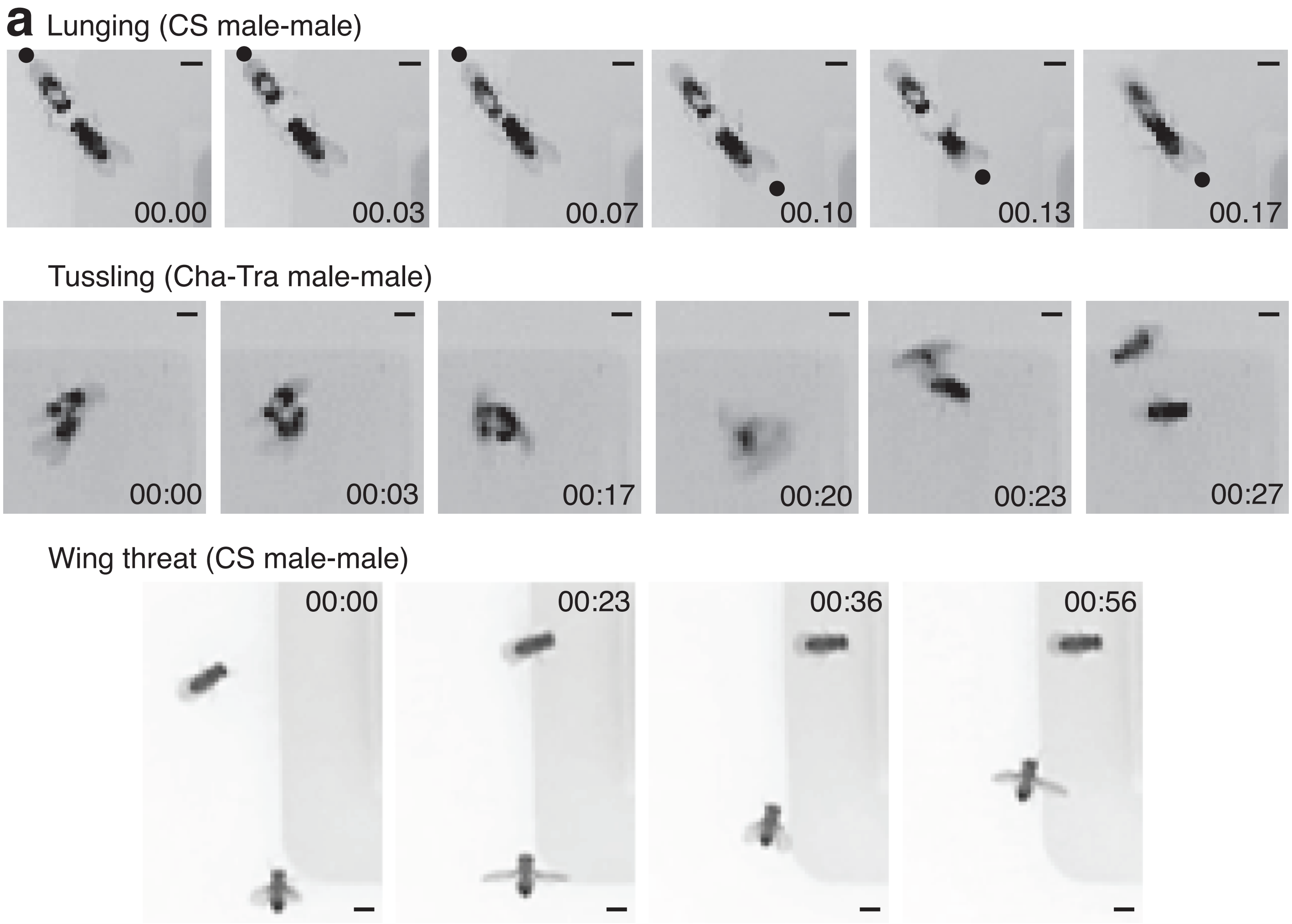
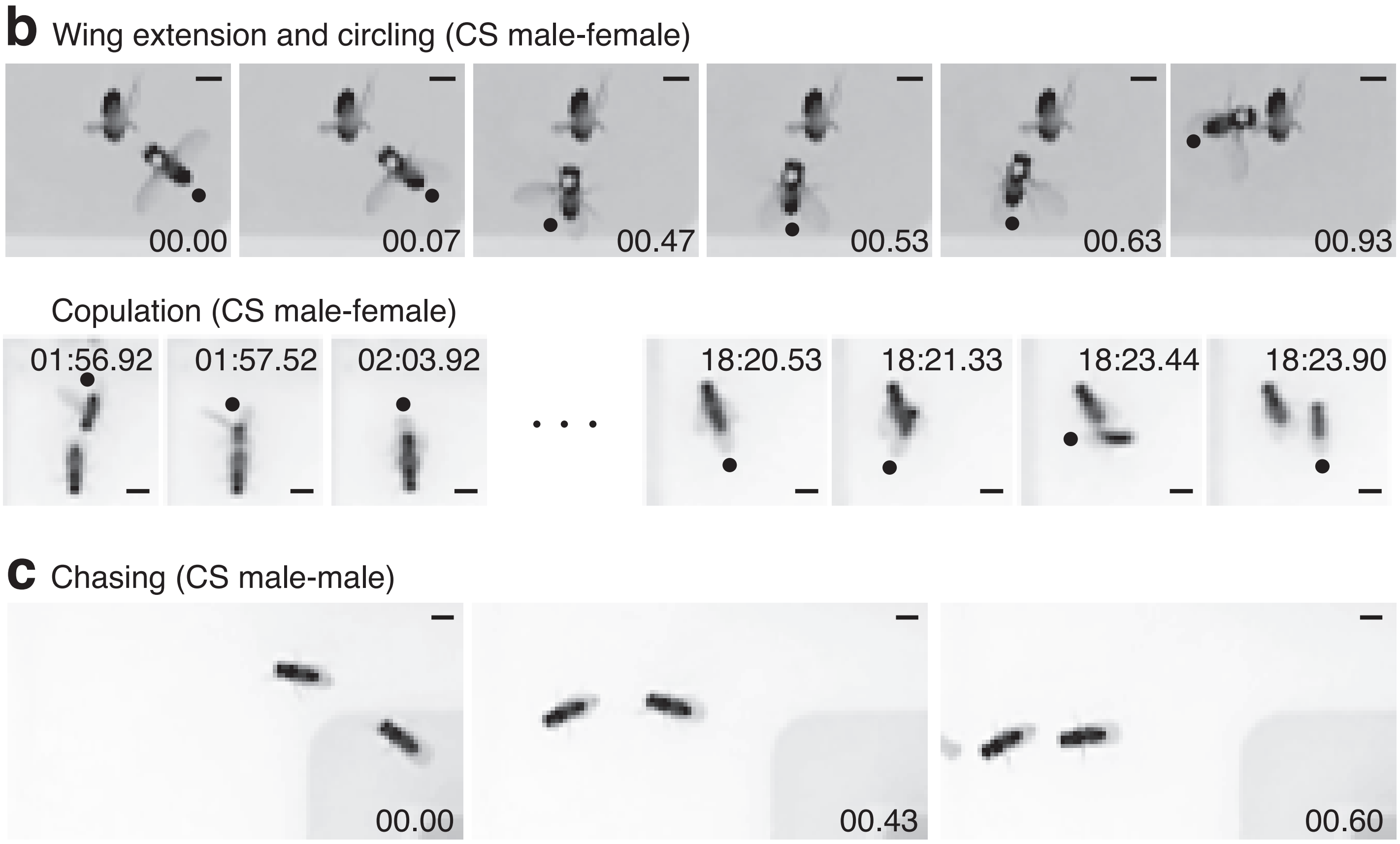
Supplementary Figure 1



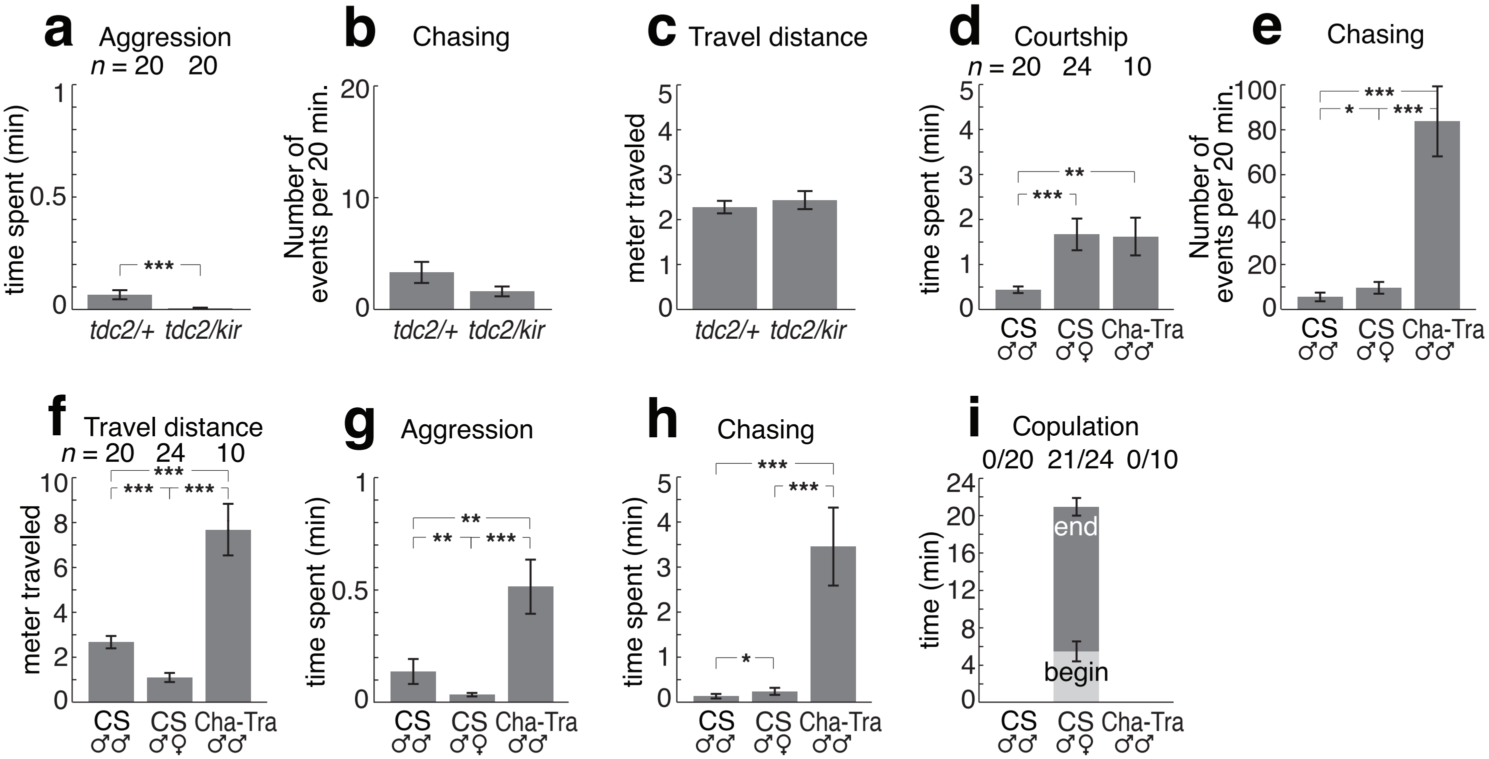
Supplementary Figure 1: Sequences of automatically detected aggressive actions, courtship actions, and chasing. (**a**) Lunging, tussling and wing threat. Only a selection of frames is shown. Time index in seconds is relative to the first frame in each action movie clip. A black dot is placed near the fly that is performing an action. Frames are shown at half of the original resolution. Bars, 1 mm.

Supplementary Figure 1

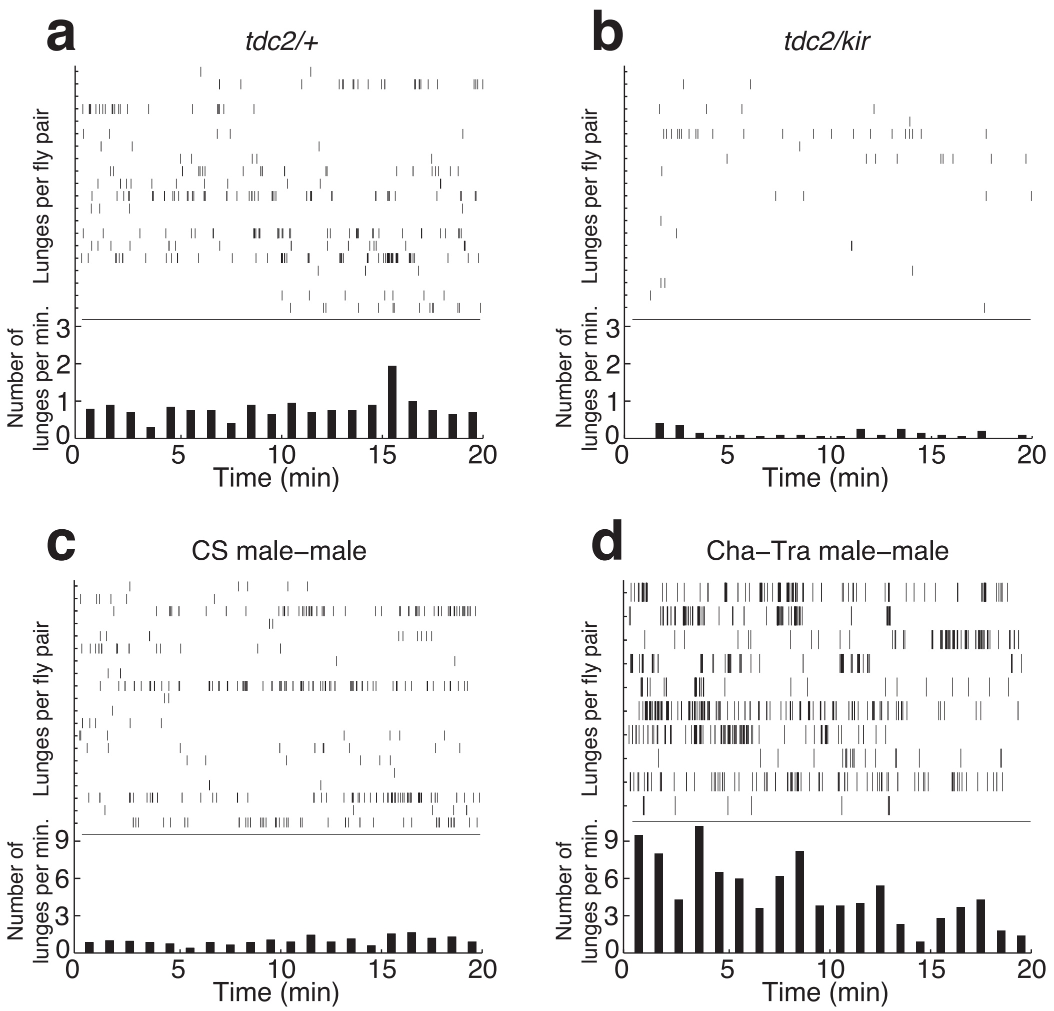


Supplementary Figure 1: Sequences of automatically detected aggressive actions, courtship actions, and chasing. (**b**) Wing extension, circling and copulation (start and end phase). Time index is relative to the first frame in each action movie clip. For copulation the time index is absolute, starting with the beginning of the assay [MM:SS] (minutes, seconds). As illustrated by the example, wing extension can occur in combination with circling. Copulation usually lasts for 16-18 minutes. (**c**) Chasing is common to both aggression and courtship. Only a selection of frames is shown. Time index in seconds is relative to the first frame in the movie clip. A black dot is placed near the fly that is performing an action. Frames are shown at half of the original resolution. Bars, 1 mm.

Supplementary Figure 2

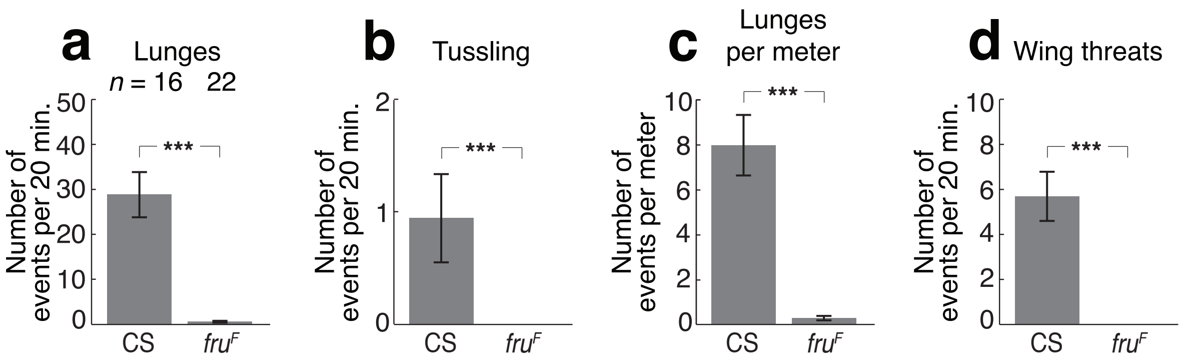


Supplementary Figure 2: Frequency and time spent performing actions. (**a**-**c**) Octopamine control (*tdc2/+*, *n* = 20) and mutant (*tdc2/kir*, *n* = 20). (**d**-**i**) CS male-male (*n* = 20), male-female (*n* = 24), and *Cha-Gal4;UAS-tra* ("Cha-Tra", *n* = 10). (**a**,**g**) Total time spent in aggressive activity. (**b**,**e**) Number of chases. (**c**,**f**) Total distance traveled per fly. (**d**) Total time spent in courtship. (**h**) Total time spent chasing. The octopamine mutant (*tdc2/kir*) shows a lower level of aggressiveness compared to the control line (**a**). The octopamine mutant shows no statistical significant difference in locomotor activity, in comparison to controls (**b**,**c**). CS and Cha-Tra male-male pairs differ by their level of locomotion (**e**,**f**). Cha-Tra males are significantly more active than CS males (**h**). (**i**) Copulation beginning and end time points. 21 out of 24 CS male-female pairs successfully copulated. Data represent mean ± s.e.m. $^*P < 0.05$, $^{**}P < 0.01$, $^{***}P < 0.001$.

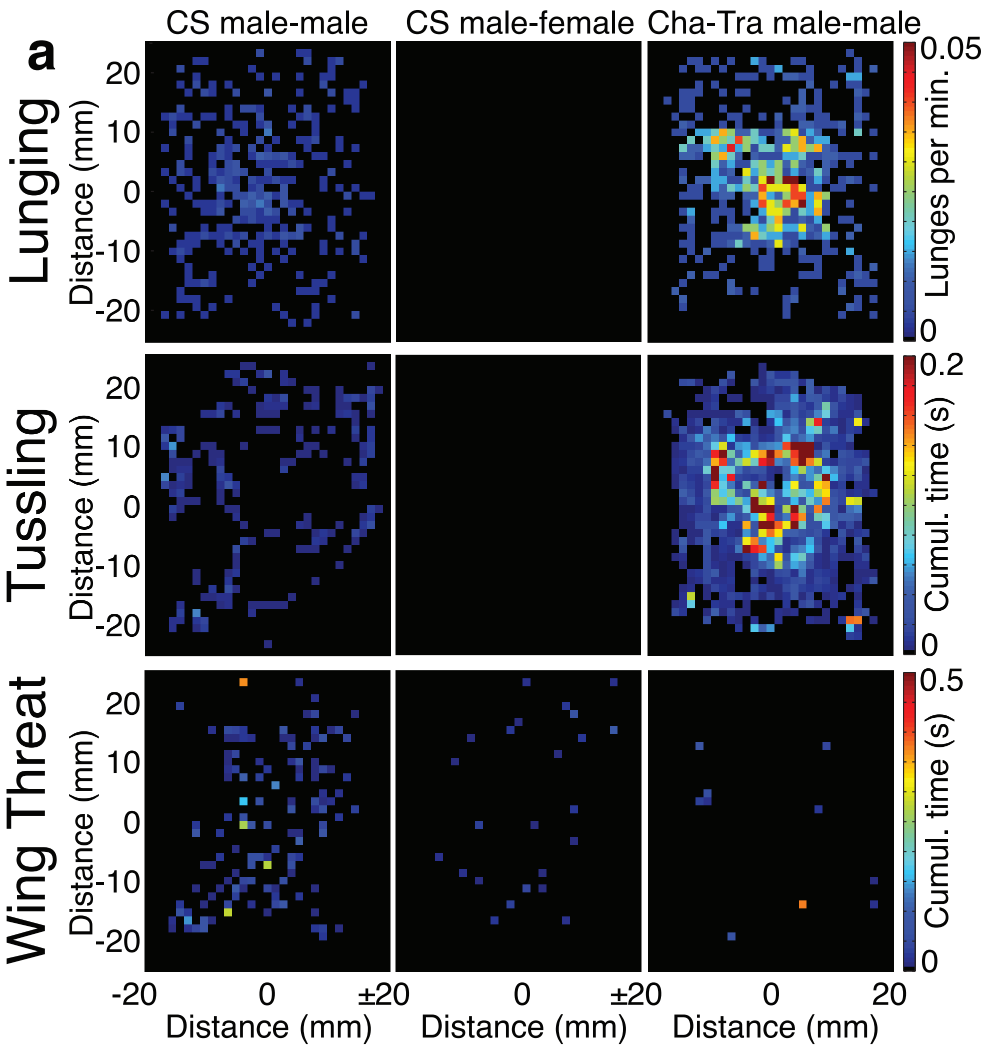


Supplementary Figure 3. Distribution of lunges over time. (**a**,**b**) Octopamine control (*tdc2/+*, *n* = 20) and mutant (*tdc2/kir*, *n* = 20). (**c**,**d**) CS male-male (*n* = 20) and *Cha-Gal4;UAS-tra* ("Cha-Tra", *n* = 10) pairs. Each spike in the upper part of the panels represents one lunge, and each row of spikes one fly pair. The histogram integrates the number of lunges over the fly pairs in one minute bins. The octopamine mutant (*tdc2/kir*) (**b**) shows a lower level of aggressiveness compared to the control line (**a**). Cha Tra males (**d**) are significantly more aggressive than CS males (**c**).

Supplementary Figure 4

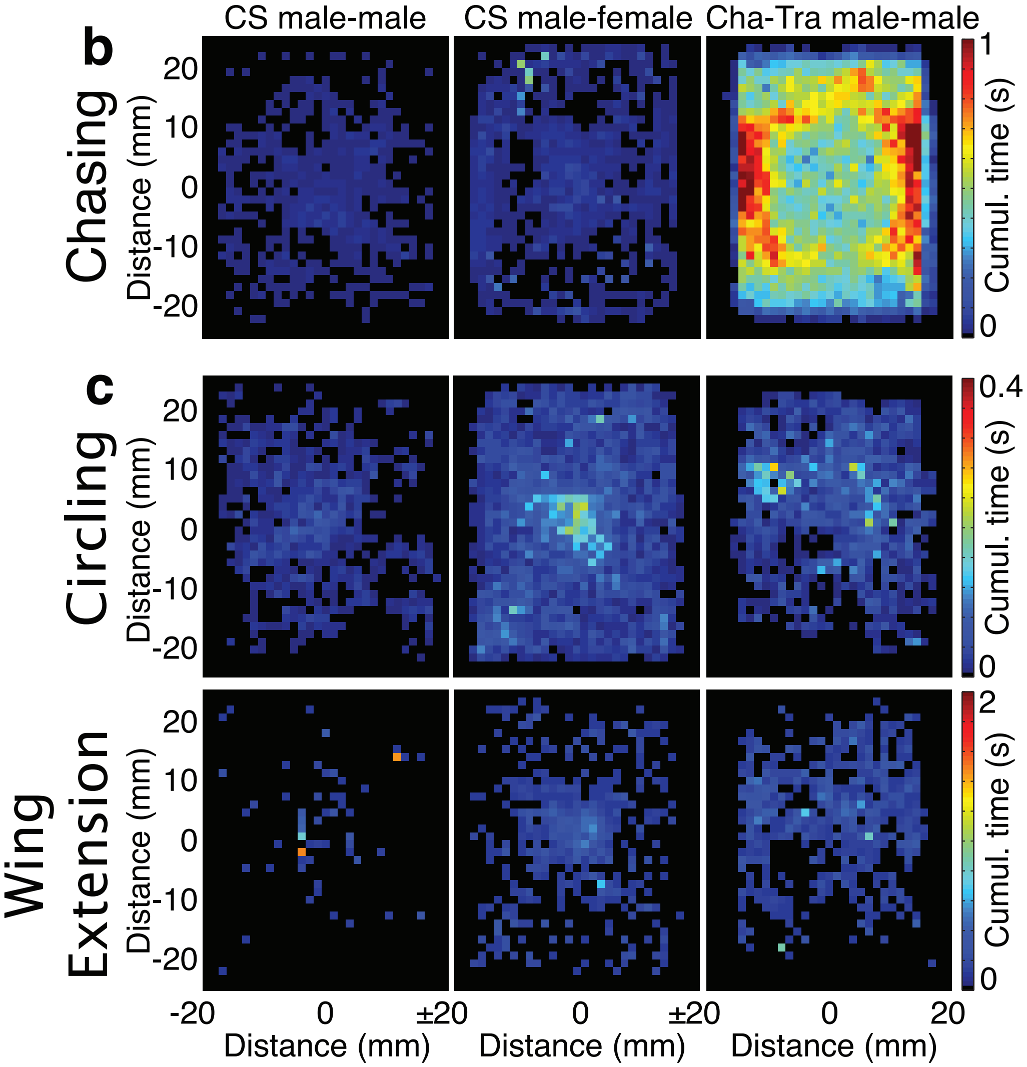


Supplementary Figure 4. Frequency and time spent performing actions for *fru*^{F}. (**a**) Number of lunges, (**b**) tussling, (**c**) lunges per meter, and (**d**) wing threats. *fru*^{F} shows a significantly lower level of aggressiveness compared to the control line. Data represent mean ± s.e.m. (CS: n = 16, *fru*^{F}: n = 22). ***$P < 0.001$.



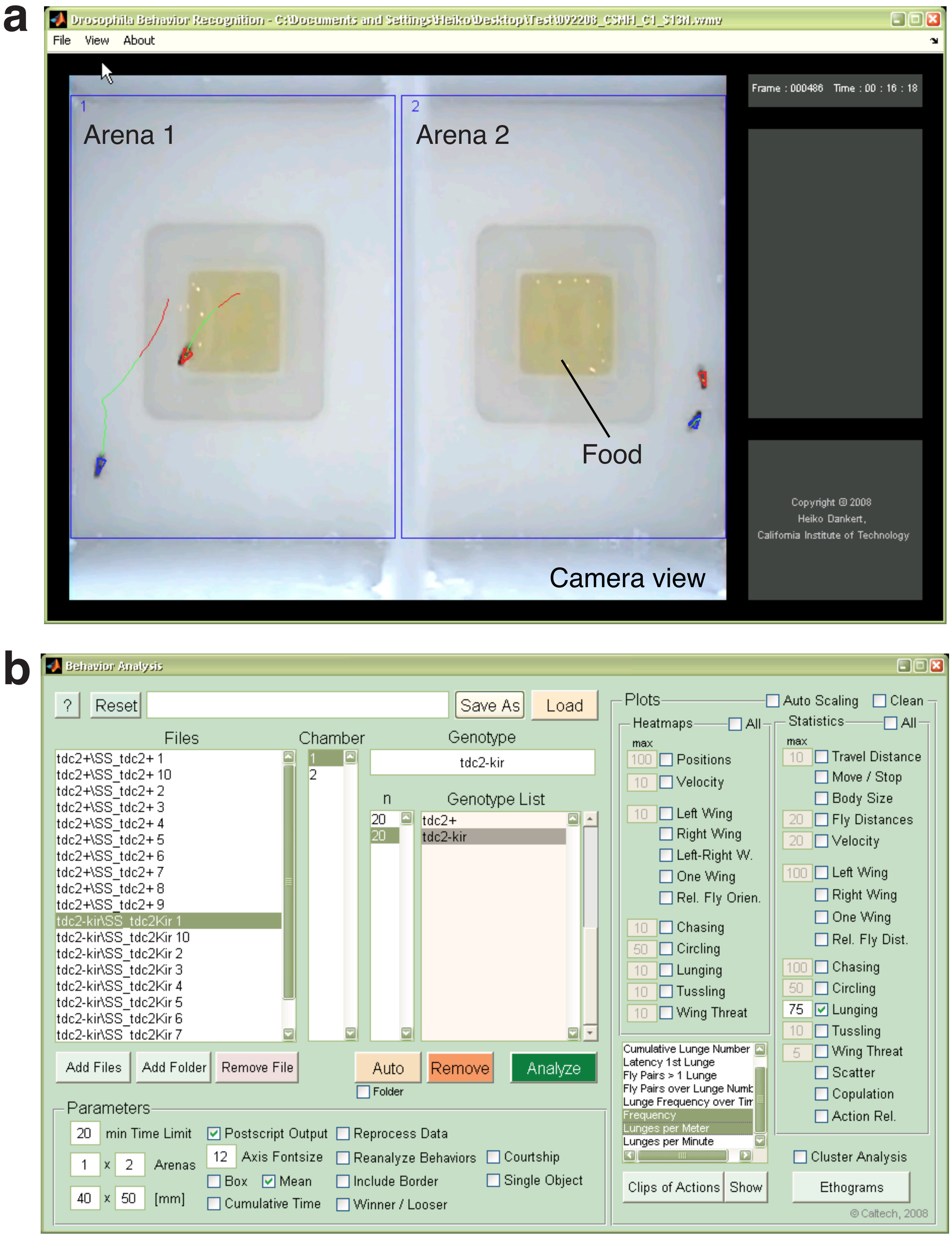
Supplementary Figure 5. Frequency of fly positions for aggressive actions, chasing and courtship actions. (**a**) Lunges (top), tussling (center row), and wing threat (bottom), for wild-type (CS) male-male (left, *n* = 20), male-female (center column, *n* = 24), and *Cha-Gal4;UAS-tra* ("Cha-Tra") male-male (right, *n* = 10) fly pairs. The color codes correspond to the cumulative sum of episodes of an action over all fly pairs. CS males perform neither lunging nor tussling, towards females and exhibit rare wing threats. The Cha-Tra males exhibit increased lunging and tussling. Additionally, Cha-Tra males perform a greater proportion of lunging and tussling on the food patch.

Supplementary Figure 5



Supplementary Figure 5. Frequency of fly positions for aggressive actions, chasing and courtship actions. (**b**) Chasing, for wild-type (CS) male-male (left, *n* = 20), male-female (center column, *n* = 24), and *Cha-Gal4;UAS-tra* ("Cha-Tra") male-male (right, *n* = 10) fly pairs. The color codes correspond to the cumulative sum of episodes of an action over all fly pairs. Cha-Tra males have a highly increased locomotor activity. (**c**) Circling (top) and wing extension (bottom).

Supplementary Figure 6



Drosophila Behavior Recognition - C:\Documents and Settings\Heiko\Desktop\Test\092208_CSMH_C1_S13H.wmv

Frame : 000486 Time : 00 : 16 : 18

Copyright © 2008
Heiko Dankert,
California Institute of Technology

Behavior Analysis

Files:
tdc2+\SS_tdc2+ 1
tdc2+\SS_tdc2+ 10
tdc2+\SS_tdc2+ 2
tdc2+\SS_tdc2+ 3
tdc2+\SS_tdc2+ 4
tdc2+\SS_tdc2+ 5
tdc2+\SS_tdc2+ 6
tdc2+\SS_tdc2+ 7
tdc2+\SS_tdc2+ 8
tdc2+\SS_tdc2+ 9
tdc2-kir\SS_tdc2Kir 1
tdc2-kir\SS_tdc2Kir 10
tdc2-kir\SS_tdc2Kir 2
tdc2-kir\SS_tdc2Kir 3
tdc2-kir\SS_tdc2Kir 4
tdc2-kir\SS_tdc2Kir 5
tdc2-kir\SS_tdc2Kir 6
tdc2-kir\SS_tdc2Kir 7

Chamber: 1, 2

Genotype: tdc2-kir

n	Genotype List
20	tdc2+
20	tdc2-kir

□ Folder

Parameters:
20 min Time Limit
1 x 2 Arenas
40 x 50 [mm]
☑ Postscript Output
12 Axis Fontsize
□ Box ☑ Mean
□ Cumulative Time
□ Reprocess Data
□ Reanalyze Behaviors
□ Include Border
□ Winner / Looser
□ Courtship
□ Single Object

Plots: □ Auto Scaling □ Clean

Heatmaps □ All (max)
100 □ Positions
10 □ Velocity
10 □ Left Wing
□ Right Wing
□ Left-Right W.
□ One Wing
□ Rel. Fly Orien.
10 □ Chasing
50 □ Circling
10 □ Lunging
10 □ Tussling
10 □ Wing Threat

Cumulative Lunge Number
Latency 1st Lunge
Fly Pairs > 1 Lunge
Fly Pairs over Lunge Numb
Lunge Frequency over Tim
Frequency
Lunges per Meter
Lunges per Minute

Statistics □ All (max)
10 □ Travel Distance
□ Move / Stop
□ Body Size
20 □ Fly Distances
20 □ Velocity
100 □ Left Wing
□ Right Wing
□ One Wing
□ Rel. Fly Dist.
100 □ Chasing
50 □ Circling
75 ☑ Lunging
10 □ Tussling
5 □ Wing Threat
□ Scatter
□ Copulation
□ Action Rel.
□ Cluster Analysis

© Caltech, 2008

Supplementary Figure 6. Software modules of the "Caltech Automated Drosophila Aggression-Courtship Behavioral Repertoire Analysis (CADABRA)". (**a**) Graphical user interface of our system for automated tracking and feature extraction (”QTrak”, Windows version). The software is shown while processing a movie. Both flies are labeled by the program, providing information on the identity, center of ellipse, head direction and previous positions of each fly. The right-hand side shows the actual frame number and time point. (**b**) Graphical User interface for post-processing the tracking information (”ANALYSIS”, Windows version). This software allows the user to select the tracking data, assign them to different genotypes, edit the genotype information, detect actions, plot 2D action histograms, statistics, ethograms, as well as produce movie clips for all actions detected.

Supplementary Table 1: Features, extracted by the tracking module per frame and fly.

feature	description
1	*x* and *y* position at time *t* (center of ellipse)
2-3	*fly length* (major axis of ellipse) and *fly area*
4-5	position of head and abdomen
6	orientation Θ (direction from center of ellipse to the head)
7-11	moving direction Θ_{move}, wing length l_L, l_R and wing angle - left Φ_L and right side Φ_R
12-13	velocity v_x, v_y and acceleration a_x, a_y
14-15	position and orientation change from *t* - 1 to *t*
16	difference between orientation Θ and moving direction Θ_{move}
17	distance to chamber's center (food-defending)
18-19	distance Δ_c and change of distance $\delta\Delta_{h-c}$ from *t* - 1 to *t* between center of ellipses
20	angle Θ_{12} between center of ellipse and head-direction of one fly and position of the other fly (lunging)
21-24	head-head Δ_{h-h}, abdomen-abdomen Δ_{t-t}, and head-abdomen Δ_{h-t} distance and change of distance
25	angle Θ_{w12} walking direction difference between both flies

Supplementary Table 2: Lunging feature ranges for pre-selection of possible lunges. ‘fly 1 or 2’ denotes the fly (1-2: measured between fly 1 and 2) and ‘time’ gives the time point *t* of feature extraction.

feature	fly 1 or 2	time	range		unit
			min	max	
velocity	1	*t*	0.5	200	mm/s
velocity	2	*t*	0	20	mm/s
acceleration	1	*t*	15	2000	mm/s^2
$\mathit{fly\ length}_{t-1} < \mathit{fly\ length}_{t}$	1				mm
length	1	*t, t* - 2	0.8	2.5	mm
Θ_{12}	1-2	*t* - 1	0	45	°
position change	1	*t*	0.05	5	mm
$\delta\Delta_{h-c}$	1-2	*t* - 1, *t*	0	1	mm
Δ_c	1-2	*t*	0.9	4	mm
$\delta\Delta_c$	1-2	*t*	-2.1	-0.15	mm

Supplementary Table 3: Features, used for final decision (lunge or no lunge) based on pre-selected frames (Supplementary Table 2). 'fly 1 or 2' denotes the fly (1, 2: measured for fly 1 and 2; 1-2: measured between fly 1 and 2) and 'time' gives the time point *t* of feature extraction.

feature	fly 1 or 2	time
velocity	1,2	t
acceleration	1,2	$t - 1$
fly length	1	$t - 2$, $t - 1$
Θ_{12}	1-2	$t - 1$
Δ_c	1-2	t
$\delta\Delta_c$	1-2	$t - 2 \rightarrow t - 1$
position change	1	$t - 1 \rightarrow t$

Supplementary Table 4: Tussling feature ranges and maintenance period. 'fly 1 or 2' denotes the fly (1, 2: measured for fly 1 and 2; 1-2: measured between fly 1 and 2) and 'time' gives the time point *t* of feature extraction. ‚Body alignment' is the orientation (with 180° ambiguity) difference between both flies.

feature	fly 1 or 2	time	range		unit	maintenance period
			min	max		
velocity	1,2	*t*	10	∞	mm/s	≥ 0.3 s
acceleration	1,2	*t*	80	∞	mm/s^2	
change of pos. change	1-2	*t* - 1, *t*	0	1	mm	
Δ_c	1-2	*t*	0	1.7	mm	
body alignment	1-2	*t*	-30	30	°	

Supplementary Table 5: Wing threat feature ranges and maintenance period. 'fly 1 or 2' denotes the fly (1-2: measured between fly 1 and 2) and 'time' gives the time point *t* of feature extraction.

feature	fly 1 or 2	time	range		unit	maintenance period
			min	max		
angle of both wings	1	*t* - 1, *t*	30	80	°	≥ 0.3 s
length of both wings	1	*t* - 1, *t*	1.1	1.9	mm	
velocity	1	*t* - 1, *t*	0.01	5	mm/s	
Δ_c	1-2	*t*	2	30	mm	
Θ_{12}	1-2	*t*	0	100	°	

Supplementary Table 6: Wing extension feature ranges and maintenance period. ‘fly 1 or 2’ denotes the fly and ‘time’ gives the time point *t* of feature extraction.

feature	fly 1 or 2	time	range		unit	maintenance period
			min	max		
wing angle Φ_L or Φ_R	1	*t*	60	90	°	≥ 1 s
wing length l_L or l_R	1	*t*	1.1	2.5	mm	
fly length	1	*t*	1.2	∞	mm/s	

Supplementary Table 7: Circling feature ranges and maintenance period. 'fly 1 or 2' denotes the fly (1-2: measured between fly 1 and 2) and 'time' gives the time point *t* of feature extraction.

feature	fly 1 or 2	time	range		unit	maintenance period
			min	max		
Δ_c	1-2	t	1	5	mm	≥ 0.7 s
$\delta\Delta_{h\text{-}c}$	1-2	t - 1, t	0	5	mm	
Θ_{12}	1-2	t	0	20	°	
velocity	1	t	0.25	∞	mm/s	
velocity	2	t	0	5	mm/s	
azimuthal velocity v_{az} wing not extended	1	t	0.25	∞	mm/s	
or:						
azimuthal velocity v_{az} wing extended	1	t	0.05	∞	mm/s	≥ 0.7 s

Supplementary Table 8: Copulation feature ranges. ‘fly 1 or 2’ denotes the fly (1-2: measured between fly 1 and 2) and ‘time’ gives the time point t of feature extraction.

feature	fly 1 or 2	time	range		unit
			min	max	
mean Δ_c	1-2	$t \pm 4.1$s	0	2	mm
standard deviation Δ_c	1-2	$t \pm 4.1$s	0	0.3	mm

Supplementary Table 9: Chasing feature ranges and maintenance period. ‘fly 1 or 2’ denotes the fly (1, 2: measured for fly 1 and 2; 1-2: measured between fly 1 and 2) and ‘time’ gives the time point *t* of feature extraction.

feature	fly 1 or 2	time	range		unit	maintenance period
			min	max		
	$\Delta_{h1\text{-}t2} < \Delta_{h2\text{-}t1}$				mm	≥ 1 s
Δ_c	1-2	*t*	3	10	mm	
$\delta\Delta_{h\text{-}c}$	1-2	*t*	0	2	mm	
Θ_{12}	1-2	*t*	0	45	°	
Θ_{w12}	1-2	*t*	0	45	°	
velocity	1,2	*t*	5	∞	mm/s	
position change	1,2	*t*	0.5	∞	mm	

- Supplementary Methods -
Automated Monitoring and Analysis of Social Behavior in *Drosophila*

Heiko Dankert[1,2], Liming Wang[2], Eric D. Hoopfer[2], David J. Anderson[2], and Pietro Perona[1]

[1] *Division of Engineering and Applied Science, California Institute of Technology, 1200 E California Blvd., Pasadena, CA 91125, USA.*

[2] *Division of Biology, Howard Hughes Medical Institute, California Institute of Technology, 1200 E California Blvd., Pasadena, CA 91125, USA.*

1 Arena

Our standard setup, as shown in **Figure 1** (main paper), consists of four double arenas, each with two arenas. An arena is made of teflon and has a rectangular base measuring 50 mm x 40 mm. A square food patch with a side length of 10 mm is placed within the center of the arena. The food patch is surrounded by 1% agar. Plexiglass walls, 115 mm tall and coated with Fluon, surround the arena. This chemical has properties similar to teflon and prevents flies from climbing the walls. A plexiglass lid covers the arena. The prototype of the arena was recently developed[1]. The food is a mixture of apple juice, sugar, and 1% agar. The recipe was described in[2].

The resolution bottleneck of our system is measuring wing position: to do this we need a resolution of at least 20 pixels per fly length (i.e., 10 pixels per mm assuming a fly length of 2 mm). Since our cameras have 640 x 480 pixel sensors, we can image a 64 mm x 48 mm area.

Allowing for some space lost to dividing walls, this allows us to image an array of 3 x 4 arenas that have 16 mm diameter as shown in **Figure SM-1**. This array follows a recently suggested and used design[3,4]. We found that this arena size is sufficient for measuring aggressive and courtship behaviors in pairs of flies. This set-up is suitable for large genetic screens with high-throughput.

2 Illumination and camera

A fluorescent ring-shaped light bulb (here: 22 W, 4100 K, 1100 lumen, "Circline T9", outside-diameter = 8 in., white reflector) with white, visible light surrounds the arenas to produce homogenous and bright illumination conditions (**Fig. 1**, main paper). These light sources have the additional advantage of being power efficient, which avoids heating the arenas and further condensation of water at the arenas walls and ceiling. The illumination has to be bright enough to provide sufficient contrast between background, fly body and wings. We measured a luminance of 850 cd/m^2 for our double-arena set-up with a neon-ring light. However, the tracking software is robust to changes in the overall level of illumination, which may be observed from experiment to experiment and are due to changes in ambient illumination and small adjustment in the position of the light source. A commercial color video CCD camera (24 bits/pixel, 640 x 480 pixel/frame, 30 frames/s, here: Sony DCR-HC 38) is placed vertically and downward-pointing above arena and light source. We prefer a longer distance in order to obtain a good approximation of a parallel projection; the distance is chosen first and the focal length is adjusted to fill the frame. This results in a resolution of $\approx$10 pixel/mm and $\approx$20 pixel/fly, given a length of $\approx$2 mm for a fly body. **Figure SM-2** shows a fly as seen by our current system.

3 Computer

Two color video CCD cameras (each 24 bits/pixel, 640 x 480 pixel/frame, 30 frames/s), and therefore four arenas, are connected in parallel via IEEE 1394a interface (400 MBit/s) to a personal computer, which allows digitization of the two video streams. For our experiments we used Intel dual-core CPUs with ≈3 GHz clock rate and at least 2 GB of memory. The video streams are captured and stored onto a hard disk using common video capturing software (here: Windows Movie Maker or Windows Media Encoder). Color movies are recorded in both setups at a frame rate of 30 Hz. In our experiments each movie was 20 minutes long (36,000 frames).

In order to reduce file size, while maintaining an image quality that was sufficient for our automatic analysis, we experimented with video compression techniques. **Figure SM-3** presents a magnified part of a frame using no compression as well as the same portion of frame compressed using different video codecs. The uncompressed image preserves the object details with an underlying evenly distributed noise. MPEG-4 V2 shows slight compression artifacts around the detailed objects and at edges. Windows Media Video 9 (WMV9) removes some amount of detail, as well as noise, without producing visible artifacts. DivX shows severe compression artifacts around the flies and produces spurious edges within the background. We chose the WMV codec giving sufficient quality and compressing 20 minute movies to sizes of ≈300 MB, as opposed to an uncompressed movie size of 33 GB. All data analysis and verifications were performed on compressed videos (WMV9).

4 Software

The software of our behavioral screening method consists of six modules: video import, ground-truthing, calibration, fly detection and tracking (pre-processing), action detection (post-processing; cf. main paper) and graphical user interface. The Video import routine was implemented in C, all other routines in Matlab. Operation of the software is completely automated: all the user has to do is specify which files have to be analyzed.

Video Import Our video import routine allows loading movies in any kind of compressed or uncompressed format assuming the appropriate video codecs are installed on the system. For speed optimization the routine decodes images in stacks of 100 frames.

Ground-Truthing For measuring the performance of our system, it is essential to have validation data, where the occurrence of any event of a particular action has been independently labeled. This is the so-called ground-truth information and obtained from two experts: one manually annotating a number of recorded movies, and another expert manually annotating the same movies. The second expert makes the final decision to accept or reject the events in the union of first and second expert's annotations. In addition, to identify any action events that were possibly missed by both human observers, the automated system is run at a low threshold to identify all possible action events. The second expert validates post-hoc any events identified by the automatic system, that were initially missed by the human observers. The ground truth is then defined as the union of the events identified and agreed on by the two human observers, together with any additional events detected by the automatic system that could be validated post-hoc by a human observer. This procedure is designed to take advantage of both the low false-positive rate of humans and the low false-negative rate of our system. Labeling software was developed and used for this purpose. **Figure SM-4** gives a screenshot of the graphical user interface of the labeling software module. The software allows a biologist to scan a movie looking for instances of a given action, and to mark the first and last frame of each video sequence where such action is observed. All collected information is saved into a text file.

Calibration The tracking software (see below) needs to know the boundaries of each arena. We call ‘calibration’ the process of obtaining such boundaries. The calibration process is fully automatic for the arena as shown in **Figure 1** (main paper). The software detects the edges of the food area using the Canny edge detector and morphological operators, and measures the length and width of the food area in pixels. Once the pixel position and size of the food patch are known, the position and size of the arena in the image are computed and used as the ROI. The ratio of the arena and food patch sizes, and mutual positions of each, is known from the design specifications of the arena. The calibration is automated in order to accommodate variations in the pixel size of the arena due to (small) variations in camera placement, focal length of the lens, and arena size.

At times, a new type of arena will be introduced. We developed an “Arena surveying GUI” in order to allow the user to measure the arena boundaries. The user draws the boundaries for each arena within the first video frame. The number of arenas is unlimited. For the conversion from pixel to millimeter, the user is asked to mark two points within the frame and provide their distance in millimeters.

Fly Detection & Tracking Here we give additional details on fly detection and tracking, which are illustrated in **Figure SM-5**. Before tracking the flies each 24 bit color frame is converted into a 8 bit brightness image I. We used a mixture of two channels, 40% red and 60% green.

We start our analysis by detecting the flies in each frame. Fly detection is accomplished by a sequence of five steps. In a first step the mean μ_I and standard deviation image σ_I are determined by averaging over 4000 randomly selected frames. The objects are separated from the background by computing a 'foreground image' $\mathbf{F}_I$:

$$F_I = 1 - \frac{I}{\mu_I + 3\sigma_I}. \tag{1}$$

It is more common to compute $(I - \mu_I)$, but we found that our ratio provides a better signal-to-noise (fly-to-background) separation than background subtraction and corrects for non-homogeneous illumination[5]. It requires a background, which is brighter than the flies. $3\sigma_I$ is added to the background to consider cases where flies do not move over long periods.

In a second step the fly bodies without the wings are segmented from the foreground image $\mathbf{F}_I$ as shown in **Figure 2a** (main paper) by fitting a Gaussian mixture model (GMM)[6] to the image brightness of the foreground image (**Fig. 2b**, main paper). The GMM consists of m = 3 components and associates each component to either background (dashed), other image parts and body (solid curves). Each component approximates the class-conditional probability density density function (pdf) that the value of a pixel in $\mathbf{F}_I$ belongs to either background, other image parts, or body (gray curve):

$$p(\mathbf{F}_I) = \sum_{m=1}^{3} p(\mathbf{F}_I|\theta_m)p(\theta_m) \tag{2}$$

where the mixing parameter $p(\theta_m)$ is optimized online and measures the fraction of pixels associated to component m and where $\sum_m p(\theta_m) = 1$. Each mixture component m is a Gaussian with mean μ and covariance matrix Σ:

$$p(\mathbf{F}_I|\theta_m) = \mathcal{N}(\mathbf{F}_I|\mu_m, \Sigma_m) \tag{3}$$

The GMM is optimized by the Expectation Maximization (EM) method[6]. The EM is initialized with μ, Σ of the previous frame. It should be noted that the EM requires an a priori selection of the model order, namely, the number of components m to be incorporated into the model. In order to assign pixels to the fly body, the second and third pdf (other parts and body) are divided by the sum of all pdfs. All pixels, whose foreground value is higher than a threshold are selected as body pixels. The threshold is defined to be the intersection of the second and third pdf (**Fig. 2b**, main paper).

In a third step the fly body pixels obtained in the second step are grouped into connected sets by using nearest-neighbor connectivity[7].

In a fourth step an ellipse is fit to each connected component: we choose the smallest ellipse containing all the pixels of the connected component (**Fig. 2b**, bottom-right, main paper). The length of the major axis of this ellipse provides an estimate of the length of the body of the fly, the orientation of the major axis an estimate of the fly's orientation and the center of the ellipse an estimate of the fly's position.

The fifth step in fly detection is to extract the wings. Our approach is to first segment each fly (body, wings) from the background, and then subtract the body pixels, obtained in step two above, in order to reveal the wings. This is not easy to accomplish: ideally background, body and wings have different $\mathbf{F}_I$ pixel values due to the transparency of the wings. However, in many cases background noise, video compression artifacts, as well as varying background patterns (border, agar, food) make it hard to clearly distinguish between wings and background. We explored two methods for overcoming this difficulty.

The first method fits a GMM taking into account the location and brightness of each pixel $(\mathbf{X}, \mathbf{F}_I)$ to segment the full fly from the background:

$$p(\mathbf{X}, \mathbf{F}_I) = \sum_m p(\mathbf{X}, \mathbf{F}_I|\theta_m)p(\theta_m), \tag{4}$$

where $p(\mathbf{X}, \mathbf{F}_I|\theta_m)$ is a Gaussian probability density function, and we assume that X and F are independent. The second way applies a fast, nonparametric and unsupervised method of automatic multi-threshold selection for image segmentation (optimal thresholds), whereas the optimal thresholds are derived from the viewpoint of discriminant analysis[8]. Both methods may be applied to those regions of a frame where fly bodies were detected.

After extensive testing the first method (GMM) was found to be too slow due to the two extra degrees of freedom and the numerous iterations necessary to fit the model. The method of optimal thresholds is 5-10x faster than the first method. Furthermore, it is better in segmenting non-Gaussian shapes.

To quantify the performance of both methods for wing segmentation we developed a GUI which picked ≈15,000 foreground images, segmented them by applying both methods. The GUI presented foreground image, optimal thresholds and the GMM segmentation side-by-side to an expert, who was asked to evaluate whether none, one, or both segmentations had properly identified the pixels belonging to the body and to the wings. 300 samples were randomly chosen for evaluation. Good segmentation was delivered in 96.3% of these images using optimal thresholds, and in 59.0% using GMM. **Figure SM-6** shows the performance of both methods on a selection of problematic, non-standard situations.

Based on these results, we utilized the method of optimal thresholds. **Figure 2c** (main paper) shows a resulting segmentation where the body, wings and background are highlighted by different colors.

The orientation Θ of a fly is computed as the orientation of the major axis of the ellipse fitting and has a 180° ambiguity (**Fig. 2e**, main paper). To remove this ambiguity the location of head and abdomen have to be determined. To do so, we make use of the combination of three sources of information: (1) the different visual appearance of the head and tail segment of each fly, (2) fly motion (when flies walk, they tend to move forward), and (3) consistency with previous frames. (1) Head-tail appearance is computed by dividing the fly along the minor axis $\mathbf{b}$ (dashed line) and comparing the brightness-value distribution of the two parts (**Fig. 2c**, bottom, main paper). Transparent wings and abdomen have a different brightness-value distribution than the head region and allow to distinguish between head and abdomen. We label ‘head’ the end of the fly whose brightness histogram of the foreground image has brighter values. (2) Retrieving the head-abdomen position only by analyzing the fly's movements is problematic, because often flies do not move (feeding, grooming, or sleeping) and sometimes they may slowly move backwards. In situations where the wings are not visible or poorly imaged, if the fly velocity is above an empirical threshold (unambiguous forward fly motion), the software uses the fly's moving-direction to decide for head-abdomen position. (3) When brightness information is ambiguous and the fly is not moving, moving slowly, or when two flies are in close proximity ($<$1.5 fly body lengths) the fly tracking information of the previous frame is used to assign head-abdomen orientation.

After determining the position of the fly's head and abdomen the wings are measured in each frame. **Figure 2c**, bottom and **Figure 2d** (both main paper) illustrate the measurement of wing length and wing angle.

The formerly discussed GUI for verification of the full fly segmentation was used to retrieve ground truth on wing angles. In addition to voting for the segmentation completeness, the expert was asked to click on the head, abdomen, and the tip of left and right wings in the foreground image for the same 300 random samples. **Figure SM-7** compares automatically measured wing angles, using optimal thresholds segmentation results of full flies, with ground truth data provided by the human expert. The correlation coefficient is 0.94 with a standard deviation of 7.9° (bias = 0.4° towards automated measurements), including 4 head-tail swaps (triangles) and 11 segmentations that were voted as not complete (crosses). The incomplete segmentations led, however, to reasonable wing angles in most cases. The post-processing software (**Supplementary Fig. 6b** online) resolves problematic issues by analyzing past and current fly orientations as well as wing angles. The software assigns both fly orientations and wing angles to the most likely value.

The fly detector will detect a single object when the two flies are very close. This anomalous situation is flagged automatically by applying a threshold on the area of the detected object. In this case the system applies a GMM with m = 2 Gaussians using pixel location **X** and pixel brightness-value $\mathbf{F}_I$ (cf. eq. 4) to resolve the bodies of the two flies. Each Gaussian is supposed to fit one fly. The optimal parameters of the model are found by maximizing likelihood using the EM algorithm. **Figure 2f** (main paper) shows two examples of how pairs of touching flies are segmented using this technique.

Fly discrimination The resolution at which our system operates (10 pix/mm) does not allow us to resolve any other feature on a fruit fly besides the head, abdomen and wings. In particular, it is impossible to discriminate individuals of the same sex. When this is needed a white dot is painted on the back of one fly. The dot appears as a dark spot in the foreground image $\mathbf{F}_I$, and as a peak in the brightness histogram of the fly's body. This peak is used to identify the labeled fly. Since the dot is small and placed in the center of the thorax it does not affect the ellipse fitting (step four, “Fly Detection & Tracking”). However, to compute the correct fly area and to make sure the dot does not cause the image of the fly body to be split into two disconnected components, it is erased by performing morphological operations on the binary image of the segmented flies. For unlabeled male-male pairs our software analyzes the past and current fly positions, along with the direction of movement. The software assigns both objects to the most likely location producing continuous fly-specific trajectories. Therefore, in principle, one may obtain individual-specific trajectories.

We have not tested how frequently the identity of the two flies is exchanged. If one fly is a female and the other one is a male, then identity may be determined directly and reliably from the apparent size of each fly, *Drosophila* females being larger than the males.

Feature extraction The tracking module extracts 25 primary and derived features per frame describing the position, velocity, and shape of each fly (ten features per fly), as well as the mutual position and velocity of the two flies (5 additional features; **Fig. 2d,e,g** and **Supplementary Table 1** online).

Velocities v_x, v_y at time t are computed by convolving fly position differences of four consecutive frames at times $t-2...t+1$ with a Gaussian filter kernel of size three:

$$\begin{bmatrix} v_{x|t} \\ \\ v_{y|t} \end{bmatrix} = \frac{1}{\Delta t} \begin{bmatrix} (x_{t-1}-x_{t-2}) & (x_t - x_{t-1}) & (x_{t+1}-x_t) \\ \\ (y_{t-1}-y_{t-2}) & (y_t - y_{t-1}) & (y_{t+1}-y_t) \end{bmatrix} \cdot [0.25 \;\; 0.5 \;\; 0.25]^T \qquad (5)$$

The convolution smoothes the estimate of fly velocity and reduces the noise. Δt denotes the time difference between two consecutive frames. The accelerations a_x, a_y at time t are computed as follows:

$$a_{x|t} = \frac{v_{x|t+1} - v_{x|t-1}}{2\Delta t} \quad a_{y|t} = \frac{v_{y|t+1} - v_{y|t-1}}{2\Delta t}. \qquad (6)$$

Example-based Classification An example-based classifier was designed to detect lunging (cf. main paper). Lunging is the concatenation of three phases: torso-raising, thrusting torso towards and collapsing onto the opponent, and pulling the opponent. Each one of these phases is highly variable in velocity and duration. The first two phases typically take only 46 ms[1] and a single lunging episode often happens in 2-3 frames when imaged at 30Hz.

The three phases are detected in a two steps. First, a simple and inexpensive criterion is used to select frames that are likely to contain lunges. During the first phase, a lunging fly will raise its torso. Thus, its length will appear to become smaller and its distance from the other fly will appear to increase. We flag all frames where this happens as ‘probable lunges’ by using the features listed in **Supplementary Table 2** online and detecting frames when these features take values outside defined ranges.

Not all ‘probable lunges’ are actual lunges and we found it impossible to define feature ranges that would uniquely detect lunges. Thus, one more step is needed for lunge detection. It is based on supervised classification: given a number of training examples, i.e. instances of lunging that were selected by an expert, we trained a classifier to decide whether a ‘probable lunge’ is actually a lunge or not. We experimented with several classification methods and found that the k-Nearest Neighbor (kNN) algorithm works best for our purpose[9]. At each frame of a movie our tracking algorithm computes 10 features per fly: position, velocity, orientation etc. (see **Supplementary Table 3** online for the complete list). Thus, each positive (lunge) and negative (non-lunge) training example is associated to a corresponding 10-dimensional feature vector at each frame. As positive training examples we select feature vectors of flies and frames that are labelled as lunges in our training database. A similar number of negative training samples is randomly chosen from frames where we detected no lunge. Since the metric of the different feature-space dimensions varies, the data need to be first ‘centered’ and then ‘sphered’; this normalization step is common practice in pattern recognition and typically improves the performance of a classifier. Centering and sphering consist of finding a new coordinate system where the set of feature vectors $\mathbf{v}$ has zero mean and its covariance is the identity. In order to sphere them, features **v** are linearly transformed to a variable $\mathbf{v}'$ such that the covariance matrix $\mathbf{c}(\mathbf{v})$ equals unity $E\{\mathbf{v'v'}^T\} = \mathbf{I}$. To do so, we perform an eigenvalue decomposition of **c** by factorizing via singular value decomposition:

$$\mathbf{c} = \mathbf{USV}. \tag{7}$$

We then transform the features so the covariance matrix of the transformed data is equal to the identity:

$$\mathbf{v}' = \mathbf{vUS}^{-1/2}. \tag{8}$$

The covariance of v' is now the identity.

Implementing a k-nearest neighbor classifier is simple: given an example to be classified, we compute its k nearest neighbors in the training set. The example is classified as a lunge if the majority of its k-nearest neighbors are also lunges, as a non-lunge otherwise. This idea may be generalized by using a fraction ν of the neighbors as the decision threshold (**Fig. 4a**, main paper).

Graphical User Interface We developed graphical user interfaces for hand-labeling the data to estimate the ground truth, as well as for running the tracking and feature extraction software ("QTrak") as shown in **Figure SM-4** and **Supplementary Figure 6a** online. The two user interfaces are similar: both allow importing movies in any kind of compressed or uncompressed format assuming the appropriate video codecs are installed on the computer. The user selects one movie, or a set of movies, and for calibration purposes within the tracking software also draws the boundaries around each arena for arrays with more than two arenas (cf. 'Calibration'). No further manual adjustments are necessary. The import routine goes sequentially through the movies, frame by frame, and passes each frame into the labeling or tracking and feature extraction module for processing. The information computed by both modules is stored into text files.

The labeling module allows the user to play a movie, stop at probable events, step forward and backward in time, frame by frame, mark visible events (start- and end frame), and classify the action regarding a set of given choices.

The tracking software is able to detect and track pairs of moving flies in parallel within multiple arenas. Its output is the trajectory (position, velocity, direction, head and abdomen position, wings) for each fly in each arena as a function of time. The tracking module currently runs ≈2.5x slower than real-time, speeding it up to real-time appears possible.

The analysis module of our software system ("ANALYSIS"), as shown in **Supplementary Figure 6b** online, is responsible for the detection and analysis of actions out of the trajectories, provided by the tracking module. The analysis includes various action statistics, comparisons, and ethograms.

Besides detecting and classifying actions, our software will output all detected actions in the form of short video-clips, one per action, which allows an operator to quickly verify the quality of the output by eye. **Supplementary Figure 1** online shows movie-clip examples of aggressive and courtship actions that were detected automatically and saved at 50% of the original spatial resolution.

5 Sources of Error

We identify six failure modes: (1) incomplete action execution, (2) blurring, (3) sudden movements, (4) identity swapping, (5) boxing, tussling and wall climbing attempts, (6) occlusion and grouping and (7) imaging effects. **Figure SM-8** gives examples of frames with problematic tracking situations. (1) Behaviors such as lunges are sometimes unusual by missing one of the 3 steps (i.e., the lunging fly would not stand on his hind-legs but simply thrusts directly towards the opponent). (2) Flies can reach velocities of up to 0.85 ms^{-1} during flight phases[10]. This is $\approx$28 mm/frame at our current system's sampling rate of 30 Hz. A fast moving fly does therefore tend to be under-sampled and appears blurred (the object is stretched over two or more frames). As long as the blurred object is distinguishable from the background (i.e., during the start and end phases of a flight) the tracking module is able to track the object). However, the semi-transparent wings can disappear in such cases, so that no information about wing position is available. Fast moving flies are not analyzed and those frames with exceptional high velocities and accelerations are discarded. Hence, no actions may be detected when flies are jumping or flying. (3, 4) Flies do also perform sudden movements such as jumping, falling, or very fast lunges and tussling. Therefore, they can appear at a very different location in the current frame compared to the previous one. The changes of movements are carried out within a few hundredths of a second. This can cause identity swapping between both opponents in case of unlabeled flies of the same sex. (5) Flies also often try to climb the arena's walls or occasionally stand near vertically while boxing or tussling. They are visible from different sides and angles. The system will track the flies correctly in these situations. Wing and dot detection as well as discrimination between head and abdomen are possible as long as these parts stay visible. The moving direction and the orientation information from the last frame are used as an additional information to find the correct orientation. For the detection of tussling the head and abdomen information are not necessary (cf. ’Tussling’ in main paper and **Supplementary Table 4** online). However, our analysis module (**Supplementary Figure 6b** online) allows the user to exclude the boundary zone (3 mm distance from the boundary) from the analysis. (6) At times, flies will come in contact and, possibly, partially occlude each other. In such cases the tracking software separates the flies by fitting two Gaussians to the object as explained in section 4 and **Figure 2f** (main paper). (7) Inhomogeneous and non-stationary illumination conditions, noise due to image compression and the camera, as well as the color and edges of the food area can increase the complexity of pattern recognition. Wings for instance are not always clearly imaged and separable from the background. These issues are solvable by using a DC light (i.e., an LED array), low or no compression and an arena with a homogenous pattern (i.e., without a food patch). A camera with higher resolution and frame rate, e.g. at 500 Hz, would dramatically increase the amount of data and therefore necessary storage and computational power for analysis, but it would only have minor effects on detection performance since the error rate is already low.

References

1. Hoyer, S. *et al.* Octopamine in male aggression of drosophila. *Curr. Biol.* **18**, 159–167 (2008).

2. Wang, L., Dankert, H., Perona, P. & Anderson, D. A common genetic target for environmental and heritable influences on aggressiveness in drosophila. *Proc Natl Acad Sci U S A* **105**, 5657–5663 (2008).

3. Dierick, H. A. A method for quantifying aggression in male drosophila melanogaster. *Nat Protoc* **2**, 2712–2718 (2007).

4. Zhou, C., Rao, Y. & Rao, Y. A subset of octopaminergic neurons are important for drosophila aggression. *Nat Neurosci* (2008 Aug 1).

5. Jähne, B. *Digital Image Processing* (Springer, 2005), 6th edn.

6. Bishop, C. *Pattern Recognition and Machine Learning* (Springer Verlag, New York, 2007), 1 edn.

7. Haralick, R. & Shapiro, L. *Computer and Robot Vision*, vol. 1, 28–48 (Addison Wesley, 1992).

8. Otsu, N. A threshold selection method from gray level histograms. *IEEE Trans. Systems, Man and Cybernetics* **9**, 62–66 (1979).

9. Shakhnarovich, G., Darrell, T. & Indyk, P. Nearest-neighbor methods in learning and vision: Theory and practice. *The MIT Press* (2006).

10. Marden, J. H., Wolf, M. R. & Weber, K. E. Aerial performance of drosophila melanogaster from populations selected for upwind flight ability. *J Exp Biol* **200**, 2747–2755 (1997).

Competing Interests The authors declare that they have no competing financial interests.

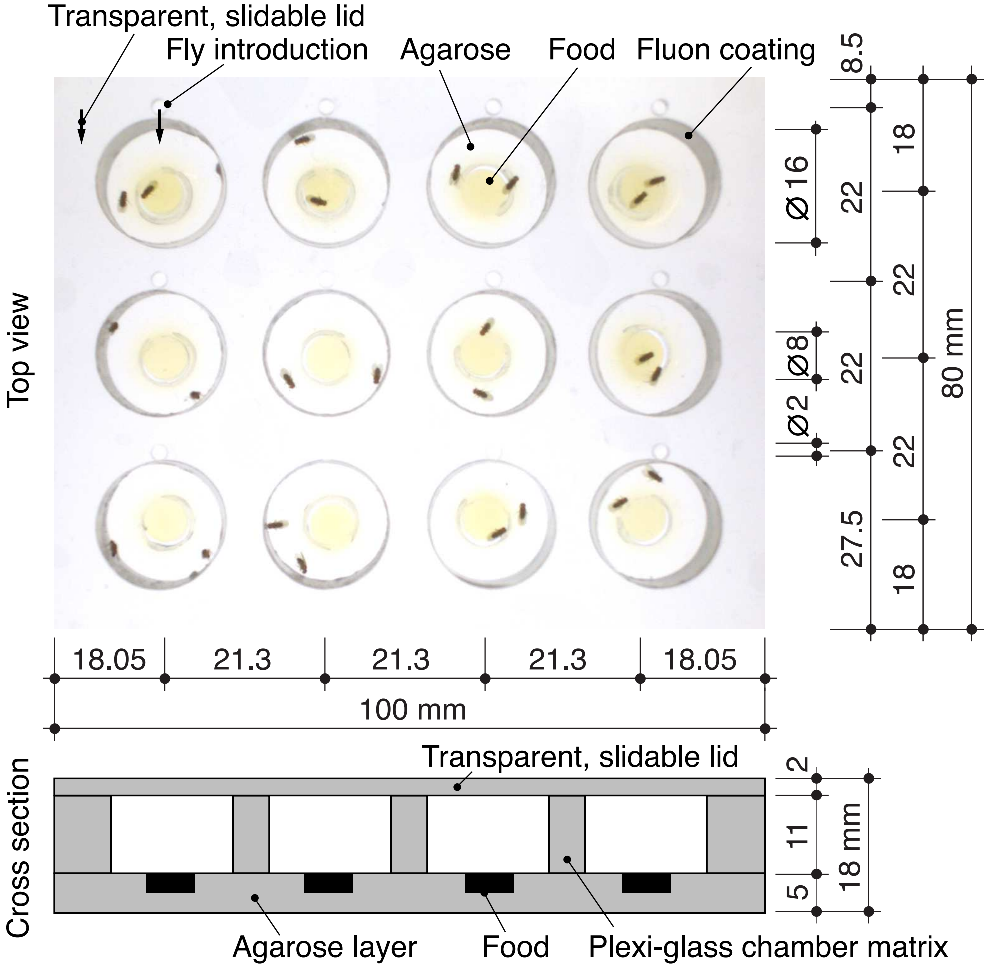


Figure 1: 12-arena setup allowing to test one genotype per movie and camera (*n* = 12). The arenas are drilled into plexiglass which lays on a 1% agar layer. A transparent and slidable plexiglass lid with holes for fly introduction lays on top of the matrix. Food is placed within the center of each arena. Alternatively the agar layer can be mixed with food. All lengths in millimeter.

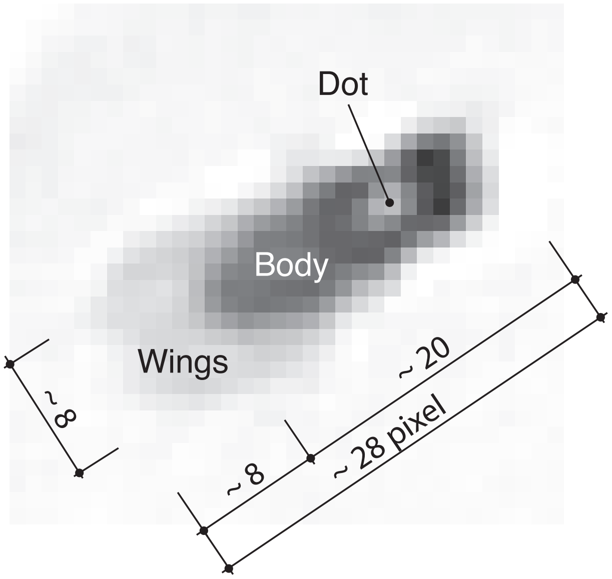


Figure 2: Dimensions of a fly in pixel as seen by our system. A white dot is painted on its back. 10 pixel ≈1 mm.

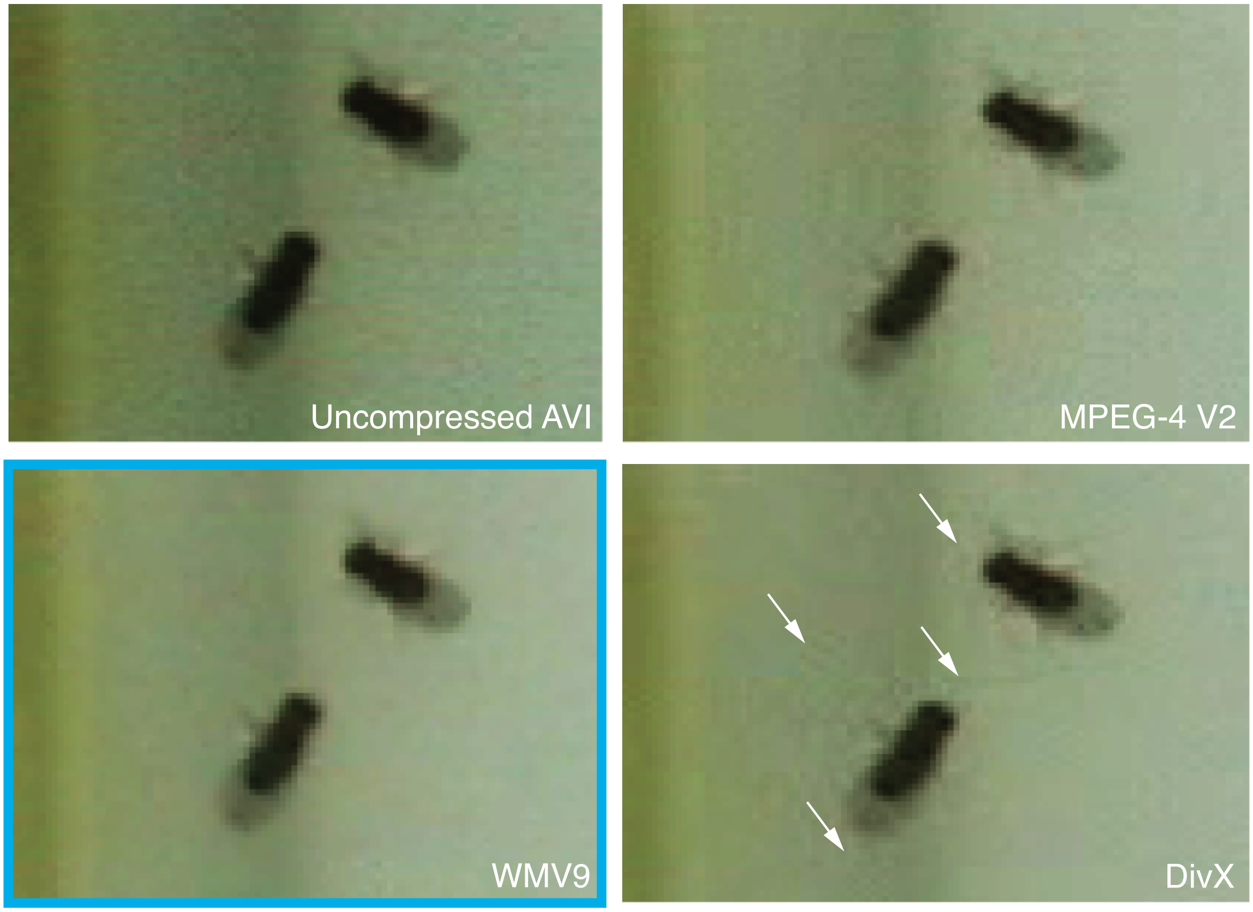
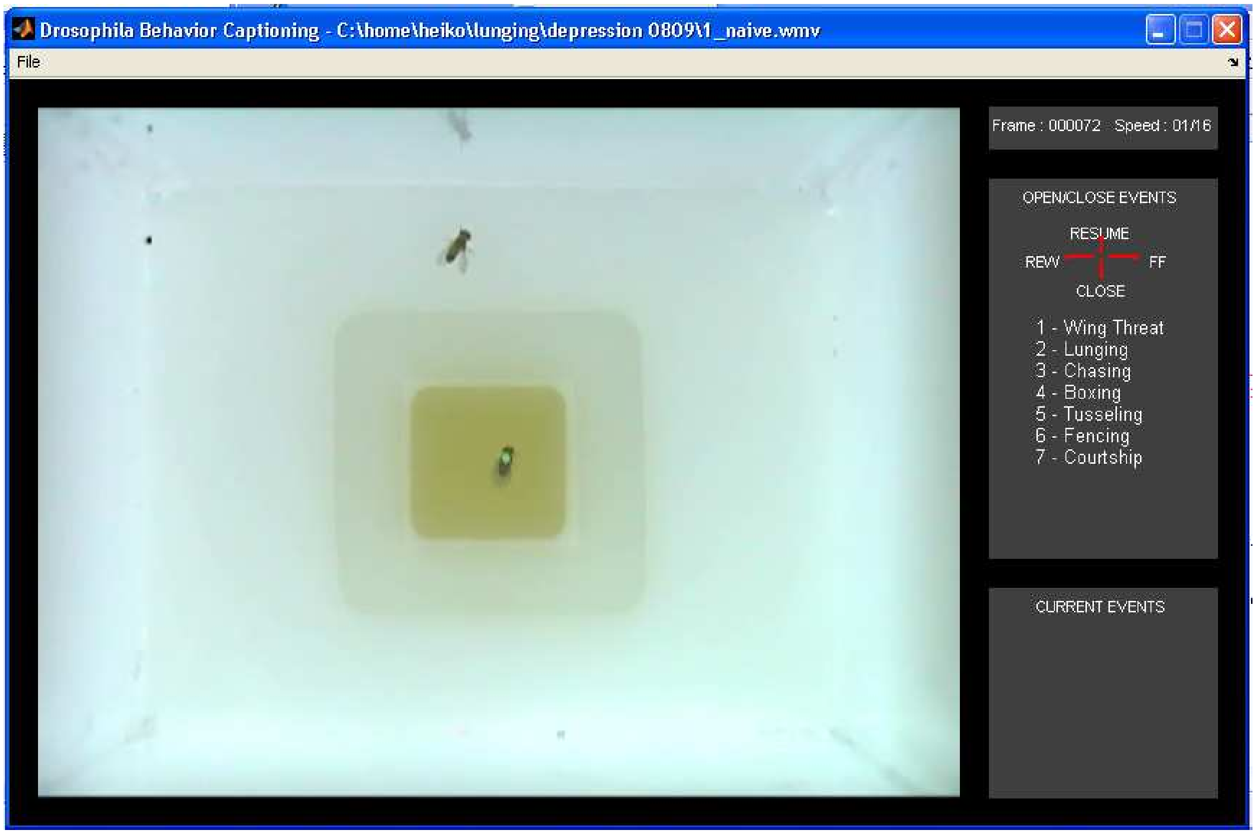


Figure 3: Magnification of a video frame showing part of an arena with two flies. The frame was coded as uncompressed AVI (top-left) and with three different compression codecs (MPEG-4 V2, WMV9, DivX). WMV9 was chosen for best quality and low noise at high compression rate. The arrows point at some of the compression artifacts that can occur and can be problematic for wing detection. In the background the three different patterns of the arena: food, agar, teflon.



Drosophila Behavior Captioning - C:\home\heiko\lunging\depression 0809\1_naive.wmv

Frame : 000072 Speed : 01/16

Figure 4: Labeling software for annotating movie sequences by hand and producing ground truth. The user can load and play a movie at different speeds, stop at any frame, go back and forward frame by frame, label start- and end-frame of an event, and chose the event type from the list on the right-hand side.

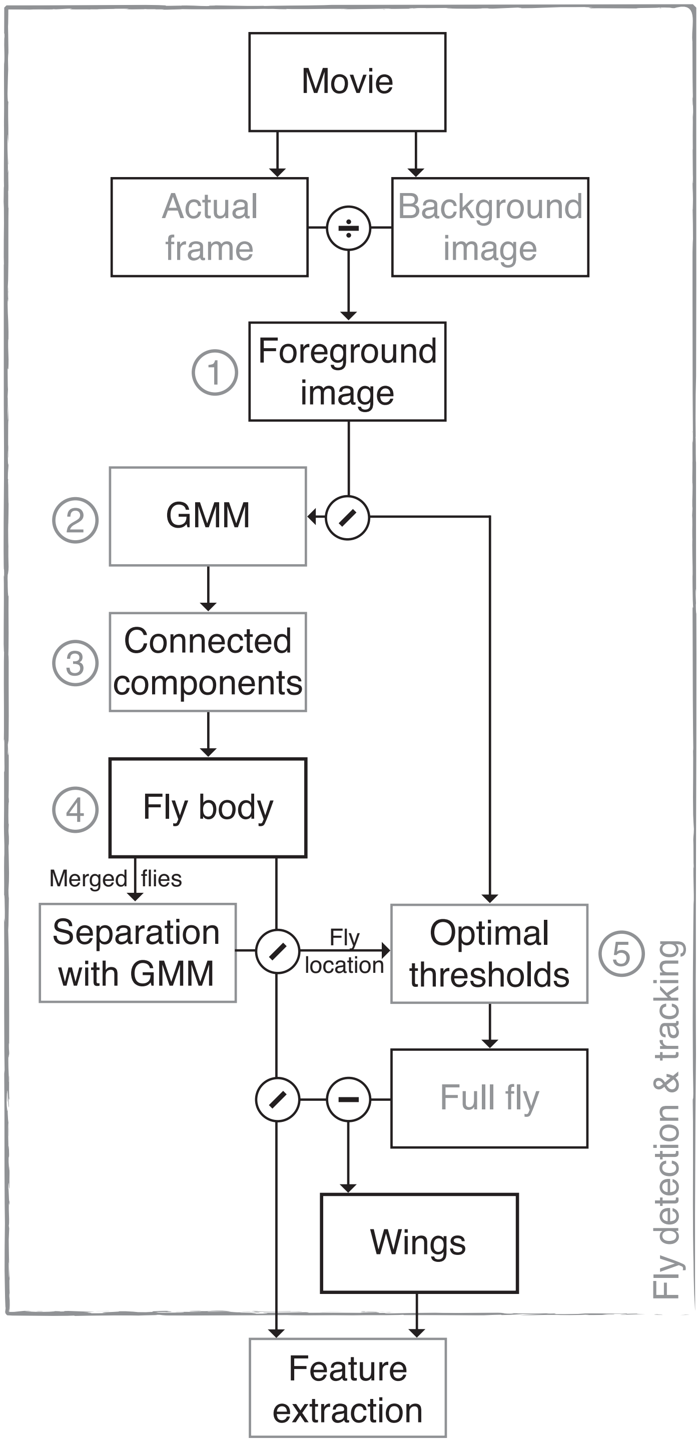


Figure 5: Fly detection and tracking scheme.

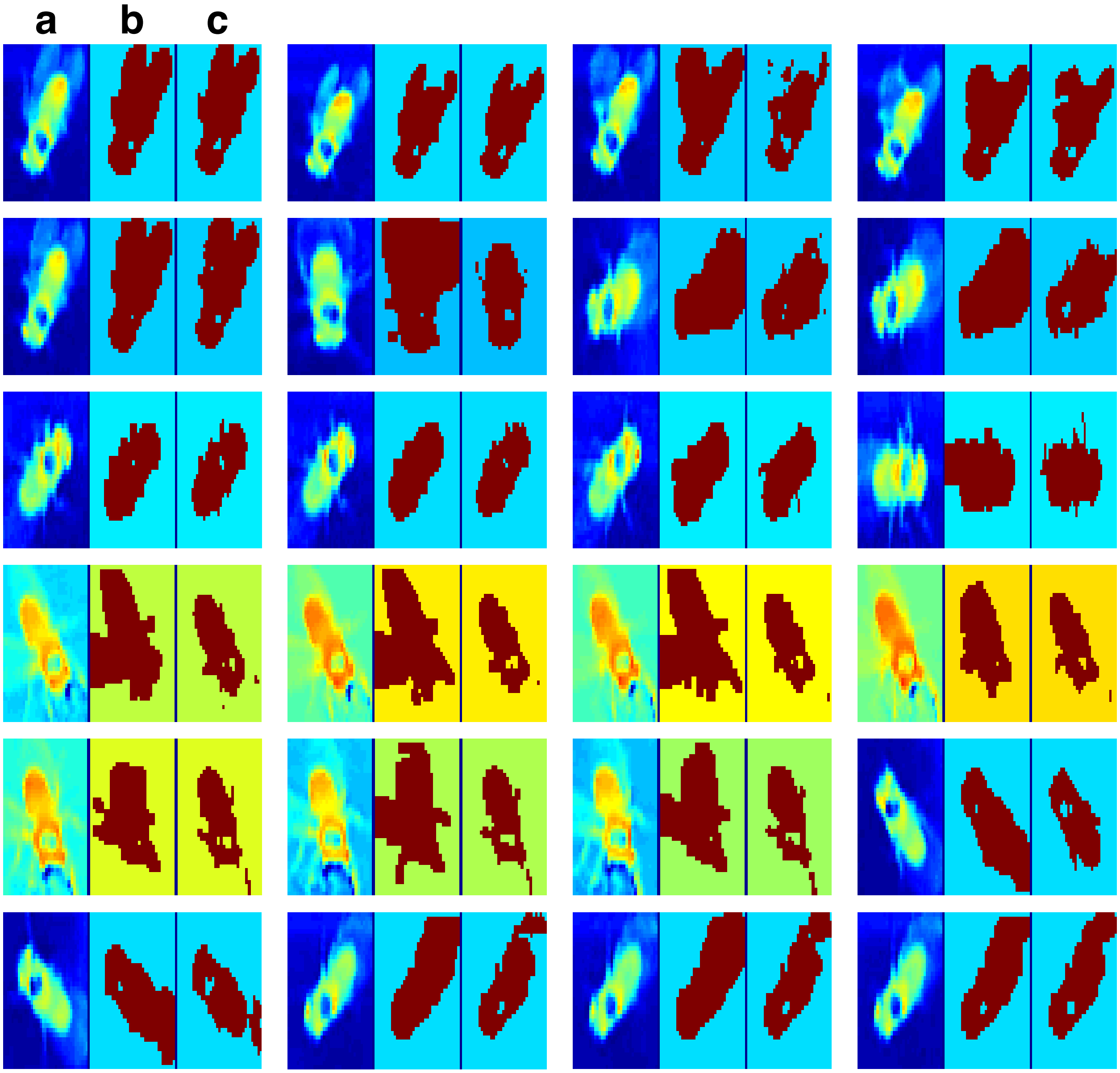


Figure 6: Segmentation results for full fly detection in problematic, non-standard situations with background noise and brightness-value close to the brightness-value of the wings. (a) 'Foreground' image, (b) optimal thresholds segmentation with five components[8], (c) GMM segmentation with five components using brightness values and pixel locations. Optimal thresholds performs 5-10x faster than GMM and is not dependent on the shape of the object.

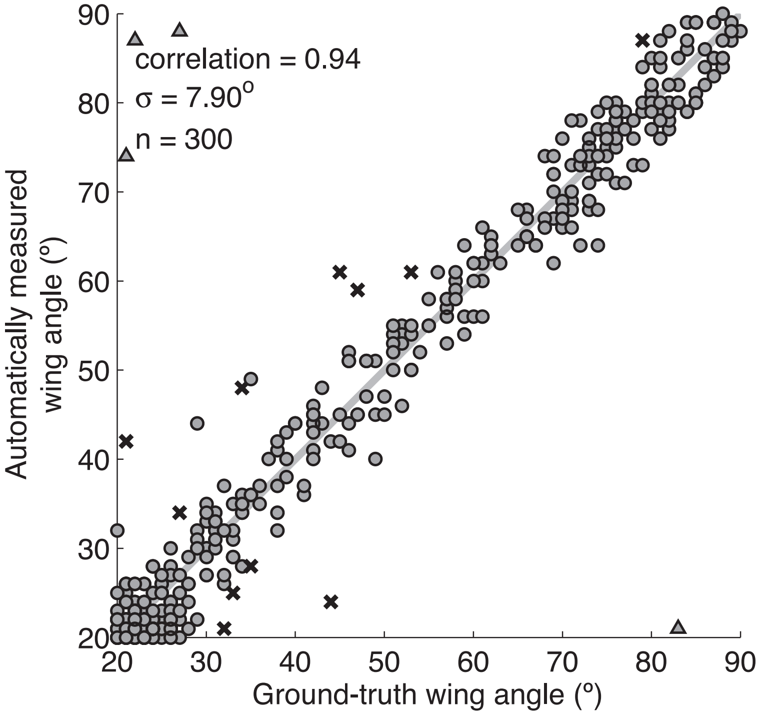


Figure 7: Scatter diagram comparing automatic wing angle with ground truth data (right and left wing). Good segmentations are indicated by dots. Segmentations that were evaluated as not complete are represented by crosses and head-tail swaps by triangles. The automatically measured wing angle is defined to be the angle between the center of fly body ellipse, the ellipse' long axis, and wing tip as described in **Figure 2d** (main paper). The ground truth wing angle is defined to be the angle between the head-abdomen center point, the head-abdomen axis, and the wing tip. This definition difference and the freedom of choice for the expert's clicks on head, abdomen, and wings causes a variation of $\approx 5^\circ$ between ground truth and automated measurements. The detectable minimum wing angle is 20°.

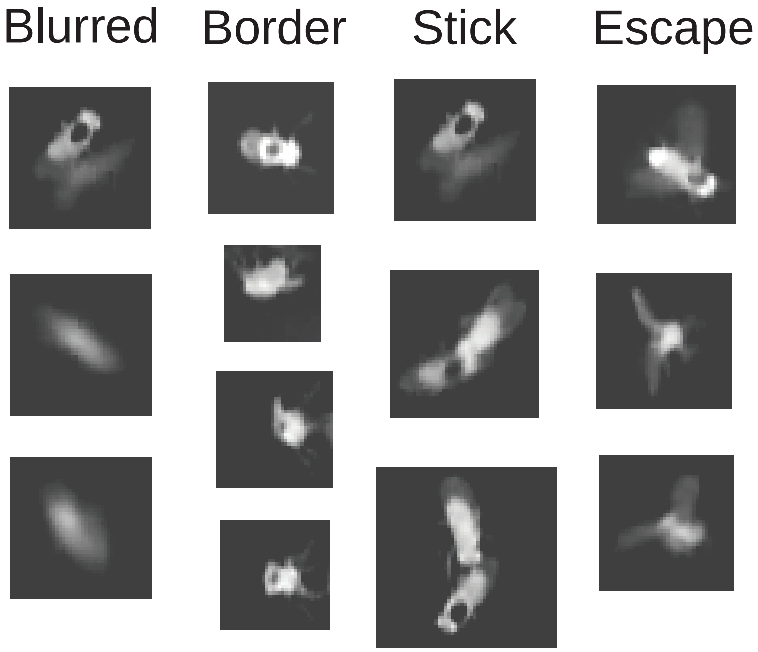


Figure 8: Examples for problematic tracking situations: flies can appear blurred due to temporal under-sampling, stay at the border, appear as one object when they are in very close proximity, and escape by flying towards the ceiling.