Flocking: don't need no stinkin' robot recognition

Benjamin T. Fine and Dylan A. Shell
Department of Computer Science and Engineering
Texas A&M University

***Abstract*— Flocking is a common and widely studied spatial behavior exhibited by groups. This work highlights inconsistencies in the presentation of motion rules used for flocking, by implementing the well known rule by Hamilton on a robot system. We address a common assumption regarding the form of input to the motion rule: detection of whole agents and suggest that such detection is not necessarily justified. Our multi-robot system successfully exhibits flocking behaviors using an alternative detection method based entirely on low level sensor data. Furthermore, we show (under certain parameter settings), the behaviors exhibited by the agent-based and sensor-based detection are equivalent. We also discuss the various dynamics and implications the chosen detection process has on the behaviors of a motion rule.**

INTRODUCTION

One of the most common collective spatial behaviors displayed by groups is aggregate motion (*e.g.*, flocking). This phenomena is seen from bacteria colonies to large migratory animals and has been studied for 60 years in disciplines such as biology, physics, and robotics. Flocking, in contradistinction to highly-coupled formation control, involves cohesive collective movement without fixed pairwise associations between agents. Still today, questions regarding the mechanisms by which these aggregates are formed and why they persist remain far from entirely resolved. This paper illustrates the delicate process of translating a prose description into an algorithmic motion rule, and explores the effects different forms of input have on the system's behavior.

A robotic implementation affords us the ability to verify the biological feasibility of a motion rule and uncover subtle inconsistencies and unrealistic assumptions for a biological agent. The chosen motion rule for this work was first proposed by Hamilton in 1971 [1] and has been the focus of many works in the biology community. Reynold's [2] motion rule is recognized and highly cited in robotics literature, however, Hamilton's work predates the work by 16 years and is well known across several disciplines. Through our treatment of Hamilton's motion rule, we show there are important details which are not considered in the majority of the literature.

For this work, perception is the process of taking raw sensor data and converting it into usable input for the motion rule. The motion rule governs the agent's behavior and, therefore, understanding the preceding perceptual processing is vital for understanding the resultant behavior. The majority of the flocking literature neither considers nor reports the process of converting the raw sensor data into usable input. As a consequence, several models employ raw sensor data which is unavailable or unrealistic for a robot, oftentimes ignoring uncertainty from noise or occlusion.

This work directly focuses on the sensory processing: three different perception functions are introduced and their implications for the underlying motion rules discussed. The three functions are (1) agent recognition, (2) agent-extrapolated and (3) sensor-based. Our data support the following relationships:

- With perfect sensing the behaviors produced by the three functions are equivalent.
- Agent-extrapolated is similar to agent-based detection due to the constraints on the sensor data.
- Sensor-based input is more sensitive to false positives (FP) where agent-based and agent-extrapolated are sensitive to false negatives (FN). [1]

In addition, we hypothesize that any motion rule's behavior is essentially independent of the perception function used under certain parameter constraints.

HAMILTON'S THEORY REVISITED

The Selfish Herd Hypothesis

Many theories (Hamilton [1], Barbosa [3], Buhl et al. [4], Simons [5]) attempt to explain the causes and mechanisms of flocking, but no single theory has had complete success. The most commonly discussed theory is Hamilton's *Selfish Herd Hypothesis* [1]. From field observations, Hamilton suggests flocking behaviors emerge due to a selfish need for survival, and that individual agents do not have a *sense* for the whole collective, but only a *sense* for agents within a certain radius.

Hamilton [1] presents the Hamilton (HA) and the Simple Nearest Neighbor (SNN) motion rules. In the HA motion rule, an agent selects the nearest agent (*nearest neighbor*) and then selects the nearest agent to

[1] A FN is when the sensing agent fails to detect an agent and a FP is when the sensing agents detects an agent which does not exists.

the *nearest neighbor*.[2] The agent will then select a way-point which is two body-lengths (*repulsion distance*) away from the *nearest neighbor*, which is on the line between the two selected agents. If the *nearest neighbor* is within the repulsion distance of the sensing agent, the way-point will be perpendicular to the line between the two selected agents.[3] In the SNN rule, the agent will move directly towards the *nearest neighbor*. If the *nearest neighbor* is within the *repulsion distance*, the sensing agent will not move.

Algorithm 1 Hamilton's original motion rule

Input: List of inputs (I) in a robot-centric coordinate frame, where $|I| > 1$.
Parameters: $R :=$ *repulsion distance*
Output: Way-point in robot-centric coordinate frame.

$a_1 \leftarrow \min(\forall i \in I, dist(i, (0, 0)))$
$a_2 \leftarrow \min(\forall i \in I, dist(i, a_1)$
if dist$(a_1, (0, 0)) > R$ **then**
    return ComputeHAWayPoint(a_1, a_2)
return ComputePerpendicularWayPoint(a_1, a_2)

Hamilton supported these motion rules by showing, through computer simulation, the agents formed densely packed clusters of agents from random starting formations and discussed how these motion rules are biologically feasible. In his simulation, the agents are homogeneous holonomic agents starting in random positions within the environment. At each simulation step, all agents would synchronously compute their next way-point and move one body-length in that direction. It must be noted that the HA rule was validated in one dimension, while SNN was validated in two dimensions.

Unfortunately, Hamilton's proposed motion rule fails to exhibit the same behaviors the motion rule was originally motivated by. In the observed biological agents [1], the group members formed one large centrally compact group and in Hamilton's simulations the agents formed many small compact groups. From the simulations, Hamilton suggested there must be a group level aggregation behavior not covered by the proposed rule. Hamilton proposed the smaller groups would then move toward the nearest group, producing a single compact group.

Extending HA

Viscido *et al.* [1] is a prime example of reexamination of the HA motion rule within the flocking literature. It illustrates how assumptions which are infeasible or not practicable for operation on robots often indicate issues that arise in any agents operating in the real world. Even though robots have different constraints to biological systems, if the motion rules have assumptions which hinder the implementation on a robot, then the argument for biological plausibility is weakened.
For example:

Synchrony: Huth and Wissel [6], Aoki [7], Viscido [8].

Non-local Sensing: Vicsek et al. [9], Levine et al. [10], Viscido et al. [8].

Perfect Sensing: Tanner et al. [11], Viscido et al. [8] [12].

The original description of the HA rule assumes the sensing agent will always have at least two agents to calculate the next way-point. In a physical system, we cannot guarantee this will hold (*e.g.*, due to occlusions created by the environment and other agents). To account for this, we chose to compute the SNN way-point when only one other agent is detected. To justify this additional behavior we must discuss our interpretation of Hamilton's work [1]. Lines 1 and 4 in Algorithm 2 show the modifications to Algorithm 1.

Algorithm 2 Extended Hamiltonian motion rule

Input: List of inputs (I) in a robot-centric coordinate frame.
Parameters: $R :=$ *repulsion distance*
$ED :=$ *exclusion distance*
Output: Way-point in robot-centric coordinate frame.

1: **if** $|I| = 0$ **then**
2:     **return**
3: **if** $|I| = 1$ **then**
4:     **return** ComputeSNNWayPoint(I)
5: $a_1 \leftarrow \min(\forall i \in I, dist(i, (0, 0)))$
6: $a_2 \leftarrow \min(\forall i \in I, dist(i, a_1))$
7: **if** $\nexists i \in I$, where dist$(a_1, a_i) > ED$ **then**
8:     **return** ComputeSNNWayPoint(a_1)
9: **if** dist$(a_1, (0, 0)) > R$ **then**
10:     **return** ComputeHAWayPoint(a_1, a_2)
11: **return** ComputePerpendicularWayPoint(a_1, a_2)

A common interpretation of Hamilton's work is that each agent attempts to individually decrease their chances of predation. Using this interpretation we cannot justify the additional SNN behavior to the HA motion rule. One may argue the SNN represents a *searching* behavior and, it stands to reason, that an agent has a higher probability of finding a group by following the sole detected agent.

However, it is unclear whether a particular agent is concerned with or even aware of its current predation risk. The observed behaviors seen in many animals, including red tail deer [1], suggest the members within the perimeter of the group all have smaller predation risk than those on the perimeter. If the agents only *desire* to be within the perimeter of the group, then the addition of the SNN behavior to the HA motion rule is justified.

For completeness, we also consider the case when the sensing agent does not detect any other agents. Here, the agent maintains the same motor commands

[2]The algorithmic description is shown in Algorithm 1.
[3]The sign of the perpendicular is chosen at random.

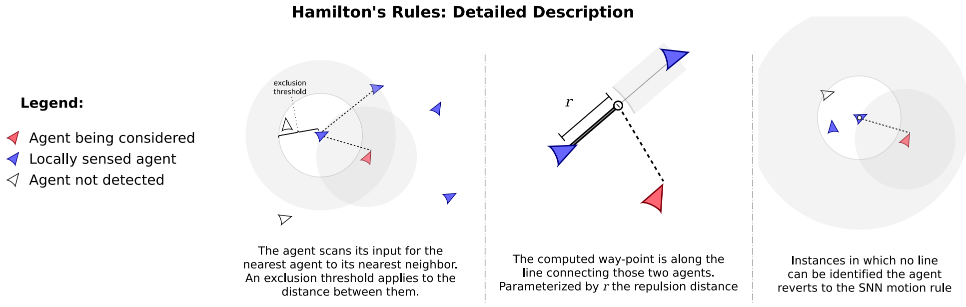

Fig. 1. Detailed pictorial description of the extended HA motion rule. The first and last frames show the effect of the *exclusion distance* on the selection of agents for way-point calculation and the middle frame shows the HA way-point calculation if two agents are selected.

as the previous computation. Figure 1 is a pictorial representation of our extended version of the HA rule.

Sensor-based versus Agent-based Detection

Agent-based input involves representing the location of each detected agent with a single point in space. Another method of translating sensor data to input for a motion rule is sensor-based detection. Sensor-based input is every sensor reading which corresponds to an agent in the field of view.

In agent-based detection an agent has an *a priori* description of an 'agent' used to distinguish *foreground* pixels from *background* pixels. *Foreground* pixels are any pixels in raw sensor data which will be used in the classification of an 'agent' where *background* pixels are all other pixels (e.g., environmental features). Figure 2 shows the data flow from the raw sensor input to the input to the motion rule for agent-based detection. The raw sensor data are first separated into *foreground* and *background* pixels based on the *a priori* description of an agent. Then the *foreground* pixels are averaged together to a single location for each detected agent.

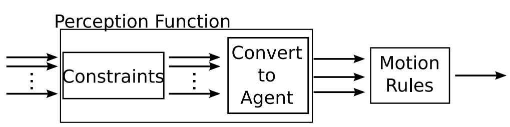

Fig. 2. The data flow from raw sensor data to the motion rule input using agent-based detection. For our implementation, the constraint is a *a priori* shape database and the *Convert to Agent* step averages the *Foreground* pixels to single (x,y)-coordinates for each detected agent.

In sensor-based detection an agent does not have an *a priori* description of an agent. The only information given to the agent is an intensity threshold used to identify *foreground* pixels from *background* pixels. Figure 3 shows the data flow from the raw sensor input to the input to the motion rule for sensor-based detection. The raw sensor data are classified as high or low intensity pixels; *foreground* and *background* pixels respectively. Then the locations of the foreground pixels are passed directly to the motion rule.

The initial set of trials using the sensor-based detection exhibited SNN behavior and not the HA behavior

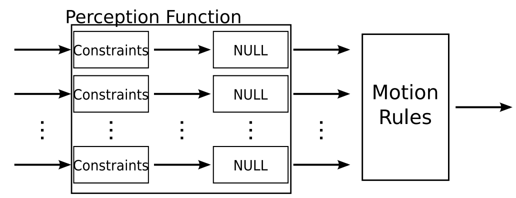

Fig. 3. The data flow from raw sensor data to motion rule input using sensor-based detection. For our implementation, the constraints for each sensor reading is a laser intensity threshold.

as expected because of an implicit assumption built into the HA motion rule. In some cases, multiple sensor readings may result from the same spatially extended agent. Algorithm 1 does not consider the case when the two chosen inputs are consecutive (or nearby) sensor readings, because agent-based detection enforces a separation distance between inputs. However, in sensor-based detection, the chosen HA way-point can be observed to be the same as the SNN way-point because the selected inputs are very nearby readings.

In the original HA rule, there are no distance constraints on the selection of the agent nearest to the *nearest neighbor*. Neither Hamilton's or Viscido's work explains what occurs when the distance between the two selected agents is less than the *repulsion distance*. If the agent's 'intent' is to find a location which will minimize its predation risk, should the agent not select a position which is feasible?[4] For our implementation the *exclusion distance* is used to find two agents which are far enough apart for a feasible point to exist on the line between the two selected agents.

The necessity of a way-point being feasible depends on the mobility and sensing frequency of the agents. If the agent will arrive at the way-point before the next sensing cycle, then the way-point should be a feasible location. However, if the agent cannot achieve the target way-point before the next sensor reading, then one could argue the *exclusion distance* is superfluous for agents in continuous motion. If the computed way-point is always more than one sensation away, then the sensing agent will never arrive at the way-point, making the feasibility

[4] That is, a position which has no agents within the *repulsion radius*.

argument moot.

Now consider when the sensing agent may reach the way-point before the next sensation. The way-point must be a feasible point in the sensing agent's configuration space, but should the way-point put the sensing agent in a location where the nearest agent is within the *repulsion radius*? Under the assumption of constant motion, it is impossible to select a way-point which is feasible in the configuration space while maintaining the *repulsion distance*. For this reason one could argue the *exclusion distance* is only needed for sensory-based input. Line 7 in Algorithm 2 adds the *exclusion distance* criterion.

The Ideal Case

In the ideal case, the detection process would have neither false negatives (FN) nor false positives (FP). In other words, every pixel in the raw sensor data will be correctly identified as either *foreground* or *background*. Given a properly calibrated *exclusion distance*, we expect the behaviors of the HA motion rule using the two detection processes to be equivalent. It follows that if the *exclusion distance* is approximately the same as the diameter of the agent, then it will function as agent-based detection.

PHYSICAL TRIALS

Each agent is an iRobot Create robot and is equipped with a Hokuyo URG-04LX-UG01 scanning laser rangefinder and ASUS Eee PC; see Figure 4. The robots are wrapped in a specific type of fiducial paper (according to the detection method). In agent-based detection, agents are detected by segmenting the scan into individual segments based on distance between consecutive pixels. Then each scan is 'time-warped' to have an equal number of readings and compared to the *a priori* database of agents [13]. For each detected segment, all of the corresponding pixels are averaged into a single (x,y)-position.

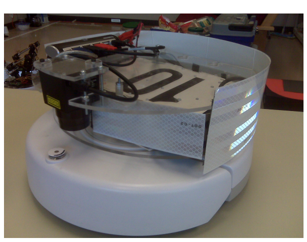

Fig. 4. A single robot wrapped in high reflectance material for sensor-based detection.

For our sensor-based detection, we have modified the Gearbox 9.07 drivers to allow for the detection of intensity. The intensity of the pixels are compared to a given threshold value. Values exceeding the threshold are labeled *foreground* pixels, or *background* otherwise.

Our trials consisted of three different starting formations (Figure 5) and various parameter settings including three different *exclusion distances* (0, 0.5, 1 meter). To compare the two perception functions we conducted trials with six robots in a obstacle free corridor approximately seven robot diameters wide and long enough to be considered infinite for these results.

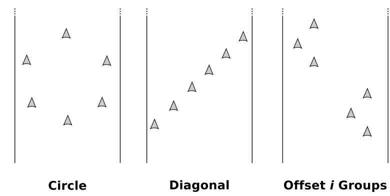

Fig. 5. Starting formations used for the presented trials.

RESULTS

Based on analysis of 63 physical trials, we have observed that sensor-based input into the HA motion rule does indeed still produce flocking behavior. This suggests that the common assumption of agent-based detection is not necessarily justified. Furthermore, these trials support the following claim: *For any motion rule designed with agent-based detection, similar behaviors can be exhibited with sensor-based detection given a particular set of parameters*. More specific to the HA motion rule, there is some range of *exclusion distances* which produce similar behaviors no matter what detection process is used.

Figures 6, 7, 8, and 9 show the *inter-agent* distance plots for all six robots for single trials. The *inter-agent* distance is the distance between the local agent and all inputs given to the HA motion rule. Each color represents the data from one of the six robots.

Figures 6 and 7 are from trials with the same parameter settings but different detection methods. Other than the sheer quantity of inputs, the *inter-agent* distances are similar in these two trials. In both plots, we see that the majority of the robots sense one other robot, and this is at a similar distance. This single robot observation is because the robots are reverting to SNN behavior.

These plots also show the effects FP and FN have on the inputs to the motion rule. Marker B highlights the sensor signature of agents that only appear briefly in Figure 6. Comparing these two figures it is apparent that both trials had a strong tendency to the SNN behaviors. These plots, along with the similarity in the observed behaviors, strongly support the claim above.

Effect of the Exclusion Distance

As expected, the *exclusion distance* did not have any observable affect on the behaviors of the HA motion rule when using agent-based detection, but it did have a substantial effect on the behaviors when using sensor-based detection. When the exclusion distance is zero, the motion rule (using sensor-based detection) always behaves identically to the SNN motion rule, even though the agents are always calculating the HA way-point (due to the implicit assumption described above). This is to be expected because the two selected agents will almost always be adjacent to each other.[5] As the *exclusion distance* increases, the behaviors start to exhibit the HA motion rule more than the SNN motion rule (after some lower-bound of the *exclusion distance*). At some point, if the *exclusion distance* becomes too large, the behaviors will again exhibit the SNN motion rule. This upper-bound exists: a trivial case is when the *exclusion distance* is equal to the size of the sensor's field of view.

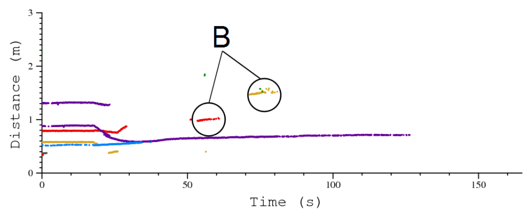

Fig. 6. A single trial of agent-based detection with an *exclusion distance* of 0. Marker B shows the issue of FN for agent-based input.

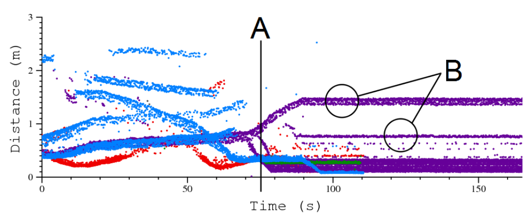

Fig. 7. A single trial of sensor-based detection with an *exclusion distance* of 0. Marker B shows the robustness to FN.

Figures 7, 8, and 9 use sensor-based input with the *exclusion distances* of 0, 0.5, and 1 meter; respectively. These three support the existence of an interval for the *exclusion distance* where the system can successfully exhibit the HA behaviors. When the *exclusion distance* is 0 and 1, we see the system starting to converge [6] to the SNN behavior (marker A in the respective figures). However, in Figure 9 (see marker C), we see two robots which detect robots which are not in the SNN formation (straight line). This means the percentage of HA way-point behaviors is higher in Figure 9 than Figure 7. When the *exclusion distance* is 0.5 we do not see any strong convergence to the SNN behavior.

[5]The only time this assumption fails is when a FP is selected as one of the two inputs

[6]This convergence is the same as described above.

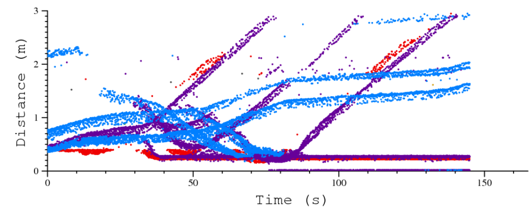

Fig. 8. A single trial of sensor-based detection with a *exclusion distance* of 0.5.

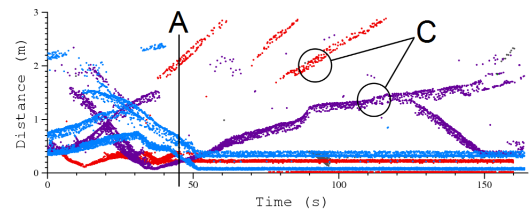

Fig. 9. A single trial of sensor-based detection with a *exclusion distance* of 1.

Discussion

We have identified two behavioral dimensions which are useful in understanding and describing the HA motion rule; *smoothness* and *behavior*. The *smoothness* dimension is the variance in the chosen way-point over a trial and the *behavior* dimension is the percentage the HA motion rule behaviors are observed. Ideally, we would want the behaviors to be completely smooth (rather than jittery) and the behaviors to always exhibit the HA motion rule.

Based on the above criteria we observed parameter settings where the sensor-based detection out performs agent-based detection. To understand why, we implemented agent-extrapolated detection. Agent-extrapolated detection uses the same *a priori* description to detect *foreground pixels*, however the *foreground pixels* are passed directly to the motion rule without being averaged. Figure 10 shows the data flow from the raw sensor data to the input to the motion rule for agent-extrapolated detection.

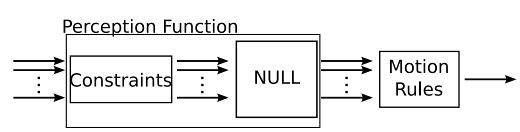

Fig. 10. The data flow from raw sensor data to motion rule input using agent-extrapolated detection. For our implementation, the constraints are the same as in agent-based detection, but the *foreground* pixels are not averaged to single (x,y)-coordinates.

We observed that the behaviors with agent-extrapolated detection are similar to the sensor-based detection with respect to the *exclusion distance*, however, the overall behaviors are more similar to the agent-based detection. We speculate this is due to the effects of sensor noise, occlusions, and the constraints used during the detection process. In the ideal case (given the proper parameter settings) we know the behaviors using all three detection processes would be equivalent. With a realistic sensor, we can expect a certain ratio of FP and FN.

From our trials, we have observed that agent-based and agent-extrapolated detection minimizes the FP while increasing the FN. Sensor-based detection minimizes the FN while increasing the FP. Agent-extrapolated detection has a similar effect on the FP/FN ratio as agent-based detection because both are using the same set of constraints on the sensor data. The only observable difference in the performance of agent-extrapolated and agent-based detection is agent-extrapolated tends to be less *smooth* than agent-based. The *smoothness* in agent-based detection is attained through the averaging of the *foreground* pixels.

Using sensor-based detection, the behaviors become considerably less *smooth* (due to FP) but the agents do not revert to the SNN behavior as often as agent-based and agent-extrapolated. Since there are fewer constraints on the selection of inputs, it is more likely to detect at least one pixel from all of the neighboring agents, which increases the probability the sensing agent can find two inputs which are at least the *exclusion distance* apart. Since two inputs are likely to be selected, we see HA behaviors more often than SNN (assuming a proper parameter set).

Conclusion

Our treatment of the HA motion rule has highlighted the necessity of implementing flocking motion rules on a multi-robot system. Without validating a motion rule on a physical system, it is difficult to prove the rule's completeness and its biological plausibility. We also showed the assumption of agent-based detection is not necessarily justified and supported this by showing similar behaviors can be observed using sensor-based input. The decision of what input is given to the motion rule can have a large impact on the exhibited behaviors. Therefore, it is important for future work to describe the assumptions on the input and how these effect the exhibited behaviors.

References

[1] W. D. Hamilton, "Geometry for the Selfish Herd," *Jour. of Theor. Bio.*, vol. 31, no. 2, pp. 295–311, 1971.

[2] C. W. Reynolds, "Flocks, herds and schools: A distributed behavioral model," *Computer Graphics* 21(4), pp. 25–34, 1987.

[3] A. Barbosa, "Foraging Strategies and Their Influence on Scanning and Flocking Behaviour of Waders," *Jour. of Avian Bio.*, vol. 26, no. 3, pp. 182–186, 1995.

[4] J. Buhl, D. J. T. Sumpter, I. D. Couzin, J. J. Hale, E. Despland, E. R. Miller, and S. J. Simpson, "From Disorder to Order in Marching Locusts," *Science*, vol. 312, no. 5778, pp. 1402–1406, 2006.

[5] A. M. Simons, "Many wrongs: the advantage of group navigation," *Trends in Ecology & Evolution*, vol. 19, no. 9, pp. 453–455, 2004.

[6] A. Huth and C. Wissel, "The simulation of the movement of fish schools," *Jour. of Theor. Bio.*, vol. 156, no. 3, pp. 365–385, 1992.

[7] I. Aoki, "Internal Dynamics of Fish Schools in Relation To Inter-Fish Distance," *Bul. of the Japanese Society of Scientific Fisheries*, vol. 48, no. 3, pp. 1081–1088, 1984.

[8] S. V. Viscido, M. Miller, and D. S. Wethey, "The Dilemma of the Selfish Herd: The Search for a Realistic Movement Rule," *Jour. of Theor. Bio.*, vol. 217, no. 2, pp. 183–194, 2002.

[9] T. Vicsek, A. Czirók, E. Ben-Jacob, I. Cohen, and O. Shochet, "Novel Type of Phase Transition in a System of Self-Driven Particles," *Phys. Rev. Let.*, vol. 75, no. 6, pp. 1226–1229, 1995.

[10] H. Levine, W.-J. Rappel, and I. Cohen, "Self-organization in systems of self-propelled particles," *Phys. Rev. E*, vol. 63, no. 1, pp. 017 101–017 104, 2000.

[11] H. G. Tanner, A. Jadbabaie, and G. J. Pappas, "Stable Flocking of Mobile Agents, Part I: Fixed Topology," in *Proc. of the IEEE CDC*, 2003, pp. 2010–2015.

[12] S. V. Viscido, "The Case for the Selfish Herd Hypothesis," *Comments on Theor. Bio.*, vol. 8, no. 6, pp. 665–684, Nov. 2003.

[13] C. D. Berndt, "Using dynamic time warping to find patterns in time series," in *AAAI-94: Knowledge discovery in databases*, 1994, pp. 229–248.