



HAL
open science

Visual collective behaviors on spherical robots

Diego Castro, Christophe Eloy, Franck Ruffier

► **To cite this version:**

Diego Castro, Christophe Eloy, Franck Ruffier. Visual collective behaviors on spherical robots. *Bioinspiration and Biomimetics*, In press, 20 (2), <10.1088/1748-3190/adaab9>. <hal-04775857v2>

HAL Id: hal-04775857

<https://hal.science/hal-04775857v2>

Submitted on 23 Jan 2025

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Copyright - All rights reserved

Visual collective behaviors on spherical robots

Diego Castro^{1,2}, Christophe Eloy² and Franck Ruffier^{1,*}

¹Aix Marseille Université, CNRS, ISM, Marseille, 13288, France

²Aix Marseille Université, CNRS, Centrale Med, IRPHE, Marseille, 13013, France

*Author to whom any correspondence should be addressed.

E-mail: franck.ruffier@cnr.fr

Accepted *Bioinspir. Biomim.*

15 January 2025

Abstract.

The implementation of collective motion, traditionally, disregard the limited sensing capabilities of an individual, to instead assuming an omniscient perception of the environment. This study implements a visual flocking model in a “robot-in-the-loop” approach to reproduce these behaviors with a flock composed of 10 independent spherical robots. The model achieves robotic collective motion by only using panoramic visual information of each robot, such as retinal position, optical size and optic flow of the neighboring robots. We introduce a virtual anchor to confine the collective robotic movements so to avoid wall interactions. For the first time, a simple visual robot-in-the-loop approach succeed in reproducing several collective motion phases, in particular, swarming, and milling. Another milestone achieved with by this model is bridging the gap between simulation and physical experiments by demonstrating nearly identical behaviors in both environments with the same visual model. To conclude, we show that our minimal visual collective motion model is sufficient to recreate most collective behaviors on a robot-in-the-loop system that be implemented using several individuals, behaves as numerical simulations predict and is easily comparable to traditional models.

Keywords: Flocking, robot-in-the-loop, Collective motion, Milling, Swarming, Optic Flow, Visual information.

1. Introduction

1.1. Related Work

Biological collective motion has inspired extensive efforts to understand, replicate, and utilize it in artificial systems. Nearly four decades ago, Reynolds *et al.* [1] proposed that collective motion could emerge from simple, individual-level rules: (i) **attraction**, the tendency to stay close to others; (ii) **alignment**, the inclination to move in the same direction as neighbors; and (iii) **avoidance**, the need to prevent collisions. Vicsek *et al.* [2] introduced a statistical model where individuals align their velocities with neighbors within a given proximity, incorporating stochastic noise to reflect individual

variation. Couzin *et al.* [3] expanded these ideas by integrating spatial interaction zones, with avoidance in the innermost region, alignment in the intermediate zone, and attraction in the outermost zone.

These **A**ttraction, **A**voidance, **A**lignment rules (“**3A**”) effectively reproduce key collective behavioral phases observed in nature. We define these phases as: **swarming**‡ (non-polarized cohesive motion), **schooling** (highly polarized motion, also known as flocking), and **milling** (circular motion). Models based on these principles are commonly referred to as “3A models” [4, 5, 6, 7].

Many “3A models” rely on artificial methods to enforce rules, such as inter-individual distances [8, 9, 10, 11], global velocity measurements [12, 13], or absolute communication networks [14, 15]. However, such approaches often diverge from the mechanisms used by animals, prompting researchers to explore more biologically plausible (“bio-plausible”) models [16, 17, 18, 19]. Vision-based mechanisms have proven particularly effective, as vision is the primary sensory modality for many species [20, 21, 22, 23].

Vision-based collective motion models have previously been tested on robotic platforms [24, 25]. Moreover, some authors have previously used only vision to successfully model collective motion [26, 27, 28, 29]. Some implementations treated vision as a selective mechanism [30, 31] or do not fully adhere to the fundamental principles of the classical 3A model by not using its structure of “alignment”, “attraction” or “avoidance” rules [27, 32]. Recently, Castro *et al.* (2024) [33] proposed a model that relies solely on visual information while maintaining the structure of 3A models. Their model replicates collective behaviors using “early vision” cues : (i) angular position, (ii) angular size in the field of view as well as (iii) **optic flow**, the visual perception of the apparent angular motion of the environment caused by relative movement. Optic flow is a key navigational tool in nature, used by insects during flight [34, 35] and by fish and birds [36, 37]. Robotic implementations of optic flow have also demonstrated significant potential [38, 39, 40, 41, 42].

1.2. Contribution

This study proposes a **robot-in-the-loop** approach to collective motion that relies exclusively on these “early vision” cues using educational spherical robots. Unlike previous models, this approach avoids reliance on global information or complex communication networks. Instead, it introduces a **visual anchor** to address environmental boundaries, eliminating the need for explicit wall modeling. Through numerical simulations and experiments involving 10 independently controlled robots, the method successfully reproduces diverse collective motion behaviors.

The article is organized as follows: Section 2.1 provides an overview of the challenges of a robotic implementation of 3A-models, with the example of the visual-

‡ Note, this “swarming” phase is a different concept to *swarm robotics*. Here, the **swarming** phase is an emerging behavior that occurs when the group moves in a non-organized motion.

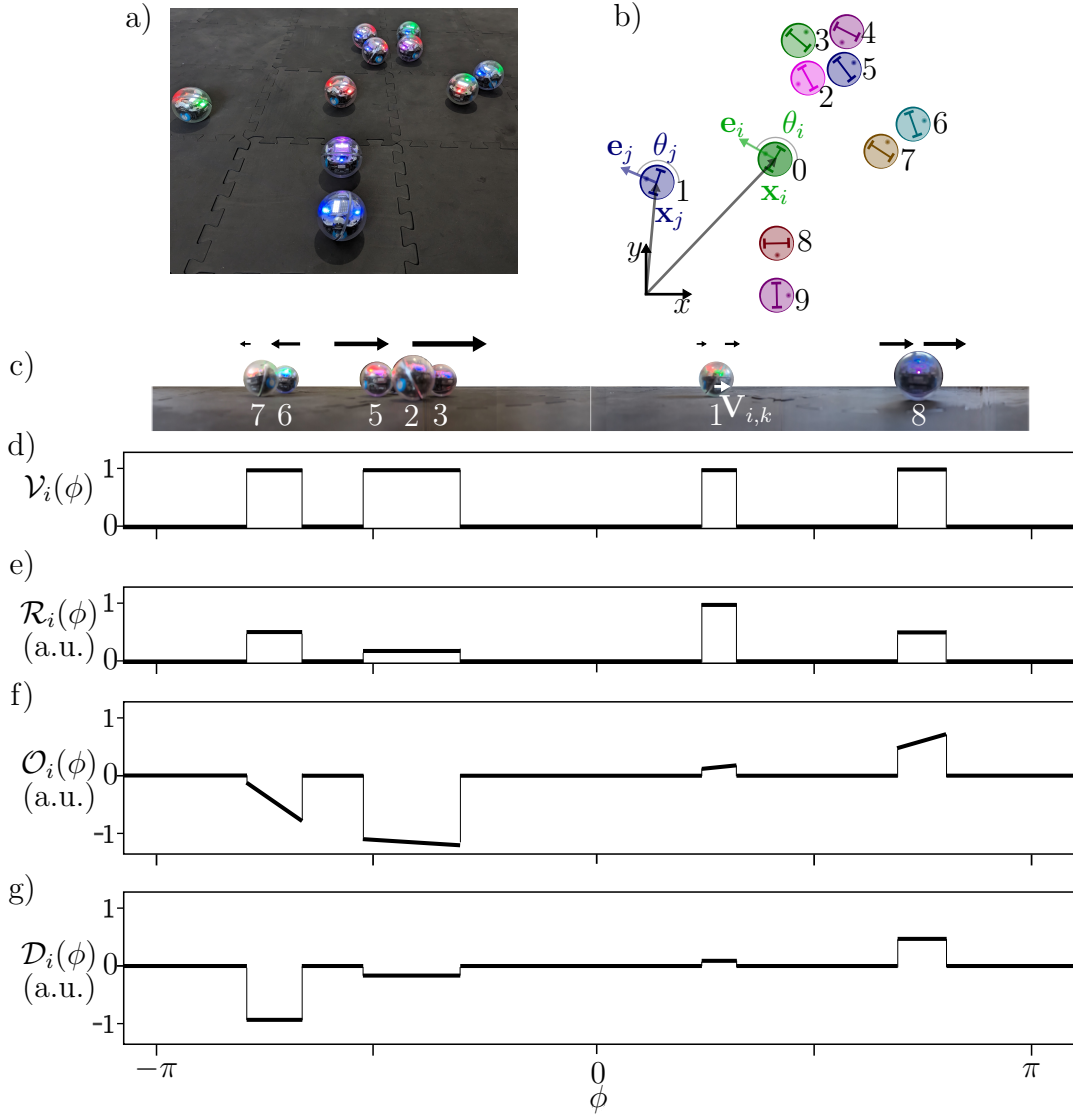


Figure 1. Modeling collective behaviors from optic flow and retinal cues [33]: a) View of the flock positions. b) Representation of flock positions after recognition at time step t . Each i -particle is described by its XY-coordinates and heading unit vector \mathbf{e}_i , with heading angle θ_i . c) Point-of-view of particle θ , showing the relative retinal positions of each particle. The arrows represent the perceived optic flow, with the white arrow indicating the apparent velocity ($\mathbf{V}_{i,k}$). Only the azimuthal component of $\mathbf{V}_{i,k}$ is shown for retinal object k as perceived by $i = 0$. d) $\mathcal{V}(\phi)$: Binary panorama representation of the field of view of particle θ . e) Apparent distance $\mathcal{R}_i(\phi)$ for particle θ . f) Optic flow $\mathcal{O}_i(\phi)$ perceived by particle 1 . g) Optic flow divergence $\mathcal{D}_i(\phi)$ perceived by particle 1 .

based collective motion model proposed by Castro *et al.* (2024) [33] (Fig.1). Section 2.2 describes the robotic platform and the model modifications for implementing on the 10 independently controlled robots including the introduction of the visual anchor concept. Section 2.4 detail the implementation of a feature matching optic flow algorithm and visual collective motion rules. Section 2.5 describes the supplementation

on original model and summarizes robotic setup, while Section 3 presents the results from simulations and robot-in-the-loop experiments. Finally, Section 4 concludes with a discussion of the results and directions for future research.

2. Methods

2.1. Robotic Implementation Challenges

Flocking models based on individual rules of attraction, avoidance, and alignment – commonly referred to as 3A-models – pose several challenges when implemented on robotic platforms. Two significant challenges arise due to differences between theoretical models and real-world implementations.

The first challenge is related to the abstraction of individual agents. Most 3A-models assume that individuals are self-propelled particles moving in a plane. However, many robots either deviate from the particle abstraction or lack natural constraints to planar motion. This raises the question: *Which robot kinematics should be chosen?* The chosen robotic platform must strike a balance between adhering to model assumptions, ensuring adequate sensing capabilities, accommodating computational constraints, and maintaining a suitable size, among other factors.

The second challenge involves transitioning from numerical simulations to physical experiments. Unlike numerical environments, experimental setups are spatially constrained. As a result, collective behaviors must be confined within the experimental area. Without addressing these constraints, implementing flocking behaviors on robots would merely involve a direct adaptation of an established model, such as the one proposed by Castro *et al.*, (2024) [33], to the chosen robotic platform (illustrated in Fig. 1).

The following sections present our solutions to these challenges.

2.2. Robotic Setup

This section describes the robotic setup for implementing a robot-in-the-loop visual flocking model. We propose to use educational spherical robots equipped with wireless control capabilities, while the robot’s sensing capabilities are provided by an external infrastructure. The external infrastructure reconstructed each binary panorama and processed the model described in Section 2.1, controlling the locomotion of each robot independently.

This robot-in-the-loop solution leverages an existing educational robotic platform, the Sphero Bolt® (Fig. 2a). These compact, spherical robots can be controlled in real-time via Bluetooth® using an educational application, which allows users to set a heading, desired velocity, and LED states, as well as read onboard sensor data. Furthermore, several open-source projects extend the functionality of these robots

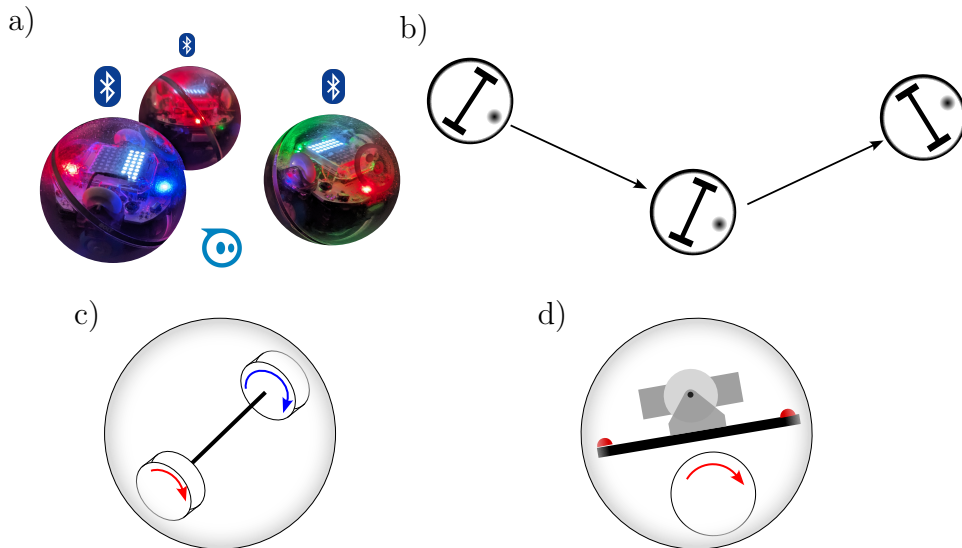


Figure 2. Spherical robotic platform. a) Sphero Bolt® wireless educational robots. b) Example of the robots’ movement, c) Heading control by differential rotation on the two opposite inner wheels, d) Induced nose-up pitch during forward displacement.

by integrating them with programming environments such as ROS, Python, and/or MATLAB®.

Each robot is equipped with a suite of sensors, including an Inertial Measurement Unit (IMU) composed of a 3-axis accelerometer, gyroscope, and magnetometer, as well as a light sensor. Additionally, the robot features four infrared (IR) proximity sensors positioned on its sides, two addressable RGB LEDs (located at the front and rear), and an 8×8 RGB LED matrix on the top.

Locomotion is facilitated by a self-balancing two-wheel enclosed system, with two passive wheels (not shown in Fig.2) maintaining continuous contact with the spherical enclosure. This dual-motor configuration enables linear movement, while stationary turning is achieved via differential motion (Fig.2b-c). The locomotion process induces a pitch displacement dependent on the robot’s speed and the friction coefficient between the spherical enclosure and the surface (Fig.2d).

This pitch displacement is partially mitigated by the robot’s dual-MCU architecture: one microcontroller unit (MCU) manages the communication protocol and high-level commands, while the other is dedicated to motor control. The latter ensures precise implementation of the desired heading and speed while handling low-level sensor readings.

To address the absence of onboard cameras and the inability to integrate them into the robots, an external infrastructure made of a bird’s-eye camera was positioned above the experimental area. The camera used is an industrial RGB global shutter model, the *Daheng Imaging MER2-230-168U3C*, with a frame rate of 168 frames per second (FPS), a resolution of 1920×1200 pixels, and compatibility with interchangeable C-mount lenses (Table 1). A 4 mm, $1/8''$, F2.0–12 non-distorting lens was selected, enabling the camera to be mounted 2.3 m above the robots while covering a 2.7×2.7 m

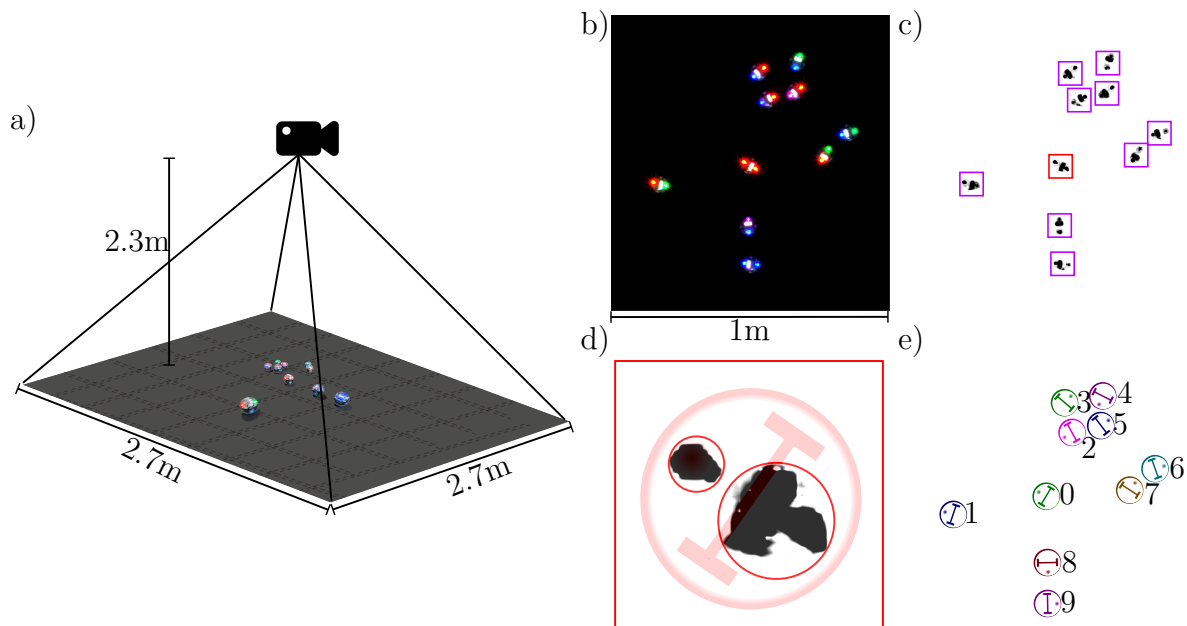


Figure 3. Experimental Setup and external infrastructure. a) Photograph of the experimental setup. b) Sample image captured by the camera (Daheng Imaging MER2-230-168U3C camera equipped with a 4 mm C-mount lens). c) Whole-robot body image segmentation: Segmentation of the robot’s body based on the captured image. d) Recognition of individual light sources and heading: Identification of the front and rear LEDs allows calculation of the robot’s heading vector. e) Robot identification and virtualization: Assignment of a unique identifier to each robot and visualization of their positions and orientations.

working area—approximately 38 times the length of a single robot (Fig. 3a).

Resolution	1920x1200
Frame rate	168 FPS
Shutter time	25 ms
Pixel Bit Depth	10 bit
Digital gain	5 dB
Pixel Data Format	Bayer RG10
Lens f-stop	11
Interface	Python 3.10
Non-distorting lens	4mm (focal length), 1/8" (optical format) , F2.0-12 (aperture)

Table 1. Daheng Imaging MER2-230-168U3C global shutter Camera Settings.

Each robot is assigned a unique color pattern, enabling the identification of its front and back. To maintain visual connectivity between these sections, the back half of the LED matrix is set to emit 50% bright white light. This configuration supports 16 distinct combinations for robot identification, with potential for further expansion using additional colors.

A bird's-eye camera captures a 1920×1200 RGB image, enabling the isolation of each light source on the robots and reduces light bleeding when several robots are in close proximity (Fig. 3b). Whole-robot segmentation is performed on this image, treating the three light sources on each robot as a single entity. This allows for the calculation of the robot's center coordinates (X,Y), which approximate its geometric center. By knowing the radius of the robots in pixels, N cropped images are extracted from these coordinates, each isolating an individual robot.

The cropped images are further segmented to recognize the front and back LEDs as well as the LED matrix. The larger area created by the two light sources facilitates accurate recognition of the robot's front and back (Fig. 3d). Additionally, the unique color patterns from this segmentation are used to assign a distinct identifier to each robot.

Finally, the robot's heading is calculated by tracing the vector formed between the center positions of the front and back LEDs. This computer vision processing yields the robot's identification, X,Y-coordinates, and heading (Fig. 3e).

The next phase, following the identification and localization of each robot, involves reconstructing the binary panorama for each robot.

2.3. Anchoring the collective behavior

In the binary panorama reconstruction, we introduce a novel component to the original model [33]: a visual anchor. The visual anchor acts as an attractive force directed toward a designated location, which, in this case, is the center of the arena. The farther an individual is from this location, the stronger the anchor's influence, causing the individual to turn and face the anchor. The anchor is described by Eq. 1, where \mathcal{R}_i represents the apparent distance function over the binary panorama observed by particle i , and a denotes the particle's radius.

$$\omega_c = \int_{-\pi}^{\pi} \frac{\mathcal{R}_i^2(\phi) \sin(\phi)}{a} d\phi, \quad (1)$$

The inclusion of an anchor bounds the flock to a place and prevent infinite expansion. The anchor is an augmented reality (AR) solution that enables each robot to see the desired center point (Fig. 4a). This anchor is omnisciently visible, regardless of any obstacles between the line of sight and the anchor location (Fig. 1b). It is a new visual channel that uses the same binary representation as the visual surroundings (Fig. 1c). The binary panorama is represented in Python as an ordered tuple list of every retinal object's rising and falling edge, with the last position always being the goal tuple. This process is performed independently for each particle without sharing information between particles.

2.4. Robotic Visual Flocking Model

Avoidance effect (\mathcal{A}) is here introduced in the attraction term (ω_{\odot}). Our previous study [33] did not include any avoidance because most collective motion models are

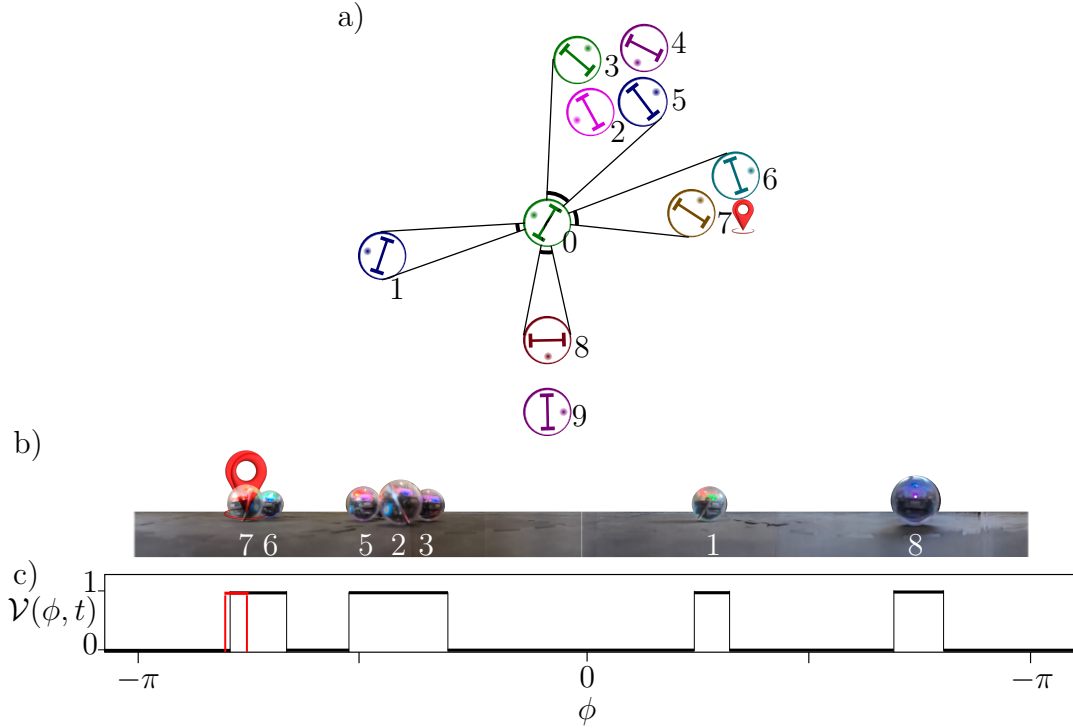


Figure 4. Binary Panorama Reconstruction. a) 2D virtual representation of the binary panorama: The red goal indicates the desired location of the flock. b) Mixed-reality perspective from particle 0: Displays the omniscient red goal as observed from the particle’s point of view. c) Graph representation of all $\mathcal{V}(\phi, t)$: $\mathcal{V}(\phi, t)$ represents the binary panorama view of the field of vision for particle i , with the anchor displayed as a distinct channel (red). This graph can be efficiently stored as a tuple list, representing the rising and falling edges of the binary panorama.

based on the assumption that all particles have some extra room on the z -axis, allowing them to avoid collisions with additional depth. This assumption is primarily based on how fish flocking data is gathered, where the fish can swim above or below each other in a few centimeters of water [43]. However, for any robotic implementation, obstacle or collision avoidance is of critical importance. Fortunately, avoidance and attraction have similar characteristics, being proportional instantaneous responses to the optical size and location of a retinal object. The key difference is in the direction of the response: while attraction is positive, avoidance is negative. Therefore, avoidance is implemented by providing a new piecewise constant function (\mathcal{A}), where attraction is inverted in situations where a particle should avoid a retinal object. This new function \mathcal{A} depends on the width of the retinal object (optical size), specifically the difference between its rising and falling edges ($\Delta\phi$). Note that the apparent distance of a retinal object ($\mathcal{R}(\phi, t)$) inversely depends on this optical size $\Delta\phi$. We choose $\mathcal{A}(\Delta\phi) = -5$ for $\Delta\phi < \pi/10$, which is equivalent to the retinal object being closer than three particle radii ($\mathcal{R} < 3a$). In all other cases, $\mathcal{A}(\Delta\phi) = 1$. The blind spot remains unchanged, giving more weight to the retinal object in front of the particle, regardless of this new function.

Alignment (ω_{\parallel}) remains unchanged with respect to previous study [33], considering

that the divergence/convergence of the optic flow already accounts for retinal objects approaching the focal object.

The optic flow for each retinal object can be decomposed in two components: the azimuthal in its center ($\overline{\mathcal{O}}$) and its radial divergence (\mathcal{D}). The former represents the apparent displacement of retinal features, while the latter represents changes in the size of those features. Divergence is a decrease in size, and convergence is an increase. With a 1D representation, these features are the rising and falling edges of the retinal object, akin to a feature-matching optic flow calculation.

Calculating optic flow as a feature-matching operation heavily relies on accurately identifying the same features across consecutive time steps. The model accomplishes this by finding the gradient of the Euclidean distance of the retinal object position and size between consecutive time steps. Regardless of flock size, there are three possible cases. If, in both time steps, there are the same number of retinal objects, the minimum of the gradient will be continuous and located on the main diagonal of a square matrix. When the number of retinal objects differs, it indicates either a separation or merge of retinal objects. In this case, there is no match, and the retinal objects are ignored in the calculation.

The apparent velocity vector \mathbf{V}_{ik} in polar coordinates is $\mathbf{V}_{ik} = [-\mathcal{D}_i, \overline{\mathcal{O}}_i] \cdot \mathcal{R}_i / U$ with the heading vector $\mathbf{e}_i = [\cos \phi, -\sin \phi]$. The supplemented visual model equations for the collective behavior are presented in Eqs. 2a-f. Note that the noise k_η and the time delay τ are set to 0 for robot-in-the-loop experiments, as they are parameter modeling real-world noise and uncertainty.

$$\dot{\mathbf{x}}_i(t) = U \mathbf{e}_i(t), \quad (2a)$$

$$\dot{\theta}_i(t) = k_\odot \omega_\odot(t) + k_\parallel \omega_\parallel(t) + k_c \omega_c(t) + k_\eta \eta(t), \quad (2b)$$

$$\omega_\odot(t) = \left\langle \int_{-\pi}^{\pi} \mathcal{R}_i^2(\phi, t - \tau) \mathcal{A}(\mathcal{R}_i(t - \tau)) b_\epsilon(\phi, t - \tau) \sin(\phi, t - \tau) d\phi \right\rangle, \quad (2c)$$

$$\omega_\parallel(t) = \left\langle \int_{-\pi}^{\pi} \frac{\mathbf{e}_i(t - \tau) \times \mathbf{V}_{ik}(t - \tau)}{U \mathcal{R}_i(\phi, t - \tau)} \cdot \hat{\mathbf{z}} b_\epsilon(\phi, t - \tau) d\phi \right\rangle, \quad (2d)$$

$$\omega_c(t) = \int_{-\pi}^{\pi} \frac{\mathcal{R}_i^2(\phi, t - \tau) \sin(\phi, t - \tau)}{a} d\phi, \quad (2e)$$

$$\frac{\mathbf{e}_i(t) \times \mathbf{V}_{ik}(t)}{\mathcal{R}_i(t)} \cdot \hat{\mathbf{z}} = -\mathcal{D}_i(\phi, t) \sin(\phi, t) + \overline{\mathcal{O}}_i(\phi, t) \cos(\phi, t). \quad (2f)$$

Where $\mathbf{x}_i(t)$ is the XY-coordinate of particle i at time t , $\mathbf{e}_i(t)$ is the heading vector of particle i at time t , k_\odot , k_\parallel , k_c , k_η are the gains of the attraction, parallel, anchor and noise respectively while $\omega_\odot(t)$, $\omega_\parallel(t)$, $\omega_c(t)$, η are the attraction, parallel, anchor and noise terms at time t . $\hat{\mathbf{z}}$ is the unitary vector that is orthogonal to the $(\hat{\mathbf{x}}, \hat{\mathbf{y}})$ -plane of movement of the robots. ϕ is the retinal angle in the field of view, is considered from the point of view of each particle and counter-clockwise is considered positive. $\mathcal{R}_i(\phi, t)$ is the apparent distance function at a retinal angle ϕ at time t for individual i . $b_\epsilon(\phi, t)$ is the blindspot function $(1 + \epsilon \cos \phi)$. $\mathbf{V}_{ik}(t)$ is the apparent velocity that an individual

i perceives of a retinal object k . Finally τ is the delay between perception and action.

The robotic platform introduces several critical processing factors – Bluetooth, image processing, internal Spheros control command, ... – that affect the overall dynamical response of the robots and its collective behavior as a whole (see Supplementary Information for more details). These extra factors results in a time delay τ with a median of 600ms. The overall functional diagram of the collective motion model is presented in Fig. 5. This model is executed independently and simultaneously for all robots; it does not involve communication between robots, sharing information, and only uses visual data. The same block could be utilized on a robotic platform using LiDAR, onboard cameras, touch sensors, ultrasound, or any other sensor that provides a binary visual panorama as output. All the information used by this model is self-referred and self-contained; there is no global reference frame, coordination, or knowledge of the flock or the environment.

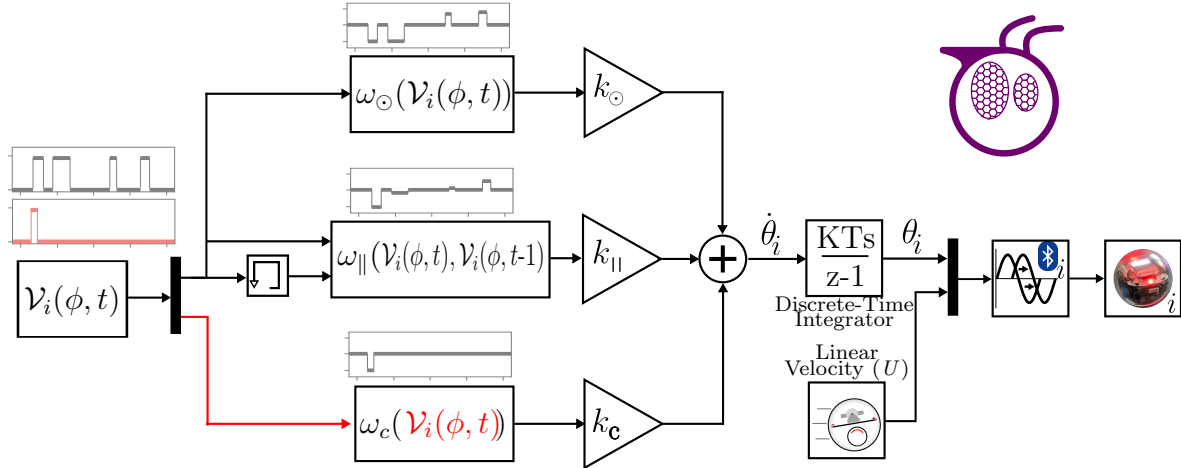


Figure 5. Functional diagram of the collective motion algorithm with robots-in-the-loop.

2.5. Robots-in-the-loop computer architecture

We propose a robots-in-the-loop flocking framework that consists of a set of N Sphero[®] Bolt robots, which lack visual sensors and are controlled via Bluetooth[®]. Their positions and headings are determined by processing images captured by an industrial 1920x1200 RGB, 164 FPS overhead camera. From these positions, a binary panorama is emulated. This binary panorama serves as the input to the collective motion model, as shown in Fig. 6.

There are two main Python programs to handle the implementation of the blocks, as shown in Fig. 6. The first one (*administrator.py*) coordinates the camera communication, data extraction, image processing, object identification, and binary panorama reconstruction (Fig. 3). The binary panorama reconstruction is transferred to N independent routines named *PyRobot.py* (the second program). Each routine

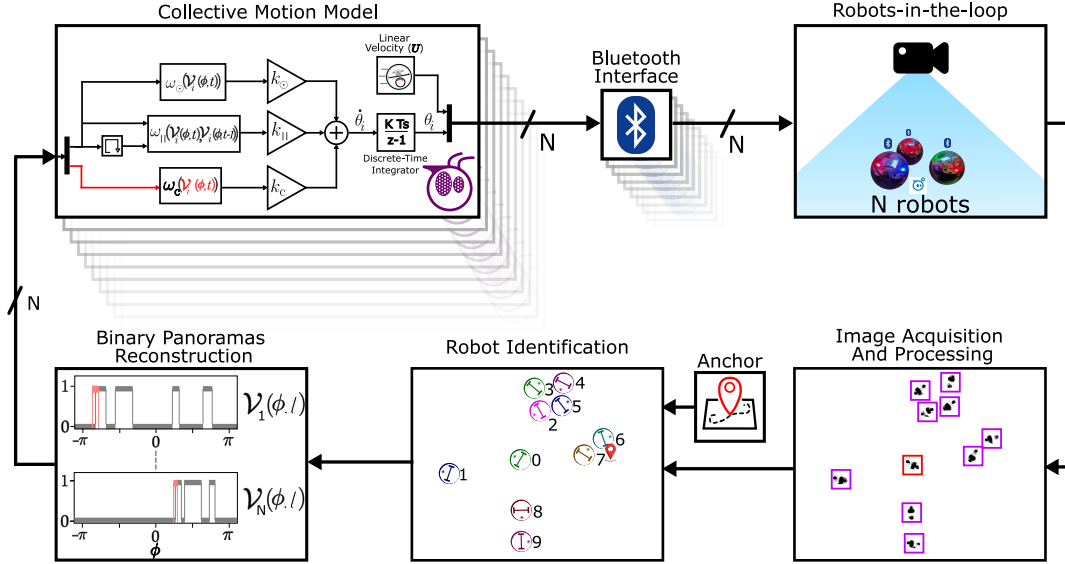


Figure 6. General Block Diagram. The N flocking robots are observed using a 1920x1200 px, 164 fps camera. The captured images are processed to determine the XY-coordinates, heading, and colors of all N robots. Based on these coordinates, each robot is uniquely identified. The XY-coordinates of the Anchor can be set at any point within the image to assist in robot identification. These coordinates are further processed to reconstruct the binary panorama for each spherical robot. Each binary panorama serves as input to an independent collective motion model, which processes the visual panoramic data. These models operate in parallel, and their outputs, in terms of heading, are transmitted to each robot via a Bluetooth interface.

processes the binary panorama of a single robot to calculate its output from the collective motion model and send the desired heading and speed via Bluetooth to that same robot (Fig. 5). Each robot receives the inputs (heading and velocity) and moves accordingly. Each *PyRobot.py* is a routine that can be executed either locally or remotely from the administrator. This setup allows a framework only bounded in the flock size to the maximum number that can fit within the camera’s recognition area and hardware-constrained Bluetooth interface capabilities of the computer. For *Windows*® machines, the theoretical maximum is seven devices per machine; in *Linux*, it is seven per interface. However, in practice, it is recommended to have a maximum of five on the same machine managing the camera and six per interface on independent machines.

2.6. Metrics

To quantitatively distinguish between different dynamical phases, we introduce three global order metrics: polarization (P), milling (M), and opacity (O) [44, 45]. These metrics are defined as follows:

$$P(t) = \left\| \overline{\mathbf{e}_i(t)} \right\|, \quad (3)$$

$$M(t) = \left\| \overline{\mathbf{y}_i(t) \times \mathbf{e}_i(t)} \right\|, \quad (4)$$

$$O(t) = \frac{1}{2\pi} \int_{-\pi}^{\pi} \overline{\mathcal{V}_i(\phi, t)} d\phi, \quad (5)$$

where the overbar denotes an average over all individuals, and the unit vector $\mathbf{y}_i(t) = \frac{\mathbf{x}_i(t) - \bar{\mathbf{x}}_i(t)}{\|\mathbf{x}_i(t) - \bar{\mathbf{x}}_i(t)\|}$ points towards particle i from the center of mass. All three metrics range within the interval $[0, 1]$. The polarization $P(t)$ measures alignment: $P(t) = 0$ corresponds to particles pointing in all directions, while $P(t) = 1$ represents a perfectly aligned school. The milling $M(t)$ quantifies the normalized angular momentum: $M(t) = 0$ corresponds to a straight-line formation, and $M(t) = 1$ represents perfect milling. The opacity $O(t)$ measures the ‘‘occupancy’’ of the visual fields: $O(t) = 0$ indicates no object in the visual field, and $O(t) = 1$ signifies that the entire visual field is obscured. To assess the performance of the phase, we quantify the number of collisions while a simulation or experiment were running. For an experiment, we counted how many frames presented a collision and how many robots were involved. For a simulation we counted collision by defining one as any time the distance between individuals was less than $2a$.

3. Results

3.1. Simulation of Supplemented Visual Model

To make the problem dimensionless, we chose $a = 1$ and $U = 1$. The blind angle parameter and noise strength were set to values that enable the reproduction of most phases: $\epsilon = 1$, $k_\eta = 0.01$. With this approach, three dimensionless parameters remain: the strengths of anchor attraction and alignment, k_c , k_\odot , and k_\parallel . The anchor parameter was chosen to confine robotic experimentation, set practically to the minimum value required to prevent the robots from reaching the arena’s border when no other law is in effect, $k_c = 0.005$.

The original visual model has been supplemented with three significant modifications, influencing the phase distribution and metrics in the parameter space (k_\odot, k_\parallel) :

- inclusion of the avoidance rule ($\mathcal{A}(\phi)$) (Fig. S7),
- addition of an anchor law (ω_c) (Fig.S8),
- introduction of a time delay (τ) (simulation only) to reflect the delay between the desired heading and a perceived change in heading when operating the Spheros robot (Fig.7).

We compared the original model with each successive addition by exploring the parameter space $(k_\odot, k_\parallel) \in [0, 0.3] \times [0, 0.3]$ for a flock of 10 individuals. For each parameter set, we ran 10 simulations over an extended duration ($\Delta t = 5000$). The mean values of P , M , and O were determined by averaging over the final 1000 time

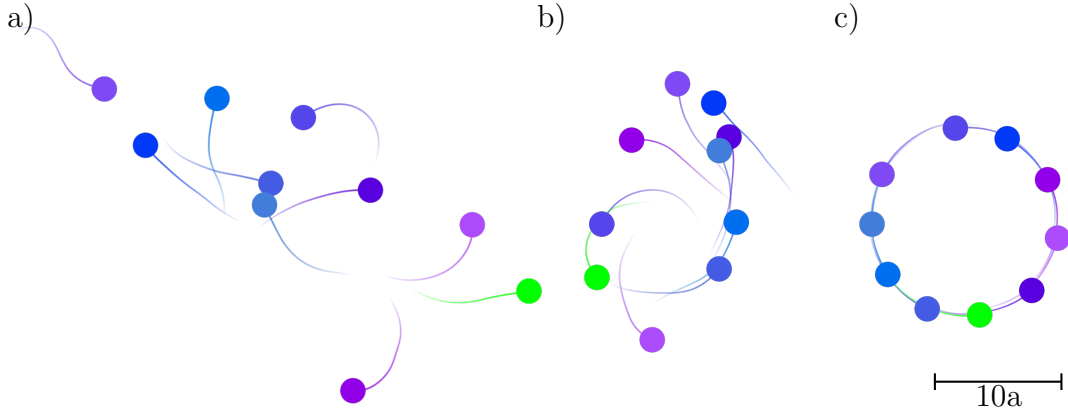


Figure 7. Illustration of the phases observed in simulation for the supplemented visual model with the addition of avoidance, anchor, and time delay as defined in Eqs. 2a, with $\tau = 2.4[a/U]$. Three phases are observed when varying the other two parameters: a) Swarming ($k_{\odot} = 0.25$, $k_{\parallel} = 0.04$) (see SVideo 1); b) Bistable ($k_{\odot} = 0.2$, $k_{\parallel} = 0.25$) (see SVideo 2); c) Milling ($k_{\odot} = 0.07$, $k_{\parallel} = 0.16$) (see SVideo 3).

units to minimize the influence of transients. The simulation results are summarized in phase diagrams (Fig. 8).

We classified the collective phases using the polarization P and milling M parameters:

Schooling when $P > 0.5$ and $M < 0.5$, Milling when $P < 0.5$ and $M > 0.5$, Swarming when $P < 0.5$ and $M < 0.5$.

3.2. Phase Diagrams of the Supplemented Model in Simulation

We further compared the supplemented model and its robotic implementation by exploring the parameter space $(k_{\odot}, k_{\parallel}) \in [0, 0.3] \times [0, 0.3]$ for a flock of 10 independently controlled robots. For robotic experiments, dimensionless parameters were adjusted to match the radius in pixels and velocity in radius per second, resulting in $a = 1$, with other parameters remaining constant ($\epsilon = 1$, $k_{\eta} = 0$, and $k_c = 0.005$).

For each parameter set, five robotic experiments were conducted over an extended period (240 sec or $\Delta t \approx 8500$). The mean values of P , M , and O were calculated by averaging over the final 1000 time units to eliminate transient effects. The experimental results are summarized in phase diagrams (Fig.9a). Examples of swarming, partial schooling in a bistable state, and milling are presented in Fig.9b, Fig.9c, and Fig.9d, respectively. Note that the bistable state has been observed in fish experiments ([46]).

The phase diagram of the original model [33] showcases the reproduction of the main three phases (swarming, schooling, and milling), with the area above $k_{\odot} > 0.15$ and $k_{\parallel} > 0.15$ being bistable (Fig. 8a). A bistable state is when the flock oscillates between milling and schooling (Fig. 8b). By adding repulsion to the model, this bistable phase is removed, but the area where milling appears shrinks to roughly half of the original while the final opacity of the flocks is reduced. Opacity is an inverse metric of flock

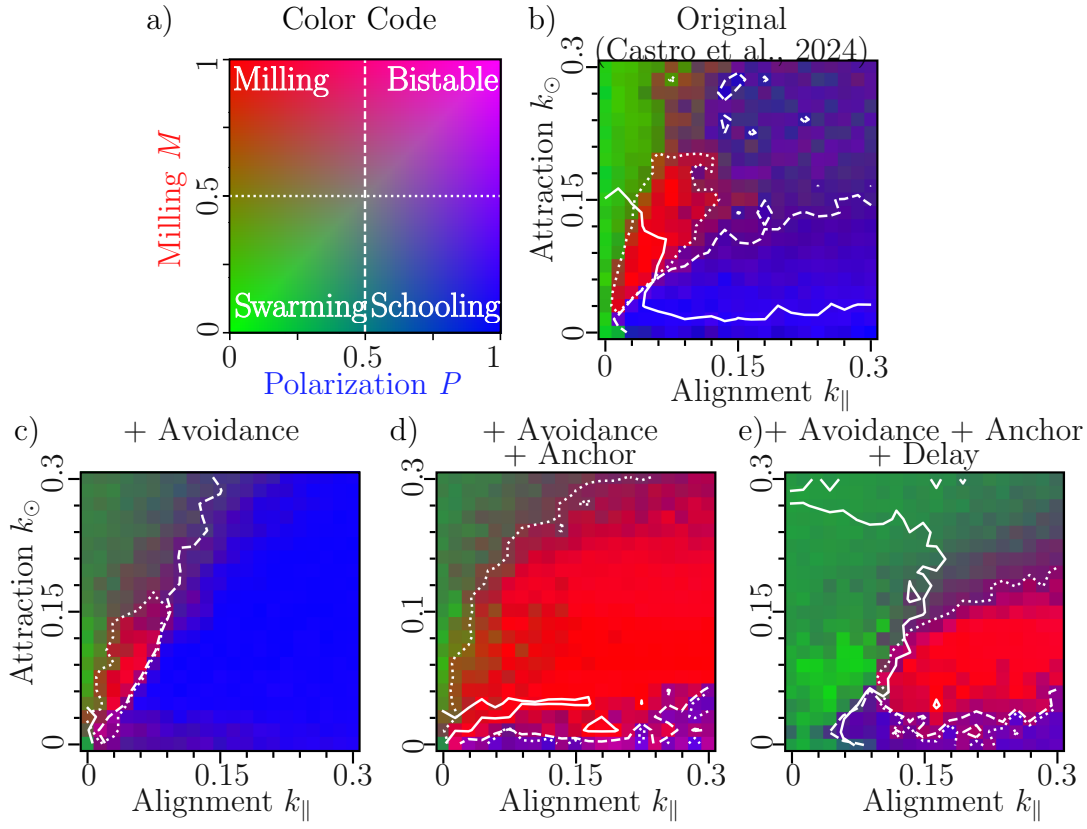


Figure 8. Simulation Phase Diagrams for 10 individuals. The dotted line indicates $P = 0.5$, the dashed line indicates $M = 0.5$, and the solid line indicates $O = 0.3$. a) Color code, b) Original visual model [33], c) Supplemented visual model with avoidance term, $\mathcal{A}(\mathcal{R}(\phi, t))$, d) Supplemented model with avoidance and anchor term, $\omega_c(t)$, e) Supplemented visual model with avoidance, anchoring, and perceived time delay ($\tau = 2.4[a/U]$).

density, and this reduction aligns with the expected effect of adding avoidance.

Fig. 8c shows the introduction of the anchor, which eliminates the schooling phase while the swarming phase area remains unchanged. This change is attributed to the turning effect generated by the anchor. When all particles are in a line, the front particles turn faster than those at the back, resulting in a turn that eventually stabilizes into a circle and a very stable milling pattern.

Adding a time delay ($\tau = 2.4[a/U]$) in the supplemented visual model (Fig. 8d) causes the swarming phase to expand to values above $k_{\odot} = 0.17$ or when $k_{\parallel} < 0.08$. For low attraction ($k_{\odot} < 0.03$), there is bistability, and the milling phase is reduced to an area for alignment above $k_{\parallel} = 0.1$ and medium attraction ($0.06 < k_{\odot} < 0.17$).

3.3. Comparison with the phase diagram of the Spheros' robots-in-the-loop

During the robotic implementation (see Fig. 9), the milling phase remains fairly stable, appearing for high alignment ($k_{\parallel} > 0.15$) across most attraction values ($k_{\odot} < 0.16$). The swarming phase is constant for low alignment ($k_{\parallel} < 0.06$) or high attraction ($k_{\odot} > 0.17$).

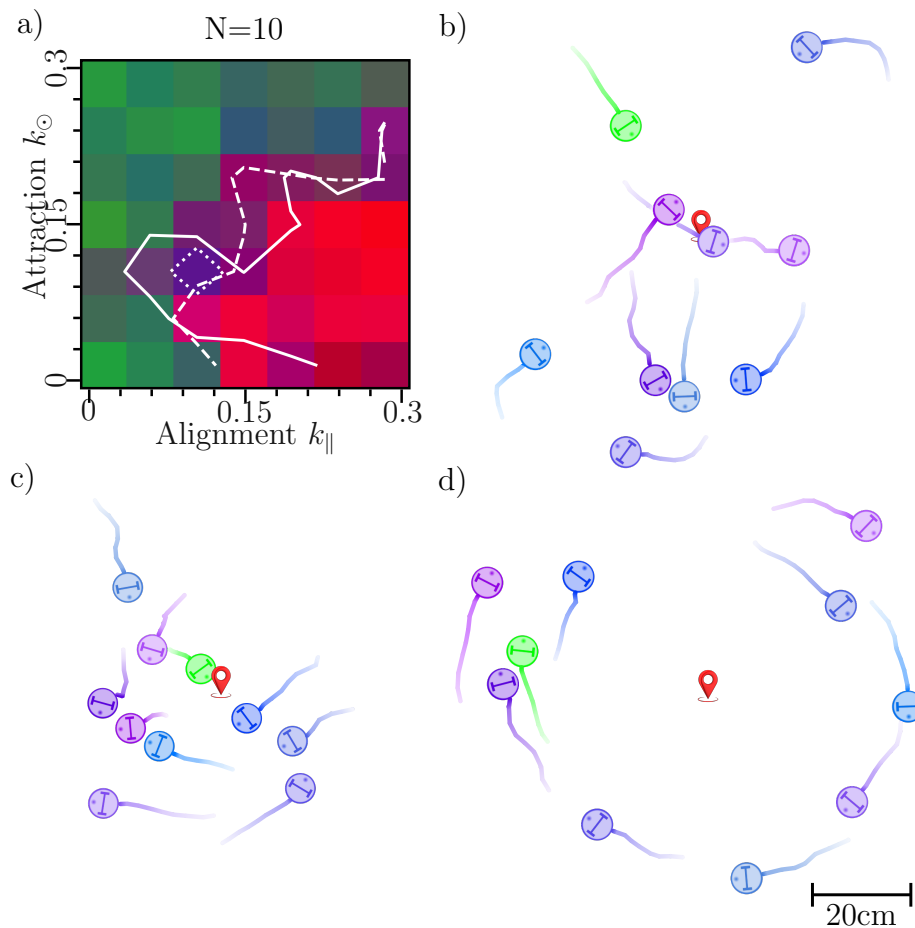


Figure 9. Experimental results for a flock of 10 independently controlled robots (with $U = 4a/sec = 0.14[m/sec]$ (the radius of Spheros robot is $a = 3.5cm$). a) Phase diagram, the dotted line is $P = 0.5$, the dashed line is $M = 0.5$, and the solid line is $O = 0.7$. b) Swarming ($k_{\odot} = 0.25$, $k_{\parallel} = 0.04$) (see raw footage SVideo 4, robotic schematic SVideo 5 or a 3rd person view SVideo 6); c) Bistable ($k_{\odot} = 0.2$, $k_{\parallel} = 0.25$) (see raw footage SVideo 7, robotic schematic SVideo 8 or a 3rd person view SVideo 9); d) Milling ($k_{\odot} = 0.07$, $k_{\parallel} = 0.16$) (see raw footage SVideo 10, robotic schematic SVideo 11 or a 3rd person view SVideo 12).

The avoidance law, subject to the same time delay, results in some collisions that resolve over time, leading to higher opacity values. Similarly, for $k_{\odot} < 0.01$, avoidance is too small, and the flock groups around the anchor without moving. This behavior is somewhat mitigated when there is enough alignment ($k_{\parallel} > 0.15$), but the flock remains more compact than in the simulations.

4. Discussion

In this article, we presented a robot-in-the-loop approach to a visual collective motion model. The robots used are practical to deploy and commercially available: the Sphero BOLT (Fig. 2). The Sphero BOLTs are sphere-like robots with a transparent shell that encloses all electronics, sensors, motors, batteries, and LEDs in a sealed design.

However, due to this sealed design, integrating vision required an external solution. We opted for an external infrastructure made of overhead camera system to recognize the robots, enabling the reconstruction of each robot’s visual field. Each robot was assigned a unique light pattern, visible through the transparent shell, to facilitate position and heading recognition via the overhead camera. (Fig. 3).

The noise effect The collective motion model presented by Castro *et al.* (2024) [33] and the supplemented version introduced in this article, incorporated noise as a component during the computation of angular velocity model output which is directly used to actuate the robot. Introducing the noise on the actuation signal accounted for the noise that errors on the heading recognition might introduce, but did not accounted for recognition errors of the position. And, by extension, the noise that those errors might introduce on the robots’ panorama reconstruction hence on the measurement of the “early vision” cues (apparent distance, optical position, and optic flow signals on Fig. 1). The effect of noise on these “early vision” cues can be particularly significant for the alignment term in the model. Alignment is calculated using optic flow and apparent distance (Eq. 2d), making it three times more sensitive to measurement noise than attraction that only uses apparent distance (Eq. 2c).

Fig. S1 illustrates the combined effect of these recognition errors in open loop, showing that the mean error is approximately 6%. Consequently, the attraction term inherently has a 6% error, while alignment error reaches around 15%. Despite this, the simulation and experimental robotic results align on a macro level as shown in Fig. 9. For comparison, the mean squared error (MSE) between the phase diagram images from robots-in-the-loop and supplemented visual model results is as low as 2.48% (for more details, see Fig. S2).

Effect of collisions We analyzed the number of collisions throughout the evolution of the flock during the examples of swarming, bistability, and milling. If a collision occurred during the robotic experiment or simulation, we counted how many individuals participated on the collision (as seen on Fig. S9). In simulation, a collision is detected when the distance between individuals is less than or equal to $2a$. In the robotic experiment, a collision is detected when the recognized object becomes much larger than a single Sphero. These collisions are calculated throughout the simulation and the robotic experiment, and are presented as additional information on the state of the flock (Fig. S10).

We found that, on average, in one video for each phase, the flock in simulation spent 14% of their time experiencing collisions. The robots exhibited a collision rate of 33% of the time (on average, three times more than the simulation). Interestingly, most simulation collisions were between two particles, and instances of collisions involving more than two particles were rare (only seen on Fig S9-b). In contrast, with the robots, most collisions caused a chain reaction, leading other particles to collide with the stopped particles (as seen on Fig S9-d). The most collisions occurred during the bistable state,

where the simulation had a 21% collision rate compared to 36% for the robots. In both cases, the fewest collisions occurred during the milling phase. The simulation only showed a 6% collision rate, while the robots exhibited a 30% collision rate. The stability of the milling phase is also seen on the minimum inter-distance that, on simulation and on the robots, tends to stabilize around $t=1200$. The timeline of this phase can be seen on Fig S9-c for simulation and Fig S9-f for robots.

The anchor's effect One notable modification to the model was the introduction of an attractive force towards the center of the area of interest, referred to as the anchor (Eq. 1). As shown in Fig. 8c, the main phase is schooling. However, when the anchor is introduced (Fig. 8d), schooling disappears. We hypothesize that the anchor, acting as a force that causes individuals to turn towards the center when they are far from it, challenges the formation of schooling. Notably, schooling is typically observed during flock translation, such as animal migration, where the flock's center of mass moves along the flock's polarized direction.

If the model bounds a flock with a wall, a schooling flock would inevitably interact with the wall. In contrast, our model introduces a force directing individuals toward the a desired location, the center of the flock. This means a schooling flock, while temporarily moving away, would eventually return toward the selected point. Moreover, the anchor acts as an unbiased attraction force, causing individuals to turn either clockwise or counterclockwise. During this turning, the flock depolarizes, leading to phase changes. As a result, any observed schooling is temporary.

The delay's effect Another significant factor contributing to the expansion of milling is the delay. A delay between recognition and actuation has been documented to influence collective motion. [47, 48, 49, 50]. If the anchor interacts with the flock with an attractive force, the delays does it with an alignment force. Because the delay, the particles tend to align with a past velocity. In other words, the polarization perceived by the particles is delayed, which leads to longer convergence on the 2 coordinated phases, only schooling or milling. This longer convergence time that the coordinated phase takes to stabilize creates opportunities for other phases to replace the current coordinated phase. Particularly the swarming phase expands because the flock itself maintains a uncoordinated cohesion on high attraction or low alignment (Fig. 8d). For the case of the bistability region, it expands on the borders of the coordinated phases, as the delay causes the flock to oscillate between milling and schooling phases more often.

Comparison between phase diagrams The comparison between the robotic (Fig. 9a) and the simulated phase diagrams (Fig. 8d) only differs for the small area of polarized bistability. The low-attraction region differs because collisions are more relevant in the experimental case than in simulation, where avoidance was included only in the attraction mechanism (low attraction equals low avoidance). In the high-attraction region, the small difference is attributable to the significant noise present in the

alignment rule, caused by the compounding effect of optic flow calculations based on retinal location and apparent distance. All three primary effects of the robotic implementation expands the area of a stable milling phase.

5. Conclusion

Regardless of these implementation challenges, the collective motion model applied on the robotic platform reproduces several collective patterns, such as swarming, milling, and a bistable phase. To the best of our knowledge, for the first time, a visual flocking implementation using spherical robots produces a phase diagram that matches almost perfectly with that of the simulation counterpart using the same visual model. Such an implementation bridges the gap between collective model simulation and real-world experiments, allowing for a framework independent of the N individuals belonging to the flock.

Future Work For future work, it would be interesting to implement an obstacle avoidance rule based on optic flow. With the divergence of the optic flow, potential collisions could be better avoided. It would also be valuable to explore how the presence of the most influential retinal object could affect the phase distribution or evaluate the current model to identify the average location of the most influential retinal object. Another interesting approach would be to introduce a true blind spot where there is no sensing. Potential applications include the evaluation of a moving anchor or transforming the concept of the anchor into a “guarding dog” individual—one individual that moves around and keeps the flock in the desired position—or to include the anchor as a static individual. Another interesting path would be the implementation to 20 or 40 individuals, or even incorporating a mixed-reality option with virtual individuals that interact with the robots to create flocks of hundreds.

References

- [1] Reynolds C W 1987 Flocks, herds and schools: A distributed behavioral model *Proceedings of the 14th annual conference on Computer graphics and interactive techniques* pp 25–34
- [2] Vicsek T, Czirók A, Ben-Jacob E, Cohen I and Shochet O 1995 *Physical review letters* **75** 1226
- [3] Couzin I D, Krause J, James R, Ruxton G D and Franks N R 2002 *Journal of theoretical biology* **218** 1–11
- [4] Aoki I 1982 *Bull. Japan. Soc. Sci. Fish.* **48** 1081–1088
- [5] Mishra S, Baskaran A and Marchetti M C 2010 *Physical Review E* **81** 061916
- [6] Vicsek T and Zafeiris A 2012 *Physics reports* **517** 71–140
- [7] Solon A P, Caussin J B, Bartolo D, Chaté H and Tailleur J 2015 *Physical Review E* **92** 062111
- [8] Chaté H, Ginelli F, Grégoire G and Raynaud F 2008 *Phys. Rev. E* **77**(4) 046113 URL <https://link.aps.org/doi/10.1103/PhysRevE.77.046113>
- [9] Bialek W, Cavagna A, Giardina I, Mora T, Silvestri E, Viale M and Walczak A M 2012 *Proceedings of the National Academy of Sciences* **109** 4786–4791
- [10] Wen G, Duan Z, Li Z and Chen G 2012 *International Journal of Robust and Nonlinear Control* **22** 1790–1805

- [11] Ashraf I, Bradshaw H, Ha T T, Halloy J, Godoy-Diana R and Thiria B 2017 *Proceedings of the National Academy of Sciences* **114** 9599–9604
- [12] Cavagna A, Del Castello L, Giardina I, Grigera T, Jelic A, Melillo S, Mora T, Parisi L, Silvestri E, Viale M *et al.* 2015 *Journal of Statistical Physics* **158** 601–627
- [13] Zhang H T, Zhai C and Chen Z 2010 *IEEE Transactions on Automatic Control* **56** 430–435
- [14] Shang Y and Bouffanais R 2014 *Scientific reports* **4** 1–7
- [15] Virág C, Vásárhelyi G, Tarcai N, Szörényi T, Somorjai G, Nepusz T and Vicsek T 2014 *Bioinspiration & biomimetics* **9** 025012
- [16] Lazarus J 1979 *Animal Behaviour* **27** 855–865
- [17] Ginelli F and Chaté H 2010 *Physical review letters* **105** 168103
- [18] Nagy M, Ákos Z, Biro D and Vicsek T 2010 *Nature* **464** 890–893
- [19] Bode N W, Franks D W and Wood A J 2011 *Journal of The Royal Society Interface* **8** 301–304
- [20] Strandburg-Peshkin A, Twomey C R, Bode N W, Kao A B, Katz Y, Ioannou C C, Rosenthal S B, Torney C J, Wu H S, Levin S A *et al.* 2013 *Current Biology* **23** R709–R711
- [21] Davidson J D, Sosna M M, Twomey C R, Sridhar V H, Leblanc S P and Couzin I D 2021 *Journal of the Royal Society Interface* **18** 20210142
- [22] Soria E, Schiano F and Floreano D 2021 *Nature Machine Intelligence* **3** 545–554
- [23] Lafoux B, Moscatelli J, Godoy-Diana R and Thiria B 2023 *Communications Biology* **6** 585
- [24] Schilling F, Soria E and Floreano D 2022 *IEEE Access* **10** 28133–28146
- [25] Mezey D, Bastien R, Zheng Y, McKee N, Stoll D, Hamann H and Romanczuk P 2024 *arXiv preprint arXiv:2406.17106*
- [26] Collignon B, Séguret A and Halloy J 2016 *Royal Society open science* **3** 150473
- [27] Bastien R and Romanczuk P 2020 *Science advances* **6** eaay0792
- [28] Krongauz D L and Lazebnik T 2023 *PLoS One* **18** e0270318
- [29] Krongauz D L, Ayali A and Kaminka G A 2024 *PLOS Computational Biology* **20** e1011796
- [30] Pearce D J G, Miller A M, Rowlands G and Turner M S 2014 *Proceedings of the National Academy of Sciences* **111** 10422–10426 URL <https://www.pnas.org/doi/abs/10.1073/pnas.1402202111>
- [31] Soria E, Schiano F and Floreano D 2019 The influence of limited visual sensing on the reynolds flocking algorithm 2019 *Third IEEE International Conference on Robotic Computing (IRC)* (IEEE) pp 138–145
- [32] Ito S and Uchida N 2024 Selective decision-making and collective behavior of fish by the motion of visual attention (*Preprint* <https://academic.oup.com/pnasnexus/article-pdf/3/7/pgae264/58645783/pgae264.pdf>) URL <https://doi.org/10.1093/pnasnexus/pgae264>
- [33] Castro D, Ruffier F and Eloy C 2024 *Phys. Rev. Res.* **6**(2) 023016 URL <https://link.aps.org/doi/10.1103/PhysRevResearch.6.023016>
- [34] Franceschini N, Ruffier F and Serres J 2007 *Current Biology* **17** 329–335
- [35] Portelli G, Ruffier F and Franceschini N 2010 *Journal of Comparative Physiology A* **196** 307–313
- [36] Serres J R, Evans T J, Åkesson S, Duriez O, Shamoun-Baranes J, Ruffier F and Hedenström A 2019 *Journal of the Royal Society Interface* **16** 20190486
- [37] Bonnen K 2023 *Current Biology* **33** R30–R32
- [38] Ruffier F and Franceschini N 2005 *Robotics and Autonomous Systems* **50** 177–194
- [39] Floreano D, Pericet-Camara R, Viollet S, Ruffier F, Brückner A, Leitel R, Buss W, Menouni M, Expert F, Juston R *et al.* 2013 *Proceedings of the National Academy of Sciences* **110** 9267–9272
- [40] Expert F and Ruffier F 2015 *Bioinspiration & biomimetics* **10** 026003
- [41] Bergantin L, Harbaoui N, Raharijaona T and Ruffier F 2021 *Journal of the Royal Society Interface* **18** 20210567
- [42] De Croon G C, Dupeyroux J J, De Wagter C, Chatterjee A, Olejnik D A and Ruffier F 2022 *Nature* **610** 485–490
- [43] Rosenthal S B, Twomey C R, Hartnett A T, Wu H S and Couzin I D 2015 *Proceedings of the National Academy of Sciences* **112** 4690–4695

- [44] Calovi D S, Lopez U, Ngo S, Sire C, Chaté H and Theraulaz G 2014 *New journal of Physics* **16** 015026
- [45] Filella A, Nadal F, Sire C, Kanso E and Eloy C 2018 *Physical review letters* **120** 198101
- [46] Lafoux B, Bernard P, Thiria B and Godoy-Diana R 2024 *Physical Review E* **110** 034613
- [47] Mijalkov M, McDaniel A, Wehr J and Volpe G 2016 *Physical Review X* **6** 011008
- [48] Cavagna A, Conti D, Creato C, Del Castello L, Giardina I, Grigera T S, Melillo S, Parisi L and Viale M 2017 *Nature Physics* **13** 914–918
- [49] Holubec V, Geiss D, Loos S A, Kroy K and Cichos F 2021 *Physical review letters* **127** 258001
- [50] Zhou Y, Wang T, Lei X and Peng X 2024 *Chaos, Solitons and Fractals* **180** 114596 ISSN 0960-0779
URL <https://www.sciencedirect.com/science/article/pii/S0960077924001474>
- [51] Castro D, Eloy C and Ruffier F 2024 Data and code of: Visual collective behaviors on spherical robots URL osf.io/ugwr9

Acknowledgements

We would also like to thank the two reviewers for their valuable suggestions and comments.

The participation of D.C. in this research was made possible by joint PhD grant from Aix-Marseille University.

D.C. and F.R. were also supported by Aix Marseille University and the CNRS (Life Science, Information Science, and Engineering and Science & technology Institutes).

The facilities for the experimental tests has been mainly provided by ROBOTEX 2.0 (Grants ROBOTEX ANR-10-EQPX-44-01 and TIRREX ANR-21-ESRE-0015).

Data Availability

Data acquired during the experiments and simulations are available online [51]

Video Accessibility

Links to supplemental videos are provided in the supplementary information and can also be found at: youtube or [51].

Author contributions statement

D.C. performed the simulations, the robotic experiments, and data consolidation. C.E. & F.R. outlined the robot in the loop experiment and supervised its implementation. All authors: Proofread the manuscript.

Supplementary information of Visual collective behaviors on spherical robots

Diego Castro^{1,2}, Christophe Eloy² and Franck Ruffier^{1,*}

¹Aix Marseille Université, CNRS, ISM, Marseille, 13288, France

²Aix Marseille Université, CNRS, Centrale Med, IRPHE, Marseille, 13013, France

*Author to whom any correspondence should be addressed.

E-mail: franck.ruffier@cnr.fr

Submitted to: *Bioinspir. Biomim.*

23 January 2025

S1 Spheros' dynamical response and time delay τ estimation

The robotic setup is composed of several steps, from setting a heading and velocity, to recognizing a XY-coordinate and heading from the image. Each step involves their own input treatment, response time and processing. As such, we performed a step response analysis for both heading and velocity independently. This analysis allows to see if there are what are significant changes from the intimate response of the simulation.

Fig. S1 a, shows the average step response for a change in the heading of the robot. The change ranged from 1 to 180 degrees. Fig. S1 b, shows the average step response for a change in the velocity of the robot. The change ranged from 1 to 25 a/sec. Fig. S1 c, shows the average step response for a change in the yaw $\dot{\theta}$ of the robot. Fig. S1 d-e, shows this speeds ramps in the measured variables. All steps responses were performed 5 times with 10 different robots. The Y-axis is a percentage of the desired heading. Finally, Fig. S1 f, shows the measurement sensitivity to changes in heading and Fig. S1 g, shows the error in following a desired yaw $\dot{\theta}$. These results show that the system has a median time delay τ of approximately 600ms in responding to any input signal change. The image processing takes from 10-50ms, the image acquisition take from 25-50ms. Leaving at best 400ms delay to the communication between the control and the robot, the inner processing and the actuation of the input. Therefore, the visual input for the collective motion model must reflect this transport delay.

S2 Direct comparison between Supplemented visual model and robot-in-the-loop Experiments

The difference between Phase Diagrams constructed from simulated and robots-in-the-loop experiments for 10 individuals is very marginal Fig. S2. The mean squared error

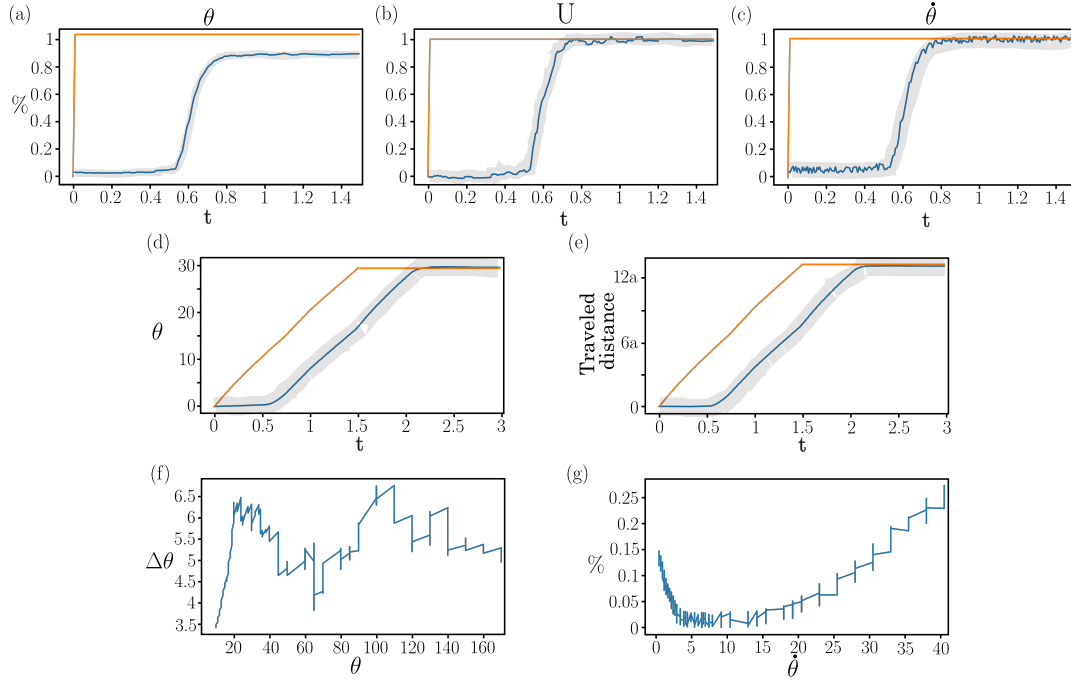


Fig. S1. Spheros' dynamical response (The grey shade is the average of normalized standard deviation) a) Average normalized heading θ step response of the 520 experiments. b) Average normalized velocity U step response [a/sec] of the 220 experiments. c) Average normalized yaw $\dot{\theta}$ step response [degrees] of the 220 experiments. d) Average heading θ ramp response [degrees/sec] of the 220 experiments. e) Average velocity step response as seen in traveled distance (with $a = 3.5\text{cm}$) of the 220 experiments. f) Heading θ measurement average sensitivity in degrees of the 520 experiments. g) Yaw $\dot{\theta}$ measurement average sensitivity in percentage of the 520 experiments.

– MSE – between the phase diagram images of robots-in-the-loop and supplemented visual model is 2.48%.

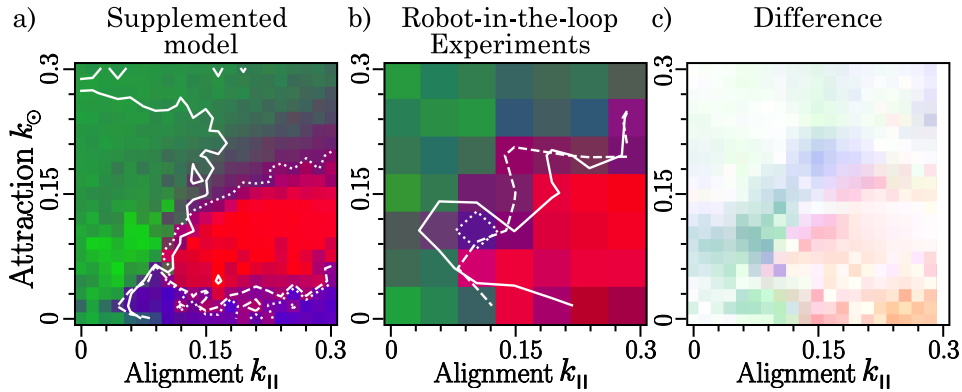


Fig. S2. Difference between the Phase Diagrams. a) Supplemented visual model (10 individuals), b) Phase Diagram of Spheros' robots-in-the-loop (10 robots), c) Raw difference image difference between both phase diagram. The mean squared error – MSE – between the phase diagram images of robots-in-the-loop and supplemented visual model is 2.48%.

S3 Image Processing Technical information

The camera details are presented on Table S1 The detailed dual image segmentation

Resolution	1920x1200
Frame rate	168fps
Pixel Size	5.86uM
Sensor Type	Sony IMX174
Optical Size	1/1.2"
Shutter Type	Global Shutter
Shutter time	52us 1s
ADC Bit Depth	10bit
Pixel Bit Depth	8bit, 10bit
Digital gain	0dB 24dB
Pixel Data Format	Bayer RG8 / Bayer RG10
Lens mount	type-C
Power Consumption	~2.7W@5V

Table S1. Camera: *Daheng Imaging MER2-230-168U3C* specifications.

example is presented on Fig. S3.

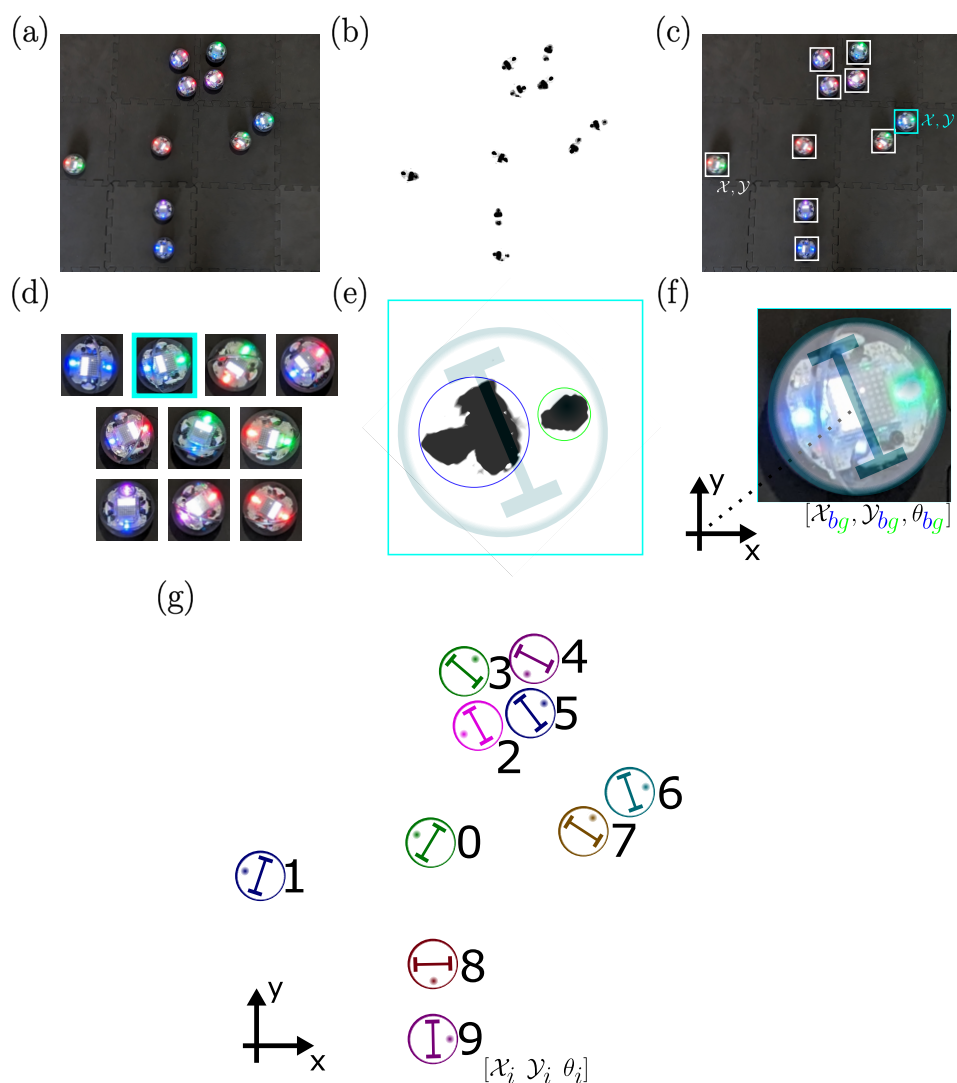


Fig. S3. Illustration of the phases observed for the original model. We observe three phases when varying the other two parameters: (a) Swarming ($k_{\odot} = 0.15$, $k_{\parallel} = 0$); (b) Schooling ($k_{\odot} = 0.07$, $k_{\parallel} = 0.3$); (c) Milling ($k_{\odot} = 0.07$, $k_{\parallel} = 0.05$).

Feature Matching Optic Flow

Fig. S4 showcases examples of what real world each of the components of the optic flow would look like. It also correlated each with a binary panorama equivalency. Finally it shows a mixed case that would be a normal optic flow generation for both image and 1D cases.

For the feature matching optic flow the gradient generated when the amount of retinal object differ between consecutive time steps is shown on Fig. S5.

S4 Digital implementation of the supplemented visual model

The model presented on Eqs. 1, are the integral form of the model that use of the visual panorama. $\mathcal{V}_i(\phi, t - \tau)$ is a piece-wise constant function that can be expressed as a set unit boxcar functions (one for each retinal object).

Each retinal object can be described as a rising $\varphi_{\uparrow k}$ and its corresponding falling edge $\varphi_{\downarrow k}$. Furthermore, from these edges, we can compute the shade's mid-point $\varphi_k = (\varphi_{\uparrow k} + \varphi_{\downarrow k})/2$, and the retinal's object half-width $\Delta\varphi_k = (\varphi_{\downarrow k} - \varphi_{\uparrow k})/2$. The apparent distance, $\mathcal{R}_i(\phi, t - \tau)$ is defined as $\mathcal{V}_i(\phi, t - \tau)$ weighted by $\frac{a}{\sin \Delta\varphi_k}$, thus depending on the inverse of the optic size of the retinal object.

This means that $\mathcal{R}_i(\phi, t - \tau)$ for a binary $\mathcal{V}_i(\phi, t - \tau)$, is a function independent of ϕ and is only be dependent on the k -retinal's object half-width ($\mathcal{R}_k(\Delta\varphi_k)$). Similarly, the avoidance function can be expressed as a function of $\Delta\varphi_k$, and is independent of ϕ . We add the subscript k , which identifies a single retinal object on the binary panorama of the individual i . Note that a retinal object k is different from an individual j , as in a retinal object there can be occlusion or aggregation of any j -particles on the i -th binary panorama. Then, we can rewrite Eq. 1c as:

$$\omega_{\odot}(t) = \left\langle \sum_k \mathcal{R}_i^2(\Delta\varphi_k, t - \tau) \mathcal{A}(\Delta\varphi_k, t - \tau) \int_{\varphi_{\uparrow k}}^{\varphi_{\downarrow k}} b_{\epsilon}(\phi, t - \tau) \sin(\phi, t - \tau) d\phi \right\rangle \quad (\text{S1})$$

where $b_{\epsilon}(\phi, t - \tau) = 1 + \epsilon \cos(\phi, t - \tau)$. And developing the integral we get

$$\omega_{\odot}(t) = \left\langle \sum_k 2\mathcal{R}_i^2(\Delta\varphi_k, t - \tau) \mathcal{A}(\Delta\varphi_k, t - \tau) \sin(\Delta\varphi_k, t - \tau) \sin(\varphi_k, t - \tau) (1 + \epsilon \cos(\varphi_k, t - \tau) \cos(\Delta\varphi_k, t - \tau)) \right\rangle \quad (\text{S2})$$

And here $\mathcal{R}_i(\Delta\varphi_k, t - \tau) = \frac{a}{\sin(\Delta\varphi_k, t - \tau)}$, Resulting in:

$$\omega_{\odot}(t) = \left\langle \sum_k \frac{2a^2 (1 + \epsilon \cos(\varphi_k, t - \tau) \cos(\Delta\varphi_k, t - \tau))}{\sin(\Delta\varphi_k, t - \tau)} \sin(\varphi_k, t - \tau) \right\rangle \quad (\text{S3})$$

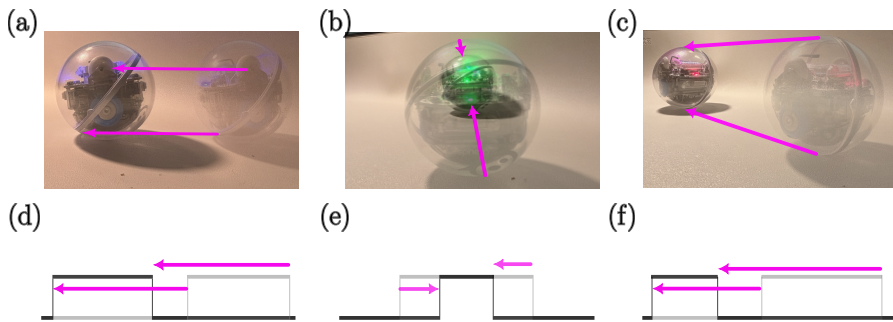


Fig. S4. Optic flow. a) Purely translational, b) Purely Convergent, c) Mixed Case, d) 1-D feature matching purely translational, e) 1-D feature matching purely convergent, f) 1-D feature matching mixed case.

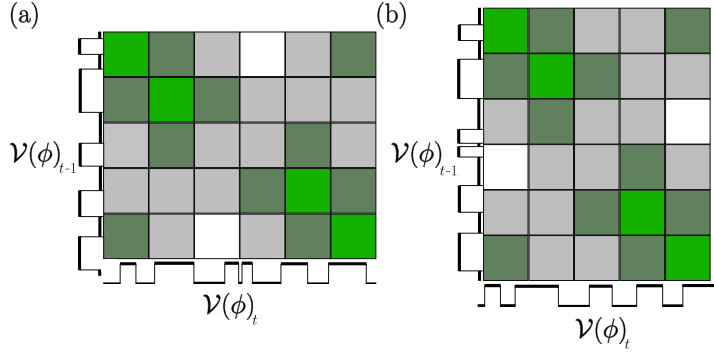


Fig. S5. Retinal object matching between consecutive time steps. a) Separation of retinal object, b) Merging of retinal objects.

By substituting the $\langle \cdot \rangle$ normalization the implemented equation is:

$$\omega_{\odot}(t) = \frac{\sum_k \frac{2a^2(1+\epsilon \cos(\varphi_k, t-\tau) \cos(\Delta\varphi_k, t-\tau))}{\sin(\Delta\varphi_k, t-\tau)} \sin(\varphi_k, t-\tau)}{\sum_k \frac{2a^2(1+\epsilon \cos(\varphi_k, t-\tau) \cos(\Delta\varphi_k, t-\tau))}{\sin(\Delta\varphi_k, t-\tau)}} \quad (\text{S4})$$

The next step is to do the same for the alignment component. First, we replace Eq. 1f in Eq. 1d resulting in:

$$\omega_{\parallel}(t) = \int_{-\pi}^{\pi} U^{-1}(\overline{\mathcal{O}}_i(\phi, t-\tau) \cos(\phi, t-\tau) - \mathcal{D}_i(\phi, t-\tau) \sin(\phi, t-\tau)) b_{\epsilon}(\phi, t-\tau) d\phi \quad (\text{S5})$$

Note, that the visual alignment can also be written in polar coordinates (ϕ, r) , with an azimuthal and radial component as follows:

$$\omega_{\parallel}(t) = \langle \omega_{\parallel, \phi}(t) + \omega_{\parallel, r}(t) \rangle \quad (\text{S6})$$

$\mathcal{D}_i(\phi, t-\tau)$ and $\overline{\mathcal{O}}_i(\phi, t-\tau)$, similarly to $\mathcal{R}_i(\phi, t-\tau)$, are piece-wise constant function which are independent of ϕ but dependent on $\Delta\varphi_k$.

Taking Eq. S5 and Eq. S6:

$$\omega_{\parallel, \phi}(t) = \sum_k \mathcal{O}_i(\Delta\varphi_k, \phi, t-\tau) \left(\int_{\varphi_{\uparrow k}}^{\varphi_{\downarrow k}} \cos(\phi, t-\tau) b_{\epsilon}(\phi, t-\tau) d\phi \right) \quad (\text{S7})$$

$$\omega_{\parallel, r}(t) = \sum_k -\mathcal{D}_i(\Delta\varphi_k, \phi, t-\tau) \left(\int_{\varphi_{\uparrow k}}^{\varphi_{\downarrow k}} \sin(\phi, t-\tau) b_{\epsilon}(\phi, t-\tau) d\phi \right) \quad (\text{S8})$$

Eq. S8 is the exact form as Eq. S1 with the weight of the integral being $\mathcal{D}_i(\Delta\varphi_k, t-\tau)$ instead of $\mathcal{R}_i^2(\Delta\varphi_k, t-\tau) \mathcal{A}(\Delta\varphi_k, t-\tau)$ resulting in:

$$\omega_{\parallel, r}(t) = \sum_k \frac{-2\mathcal{D}_i(\Delta\varphi_k, t-\tau)}{\mathcal{R}_i(\Delta\varphi_k, t-\tau)} \sin(\varphi_k, t-\tau) (1 + \epsilon \cos(\varphi_k, t-\tau) \cos(\Delta\varphi_k, t-\tau)) \quad (\text{S9})$$

As for Eq. S7 the developing the integral and assuming $\Delta\phi \approx \sin \Delta\phi$ we have:

$$\omega_{\parallel, \phi}(t) = \sum_k \frac{\mathcal{O}_i(\Delta\varphi_k, t-\tau)}{\mathcal{R}_i(\Delta\varphi_k, t-\tau)} (2 \cos(\varphi_k, t-\tau) (1 + \epsilon \cos(\Delta\varphi_k, t-\tau) \cos(\varphi_k, t-\tau)) + \epsilon(1 - \cos(\Delta\varphi_k, t-\tau))) \quad (\text{S10})$$

Assuming that $\cos \Delta\varphi_k \approx 1$ and using Eq. 1f, then Eqs. S9-S10 become:

$$\omega_{\parallel}(t) \approx \left\langle \sum_k 2 \frac{\mathbf{e}_i(t-\tau) \times \mathbf{V}_{ik}(t-\tau)}{U\mathcal{R}_i^2(\Delta\varphi_k, t-\tau)} b_{\epsilon}(\varphi_k, t-\tau) \right\rangle \quad (\text{S11})$$

Substituting the normalization and the blindspot, Eq. S11 becomes:

$$\omega_{\parallel}(t) \approx \frac{\sum_k 2 \frac{\mathbf{e}_i(t-\tau) \times \mathbf{V}_{ik}(t-\tau)}{U\mathcal{R}_i^2(\Delta\varphi_k, t-\tau)} (1 + \epsilon \cos(\varphi_k, t-\tau))}{\sum_k 2 \frac{\|\mathbf{V}_{ik}(t-\tau)\|}{U\mathcal{R}_i^2(\Delta\varphi_k, t-\tau)} (1 + \epsilon \cos(\varphi_k, t-\tau))} \quad (\text{S12})$$

S5 Original Visual Collective Motion Model (Castro *et al.*, 2024) [1]

Our model considers a system of N self-propelled particles confined to a plane (Fig. 1a-b). Each particle is represented as an object with a radius a , moving with a constant speed U . The position of each particle is denoted by \mathbf{x}_i , and its orientation is described by the unit vector $\mathbf{e}_i = [\cos \theta_i, \sin \theta_i]$. The collective motion model interacts with the particles by adjusting their angular velocity at each time step.

$$\dot{\mathbf{x}}_i = U\mathbf{e}_i, \quad (\text{S13})$$

$$\dot{\theta}_i = k_{\odot} \omega_{\odot} + k_{\parallel} \omega_{\parallel} + k_{\eta} \eta, \quad (\text{S14})$$

We represent the visual surroundings of the particles as retinal objects on a one-dimensional (1-D) binary panorama (Fig. 1c). This binary panorama is centered on the focal particle's heading, marking the presence of objects along the line of sight and generating binary retinal objects (Fig. 1d). Since all particles are perceived with the same magnitude, multiple particles may aggregate into a single, larger retinal object. Consequently, there are typically more i -particles than k -retinal objects. A apparent distance ($\mathcal{R}_i(\phi)$) is computed from this binary panorama (Fig. 1e), which is then used to calculate the attraction component of the angular velocity, ω_{\odot} (Eq. S15).

The optic flow perceived by each particle (illustrated by the black arrows in Fig. 1c) can be decomposed into two components: a radial component representing the translational optic flow (the average of the function shown in Fig. 1f), and an azimuthal component representing the divergence or convergence of the optic flow (Fig. 1g). The apparent velocity vector for each retinal object ($V_{i,k}$) can be determined using these two components of optic flow. This apparent velocity is employed to compute the alignment component of the angular velocity, ω_{\parallel} (Eq. S16). Finally, the individuality of each particle is incorporated into the angular velocity as a standard Wiener process (η).

$$\omega_{\odot} = \left\langle \int_{-\pi}^{\pi} \mathcal{R}_i^2(\phi) b_{\epsilon}(\phi) \sin \phi d\phi \right\rangle, \quad (\text{S15})$$

$$\omega_{\parallel} = \left\langle \int_{-\pi}^{\pi} \frac{\mathbf{e}_i \times \mathbf{V}_{ik}}{U\mathcal{R}_i(\phi)} b_{\epsilon}(\phi) d\phi \right\rangle, \quad (\text{S16})$$

$$\frac{\mathbf{e}_i \times \mathbf{V}_{ik}}{U\mathcal{R}_i(\phi)} = \frac{-\mathcal{D}_i(\phi) \sin \phi + \overline{\mathcal{O}}_i(\phi) \cos \phi}{U}. \quad (\text{S17})$$

$b_\epsilon(\phi)$ is a parametric representation of an optic blind spot. The operand $\langle \cdot \rangle$ denotes a normalization that ensures the resulting function is of order $O(1)$. Conceptually, it functions like a min-max normalization on the amount of rotation for a single time step.

This model is made dimensionless by using a as the spatial unit and a/U as the temporal unit. The use of these spatial and temporal units creates a parameter space with the weights of each component of the angular velocity $(k_\odot, k_\parallel, k_\eta)$. Some Results can be seen on Fig. S6.

S6 Observed phases

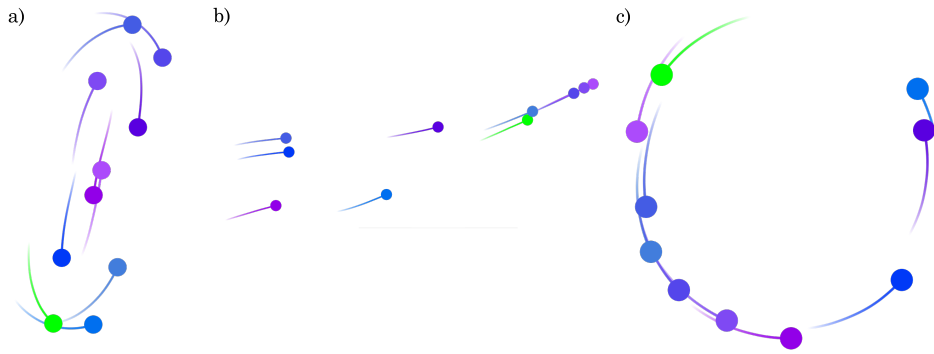


Fig. S6. Illustration of the phases observed for the raw original model [1]. We observe three phases when varying the other two parameters: (a) Swarming ($k_\odot = 0.15$, $k_\parallel = 0$) (see Svideo 13); (b) Schooling ($k_\odot = 0.07$, $k_\parallel = 0.3$) (see Svideo 14); (c) Milling ($k_\odot = 0.07$, $k_\parallel = 0.05$) (see Svideo 15).

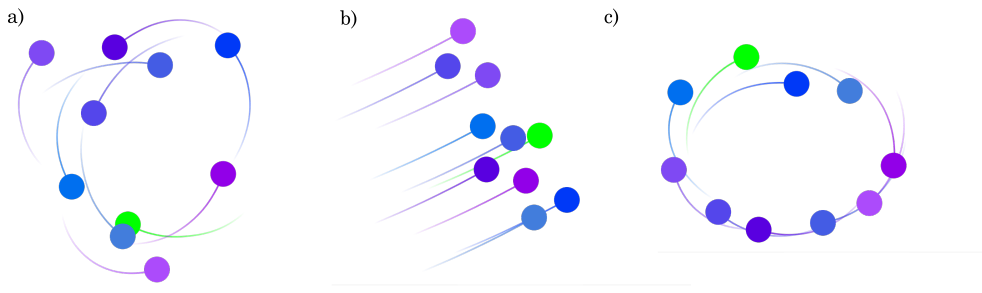


Fig. S7. Illustration of the phases observed for the visual model supplemented only with the avoidance. We observe three phases when varying the other two parameters: (a) Swarming ($k_\odot = 0.07$, $k_\parallel = 0$) (see Svideo 16); (b) Schooling ($k_\odot = 0.15$, $k_\parallel = 0.3$) (see Svideo 17); (c) Milling ($k_\odot = 0.07$, $k_\parallel = 0.05$) (see Svideo 18).

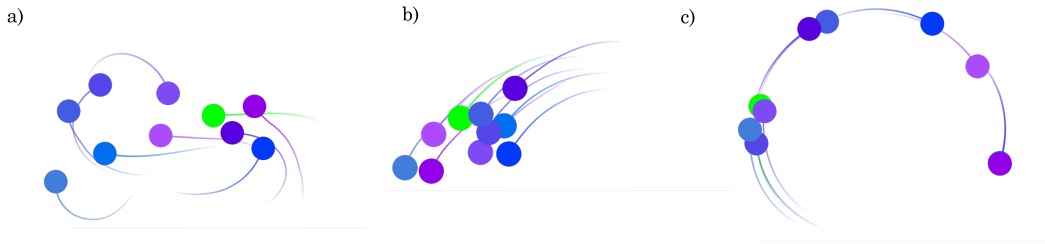


Fig. S8. Illustration of the phases observed for the visual model supplemented only with the avoidance and the visual anchor. We observe three phases when varying the other two parameters: (a) Swarming ($k_{\odot} = 0.3$, $k_{\parallel} = 0$) (see Svideo 19); (b) Schooling ($k_{\odot} = 0$, $k_{\parallel} = 0.3$) (see Svideo 20); (c) Milling ($k_{\odot} = 0.07$, $k_{\parallel} = 0.3$) (see Svideo 21).

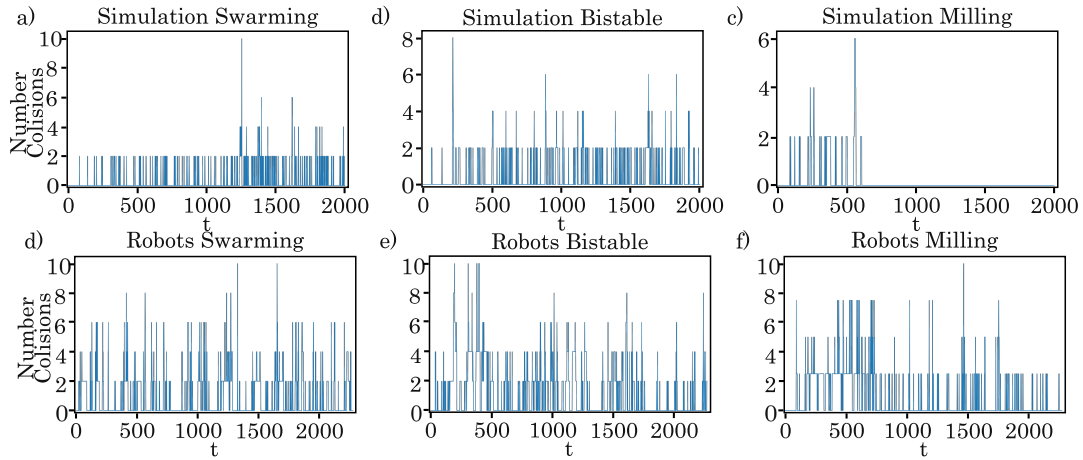


Fig. S9. Number of collisions during the evolution with a set of parameters (a) Swarming simulation; (b) Schooling simulation; (c) Milling simulation; (d) Swarming robot-in-the-loop; (e) Schooling robot-in-the-loop; (f) Milling robot-in-the-loop.

S7 Occlusion Analysis

References

- [1] Castro D, Ruffier F and Eloy C 2024 *Phys. Rev. Res.* **6**(2) 023016 URL <https://link.aps.org/doi/10.1103/PhysRevResearch.6.023016>

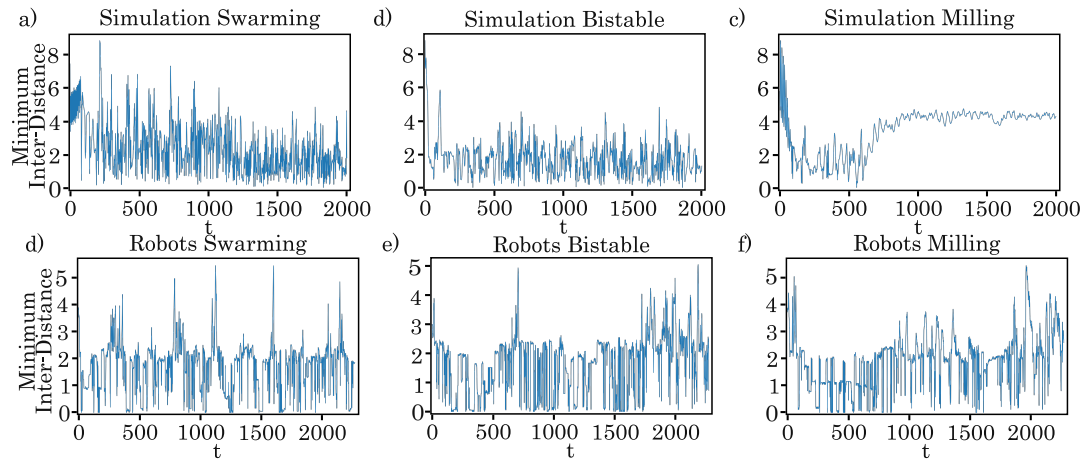


Fig. S 10. Minimum inter-distance found during the evolution with a set of parameters (a) Swarming simulation; (b) Schooling simulation; (c) Milling simulation; (d) Swarming robot-in-the-loop; (e) Schooling robot-in-the-loop; (f) Milling robot-in-the-loop.