PNAS RESEARCH ARTICLE | COLLECTIVE BEHAVIOUR

Collective Motion Models in Multiple-Object Tracking

Jure Pahor[a,1], Timotej Kroflič[a,1], and Jure Dežman[a,1]

This manuscript was compiled on November 11, 2026

NOTE: This section will be added at a later milestone. NOTE: This section will be added at a later milestone. NOTE: This section will be added at a later milestone. NOTE: This section will be added at a later milestone. NOTE: This section will be added at a later milestone. NOTE: This section will be added at a later milestone. NOTE: This section will be added at a later milestone.

Multiple-Object Tracking | Collective Motion | Motion Models

Significance

Tracking many animals at once in video is central to studying collective behaviour, yet automatic trackers often swap identities when similar-looking individuals pass close to or behind one another. Most widely used trackers predict the next position of each animal as if it moved alone, in a straight line. Animals in groups, however, coordinate their motion with their neighbours. We bring a classic model of collective motion, in which individuals are attracted to, avoid and align with nearby group members, into the prediction step of a standard tracker. When a fish is briefly hidden, its predicted path follows the group rather than a straight line.

Introduction

Multiple-object tracking (MOT) aims to localise all objects of interest in a video and link them over time into trajectories with consistent identities (1). The dominant approach is *tracking-by-detection*, where a detector finds objects in each frame and these detections are then associated across frames (1, 2). Such trackers can operate *offline*, using the whole video at once (3), or *online*, using only past and current frames (4–7). A different line of work tracks objects through segmentation, most notably with the Segment Anything Model (SAM) family (8, 9), propagating object masks through the video instead of matching boxes (10).

Many online trackers build on *Simple Online and Realtime Tracking (SORT)* (4), which predicts each object's next bounding box with a Kalman filter (11) and matches these predictions to new detections by their overlap (IoU). Despite its simplicity, SORT and its successors (5–7, 12) remain competitive. They all retain the same constant-velocity motion model, which assumes that each object keeps moving in the direction and at the speed estimated from its previous frames. This ignores both changes in motion, such as accelerations and turns, and the influence of nearby objects, since every object is predicted independently of all others (4).

Other motion models, such as constant-acceleration or turn-rate models, have also been integrated into SORT-style trackers (13, 14), yet they too predict every object in isolation. In animal groups, however, individuals do not move independently. Models of collective behaviour describe each individual as a separate entity that steers according to simple rules relative to its neighbours: it moves away from those that are too close (separation), matches the heading of nearby ones (alignment), and moves towards those further away (cohesion) (15–17). Fitting such models to recorded fish trajectories has shown that real fish indeed interact through attraction, repulsion and alignment with a limited number of neighbours (18, 19). Interaction models of this kind have been used to improve pedestrian tracking (20), but to our knowledge collective motion models have not been used as the motion model of SORT-style trackers.

In this work, we replace the independent constant-velocity prediction with a collective motion model. We build on OC-SORT (7), a SORT (4) successor designed to be robust to occlusion and non-linear motion, and leave its association steps unchanged. Following the boids model (16), each track perceives the other tracks within a limited radius and field of view, and their positions and velocities give rise to cohesion, separation and alignment forces. The weighted sum of these forces enters the Kalman filter as an acceleration, so an occluded fish keeps moving with its group instead of along a straight line, while a fish with no neighbours is predicted exactly as before. We evaluate the approach on two fish datasets, SweetFish (21) and 3D-ZeF (22), in which many similar-looking individuals swim close together and frequently occlude one another.

Author affiliations: [a] Faculty of Computer and Information Science, University of Ljubljana, Večna pot 113, SI-1000 Ljubljana, Slovenia

Author contributions: J.P., T.K. and J.D. designed research, performed research, analyzed data and wrote the paper.

[1] J.P., T.K. and J.D. contributed equally to this work.

Our contributions are as follows:

- a collective motion model for tracking-by-detection, derived from the boids model;
- integration of the derived model into OC-SORT (7) as a drop-in replacement of the constant-velocity motion model;
- evaluation of the approach on two fish datasets SweetFish (21) and 3D-ZeF (22).

Methods

We follow the *tracking-by-detection* paradigm. At every frame t, a detector provides a set of boxes, each with centre (u, v), width w, height h and a confidence score. OC-SORT (7) maintains a set of tracks and, at every frame, predicts the state of each track with a motion model, associates the predictions with the new detections and updates the matched tracks with their detections. Unmatched detections start new tracks, and tracks that stay unmatched for too long are deleted. We keep the association of OC-SORT unchanged and modify only its motion model.

Motion model.

Constant-velocity Kalman filter (OC-SORT). Like the original SORT (4), OC-SORT (7) models each track with a linear Kalman filter (11) with state

$$\mathbf{x} = [u, v, s, r, \dot{u}, \dot{v}, \dot{s}]^\top, \qquad [1]$$

where $s = wh$ is the box area and $r = w/h$ its aspect ratio, which is assumed constant within the motion model. A detection is observed as $\mathbf{z} = [u, v, s, r]^\top$. The filter alternates between a predict and an update step,

$$\textit{predict} \begin{cases} \hat{\mathbf{x}}_{t|t-1} = \mathbf{F}\hat{\mathbf{x}}_{t-1|t-1} \\ \mathbf{P}_{t|t-1} = \mathbf{F}\mathbf{P}_{t-1|t-1}\mathbf{F}^\top + \mathbf{Q} \end{cases} \qquad \textit{update} \begin{cases} \mathbf{K}_t = \mathbf{P}_{t|t-1}\mathbf{H}^\top(\mathbf{H}\mathbf{P}_{t|t-1}\mathbf{H}^\top + \mathbf{R})^{-1} \\ \hat{\mathbf{x}}_{t|t} = \hat{\mathbf{x}}_{t|t-1} + \mathbf{K}_t(\mathbf{z}_t - \mathbf{H}\hat{\mathbf{x}}_{t|t-1}) \\ \mathbf{P}_{t|t} = (\mathbf{I} - \mathbf{K}_t\mathbf{H})\mathbf{P}_{t|t-1} \end{cases} \qquad [2]$$

where the transition matrix $\mathbf{F}$ adds the velocities to u, v and s over one frame, $\mathbf{H}$ selects the first four state entries, and $\mathbf{Q}$ and $\mathbf{R}$ are the process and observation noise covariances. The covariance $\mathbf{P}$ expresses the current uncertainty of the state estimate, and the Kalman gain $\mathbf{K}_t$ weighs the prediction against the detection: the more uncertain the prediction is relative to the detection, the closer the update moves the estimate towards the detection. When a track has no matching detection, the update is skipped and the prediction is kept as the estimate, $\hat{\mathbf{x}}_{t|t} = \hat{\mathbf{x}}_{t|t-1}$, $\mathbf{P}_{t|t} = \mathbf{P}_{t|t-1}$. Each track is therefore predicted independently of all others, and an occluded track continues along a straight line regardless of the group it belongs to.

Collective motion model (ours; NOTE: perhaps not the final version). We replace this independent prediction with one in which tracks interact, following the boids model (16, 23). For track i, let $\mathbf{c}_i = (u_i, v_i)^\top$ and $\dot{\mathbf{c}}_i = (\dot{u}_i, \dot{v}_i)^\top$ be the centre and velocity of its estimate $\hat{\mathbf{x}}_{t-1|t-1}$, and $\mathbf{d}_i = \dot{\mathbf{c}}_i / \|\dot{\mathbf{c}}_i\|$ its heading. Track i perceives the other tracks within a radius r_v and a field of view ϕ,

$$\mathcal{N}_i = \left\{ j \neq i \;:\; \|\mathbf{c}_j - \mathbf{c}_i\| \leq r_\mathrm{v},\;\; \frac{(\mathbf{c}_j - \mathbf{c}_i)\cdot \mathbf{d}_i}{\|\mathbf{c}_j - \mathbf{c}_i\|} > \cos\phi \right\}. \qquad [3]$$

Let $n_i = |\mathcal{N}_i|$ be the number of neighbours of track i. We summarise them by their mean position, their mean velocity and a repulsion vector,

$$\bar{\mathbf{c}}_i = \frac{1}{n_i}\sum_{j\in\mathcal{N}_i} \mathbf{c}_j, \qquad \bar{\dot{\mathbf{c}}}_i = \frac{1}{n_i}\sum_{j\in\mathcal{N}_i} \dot{\mathbf{c}}_j, \qquad \mathbf{g}_i = \sum_{j\in\mathcal{N}_i} \frac{\mathbf{c}_i - \mathbf{c}_j}{\|\mathbf{c}_i - \mathbf{c}_j\|^2},$$

where each neighbour in $\mathbf{g}_i$ pushes track i away from itself, and closer neighbours push harder. From these we obtain three drives: attraction towards the centre of the neighbours (cohesion), repulsion from nearby neighbours (separation) and alignment with their mean velocity,

$$\mathbf{F}_{A,i} = \frac{\bar{\mathbf{c}}_i - \mathbf{c}_i}{\|\bar{\mathbf{c}}_i - \mathbf{c}_i\|}, \qquad \mathbf{F}_{R,i} = \frac{\mathbf{g}_i}{\|\mathbf{g}_i\|}, \qquad \mathbf{F}_{P,i} = \frac{\bar{\dot{\mathbf{c}}}_i - \dot{\mathbf{c}}_i}{\|\bar{\dot{\mathbf{c}}}_i - \dot{\mathbf{c}}_i\|}. \qquad [4]$$

To avoid division by zero, a vector with zero norm is normalised to the zero vector.

As in the boids model (23), each drive is a steering force that requests an acceleration in its direction. Following the intuition of Newton's second law, $\mathbf{a} = \mathbf{F}/m$, so for an agent of unit mass the force is the acceleration, and the weighted sum of the drives gives the acceleration of the track,

$$\mathbf{a}_i = w_A\mathbf{F}_{A,i} + w_R\mathbf{F}_{R,i} + w_P\mathbf{F}_{P,i}, \qquad \mathbf{a}_i = \mathbf{0} \text{ if } \mathcal{N}_i = \emptyset. \qquad [5]$$

Since the drives are unit vectors, each weight is the change in velocity, in pixels per frame (unit of time), that its drive causes in one frame. The acceleration does not move the track directly but changes its velocity, which then moves its

position. Over one discrete time step, $\Delta t = 1$ frame, the centre and its velocity then follow the familiar equations of motion,

$$\dot{\mathbf{c}}_{i,t} = \dot{\mathbf{c}}_{i,t-1} + \mathbf{a}_i\,\Delta t, \qquad \mathbf{c}_{i,t} = \mathbf{c}_{i,t-1} + \dot{\mathbf{c}}_{i,t-1}\,\Delta t + \tfrac{1}{2}\mathbf{a}_i\,\Delta t^2,$$

which extends the *predict* step of Equation [2] by:

$$\hat{\mathbf{x}}_{t|t-1} = \mathbf{F}\hat{\mathbf{x}}_{t-1|t-1} + \mathbf{B}\mathbf{a}_i, \qquad \mathbf{B}^\top = \begin{bmatrix} \frac{1}{2}\Delta t^2 & 0 & 0 & 0 & \Delta t & 0 & 0 \\ 0 & \frac{1}{2}\Delta t^2 & 0 & 0 & 0 & \Delta t & 0 \end{bmatrix}, \tag{6}$$

where $\mathbf{F}\hat{\mathbf{x}}_{t-1|t-1}$ is the unchanged constant-velocity prediction and $\mathbf{B}$ adds the displacement and velocity change caused by $\mathbf{a}_i$ to the centre and its velocity, leaving the area and aspect ratio unaffected. Without neighbours, $\mathbf{a}_i = \mathbf{0}$ and Equation [6] reduces to the constant-velocity model. As $\mathbf{a}_i$ is treated as a known input, it adds no uncertainty, thus the covariance prediction of $\mathbf{P}_{t|t-1}$ and the *update* step of Equation [2] are unchanged. All accelerations are computed from the estimates at $t-1$ before any track is predicted, so the tracks move synchronously. During occlusion, the repeated small accelerations gradually turn the velocity of the track, so its predicted path bends towards its neighbours instead of continuing in a straight line.

Data association (OC-SORT). OC-SORT (7) extends SORT (4) with three observation-centric components: a re-update that corrects lost tracks once they are found again (ORU), and two association rounds (OCM & OCR), each solved with the Hungarian matching algorithm (24).

Observation-centric re-update (ORU; NOTE: perhaps not the final version). While a track is lost, its filter only predicts, so when the track is found again its velocity still reflects its motion before the gap. A plain update with the new observation corrects the position but largely keeps this outdated velocity, while re-initialising the track would discard the velocity altogether. Yet it is the velocity at re-detection that determines the predictions that follow. ORU therefore rolls the filter of a track lost after its last observation $\mathbf{z}_{t_1}$ and matched again to $\mathbf{z}_{t_2}$ back to t_1, and re-runs it over the gap on the virtual trajectory

$$\tilde{\mathbf{z}}_t = \mathbf{z}_{t_1} + \frac{t - t_1}{t_2 - t_1}(\mathbf{z}_{t_2} - \mathbf{z}_{t_1}), \qquad t_1 < t < t_2, \tag{7}$$

alternating predict and update steps (Equation [2]) with $\tilde{\mathbf{z}}_t$ in place of the missing observations, so that the track continues from t_2 with its average motion over the gap. As in the original work, we use this linear trajectory, which performed best among the alternatives compared there (7), and run the replay without our control input, as every replayed frame is already anchored by a virtual observation.

Observation-centric momentum (OCM). In the first association round, every pair of a predicted track and a detection receives a cost that combines overlap and motion direction,

$$C(\hat{\mathbf{X}}, \mathbf{Z}) = C_{\text{IoU}}(\hat{\mathbf{X}}, \mathbf{Z}) + \lambda C_v(\mathcal{Z}, \mathbf{Z}), \tag{8}$$

where $\hat{\mathbf{X}}$ are the predicted tracks, $\mathbf{Z}$ the new detections and $\mathcal{Z}$ the previous observations of each track. C_{IoU} is the negative IoU between predicted and detected boxes. C_v penalises the angle $\Delta\theta = |\theta_{\text{track}} - \theta_{\text{intention}}|$ between the direction in which a track has been moving, given by two of its observations Δt frames apart, and the direction from its earlier observation to the new detection. The Hungarian algorithm then selects the one-to-one assignment with the lowest total cost, and matches with IoU below a threshold are rejected.

Observation-centric recovery (OCR). Tracks and detections left unmatched enter a second association round, in which the cost is the negative IoU alone, computed against the last observation of each track instead of its prediction.

Only the first round uses the predicted boxes. Our motion model therefore affects tracking through the IoU term of Equation [8].

Experimental setup (NOTE: not conducted yet, metrics and datasets hold)

Datasets. We evaluate our approach against the OC-SORT (7) baseline on two fish datasets.

SweetFish (21) contains top-view recordings of a school of 10 juvenile sweetfish in a shallow tank, where the fish move in an approximately two-dimensional plane. Only its training video, with 10,000 frames at 2456×2058 pixels and 30 frames per second, is annotated, with boxes of a fixed size centred on each fish. Here we note that the actual video dimensions and FPS differs from the one described in original paper (21).

3D-ZeF (22) contains eight sequences with 1 to 10 zebrafish in an aquarium, each recorded simultaneously by a top and a front camera at 2704×1520 pixels and 60 frames per second.

Evaluation metrics. We evaluate all experiments with TrackEval (25) and report the standard MOT metrics (26). As our motion model only changes the predictions used for association, we focus on HOTA (27), which combines the detection accuracy (DetA) and the association accuracy (AssA) by their geometric mean, together with the association measures AssA, IDF1 (28), the F1 score of detections assigned to the correct identity, and the number of identity switches (IDSW). For completeness, we also report DetA and MOTA (29), which combines false negatives, false positives and identity switches.

Detector. NOTE: This section will be added at a later milestone.

Results

NOTE: This section will be added at a later milestone.

Discussion

NOTE: This section will be added at a later milestone.

Conclusion

NOTE: This section will be added at a later milestone.

1. W Luo, et al., Multiple object tracking: A literature review. *Artif. Intell.* **293**, 103448 (2021).
2. P Dendorfer, et al., Motchallenge: A benchmark for single-camera multiple target tracking: P. dendorfer et al. *Int. J. Comput. Vis.* **129**, 845–881 (2021).
3. G Brasó, L Leal-Taixé, Learning a neural solver for multiple object tracking in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. (IEEE), pp. 6246–6256 (2020).
4. A Bewley, Z Ge, L Ott, F Ramos, B Upcroft, Simple online and realtime tracking in *2016 IEEE International Conference on Image Processing (ICIP)*. pp. 3464–3468 (2016).
5. N Wojke, A Bewley, D Paulus, Simple online and realtime tracking with a deep association metric in *2017 IEEE international conference on image processing (ICIP)*. (IEEE), pp. 3645–3649 (2017).
6. Y Zhang, et al., Bytetrack: Multi-object tracking by associating every detection box in *European conference on computer vision*. (Springer), pp. 1–21 (2022).
7. J Cao, J Pang, X Weng, R Khirodkar, K Kitani, Observation-centric sort: Rethinking sort for robust multi-object tracking in *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. (IEEE), pp. 9686–9696 (2023).
8. N Ravi, et al., Sam 2: Segment anything in images and videos in *International Conference on Learning Representations*. Vol. 2025, pp. 28085–28128 (2025).
9. N Carion, et al., Sam 3: Segment anything with concepts in *International conference on learning representations*. Vol. 2026, pp. 138846–138923 (2026).
10. J Pelhan, A Lukezic, M Kristan, Ugo: Unified architecture for general multi-object tracking by segmentation. *arXiv preprint arXiv:2609.37339* (2026).
11. RE Kalman, A new approach to linear filtering and prediction problems. *J. Basic Eng.* **82**, 35–45 (1960).
12. C Li, J Ma, L Shen, Y Chen, L Chen, Occlusion-aware sort: Observing occlusion for robust multi-object tracking in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. pp. 42560–42570 (2026).
13. X Xiao, et al., A multi-fish tracking and behavior modeling framework for high-density cage aquaculture. *Sensors* **26** (2026).
14. GK Adari, Arc-sort: Adaptive, robust, compensated multi-object tracking with interacting multiple models. *arXiv preprint arXiv:2610.22306* (2026).
15. I AOKI, A simulation study on the schooling mechanism in fish. *NIPPON SUISAN GAKKAISHI* **48**, 1081–1088 (1982).
16. CW Reynolds, Flocks, herds and schools: A distributed behavioral model in *Proceedings of the 14th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '87. (Association for Computing Machinery, New York, NY, USA), p. 25–34 (1987).
17. ID COUZIN, J KRAUSE, R JAMES, GD RUXTON, NR FRANKS, Collective memory and spatial sorting in animal groups. *J. Theor. Biol.* **218**, 1–11 (2002).
18. Y Katz, K Tunstrøm, CC Ioannou, C Huepe, ID Couzin, Inferring the structure and dynamics of interactions in schooling fish. *Proc. Natl. Acad. Sci.* **108**, 18720–18725 (2011).
19. FJH Heras, F Romero-Ferrero, RC Hinz, GG de Polavieja, Deep attention networks reveal the rules of collective motion in zebrafish. *PLOS Comput. Biol.* **15**, 1–23 (2019).
20. S Pellegrini, A Ess, K Schindler, L van Gool, You'll never walk alone: Modeling social behavior for multi-target tracking in *2009 IEEE 12th International Conference on Computer Vision*. pp. 261–268 (2009).
21. MM Itoh, Q Hu, T Niizato, H Kawashima, K Fujii, Fish tracking challenge 2024: A multi-object tracking competition with sweetfish schooling data (2024).
22. M Pedersen, JB Haurum, SH Bengtson, TB Moeslund, 3d-zef: A 3d zebrafish tracking benchmark dataset in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. (IEEE), pp. 2423–2433 (2020).
23. CW Reynolds, , et al., Steering behaviors for autonomous characters in *Game developers conference*. (San Francisco, CA), Vol. 1999, pp. 763–782 (1999).
24. HW Kuhn, The hungarian method for the assignment problem. *Nav. Res. Logist. Q.* **2**, 83–97 (1955).
25. AH Jonathon Luiten, Trackeval (https://github.com/JonathonLuiten/TrackEval) (2020).
26. M Bashar, et al., In pursuit of many: A review of modern multiple object tracking systems (2026).
27. J Luiten, et al., Hota: A higher order metric for evaluating multi-object tracking. *Int. J. Comput. Vis.* **129**, 548–578 (2020).
28. E Ristani, F Solera, R Zou, R Cucchiara, C Tomasi, Performance measures and a data set for multi-target, multi-camera tracking in *European conference on computer vision*. (Springer), pp. 17–35 (2016).
29. K Bernardin, R Stiefelhagen, Evaluating multiple object tracking performance: the clear mot metrics. *EURASIP journal on image video processing* **2008**, 246309 (2008).