_2}, \ldots, r_{i_m}$ are all distinct. Furthermore, since $V_1(0) \equiv V_2(0) \equiv \cdots \equiv V_m(0)$ and $V_{i_\ell}(0)$ is a subtree of $V_\ell(0)$ connected to the root of $V_\ell(0)$ by an edge with label $((a,b),(c,d))$ for each ℓ, we have $V_{i_1}(0) \equiv V_{i_2}(0) \equiv \cdots \equiv V_{i_m}(0)$. Thus there are at least m robots (including r_{m+1}) having a view equivalent to that of r_{m+1}. But then there must be exactly m such robots, since there cannot exist more than m such robots by the definition of m. The lemma follows from this observation. □

LEMMA 4.2. *At time* 0*, the robots in the same group form a regular* m*-gon, and the regular* m*-gons formed by all the groups have a common center.*[7] *(See Figure* 4.3*.)*

Proof. Suppose that $V_1(0) \equiv V_2(0) \equiv \cdots \equiv V_m(0)$, that is, $r_1, r_2, \ldots, r_m$ have an equivalent view at time 0. Consider the initial positions $p_1(0), p_2(0), \ldots, p_m(0)$ of these robots. Clearly, at least one of $p_1(0), p_2(0), \ldots, p_m(0)$ is a corner of the convex hull C of $\{p_1(0), p_2(0), \ldots, p_m(0)\}$. Then, since $V_1(0) \equiv V_2(0) \equiv \cdots \equiv V_m(0)$, each of $p_1(0), p_2(0), \ldots, p_m(0)$ must be a corner of C. Without loss of generality, assume that $p_1(0), p_2(0), \ldots, p_m(0)$ occur in counterclockwise order around the convex hull. (See Figure 4.3.)

Since $V_1(0) \equiv V_2(0) \equiv \cdots \equiv V_m(0)$, the internal angles of C at the corners $p_1(0), p_2(0), \ldots, p_m(0)$ must all be identical, and the lengths of the edges of the convex hull must all be identical. (If $\overline{p_1(0)p_2(0)}$ looks shorter than $\overline{p_2(0)p_3(0)}$ to r_2, then $\overline{p_2(0)p_3(0)}$ should look shorter than $\overline{p_3(0)p_4(0)}$ to r_3, and so on, leading to a conclusion that $\overline{p_1(0)p_2(0)}$ is shorter than $\overline{p_1(0)p_2(0)}$, a contradiction.) Thus $p_1(0), p_2(0), \ldots, p_m(0)$ form a regular m-gon.

Suppose that at time 0, $r_{m+1}, r_{m+2}, \ldots, r_{2m}$ also have an equivalent view and that their respective positions $p_{m+1}(0), p_{m+2}(0), \ldots, p_{2m}(0)$ appear in counterclockwise order around the regular m-gon they form. Then again, since $V_1(0) \equiv V_2(0) \equiv \cdots \equiv V_m(0)$, the position of $p_{m+1}(0)$ relative to p_1 is the same as the position of $p_{m+2}(0)$ relative to p_2, and so on. (See Figure 4.3.) So the regular m-gon formed by $p_1(0), p_2(0), \ldots, p_m(0)$ and the regular m-gon formed by $p_{m+1}(0), p_{m+2}(0), \ldots, p_{2m}(0)$ have the same center. □

[7] A regular 2-gon is simply a line segment whose center is the midpoint of the endpoints.

LEMMA 4.3. *For any algorithm ψ, if the robots are synchronized, then at any time instant t, the robots in the same group form a regular m-gon and the regular m-gons formed by all the groups have a common center.*

Proof. Suppose that $V_1(0) \equiv V_2(0) \equiv \cdots \equiv V_m(0)$, that is, $r_1, r_2, \ldots, r_m$ have an equivalent view at time 0. Now, since the initial distribution of the robots looks identical to $r_1, r_2, \ldots, r_m$, the new positions they compute using ψ in their respective $Z_1, Z_2, \ldots, Z_m$ are all identical. Also, since $V_1(0) \equiv V_2(0) \equiv \cdots \equiv V_m(0)$, the center of the regular m-gon that $r_1, r_2, \ldots, r_m$ form at time 0 has the same x-y coordinates in all of $Z_1, Z_2, \ldots, Z_m$. This means that $r_1, r_2, \ldots, r_m$ move in a symmetric manner relative to the center of the regular m-gon, and thus at time 1 they again form a regular m-gon with the same center. The same applies to all n/m groups, and since the robots are synchronized, at time 1 they together form a collection of n/m regular m-gons all having the same center. Since the robots in the same group have observed the same robot distributions, their next moves at time 1 are also symmetric relative to the center of the regular m-gon they currently form. Therefore, again, at time 2 the robots form a collection of n/m regular m-gons all having the same center. Continuing in the same manner, we can prove that at any time instant t the robots form a collection of n/m regular m-gons all having the same center. □

Since the robots may happen to be synchronized, by Lemma 4.3 there exists an algorithm ψ for forming a pattern similar to F starting from the given initial configuration only if F can be partitioned into n/m regular m-gons all having the same center. Conversely, we have the next lemma.

LEMMA 4.4. *For any multiset F of points that can be partitioned into n/m regular m-gons all having the same center, there exists an algorithm ψ for forming a pattern similar to F starting from the initial configuration. (The algorithm does not depend on the initial configuration.)*

Proof. We fix a total ordering over views and we fix an ordering of the n/m regular m-gons in F. The idea is to move the robots in the jth group in the ordering of the views to the corners of the jth regular m-gon, as in the case $m = 1$. Specifically, first the robots execute Algorithm $\psi_{getview}$ and obtain their views. The robots in the first group need not move any more, since the m-gon they form is similar to the corners of the first m-gon of F (except when the first m-gon is a point, in which case the robots must move to the center of the m-gon they form). Each robot in the second group computes the position of a corner of the second m-gon of F (relative to the location of the first m-gon of F) that is closest to its current position, breaking ties in any deterministic manner. (If the first m-gon is a point and the second m-gon is not, then the robots in the second group need not move.) The robots in other groups also compute their final positions in a similar manner. Then the robots move to their respective final positions and form a pattern similar to F. □

The following theorem summarizes the discussion given above.

THEOREM 4.5. *Let m be the size of a largest subset of robots having an equivalent view at time 0. Let F be a multiset of n points. There exists an algorithm ψ for forming a pattern similar to F, starting from the given initial configuration iff either (1) $m = 1$ or (2) $m \geq 2$ and F can be partitioned into n/m regular m-gons all having the same center.*

Proof. The theorem follows from Lemmas 4.3 and 4.4. □

We introduced in section 3 a predicate π_{point} such that $\pi_{point}(p_1, \ldots, p_n) =$ true iff $p_i = p_j$ for any $1 \leq i, j \leq n$. Consider another predicate $\pi_{regular}$, where $\pi_{regular}(p_1, \ldots, p_n) =$ true iff $p_1, \ldots, p_n$ form a regular n-gon. The following theo-

rem, which follows as a corollary to Theorem 4.5, states that these two are the only predicates for which the formation problem is solvable.

THEOREM 4.6. *There exists an algorithm for solving the formation problem for a predicate π iff either $\pi = \pi_{point}$ or $\pi = \pi_{regular}$.*

Proof. The if part for $\pi_{regular}$ follows immediately from Theorem 4.5, and that for π_{point} follows from Theorem 4.5 and the observation that, for any m that divides n, a point can be viewed as a collection of n/m degenerate regular m-gons all having the same center. The only-if part follows from the fact that if $m = n$, where m is the size of a largest subset of robots having an equivalent view at time 0, then by Theorem 4.5 an algorithm exists for the formation problem only for a single regular n-gon (which reduces to a point if the polygon is degenerate). □

5. Concluding remarks. We formally modeled the system of anonymous mobile robots and characterized the class of geometric patterns that the robots can form. In this section, we discuss other related issues.

5.1. Agreement on a common *x*-*y* coordinate system. In section 1 we briefly mentioned that the agreement problem on a common x-y coordinate system is reducible to the formation problem of certain geometric patterns. By Theorem 4.6 it is always possible for the robots to form a point and a regular n-gon, hence the robots can always agree on both the origin and unit distance (of a common x-y coordinate system). On the one hand, the agreement problem on direction is unsolvable in general, since otherwise the formation problem of a line segment would be solvable, contradicting Theorem 4.6. On the other hand, it can be shown that if the robots have a sense of direction (i.e., their local coordinate systems agree on the positive x direction), then they have distinct views at time 0 (i.e., $m = 1$ where m is as defined in section 4). As we have shown, in this case the robots can form (a pattern similar to) any geometric pattern. This means that the difficulty of forming certain geometric patterns lies in the difficulty of agreeing on direction (and break symmetry).

5.2. Issues of fault tolerance. As we mentioned in section 1, Debest [2] discussed the problem of forming a circle from the viewpoint of self-stabilizing systems. Algorithms for controlling robots must be sufficiently robust against sensor and control errors. Oblivious algorithms are, by definition, self-stabilizing in the sense that they achieve their goal even in the presence of a finite number of sensor and control errors. In contrast, nonoblivious algorithms are sensitive to errors in general, and it is a challenging open problem to enhance fault tolerance in such algorithms.

Another interesting issue in fault tolerance arises when the number of robots changes dynamically a finite number of times during the execution of an algorithm, where by this we mean that a robot becomes visible (or invisible) when it is added to (or removed from) the system. Again by definition, an oblivious algorithm correctly solves the given problem even if the number of robots changes a finite number of times. One way to make nonoblivious algorithms robust against such changes is to adopt an additional assumption that, if the number of robots changes, then it never changes again until all robots have noticed the change. Under this assumption, it can be shown that any nonoblivious algorithm works correctly when it is modified so that a robot noticing a change in the number of robots "resets its memory and restarts the algorithm" (i.e., it ignores the pairs (Q_ℓ, p_ℓ) for the observations made previously).

5.3. Time complexity. Since a robot may remain inactive for an unpredictable period of time, we cannot use the total number of steps for measuring the time complexity of a formation algorithm. An alternative measure of the complexity of

an algorithm is the total distance that a robot must move to form a given pattern. Under this measure, a robot moves over distance $O(d)$ by the method used in the proof of Lemma 4.4, where d is the diameter of the smallest enclosing circle of the initial positions of the robots. (Note that the total distance that a robot moves while executing $\psi_{getview}$ can be limited to $O(1)$.) The bound of $O(d)$ is tight for some patterns (e.g., a point), since a robot can move at most a constant distance at a time.

5.4. Other open problems. Algorithms for solving a formation problem based on the method given in the proof of Lemma 4.4 are nonoblivious. Thus Theorem 4.6 implies that a point and a regular n-gon can be formed by n robots regardless of the initial distribution $P(0)$ and the activation schedule $\mathcal{A}$, by a *nonoblivious* algorithm. An interesting question is whether these patterns can also be formed, regardless of $P(0)$ and $\mathcal{A}$, by an *oblivious* algorithm. For the case of a point we already have the answer: an oblivious algorithm for forming a point exists for the case $n \geq 3$ (Theorem 3.4), but not for the case $n = 2$ (Theorem 3.1). However, the question remains open for the formation of a regular n-gon. We are currently working on this issue and also are conducting similar investigations on (1) randomized algorithms, (2) the case in which the motion of a robot is not instantaneous, and (3) the three-dimensional case.

REFERENCES

[1] G. Beni, S. Hackwood, and X. Liu, *High-order strictly local swarms*, in Distributed Autonomous Robotic Systems, Asama, Fukuda, Arai, and Endo, eds., Springer-Verlag, New York, 1994, pp. 267–278.

[2] X. A. Debest, *Remark about self-stabilizing systems*, Communications of the ACM, 38 (1995), pp. 115–117.

[3] B. R. Donald, *Information invariants in robotics: Part* I*—State, communication, and side-effects*, in Proceedings of the IEEE International Conference on Robotics and Automation, Atlanta, GA, 1993, pp. 276–283.

[4] B. R. Donald, *Information invariants in robotics: Part* II*—Sensors and computation*, in Proceedings of the IEEE International Conference on Robotics and Automation, Atlanta, GA, 1993, pp. 284–290.

[5] K. Fujimura, *Model of reactive planning for multiple mobile agents*, in Proceedings of the IEEE International Conference on Robotics and Automation, Sacramento, CA, 1991, pp. 1503–1509.

[6] T. Fukuda and S. Nakagawa, *Approach to the dynamically reconfigurable robot systems*, Journal of Intelligent and Robotics Systems, 1 (1988), pp. 55–72.

[7] Y. Kawauchi, M. Inaba, and T. Fukuda, *A principle of decision making of cellular robotic system (CEBOT)*, in Proceedings of the IEEE International Conference on Robotics and Automation, Los Alamitos, CA, 1993, pp. 833–838.

[8] S. Kokaji, *A fractal mechanism and a decentralized control method*, in Proceedings of the USA-Japan Symposium on Flexible Automation, Minneapolis, MN, 1988, pp. 1129–1134.

[9] M. J. Mataric, *Designing emergent behaviors: From local interactions to collective intelligence*, in From Animals to Animates 2: Proceedings of the Second International Conference on Simulation of Adaptive Behavior, Meyer, Roitblat, and Wilson, eds., MIT Press, Cambridge, MA, 1993, pp. 432–441.

[10] S. Murata, H. Kurokawa and S. Kokaji, *Self-assembling machine*, in Proceedings of the IEEE International Conference on Robotics and Automation, San Diego, CA, 1994, pp. 441–448.

[11] L. E. Parker, *Designing control laws for cooperative agent teams*, in Proceedings of the IEEE International Conference on Robotics and Automation, Los Alamitos, CA, 1993, pp. 582–587.

[12] M. Schneider, *Self–stabilization*, ACM Computing Surveys, 25 (1993), pp. 45–67.

[13] K. Sugihara and I. Suzuki, *Distributed motion coordination of multiple mobile robots*, in Proceedings of the 5th IEEE International Symposium on Intelligent Control, Philadelphia, PA, 1990, pp. 138–143.

[14] K. Sugihara and I. Suzuki, *Distributed algorithms for formation of geometric patterns with many mobile robots*, Journal of Robotic Systems, 13 (1996), pp. 127–139.

[15] I. Suzuki and M. Yamashita, *Formation and agreement problems for anonymous mobile robots*, in Proceedings of the 31st Annual Allerton Conference on Communication, Control, and Computing, University of Illinois, Urbana, IL, 1993, pp. 93–102.

[16] O. Tanaka, *Forming a Circle by Distributed Anonymous Mobile Robots*, Bachelor thesis, Department of Electrical Engineering, Hiroshima University, Hiroshima, Japan, 1992.

[17] J. Wang and G. Beni, *Cellular robotic systems: Self-organizing robots and kinetic pattern generation*, in Proceedings of the 1988 IEEE International Workshop on Intelligent Robots and Systems, Tokyo, Japan, 1988, pp. 139–144.

[18] L. L. Whitcomb, D. E. Koditschek, and J. B. D. Cabrera, *Toward the automatic control of robot assembly tasks via potential functions: The case of 2-D sphere assemblies*, in Proceedings of the IEEE International Conference on Robotics and Automation, Nice, France, 1992, pp. 2186–2191.

[19] M. Yamashita and T. Kameda, *Computing on anonymous networks Part* I*: Characterizing the solvable cases*, IEEE Transactions on Parallel and Distributed Systems, 7 (1996), pp. 69–89.

[20] M. Yamashita and T. Kameda, *Computing on anonymous networks Part* II*: Decision and membership problems*, IEEE Transactions on Parallel and Distributed Systems, 7 (1996), pp. 90–96.