LNCS 3172

Marco Dorigo Mauro Birattari
Christian Blum Luca M. Gambardella
Francesco Mondada Thomas Stützle (Eds.)

Ant Colony Optimization and Swarm Intelligence

4th International Workshop, ANTS 2004
Brussels, Belgium, September 2004
Proceedings

Springer

Gathering Multiple Robotic A(ge)nts with Limited Sensing Capabilities

Noam Gordon, Israel A. Wagner, and Alfred M. Bruckstein

Center for Intelligent Systems, CS Department
Technion – Israel Institute of Technology, 32000 Haifa, Israel
{ngordon,wagner,freddy}@cs.techniohn.ac.il

Abstract. We consider a swarm of simple ant-robots (or *a(ge)nts*) on the plane, which are anonymous, homogeneous, memoryless and lack communication capabilities. Their sensors are range-limited and they are unable to measure distances. Rather, they can only acquire the directions to their neighbors. We propose a simple algorithm, which makes them gather in a small region or a point. We discuss three variants of the problem: A continuous-space discrete-time problem, a continuous-time limit of that problem, and a discrete-space discrete-time analog. Using both analysis and simulations, we show that, interestingly, the system's global behavior in the continuous-time limit is fundamentally different from that of the discrete-time case, due to hidden "Zenoness" in it.

1 Introduction

The problem of gathering a swarm of robots in a small region or a point on the plane is a fundamental one. From a practical standpoint, it may be useful for collecting multiple ant robots after they have performed a task in the field, for enabling them to start a mission, after being initially dispersed (e.g., parachuted), or even for aggregating many nano-robots in a self-assembly task. From a theoretical standpoint, the gathering problem is linked to *agreement* problems (as it may imply or be implied by agreement on a reference location) and is the most basic instance of the *formation* problem, i.e., the problem of arranging multiple robots in a certain spatial configuration. The problem is most challenging when the robots are ant-like – having very limited abilities, e.g. myopic, disoriented and lacking explicit communication capabilities.

Several theoretical works on this subject exist. Suzuki and Yamashita suggested an agreement procedure, where the agents communicate data through their movements, in order to agree on a meeting point [16]. Schlude suggested the use of a *contraction point*, which is invariant to their moves toward it [14]. Ando et al. suggested an algorithm for myopic robots, which move according to the exact locations of nearby robots [1]. Lin et al. also provided an algorithm for mypoic robots and related to the case of a limited field of view [11]. Prencipe et al. suggested an algorithm for myopic robots, which relies on a common compass [7], as well as an algorithm which creates a unique point of multiplicity (i.e., which contains several agents) to which all agents move [6]. Gordon et al.

M. Dorigo et al. (Eds.): ANTS 2004, LNCS 3172, pp. 142–153, 2004.
© Springer-Verlag Berlin Heidelberg 2004

suggested a simple gathering algorithm on the grid [8]. Others explored cyclic pursuit behaviors (where robots are cyclicly ordered and pursue each other accordingly), which, in some cases, lead to gathering [2, 3, 12]. Sugihara et al. suggested a simple behavior which makes robots fill any convex shape and evenly distribute inside it [15]. Also related is the work of Melhuish et al. [13], who considered aggregation of robots around a beacon in a noisy environment, and demonstrated a way to control the swarm size using minimal communication.

These works rely on some strong assumption about the robots (or *agents* as we shall call them henceforth): Some rely on labelling (e.g., pursuit), some on common orientation, and many on infinite range visibility. Furthermore, all works rely on the agents' ability to measure their mutual distances (except for a few pursuit strategies).

In this work, we suggest a simple gathering algorithm, which relies on very few capabilities: Our agents are both anonymous, homogenous, memoryless, asynchronous, myopic and are *incapable of measuring mutual distances*. It is similar in idea to the polygon-filling algorithm of Sugihara et al. [15]. However, they did not consider visibility limitations at all, so their algorithm could not be used as is, under our imposed limitations.

The inspiration and motivation for this work comes from experiments with real robots in our lab [5], made from LEGO parts and very simple sensors, which are range-limited and do not provide usable distance measurements.

We consider three flavors of the problem, differing mostly in the way time and space are modelled (continuous vs. discrete). Using simulations and analysis, we discuss some interesting implications of these differences on the resulting swarm behaviors. Due to limited space considerations, we have omitted or abridged most of our proofs. These will appear in a forthcoming extended paper.

2 An Asynchronous Gathering Algorithm on the Plane

In this section we present the continuous-space discrete-time case. We begin with the model of the world and its inhabitants.

The *world* consists of the infinite plane $\mathbb{R}^2$ and n point *agents* living in it. We adapt Suzuki and Yamashita's convenient way of modelling a system of asynchronous agents [16]: *Time* is a discrete series of *time steps* $t = 0, 1, \ldots$. At each time step, each agent may be either *awake* or *asleep*, having no control over the scheduling of its waking times. A sleeping agent does nothing and sees nothing, i.e., it is unaware of the world's state. When an agent wakes up, it is able to move instantly to a point on the plane within a distance σ (the *maximum step length*) according to its algorithm. The agent is able to see only the agents within distance V (the *visibility radius* or *range*). However, it cannot measure its *distance* from them. It only knows the *directions* in which the nearby agents are found, i.e., the input is a cyclic list of *angles* $\theta_1, \ldots \theta_m$ (relative to some arbitrary direction, e.g., the agent's heading). There are no collisions. Several agents may occupy the same point[1]. All agents are *memoryless*, *anonymous* (they cannot be

[1] In this case, they have undefined relative directions and are simply mutually ignored.

distinguished by their appearance) and *homogenous* (they lack any individuality, such as a name or ID, and perform the same algorithm).

Regarding the waking times of the agents, we make the following assumption. We say that the agents are *strongly asynchronous*: For any subset G of the agents and in each time step, the probability that G will be the set of waking agents is bounded from below by some constant $\varepsilon > 0$. This implies that each agent will always wake up again in finite expected time.

Define the *mutual visibility graph* of the world as an undirected graph with n vertices, representing the agents, and an edge between each pair of agents, if and only if they can see each other, i.e., the distance between them is at most V. Unless noted otherwise, we assume that this graph is initially connected.

2.1 Maintaining Visibility

We now present a sufficient condition on any movement algorithm for maintaining mutual visibility between agents. In what follows, denote a disc of radius r and center a (where a may signify the location of agent a) by $B_r(a)$.

Let a and b be two agents at some distance $d \leq V$ apart. If their next movement is confined to $B_{V/2}\left(\frac{a+b}{2}\right)$, they will remain visible, by definition. However, since the agents *cannot* measure d, they must consider all possible values of $d \in [0, V]$. Therefore, each agent's next move must rely within the intersection of all discs of diameter V, centered at all possible midpoints between a and b. It is easy to show that, for agent a, this is equivalent to $B_{V/2}(a) \cap B_{V/2}(r)$, where r is a point at a distance $V/2$ from a, in b's direction (cf. Fig. 1(a)). More generally, when a sees several other agents $b_1, \ldots b_m$, it is allowed to move to any point within the intersection of $m+1$ discs of diameter V, one is $B_{V/2}(a)$, and the other m discs centered at a distance $V/2$ from a, in the directions of $b_1, \ldots b_m$ (cf. Fig. 1).

Denote by ξ the *largest angle between consecutive agents* in agent a's (cyclic) list of input angles $\theta_1, \ldots \theta_m$. It is straightforward to show that the allowable movement region is empty if and only if $\xi < \pi$, i.e., when the agent is "sur-

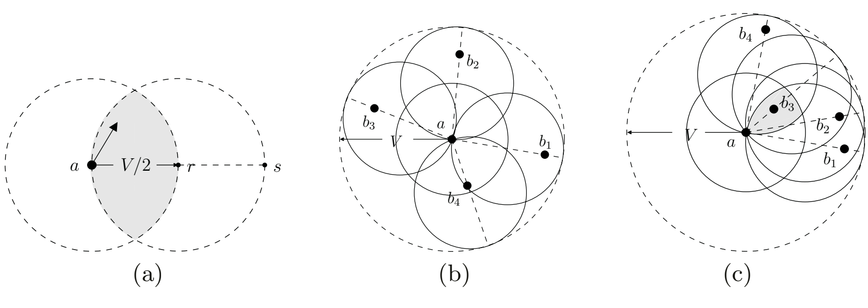


Fig. 1. Maintaining visibility. (a) a must remain within the shaded area to maintain visibility with b, which is somewhere on the line segment $\overline{as}$, as far as a knows. (b) a is surrounded and cannot move. (c) a can move only within the shaded area.

rounded" by other agents (cf. Fig. 1(b)). Otherwise, the allowable region is not empty, and is calculated as follows: Let θ_k and θ_{k+1} be the directions which form the angle $\xi = \theta_{k+1} - \theta_k$. Then the allowable region is the intersection of three discs of diameter V, centered around a and around the two points at a distance $V/2$ in directions θ_k and θ_{k+1} from a, respectively (cf. Fig. 1(c), 2). The following lemma follows from the above geometric arguments.

Lemma 1. *If each agent confines its movements to the allowable region defined above, then existing visibility will be maintained.*

A direct corollary to Lemma 1 is that a connected visibility graph will remain connected forever.

2.2 The Gathering Algorithm

Denote by ψ the complementary angle of ξ defined above (i.e., ψ is the angle of the *smallest wedge* containing all visible agents). The algorithm works as follows: Agents which are surrounded by other agents do not move, while other agents move as far as they can on the *bisector* of ψ. The idea is that outermost agents generally move *inside* the region containing the agents, gradually making it shrink.

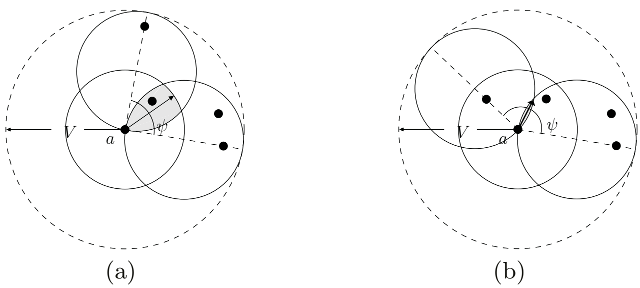


Fig. 2. Agent a's movement (assuming $\sigma > V/2$). (a) $\psi < 2\pi/3$. Travelling distance is $V/2$. (b) $\psi > 2\pi/3$. Travelling distance is $V\cos(\psi/2)$.

The exact movement rule is as follows: *If* $\psi \geq \pi$, *then do not move. Otherwise, move along the bisector of* ψ, *a distance*

$$\mu_\psi = \min\left(V/2, V\cos\left(\psi/2\right), \sigma\right) \quad . \tag{1}$$

σ is the physical constraint, while the first two constraints express the maximum possible travelling distance, easily derived from the definition of the allowed movement region stated above (See Fig. 2). For convenience, we denote the *maximum possible step length* by $\mu = \min\left(V/2, \sigma\right)$.

It is worth noting that the allowable region is symmetrical about the bisector of ψ, and that the farthest point on this region is on the bisector. In other words, the bisector can be seen as the "natural" direction to move along, just for the sake of maintaining visibility. Interestingly, this movement direction was originally chosen by Sugihara et al., even though they were not at all concerned with range-limited visibility or step size.

We defer the discussion of the system's behavior to Sect. 5, after the other variants of the model are presented.

3 The Continuous-Time Limit

In this section we discuss the continuous-time limit behavior of the system described above, as follows. Assume that the agents are synchronous, i.e., active at all times. Denote the physical duration of each time step by $\triangle t$ and let $\sigma = v \triangle t$ for some constant v. It follows from (1) that, for $\sigma < V/2$, $\mu_\psi = \sigma$ for $0 \le \psi < 2\cos^{-1}(\sigma/V)$. Thus, in the limit $\triangle t \to 0$, we get $\mu_\psi = \sigma = v \triangle t$ for all $0 \le \psi < \pi$, and the movement rule becomes: *If $\psi < \pi$, move along the bisector of ψ at a (constant) speed v. Otherwise, do not move.*

Lemma 1 still holds. In fact, it is not hard to see that an even stronger result holds here: For any two mutually visible agents, their distance is non-increasing.

3.1 Collinearity, Varying Speeds and Zenoness

This simple movement rule (either move at a constant speed v or stand), exhibits a seemingly paradoxical behavior: Agents will sometimes move at *varying* speeds! To see this, consider the following scenario. Let a, b and c be three collinear agents and denote their wedge angles by ψ_a, ψ_b and ψ_c, respectively. b is the middle one, and $\psi_b = \pi$ is determined by the locations of a and c. Assume that $\psi_a, \psi_c < \pi$ (so that a and c move) at time $t = 0$. After an *arbitrarily short* time, as a result of their displacement, ψ_b slightly decreases and, therefore, b starts moving as well, in a normal direction to the segment $\overline{ac}$. Collinearity is now seemingly broken. However, within an arbitrarily small time, b necessarily moves farther enough so that it is "ahead" of $\overline{ac}$, and the bisector of ψ_b now flips and points backwards. As a result, b returns backwards until it crosses $\overline{ac}$ again. This process repeats itself over time, while a and c keep moving. Now, since the deviations of b from $\overline{ac}$ are also arbitrarily small, b effectively remains collinear with a and c. In summary, the following claim holds:

Proposition 1. *Let a, b and c be three collinear agents, where b is the middle one. They will remain collinear as long as in ψ_b is determined by a and c.*

When integrated over time, this "chattering" movement back and forth (at a constant speed v) becomes a smooth movement of b, always on $\overline{ac}$, at a speed generally not equal to v. Take, for instance, the symmetric case, where both a and c move at an angle $\pi/4$ relative to $\overline{ac}$. Then b will move in a straight line, normal to $\overline{ac}$, at a speed of $v/\sqrt{2}$.

This seemingly paradoxical behavior stems from the fact that agent b performs an infinite number of discrete direction switches over a finite period of time. This phenomenon is known as *Zenoness* in hybrid systems theory (See, e.g., [9]), and it shows that, in the continuous-time limit, our system is ill-posed and physically unrealizable. However, had we introduced a slight delay in the agents' responses, we would get a consistent behavior where agent b "oscillates" around $\overline{ac}$ at a finite rate. Indeed, we observed such a behavior in simulations of the discrete-time system of Sect. 2 (For a related discussion, cf. [4]).

3.2 Correctness and Termination

We now present a formal proof of correctness and termination in finite time of the algorithm. Denote the *convex hull* of the configuration (i.e., of the positions of all agents) at time t by $CH(t)$, and the number of its (strictly convex) corners by m. Denote the corner agents by a_i ($i = 1, \ldots, m$), and the inner angle at each corner a_i by ϕ_i. Denote the angle formed between a_i's movement direction and one of the adjacent edges of $CH(t)$ by α_i (Clearly, $0 \leq \alpha_i \leq \phi_i$). Denote the length of the edge adjacent to a_i by P_i, and the total perimeter of $CH(t)$ by $P = \sum_{i=1}^{m} P_i$.

Proposition 2. *$\dot{P}$ is negative and bounded away (by a constant) from zero, as long as $P > 0$.*

Proof. (abridged) Since the corner agents do not necessarily see each other, a corner agent may not necessarily move along the bisector of ϕ_i. However, since there are no agents outside $CH(t)$, and according to Lemma 1 the visibility graph is connected, it is guaranteed that each corner agent observes at least one other agent and therefore moves inside $CH(t)$.

Observe a single edge of $CH(t)$, connecting a_i and a_{i+1}. During an arbitrarily short period of time dt, these agents move into $CH(t)$ a distance $v\,dt$, at angles α_i and $\phi_{i+1} - \alpha_{i+1}$ relative to the edge, respectively. Any other agents on the edge, or arriving at the edge, will remain on it, according to Prop. 1. Therefore, they have no effect on it. It can be shown that, as a result, the edge will be shortened by the following amount:

$$P_i(t + dt) - P_i(t) = -v\,dt\,\left(\cos \alpha_i + \cos\left(\phi_{i+1} - \alpha_{i+1}\right)\right) + \mathbf{o}\left(dt\right) \quad . \qquad (2)$$

Summing over all m corners, dividing by dt and letting $dt \to 0$, we get:

$$\begin{aligned} \dot{P} &= -v \sum_{i=1}^{m} \left[\cos \alpha_i + \cos\left(\phi_i - \alpha_i\right)\right] \\ &\leq -v \sum_{i=1}^{m} \left[1 + \cos\left(\phi_i\right)\right] \\ &\leq -v\,\left[1 + \cos\left(\pi\,\left(1 - 2/n\right)\right)\right] \quad . \end{aligned} \qquad (3)$$

The first inequality holds because, for any fixed $\phi_i < \pi$, the expression $\cos \alpha_i + \cos\left(\phi_i - \alpha_i\right)$ is minimal when $\alpha_i = 0$. The second inequality is straightforward, and the resulting expression on the right side is a negative constant, dependent only on n, which is finite. □

Theorem 1. *Beginning with any initial configuration, whose visibility graph is connected, all agents will gather in a single point, in finite time.*

Proof. This is a direct corollary of Prop. 2. □

Another obvious property from the above analysis is that for any two time instants t_1, t_2, if $t_1 < t_2$ then $CH(t_1) \supset CH(t_2)$. A corollary of this is that if the visibility graph is initially *not* connected, none of its connected components will ever merge (or split, of course), and therefore each group of agents, corresponding to a connected component, will gather in exactly one separate point.

4 A Discrete Analog

This section presents a discrete-space discrete-time analog of the problem presented in Sect. 2. The asynchronous operation of the agents is retained, with the world now being the infinite rectangular grid ($\mathbb{Z}^2$).

On the grid, we measure distances with *infinity norms*. Formally, the distance between two points $p_1 = (x_1, y_1)$ and $p_2 = (x_2, y_2)$ on the grid is

$$\|p_1 - p_2\|_\infty = \max\left(x_1 - x_2, y_1 - y_2\right) .$$

Accordingly, let $V \geq 1$ be the visibility range of the agents. Then the visible area of an agent is a $(2V + 1) \times (2V + 1)$ square, centered at the agent. An agent cannot measure the exact distance from a visible agent. Rather, it can only measure the signs (positive, negative or zero) of $x_1 - x_2$ and $y_1 - y_2$. In the grid world, an agent may move only to one of the four neighboring cells.

Define a move as *allowable* if and only if there are *no* visible agents "behind" the agent and there *exists* a visible agent "before" it, as it looks in that direction (e.g., the agent is allowed to move in the *positive* x direction if and only if there is *no* visible agent with a *smaller* x *coordinate* and there *exists* a visible agent with a *larger* x *coordinate*. See Fig. 3).

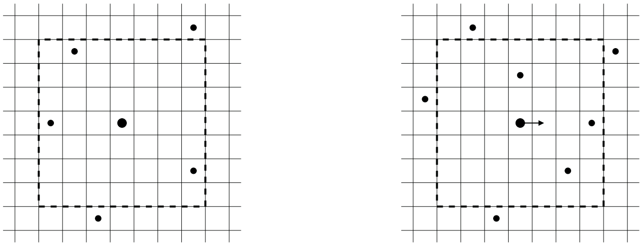
The movement rule is as follows: *The agent randomly picks an allowable move, if there is any, and performs it* (There can be 0 to 2 allowable moves. In case of two allowable moves, the agent tosses a coin). Intuitively, this rule makes agents get closer to, but not move away from, each other. Thus, visibility is maintained, and we expect the area occupied by all agents to shrink. We now present a proof for this.

Define the *bounding box* $BB(t)$ of the agents as the smallest enclosing rectangle (oriented with the grid's axes) which contains all of the agents.

Proposition 3. *When following the movement rule defined above, mutually visible agents remain visible.*

Proposition 4. *The bounding box of the agents is monotonically non-inflating, i.e.,* $BB(t + 1) \subseteq BB(t)$ *for all* t*.*

Proposition 5. *At least one of the bounding box's sides eventually move inwards (as long as* BB *is not a single cell).*



(a) (b)

Fig. 3. The proposed gathering algorithm on the grid ($V = 3$). (a) The center agent is surrounded and cannot move. (b) The center agent can move only to the right.

Proof. (abridged) Observe w.l.o.g. all agents which reside on the *upper side* of the bounding box (i.e., all agents with the maximum y coordinate), and denote their number by $X(t)$. If the bounding box's height is 1 (i.e., $X = n$), then clearly the leftmost and rightmost agents move inwards (when they eventually wake up), narrowing the bounding box. Otherwise, due to the connectedness of the visibility graph (Prop. 3), at least one agent on the upper side observes another agent, which resides *below* the upper side. Therefore, that agent is allowed to move downwards and, from the strong asynchronicity assumption, there is a probability of at least $\varepsilon/2$ that it indeed wakes up and chooses to move in that direction (and no other agent wakes up and moves up), lowering the number of agents on the upper side. Thus, all states $X = k,\ 0 < k < n$ are transitional and connected to either one of the states $X = 0$ and $X = n$, which are obviously trapping (due to Prop. 4). Therefore, eventually either $X = 0$ or $X = n$ will occur, meaning that either the upper side has moved down or the lower side has moved up all the way to merge with the upper side, respectively. □

Theorem 2. *Beginning with any initial configuration, whose visibility graph is connected, all agents will eventually gather in a single cell.*

Proof. The proof follows immediately by applying Proposition 5 repeatedly. □

5 Discussion

We performed extensive simulations of the problem described in Sect. 2. Not only did the simulations validate the correctness of the algorithm, but they also revealed some interesting global behaviors of the swarm. We discuss them qualitatively, and compare them to the continuous-time limit behavior (which apparently cannot be simulated, due to its Zenoness). We argue that, in the limit, the system's behavior is *fundamentally* different.

The system's evolution is clearly divided into two phases (1) A *contraction phase*, where the area occupied by the agents contracts until all agents become a small, dense cluster, whose diameter is in the order of μ (the maximum step

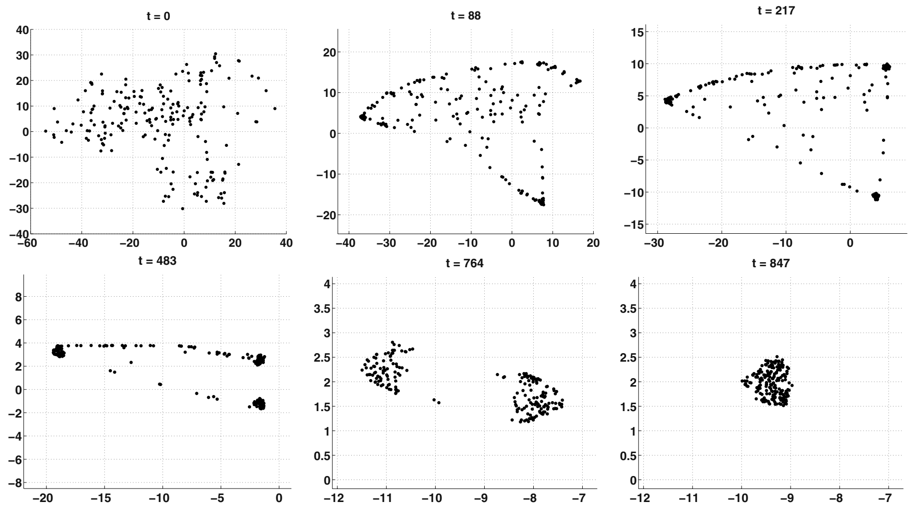


Fig. 4. A typical run of the continuous-space discrete-time algorithm. Here $n = 200, \sigma = 1, V = 10$, and each agent wakes up in each time step with probability $p = 0.6$. Note that the scale changes between frames.

size); (2) A *wandering phase*, where the cluster stops contracting and begins to wander indefinitely in the plane.

5.1 The Contraction Phase

Observe Fig. 4, which shows several snapshots of one particular run of the system, beginning with a random configuration with a connected visibility graph. Consider the area occupied or "guarded" by the agents (informally speaking, the area enclosed by laying a "fence" between each pair of mutually visible agents). It is intuitively expected that the agents on the convex segments of its boundary would move inside, those on the concave segments would either move outside or stand, while the interior agents would generally stay inside[2]. What intrigues us is the evolution of that boundary over time. Evidently, the occupied region shrinks and its boundary contracts. As expected, the moving boundary "sweeps" more and more agents, and becomes a "belt" which accumulates most of the "mass" in the system. Moreover, a peculiar and less obvious phenomenon is evident from the simulations: The build-up of mass on the boundary belt is *not* uniform. Segments with high curvature (i.e., where the boundary's course bends sharply) tend to absorb more agents than (and from) segments with lower curvature. In addition, the former segments' curvature becomes even higher, while the latter segments tend to straighten slightly. This interaction between *mass* and *curvature* along the belt creates a positive feedback process: The large-scale

[2] We say "generally", since it is indeed possible that agents move a short distance outside the occupied region, if they are close enough to the boundary.

shape of the occupied region becomes an approximate polygon, where the curved segments turn into large clusters which form the corners, while the less curved segments become nearly straight edges between the corners. As the polygon contracts, edges gradually merge, until a triangle or a two-cluster "dipole" remains, ultimately collapsing into one dense cluster.

What causes this positive feedback process? An informal explanation for one direction can be illustrated if we consider the following approximated behavior: Replace the boundary with a smooth contour with equally spaced agents on it, and let each agent move a short distance μ along its normal. Clearly, the higher the curvature, the closer adjacent agents will become.

The other direction's possible explanation is easier to observe in one dimension first. Consider the following "leaping frogs" game: Imagine m frogs on the real axis $\mathbb{R}$. Their movement rule is that, at each time step, only the leftmost frogs leap a distance σ to the right. What is the average speed of the pack? Assuming that no frog lands exactly on another frog, then it is clearly σ/m, as only one of the m frogs move at each time step. Back in two dimensions, we notice that the boundary belt is especially *thick* in the denser segments, and the "cross sections" of these segments typically contain more agents. However, only the outermost agents (analogous to the leftmost frogs) move in each step, lowering the average contraction speeds of these segments, which, as a result, "lag behind" and become more corner-like.

If we set $\sigma = v\triangle t$, where $\triangle t$ is the physical duration of each time step, the average speed of the frog pack should be v/m, regardless of the size of $\triangle t$. However, when we take $\triangle t \to 0$, a strange thing happens. The frogs' behavior changes from "leaping over" to "sweeping" – The leftmost frog simply moves at a constant speed v, eventually joined by all frogs. Analogously, in the continuous-time system of Sect. 3, due to Prop. 1, when a moving segment of the boundary meets an internal agent, it sweeps it along, without slowing at all. Thus, in this model, the "mass" of the boundary (which has no thickness in this case) does not affect its contraction speed, and the mass–curvature feedback link is broken. Therefore, we conjecture that the shape evolution of the occupied region's boundary is fundamentally different than that of the discrete-time case – No sharp corners and straight edges will be formed. Rather, convex segments will quickly contract, and the region will have a much rounder and smoother shape.

5.2 The Wandering Phase

Since the agents do not measure distances, their steps' sizes are invariant to the configuration's scale. The smaller the region they occupy, the relatively longer their steps become. Once the diameter of the occupied region is in the order of μ, the moving agents tend to "leap" over the region, rather than enter it. As a result, the region drifts rather than contracts[3]. Due to the random nature of the agents' activity schedules, the drift direction is also random. Thus, the

[3] Formally, we may choose to define the cluster's location as its center of mass, the center of the smallest enclosing circle, etc.

movement of the cluster is a random walk. We call it a *composite random walk*, as it is composed of the deterministic (yet randomly scheduled) movements of many agents. Just as with the boundary evolution case above, here more mass means slower wandering.

Random walks in two dimensions are known to be recurrent, which implies that two random walkers are bound to meet eventually (See, e.g., [10]). This has an important implication on our problem: If the visibility graph is initially *not* connected, then each connected component becomes an independent composite random walker. However, due to their recurrent nature, these clusters will eventually meet and merge. Thus, the proposed algorithm may eventually gather all agents, *even though the visibility graph is not connected initially.* Of course, we do not mean that it will work for any initial condition. The composite random walkers must be able to meet. This may be false for clusters of one agent (which doesn't move at all) or two agents (which always remain on one line). We conjecture that for clusters of three or more non-collinear agents, this is guaranteed, and a sufficient condition for the eventual success of the algorithm would be the existence of such a cluster. It should be noted that, as our simulations show, the merging process is generally agonizingly slow. Still, from a theoretical standpoint, merging should occur eventually.

Yet again, we see a significant difference between this behavior and the continuous-time limit behavior, where, as we showed in Sect. 3, gathering in a single point occurs, rather than wandering, and there is no hope of ever merging two connected components of the visibility graph.

6 Conclusion

The algorithm proposed in this paper is an example of how very simple individual behaviors can yield complex global behaviors of the swarm. In each of the three variants we presented, we showed how different global processes lead to the ultimate goal of gathering. In the continuous-space discrete-time model, the global shape of the swarm becomes an approximate polygon and converges to a wandering cluster. In the continuous-time limit, the swarm shape is much smoother, and the agents converge to a static point. Our main contribution is that we consider the gathering problem with such severe limits on the sensory abilities of the agents (being both myopic and unable to measure distances), in addition to being anonymous and memoryless.

In forthcoming papers, we shall provide an analytic proof for the convergence of the first variant of the algorithm, and a more rigorous analysis of the swarm's shape evolution. We will also explore the application of the proposed algorithm to row-straightening and formation of other convex shapes. Further work should relate to the effect of noise and error, both in sensing and movement, on the resulting global behavior. Clearly, the proposed algorithm is sensitive to such errors, as the connectivity of the visibility graph may be broken. It would be interesting to analyze the effects, and devise methods to overcome them.

References

1. H. Ando, Y. Oasa, I. Suzuki, and M. Yamashita. A distributed memoryless point convergence algorithm for mobile robots with limited visibility. *IEEE Trans. on Robotics and Automation*, 15(5):818–828, 1999.
2. A. M. Bruckstein, N. Cohen, and A. Efrat. Ants, crickets and frogs in cyclic pursuit. Technical Report CIS-9105, Technion – IIT, 1991.
3. A. M. Bruckstein, C. L. Mallows, and I. A. Wagner. Probabilistic pursuits on the grid. *American Mathematical Monthly*, 104(4):323–343, April 1997.
4. A. M. Bruckstein and O. Zeitouni. A puzzling feedback quantizer. Technical Report 879, EE Department, Technion IIT, May 1993.
5. The Center of Intelligent Systems, Technion IIT web site: *http://www.cs.technion.ac.il/Labs/Isl/index.html*.
6. M. Cieliebak, P. Flocchini, G. Prencipe, and N. Santoro. Solving the robots gathering problem. In *Proc. of ICALP 2003*, 2003.
7. P. Flocchini, G. Prencipe, N. Santoro, and P. Widmayer. Gathering of autonomous mobile robots with limited visibility. In *Proc. of STACS 2001*, 2001.
8. N. Gordon, I. A. Wagner, and A. M. Bruckstein. Discrete bee dance algorithms for pattern formation on a grid. In *Proc. of IEEE Intl. Conf. on Intelligent Agent Technology (IAT03)*, pages 545–549, October 2003.
9. T. A. Henzinger. The theory of hybrid automata. In *Proc. of LICS*, pages 278–292, 1996.
10. B. D. Hughes. *Random Walks and Random Environments*, volume 1. Oxford University Press, 1995.
11. Z. Lin, M. E. Broucke, and B. A. Francis. Local control strategies for groups of mobile autonomous agents. *IEEE Trans. on Automatic Control*, 49(4):622–629, April 2004.
12. J. A. Marshall, M. E. Broucke, and B. A. Francis. A pursuit strategy for wheeled-vehicle formations. In *Proc. of CDC03*, pages 2555–2560, 2003.
13. C. R. Melhuish, O. Holland, and S. Hoddell. Convoying: using chorusing to form travelling groups of minimal agents. *Robotics and Autonomous Systems*, 28:207–216, August 1999.
14. K. Schlude. From robotics to facility location: Contraction functions, weber point, convex core. Technical Report 403, CS, ETHZ, 2003.
15. K. Sugihara and I. Suzuki. Distributed algorithms for formation of geometric patterns with many mobile robots. *Journal of Robotic Systems*, 13(3):127–139, 1996.
16. I. Suzuki and M. Yamashita. Distributed anonymous mobile robots: Formation of geometric patterns. *SIAM Journal on Computing*, 28(4):1347–1363, 1999.