Automatic tracking of mouse social posture dynamics by 3D videography, deep learning and GPU-accelerated robust optimization

Social interactions powerfully impact both the brain and the body, but high-resolution descriptions of these important physical interactions are lacking. Currently, most studies of social behavior rely on labor-intensive methods such as manual annotation of individual video frames. These methods are susceptible to experimenter bias and have limited throughput. To understand the neural circuits underlying social behavior, scalable and objective tracking methods are needed. We present a hardware/software system that combines 3D videography, deep learning, physical modeling and GPU-accelerated robust optimization. Our system is capable of fully automatic multi-animal tracking during naturalistic social interactions and allows for simultaneous electrophysiological recordings. We capture the posture dynamics of multiple unmarked mice with high spatial (~2 mm) and temporal precision (60 frames/s). Our method is based on inexpensive consumer cameras and is implemented in python, making our method cheap and straightforward to adopt and customize for studies of neurobiology and animal behavior. We are currently combining our method with recordings of single neurons in the somatosensory cortex of freely interacting mice. Using our 3D tracking method, we can automatically relate the firing patterns of these neurons to social events (nose-to-nose touch, anogenital sniffing, mounting), posture and movement kinematics, postures and movement displayed by the social interaction partner, and spatial features (heading direction, spatial location).

Recording setup

We use inexpensive consumer depth cameras (Intel D435) to image socially interacting mice from four directions (**Fig. 1**). The cameras are synchronized to each other and to neural recording via TTL pulses and blinking LEDs. Our recording software (open source, written in python) captures depth images, color images (or – if visual darkness is preferred – infrared images), and automatically extracts the blinking LED pulses. At 60 fps, the camera frame rate is very stable (jitter across all cameras: ±26 µs, almost no dropped frames). Dropped frames and clock-drift between the cameras (~49 µs/min) is automatically detected and corrected for.

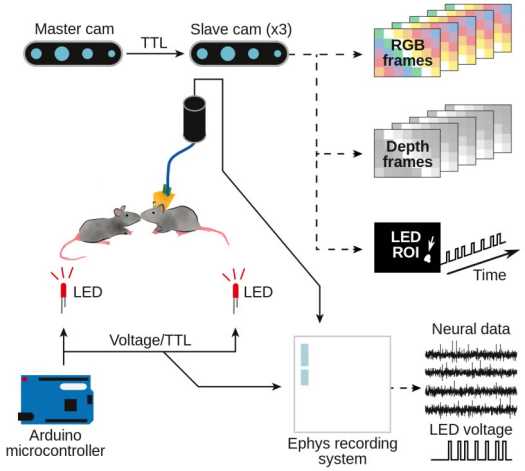

Fig 1. Schematic of our recording system

Data processing

From the depth images, we reconstruct the behavioral scene as a 3D pointcloud. In the color images, we use a convolutional neural network (stacked hourglass [1], trained with part affinity fields [2]) to detect four keypoints types: the mice's noses, ears, tails, and the neural implant. By aligning the depth and the color images, we can place these keypoints in 3D space (**Fig. 2**). For some social behaviors, our experiments are done in visual darkness, so we detect the keypoints in infrared images rather than color images, and use an augmentation pipeline to train the network to disregard the speckle pattern generated by the depth camera's infrared laser projectors.

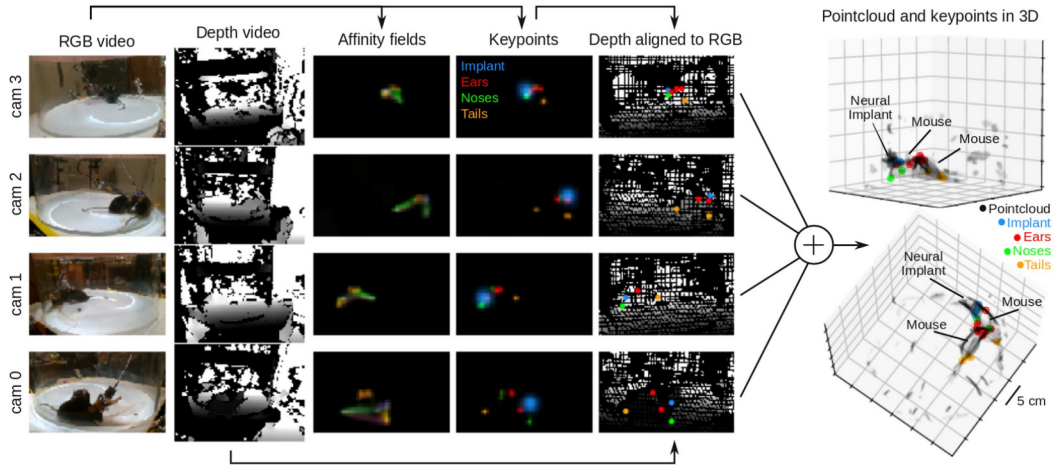

Fig 2. Using convolutional neural networks and 3D videography to estimate behavior

Fully automatic GPU-accelerated tracking algorithm

To estimate the mice's body postures and movements from the pre-processed 3D data, we use a combination of parametric shape modeling, particle filtering and Bayesian tracking. We model the body surface of the mice as stretchy ellipsoids (**Fig. 3**) and optimize a loss function, that makes the surface of the ellipsoids align to the 3D surface. The loss function has additional terms that use the detected keypoints and physical constraints to regularize the body models (the mouse bodies cannot overlap, nose keypoints must be near the tip of the nose ellpsoid, etc.). The loss function is minimized with a custom GPU-accelerated particle filter for every frame, thus robust to local minima, but provides a noisy estimate. Thus, we have a second Bayesian filter (kernel-recusive least squares [3]) that integrates noisy particle filter estimates across frames. This approach makes our tracking algorithm extremely robust, even in challenging situations, where the mice are closely interacting: The algorithm does not get stuck (thanks to the particle filter-based loss landscape exploration) and the algorithm does not flip the identity of the mice (because the across-frame filtering enforces spatiotemporal continuity).

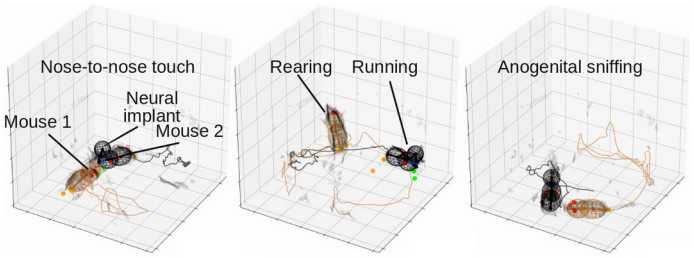

Fig 3. Parametric body models fitted to 3D data

Open source 3D social behavior analysis for the systems neuroscience community

We provide a comprehensive library python functions and example notebooks for recording, processing and viewing the behavioral data (**Fig 4.**). The multi-animal 3D posture/movement is rich and can be analyzed in multiple ways. For example, by defining behavioral events of a priori interest (e.g. nose-to-genital touch), by training a classifier to reproduce human annotation of behavioral categories (e.g. [4, 5]), or by unsupervised approaches that learn the behavioral categories from the data itself (e.g. [6–11]). For example, in our own work, we use the GPU-accelerated deep probabilistic language Pyro [12] to fit 'sticky' hidden Markov models [13] to movement kinematics to automatically classify movement states (**Fig. 5**). We are currently using the system with silicone probe recording from somatosensory cortex, relating spike patterns to social events, posture and movement kinematics, postures and movement displayed by the social interaction partner, and spatial features (heading direction, spatial location). For these experiments, it is key that our system not only classifies behavior, but <u>extracts continuous posture/movement data</u>. Neural activity is modulated by motor signals and vestibular signals across sensory cortices [14] and to understand how neural circuits process social cues during social interactions, these "low-level" motor and posture related confounds must be regressed out.

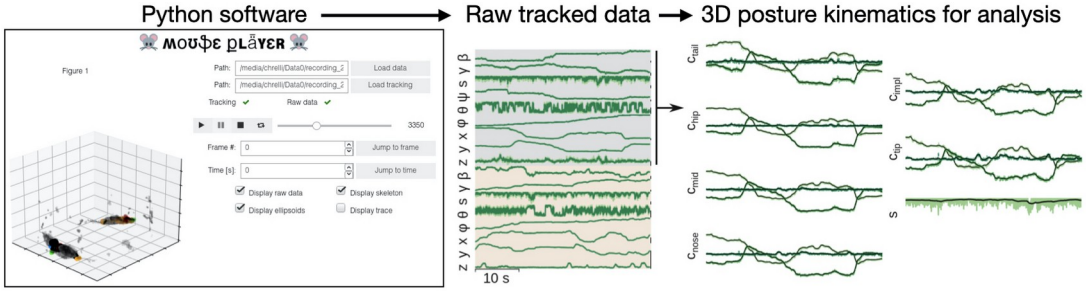

Fig 4. Python GUI for exploring 3D data, with automatic routines for exporting posture data

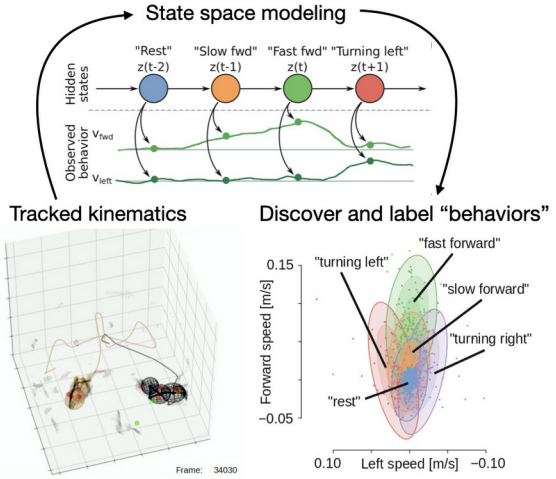

Fig 5. Using state space modeling to classify behavior, e.g. here: 5 running/turning states

[1] Newell et al. 2016 ArXiv160306937 Cs, [2] Cao et al. 2017 ArXiv161108050 Cs, [3] Lázaro-Gredilla et al. 2011 IEEE Int. Workshop Mach. Learn. Signal Process. Pp 1–6 [4] Nilsson et al. bioRxiv 2020.04.19.049452 [5] Segalin et al. bioRxiv 2020.07.26.222299 [6] Berman et al. J R Soc Interface 2014 11:20140672, [7] Wiltschko et al. 2015 Neuron 88:1121–1135 [8] Calhoun et al. 2019 Nat Neurosci 22:2040–2049 [9] Johnson et al. 2020 Curr Biol 30:70-82.e4 [10] Werkhoven et al. 2019 BioRxiv 779363 [11] Tao et al. 2019 eLife 8:e41235 [12] Bingham et al. 2018 ArXiv181009538 Cs Stat [13] Fox et al.2011 Ann Appl Stat 5:1020–1056 [14] Parker et al. 2020 Trends Neurosci 43:581–595