

Pixel personality for dense object tracking in a 2D honeybee hive

Katarzyna Bozek^a, Laetitia Hebert^a, Alexander S Mikheyev^a & Greg J Stephens^{a,b}

^a*Okinawa Institute of Science and Technology, 1919-1 Tancha Onna-son, Kunigami-gun, Okinawa 904-0495, Japan*

^b*Department of Physics and Astronomy, VU University Amsterdam, 1081 HV Amsterdam, The Netherlands*

Tracking large numbers of densely-arranged, interacting objects is challenging due to occlusions and the resulting complexity of possible trajectory combinations, as well as the sparsity of relevant, labeled datasets. Here we describe a novel technique of collective tracking in the model environment of a 2D honeybee hive in which sample colonies consist of $N \sim 10^3$ highly similar individuals, tightly packed, and in rapid, irregular motion. Such a system offers universal challenges for multi-object tracking, while being conveniently accessible for image recording. We first apply an accurate, segmentation-based object detection method to build initial short trajectory segments by matching object configurations based on class, position and orientation. We then join these tracks into full single object trajectories by creating an object recognition model which is adaptively trained to recognize honeybee individuals through their visual appearance across multiple frames, an attribute we denote as pixel personality. Overall, we reconstruct $\sim 46\%$ of the trajectories in 5 min recordings from two different hives and over 71% of the tracks for at least 2 min. We provide validated trajectories spanning 3,000 video frames of 876 unmarked moving bees in two distinct colonies in different locations and filmed with different pixel resolutions, which we expect to be useful in the further development of general-purpose tracking solutions.

Introduction

From cells in tissues to human crowds, the tracking of densely packed, active objects is a challenging and important problem. Indeed, the understanding of collective phenomena in biology across scales – from cellular interactions to the organization of animal groups – requires efficient tracking methods to connect ensemble behavior with the mechanisms of individual components (see e.g. [1, 2]). The automated tracking of individual objects in dense groups based on video recording is also of practical interest for the implementation of monitoring frameworks.

Social insects are an important category of animal groups whose reproduction and survival depend on the intrinsic organization and cooperation of the entire colony. Honeybees, an example of social insects, live in large organized family groups and perform activities unavailable to solitary insects, such as nest construction, division of labor, or group defense [3]. Due to long-established human husbandry, bees are accessible for observation with the use of an *observation beehive*. In such a hive the entire colony is placed on one surface of a honeycomb covered with glass. Tight spacing between the glass and the comb constrains the colony to an approximately 2D environment in which colony activities are performed by thousands of individuals in dense configurations and constant motion. The complexity of tracking within the observation hive environment is representative of other systems and thus of general interest.

We propose a new method for tracking of a large number of highly similar individuals in dense configurations. We adapt a previous method for identifying bee position and orientation [4] by retraining with samples from our imaging arrangement. We next create initial track fragments with a basic matching approach using

the difference in object class, orientation and position across neighboring frames. Finally, we iteratively learn visual features of hundreds of individuals within the short tracks, effectively assigning a unique “pixel personality” to each object. We use these learned visual features to match track fragments belonging to the same individual into full trajectories spanning the entire recording.

With the remarkable rise of deep learning, there has been surprisingly little progress with multi-target tracking, in part due to the lack of sufficiently large and annotated datasets. As an outcome of our work, we also provide a unique dataset of a large number of trajectories in a dense and challenging environment with nearly identical objects.

Related work

Among the diverse applications of multi-object tracking, crowd tracking has received the broadest attention in computer vision due to applications in surveillance systems, robotics, and human-computer interaction environments. Given the difficulty of tracking multiple moving, interacting, and visually overlapping targets, numerous approaches have explored the spatiotemporal patterns within human crowds and used local motion correlations to aid the tracking task [5–8].

Expanding tracking beyond locally correlated groups necessitates the use of additional cues to prevent identity swaps. Examples include temporal and spatial linking through quadratic programming [9], using appearance and motion cues [10], blending appearance to multiple hypothesis tracking [11], or integrating motion, time, position, size, and appearance into a single affinity model [12]. With the broad adoption of deep learning for image analysis, convolutional neural networks (CNNs) have opened up new possibilities of extracting visual cues in

the task of multi-object tracking. In particular, siamese architectures have attracted broad interest for their ability of associating detection pairs [13].

Despite the capacity of recurrent neural networks (RNNs) to learn temporal patterns, few studies have shown their application to multi-object tracking. However, a notable example [14] suggested an integrated architecture for trajectory association including object birth and death based on existing detections. Trained end-to-end, this model has shown high performance in simulated data and crowd recordings. RNNs have also been applied to space occupancy prediction based on unsupervised representation learning in the context of multi-object tracking [15, 16].

The need to create standards for the evaluation of tracking methods has inspired the creation of the Multi-Object Tracking benchmark, with both human tracks and also elements of scene and other object classes introduced in the benchmark’s 2016 release (MOT16) [17]. This dataset includes a total of 1,276 tracks within 14 recordings of up to 1,000 frames, addressing major challenges in scene tracking such as varying conditions, viewpoints, and a moving camera.

Compared to the MOT16 benchmark, honeybee colonies contain a significantly higher number of individuals ($\sim 1,000$ each), a number prohibitively large for the comprehensive exploration of track association hypotheses. Individual bees are generally of the same size, largely similar, and not available for recording in isolation, which limits the usefulness of comprehensive appearance models [18]. The beehives also exhibit a varied, uneven background composed of comb cells, brood, and food, the viewpoint of the recordings is static, and object classes are limited to two types of bee body postures.

Due to the difficulties of multi-object tracking, studies in biological systems often produce collective recordings under controlled laboratory conditions or with the marking of individuals. Typical experimental conditions involve plain 2D or 3D arenas with uniform illumination and high foreground-background contrast [18, 19]. The controlled aspect of these arrangements allows for accurate object detection as well as extraction of comprehensive appearance cues from large volumes of recordings of each individual [18]. The marking of individuals with unique barcodes has also been extensively used in the study of social insects [2, 20, 21], but has important feasibility limitations in large colonies with short generation time spans. Both the laboratory conditions and marking itself can also have an important effect on behavior, necessitating versatile methods for marker-less tracking of groups of individuals in their natural environments. Such methods would also create the opportunity for collective tracking in systems that cannot be observed under controlled conditions and that cannot be easily marked.

Data

We collected two video recordings from observation beehives in two different locations and filmed with cameras of resolution (5120×5120 pixels and 3860×2160 pixels, respectively) at 30 fps (Fig. S1) for 5 min. The images of the first recording were downsized by a factor of two along both axes to reach a similar spatial resolution of the second recording, in which a bee’s largest dimension is ~ 80 pixels. Images, a segmentation tutorial and resulting tracked trajectories are available at <https://groups.oist.jp/bptu/honeybee-tracking-dataset>.

Object detection

We followed and improved a previous segmentation approach for the efficient estimation of honeybee position and orientation [4] by combining the orientation and position loss functions into one architecture and adding one convolutional layer (Fig. S2). This modified network network enables the simultaneous computation of both object orientation and class.

We used two object classes corresponding to a fully visible bee and an abdomen of a bee which is inside of a comb cell. For the abdomen class, the orientation angle is not defined and is set to 0. We retrained the modified architecture on a previous dataset [4], downsized by a factor of two on both axes to match the resolution of our video recordings. We used ellipse-shaped segmentation labels for marking fully visible bees and round-shaped for marking abdomens of bees inside comb cells. Both labels were centered over reported bee central point position, ellipse-shaped labels were additionally rotated to align with the bee body axis. We resized the segmentation labels to $r1 = 6.7$ pixels and $r2 = 11.7$ pixels for the semi-minor and semi-major axes of the ellipse-shaped segmentation label of a fully visible bee and $r = 6.7$ pixels radius for the round-shaped segmentation label of bee abdomens inside the comb cells (Fig. S3). This resizing produced more precise position estimation and improved bee separation in dense configurations. The segmentation output was post-processed as described previously [4].

To improve the accuracy of the detection model, which was trained on the original data [4], we retrained the network on a small number of frames from our video recordings. We performed the labelling in an iterative manner, where predictions of the detection model were first manually corrected using a custom interface (Fig. S4), and then used to retrain the detection model. In each iteration, two frames of each recording spaced by 0.1 s were corrected and the network was trained for 5 iterations on image patches of 256×256 pixels covering the newly labeled frames. In each iteration we noticed a decrease by two of the time spent on labeling corrections of the following two frames and we stopped the process when no time gain remained, which happened when 32 frames of each recording were labeled. The final network used

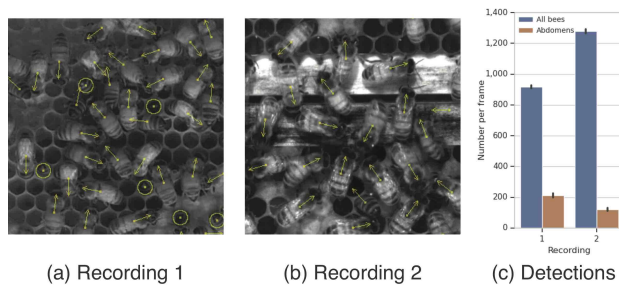


FIG. 1: Reliable object detection is the foundation of our dense tracking approach. (a, b) Example detections from two recordings. Body centers and head-tail orientation are marked with arrows, abdomens of bees inside comb cells are marked with circles. (c) Average number of detections per frame with vertical black lines denoting the standard deviation. The low variation in the number of detected objects is consistent with a constant and small false positive rate of ~ 0.06 [4].

for object detection was retrained for 30 iterations on the newly labeled set of frames.

The training of the segmentation network and the detection of bees were performed on image patches of size 256×256 pixels. Given the higher error rates of object detection along the image boundaries [4], we overlapped the patches by a margin of 25 pixels and results within margins were discarded. Both recordings captured images at 30 fps and we used a segment of 5 min down-sampled to 10 fps. Our segmentation procedure resulted in 2,756,532 and 3,840,312 detections from recording 1 and 2, respectively with accuracy (after removing image margins) of ~ 0.06 false positive rate, an average position error of 4.9 pixels and an orientation angle error of 9.7° (Fig. 1), thus providing a strong foundation for trajectory reconstruction.

Constructing track fragments using configuration-based matching

For each frame, the detection algorithm provides a list of entries x, y, c, α , where x, y are object coordinates, c is the object category (fully visible bee or an abdomen only), and α is the orientation angle. We combine these four features into a similarity measure D_{ij} to identify object detections across neighboring frames,

$$D_{ij} = d_{ij} + w|c_i - c_j| + w|\sin(\alpha_i - \alpha_j)| + |d_{x_{ij}}| + |d_{y_{ij}}|,$$

where d_{ij} is the Euclidian distance between objects i and j , w is a scaling factor of the class and angle differences between the objects, and $d_{x_{ij}}$ and $d_{y_{ij}}$ are differences in the motion vectors along the x and y axis in this matching step compared to the previous one of the same objects i and j . Since the orientation angle is based on the body axis estimation [4], the angle component of D_{ij} penalizes deviations from axis orientation rather than the head-tail orientation. We set the scaling factor $w = 20$.

Using D_{ij} , objects in the following frames of each recording were matched in an iterative manner, Fig. 2. Starting from lowest values of D between two consecutive video frames the respective objects were matched into a track, until no pairs of objects with $D < 50$ remained. Matching was done between following pairs of frames and tracks for which no match was found over the last 10 frames were ended. Only tracks of length > 30 were kept as shorter trajectories were considered as individual detections. This greedy detection matching is reminiscent of the Hungarian algorithm, however we do not search for a complete matching as occlusions and detection errors can lead to wrong associations. We only join detections that are sufficiently close which ensures that the track fragments are correct and thus leaving the task of expanding these fragments to the appearance-based, pixel personality matching described below.

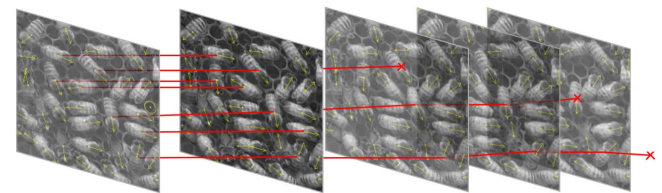


FIG. 2: Object detections are joined into short track fragments using a simple distance metric. We join the nearest detections in pairs of consecutive video frames using a distance measure composed of position, orientation angle, object class and direction of movement. The joining procedure results in short track fragments of varying lengths as well as start and end points.

The matching procedure resulted in 8,895 and 8,565 track fragments in recordings 1 and 2, respectively. The fragments have various lengths (Fig. S5) and span a total of $\sim 10^6$ detections in each recording. Within the first 30 s of the recording we searched for a frame with the highest number of tracks with length > 100 . These tracks represented the initial track set that was extended further as described in the next section. This set contains 794 initial trajectories from recording 1 and 1,115 in recording 2. We also note that 36 and 137 trajectories respectively, spanning $> 95\%$ of frames, were considered complete and not extended further.

Joining track fragments using appearance-based matching

The track segments obtained using the matching criteria defined above are fragmented due to imperfect detections, occlusions, and irregular, sometimes rapid motion. To connect the fragmented tracks and individual detections into trajectories spanning the entire recording, we devised a matching approach exploiting appearance, space, and temporal cues, as well as taking into account the space occupancy of other tracked objects.

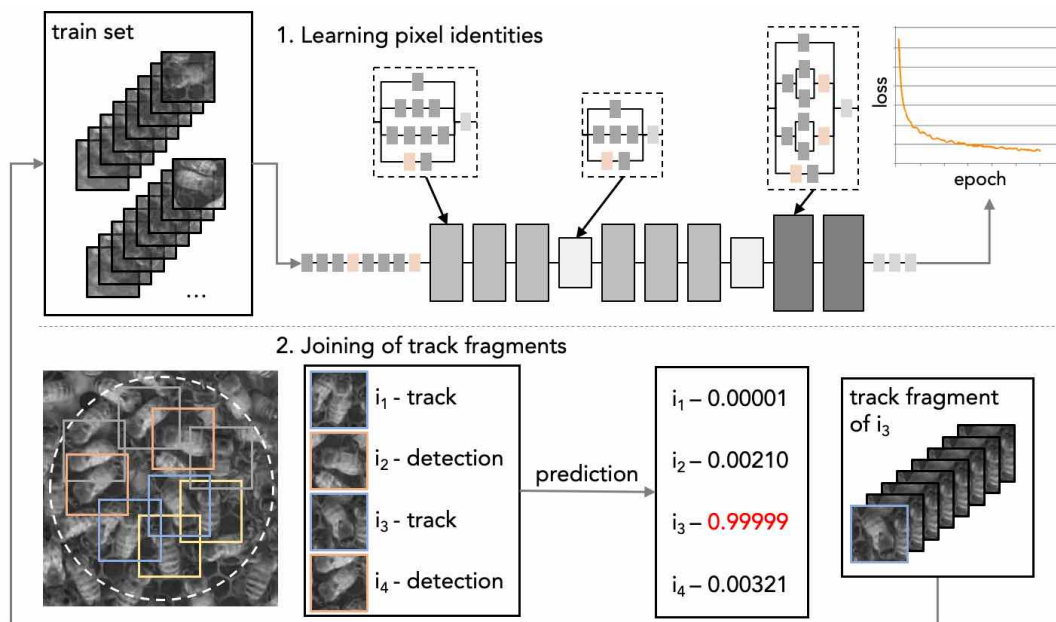


FIG. 3: Overview of the track joining algorithm. (upper left) Initial track fragments from different individuals are used as a training set for the object identity recognition network [22]. After the training loss function decreases below a predefined value, we consider track extensions through the joining of fragments. For a given object we examine the video frame immediately following the current end of the assembled trajectory and consider all detections within a predefined similarity (bottom-left, dashed line) from the last point in the trajectory. Among these detections, some belong to trajectories of other objects (yellow squares), some are part of a track and are not at the beginning of track (gray squares), some are part of a track and are at the beginning of this track (blue squares), and some represent individual detections (single-frame fragments, orange squares). For detections belonging to the latter two categories we use the Inception V3 network to predict detection identity. If the predicted identity of any of the tested detections matches the given object (here, i_3) this detection is added to the trajectory. If the chosen detection is at the beginning of a track fragment, all detections belonging to this fragment are added to the assembled trajectory. Finally, the network is retrained on the updated training set and the process repeats.

We captured cues to individual identity across varying visual appearances, an attribute we denote as pixel personality, by using the object recognition network Inception V3 [22]. We also note previous approaches that exploit subtle but consistent visual differences for multi-animal tracking [18, 23]. The initial track segments described above were used as a training set for learning object identities. Images of individuals from each track were cropped to the size 80×80 pixels centered over the detected positions of each individual. Only N occurrences from the latter part of the individual's track were incorporated into the training set. If a trajectory was shorter than N , the occurrences were repeated until size N was obtained. These individual instances were stored in memory, preserving their order in the recording. We also randomly sampled images of background from the space in between the bee detections from different video frames. We collected 10^4 background images and selected $N = 250$ which allows for feasible processing time.

Starting with our assembled training set, including background images and images from separate objects labeled as a distinct category, we iteratively performed training and matching steps, Fig. 3. In the train step, the network is trained until the loss decreases below a se-

lected value ($L < 0.01$). In addition, two data augmentation operations are randomly performed on the train set – masking and flipping. Masking is aimed at reducing the background influence on the object recognition and consists of covering the image's outer parts and leaving only the center either square- or round-shaped (Fig. S6). Flipping is performed along the x- and y-axes. The loss function of background images is down-scaled by a factor of 0.1 to compensate for their higher number compared to other object categories.

After the network reaches a predefined loss L , the matching step is performed. The object identities learned by the network are used to find the most probable object detections in the video frame following the frames of the given object instances contained in the train set. Specifically, for a given trajectory k that ends at frame t at position p , we search for detections in the frame at $t + d_t$ which fulfill the following criteria: (1) are not already assigned to another object, (2) are not part of a track fragment or are within first 10 instances of a track fragment, (3) are no further than a distance $D\sqrt{d_t}$ from p , where $D = 80$ pixels is approximately the largest object dimension size and d_t is the difference (in frames) between currently tested frame and the last frame of tra-

Recording	MT2↑	MT5↑	ML2↓	ML5↓	Det. err.	ID swaps	Occlusion	Lost	Assoc. MT2	Assoc. MT5
Rec. 1	0.74	0.46	0.01	0.12	0.03	0.19	0.10	0.17	981	10,681
Rec. 2	0.71	0.46	0.02	0.16	0.02	0.30	0.03	0.10	470	3,772

TABLE I: Summary results. The tracking quality metrics shown are: trajectories tracked for at least 1 *min* 40 *sec* (MT2), for at least 4 *min* (MT5), for less than 10 *sec* (ML2), less than 30 *sec* (ML5). Within the trajectories not counted as MT5 sources of errors were categorized as detection error (*Det. err.*), identity swap (*ID swaps*), occlusion (*Occlusion*), or trajectory loss due to other reasons, for example, lack of detections or lack of detections recognized as the respective object (*Lost*). Columns *Assoc. MT2* and *Assoc. MT5* list the number of track and detection associations performed using the object recognition network described previously.

jectory k (Fig. 3). Detections fulfilling all three criteria are cropped from the video frame and tested with the object recognition network for their similarity to the object in trajectory k .

From the set of tested detections, the one showing the highest softmax score that is above a cutoff score of $s = 0.1$ in the object category k is selected for extending trajectory k . This detection is added to the trajectory and the train set with label k . The earliest instances of trajectory k are removed from the train set keeping the number of instances of label k at the same number N . If this detection belongs to a track fragment (recall that the detection can be at most 10 frames from the start of a track fragment), the entire track is added to the train set with the label k and the same number of earlier instances of this trajectory are removed from the train set. If no tested detection shows a score above s or no detection is found fulfilling the location and occupancy criteria, trajectory k is not extended in frame $t + d_t$ and d_t is increased by 1.

After the matching step, the iteration is repeated with the updated train set. The trajectories that either have been completed through matching or failed to be matched over the last 50 iterations are removed from processing. In training step we set the learning rate to 0.00005 and used the Adam optimizer [24], to ensure gradual learning of the partially altered train set. The solution was implemented in tensorflow and deployed on IBM power system with four P100 GPU's. The matching step was computed in parallel on 160 cores of the computational node with train set modification, test set construction, and filtering of results performed by parallel process pools, Fig. S7.

For both recordings, network training and detection matching were first performed on half of the objects for 100 iterations, after which the procedure was done for another 900 iterations on all objects whose trajectories were yet unfinished. Reducing the number of tracked objects allowed for faster training epochs in the initial 100 iterations during which several trajectories were completed. The total of 1000 iterations was completed during one week using one computation node for each recording.

Results

To quantify the accuracy of our results in comparable format, we adopted the metrics used in the MOT16 benchmark [17]. Importantly however, in the context of a honeybee hive, ground truth detection of the trajectories would be prohibitively costly to obtain and thus we omit Multiple Object Tracking Accuracy (MOTA) and Multiple Object Tracking Precision (MOTP). Even so, we provide other track quality measures which capture the effectiveness of our solution.

We used two tracking quality measures of MOT16 – mostly tracked (MT) and mostly lost (ML) track – that we quantified over the first 2 min and the entire 5 min of both recordings. According to MOT16, a target is mostly tracked if it is successfully tracked for at least 80% of its life span. It is irrelevant for this measure whether the identity remains the same throughout the track. Mostly lost tracks are those only recovered for less than 20% of the tested time span. Also, we did not consider as track discontinuities temporary identity changes after which the track comes back to its original identity.

Trajectories counted as MT2 in Table I are those tracked for at least 100s, MT5 are those tracked for at least 4 min, ML2 tracks are recovered for less than 10s, and ML5 for less than 30s. Within the trajectories not counted as MT5 we additionally assessed the sources of errors as detection error, identity swap, occlusion, or trajectory loss due to lack of detections that were recognized as the respective object.

We found 74% and 71% of trajectories mostly tracked over 2 min in recording 1 and 2, respectively. 46% of trajectories were mostly tracked over the entire 5 min of both recordings. Few trajectories were mostly lost with 12% and 16% classified as ML5 in recording 1 and 2, respectively. Unsurprisingly, identity swaps among the highly similar honeybees were the predominant cause of tracking errors with 19% and 30% trajectories lost due to identity swap in both recordings, respectively. Trajectory loss is the second most frequent reason for tracking error. This type of error might result from the inaccuracy of the object recognition network, as well as from previous incorrect associations of tracks that might have incorrectly

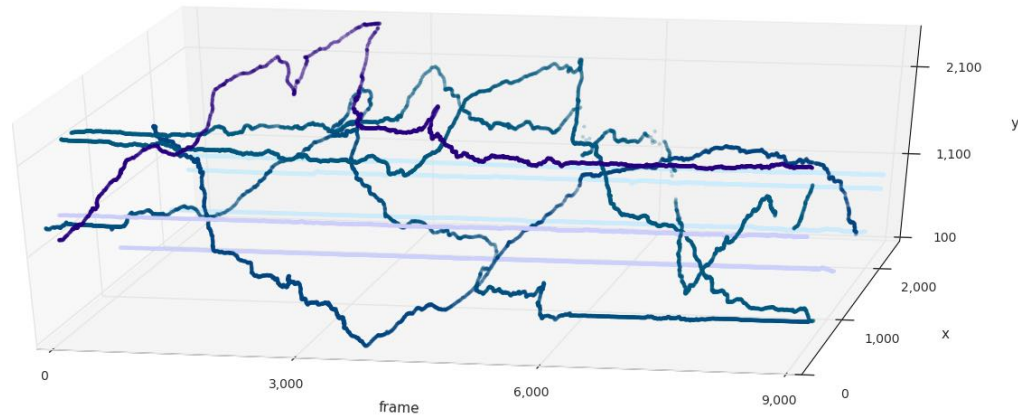


FIG. 4: Tracking reveals a variety of movement dynamics within the hive. Darker colors represent example trajectories of rapidly moving bees, while bright colors represent stationary bees.

assigned the given object's track to another object in one of the previous iterations of the track matching algorithm leaving no possible detections to associate to this object's trajectory. Trajectory loss due to occlusions (10% and 3% trajectories, respectively) might be caused by a dramatic change in object's visual cues, not captured in the train set as the object was occluded, as well as, if the occlusion lasts longer than 50 frames the trajectory was stopped according to the algorithm design. Importantly, only a very small number of trajectories (2–3%) were lost due to detection errors, corroborating the high accuracy of the object detection method.

The setting of this important biological system is substantially different than those considered in the MOT16 benchmark. The density of objects, number of object classes, visual variability of the scene, length and size of the recordings differentiate our study from the videos included in the benchmark. Nevertheless, our results according to the MT and ML measures (22% and 37%, respectively in MOT16) are both promising and substantially improved. Similarly, we are currently lacking ground truth detections to comprehensively estimate MOTA and MOTP measures. However, the low percentage of trajectory loss due to detection errors suggest that our detection accuracy is similar or better than the benchmark results.

We also note that our solution addresses several important challenges of tracking in dense environments such as rapid, irregular motion (MovieS1), body posture changes (MovieS2), temporary occlusions (MovieS3), or densely

packed environments (MovieS4). Also, 70% of objects tracked over the first 2 min and 46% of objects tracked over 5 min represent a significant proportion of the densely packed comb (MovieS5 and MovieS6).

Currently limited to 5 min, our solution can be straightforwardly parallelized across computational nodes and time segments, thus allowing for longer term tracking. Even within 5 min, essential behavioral dynamics can be observed (Fig. 4, S8, S9) enabling the quantitative analysis of collective properties of a naturally behaving honeybee colony at a single-organism resolution. The capability to track without tags also facilitates the analysis of collective replicates and is relevant in systems where population fluctuations are important such as dividing cells. While we developed our tracking solution in the biologically-relevant and interesting system of a honeybee hive, we expect our method to easily generalize to other contexts.

Acknowledgments & Data

Funding for this work was provided by the OIST Graduate University to ASM and GS. Additional funding was provided by KAKENHI grants 16H06209 and 16KK0175 from the Japan Society for the Promotion of Science to ASM. We are grateful to Yoann Portugal for assistance with colony maintenance and image acquisition. Data consisting of videos, detection and resulting trajectories for two beehive recordings are available here: <https://groups.oist.jp/bptu/honeybee-tracking-dataset>.

[1] J. D. Crall, C. M. Switzer, R. L. Oppenheimer, A. N. Ford Versypt, B. Dey, A. Brown, M. Eyster, C. Guérin,

N. E. Pierce, S. A. Combes, and B. L. de Bivort, *Science* **362**, 683 (2018).

- [2] D. P. Mersch, A. Crespi, and L. Keller, *Science* **340**, 1090 (2013). pp. 234–241.
- [3] T. D. Seeley, *The Wisdom of the Hive: the social physiology of honey bee colonies* (Harvard University Press, 2009).
- [4] K. Bozek, L. Hebert, A. S. Mikheyev, and G. J. Stephens, in *Computer Vision and Pattern Recognition (CVPR), 2017 IEEE* (2017).
- [5] L. Kratz and K. Nishino, in *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on* (ieeexplore.ieee.org, 2010) pp. 693–700.
- [6] I. Ali and M. N. Dailey, in *Advanced Concepts for Intelligent Vision Systems*, Lecture Notes in Computer Science, edited by J. Blanc-Talon, W. Philips, D. Popescu, and P. Scheunders (Springer Berlin Heidelberg, 2009) pp. 540–549.
- [7] W. Ge, R. T. Collins, and R. B. Ruback, *IEEE Trans. Pattern Anal. Mach. Intell.* **34**, 1003 (2012).
- [8] M. Rodriguez, J. Sivic, I. Laptev, and J. Y. Audibert, in *2011 International Conference on Computer Vision* (2011) pp. 1235–1242.
- [9] R. Henschel, L. Leal-Taixé, B. Rosenhahn, and K. Schindler, (2016), arXiv:1607.07304 [cs.CV].
- [10] Bing Wang, Gang Wang, Kap Luk Chan, and Li Wang, *IEEE Trans. Pattern Anal. Mach. Intell.* **39**, 589 (2017).
- [11] C. Kim, F. Li, A. Ciptadi, and J. M. Rehg, in *2015 IEEE International Conference on Computer Vision (ICCV)* (2015) pp. 4696–4704.
- [12] C. Kuo and R. Nevatia, in *CVPR 2011* (2011) pp. 1217–1224.
- [13] L. Leal-Taixé, C. Canton-Ferrer, and others, *Proc. IEEE* (2016).
- [14] A. Milan, S. H. Rezatofighi, A. Dick, I. Reid, and K. Schindler, (2016), arXiv:1604.03635 [cs.CV].
- [15] P. Ondruska, J. Dequaire, D. Z. Wang, and I. Posner, (2016), arXiv:1604.05091 [cs.LG].
- [16] P. Ondruska and I. Posner, (2016), arXiv:1602.00991 [cs.LG].
- [17] A. Milan, L. Leal-Taixé, I. Reid, S. Roth, and K. Schindler, arXiv:1603.00831 [cs] (2016), arXiv:1603.00831.
- [18] F. Romero-Ferrero, M. G. Bergomi, R. Hinz, F. J. H. Heras, and G. G. de Polavieja, (2018), arXiv:1803.04351 [cs.CV].
- [19] M. Kabra, A. A. Robie, M. Rivera-Alba, S. Branson, and K. Branson, *Nat. Methods* **10**, 64 (2013).
- [20] F. Wario, B. Wild, M. J. Couvillon, R. Rojas, and T. Landgraf, *Front. Ecol. Evol.* **3** (2015).
- [21] T. Gernat, V. D. Rao, M. Middendorf, H. Dankowicz, N. Goldenfeld, and G. E. Robinson, *Proc. Natl. Acad. Sci. U. S. A.* **115**, 1433 (2018).
- [22] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, (2015), arXiv:1512.00567 [cs.CV].
- [23] A. Pérez-Escudero, J. Vicente-Page, R. C. Hinz, S. Arganda, and G. G. de Polavieja, *Nat. Methods* **11**, 743 (2014).
- [24] D. Kingma and J. Ba, (2014), arXiv:1412.6980 [cs.LG].
- [25] O. Ronneberger, P. Fischer, and T. Brox, in *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, Lecture Notes in Computer Science, edited by N. Navab, J. Hornegger, W. M. Wells, and A. F. Frangi (Springer International Publishing, 2015)

Supplementary Material

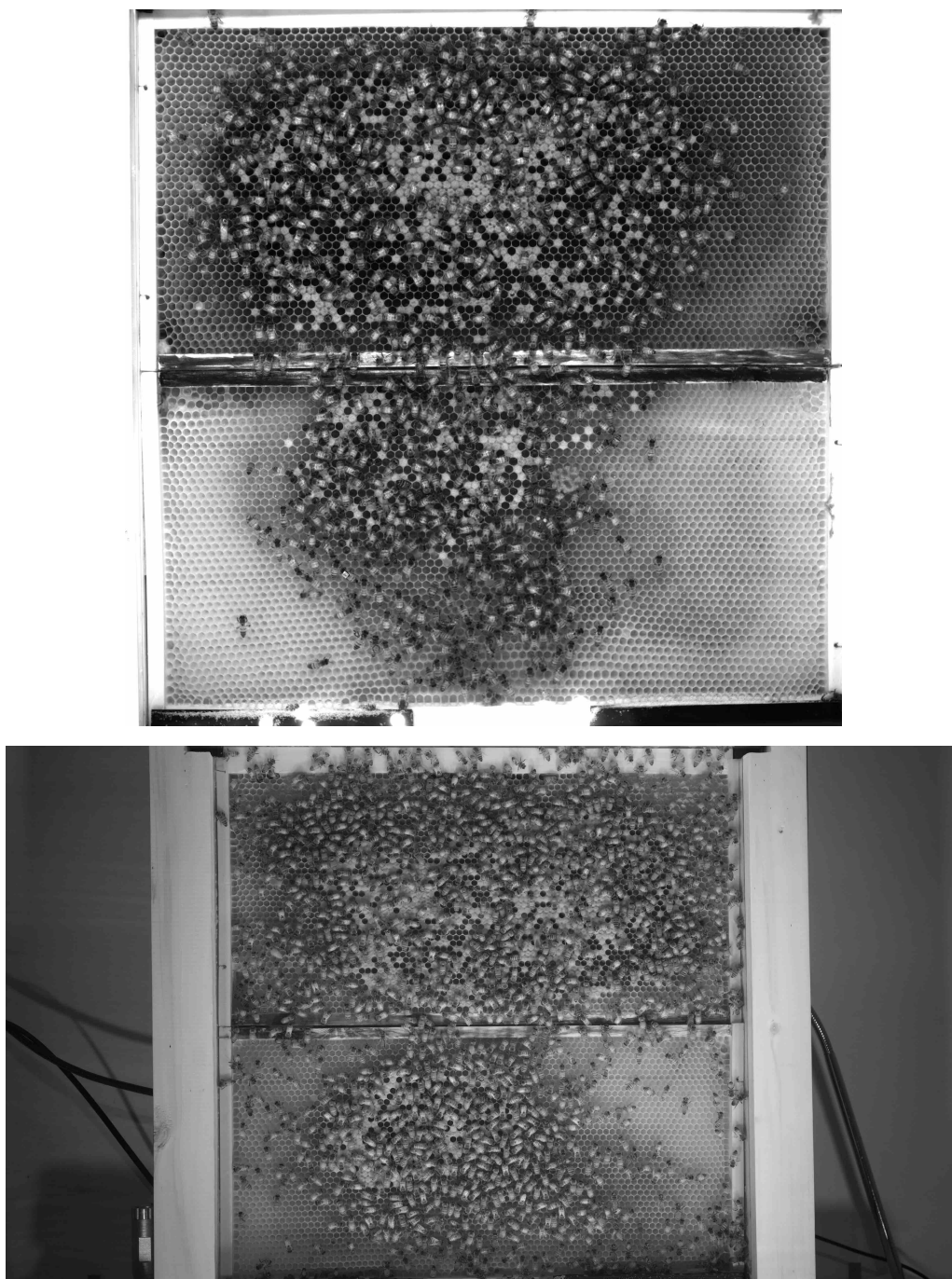


FIG. S1: Example frames from recording 1 (upper image) and 2 (lower image).

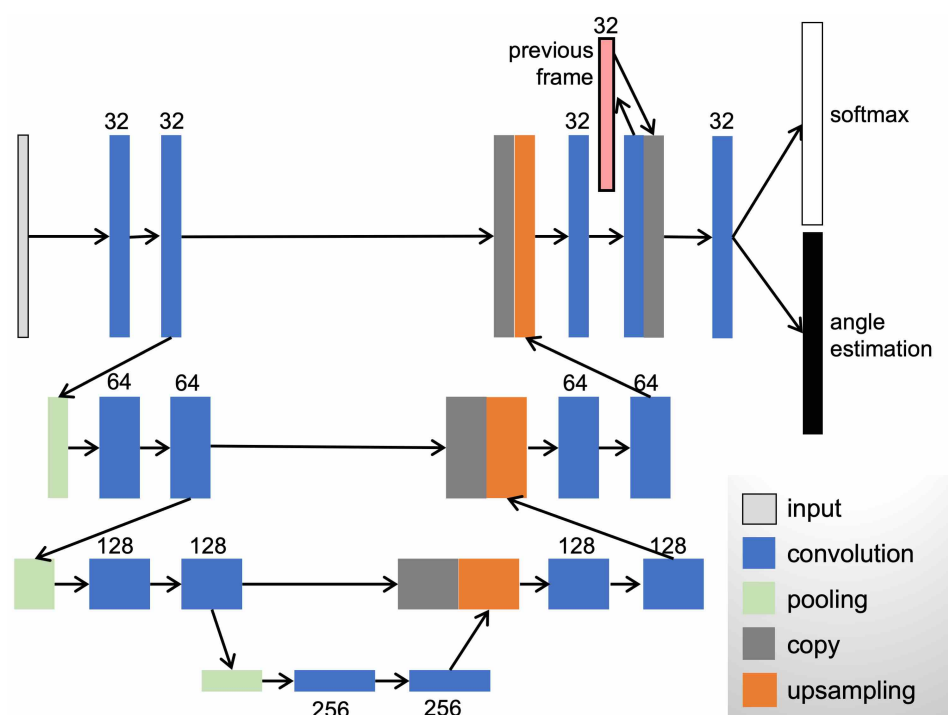


FIG. S2: UNet architecture used for object detection. The network was reduced to $\sim 6\%$ of the parameters compared to the original UNet [25]. A temporal component was added (pink box) to incorporate information from the previous video frame as a prior to improve the prediction of object positions in the following frame. An additional convolutional layer was added before the final two loss function layers – softmax for class prediction and angle loss function as proposed in [4].

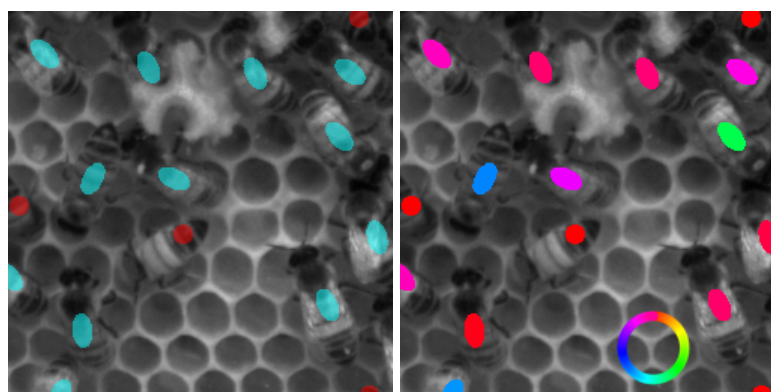


FIG. S3: Example labels and images used for object detection. The Image size was 256×256 pixels, segmentation labels indicate object class (left panel) and orientation angle (right panel, angle color code indicated in the inserted circle). Segmentation labels are sized to $r_1 = 6.7$ pixels and $r_2 = 11.7$ pixels for the semi-minor and semi-major axes of the ellipse-shaped label and $r = 6.7$ pixels radius for the round-shaped label.

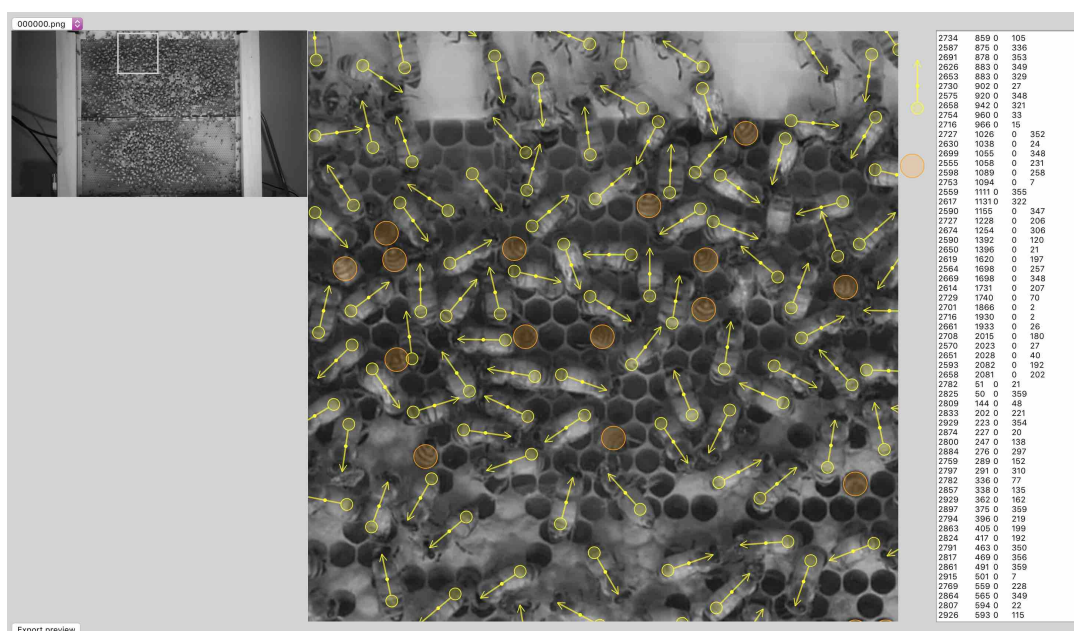


FIG. S4: Labeling interface. The interface was implemented in JavaScript. Labeling was done through dragging and dropping the yellow arrow and orange round symbols on the labelled image. The round end of the arrow allows for rotating it to align with the bee body axis. The entire video frame is labeled by scrolling with the use of square shaped zoom in the left image. An editable list of all annotations is displayed in the right panel.

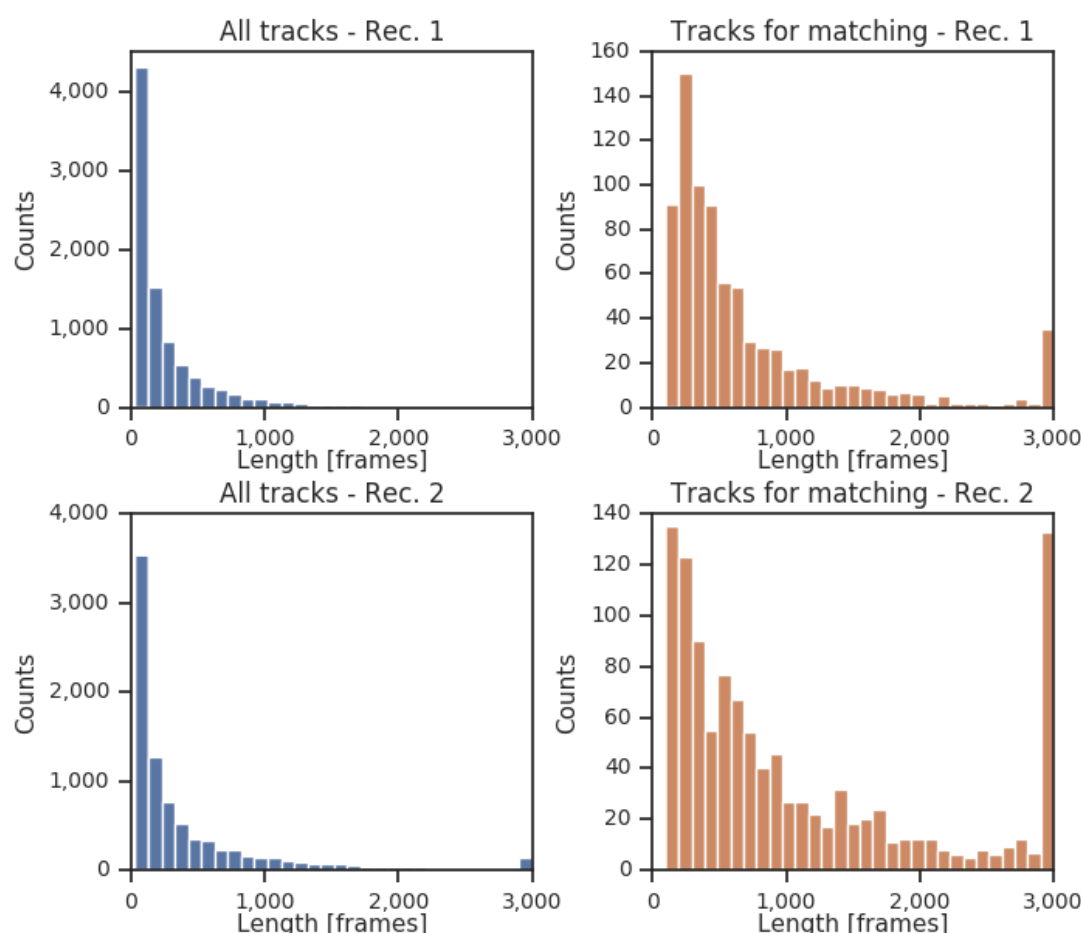


FIG. S5: Distributions of track fragment lengths. The Distributions of the lengths of track fragments obtained through position matching are shown for recording 1 (upper panels) and 2 (bottom panels). All tracks are included in the panels on the left, right-side panels show lengths of track fragments in the initial train set before the start of the track matching algorithm. Recording 2 contains a larger number of individuals with slower movement which resulted in a longer average track length compared to recording 1.

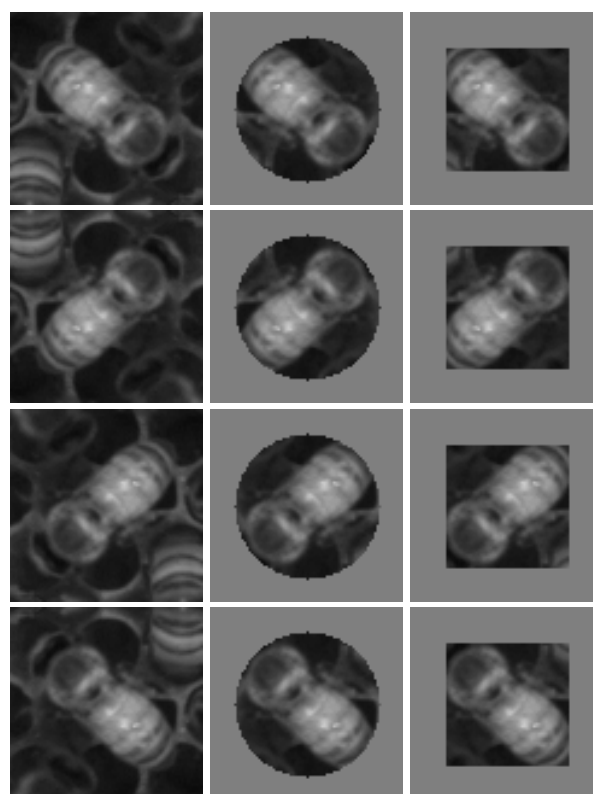


FIG. S6: Augmentation operations. All possible combinations of augmentation operations are shown on an example image (upper left image). These operations include flipping along 2 axes and round- or square-shaped masking of the outer parts of the image.

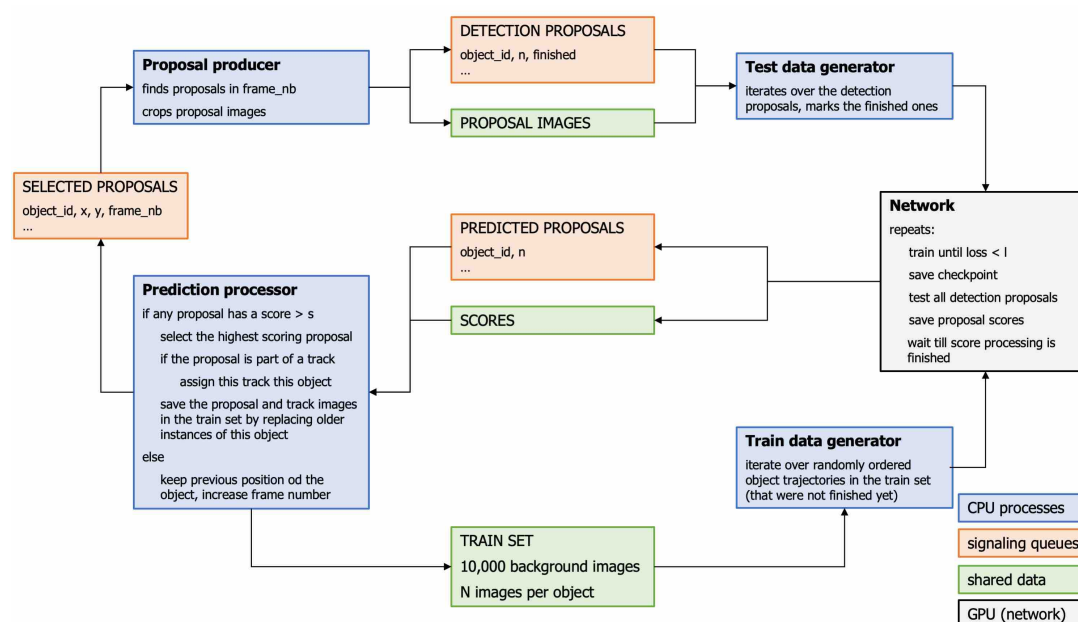


FIG. S7: Training and tracking loop implementation. Four process pools (blue boxes) are used for preparing images for the train and test and for processing of the network predictions. Queues (orange boxes) are used for process synchronization, images in the train and test sets are placed in shared data structures (green boxes).

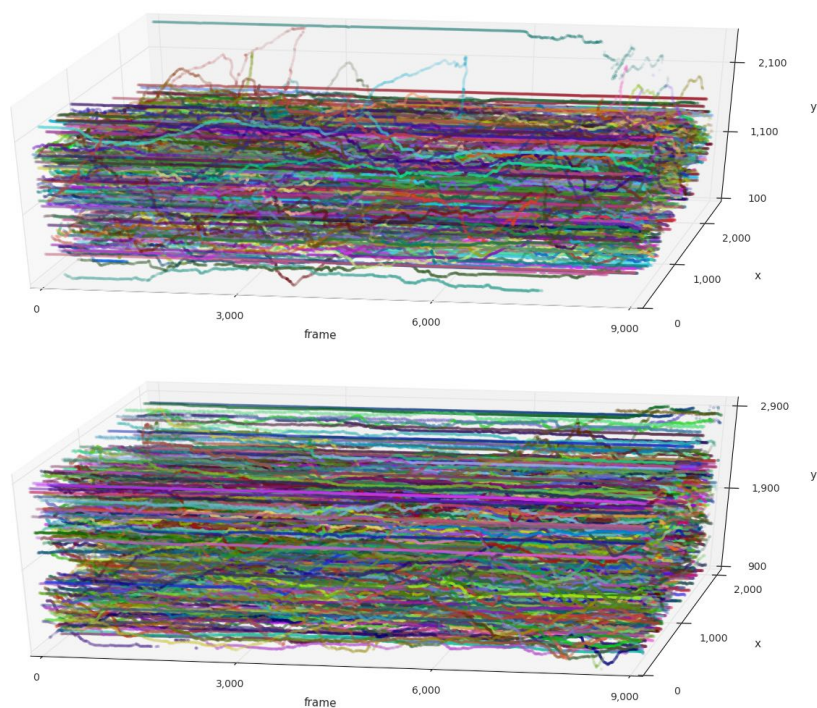


FIG. S8: The reconstructed trajectories of 4 min and longer in recording 1 (upper plot) and 2 (bottom plot).

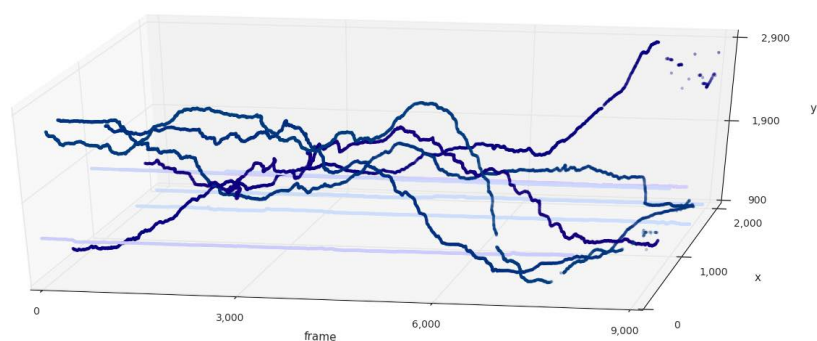


FIG. S9: Example trajectories of objects with different movement dynamics. Analogous to Fig. 4, examples of fast moving and stationary object trajectories from recording 2 are shown in dark and bright colors, respectively.