Lab Animal | 246 VOL 50 | September 2021 | 246–254 | www.nature.com/laban
https://doi.org/10.1038/s41684-021-00811-1

# A review of 28 free animal tracking software: current features and limitations

Veronica Panadeiro[1], Alvaro Rodriguez[1,2,3,*], Jason Henry[4], Donald Wlodkowic[4] and Magnus Andersson[5]

## Abstract

Well-quantified laboratory studies can provide a fundamental understanding of animal behavior in ecology, ethology and ecotoxicology research. These types of studies require observation and tracking of each animal in well-controlled and defined arenas, often for long timescales. Thus, these experiments produce long time-series and a vast amount of data that require software applications to automate the analysis and reduce manual annotation. In this Review, we examine 28 free software applications for animal tracking to guide researchers in selecting the software that might best suit a particular experiment. We also review the algorithms in the tracking pipeline of the applications, explain how specific techniques can fit different experiments, and, finally, expose each approach's weaknesses and strengths. Our in-depth review includes last update, type of platform, user-friendliness, off- or online video acquisition, calibration method, background subtraction and segmentation method, species, multiple arenas, multiple animals, identity preservation, manual identity correction, data analysis and extra features. We found, for example, that out of 28 programs, only three include a calibration algorithm to reduce image distortion and perspective problems that affect accuracy and can result in substantial errors when analyzing trajectories and extracting mobility or explored distance. In addition, only four programs can directly export in-depth tracking and analysis metrics, only five are suited for tracking multiple unmarked animals for more than a few seconds and only 11 have been updated in the period 2019-2021.

## Introduction

Animal behavior studies are fundamental in ecology, ethology, ecotoxicology, neuroscience and many other fields[1,2]. These studies can be performed in a wide variety of ways, ranging from observational tests in natural conditions to experimental trials in a laboratory environment. To compare experiments in laboratory conditions, performed by different research groups and with different organisms, it is important that these types of experiments are implemented in carefully controlled conditions and that they use standardized and repeatable protocols. Therefore, these experiments often use model organisms that have been widely studied, such as zebrafish or rodents, and take place in well-defined environments, so-called arenas. The size of the arena and the number of animals in an arena should also be carefully considered and controlled not to bias the study's outcome.

Standardized tests can measure an organism's activity in different arenas (Fig. 1a). For example: open arenas and plus-mazes are common in anxiety or motivation studies[3]; T-maze or

* Corresponding author. E-mail: a.tajes@udc.es
[1] Department of Computer Science, University of A Coruña. Campus de Elviña, 15071-A Coruña, Spain.
[2] Center for Research in Information and Communication Technologies (CITIC), University of A Coruña. Campus de Elviña, 15071-A Coruña, Spain.
[3] Department of Basic Psychology I, UNED, Ciudad Universitaria, 28040-Madrid, Spain.
[4] The Neurotox Laboratory, School of Science, RMIT University, 3083-Melbourne, VIC, Australia.
[5] Department of Physics, Umeå University, 90187-Umeå, Sweden.

Lab Animal | 246 VOL 50 | September 2021 | 246–254 | www.nature.com/laban
https://doi.org/10.1038/s41684-021-00811-1

Y-maze test arenas are commonly used for memory and spatial learning studies[4]; and 3-chambered arenas are used to measure social approach or fear responses[5]. Other standard tests include water mazes[6], elevated mazes[7], arenas with light-dark transitions[8] and arenas with thermal gradients or hot plates[9]. Stimuli or distractors can also be added to these arenas to measure behavioral changes in different conditions[10]. Although these arenas have different layouts, the way the data (video files) are acquired is similar (Fig. 1b).

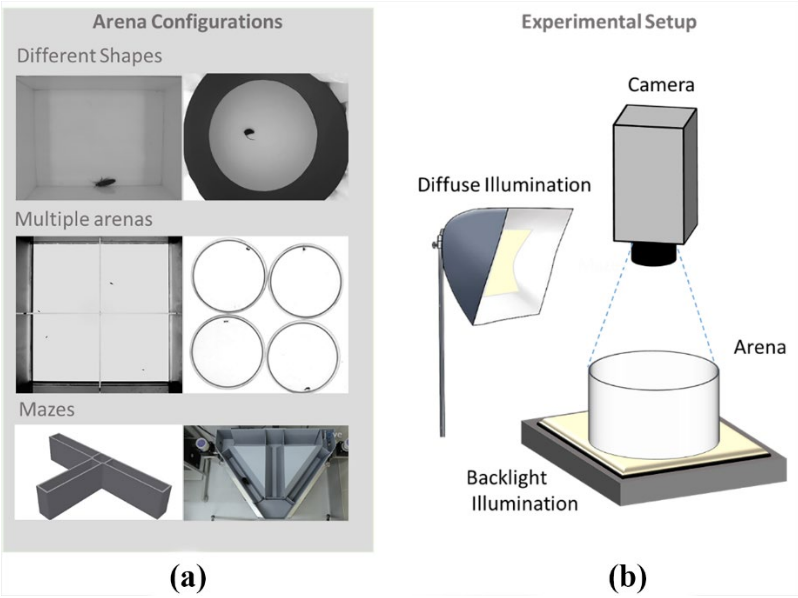


**Fig. 1| Overview of typical setups used in tracking experiments. a,** Examples of different arena configurations commonly used for insects, rodents and fish. **b,** Experiments in laboratory conditions should be carefully designed for front or back illumination depending on the type of organism under study and the arena used. Also, consider using diffuse illumination if details in the appearance of the animals are important, or backlight illumination to achieve high contrast for motion tracking. Make sure that no reflections from the surrounding environment or the lamp are seen.

Regardless of the arena design or organism used, most behavioral studies use video recording and analysis techniques in which the video framerate, field of view as well as the resolution are adjusted to capture the organism's motion with the required temporal and spatial resolution[11]. These experiments can produce a vast amount of image data, which can be very time-consuming to analyze, especially if manual annotation is used. Also, manual annotation can introduce human errors and biases, which can reduce the accuracy of the results[12]. Therefore, objective and automatic approaches for animal analysis are needed. Software and algorithms that can track and analyze the organism's position under study are critical for efficient research.

In general, these software packages can help to solve two main issues: detecting the position of animals via tracking algorithms; or detecting the positions of the animals' parts, so-called pose estimation. Pose estimation typically requires a previous tracking analysis. Therefore, as tracking is the main focus of this work, we refer only to the tracking software and the relevant tracking stages of pose estimation software. We exclude explicitly pose estimation software such as DeepPoseKit and DeepLabCut that are based on the extraction of images with distinct postures and the manual annotation of body parts to train machine learning models[13–15]. We think that these techniques are more oriented towards the detection of the behavior of a single animal at a fine scale and are not as directly relevant for general tracking applications. Also, they are sufficiently different in their theoretical approach to grant a separated analysis in another Review.

Tracking applications are available as free or commercial tools. Commercial software usually offer more features and flexibility, especially regarding input video formats and statistical outputs. However, many researchers cannot afford these expensive tools and the algorithms used by these software are often not transparent to the user. Thus, we focus this analysis on free tracking software. Several new types and versions are published each year. The continuous release of new software makes it challenging to select the appropriate software for a particular experiment or understand each option's limitations and differences. For example, many applications use the same pipeline and the same processing techniques, but there might be steps in the algorithm that are a limiting factor for some types of experiments. In addition, some programs are limited and specific in what they can do and what they can analyze.

With these considerations, we conducted our analysis using a systematic search of tracking applications on Google Scholar and Research Gate, using the software published from 2008 unti 2020 as inclusion criteria. We based our analysis primarily on the descriptions of the software in their respective papers and the published results from their authors. Additionally, we tested some of the analyzed software to gather information that was not available from these sources.

In this Review, we compare 28 tracking applications in a comprehensive and accessible way to help and guide researchers within the field of behavior studies. In contrast to previous reviews that focused on specific research fields and organisms[16,17], or explored only specific features such as social interactions[18], or did not include an in-depth discussion[19], or included only a handful of software[18,19], we point out the weaknesses in every step of the pipeline, identifying the problems that might face behavioral researchers when choosing a specific software. Also, we provide a detailed taxonomy of the main algorithms and their limitations, and show the emerging opportunities in animal tracking software development.

## General pipeline of a tracking program

To understand how a tracking application operates, it is essential to decompose the pipeline involved in the tracking process. Starting from video recording and ending in the analysis of animal trajectories, all the tracking applications we studied here use a similar sequence of algorithms. We show the general workflow of these tracking applications (see Fig. 2) and describe each step of the pipeline below. It is important to recall that some applications do not include all the steps shown in the figure. For example, some applications lack the calibration procedure (Fig. 2b) and the possibility to analyze trajectories (Fig. 2d).

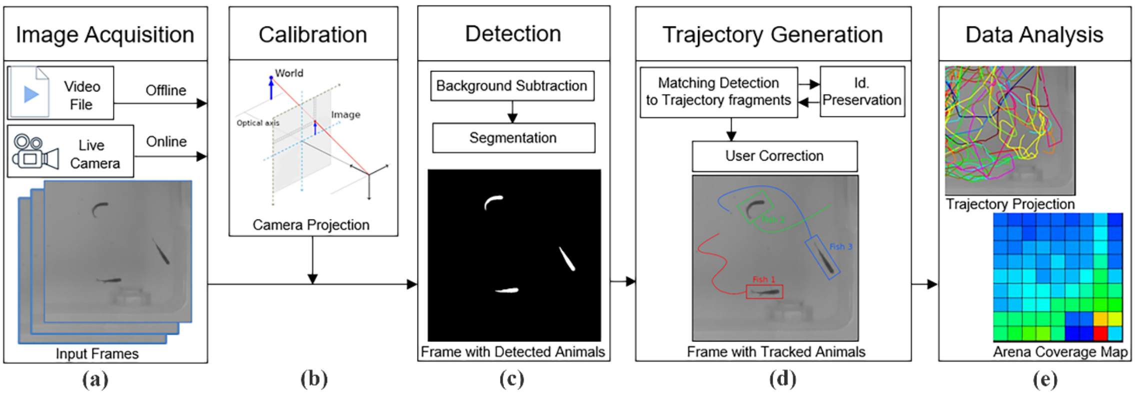


**Fig. 2| Illustration showing the general workflow of a tracking program; how an image frame is processed from video to analysis. a,** Image acquisition. **b,** Calibration. **c,** Detection. **d,** Trajectory generation. **e,** Data analysis. See text for full description.

**Image acquisition.** This step defines how video images are obtained. There are two main approaches: offline, in which a sequence of images feeds the algorithm from a video file; or online, in which a live camera streams each image. An advantage of using the offline mode is that complex algorithms can use extensive computations and access future and past frames to process a given image. The online method uses real-time computations, which require less storing capacity but increase the computational requirements, as each frame needs to be analyzed with speed similar to the camera framerate to avoid losing data. As a result, the online method is currently only suitable for its use with relatively simple algorithms.

**Calibration.** This step determines the relationship between the camera's natural units (pixels) and the real-world units. A calibration algorithm that can correct distortions in the images caused by misalignment, projection errors and the camera lens, is valuable.

**Detection.** This step consists of finding the spatial position of the animals in each image of a video sequence. Detection algorithms can include different steps, such as background subtraction to remove stationary features from the environment, segmentation to separate the objects that represent potential animals and a filtering step to remove false detections.

**Trajectory generation.** This step associates the detected animals in each image to the tracked individuals. In this step, a set of trajectories are generated, each defined by an animal's positions in the subsequent images of the video. Trajectory generation is a trivial task for a single animal in an open arena without occlusions or reflections. However, in arenas with uneven illumination conditions, with multiple targets, possible occlusions or complex backgrounds, trajectory generation requires identity preservation algorithms to track animals when an occlusion or a crossing occurs. In general, trajectory generation is a very complex task that usually requires manual correction, especially with a large number of animals.

**Data analysis.** Analysis involves extracting information from each trajectory to obtain the behavioral data required by the experiment. This extracted data consists of statistics related to moving velocity, distance, animal orientation, rate of explored territory or activity rate. Data analysis is usually a post-processing step done after all other tracking tasks are completed.

## Results and discussion

To compare how the 28 selected free programs handle the different steps in the tracking pipeline, we analyze and evaluate their main characteristics and functionalities for each step. Since there are no public datasets and standard metrics, we cannot make a direct quantitative comparison of their performance. In addition, these programs have been developed to use different video files, are optimized for specific arenas and are aimed at tracking different animals. Therefore, we point out their most important limitations, study the benefits and drawbacks of the techniques and algorithms, evaluate the areas that require further research, and finally, comment on the most important missing features of the software. We list these features in two tables that follow the tracking pipeline structure (see Fig. 3 and 4) and discuss each step below.

**Video Acquisition**. Behavioral experiments are often long and require a relatively large sample size; thus, they usually generate a vast amount of video data. The quality of the video data, and therefore the results extracted from it, are heavily dependent on experimental conditions such as illumination, camera position, camera resolution, optical quality and background color. The influence of these parameters in the behavioral results has rarely been studied, and most researchers are not familiar with the variables that need to be considered or how to adjust them[20].

It is also important to note that parameters such as resolution, codec configuration or frame rate, can affect the computational cost of tracking algorithms exponentially. It is recommended that each animal is composed of at least 50 pixels in a video[21,22] and that the frame rate is high enough, so the animal position overlaps in two consecutive frames. However, most tracking applications are limited by the maximum resolution that they can handle. For example, *EthoWatcher*[23,24] is restricted to a maximum video resolution of 320x240 pixels, and *MouseMove*[25] can handle only resolutions up to 640x480 pixels[22]. Thus, these applications are somewhat limited in the field of view and the number of organisms they can track if one wants to fulfill the requirements presented above. To our knowledge, only *anTraX*[26], *Automated Planar Tracking*[27] and *ToxTrac*[22,28,29] have been successfully tested with resolutions higher than 1920x1080. Other programs such as *Idtracker*[21,30] and *Idtracker.ai*[31,32] *or Ctrax*[33,34] are in our experience also able to handle videos with such resolutions.

Lab Animal | 246 VOL 50 | September 2021 | 246–254 | www.nature.com/laban
https://doi.org/10.1038/s41684-021-00811-1

**Fig. 3| Tracking software comparison: Last update, platform, video acquisition, calibration and detection.** * +: Bad; programing or high-level domain specific knowledge is required, the software needs to be compiled, has complex installation requirements or is difficult to use. ++: Ok; requires relevant domain specific knowledge to install and run. It is suitable for laboratories or academia. +++: Good; the software is easy to install and run and it is suitable for domestic use. *NA*: Information not available. CNN: convolutional Neural Network. GPU: graphic processing unit. MCR: Matlab Compiler Runtime. OS: Operating system.

| Software | Last update | Platform | User Friendly* | Image Acquisition | Calibration | Detection | | |
|---|---|---|---|---|---|---|---|---|
| | | | | | | Background Subtraction | Segmentation | Species |
| *ABC Tracker*[35,36] | 2020 | Windows | +++ | Offline | No | User selects background objects + morphological operations | User marks on the image + support vector machines (SVM) | Any |
| *Animapp*[47] | 2020 | Android, Windows, Mac, Linux | ++ | Offline | Yes (Manual scaling) | No | Thresholding | Any |
| *anTraX*[26] | 2020 | Mac, Linux, needs Matlab MCR | ++ | Offline | No | Averaged model | Thresholding | Small Insects |
| *Automated Planar Tracking*[27] | 2016 | Windows | + | Offline | No | No | Thresholding + head detection + shape model | Fish |
| *BEMOVI*[48,49] | 2015 | Windows, Mac, Linux, (needs ImageJ and R) | + | Offline | Yes (Manual scaling) | Dynamic difference image segmentation | Thresholding | Micro-organisms |
| *BioSense*[38,39] | 2018 | Windows, Mac, Linux | *NA* | Offline and online | Yes (Manual scaling) | Gaussian mixture model | Thresholding + filter by size | Any |
| *BioTrack*[77,78] | 2018 | Mac (experimental), Linux | + | Offline | No | Yes (*NA*) | Requires segmented input | Any (model required) |
| *Biotracker*[53] | 2018 | Windows, Mac, Linux | +++ | Offline | No | Averaged model | Ellipse fitting | Any |
| *Ctrax*[33,34] | 2009 | Windows, Mac, Linux | + | Offline | No | Gaussian mixture model | Thresholding + ellipse fitting | Flies |
| *EthoWatcher*[23,24] | 2012 | Windows | +++ | Offline and online | Yes (Manual scaling) | Reference frame | Thresholding | Any |
| *FIMTrack*[79,80] | 2014 | Windows, Mac, Linux | *NA* | Offline | No | No | Trained CNN | Any |
| *Fish CnnTracker*[58] | 2017 | Not available | *NA* | Offline | No | No | Thresholding + head detection | Fish |
| *Idtracker.ai*[31,32] | 2018 | Windows, Linux, needs GPU | ++ | Offline | No | Averaged model | Thresholding + filter by size | Any |
| *Idtracker*[21,30] | 2020 | Windows, needs Matlab MCR | +++ | Offline | No | Averaged model | Thresholding + filter by size | Any |
| *MARGO*[40,41] | 2020 | Windows, needs Matlab MCR | +++ | Offline and online | Yes (Manual scaling, automatic distortion correction) | Reference frame | Thresholding + filter by size | Any |
| *Mot*[64,65] | 2013 | Windows, Linux, needs Matlab | + | Offline | No | Averaged model | Thresholding + ellipse fitting | Mice |
| *Mouse Tracking*[56,57] | 2019 | OS not specified, needs Python and Tensorflow | + | Offline | No | Deep learning model | Ellipse fitting | Mice |
| *MouseMove*[25] | 2015 | Windows (needs LabView Runtime) | +++ | Offline | Yes (Manual scaling) | Reference video | Thresholding | Mice |
| *Multi-Animal Tracker*[59,60] | 2017 | Windows, Linux, needs Matlab | + | Offline | No | No | User marks on image + feature analysis + machine learning (*NA*) | Any |
| *Multi-Worm Tracker*[42,43] | 2013 | Windows, Linux, needs Anaconda | + | Offline and online | No | Decaying average model | Thresholding + filter by size | Worms |
| *Pathtrackr*[50,51] | 2019 | Windows, Mac | + | Offline | No | Reference frame | User mark + blob detection, bounding box analysis | Any |
| *QTrack*[12,74] | 2009 | Windows, Linux | + | Offline | Yes (*NA*) | Remove average color | Fitting Gaussian mixture model | Fly |
| *RAT*[75,76] | 2020 | Embedded in specific hardware | + | Online | No | No | Dynamic thresholding | Mice |
| *SpectralTL*[52] | 2014 | Windows, needs Matlab | + | Offline | No | Reference frame | Thresholding | Any |
| *SwisTrack*[44,45] | 2008 | Windows, Mac, Linux | +++ | Offline and online | Yes (Manual scaling, automatic distortion correction) | Reference frame, moving average background | Edge filtering | Any |
| *ToxTrac*[22,28,29] | 2021 | Windows | +++ | Offline | Yes (Manual scaling, automatic distortion correction) | Gaussian mixture model | Thresholding + filter by size | Any |
| *Tracktor*[19,68] | 2019 | Windows, Mac, Linux | + | Offline | No | No | Thresholding (adaptive) | Any |
| *UMATracker*[54,55] | 2019 | Windows, Mac | + | Offline | No | Gaussian mixture model | Thresholding | Any |

Red: functionality or characteristic not present or lacking when compared with the best software in the evaluated characteristic.
Light grey: limited functionality or characteristics, compared with the best software in the evaluated characteristic.

Lab Animal | **246** VOL 50 | September 2021 | 246–254 | www.nature.com/laban
https://doi.org/10.1038/s41684-021-00811-1

**Fig. 4| Tracking software comparison: Trajectory generation, data analysis and extra features.** *: Minimum-maximum number of animals and in brackets, the time the software is able to preserve their identity according to the results published by their authors. *NA*: Information not available. CNN: convolutional Neural Network.

| Software | Trajectory Generation | | | | Data Analysis | Extra Features |
|---|---|---|---|---|---|---|
| | Multiple Arenas | Multiple Animals (Time)* | Id. Preservation | Manual Id Correction | | |
| *ABC Tracker*[35,36] | *NA* | 2 - 30 (minutes) | Forward and backward particle filtering + stationary tracking robust to contact. | Yes | Direction | *NA* |
| *Animapp*[47] | No | No | No | No | Speed and distance | App for smartphone, batch processing in desktop version. |
| *anTraX*[26] | No | 2- 36 (hours) | Markers placed on animals + Deep learning classification (CNN) | No | Position, direction and speed | Can be combined with pose stimation software such as JAABA[71]. |
| *Automated Planar Tracking*[27] | No | 2 - 20 (seconds) | Head cross correlation | No | No | No |
| *BEMOVI*[48,49] | No | 2 – hundreds (seconds) | No | No | No | No |
| *BioSense*[38,39] | No | 2 - 8 (seconds) | No | No | Arena coverage, direction, regions of interest, speed and distance | No |
| *BioTrack*[77,78] | No | 2 - 3 (seconds) | No | No | No | No |
| *Biotracker*[53] | No | 2 - 11 (seconds) | No | No | No | Modular expandable framework |
| *Ctrax*[33,34] | No | 2 – 159 (seconds) | No | Yes | Behavior recognition (flies) | Batch processing |
| *EthoWatcher*[23,24] | No | No | No | No | Speed, distance, frequency and duration of each behavior. | User behavior annotation |
| *FIMTrack*[79,80] | No | 2 - 10 (none) | No | Yes | Position, direction and speed | No |
| *Fish CnnTracker*[58] | No | 2 - 11 (minutes) | Deep learning classification (CNN) | No | No | No |
| *Idtracker.ai*[31,32] | No | 2 (hours) - 100 (minutes) | Deep learning occlusion detection (CNN) + deep learning classification (CNN) + Bayesian analysis | Yes | No | Multiple files videos |
| *Idtracker*[21,30] | No | 2 - 20 (minutes) | Texture analysis + Bayesian analysis | Yes | No | Multiple file videos, allows multiple sessions |
| *MARGO*[40,41] | Yes | 2 - hundreds (seconds) | No | No | Position, size, direction and speed | Batch processing and hardware integration |
| *Motr*[64,65] | No | 2 – 6 (hours) | Markers placed on animals + Markov models + Bayesian analysis | No | Position, size and orientation | Allows multiple sessions |
| *Mouse Tracking*[56,57] | No | No | No | No | Speed and distance | No |
| *MouseMove*[25] | No | No | No | No | Activity, laterality, regions of interest, speed and distance | Batch processing |
| *Multi-Animal Tracker*[59,60] | No | 2 – *NA* (seconds) | No | No | Behavior recognition (worms), direction, regions of interest, speed and distance | No |
| *Multi-Worm Tracker*[42,43] | No | 2 - 120 (seconds) | No | No | Arena coverage, direction, population and individual info, postural information, speed and distance | No |
| *Pathtrackr*[50,51] | No | No | No | No | Arena coverage, direction, speed and distance | No |
| *QTrack*[12,74] | No | 2 (*NA*) | Separation of occlusions and contacts | No | Location, orientation and wing posture (flies) | Behavior analysis |
| *RAT*[75,76] | No | No | No | No | Position and speed | Hardware integration and wireless use |
| *SpectralTL*[52] | No | No | No | No | Speed and distance | No |
| *SwisTrack*[44,45] | No | 2 – 30 (seconds) | No | No | No | No |
| *ToxTrac*[22,28,29] | Yes | 2 - 11 (minutes) | Occlusion detection + texture analysis + Bayesian analysis + Hungarian optimization | No | Activity, arena coverage, population and individual info, regions of interest, speed and distance | Batch processing, multiple file videos |
| *Tracktor*[19,68] | No | 2 - 8 (*NA*) | K-means classification + Hungarian optimization | No | Direction, regions of interest, speed and distance | No |
| *UMATracker*[54,55] | No | 2 - 15 (*NA*) | Bayesian analysis, optical flow, K-means, prediction-correction | Yes | Interaction graph, regions of interest | Modular expandable framework |

Red: functionality or characteristic not present or lacking when compared with the best software in the evaluated characteristic.
Light grey: limited functionality or characteristics, compared with the best software in the evaluated characteristic.

Lab Animal | 246 VOL 50 | September 2021 | 246–254 | www.nature.com/laban
https://doi.org/10.1038/s41684-021-00811-1

The frame rate or the number of frames in a video file are also computational limiting factors. Since *Idtracker*[21,30] and *Idtracker.ai*[31,32] require high computation times for each frame even on moderate resolutions, analyzing large data sets using these software programs can be very time-consuming. Finally, *ABC tracker*[35,36] can only handle videos shorter than 10 minutes restricting the use of this software to short-time experiments[37].

The most important limitation of video acquisition is the processing speed and memory required to run tracking algorithms. This limitation is currently the bottleneck in behavioral experiments. The ability to process high-resolution high-framerate videos in real-time (online analysis) would revolutionize behavioral experiments by markedly decreasing the analysis timescale and reducing the need for video data storage. Currently, real-time tracking is only possible using simple algorithms that do not work with complex backgrounds and multiple targets. Out of the 28 applications, *BioSense*[38,39], *EthoWatcher*[23,24], *MARGO*[40,41], *Multi-Worm Tracker*[42,43] and *SwisTrack*[44,45] offer both online and offline video acquisition modes, whereas the other programs only operate in offline mode.

**Calibration**. Camera calibration is a process that allows the user to obtain measurements in real-world coordinates. In addition, calibration can also include removing image distortion and perspective errors, which occur when imaging a 2D surface with a fixed camera; it is important to recall that the distance from the center to the edges is not linearly increasing. Although calibration is a critical feature of animal tracking software to obtain reliable data, only 9,of the 28 analyzed programs offer a calibration function. See table 1, "Calibration" column.

Calibration techniques are commonly based on the use of the pinhole mathematical camera model[46] to solve the equations that describe the projection of a point in the real world to the image plane through the lens of an ideal camera. This model takes into account not only the pixel scale but also the rotation of the camera with respect to the arena, also allowing the estimation and removal of lens distortion. This technique requires to solve a complex equation system and to use a calibration pattern. Only *SwisTrack*[44,45], *MARGO*[40,41] and *ToxTrac*[22,28,29] implement this technique.

Most calibration techniques do not take advantage of the pinhole model and use a simple scale transformation. A scale transformation converts image coordinates to world coordinates by multiplying them by a constant factor. The programs *Animapp*[47], *BEMOVI*[48,49], *BioSense*[38,39], *EthoWatcher*[23,24], *MARGO*[40,41], *MouseMove*[25] and *ToxTrac*[22,28,29] use this approach. This technique is more straightforward for the user, but far less flexible and accurate since image distortion is not taken into account. Only *SwisTrack*[44,45], *ToxTrac*[22,28,29] and *MARGO*[40,41] have both calibration systems.

The lack of calibration options in available animal tracking software is surprising. We believe that this issue illustrates the deep gap between the considerations of the academic community that develops tracking software and the actual laboratory needs of software users'.

**Detection**. The detection step consists of finding the animals of interest in the images. In table 1, we divide the detection step into three different sections:

*Background subtraction.* Background subtraction algorithms aim to remove features of the environment that can interfere with animal detection. Background subtraction is a key feature when recording animals in a natural setting with dynamic lighting conditions or in aquatic environments where images are changed by reflections, shadows and other artifacts. Two most common types of background subtraction techniques are: background subtraction techniques based on static images, and background subtraction techniques based on dynamic models.

Static techniques commonly use a reference frame or video of the background without animals, such as in *EthoWatcher*[23,24], *Pathtrackr*[50,51], *MARGO*[40,41], *MouseMove*[25], *SpectralTL*[52] and *SwisTrack*[44,45]; or estimate a background by averaging the frames of the video, such as in *anTraX*[26], *Biotracker*[53], *Idtracker*[21,30] and *Idtracker.ai*[31,32]. Static background techniques are easy to implement and are effective when detecting stationary or moving animals, if the background

https://doi.org/10.1038/s41684-021-00811-1

objects and illumination does not change during the experiment. Otherwise, these techniques should not be used.

Dynamic techniques use moving or decaying average models, such as in *Multi-Worm Tracker*[42,43], *SwisTrack*[44,45]; or Gaussian mixture models, such as in *BioSense*[38,39], *Ctrax*[33,34], *ToxTrac*[22,28,29] and *UMATracker*[54,55]. Dynamic techniques can account for illumination changes or other gradual changes in the background. This factor is important when running long-time experiments in which the sun is used as a light source. However, these techniques are not reliable when detecting animals that remain static during a substantial part of the experiment.

The only program that approaches background subtraction with an innovative technique is *Mouse Tracking*[56,57]. *Mouse Tracking*[56,57] uses a deep learning algorithm to separate the pixels from the background and foreground. This strategy is more robust than other techniques but is also complex and requires a massive amount of training data, which makes the method very computationally heavy.

From our experience, there is room for improvement in background subtraction algorithms given thay only a few studies have addressed situations with low and/or changing contrast within the background.

*Segmentation.* Segmentation is performed immediately after background subtraction and usually consists of a technique aimed at separating potential animals in the image and a filtering step that removes possible false positives. The most common segmentation technique is based on so-called thresholding. Thresholding is a simple segmentation method that uses a reference value to separate pixels' regions of the image of different brightness. Thresholding is based on contrast and requires the animals to appear as bright objects in a dark background or as dark objects in a bright background. The main advantage of thresholding is that it is a very computationally efficient technique. However, thresholding is very sensitive to false-positives or false-negatives in non-uniform images. Most of the analyzed software use a variation of this technique.

To increase the sensitivity and robustness of detection, a few techniques locate specific animal-features on the image. This method limits the thresholding step given that it can only be applied to specific animals with certain body shapes and therefore can not be used for general tracking. Examples of software using this strategy are *Ctrax*[33,34] and *Mouse Tracking*[56,57], which use an ellipse fitting strategy to search for circular shaped objects. *Automated Planar Tracking*[27] and *Fish CnnTracker*[58], on the other hand, rely on locating the specific shape of the fish head to improve detection.

The only applications that provide some innovations in animal detection are *ABC Tracker*[35,36], *Multi-Animal Tracker*[59,60] and *Pathtrackr*[50,51]. These programs use a system that requires the user to mark each animal's location in a few frames on the video and use a machine learning technique to locate each animal in the remaining frames. This approach provides a more robust detection system than other algorithms. *ABC Tracker*[35,36] successfully uses this approach by applying support vector machines (SVM) and obtains robust results with a user-friendly experience.

*Species.* Most of the software studied are versatile and can be used for different species. Out of the 28 tested applications, 16, can be used for tracking any type of animal, whereas 11, are designed for a specific animal type and 1 require expanding the software with specific animal models (see column "Species" in table 1).

**Trajectory generation**. The challenge of trajectory generation is to associate potential targets with previous trajectories. That is, to associate a set of detections to a group of animals, where we know the trajectories of these animals before the current frame. The most common technique to solve this problem is by using the Kalman filter, which is a prediction-correction technique[61]. With this technique, one can estimate an animal's position in the next frame based on its previous known positions by assuming a constant speed or a constant acceleration model. Then, in the next frame, the predicted positions are compared with the actual detections using a Hungarian optimization technique[62]. The Kalman filter is very efficient computationally, and most tracking programs that we are aware of have implemented a variation of this algorithm or use similar techniques, for example, Particle Filters[63].

However, the Kalman filter and other similar techniques are not reliable in animal tracking scenarios where occlusions or multiple interacting animals exist. The reason for this limitation is that these techniques are not able to keep the identity of the objects and use only spatial information to match the trajectories. A typical example of a situation where these techniques fail is when two animals cross paths and change direction after the collision. When this situation occurs, the algorithm will lose track of the animals for a brief moment. It will then search for the animals' new positions, assuming they continued moving in the same direction. Asa consequence, animals' identities will be switched.

Preserving the identity of multiple individuals (Id. preservation) after an occlusion is currently the main limitation in the trajectory generation step. The complexity of this problem is illustrated in a 2014 study[21], in which Pérez-Escudero and colleagues analyzed a scenario with multiple interacting animals. In this scenario, when solving correctly 99% of all crossings, only 11% of the animals were correctly identified after 2 minutes of tracking owing to error propagation[22]. In summary, preserving the identity is complex and computationally expensive, and only a few offline tracking applications offer major contributions to this field.

*Motr*[64,65] uses an Id. preservation that relies on marking the individuals with visually distinctive markers that can be easily identified automatically. This technique is reliable and allows tracking animals for long periods of time and in multiple sessions. However, many modern techniques try to avoid placing markers or sensors on the animals given that the markers can be impractical and sometimes interfere with the experiment by affecting animal behavior[66,67]. Programs such as *Automated Planar Tracking*[27], *Tracktor*[19,68] and *UMATracker* offer Id. preservation algorithms that in our opinion have only marginally improved the basic Kalman strategy and are not robust and reliable for some specific situations.

*Idtracker*[21,30] and *ToxTrac*[22,28,29] use a strategy based on a probabilistic texture analysis to analyze animals' similarity between collisions. *Idtracker*[21,30] was one of the first applications that seriously approached this issue when tracking multiple targets and uses a complex algorithm based on a Bayesian analysis with a similarity metric to compare the objects' texture. *ToxTrac*[22,28,29], on the other hand, uses a combination of a similarity analysis with a Hungarian algorithm to manage the identity preservation of multiple targets. This technique builds on top of a very fast tracking algorithm that can handle simultaneous tracking in multiple arenas, resulting in one of the most flexible free tracking tools for trajectory generation.

Traditional probabilistic texture analysis is not capable of tracking many targets or for very long times, but the approach is very useful in short experiments with small groups of animals, for which body shape and appearance do not change much in comparison to their position or posture. Based on the results reported by *ToxTrac*[22,28,29] and *Idtracker*[21,30], we recommend using these programs with groups of up to 5 animals in videos no longer than 20 minutes. Between these two techniques, especially for users with limited computational speed, we think *ToxTrac*[22,28,29] is a better alternative because it requires substantially less processing time[19,28].

*Idtracker.ai*[31,32], and *Fish CnnTracker*[58] use an approach based on deep learning models called convolutional neural networks (CNNs). CNNs are optimized for image classification tasks and are among the most powerful image classification techniques nowadays, outperforming those based on traditional probabilistic texture analysis. The current drawbacks of these models are that they require specific training data and cannot be used in real-time applications even with optimized hardware, such as graphics processing units (GPUs). While some techniques such as transfer learning can mitigate some of these issues, none of the software in the list use this approach.

*Idtracker.ai*[31,32] is an extension of *Idtracker*[21,30], which combines two different deep learning algorithms: one to detect occlusions and one to identify targets with classification analysis. *Idtracker*[21,30] and *Idtracker.ai*[31,32] present the most solid identity preservation techniques on the market. However, the computational time required to analyze a standard experiment can be as high as one hour per frame if not using expensive GPU computing hardware, making it unpractical to run on a standard laboratory computer. *Fish CnnTracker*[58] offers a less flexible approach only suitable for fish tracking and is also less accurate[28] than *Idtracker.ai*[31,32]. *Idtracker.ai*[31,32] achieves the best accuracy in Id. preservation to date. In a set of experiments, Romero-Ferrero et al. manually reviewed a significant sample of the crossings for individual animals, and *Idtracker.ai*[31,32] was able to track up to 100 zebrafish and fruit flies for 10 minutes, or 4 mice for one hour[31]. However, in our opinion, the computation times this technique requires makes *Idtracker.ai*[31,32] not suitable for most common scenarios, given that many behavioral labs do not have a high-end computer with a state of the art GPU, and that processing an hour video with *Idtracker.ai*[31,32], with a modern GPU, can take more than one day of computation in extreme situations[32]. Despite this limitation, it is important to note that *Idtracker.ai*[31,32] is the best of the reviewed software to analyze experiments with large groups of unmarked animals.

*ABC Tracker*[35,36] offers a novel approach that uses forward and backward particle filtering. A particle filter is a selection-prediction-measurement solution of similar complexity to the Kalman filter. The particle filter uses a set of samples called particles to estimate the internal states in dynamical systems from partial observations and with random perturbations. This filter is also easy to parallelize and can be more accurate than Kalman filter[69]. In our opinion, the main novelty of *ABC Tracker*[35,36] is that it combines with this prediction scheme an algorithm to track stationary objects based on a local search strategy that, in practice, is able to solve most animal interactions when the scenario does not involve complex occlusions in a 3D space. *ABC Tracker*[35,36] can currently track up to 30 animals in a video for less than 10 minutes[37], obtaining very good results in these situations. In our opinion, *ABC Tracker*[35,36] has also the most intuitive interface for analyzing experiments with multiple individuals.

Finally, *anTraX*[26]*,* combines the use of color tags to mark individuals, CNNs and a graph-based approach. According to the authors, these features should allow the tracking of dozens of marked individuals for hours, if not days. However, given the the use of tags and the lack of validation results forspecific timeit is unclear how robust is the algorithm compared to other techniques.

In summary, Id. preservation algorithms for multiple interacting animals are still insufficient when looking at the computational performance and accuracy that are required for behavioral tests and analysis. So far, Id. preservation has not been accomplished when running multiple arenas simultaneously, with animal sizes smaller than 50 pixels or with online image acquisition techniques.

**Data Analysis**. Nowadays, automatic behavior recognition can be achieved using annotated video datasets to train machine-learning classifiers[70]. Kabra et al. proposed an automatic animal behavior annotator that led to the creation of individual and social behavior classifiers for organisms, such as mice and larval flies[71]. Robie et al. proposed a similar strategy to detect patterns of social interactions[18]. Using these techniques, some works studied complex behaviors such as mating and feeding in mice[72], or behavioral responses of larval fish to chemicals[73]. Despite this development, only a minority of the analyzed software include functions for complex behavior recognition, and they are always limited to specific behaviors in particular scenarios. For example, *Multi-Animal Tracker*[59,60] was used to detect pirouette movements in worms, *Ctrax*[33,34] was used to detect touch and chase social behaviors in flies, *QTrack*[12,74] detects specific courtship behaviors also in flies; and finally, *MouseMove*[25] can quantify unilateral locomotor deficits in mice.

In general, tracking applications provide only movement metrics. Thus, most tracking applications are limited to movement, orientation, and zone exploration metrics. Furthermore, only a fraction of the assessed software in this Review provides a useful array of these metrics, that is: *BioSense*[38,39], *MouseMove*[25], *Multi-Animal Tracker*[59,60], *Multi-Worm Tracker*[42,43] and *ToxTrac*[22,28,29]. These software provide advanced toolkits that allow non-programmers to analyze parameters such as movement, time spent in selected areas, changes in direction, and time spend

Lab Animal | 246 VOL 50 | September 2021 | 246–254 | www.nature.com/laban
https://doi.org/10.1038/s41684-021-00811-1

moving. An important note is that these software packages can also provide individual or population metrics.

We conclude that more work is needed to integrate behavior recognition in tracking software, and we believe that algorithms for automatic detection of stress and other complex behaviors would represent a true innovation if included in tracking tools.

**Extra Features**. Some tracking applications implement extra features to facilitate user experience or to add versatility. The most useful extra feature, from our point of view, is the ability to analyze video files that have been split into multiple files. This feature is explicitly supported currently by *Idtracker*[21,30], *Idtracker.ai*[31,32] and *ToxTrac*[22,28,29]. Another important feature is the possibility of processing a batch of video files using the same camera configuration, allowing the user to adjust the parameters only once for a set of experiments. Only *Animapp*[47], *Ctrax*[33,34], *idTracker*[21,30], *MARGO*[40,41], *MouseMove*[25] and *ToxTrac*[22,28,29] implement this functionality.

The possibility of controlling hardware peripherals such as external sensors, lights or temperature sources is a nice feature implemented in *MARGO*[40,41] and *RAT*[75,76]. This feature allows measuring parameters such as reaction times to stimuli, planning long-term tests without supervision and modifying the stimuli according to behaviors creating a closed control loop.

*UMATracker*[54] and *BioTracker*[53] implement a modular approach to facilitate the development and integration of new processing modules. We think that this addition can be useful in collaboration with the community to integrate new features into the software. However, taking full advantage of this feature requires a constant level of support and commitment that may not be realistic.

Finally, we would like to highlight that *Animapp*[47] includes an Android application that directly analyzes images using the smartphone camera. With the increased computational power of smartphones, we believe that this type of program can be useful for simple studies performed in a field environment.

## Conclusions

Current tracking software need to balance robustness, accuracy, and processing speed. As a rule of thumb, higher robustness and accuracy require complex algorithms that reduce processing speed. Therefore, online processing programs use simple algorithms that increase efficiency to reach real-time performance, but with the trade-off of less robustness and accuracy. Offline software can take advantage of more complex processing algorithms but require a computational capacity that is not achievable for all users or suitable for every experiment. We tried to highlight these differences and the contributions of each application to the animal tracking field.

Our assessment shows that all applications share more-or-less the same pipeline and very few of them offer a unique or revolutionary approach; the use of CNN networks for Id. Preservation being the most relevant new contribution. However, our biggest concern is the lack of usability of recent software. Out of the 28 tested programs, only four: *IdTracker*[21,30], *IdTracker.ai*[31,32], *ABC Tracker*[35,36] and *Toxtrac*[22,28,29] provide innovative algorithms, useful features and user-friendly interfaces. We believe that the main reason for this lack of usability is the existence of a gap between software design and their intended use in a laboratory. Most software packages are not easy-to-use, require tuning of several and complex parameters for each experiment, and do not include important features such as calibration options. In addition, most programs do not offer data analysis tools beyond the most basic ones and cannot extract valuable behavioral metrics. In our opinion, developers of tracking tools must change their paradigm from creating programs that can be published to creating programs that are useful and easy to use.

Finally, we want to draw attention to the lack of complete, open, and well-labeled datasets that provide a standard reference for validation and accuracy testing. Such datasets would give researchers an objective tool for a quantitative comparison of tracking programs[6].

**Author Contributions**

A.R. co-wrote the manuscript and performed the majority of the analysis of the tracking software. V.P. co-wrote the manuscript and assisted with the analysis of the tracking software. J.H., D.W. and M.A. revised and edited the manuscript and assisted with the analysis of the tracking software.

**Competing interests**

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

**Publisher's note**: Springer nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

## References

1. Dell, A. I. *et al.* Automated image-based tracking and its application in ecology. *Elsevier* **29**, 417–428 (2014).
2. Hajar, R. Animal testing and medicine. *Hear. Views* **12**, 42 (2011).
3. Carola, V., D'Olimpio, F., Brunamonti, E., Mangia, F. & Renzi, P. Evaluation of the elevated plus-maze and open-field tests for the assessment of anxiety-related behaviour in inbred mice. *Behav. Brain Res.* **134**, 49–57 (2002).
4. Olton, D. S. Mazes, maps, and memory. *Am. Psychol.* **34**, 583 (1979).
5. Silverman, J. L., Babineau, B. A., Oliver, C. F., Karras, M. N. & Crawley, J. N. Influence of stimulant-induced hyperactivity on social approach in the BTBR mouse model of autism. *Neuropharmacology* **68**, 210–222 (2013).
6. Cirulli, F., Berry, A. & Alleva, E. Intracerebroventricular administration of brain-derived neurotrophic factor in adult rats affects analgesia and spontaneous behaviour but not memory retention in a Morris Water Maze task. *Neurosci. Lett.* **287**, 207–210 (2000).
7. Borta, A. & Schwarting, R. K. W. Inhibitory avoidance, pain reactivity, and plus-maze behavior in Wistar rats with high versus low rearing activity. *Physiol. Behav.* **84**, 387–396 (2005).
8. Kulesskaya, N. & Voikar, V. Assessment of mouse anxiety-like behavior in the light--dark box and open-field arena: role of equipment and procedure. *Physiol. Behav.* **133**, 30–38 (2014).
9. Lee, H., Iida, T., Mizuno, A., Suzuki, M. & Caterina, M. J. Altered thermal selection

[6] The software *BioTrack*[77,78] and *FIMTrack*[79,80] is not specifically addressed in the text, and only referenced in Fig 3. And Fig 4.

behavior in mice lacking transient receptor potential vanilloid 4. *J. Neurosci.* **25**, 1304–1310 (2005).

10. Woodley, C. M., Urbanczyk, A. C., Smith, D. L. & Lemasson, B. H. Integrating Visual Psychophysical Assays within a Y-Maze to Isolate the Role that Visual Features Play in Navigational Decisions. *JoVE (Journal Vis. Exp.* e59281 (2019).

11. Jonsson, M. *et al.* High-speed imaging reveals how antihistamine exposure affects escape behaviours in aquatic insect prey. *Sci. Total Environ.* **648**, 1257–1262 (2019).

12. Dankert, H., Wang, L., Hoopfer, E. D., Anderson, D. J. & Perona, P. Automated monitoring and analysis of social behavior in Drosophila. *Nat. Methods* **6**, 297–303 (2009).

13. Pereira, T. D. *et al.* Fast animal pose estimation using deep neural networks. *Nat. Methods* **16**, 117–125 (2019).

14. Graving, J. M. *et al.* DeepPoseKit, a software toolkit for fast and robust animal pose estimation using deep learning. *Elife* **8**, e47994 (2019).

15. Mathis, A. *et al.* DeepLabCut: markerless pose estimation of user-defined body parts with deep learning. *Nat. Neurosci.* **21**, 1281–1289 (2018).

16. Franco-Restrepo, J. E., Forero, D. A. & Vargas, R. A. A Review of Freely Available, Open-Source Software for the Automated Analysis of the Behavior of Adult Zebrafish. *Zebrafish* **16**, zeb.2018.1662 (2019).

17. Husson, S. J., Costa, W. S., Schmitt, C. & Gottschalk, A. Keeping track of worm trackers. in *WormBook: The Online Review of C. elegans Biology [Internet]* (WormBook, 2018).

18. Robie, A. A., Seagraves, K. M., Egnor, S. E. R. & Branson, K. Machine vision methods for analyzing social interactions. *J. Exp. Biol.* **220**, 25–34 (2017).

19. Sridhar, V. H., Roche, D. G. & Gingins, S. Tracktor: image-based automated tracking of animal movement and behaviour. *Methods Ecol. Evol.* (2018).

20. Henry, J., Rodriguez, A. & Wlodkowic, D. Impact of digital video analytics on accuracy of chemobehavioural phenotyping in aquatic toxicology. *PeerJ* **7**, e7367 (2019).

21. Pérez-Escudero, A., Vicente-Page, J., Hinz, R. C., Arganda, S. & de Polavieja, G. G. idTracker: tracking individuals in a group by automatic identification of unmarked animals. *Nat. Methods* **11**, 743–748 (2014).

22. Rodriguez, A. *et al.* ToxTrac: A fast and robust software for tracking organisms. *Methods Ecol. Evol.* **9**, 460–464 (2018).

23. Junior, C. F. C. *et al.* ETHOWATCHER: validation of a tool for behavioral and video-tracking analysis in laboratory animals. *Comput. Biol. Med.* **42**, 257–264 (2012).

24. Junior, C. F. C. *et al.* EthoWatcher. http://ethowatcher.paginas.ufsc.br/ (2019).

25. Samson, A. L. *et al.* MouseMove: an open source program for semi-automated analysis of movement and cognitive testing in rodents. *Sci. Rep.* **5**, (2015).

26. Gal, A., Saragosti, J. & Kronauer, D. J. C. anTraX, a software package for high-throughput video tracking of color-tagged insects. *Elife* **9**, e58145 (2020).

27. Wang, S. H., Cheng, X. E., Qian, Z.-M., Liu, Y. & Chen, Y. Q. Automated planar tracking the waving bodies of multiple zebrafish swimming in shallow water. *PLoS One* **11**, e0154714 (2016).

28. Rodriguez, A., Zhang, H., Klaminder, J., Brodin, T. & Andersson, M. ToxId: an efficient algorithm to solve occlusions when tracking multiple animals. *Sci. Rep.* **7**, 14774 (2017).

29. Rodriguez, A. *et al.* Toxtrac. https://sourceforge.net/projects/toxtrac/ (2019).

30. Pérez-Escudero, A., Vicente-Page, J., Hinz, R. C., Arganda, S. & de Polavieja, G. G. idTracker. http://www.idtracker.es/ (2019).

31. Romero-Ferrero, F., Bergomi, M. G., Hinz, R., Heras, F. J. H. & de Polavieja, G. G. idtracker.ai: Tracking all individuals in large collectives of unmarked animals. *Nat. Methods* **16**, 179–182 (2019).

32. Romero-Ferrero, F., Bergomi, M. G., Hinz, R., Heras, F. J. H. & de Polavieja, G. G. idtracker.ai. https://idtracker.ai/ (2019).

33. Branson, K., Robie, A. A., Bender, J., Perona, P. & Dickinson, M. H. High-throughput ethomics in large groups of Drosophila. *Nat. Methods* **6**, 451–457 (2009).

34. Branson, K., Robie, A. A., Bender, J., Perona, P. & Dickinson, M. H. Ctrax. http://ctrax.sourceforge.net/ (2019).

35. Shin, M. C. ABC Tracker. http://abctracker.org/ (2018).

36. Rice, L. A. A beginning-to-end system for efficiently gathering tracking data on multiple targets. (The University of North Carolina at Charlotte, 2016).

37. David Farynyk. ABC Tracker Support. https://abctracker.atlassian.net/wiki/spaces/ABCTS/pages/458795/FAQ (2020).

38. Patman, J., Michael, S. C. J., Lutnesky, M. M. F. & Palaniappan, K. BioSense: Real-Time Object Tracking for Animal Movement and Behavior Research. in *2018 IEEE Applied Imagery Pattern Recognition Workshop (AIPR)* 1–8 (2018).

39. Patman, J., Michael, S. C. J., Lutnesky, M. M. F. & Palaniappan, K. BioSense. https://www.mantisrt.com/ (2019).

40. Werkhoven, Z., Rohrsen, C., Qin, C., Brembs, B. & de Bivort, B. MARGO (Massively Automated Real-time GUI for Object-tracking), a platform for high-throughput ethology. *PLoS One* **14**, e0224243 (2019).

41. Werkhoven, Z., Rohrsen, C., Qin, C., Brembs, B. & de Bivort, B. MARGO. https://github.com/de-Bivort-Lab/margo (2020).

42. Swierczek, N. A., Giles, A. C., Rankin, C. H. & Kerr, R. A. High-throughput behavioral analysis in C. elegans. *Nat. Methods* **8**, 592 (2011).

43. Swierczek, N. A., Giles, A. C., Rankin, C. H. & Kerr, R. A. Multi-Worm Tracker. https://sourceforge.net/projects/mwt/ (2019).

44. Correll, N. *et al.* SwisTrack: A tracking tool for multi-unit robotic and biological systems. in *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems* 2185–2191 (2006).

45. Mario, E. Di *et al.* SwisTrack. https://sourceforge.net/projects/swistrack (2019).

46. Zhang, Z. A flexible new technique for camera calibration. *IEEE Trans. Pattern Anal. Mach. Intell.* **22**, 1330–1334 (2000).

47. Rao, S. R. *et al.* Small Animal Video Tracking for Activity and Path Analysis Using a Novel Open-Source Multi-Platform Application (AnimApp). *Sci. Rep.* **9**, (2019).

48. Pennekamp, F., Schtickzelle, N. & Petchey, O. L. BEMOVI, software for extracting

behavior and morphology from videos, illustrated with analyses of microbes. *Ecol. Evol.* **5**, 2584–2595 (2015).

49. Pennekamp, F., Schtickzelle, N. & Petchey, O. L. BEMOVI. http://bemovi.info/ (2015).

50. Harmer, A. M. T. & Thomas, D. B. pathtrackr: An r package for video tracking and analysing animal movement. *Methods Ecol. Evol.* **00**, 1–7 (2019).

51. Harmer, A. M. T. & Thomas, D. B. Pathtrackr. https://github.com/aharmer/pathtrackr (2019).

52. Madan, C. R. & Spetch, M. L. Visualizing and quantifying movement from pre-recorded videos: the spectral time-lapse (STL) algorithm. *F1000Research* **3**, (2014).

53. Mönck, H. J. *et al.* Biotracker: An open-source computer vision framework for visual animal tracking. *arXiv Prepr. arXiv1803.07985* (2019).

54. Yamanaka, O. & Takeuchi, R. UMATracker: an intuitive image-based tracking platform. *J. Exp. Biol.* **221**, jeb182469 (2018).

55. Yamanaka, O. & Takeuchi, R. UMATracker. http://ymnk13.github.io/UMATracker/ (2019).

56. Geuther, B. Q. *et al.* Robust mouse tracking in complex environments using neural networks. *Commun. Biol.* **2**, 124 (2019).

57. Geuther, B. Q. *et al.* Mouse Tracking. https://github.com/KumarLabJax/MouseTracking (2019).

58. Zhiping, X. U. & Cheng, X. E. Zebrafish tracking using convolutional neural networks. *Sci. Rep.* **7**, 42815 (2017).

59. Itskovits, E., Levine, A., Cohen, E. & Zaslaver, A. A multi-animal tracker for studying complex behaviors. *BMC Biol.* **15**, 29 (2017).

60. Itskovits, E., Levine, A., Cohen, E. & Zaslaver, A. Multi-Animal Tracker. https://github.com/itskov/MultiAnimalTrackerSuite (2019).

61. Cuevas, E. V, Zaldivar, D. & Rojas, R. Kalman filter for vision tracking. *Freie Univ. Berlin Tech. Pap.* (2005) doi:10.17169/refubium-22852.

62. Kuhn, H. W. The Hungarian method for the assignment problem. *Nav. Res. Logist. Q.* **2**, 83–97 (1955).

63. Del Moral, P. Nonlinear filtering: Interacting particle resolution. *Comptes Rendus l'Académie des Sci. I-Mathematics* **325**, 653–658 (1997).

64. Ohayon, S., Avni, O., Taylor, A. L., Perona, P. & Egnor, S. E. R. Automated multi-day tracking of marked mice for the analysis of social behaviour. *J. Neurosci. Methods* **219**, 10–19 (2013).

65. Ohayon, S., Avni, O., Taylor, A. L., Perona, P. & Egnor, S. E. R. Motr. http://motr.janelia.org/ (2019).

66. Rodriguez, A. *et al.* Optical fish trajectory measurement in fishways through computer vision and artificial neural networks. *J. Comput. Civ. Eng.* **25**, 291–301 (2011).

67. Rodriguez, A., Bermúdez, M., Rabuñal, J. & Puertas, J. Fish tracking in vertical slot fishways using computer vision techniques. *J. Hydroinformatics* (2014).

68. Sridhar, V. H., Roche, D. G. & Gingins, S. Tracktor. https://github.com/vivekhsridhar/tracktor (2019).

69. Rao, G. M. & Satyanarayana, C. Visual Object Target Tracking Using Particle Filter: A Survey. *Image, Graph. Signal Process.* 57–71 (2013) doi:10.5815/ijigsp.2013.06.08.

70. Datta, S. R. Q&A: Understanding the composition of behavior. *BMC Biol.* **17**, 1–7 (2019).

71. Kabra, M., Robie, A. A., Rivera-Alba, M., Branson, S. & Branson, K. JAABA: interactive machine learning for automatic annotation of animal behavior. *Nat. Methods* **10**, 64 (2013).

72. Linck, V. de M. *et al.* Effects of inhaled Linalool in anxiety, social interaction and aggressive behavior in mice. *Phytomedicine* **17**, 679–683 (2010).

73. Steele, W. B., Mole, R. A. & Brooks, B. W. Experimental protocol for examining behavioral response profiles in larval fish: application to the neuro-stimulant caffeine. *JoVE (Journal Vis. Exp.* e57938 (2018).

74. Dankert, H. QTrack (Cadabra). http://www.vision.caltech.edu/cadabra/ (2009).

75. Krynitsky, J. *et al.* Rodent Arena Tracker (RAT): A Machine Vision Rodent Tracking Camera and Closed Loop Control System. *eNeuro* **7**, (2020).

76. Krynitsky, J. *et al.* RAT. https://hackaday.io/project/162481-rodent-arena-tracker-rat (2020).

77. Feldman, A., Hybinette, M. & Balch, T. The multi-iterative closest point tracker: An online algorithm for tracking multiple interacting targets. *J. F. Robot.* **29**, 258–276 (2012).

78. Hrolenok, B., Quitmeyer, A., Motter, S., Stolarsky, D. & Migliozzi, B. L. R. Bio-tracking. http://www.bio-tracking.org/ (2012).

79. Risse, B., Berh, D., Otto, N., Klämbt, C. & Jiang, X. FIMTrack: An open source tracking and locomotion analysis software for small animals. *PLoS Comput. Biol.* **13**, e1005530 (2017).

80. Risse, B., Berh, D., Otto, N., Klämbt, C. & Jiang, X. FIMTrack. https://www.uni-muenster.de/PRIA/en/FIM/download.shtml (2017).