DOI: 10.1111/cgf.70225

COMPUTER GRAPHICS *forum*
Volume 44 (2025), number 6, e70225

Herds From Video: Learning a Microscopic Herd Model From Macroscopic Motion Data

Xianjin Gong,[1] James Gain,[2] Damien Rohmer,[1] Sixtine Lyonnet,[1] Julien Pettré[3] and Marie-Paule Cani[1]

[1]LIX, Ecole Polytechnique/CNRS, IP Paris, Palaiseau, France
{xianjin.gong, Damien.Rohmer, sixtine.lyonnet, marie-paule.cani}@polytechnique.edu
[2]University of Cape Town, Cape Town, South Africa
jgain@cs.uct.ac.za
[3]Univ Rennes, Inria, CNRS, IRISA, Rennes, France
julien.pettre@inria.fr

Abstract

We present a method for animating herds that automatically tunes a microscopic herd model based on a short video clip of real animals. Our method handles videos with dense herds, where individual animal motion cannot be separated out. Our contribution is a novel framework for extracting macroscopic herd behaviour from such video clips, and then deriving the microscopic agent parameters that best match this behaviour. To support this learning process, we extend standard agent models to provide a separation between leaders and followers, better match the occlusion and field-of-view limitations of real animals, support differentiable parameter optimization and improve authoring control. We validate the method by showing that once optimized, the social force and perception parameters of the resulting herd model are accurate enough to predict subsequent frames in the video, even for macroscopic properties not directly incorporated in the optimization process. Furthermore, the extracted herding characteristics can be applied to any terrain with a palette and region-painting approach that generalizes to different herd sizes and leader trajectories. This enables the authoring of herd animations in new environments while preserving learned behaviour.

Keywords: animation, behavioural animation

CCS Concepts: • Computing methodologies → Simulation by animation; Physical simulation;

1. Introduction

Simulating the collective motion of herds, packs, schools and swarms of animals and insects has a long history in biology [GLR96], where it is used to conduct *in silico* experiments on animal perception and behaviour, physics [VZ12], where it is used to explore theories of emergence both within and beyond the animal domain, and computer graphics [Rey87], where it is applied to animate animal groupings in film, games and virtual environments. In these domains, the most common strategy is to simulate individual members using a 'microscopic' agent model, such as Boids [Rey87, R*99] or the Social Force Model [LJ14], designed to mimic the core behaviours of cohesion, alignment and collision avoidance exhibited by real animals. Collective behaviour then arises as an emergent 'macroscopic' property of the interaction between individual agents and their environment.

Unfortunately, replicating real animal behaviour using agent models invariably requires careful tuning, based on a deep understanding of the model parameters and their second-order effects on the simulation. One way to circumvent this often lengthy and frustrating trial-and-error process would be to automatically derive model parameters from real motion data [LWPCL15].

An obvious route to achieving this is to extract the position and velocity of individual animals from video frames and then use inverse differential optimization to find agent parameters that induce matching herd behaviour, as in the work of Wolinski *et al.* [WGO*14]. However, this is both computationally costly, and made difficult by a lack of accurate data on animal behaviour. Given that very few curated datasets exist compared to the large number of species moving in herds, our idea is to work from accessible non-curated video footage. However, this is again challenging and error

prone because of the hundreds of individuals in a typical herd combined with the hundreds of frames in even relatively short video clips. In addition, animals in a herd often move shoulder to shoulder making them difficult to distinguish.

In this work, we present a solution for learning microscopic agent models from video clips of herds, which is robust to dense herds where individual animals cannot be distinguished. We achieve this by extracting macroscopic gridded density and velocity data from the videos, using colour matching and dense optical flow. We assume video shot from an aerial perspective, typically using a drone, and compensate for camera motion using inter-frame region tracking to establish a fixed coordinate system.

The extracted density and velocity fields are used as the basis for Stochastic Gradient Descent optimization of the parameters of a microscopic herd model, loosely inspired by Boids [Rey87]. We found that changes to the basic alignment, cohesion and avoidance rules of Boids were necessary to improve differentiability and align the agent's perception more closely with real herding animals. In addition, we distinguish between leader and follower behaviour, enabling animals to change their role over time, depending on the arrangement of their perceived neighbours. While a follower's direction of motion is computed from interaction forces, we introduce a new, stochastic model for their speed of motion, which is a function of local density and can be independently learnt from a video clip.

Once extracted, the herd parameters can be transferred to a new setting with different environmental obstacles, herd size and maximum speed, and with different routing for leaders (which can be painted onto the terrain as a navigation field). When generating a new animation, the herd model preserves the local dynamic distribution patterns learnt from the video. To further support authoring, behaviour brushes are provided, enabling users to paint specific learnt behaviour on different parts of the terrain. In this way, the herd can be directed to transition between high-level patterns, such as milling, swarming and schooling, as required.

To summarize, our contributions are as follows:

1. A framework for robustly extracting microscopic herd behaviours from short video clips, by matching macroscopic velocity and density fields.
2. A novel microscopic herd model, supporting differentiable optimization, which distinguishes between leader and follower dynamics, untangles speed from direction of motion and better aligns the forces driving the latter with actual animal perception.
3. Support for authoring new herd animations, directed using a navigation field for leaders and the learnt herd dynamics for followers.

2. Related Work

Animal herds generate visually compelling movement patterns [PGM14, PFO*23], which explains the focus on crowd simulation in computer graphics, beginning with Reynolds' seminal work on Boids [Rey87]. Reynolds' key inspiration is that simple local rules, governing inter-individual interactions, can explain and cause the emergence of large-scale patterns. Other fields, such as physics have taken up this idea for analogous physical systems, such as charged particles in electromagnetic fields [HM98, VCBJ*95, VZ12].

The ongoing crowd and herd research in computer graphics strives to create visual simulations that satisfy two conflicting goals: achieving a close visual resemblance to real crowds and herds, while providing high-level control over authored animations. The field has progressed towards these goals by proposing various macroscopic [TCP06], microscopic [MT01, PPD07, GCC*10] and hybrid models [NGCL09]. Microscopic approaches are more widely used in practice for animation because they generally produce higher-quality individual motion. This general approach has been explored in depth and recent developments are reported by Van Toll and Pettré [vTP21]. Nevertheless, the question of how to configure such simulations, in particular using data sources to generate animations with matching behaviour, remains largely unanswered. Our work falls into this category.

The problem of setting simulation parameters is part of the more general challenge of effectively authoring crowd animations, for which Lemonari *et al.* [LBC*22] provide a useful survey. In this regard, our primary objective is to learn collective animal behaviour from video examples, as well as to control the global trajectory of the resulting simulated herds.

Along similar lines, methods exist to imitate pedestrian trajectories [LCL07, FR12, CC14], or to replicate how humans interact locally in crowds [LCHL07, JCP*10]. While these approaches focus specifically on human crowds, some work has also been carried out on herds or swarms of animals and insects [LWPCL15, RWJM16, XYWJ20]. However, our aims differ: We seek to process low-quality video data and extract an individual model while reproducing the macroscopic behaviour of animals of any species. Of course, recent developments in deep learning have led to the possibility of directly modelling human trajectories and reproducing their characteristics, linked to biomechanics and social interactions [AGR*16, GJFF*18, AHP19, YMW22]. These deep models focus particularly on predicting human trajectories, which can be reduced to a simulation problem. Importantly, their reliance on sufficient high-quality training data is a limiting factor, with some resorting to synthetic data [RLBP*23]. The ability of those approaches to capture real social and interaction-driven behaviours is also questioned [SBK*22].

Finally, we note that our work is not a first attempt to model collective animal motion from data. In the field of biology, data-driven modelling is employed to study collective behaviour [CLN*14, HR16]. While these models collect response functions based on real data, the recording conditions are highly constrained, the herd sizes limited and the issue of animation control is not addressed.

Compared to the considerable body of crowd research, which we have only touched on lightly, the specific focus of our approach is extracting collective behaviour from single video clips of short duration for any animal species exhibiting herding, in such a way that the extracted and source motion align visually.

The work of Courty and Corpetti [CC07] is positioned closest to ours. They explore the video-driven parameterization of a Social

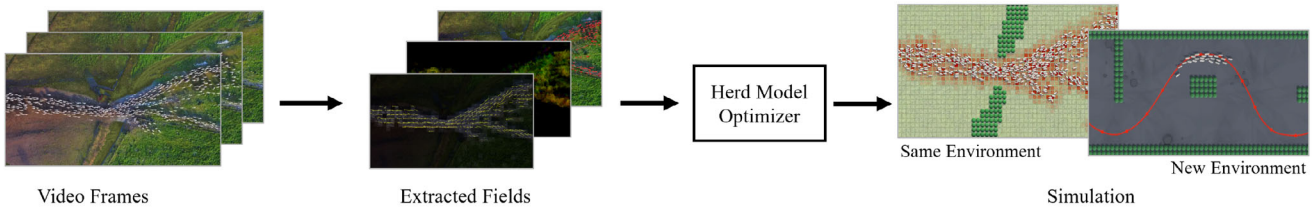

Figure 2: *Learning and authoring pipeline. Starting with frames from video clips, grid-based density and velocity fields are extracted. This data is used to optimize our herd model with perception-related force parameters. Finally, the learnt herd can be deployed either in an environment that matches the source clip or a new user-defined environment.*

Force model. However, they employ synthetic videos and do not focus on preserving emergent behaviours, such as herding.

In conclusion, the framework presented here is the first to offer a practical solution to the problem of analysing animal herds in short video clips and then synthesizing analogous behaviour applicable to new animation settings.

3. Overview

Our method takes as input an overhead video clip of a few seconds duration depicting an animal herd in motion. Such videos are commonly captured by drones or low-altitude aircraft. The animal herd may include a large number of animals—a few hundred in our examples—including situations where the animals are walking shoulder to shoulder, which makes the detection and tracking of individual animals challenging. In the following, we assume that the herd is moving over flat ground, and that there is sufficient contrast in colour between the animals, the background and static obstacles, so as to classify these elements into three distinct categories, using standard vision-based techniques.

The objective of this work is twofold. First, we aim to capture the collective behaviour of animals faithfully, as observed in the reference video. Second, we want to enable the transfer of this behaviour to different contexts, such as a different number and initial arrangement of animals, different animal trajectories and different distributions of environmental obstacles. To this end, we propose an extended Boids model, featuring two contributions: (i) an implementation of perception for simulating the collective behaviour of large-scale herds (with an analysis of its impact), and (ii) trajectory authoring via role-switching that preserves learnt behaviours

Role-switching was previously introduced by Hartman and Benes [HB06], but our approach differs in scope and implementation. Their focus is on reproducing the specific 'leader game' behaviour of birds and they do not consider the earthbound restrictions of terrestrial animals. As such, leaders undergo procedural acceleration and deceleration in 3D space with respect to their herds, but do not lead the group along a pre-defined trajectory or towards a target position. In our case, our terrestrial agents follow navigation fields when dynamically classified as leaders, thus serving the desired navigation without disturbing the process of behavioural learning from video. Notably, unlike many previous herd models, followers automatically connect to leaders via alignment and cohesion forces, eliminating the need to define an additional force specifically for this interaction.

Figure 2 summarizes our processing pipeline. First, the input video is processed in order to extract density and velocity field data. These fields provide a macroscopic description of the herd motion in the video and we fit a microscopic vision-aware force-based agent model to match them greedily, making the herd model reusable in a different context.

Subsequently, the user can create a new scene with obstacles and a trajectory field for leader agents, and instantiate a new herd, while re-using the herd parameters learnt from one or more videos, thereby retrieving their collective behaviour.

The remainder of the paper is organized as follows: Our method for extracting macroscopic fields for any input herd video clip is presented in Section 4. Then, Section 5 details the parameters of our new microscopic herd model. The optimization algorithm is described in Section 6. We finally present our validation and authoring results in Section 7. In particular, we show that our optimized microscopic herd model is able to capture the high-level macroscopic behaviours of the real herd in the input clip, expressed through several common metrics including polarity, angular momentum and aspect ratio relative to the primary movement direction, while being transferable to new initial conditions, herd sizes, target trajectories and environmental obstacles (see Figure 1).

4. Macroscopic Fields From Video Clips

In this section, we present the processing applied to the input video in order to extract the macroscopic fields used to optimize our agent-based model. Although we describe the steps used to compute these fields for the sake of completeness, we emphasize that the specific video processing and annotation techniques used here can be modified independently, without affecting the learning process or the overall effectiveness of our method.

Footage stabilization. Real-video footage of animal herds typically contains camera motion, rotation or zoom over time, since it is usually captured by drones or small aircraft. Our initial preprocessing step addresses this by stabilizing the footage, creating a global coordinate system that allows for consistent position comparisons across different frames.

Figure 1: *Our method can simulate individual agents to replicate herd behaviour learnt from a video containing many animals. [Left and middle] The original video (lower-right) and our simulation (upper-left), optimized to fit the macroscopic density and velocity fields over a coarse grid. [Right] An authored simulation in which a herd transitions between the two illustrated behaviours, featuring narrow and broad formations.*

To this end, the user manually specifies two rectangular regions (in two different, arbitrary video frames) to be identified as being in correspondence. We then compute an automatic region tracking over all frames of the video, using the *Channel and Spatial Reliability Tracking* (CSRT) algorithm [LVCZ*18], and apply time window smoothing over nine frames to remove jitter. Since our goal is to extract averaged macroscopic values for the herd distribution and motion, we employ a low resolution spatial grid G_{xy} containing $N_{grid} \times N_{grid}$ cells for storing these values. The change of coordinates from the video frame to the global coordinate system is expressed as follows: For each frame, the CSRT algorithm provides the current region centre $\mathbf{c}$, two unit and orthogonal vectors $\mathbf{u}_1$ and $\mathbf{u}_2$ representing, respectively, the two principal axes of the tracked rectangular region and a scaling factor $s > 0$, taking into account the change in length related to the zoom. The global coordinates $\mathbf{p} \in \mathbb{R}^2$ can then be computed from the local frame $\mathbf{p}_{frame}$ as

$$\mathbf{p} = s\big((\mathbf{p}_{frame} - \mathbf{c}) \cdot \mathbf{u}_1\,,\ (\mathbf{p}_{frame} - \mathbf{c}) \cdot \mathbf{u}_2\big)/N_{grid}. \qquad (1)$$

Computing macroscopic fields. Three fields are then computed on each grid cell, namely, the density of the animals ρ_{xy}, the spatial velocity $\mathbf{V_{xy}}$ and the velocity variance $\mathbf{S_{xy}}$. Note that all fields are computed for the video frame at the highest resolution before being transferred to the global coordinate frame, using Equation (1). The animal density is obtained by applying colour-based segmentation of the input video, and counting all body-pixels falling within a given grid cell G_{xy}. The velocity estimation is obtained from a dense optical flow using the Gunnar–Farnebäck algorithm [Far03]. For a given grid cell G_{xy}, the mean velocity is denoted by $\mathbf{V_{xy}}$, and associated variance by $\mathbf{S_{xy}}$. Finally, users are able to manually paint an additional annotation into the global grid to identify regions with impassable obstacles.

5. A Differentiable Herd-Agent Model

In choosing a suitable agent model, we were motivated by three considerations: conceptual and computational simplicity, amenability to parameter optimization and realistic emergent behaviour.

In this regard, the Boids model [Rey87] strikes the best balance among these criteria. An individual Boids agent employs an aggregation of simple forces for cohesion, aggregation and separation. Each component force considers nearby agents within a pre-defined separate radial distance. This computational simplicity is appealing because it enables faster optimization and simulation of larger herds, as compared to more complex candidates, such as reciprocal collision avoidance [vdBPS*08] and social force models [LJ14]. It is also more conceptually accessible, which makes fine-tuning of learnt behaviour easier for artists and game developers, should this be necessary.

Furthermore, although the original model cuts off force contributions at various distance thresholds, creating a derivative discontinuity at these boundaries, this can be rectified by introducing a Gaussian distance weighting that ensures the differentiability required by many optimization schemes.

Finally, in terms of realism, it has been shown that Boids-like models can perform comparably on vorticity, and separation and re-grouping measures of emergent behaviour with dense herds [WGO*14] and can also be configured for a spread of motion patterns [CLN*14], such as milling (circular movement around a central core), schooling (coordinated movement with close alignment) and swarming (chaotic movement but with a dominant overall direction).

While the Boids model provides a sound starting point, in order to support both learning and authoring, we introduce a number of modifications: a division of agents into leader and follower roles to control herd navigation, Gaussian weighting to enable differentiable optimization, an improved perceptual model that incorporates occlusion and separate control over the speed and magnitude of agent motion.

5.1. Leaders and followers

During both the learning and subsequent authoring phases, it is necessary to have a mechanism for directing the herd as a whole. Inspired by Herbert-Read's study [HR16] of coordinated movement in real groups of animals, we address this by sorting the agents, at each simulation step, into two classes as follows: leaders and followers (as shown in Figure 3). Note that an agent can switch dynamically from one class to the other during the simulation.

We classify agents as leaders if up to two neighbours appear in their local visual field. Leaders ignore the Boids force calculations and instead follow either the velocity field $\mathbf{V_{xy}}$ extracted from video in the case of optimization (see Section 6) or a navigation field provided during authoring (see Section 7.3). All other agents are clas-

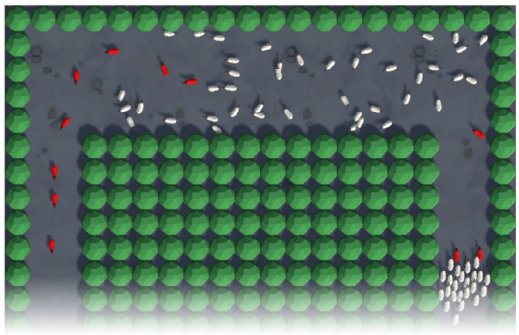

Figure 3: *Agent classification: Red agents are designated as leaders, since the number of neighbours within their perceptual range is ≤2. They follow the navigation field. White agents are followers and they obey our new herd-agent model.*

sified as followers and obey our force calculations. This allows the best of both worlds: path navigation initiated by the leaders, while followers comply with the learnt herd behaviour.

5.2. Updating directional velocity

To ensure our model captures animal behaviour in real herd videos, we depart for previous microscopic crowd models in how an agent's velocity is computed from the set of social forces.

Specifically, we decouple the speed and direction of motion. An agent's updated direction is computed for each timestep based on the previous direction and an integration of applied forces. An agent's updated speed is based on a stochastic model, which from our experiments provides a closer match to actual animal behaviour. Another advantage of this decoupling is the control it provides over the speed of the herd when the learned model is re-used, a feature difficult to achieve with standard entangled microscopic models.

Let us detail this change: For each agent a_i at frame k for timestep Δt, we update velocity component-wise, as follows:

$$\mathbf{V_i^{k+1}} = v_i^{k+1}\mathcal{N}(\mathbf{V_i^k} + \Delta t\mathbf{F_i^k}), \tag{2}$$

where the speed term v_i^{k+1} is a smoothed version of Brownian particle motion with friction, which will be detailed in Section 6.2, and $\mathcal{N}(\mathbf{V}) = \mathbf{V}/\|\mathbf{V}\|$ is the normalized version of V.

Here, the normalized direction term $\mathcal{N}(\mathbf{V_i^k} + \Delta t\mathbf{F_i^k})$ is dictated by the total force $\mathbf{F_i^k}$ exerted by neighbouring agents and environmental obstacles. This force is a weighted sum of a cohesion force $\mathbf{FC_i^k}$, alignment force $\mathbf{FA_i^k}$, agent avoidance force $\mathbf{FV_i^k}$ and obstacle avoidance force $\mathbf{FO_i^k}$, as follows:

$$\mathbf{F_i^k} = \Omega \cdot \left(W_{FC} \cdot \mathbf{FC_i^k} + W_{FV} \cdot \mathbf{FV_i^k} + W_{FA} \cdot \mathbf{FA_i^k} + W_{FO} \cdot \mathbf{FO_i^k}\right) \tag{3}$$

where W_{FC}, W_{FV}, W_{FA} and W_{FO} are weights on the component forces, to be found by optimization. We also introduce a sensitivity modifier Ω (10 in our case) on the total force, which controls how rapidly agents respond to their neighbours, but which is not included in the optimization.

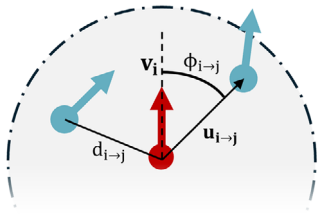

Figure 4: *Perceptual terms: Distance to neighbour $d_{i\to j}$ and bearing angle $\phi_{i\to j}$ are defined by the relative position of the current agent a_i and its neighbour a_j.*

5.3. Perception-related directional forces

Before delving into the details of the component forces, it is first necessary to define our perception mechanism, which represents a departure from the standard Boids approach. Instead of including all neighbouring agents closer than a threshold distance, we exclude agents that are occluded, which moves our model closer to actual animal perception. We cast a ray from the focal agent to a potential neighbour and reject the neighbour if there are any intersected obstacles (either dynamic or static) between them. In the results section, we will show that, not only does this restriction make our model more realistic, but it also helps speed up the optimization process. Furthermore, we introduce an angle-based term, which better models the field-of-view of real animals.

To be specific, each component force depends on separate distance and direction terms (see Figure 4). We record the distance $d_{i\to j}$ from a focal agent a_i to its neighbour a_j and the bearing angle $\phi_{i\to j}$ between the forward unit direction vector $\mathbf{v_i}$ of a_i and the unit vector $\mathbf{u_{i\to j}}$ pointing towards a_j. A bearing angle $\mathbf{u_{i\to o}}$ for an obstacle Obs_o can be defined on a similar basis.

Each component force is governed by a pair of weights calculated using a separate Gaussian fall-off function for radial WR_F^{ij} and angular WA_F^{ij} perception, with range limits R_F and Φ_F, respectively, as follows:

$$WR_F^{ij} = exp\left(-0.5 \cdot \frac{d_{i\to j}^2}{R_F^2}\right), WA_F^{ij} = exp\left(-0.5 \cdot \frac{\phi_{i\to j}^2}{\Phi_F^2}\right). \tag{4}$$

In effect, there are eight additional optimizable parameters for a total of 12, two for each of cohesion (R_{FC} and Φ_{FC}), alignment (R_{FA} and Φ_{FA}), agent avoidance (R_{FV} and Φ_{FV}) and obstacle avoidance (R_{FO} and Φ_{FO}). This means that that an agent can have different sensitivity (field of view and range of vision) depending on the type of force.

Finally, the component forces are defined as follows:

$$\mathbf{FC_i^k} = \sum_{a_j \in Neighbour(a_i)} WR_{FC}^{ij} \cdot WA_{FC}^{ij} \cdot d_{i\to j} \cdot \mathbf{u_{i\to j}}$$

$$\mathbf{FV_i^k} = \sum_{a_j \in Neighbour(a_i)} -WR_{FV}^{ij} \cdot WA_{FV}^{ij} \cdot d_{i\to j} \cdot \mathbf{u_{i\to j}}$$

$$\mathbf{FA_i^k} = \sum_{a_j \in Neighbour(a_i)} WR_{FA}^{ij} \cdot WA_{FA}^{ij} \cdot \mathbf{v_j}$$

$$\mathbf{FO_i^k} = \sum_{Obs_o \in Neighbour(a_i)} -WR_{FO}^{io} \cdot WA_{FO}^{io} \cdot d_{i\to o} \cdot \mathbf{u_{i\to o}} \quad (5)$$

5.4. Updating speed

Although our model primarily governs changes in the direction of herd agents, in the interests of realism, it is important to also control changes in their speed. To replicate the source behaviour of the herd as closely as possible, we use an Ornstein–Uhlenbeck (OU) process proposed in Uhlenbeck and Ornstein [UO30], which emulates Brownian particle motion with friction. This makes use of a global mean and variance in speed derived from the video data or supplied by the user during authoring.

At frame k, we define the speed of agent a_i as v_i^k, the herd's mean speed as v_μ and the standard deviation of herd speed as v_σ. Then, according to an OU process, the change in speed of an agent is

$$v_i^{k+1} - v_i^k = -\theta(v_i^k - v_\mu)\Delta t + v_\sigma\sqrt{\Delta t}N(0,1) \quad (6)$$

where θ (1 in our case) is the mean reverting rate, $N(0, 1)$ is the random normal distribution, and Δt is the time interval between frames.

This is a relatively straightforward model for speed adjustment. Nevertheless, it is amenable to the incorporation of additional features. For instance, terrain information, such as slope and surface properties (*e.g.* snow, sand and water), could be incorporated into the grid and then used to adjust speed. In addition, speed could be modulated according to local herd density. We leave these improvements to future work.

6. Learning Behaviour via Differentiable Optimization

Let us now describe how our herd-agent model from Section 5 can be tuned to match the macroscopic fields extracted from a video clip (see Section 4). Since our agent model disentangles the directional and speed components of velocity, learning needs to take place in two stages.

6.1. Learning the force parameters for directional velocity

The first learning stage involves tuning the directional force parameters (force weights and perception range limits) via differentiable optimization, using stochastic gradient descent. In total, there are 12 parameters to optimize, three per component force.

As is typical with optimization, the main challenge lies in finding a meaningful error function to minimize. Thanks to our disentangled model for velocity, we can focus on a match for mean direction (*i.e.* normalized velocity) at the individual agent level or a match for direction variance at the cell level, instead of having to achieve both speed and direction (*i.e.* full velocity) simultaneously. In practice, this makes learning much more efficient.

We define the error function for frame k in two parts as follows: First, we define the error in mean direction as the accumulated difference between normalized motion vectors of the simulated agents ($\mathbf{V_i}$) and video target ($\mathbf{U_{xy}}$):

$$Error_{mean}^k = \sum_i \|\mathcal{N}(\mathbf{V_i^k}) - \mathcal{N}(\mathbf{U_{xy}^k})\|, \quad (7)$$

where the simulated agent a_i is located in cell xy of the grid, and N is again the vector normalization function.

Second, we calculate the error in directional variance by summing over all individuals the angular variance of the individual's direction from that of the cells it inhabits when compared to a cell-based target variance ($\mathbf{S_{xy}^k}$), as

$$Error_{var}^k = \sum_i \|Var_{i\in xy}\big[Angle(\mathcal{N}(\mathbf{V_i^k}), \mathcal{N}(\bar{\mathbf{V}}_{\mathbf{xy}}^{\mathbf{k}}))\big] - (\mathbf{S_{xy}^k})\|, \quad (8)$$

Here, agent velocity is calculated per time-step in the expected fashion:

$$\mathbf{V_i^k} = \mathbf{V_i^{k-1}} + \Delta t \cdot \mathbf{F_i^{k-1}}. \quad (9)$$

What remains is to define a target motion vector $\mathbf{U_{xy}^k}$ that blends the macroscopic velocity and density fields extracted from video (see Section 4), neither of which suffices alone.

From the velocity field, we extract the normalized mean velocity $\mathcal{N}(\mathbf{V_{xy}^k})$ of a particular grid cell. As shown in Section 7.2.3, it is not possible using mean velocity alone to fully capture motion from a video. Indeed, the observed changes of density in the input video are also a strong indicator of the direction of motion, and we take them into account as follows. We define a density matching vector $\mathbf{D_{xy}^k}$ using the gradient of the difference between the simulated (ρ'_{xy}) and target (ρ_{xy}) density fields:

$$\mathbf{D_{xy}^k} = -\nabla(\rho'_{xy} - \rho_{xy}). \quad (10)$$

The intuition is that positive or negative values represent, respectively, an over- or under-supply of agents in a grid cell of the simulation relative to the target. The inverse gradient thus supplies a greedy direction for agents to follow so as to correct this imbalance (as shown in Figure 5).

The final target direction $\mathbf{U_{xy}^k}$ is defined as a weighted linear combination of this normalized density gradient and the normalized mean velocity:

$$\mathbf{U_{xy}^k} = \beta\mathcal{N}\big(\mathbf{D_{xy}^k}\big) + (1-\beta)\mathcal{N}\big(\mathbf{V_{xy}^k}\big), \quad (11)$$

where the weight β (we used values between 0.5 and 0.8) is used to prioritize either density or velocity field matching. Prioritizing density yields more accurate position distributions, at the risk of losing alignment. Prioritizing velocity may improve movement coordination, but at the expense of positional accuracy.

The error in mean direction $Error_{mean}$ is usually sufficient for finding parameters based on video inputs. However, convergence of the optimization is not guaranteed in cases where the movement of agents in a cell is fully random, as this results in an unstable $\mathbf{V_{xy}^k}$ for error calculations. We can set $\beta = 1$ to match on the

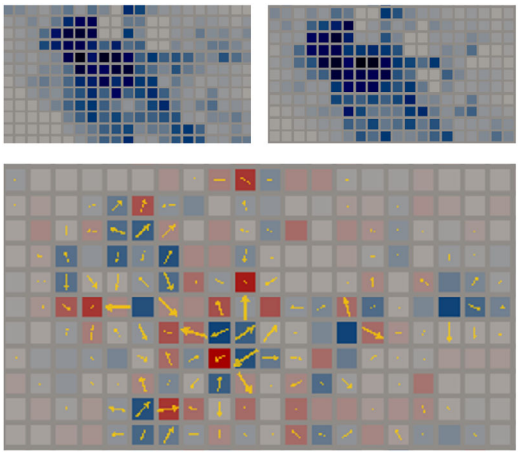

Figure 5: *Density difference field. [Upper left] simulated density field ρ'. [Upper right] extracted density field ρ. [Bottom] difference field and target direction, with blue indicating that the simulated density is too high, and red that it is too low. The arrows show the direction of motion for correction.*

density field alone, but, because alignment is never optimized towards a steady target, it is still difficult to reach convergence. During experiments, we noticed that the variance in an agent's velocity direction is strongly influenced by the alignment force, while avoidance and cohesion forces act to adjust the spacing between agents. Accordingly, we use $Error_{var}$ to optimize alignment parameters separately, such that the alignment force can be fixed with or without information from the velocity field. Using this error term, we can even extend our application to learn collective behaviour from a single image in cases where individual agents are randomly aligned. A concrete example will be provided in the next section.

During learning, we exclude agents identified as leaders from the optimization process. Instead, they directly follow the mean velocity field with no deviation. It is worth noting that the classification of leaders and followers occurs on a per-frame basis, so that these roles can change dynamically over the course of video matching.

One slight complication is that agents may need to be removed or added during optimization. First, in many video clips, animals enter or exit from outside the camera's field of view. Second, if the agent parameters do not align well, then for later timesteps, the simulation will become ever more divergent, hampering convergence. To detect these cases, we calculate density matching error as the sum of absolute differences between the simulated and target density fields. If this exceeds a pre-defined threshold, then we remove agents from the densest cells and add agents to the sparsest cells.

If we treat the agent model as a function, with agent parameters as input and the collective herd behaviour as output, it can be characterized as non-injective (many inputs map to a single output). This lack of injectivity may give rise to flat (zero gradient) regions in the optimization landscape. To avoid becoming trapped, we use a fixed update step size and random update direction when a zero error gradient is encountered.

6.2. Extracting speed parameters

Our model is now able to learn force parameters that best match the salient direction of motion in a video clip. Next, we seek to match the speed of motion. This is achieved in two steps, as follows.

First, we compute the mean and standard deviation of the speed of animals in the video. Although we do not identify individual animals in the source video, we can, nevertheless, infer these statistics using the dense optical flow of body-pixels (see Section 4.) We begin with an assumption of relatively homogeneous body sizes within a given video, so that each animal occupies roughly the same number of image pixels N_{bp}. Then we calculate the mean and standard deviation for the speed of body-pixels, v_μ^{bp} and v_σ^{bp}, respectively. Finally, we obtain the animal-specific statistics: $v_\mu = v_\mu^{bp}$ and $v_\sigma = v_\sigma^{bp}/\sqrt{N_{bp}}$, which are fed into Equation (6) to generate the stochastic speed of the agents.

Second, we account for speed-density correlations by constructing a distribution function of speed with respect to density, based on the extracted density field. The change in speed provided by the OU process can then be adjusted according to the local herd density. This approach significantly improves the local alignment between the derived and observed speed patterns.

7. Results

In this section, we present a variety of results, including validations based on biological measures of emergent behaviour, ablation studies for our implementation choices and case studies in authoring new herd animations based on learnt behaviours.

For these experiments, we used one image and six input video clips with durations ranging between 3 and 13 s. The still image is labelled as 'seals' and is a drone photo of a seal herd distributed on an ocean shoreline.The video clips are labelled as 'ants-chaotic', 'ants-circling', 'sheep-broad', 'sheep-narrow', 'sheep-multi-lines', and 'duck-milling'. In total, they encompass four different animal species and a variety of emergent behaviours.

The two ant clips showcase, respectively, uncoordinated quasi-random motion ('ants-chaotic') and a coordinated circling motion ('ants-circling') around a central object. The three sheep clips represent a herd advancing along a wide front with varied velocity and behaviour, such as grazing ('sheep-broad') and a herd migrating in a long but narrow-fronted arrangement ("sheep-narrow"). Anisotropic behaviour is apparent in the third clip ('sheep-multi-lines') where sheep evidence different spacing from neighbours ahead and behind as compared to left and right. This leads to different dividing and merging behaviours when they meet obstacles. Finally, we have the 'duck-milling' clip which captures self-rotation behaviour without external factors forcing that.

All experiments were performed on a PC equipped with an AMD Ryzen 7 9800X3D processor (4.70 GHz) and an NVIDIA GeForce RTX4060Ti. Video data extraction was implemented in Python, while Unity with C# was used for building the virtual environments and handling physical interactions, such as collision detection. The herd simulation and the parameter optimization updates were implemented in C++, via a DLL plugin.

Our method is not optimized, but nevertheless achieves computational performance of approximately 0.2 s per frame for the preprocessing phase, in which macroscopic density and velocity fields are extracted from video. For herds of size 500, 1000 and 1500 agents, the optimization process takes 0.1, 0.3 and 1.2 s per frame, respectively, and the simulation takes 0.02, 0.05 and 0.3 s. Note that the time required for optimization and simulation may vary depending on local density, as ray casting is performed for each neighbour to test for occlusion. Higher densities result in longer simulation times.

Data availability statement. Data sharing is not applicable to this article as no datasets were generated or analysed during the current study. The source code can be found at https://github.com/XavierScor/HerdFromVideo.

7.1. Validation

7.1.1. *Ground-truth convergence*

Before operating on real video data, we first performed a pilot study to test whether our macroscopic approach could converge accurately to a known simulated ground truth. We created a test environment, manually set the parameters of our agent model, ran a simulation and then computed gridded velocity and density fields in a format that matched our video extraction. We then performed several optimization runs with different initial parameters to test for convergence between learnt and ground truth agent parameters.

All optimized parameters are expressed in a normalized range in [0, 1]. The cumulative error over the 12 parameters thus lies in [0, 12]. We report an average accumulated error of 0.72, where the averaging is performed over the results obtained from different initial conditions, or, in other words, an average error of 6% per parameter.

Figure 6 shows side-by-side frames at $t = 1.9$ s from the ground truth and the simulation using the learnt parameters.

7.1.2. *Measures of emergent behaviour*

It is important in any assessment of herd behaviour to carefully consider the emergent (macroscopic) properties. In this regard, the typical emergent measures used in the physics, biology and graphics literature are as follows:

- Density. How closely grouped or spread the individuals of the herd are relative to their neighbours [PFO*23].

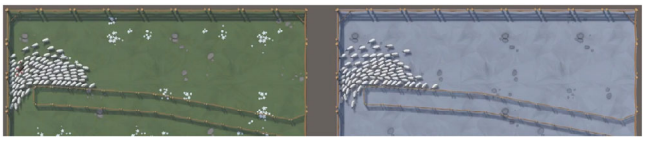

Figure 6: *Convergence to a manual ground truth. [Left] The target herd with manually configured parameters. [Right] A herd simulated using parameters learnt from the target on the left via optimization.*

- Speed. How fast the herd moves over time, usually measured by the mean and standard deviation of the magnitude of agent velocity.
- Shape. What is the shape of the convex hull (or some other enclosing shape) of the herd in relation to its prevailing direction of motion? A typical measure is the aspect ratio of the motion-aligned bounding box fitted around the members of the herd [PFO*23, HH11]. The lowest aspect ratio occurs when a herd files behind a single leader. Conversely, the highest aspect ratio involves all members moving abreast in a row.
- Polarity. How closely aligned the members of the herd are in terms of their movement direction [CLN*14, PCHFJ16]. It is commonly measured by

$$P(t) = 1/N \left\| \sum_{i=1}^{N} \mathcal{N}(\mathbf{V_i}(t)) \right\|, \tag{12}$$

where $P(t) \in [0, 1]$ is the polarity (or sometimes order [VZ12]) at timestep t, N is the number of individuals in the herd and $\mathcal{N}(\mathbf{V_i}(t))$ is the normalized movement vector of an individual.
- Angular momentum. This indicates the extent to which the herd circles around a common centre, and is often used as an indication of milling behaviour [CLN*14].

$$M(t) = 1/N \left\| \sum_{i=1}^{N} \mathcal{N}(\mathbf{R_i(t)}) \times \mathcal{N}(\mathbf{V_i(t)}) \right\|, \tag{13}$$

where $\mathbf{R_i(t)}$ is the vector connecting the agent position to the centre of the rotational behaviour. By default, we consider this to be the centre of the image.
- Spatial organization. Certain animals adopt characteristic lattice-like formations while moving collectively. For instance, birds have a tendency to avoid flying nose to tail in formation. It is very common to use a radial density histogram to capture this [PFO*23, HR16, PGM14, PCHFJ16]. This shows the expected density of neighbours around an individual, plotted as a function of neighbour distance and angular placement relative to the direction of motion.
- Overall motion pattern. High-level descriptive terms are often used to characterize the global movement pattern of herds. These include milling (moving in a circular pattern around a central core), schooling (members act with strong coordination and high polarity moving in a single dominant direction) and swarming (individual members move chaotically but nevertheless move in a collective direction). It is possible to identify such motion patterns using a suite of other measures. However, these terms remain rather nebulous and ill-defined.

For the purposes of validation, we handle these measures in different ways. Density (along with velocity) is integral to our learning process and so easily evaluated using the density component of the optimization error function (Equation 7). Shape, polarity and angular momentum statistics can be derived from the velocity and density fields of the source and simulation and provide independent verification of match quality. Finally, the overall motion pattern is a qualitative rather than quantitative attribute, which is best assessed by directly examining video outputs (see Figures 1 and 17).

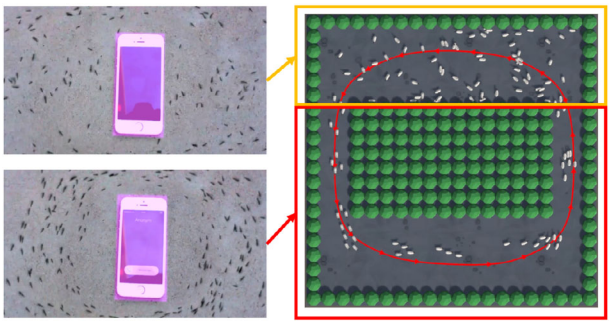

Figure 17: *Authoring an anthill. A counterclockwise guidance arrow is drawn around a large square obstacle and swarming (red rectangle) and milling (blue rectangle) behaviours are assigned from corresponding video clips (on the left).*

Speed was directly measured from the extracted macroscopic fields as a function of density (Section 6.2) so it matches the input by construction. We therefore excluded it from the validation.

Unfortunately, the remaining measure, namely spatial organization, cannot be usefully assessed in our framework, because it would require the ability to detect individual animals in the source video. We will directly show the result by comparing simulated and target frame images.

We conducted our validation experiments by dividing each of the five video clips in half. The first half was used to learn agent parameters, while the second half was used for assessment, by initializing based on the first frame of the second half and then comparing simulation runs, with parameters before and after optimization, against the video on a frame-by-frame basis. For density and velocity, we used a relatively coarse 10×10 grid to accumulate error since this captures behaviour at a larger scale than the finer resolution used for optimization. In this regard, we look for a significant drop in error before and after parameter optimization.

For shape, polarity and angular momentum, we made an assumption of consistency, based on observing generally coherent herd behaviour within each clip. Given this assumption, the standard deviation of each measure can be used as a rough bound on the acceptable mean error, allowing us to claim equivalent behaviour in such case.

7.1.3. *Case study: funnelled sheep herd*

We began validation with the simplest case: 'sheep-narrow'. In this clip, all sheep move in the same direction from right to left with a greater uniform local distribution than other cases. Accordingly, polarization is close to the maximum (at 0.971) and angular momentum to the minimum (at 0.039). To account for the limited camera view, in optimizations, agents moving outside the frame are removed, while new agents are generated in predefined entry cells to maintain a population count consistent with the video.

Table 1 shows our mean simulated error values to be of the same order of magnitude as the STD of the corresponding measures from video, meaning that our simulated result reproduces equivalent behaviour based on polarization and angular momentum. However, in this example, the camera focused on a subset of the herd, leaving the aspect ratio undefined, which is something we address in Section 7.1.5.

Table 1: *Sheep-narrow results.*

		Video measure	Simulated value	Simulated error
Polarization	Mean	0.971	0.958	0.016
	STD	0.015	0.018	N/A
Angular momentum	Mean	0.039	0.028	0.016
	STD	0.024	0.019	N/A

Figure 7: *Simulation based on 'sheep-narrow'. [Left] Target video frame. [Middle] Learnt simulation. [Right] Simulation with random parameters before optimization.*

Table 2: *Duck-milling results.*

		Video measure	Simulated value	Simulated error
Polarization	Mean	0.288	0.212	0.056
	STD	0.065	0.007	N/A
Angular momentum	Mean	0.839	0.916	0.060
	STD	0.042	0.015	N/A

Figure 7 shows the herd simulation before and after optimization. We performed multiple experiments with different random initial starting parameters and they all converged to the same optimum. During the course of optimization, accumulated density (ε_D) and velocity (ε_E) errors decrease from typical starting values of 3.614 and 2.114 to converge on 2.114 and 0.120, respectively.

7.1.4. *Case study: milling ducks*

The 'duck-milling' clip demonstrates the other extreme of polarization and angular momentum (at 0.288 and 0.839, as show in Table 2). This shows a flock of ducks circling around a point in a river. This is an example of a spiral motion without trajectory authoring (as opposed to 'ants-circling', which is guided by a user-specified trajectory around an obstacle in Figure 17). Again, the video frame does not always contain the entire flock, so the aspect ratio cannot be assessed.

To emulate the confines of the river banks, we add obstacles, shown with a grass texture in Figure 8, on the left and right of our simulated scene. Figure 8 [right] shows a flock simulation with ran-

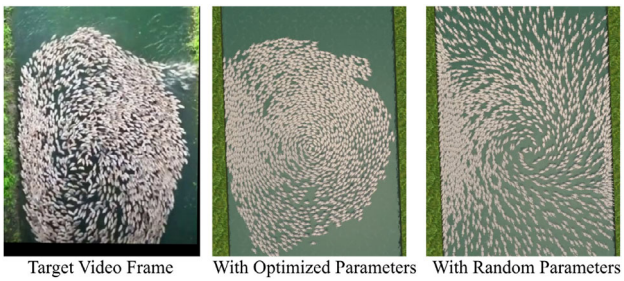

Figure 8: *Simulation based on 'duck-milling'. [Left] Target video frame. [Middle] Learnt simulation. [Right] Simulation with random parameters before optimization.*

Table 3: *Sheep-broad results.*

		Video measure	Simulated value	Simulated error
Polarization	Mean	0.771	0.842	0.069
	STD	0.020	0.008	N/A
Angular momentum	Mean	0.202	0.277	0.068
	STD	0.025	0.015	N/A
Aspect ratio	Mean	0.892	0.889	0.038
	STD	0.074	0.028	N/A

dom parameters set for high avoidance and low cohesion and alignment ($\varepsilon_D = 1.232, \varepsilon_V = 1.266$). Figure 8 [middle] is a frame from simulation after optimization ($\varepsilon_D = 0.585, \varepsilon_V = 0.207$). Here optimization is hampered because the flock is clipped along the bottom edge in the video, leading to incorrect matching in the bottom right of the simulation. Despite this, our model still learns correct milling behaviour as confirmed by the polarization and angular momentum measures.

7.1.5. *Case study: complex sheep herd*

The previous clips have two shortcomings. First, they do not capture the entire herd or flock, making it impossible to evaluate aspect ratio. Second, herd behaviour takes place in relatively unobstructed environments. We overcome this by including the 'sheep-broad' clip, which captures in its entirety a large herd of approximately 1300 sheep crossing a field littered with bushes and rocky outcroppings.

This is a challenging case because the terrain is complex, there is significant interaction between the sheep and environmental obstacles, and there is out-of-frame influence from sheepdogs and humans. Despite these challenges, we achieve simulation error rates within the same order of magnitude as the STD of video results for polarization, angular momentum and aspect ratio (see Table 3).

Although the match between target and simulation is improved through optimization (from errors of $\varepsilon_D = 2.25, \varepsilon_V = 1.26$ down to $\varepsilon_D = 1.96, \varepsilon_V = 0.44$), as shown in Figure 9, the match is not exact due to these extraneous factors. Nevertheless, the simulated herd exhibits the correct overall movement, spatial organization and splitting and merging behaviour around obstacles, without the need for trajectory authoring.

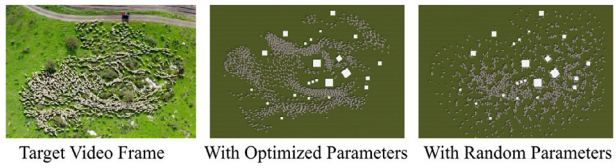

Figure 9: *Simulation based on 'sheep-broad'. [Left] Target video frame. [Middle] Learnt simulation. [Right] Simulation with random parameters before optimization.*

Table 4: *Occlusion ablation results. Columns show the standard deviation in the original source video (except for density and velocity, which are cumulative metrics), and error, with and without perceptual occlusion for various macroscopic properties.*

	Video STD	Without occlusion	With occlusion
Density	N/A	0.09	0.08
Velocity	N/A	30.80	21.67
Polarization	0.04	0.10	0.04
Angular momentum	0.05	0.11	0.07
Aspect ratio	0.04	0.05	0.03

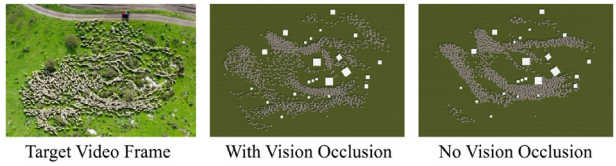

Figure 10: *Assessing the impact of vision occlusion. [Left] A target video frame. [Middle] Learnt simulation* with *vision occlusion. [Right] Learnt simulation* without *vision occlusion.*

7.2. Ablation studies

7.2.1. *Ablation of vision occlusions*

We also undertook a comparison between our agent model with and without vision occlusion. Including perceptual occlusion tends to create more disparity in agent response because even nearby agents can have quite different views of the environment depending on the arrangement of obstacles and other agents. We undertook an ablation study using the 'sheep-broad' video clip and found that incorporating occlusion always reduces error (see Table 4), in many cases from above to below the standard deviation of the original data source. In addition, we have included a side-by-side visual comparison in Figure 10 showing the impact this can make over the course of a simulation.

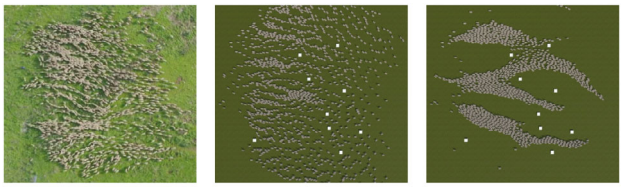

Figure 11: *Simulation based on 'sheep-multi-lines'. [Left] Target video frame. [Middle] Learnt simulation. [Right] Simulation with random parameters before optimization.*

Table 5: *Sheep-multi-lines results.*

		Video measure	Simulated value	Simulated error
Polarization	Mean	0.763	0.922	0.069
	STD	0.044	0.017	N/A
Angular momentum	Mean	0.008	0.277	0.017
	STD	0.020	0.015	N/A
Aspect ratio	Mean	0.423	0.633	0.179
	STD	0.111	0.033	N/A

7.2.2. *Ablation of angular perception*

The literature posits a link between spatial organization, particularly anisotropic separation and the angular range of animal vision [HR16, PGM14]. However, it is unclear what impact this has in our framework, which is an important concern given the additional computation cost involved. Accordingly, we performed an ablation study on the impact of angular perception ranges on anisotropic spatial organization.

We based our experiments on the 'sheep-multi-lines' clip, which shows the formation of linear structures in a large sheep herd, where the nose-to-tail separation between individuals is much smaller than the orthogonal shoulder-to-shoulder separation (see Figure 11 [left]).

First we show that our model can reproduce the collective behaviour based on the polarization, angular momentum and aspect ratio measures (see Table 5 and Figure 11). The learnt angular ranges for avoidance, cohesion, alignment and obstacle avoidance forces are 1.000, 0.271, 0.886 and 0.442, respectively. This means that all neighbours within the perception radius influence avoidance, but neighbours outside bearing angles of 48ºhave little cohesion influence on the focal agent.

Next, we built a new ablation environment (see Figure 12) with three obstacles, no guide trajectories and agents facing upward. In Figure 12 [left], we did not impose any limits on angular range, resulting in a lattice-like structure with agents locked into position. In Figure 12 [middle], we applied the learnt angular ranges from 'sheep-multi-lines', giving rise to more natural split-and-merge behaviour. Finally, in Figure 12 [right], we removed the obstacles and further restricted the cohesion and alignment angular range, which causes strongly anisotropic behaviour even in unobstructed environments.

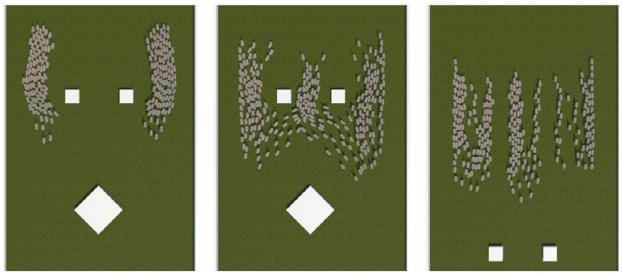

Figure 12: *Simulation with different angular range. [Left] All angular range to 1. [Middle] Parameters from optimization. [Right] Cohesion angular range set to 0.100, alignment angular range set to 0.300.*

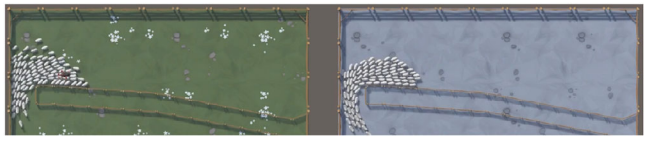

Figure 13: *Assessing the impact of density matching. [Left] The target herd with the same parameters as in Figure 6. [Right] A herd simulated using parameters learnt from the target, but without density matching.*

7.2.3. *Ablation of density matching*

Solely matching on the velocity direction field during optimization does not adequately capture motion from video (as mentioned in Section 6). This is demonstrated in Figure 13, where the target behaviour (on the left) is poorly matched by the behaviour optimized without a density term (on the right). This should be compared to the distribution in Figure 6, which incorporates density and represents a closer correspondence. Total average error values (0.72 with and 0.85 without density) confirm this.

7.2.4. *Ablation of direction variance matching*

Up to this point, we have concentrated on examples with little dispersion in agent direction within a cell. However, swarms and herds (such as ants and locusts) often move in a more uncoordinated fashion, and the resulting unstable velocity field can hamper convergence. To overcome this, we include a direction variation matching term in our optimization, which brings the added benefit of enabling optimization on a single input image, in cases where alignment is less dominant.

For ablation, we performed optimization on the 'seals' image (see Figure 14 [left]) with and without density matching. Here, the seals are randomly aligned, but with differing density in three zones (ocean demarcated in red, shoreline in yellow and in-shore in blue). We thus performed optimization and simulation with separate parameters for each zone. We set seal speed to 3.0 and 0.5 body lengths per second in water and on land, respectively, based on known body-length and movement characteristics.

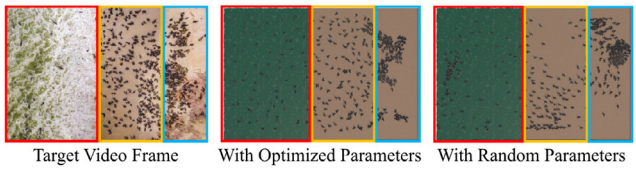

Figure 14: *Simulation based on 'seals' image. [Left] Target image. [Middle] Learnt simulation with direction variance matching. [Right] Learnt simulation without direction variance matching.*

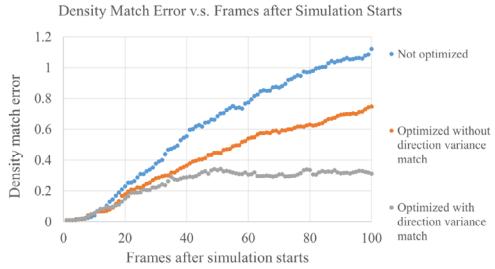

Figure 15: *Density error for simulations before and after optimization with and without direction variance error.*

Without direction variance matching (see Figure 14 [right]), the seals move and cluster in an unrealistically coordinated and aligned fashion. With direction variance (see Figure 14 [middle]) set at a uniform target of 3000 for all cells, based, by inspection, on random orientation of seals in the image, the visual match is much closer. This is quantitatively supported by lower density matching error, as shown in Figure 15.

7.2.5. *Speed matching and its relationship with density*

To highlight the necessity for speed matching, we simulated the 'sheep-narrow' clip using a fixed maximum allowed speed for the agent model. In the simulation, agents exhibited a consistent speed close to the allowed maximum, with a negligible standard deviation of $< 10^{-5}$. However, the observed standard deviation in the video footage was much larger, at 1.78, indicating that the speed of animals in the clip varies significantly more than evidence by the simulation. By employing the OU process, we were able to more accurately match both the mean and standard deviation of animal speeds.

Figure 16 reports the relationship between speed and animal density measured for the 'sheep-narrow' clip. Interestingly, at very low densities, the sheep move slowly, contrary to the expectation that they should move faster in obstacle-free areas. This behaviour is likely due to the sheep grazing and wandering in sparse areas. As density increases, speed initially rises, reflecting movement towards the centre of the migrating group. However, at very high densities, speed decreases again, likely due to crowding and friction as the animals move in close proximity. This dynamic relationship between speed and density highlights the necessity of incorporating such behaviours in our simulation model.

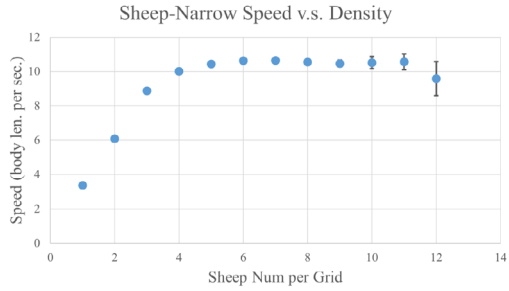

Figure 16: *Relationship between velocity and density in 'sheep-narrow', where the few isolated animals go much slower, and the most densely packed ones need to slightly slow down.*

7.3. Authoring

Once herd parameters have been learnt from a series of video-clips, users can author herds of any size in new environments and control their overall paths by drawing guidance arrows, providing a destination goal, or through a combination of the two.

Once guiding arrows in the form of oriented polylines have been drawn onto the landscape by a user, they are transformed into a single grid-based normalized navigation field $\mathbf{GF_{ij}}$, to be subsequently used by leader agents. This is carried out as follows: Each guiding arrow is sub-divided into oriented line segments whose lengths accord with the resolution of $\mathbf{GF_{ij}}$. Next, every grid cell intersected by one or more arrows is assigned the average direction of the incident line segments. Then line-segment directions are propagated iteratively to neighbouring cells and averaged out to a fixed user-specified radius, after which an exponential decay is applied.

During propagation, a cancelling field around obstacles is also established to prevent navigating on a collision course. In cases where the guidance field is going to be propagated from a grid-cell G_{ij} to an inaccessible obstacle cell G_{mn}, we define a cancelling field $\mathbf{CF_{ij}}$, as follows:

$$\mathbf{CF_{ij}} = \begin{cases} \mathbf{0}, & \text{if } \mathbf{GF_{ij}} \cdot \mathbf{V_{ij\to mn}} < 0 \\ -(\mathbf{GF_{ij}} \cdot \mathbf{V_{ij\to mn}})\mathbf{V_{ij\to mn}}, & \text{otherwise,} \end{cases} \tag{14}$$

where $\mathbf{V_{ij\to mn}}$ is the unit vector from G_{ij} to G_{mn}. Cancelling fields are also propagated out to a pre-set radius, and can be summed up to the navigation field to create a final guidance field.

Inspired by the modified Disjkstra's algorithm proposed in Patil *et al.* [PvdBC*11], we further provide the possibility to define the navigation field in terms of a single target destination. The results from this algorithm can be blended with the guidance field for additional flexibility.

In addition, users are able to control the speed of agents either by applying a speed-density correlation from a video clip or by directly painting a maximum speed cap into the environmental grid.

We constructed two test cases to demonstrate herd authoring in action:

1. Anthill (Figure 17). This depicts an environment with a large central obstacle surrounded by narrow vertical and broader hor-

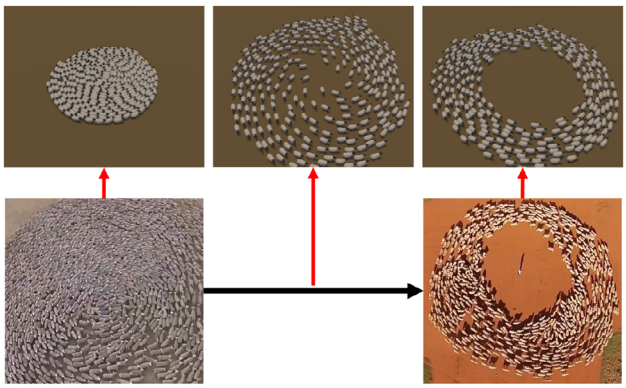

Figure 18: *Continuous transition from filled milling to hollow milling.*

izontal lanes. A circular guidance arrow was drawn around the obstacle. The upper (red bordered) and lower (blue bordered) regions were assigned 'ant-swarm' and 'ant milling' parameters, respectively.

2. Sheep pasture (Figure 1 [right]). This represents a corridor with various obstructions. The sheep herd is directed to follow a sinusoidal guidance arrow, with 'sheep-narrow' and 'sheep-broad' parameters painted on the left- and right-hand sides, respectively.

As demonstrated by these figures and the accompanying video, our authoring framework can be used to successfully transplant desired herding behaviours from video clips to authored scenes.

7.4. Visual continuity of simulated behaviour

A concern when agents switch roles or change control parameters is the potential introduction of motion discontinuities. Fortunately, since our model is forced-based, transitions are gradual even when parameter changes are abrupt.

While smooth transitions were observed in previous test cases (sheep in Figure 1 [right] and ants in Figure 17), these involved an authored guidance field and changes were only applied to agents as they crossed zone boundaries. To provide clearer evidence of the model's intrinsic smoothing capabilities, we designed a test case without authoring, focusing on a sheep herd transitioning between milling patterns.

This is illustrated in Figure 18: the top left and top right images show distinct filled and hollow milling behaviours learnt from video clips (bottom left and right). When the control parameters are switched discontinuously from filled to hollow, the simulation does not exhibit any jarring motion. Instead, as shown in the top middle image, the herd passes through a smooth, visually continuous intermediate state. This demonstrates that our model can achieve seamless behavioural changes directly from discontinuous parameter updates, without requiring explicit interpolation between parameter sets. The inherent dynamics of our force-based framework naturally smooth the transition.

Figure 19: *Integration with existing ecosystems. [Left] Herd of reindeer with our parameters during free wandering. [Right] A wolf pack that tends to walk in lines when searching for deer.*

7.5. Integration with fully simulated ecosystems

One of our stated aims is to allow straightforward deployment of learnt collective behaviours. To demonstrate integration with existing systems, we enhanced a virtual environment depicting a prehistoric valley to include more natural animal bahaviours when they are freely wandering in the valley (see Figure 19). We utilized one set of learnt parameters for the herd of reindeer. Leveraging the force-based nature of our model, an additional pre-defined force is included for predator avoidance in response to wolf predation.

Wolves were simulated with another set of parameters to govern pack movement during hunting. Upon detecting the deer, the wolves initially accelerated while adhering to our agent model to maintain formation. Once close to the herd, they transitioned to a pre-defined behavioural model, implementing specific hunting strategies, designed separately and informed by biological studies, to isolate and target vulnerable deer.

This example highlights our model's ability to preserve learnt collective behaviours while remaining compatible with external applications, contributing to more naturalistic agent simulations within virtual environments.

7.6. Limitations

Here, we note a few limitations of our model. First, the video preprocessing used to extract density and velocity values is relatively simple, lacks robustness in the case of large camera deformations and requires manual input. This processing could be replaced by more advanced computer-vision techniques, which would remove the requirement for camera stabilization and colour contrast between the animal species and background, without impacting the remainder of the pipeline. Second, our force-model is limited to a subset of herd behaviour. For instance, the separation of direction and speed simplifies learning and enhances authoring for generally homogeneous herds. However, our model cannot capture the behaviour of a crowd where agents have markedly different speeds, such as a mixture of slow, older animals and fast, younger ones. Lastly, the parameters in our extended Boids model do not establish a one-to-one correspondence with the resulting behaviour. Different parameter sets can lead to similar behaviours, which means the optimized parameters may differ from those expected for real animals. For instance, the perceptual range and field of view may not align with known biological traits for a species. However, per-

forming coupled optimization across multiple videos with the same species could help refine these parameters towards more biologically plausible values.

8. Conclusion

In this work, we have proposed a framework for learning general herd behaviour from videos, by optimizing the parameters of an agent-based simulation. This is based on a modified Boids model, which integrates a per-agent perception system with dynamically assigned roles. Agents in a leader role follow a local preset trajectory, while those in a follower role are influenced by neighbouring agents and environmental obstacles. The parameters of the Gaussian force model are optimized to greedily match macroscopic density and velocity direction fields extracted from a video, while the speed and its correlation with density are extracted independently.

We have demonstrated that our approach successfully reproduces various herd configurations, with emergent behaviours that quantitatively align with the aspect ratio, polarization, and angular momentum of the reference video. In addition, our method can be used as an authoring tool to map and simulate specific herd behaviours in new environments, including modification of the herd size, the speed and trajectory of animals, and the position of obstacles. More broadly, our approach is well-suited to a painting interface, in which users are able to select a brush, representing a particular herding behaviour, and then paint with it on specific regions of a virtual terrain, resulting in distinct animal behaviours in different areas.

In terms of future work, adding additional parameters to the optimized force model would enable a broader range of scenarios. For instance, accounting for correlations between terrain slope and animal behaviour would improve simulation accuracy for real-world topographies. Another potential improvement would involve modelling the interaction between different species, such as predator-prey dynamics, so as to better represent the complex dynamics observed in nature. Another avenue would be to capture the behaviour of animals at multiple scales. While our model is able to fit to macroscopic emergent behaviour, such as the global herd profile, it does not handle sub-clusters that may exhibit individual trajectories. For instance, ants might follow individual locally-oscillating trajectories that cannot be captured at the level of resolution of our study. Introducing a multi-scale representation of the density and velocity grid could allow us to extract different behaviours at varying levels of detail. Finally, at the authoring level, we would like to study the possibility of smoothly interpolating between different behaviours. Indeed, a simple interpolation in the Boids parameter space is not likely to lead to the expected transition between emergent behaviours. Incorporating additional differentiable loss functions in the optimization process (such as aspect ratio) might well open such an avenue.

Acknowledgements

Xianjin Gong conducted this research as part of a master's internship supported by LIX, École Polytechnique. We are grateful to Pierre Ecormier-Nocca for laying a solid foundation with his early investigations and to Pooran Memari for providing valuable insight. We also wish to thank Elouan Gros for his inspiration regarding herd editing.

References

[AGR*16] ALAHI A., GOEL K., RAMANATHAN V., ROBICQUET A., FEI-FEI L., SAVARESE S.: Social LSTM: Human trajectory prediction in crowded spaces. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2016), pp. 961–971.

[AHP19] AMIRIAN J., HAYET J.-B., PETTRÉ J.: Social ways: Learning multi-modal distributions of pedestrian trajectories with GANs. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops* (2019).

[CC07] COURTY N., CORPETTI T.: Crowd motion capture. *Computer Animation and Virtual Worlds 18*, 4-5 (2007), 361–370.

[CC14] CHARALAMBOUS P., CHRYSANTHOU Y.: The PAG crowd: A graph based approach for efficient data-driven crowd simulation. *Computer Graphics Forum 33*, 8 (2014), 95–108.

[CLN*14] CALOVI D. S., LOPEZ U., NGO S., SIRE C., CHATÉ H., THERAULAZ G.: Swarming, schooling, milling: Phase diagram of a data-driven fish school model. *New Journal of Physics 16*, 1 (2014), 015026.

[Far03] FARNEBÄCK G.: Two-frame motion estimation based on polynomial expansion. In *Image Analysis*. J. Bigun and T. Gustavsson (Eds.) Springer, Berlin Heidelberg (2003), pp. 363–370.

[FR12] FLAGG M., REHG J. M.: Video-based crowd synthesis. *IEEE Transactions on Visualization and Computer Graphics 19*, 11 (2012), 1935–1947.

[GCC*10] GUY S. J., CHHUGANI J., CURTIS S., DUBEY P., LIN M., MANOCHA D.: PLEdestrians: A least-effort approach to crowd simulation. In *ACMSIGGRAPH/Eurographics Symposium on Computer Animation, SCA 2010* (2010), Association for Computing Machinery Inc., pp. 119–128.

[GJFF*18] GUPTA A., JOHNSON J., FEI-FEI L., SAVARESE S., ALAHI A.: Social GAN: Socially acceptable trajectories with generative adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2018), pp. 2255–2264.

[GLR96] GUERON S., LEVIN S. A., RUBENSTEIN D. I.: The dynamics of herds: From individuals to aggregations. *Journal of Theoretical Biology 182*, 1 (1996), 85–98.

[HB06] HARTMAN C., BENES B.: Autonomous boids. *Computer Animation and Virtual Worlds 17* (2006), 199–206. https://api.semanticscholar.org/CorpusID:15720643.

[HH11] HEMELRIJK C. K., HILDENBRANDT H.: Some causes of the variable shape of flocks of birds. *PLoS One 6*, 8 (Aug. 2011), 1–13.

[HM98] HELBING D., MOLNAR P.: Social force model for pedestrian dynamics. *Physical Review E 51* (May 1998), 4282.

[HR16] HERBERT-READ J. E.: Understanding how animal groups achieve coordinated movement. *Journal of Experimental Biology 219*, 19 (Oct. 2016), 2971–2983.

[JCP*10] JU E., CHOI M. G., PARK M., LEE J., LEE K. H., TAKAHASHI S.: Morphable crowds. *ACM Transactions on Graphics 29*, 6 (Dec. 2010), 1–10.

[LBC*22] LEMONARI M., BLANCO R., CHARALAMBOUS P., PELECHANO N., AVRAAMIDES M., PETTRÉ J., CHRYSANTHOU Y.: Authoring virtual crowds: A survey. *Computer Graphics Forum 41*, 2 (2022), 677–701.

[LCHL07] LEE K. H., CHOI M. G., HONG Q., LEE J.: Group behavior from video: A data-driven approach to crowd simulation. In *Proceedings of the 2007 ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (2007), pp. 109–118.

[LCL07] LERNER A., CHRYSANTHOU Y., LISCHINSKI D.: Crowds by example. *Computer Graphics Forum 26*, 3 (2007), 655–664.

[LJ14] LI Z., JIANG Y.: Friction based social force model for social foraging of sheep flock. *Ecological Modelling 273* (2014), 55–62.

[LVCZ*18] LUKEZIC A., VOJIR T., CEHOVIN ZAJC L., MATAS J., KRISTAN M.: Discriminative correlation filter tracker with channel and spatial reliability. *International Journal of Computer Vision 126*, 7 (Jan. 2018), 671–688.

[LWPCL15] LI W., WOLINSKI D., PETTRÉ J., LIN M. C.: Biologically-inspired visual simulation of insect swarms. *Computer Graphics Forum 34* (2015), 425–434.

[MT01] MUSSE S. R., THALMANN D.: Hierarchical model for real time simulation of virtual human crowds. *IEEE Transactions on Visualization and Computer Graphics 7*, 2 (2001), 152–164.

[NGCL09] NARAIN R., GOLAS A., CURTIS S., LIN M. C.: Aggregate dynamics for dense crowd simulation. In *SIGGRAPH Asia'09: ACM SIGGRAPH Asia 2009 Papers* (New York, NY, USA, 2009), Association for Computing Machinery.

[PCHFJ16] PITA D., COLLIGNON B., HALLOY J., FERNÁNDEZ-JURICIC E.: Collective behaviour in vertebrates: A sensory perspective. *Royal Society Open Science 3*, 11 (2016), 160377.

[PFO*23] PAPADOPOULOU M., FÜRTBAUER I., O'BRYAN L. R., GARNIER S., GEORGOPOULOU D. G., BRACKEN A. M., CHRISTENSEN C., KING A. J.: Dynamics of collective motion across time and species. *Philosophical Transactions of the Royal Society B: Biological Sciences 378*, 1874 (2023), 20220068.

[PGM14] PERNA A., GREGOIRE G., MANN R.: On the duality between interaction responses and mutual positions in flocking and schooling. *Movement Ecology 2* (Oct. 2014), 22.

[PPD07] PARIS S., PETTRÉ J., DONIKIAN S.: Pedestrian reactive navigation for crowd simulation: A predictive approach. *Computer Graphics Forum 26*, 3 (2007), 665–674.

[PvdBC*11] PATIL S., VAN DEN BERG J., CURTIS S., LIN M., MANOCHA D.: Directing crowd simulations using navigation fields. *IEEE Transactions on Visualization and Computer Graphics 17* (Mar. 2011), 244–54.

[R*99] REYNOLDS C. W.: Steering behaviors for autonomous characters. In *Game Developers Conference* (1999), vol. *1999*, pp. 763–782.

[Rey87] REYNOLDS C. W.: Flocks, herds and schools: A distributed behavioral model. In *SIGGRAPH'87: Proceedings of the 14th Annual Conference on Computer Graphics and Interactive Techniques* (New York, NY, USA, 1987), Association for Computing Machinery, pp. 25–34.

[RLBP*23] REMPE D., LUO Z., BIN PENG X., YUAN Y., KITANI K., KREIS K., FIDLER S., LITANY O.: Trace and pace: Controllable pedestrian animation via guided trajectory diffusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2023), pp. 13756–13766.

[RWJM16] REN J., WANG X., JIN X., MANOCHA D.: Simulating flying insects using dynamics and data-driven noise modeling to generate diverse collective behaviors. *PloS One 11*, 5 (2016), e0155698.

[SBK*22] SAADATNEJAD S., BAHARI M., KHORSANDI P., SANEIAN M., MOOSAVI-DEZFOOLI S.-M., ALAHI A.: Are socially-aware trajectory prediction models really socially-aware? *Transportation Research Part C: Emerging Technologies 141* (2022), 103705.

[TCP06] TREUILLE A., COOPER S., POPOVIĆ Z.: Continuum crowds. *ACM Transactions on Graphics 25*, 3 (July 2006), 1160–1168.

[UO30] UHLENBECK G. E., ORNSTEIN L. S.: On the theory of the Brownian motion. *Physical Review 36* (Sep. 1930), 823–841.

[VCBJ*95] VICSEK T., CZIRÓK A., BEN-JACOB E., COHEN I., SHOCHET O.: Novel type of phase transition in a system of self-driven particles. *Physical Review Letters 75*, 6 (1995), 1226.

[vdBPS*08] VAN DEN BERG J., PATIL S., SEWALL J., MANOCHA D., LIN M.: Interactive navigation of multiple agents in crowded environments. In *I3D'08: Proceedings of the 2008 Symposium on Interactive 3D Graphics and Games* (New York, NY, USA, 2008), Association for Computing Machinery, pp. 139–147.

[vTP21] VAN TOLL W., PETTRÉ J.: Algorithms for microscopic crowd simulation: Advancements in the 2010s. *Computer Graphics Forum 40*, 2 (2021), 731–754.

[VZ12] VICSEK T., ZAFEIRIS A.: Collective motion. *Physics Reports 517*, 3 (2012), 71–140.

[WGO*14] WOLINSKI D., GUY S., OLIVIER A.-H., LIN M., MANOCHA D., PETTRE J.: Parameter estimation and comparative evaluation of crowd simulations. *Computer Graphics Forum 33* (May 2014), 303–312.

[XYWJ20] XIANG W., YAO X., WANG H., JIN X.: FASTSWARM: A data-driven framework for real-time flying insect swarm simulation. *Computer Animation and Virtual Worlds 31*, 4-5 (2020), e1957.

[YMW22] YUE J., MANOCHA D., WANG H.: Human trajectory prediction via neural social physics. In *European Conference on Computer Vision* (2022), Springer, pp. 376–394.

Supporting Information

Additional supporting information may be found online in the Supporting Information section at the end of the article.

Video S1