

1

2

3 **FlyLimbTracker: an active contour based approach for leg**
4 **segment tracking in unmarked, freely behaving *Drosophila***

5

6 Virginie Uhlmann^{1,¶,*}, Pavan Ramdya^{2,3,¶,#b,*}, Ricard Delgado-Gonzalo^{1,#a},

7

Richard Benton³, Michael Unser¹

8

9 ¹ Biomedical Imaging Group, École Polytechnique Fédérale de Lausanne
10 (EPFL), Lausanne, Switzerland

11 ² Institute of Microengineering, École Polytechnique Fédérale de Lausanne
12 (EPFL), Lausanne, Switzerland

13 ³ Center for Integrative Genomics, Faculty of Biology and Medicine, University
14 of Lausanne, Lausanne, C H-1015, Switzerland.

15 ^{#a} Current Address: Centre Suisse d'Électronique et Microtechnique (CSEM),
16 Neuchâtel, CH-2002, Switzerland.

17 ^{#b} Current Address: Division of Biology and Bioengineering, California Institute
18 of Technology, Pasadena, California, United States of America

19 ¶ These authors contributed equally to this work

20 * Corresponding authors:

21 Email: ramdya@caltech.edu (PR), or virginie.uhlmann@epfl.ch (VU)

22 **Short title**

23 FlyLimbTracker tracks leg segments in freely behaving *Drosophila*

24 **Abstract**

25 Understanding the biological underpinnings of movement and action requires
26 the development of tools for precise, quantitative, and high-throughput
27 measurements of animal behavior. *Drosophila melanogaster* provides an ideal
28 model for developing such tools: the fly has unparalleled genetic accessibility
29 and depends on a relatively compact nervous system to generate
30 sophisticated limbed behaviors including walking, reaching, grooming,
31 courtship, and boxing. Here we describe a method that uses active contours
32 to semi-automatically track body and leg segments from video image
33 sequences of unmarked, freely behaving *Drosophila*. We show that this
34 approach is robust to wide variations in video spatial and temporal resolution
35 and that it can be used to measure leg segment motions during a variety of
36 locomotor and grooming behaviors. FlyLimbTracker, the software
37 implementation of this method, is open-source and our approach is
38 generalizable. This opens up the possibility of tracking leg movements in
39 other species by modifications of underlying active contour models.

40

41

42

43

44 **Author Summary**

45 In terrestrial animals, including humans, fundamental actions like locomotion
46 and grooming emerge from the displacement of multiple limbs through space.
47 Therefore, precise measurements of limb movements are critical for
48 investigating and, ultimately, understanding the neural basis for behavior. The
49 vinegar fly, *Drosophila melanogaster*, is an attractive animal model for
50 uncovering general principles about limb control since its genome and
51 nervous system are easy to manipulate. However, existing methods for
52 measuring leg movements in freely behaving *Drosophila* have significant
53 drawbacks: they require complicated experimental setups and provide limited
54 information about each leg. Here we report a new method - and provide its
55 open-source software implementation, FlyLimbTracker - for tracking the body
56 and leg segments of freely behaving flies using only computational image
57 processing approaches. We illustrate the power of this method by tracking fly
58 limbs during five distinct walking and grooming behaviors and from videos
59 across a wide range of spatial and temporal resolutions. Our approach is
60 generalizable, allowing researchers to use and customize our software for
61 limb tracking in *Drosophila* and in other species.

62

63 **Introduction**

64 Many terrestrial animals rely on complex limb movements to locomote, groom,
65 court, mate, and fight. Discovering how these and other fundamental
66 behaviors are orchestrated by the nervous system will require manipulations
67 of the genome and nervous system as well as quantitative measurements of
68 behavior. The vinegar fly, *Drosophila melanogaster*, is an attractive model
69 organism for uncovering the neural and genetic mechanisms underlying
70 behavior. First, it boasts formidable genetic tools that allow experimenters to
71 remotely activate, silence, visualize and modulate specific gene function in
72 identified neurons [1]. Second, a number of sophisticated methods have been
73 developed that permit robust tracking of *Drosophila* body movements – a
74 promising set of tools for high-throughput screens [2-7].

75 By contrast, similarly robust methods with the precision required to
76 semi-automatically track leg segments are largely absent. State-of-the-art
77 approaches suffer from several drawbacks. For example, the most precise
78 methods require the manual placement of visible markers on tethered animals
79 [8] as well as sophisticated fluorescence-based optics (for another example in
80 cockroaches see [9]). Marking insect leg segments is a time-consuming
81 process that limits experimental throughput. On the other hand, the most high-
82 throughput approach for marker-independent leg tracking in freely behaving
83 *Drosophila* uses complex optics to measure Total-Internal-Reflection
84 Fluorescence (TIRF) when the distal leg tips (claws) of walking animals
85 scatter light transmitted through a transparent floor [10]. Although this method
86 can resolve the claws of each leg it cannot detect their segments. Thus, it
87 provides only binary information about whether or not a leg is touching the

88 surface and cannot resolve the velocity of legs during swing phases, stance
89 adjustments, or non-locomotive limb movements such as reaching [11] or
90 grooming [12].

91 Here we describe a new method that permits semi-automated, marker-
92 free tracking of the body and leg segments of freely walking *Drosophila*. We
93 implement this method in an open source software plugin for Icy named
94 FlyLimbTracker. Our approach uses active contours (i.e., snakes) to process
95 objects in high-frame-rate image sequences. Thus, it does not require
96 complicated optical setups. While there are a number of active contour
97 algorithms [13], here we use parametric spline-snakes. These global-purpose,
98 semi-automated image segmentation algorithms are typically used in two
99 steps. First, the user roughly initializes a curve to a feature in an image (e.g.,
100 a fly's body or leg). Second, the curve's shape is automatically optimized to fit
101 the boundaries of the object of interest. Therefore, segmentation algorithms
102 using spline-snakes are composed of two major components: a *spline curve*
103 or *model* that defines how the snake is represented in the image, and a *snake*
104 *energy* that dictates how the curve is deformed in the image plane during
105 optimization. Spline-snake models have a number of advantages to other
106 approaches: they are (i) composed of only a few parameters, (ii) very flexible,
107 (iii) amenable to easy manual edits, and (iv) formed from continuously defined
108 curves that permit refined data analysis. Such models have therefore become
109 widely used for image segmentation in medium-throughput biological
110 applications [14,15]. Using this approach, we show that FlyLimbTracker can
111 semi-automatically track freely walking or grooming *Drosophila melanogaster*
112 in video data that spans a wide range of spatial and temporal resolutions.

113 FlyLimbTracker is written as a plug-in for Icy, an open-source, community-
114 maintained, and user-friendly image processing environment for biological
115 applications [16-18]. This makes it amenable to customization for behavioral
116 measurements in other species.

117 **Materials and Methods**

118 ***Drosophila behavior experiments***

119 We performed experiments using adult female *Drosophila*
120 *melanogaster* of the *Canton-S* strain at 2-4 days post-eclosion. Flies were
121 raised on a 12 h light:12 h dark cycle at 25°C. Experiments were performed in
122 the late afternoon Zeitgeber time after flies were starved for 4-6 h in
123 humidified 25°C incubators.

124 During experiments, we placed flies in a custom designed acrylic arena
125 (pill shaped: 30 mm x 5 mm x 1.2 mm) illuminated by a red ring light
126 (FALCON Illumination MV, Offenau, Germany). We captured behavioral video
127 using a high-speed (236 frames-per-second), high-resolution (2560 x 918
128 pixels) camera (Gloor Instruments, Uster Switzerland).

129

130 ***Automated body and leg tracking***

131 FlyLimbTracker is implemented in Java as a freely available plug-in for
132 Icy, a cross-platform, multi-purpose image processing environment [16].
133 Briefly, FlyLimbTracker performs leg segment tracking in several steps. First,
134 the user is asked to manually initialize the position of a fly's body and leg
135 segments in a single frame of the image sequence. This information is
136 combined with image features to propagate body and leg segmentation to the

137 frames immediately preceding, or following this first frame. At any time, the
138 user can stop, edit, and restart automated segmentation. Manual corrections
139 are taken into account when tracking is resumed.

140 To perform image segmentation, FlyLimbTracker uses active contour
141 models (i.e., snakes). A snake [19] is defined as a curve that is optimized from
142 an initial position - usually specified by the user - toward the boundary of an
143 image object. Evolution of the curve's shape results from solving an
144 optimization problem in which a cost function, or snake energy, is minimized.
145 Thus, snakes are an effective hybrid, semi-automated algorithm in which user
146 interactions define an initial position from which automated segmentation
147 proceeds [20,21]. Specifically, FlyLimbTracker first uses a *closed* snake to
148 segment the *Drosophila* body into a head, thorax, and abdomen. Then, *open*
149 snakes are used to model each of the fly's legs. Manual mapping of these
150 snakes onto the fly in an initial frame is the basis for subsequent tracking.

151 *Drosophila body model*

152 We designed a custom snake model to segment and track the
153 *Drosophila* body. In our model, the fly's body is defined as a 2-dimensional
154 closed curve \mathbf{r} :

$$155 \quad \mathbf{r}(t) = \begin{pmatrix} r_1(t) \\ r_2(t) \end{pmatrix} = \sum_{k=0}^{M-1} \mathbf{c}[k] \varphi_M(Mt - k),$$

156 with $t \in [0, M)$, where $\mathbf{c}[k] = \{(c_1[k] \ c_2[k])^T\}_{k \in \mathbb{Z}}$ is an M -periodic sequence of
157 control points and $\varphi_M(t) = \sum_{n=-\infty}^{\infty} \varphi(t - Mn)$ the M -periodization of a basis
158 function φ . For a thorough description of the spline snake formalism, see [13].
159 The proposed model for the body of the fly consists of an $M=18$ nodes snake
160 using the ellipse-reproducing basis [22]

$$\varphi(t) = \begin{cases} \frac{\cos\left(\frac{2\pi|t|}{M}\right)\cos\left(\frac{\pi}{M}\right) - \cos\left(\frac{2\pi}{M}\right)}{1 - \cos\left(\frac{2\pi}{M}\right)}, & 0 \leq |t| < \frac{1}{2}, \\ \frac{1 - \cos\left(\frac{2\pi\left(\frac{3}{2} - |t|\right)}{M}\right)}{2\left(1 - \cos\left(\frac{2\pi}{M}\right)\right)}, & \frac{1}{2} \leq |t| < \frac{3}{2}, \\ 0, & |t| \geq \frac{3}{2}. \end{cases}$$

161 To optimize the snake automatically from a coarse initial position to the
 162 precise boundaries of the fly's body, we define a snake energy composed of
 163 three elements:

$$164 \quad E_{\text{body}} = E_{\text{edge}} + E_{\text{region}} + E_{\text{shape}}.$$

165 The first element E_{edge} is an edge-based energy term relying on
 166 gradient information to detect the body contour, which is formally expressed
 167 as

$$168 \quad E_{\text{edge}} = - \oint_{\mathbf{r}} \mathbf{k}^T (\nabla I(x, y) \times d\mathbf{x}),$$

169 where $d\mathbf{x}$ is the infinitesimal vector tangent to the snake, $\nabla I(x, y)$ the in-plane
 170 gradient of the image at position (x, y) , and $\mathbf{k} = (0, 0, 1)$ is the vector
 171 orthonormal to the image plane. The energy term is negative since it has to be
 172 minimized during the optimization process. Using Green's theorem, we can
 173 transform the line integral into a surface integral:

$$174 \quad E_{\text{edge}} = - \int_{\Omega} \Delta I(\mathbf{x}) d\mathbf{x}.$$

175 The second term, E_{region} , is a region energy term that uses region
 176 statistics to segment the object from the background. Specifically, it is
 177 computed as the intensity difference between the region enclosed by the
 178 snake and the region surrounding it, as

179
$$E_{\text{region}} = \frac{1}{|\Omega|} \left(\int_{\Omega} I(\mathbf{x}) d\mathbf{x} - \int_{\Omega_{\lambda} \setminus \Omega} I(\mathbf{x}) d\mathbf{x} \right),$$

180 where I is the image and $|\Omega|$ the signed area of the snake, which is defined as

181
$$|\Omega| = \oint_{\mathbf{r}} x_2 dx_1.$$

182 Minimizing this term encourages the snake to maximize the contrast between
183 the area it encloses and the background. For more details about the edge and
184 region energy derivations, see [23,24].

185 Finally, the last term, E_{shape} , corresponds to the shape-prior energy
186 contribution detailed in [25]. This term measures the similarity between the
187 snake and its projection on a given reference curve. It therefore encourages
188 the convergence of the contour to an affine transformation of the reference
189 shape. The smoothness and regularity of the reference are preserved.
190 Moreover, this term prevents the formation of loops and aggregation of nodes
191 during the optimization process. In our case, the reference shape is a
192 symmetric 18-node fly body contour (Fig. 1A,F).

193

194 **Figure 1. FlyLimbTracker uses active contour models to annotate the**
195 ***Drosophila* body and legs. (A)** The body model is a closed snake consisting
196 of 18 control points ($c[0]$ to $c[17]$). Control points $c[0]$ and $c[9]$ correspond,
197 respectively, to the posterior-most position on the abdomen and the anterior-
198 most position on the head. All other control points are symmetric along the
199 anteroposterior axis of the body (e.g., control points $c[3]$ and $c[15]$). **(B)** Six
200 leg anchor positions (yellow) between the coxa and thorax are defined
201 empirically based on a linear combination of distances from the head-thorax
202 boundary, the thorax-abdomen boundary, and a distance from the thoracic
203 midline. These positions are then shifted depending on how the body model is

204 optimally deformed to fit the contours of a specific animal. **(C)** The leg model
205 consists of four control points including a thorax-coxa attachment $I[0]$, the
206 femur-tibia joint $I[1]$, the tibia-tarsus joint $I[2]$, and the pretarsus/claw $I[3]$. For
207 simplicity, control points for only a single leg are shown. **(D)** In sum, 27
208 positions are calculated for each fly per frame: a centroid (0), anterior point
209 (A), posterior point (P), as well as the body anchor, first intermediate, second
210 intermediate and tip for each of the six legs. Our data labeling convention is
211 as follows. Right and left legs are numbered 1 to 3 (front to rear) and 4 to 6
212 (front to rear), respectively. Each leg has four control points labeled 1 to 4 in
213 the units digit that correspond the body anchor (1), leg joints (2 and 3), and
214 claw (4). In each label, the leg number is shown in the tenths digit and the
215 control point in the units digit. For example, the label “11” refers to the body
216 anchor of the right prothoracic leg 1. For simplicity, only the control points for
217 leg 3 are shown. **(E)** An example raw image of the ventral surface of a fly
218 used for segmentation. **(F)** This image is first segmented using the parametric
219 body snake consisting of 18 control points (red and blue crosses). **(G)**
220 Subsequently, leg segmentation is initialized through automatic tracing from
221 body anchor points to user-defined leg tips. From this initialization, an
222 annotation is performed using open snakes consisting of four control points
223 (yellow crosses). **(H)** Body and **(I)** leg segment tracking annotation for flies
224 during a 455-frame (1.93 s) sequence. Annotation results (red) and the
225 centroid in **H** or leg tip positions in **I** (blue) for each frame are overlaid.

226

227 To automatically optimize the snake, we modified the position of the
228 control points by minimizing the energy using a Powell-like line-search

229 method [26], a standard unconstrained optimization algorithm that converges
230 quadratically to an optimal solution. First, one direction is chosen depending
231 on the partial derivatives of the energy, which is computed using finite
232 differences. Second, a one-dimensional minimization of the energy function is
233 performed in the selected direction. Finally, a new direction is chosen using
234 the partial derivatives and enforcing conjugation properties. These steps are
235 repeated until convergence. The final configuration of the control points
236 provides an accurate description of the orientation and size of the fly body.

237 In practice, the algorithm depends on initial user input to coarsely
238 locate the fly in a frame of the image sequence. Following a single mouse
239 click, a two-step multiscale optimization scheme inspired by [24] is initiated. A
240 spherical active contour composed of 3-control points is first created, centered
241 at the mouse position. This snake is optimized using $E_{\text{edge}} + E_{\text{region}}$ to form an
242 elliptic curve surrounding the fly. In this way, the major axis of the elliptical
243 snake will be aligned with the anteroposterior axis of the fly, and the minor
244 axis will be perpendicular to it.

245 The 3-point elliptical snake fit to the body of the fly can be expressed
246 as follows [23]:

$$247 \quad \mathbf{r}(t) = \mathbf{R}_0 + \mathbf{R}_1 \cos(2\pi t) + \mathbf{R}_2 \sin(2\pi t),$$

248 where

$$249 \quad \mathbf{R}_0 = \frac{1}{3} \sum_{k=0}^2 \mathbf{c}[k], \quad \mathbf{R}_1 = \sum_{k=0}^2 h_c[k] \mathbf{c}[k], \quad \mathbf{R}_2 = \sum_{k=0}^2 h_s[k] \mathbf{c}[k],$$

250 and

$$251 \quad h_c[k] = \frac{2}{3} \cos\left(\frac{\pi}{3}\right) \cos\left(\frac{2\pi k}{3}\right), \quad h_s[k] = \frac{2}{3} \cos\left(\frac{\pi}{3}\right) \sin\left(\frac{2\pi k}{3}\right).$$

252 Relating this to the general parametric equation of an ellipse of major axis a ,
253 minor axis b , and center $(x_c \ y_c)^T$ allows us to extract the parameters of the 3-

254 control point snake fit to the fly's body. Namely, $(x_c \ y_c)^T = \mathbf{R}_0$,
255 $a = \max(\|\mathbf{R}_1\|, \|\mathbf{R}_2\|)$ and $b = \min(\|\mathbf{R}_1\|, \|\mathbf{R}_2\|)$. By knowing a , the
256 orientation of the ellipse in the image can be computed.

257 The ellipse fit is then replaced by an 18-node fly-shaped closed snake
258 that has been rotated and dilated to match the ellipse's length and orientation
259 (Fig. 1A). An ambiguity results since two potential snake models can be
260 initialized for a given ellipse, with opposite anteroposterior axis orientation. To
261 resolve this ambiguity, both potential snake orientations are optimized on the
262 image using E_{body} in addition to E_{edge} and E_{region} . The solution with the lowest
263 cost (i.e., energy value at convergence) is used.

264 *Drosophila leg model*

265 Once the fly's body is properly segmented, open snake models for
266 each of its legs are then added. First, the positions of leg coxa-thorax
267 attachment points (hereafter referred to as *anchors*) are automatically
268 computed based on the body segmentation. The location of the six leg
269 anchors with respect to the reference body model have been empirically
270 determined as linear combinations of three axes defined by the head-thorax
271 junction, the thorax-abdomen junction and the thorax length (Fig. 1B). These
272 locations are then adapted according to an individual fly-specific deformation
273 of the body model.

274 User input is required to initialize the positions of each leg prior to
275 tracking. Initialization is based on a single click for each leg: the user indicates
276 the claw (hereafter referred to as *tip*) of each leg through mouse-clicks on the
277 selected frame. The click location is assigned to the most likely body anchor
278 using a probabilistic formulation based on the distance and intersection with

279 the fly's body model and that of other leg models. Once a leg tip and a leg
280 anchor have been paired, a dynamic programming method [27] is initiated to
281 automatically trace the leg from the anchor to the tip. To facilitate this process,
282 the fly's legs are enhanced by processing the segmented image frame using a
283 ridge detector [28].

284 Dynamic programming is a method that yields the globally optimal
285 solution for a given separable problem. In particular, it can be used to
286 implement algorithms solving shortest path problems. Dynamic programming
287 relies on a graph-based representation: the shortest path is represented as a
288 sequence of successive nodes in a graph that minimize a cost function. To
289 trace a leg from its anchor to its tip, we build a graph by interpolating image
290 pixels along the two axes using a straight segment linking the anchor to the tip
291 (axis \mathbf{k}) and its normal vector (axis \mathbf{u}). The cost of the path at index $k + 1$
292 along axis \mathbf{k} is then given by:

$$293 \quad C[k + 1] = C[k] + \lambda \left(\frac{1}{L_S} \sum_{(x,y) \in S} I_{\text{ridge}}(x, y) \right) + (1 - \lambda) |u_k - u_{k+1}|,$$

294 where $C[i]$ is the cost of the path at location i on axis \mathbf{k} , S is the collection of
295 image pixels (x, y) in the segment between node (k, u_k) and $(k + 1, u_{k+1})$, L_S
296 is the pixel length of this segment, I_{ridge} is the ridge-filtered version of current
297 frame, and $\lambda \in [0, 1]$ is a weighting coefficient. The first term corresponds to a
298 discretized integral of the image in the segment linking nodes k and $k + 1$,
299 and therefore tends to favor paths going through low pixel values. The second
300 term is composed of the distance along axis \mathbf{u} between two successive
301 nodes. As a result, the optimal path follows relatively bright (or dark) regions
302 in the image with respect to the background, while retaining a certain level of
303 smoothness. The relative contribution of each term is determined by λ .

304 In contrast to body segmentation, leg segmentation uses open rather
305 than closed snakes. Fly legs are parameterized by a curve composed of
306 $M = 4$ control points (Fig. 1C,G). For each leg, the body anchor, $\mathbf{I}[0]$, is
307 considered fixed. The discrete path obtained through dynamic programming is
308 used to initialize the leg snake. The rationale behind this two-step procedure
309 is two-fold. First, dynamic programming is very robust and can therefore
310 effectively trace the leg from a body anchor to its tip. However, since it is a
311 discrete approach, it is computationally expensive. By contrast, snake-based
312 methods are more likely to diverge when initialized far from their target but are
313 computationally inexpensive since only a few control points need to be stored
314 to characterize a given curve. Therefore, we combined these approaches by
315 first finding a path to define each leg using dynamic programming and then
316 transforming this path into a parametric curve for optimization. The parametric
317 representation of the leg snake curve is defined as

318
$$\mathbf{s}(t) = \begin{pmatrix} s_1(t) \\ s_2(t) \end{pmatrix} = \sum_{k=0}^{M-1} \mathbf{I}[k] \varphi(Mt - k),$$

319 where $t \in [0, M - 1]$ and $\mathbf{I}[k] = \{(l_1[k] \ l_2[k])^T\}_{k \in \mathbb{Z}}$ are the leg snake control
320 points. Since *Drosophila* legs are composed of relatively straight segments
321 between each joint, we use linear splines as basis functions $\varphi(t)$. The leg
322 control points are therefore linked through linear interpolation and each
323 control point has a unique identifier that can be used for subsequent data
324 processing (Fig. 1D). Figure 1E-G illustrates the full process of taking a single
325 raw image (Fig. 1E) and using active contours to segment the body (Fig. 1F)
326 and legs (Fig. 1G).

327

328 *Segmentation propagation (tracking)*

329 High frame-rate videos ensure that the displacement of a fly's body
330 between successive frames is small. FlyLimbTracker takes advantage of this
331 fact to propagate body and leg snakes from one frame to the next during
332 tracking. The body snake in frame $t+1$ is therefore segmented by optimizing a
333 contour initialized as the corresponding snake from frame t using the body
334 snake energy previously described. This approach is sufficient to obtain good
335 segmentation provided that there is some overlap between the animal's body
336 in frames t and $t+1$.

337 Compared with the body, leg displacement can be larger between
338 frames. Therefore, leg snakes require a more sophisticated algorithm to be
339 propagated during tracking. First, the anchor of each leg is automatically
340 computed from the newly propagated fly body. Since each leg is modeled as
341 a 4-node snake, the three remaining leg snake control points are optimized
342 using the snake energy

$$343 \quad E_{\text{leg}} = E_{\text{ridge}} + E_{\text{EDT}} + E_{\text{segments}} + E_{\text{extremity}}.$$

344 The first term corresponds to the integral along the leg in the current frame
345 filtered by a ridge detector [28], i.e.,

$$346 \quad E_{\text{ridge}} = \int_C I_{\text{ridge}} ds = \int_0^1 I_{\text{ridge}}(\mathbf{r}(t)) |\mathbf{r}'(t)| dt.$$

347 Analogous with the first term, the second term is computed as the integral
348 along the leg of the Euclidean distance transform (EDT, [29]) in the current
349 frame where

$$E_{\text{EDT}} = \int_C I_{\text{EDT}} ds = \int_0^1 I_{\text{EDT}}(\mathbf{r}(t)) |\mathbf{r}'(t)| dt.$$

350 Each of the linear segments comprising a fly's legs should be roughly
351 constant in length across a video, aside from changes introduced by

352 projecting the three-dimensional legs onto two-dimensional images. Taking
353 this consistency into account, the third term of the leg energy penalizes
354 solutions for which the leg joint positions result in leg segments whose lengths
355 vary considerably from one frame to the next. This prevents unrealistic
356 configurations of the leg joints that yield excessively long leg segments
357 compared with neighboring annotated frames.

358 Finally, the fourth term is used to determine the leg tip position at
359 time t , denoted $\mathbf{l}_t[3]$. Since the distal tip of the leg may move considerably
360 between successive frames, we designed a dedicated energy term to attract
361 the tip toward candidate locations in the image. These candidate locations are
362 defined by minima after the image is filtered using a Laplacian-of-Gaussian
363 (LoG, [30]). A potential map of tip candidates is then created according to:

$$E_{\text{extremity}} = 1 - w_{\mathbf{p}^*} e^{-\frac{\|\mathbf{l}_t[3] - \mathbf{p}^*\|^4}{\sigma^2}},$$

364 where

$$\mathbf{p}^* = \operatorname{argmin}_{\mathbf{p} \in P} \|\mathbf{l}_t[3] - \mathbf{p}\|^2$$

365 is the tip candidate closest to $\mathbf{l}_t[3]$, $w_{\mathbf{p}^*} \in [0,1]$ its associated weight, and σ^2
366 a fixed parameter determining the width of the attraction potential of the tip
367 candidates. The weight $w_{\mathbf{p}^*}$ is a measure of how tip-like \mathbf{p}^* is, and is
368 computed based on the magnitude of the LoG filter response. A strong weight
369 results in a deeper potential, and is therefore more likely to attract $\mathbf{l}_t[3]$.

370 In summary, the four anchor points characterizing each leg are
371 propagated as follows. First, the leg body anchors are determined using the
372 body model. Second, the remaining three control points (two leg joints and tip)
373 are shifted by optimizing a cost function that incorporates both image

374 information (E_{ridge} and E_{EDT}) and a smoothness constraint (E_{segments}). Finally,
375 the tip is further constrained using an estimation of how tip-like the image is at
376 candidate locations.

377

378 *Data output*

379 Once the full image sequence is annotated, data can be extracted as a
380 CSV file for each fly. These measurements include the locations of three
381 reference points on the fly's body (A, P, and O), as well as each of the legs'
382 anchor points (see Fig. 1D for the labeling convention).

383 FlyLimbTracker is linked to Icy's Track Manager plugin (Publication Id:
384 ICY-N9W5B7) via the *extract tracks* buttons (see interface description in the
385 Appendix), allowing additional data to be extracted. In particular,
386 segmentations of the fly's body (Fig. 1H) and legs (Fig. 1I) can be visualized
387 across the entire sequence, illustrating their entire trajectories. Each individual
388 control point of the leg snakes or the body snake's centroid can be
389 independently visualized. Note that tracks are also numbered according to the
390 labeling convention in Fig. 1D.

391

392 *Software and data availability*

393 User instructions, FlyLimbTracker software, and sample data can be
394 found at:

395 <http://bigwww.epfl.ch/algorithms/FlyLimbTracker/>

396 **Results**

397 FlyLimbTracker performs semi-automated body and leg tracking. First,

398 the user manually initializes the positions of the fly's body and leg segments in
399 a single, arbitrarily chosen frame of the image sequence (Fig. 2A). These
400 manual annotations are then used to automatically propagate segmentation to
401 prior, or subsequent frames (Fig. 2B). During automated segmentation, the
402 user can interrupt tracking to correct errors (Fig. 2C). When FlyLimbTracker is
403 restarted, the automated segmentation continues, taking into account these
404 user edits.

405

406 **Figure 2. FlyLimbTracker workflow. (A)** The user manually indicates the
407 approximate location of the fly's body in an arbitrarily chosen video frame (t_1).
408 FlyLimbTracker then optimizes a closed active contour model that
409 encapsulates the fly's body in the correct orientation. The user then manually
410 indicates the location of each leg's tip. FlyLimbTracker then optimizes an
411 open active contour model that runs across the entirety of each leg. **(B)** The
412 user then runs FlyLimbTracker's automatic tracking algorithm to propagate
413 body and leg models to subsequent video frames (or prior frames if run in
414 reverse). **(C)** Either during or after automated tracking, the user can look for
415 tracking errors. After manually correcting these errors, the user can re-run
416 automatic tracking. In each image, the frame number is indicated.

417

418 *Algorithm robustness*

419 FlyLimbTracker can be used to segment and track fly bodies and legs
420 in videos spanning a wide range of spatial and temporal resolutions.
421 Resolution determines the nature of the annotation process: high-resolution

422 data tracking is more automated, while low resolution data requires more user
423 intervention. To quantify the dependence of computing time and the number
424 of user interventions on data quality, we systematically varied the spatial and
425 temporal resolutions of videos featuring five common *Drosophila* behaviors:
426 walking straight, turning, foreleg grooming, head grooming, and abdominal
427 grooming. Raw videos were originally captured at 236 fps and at 2560 x 918
428 pixel resolution (Supplementary Videos 1-5).

429 First, we studied FlyLimbTracker's robustness to variations in spatial
430 resolution. We down-sampled each of the five videos by a factor of N , where
431 $N \times N$ pixels were averaged. This resulted in image sequences N times
432 smaller along both spatial dimensions but with an identical temporal resolution
433 of 236 fps (Fig. 3A). Alternatively, to vary temporal resolution, we down-
434 sampled each video by a factor of N , where only one frame from every N was
435 retained. This resulted in image sequences of varying temporal resolution but
436 consistently high spatial resolution of 2560 x 918 pixels (Fig. 3B).

437

438 **Figure 3. Sensitivity of leg tracking to changes in spatial or temporal**
439 **video resolution. (A)** Sample video image (top-left) after 2x (top-right), 4x
440 (bottom-left), or 8x (bottom-right) spatial down-sampling. **(B)** Representations
441 of the difference between successive images (t_1 and t_2 overlaid in magenta
442 and green, respectively) for different frame rate videos after temporal down-
443 sampling. **(C-D)** The number of corrections required per node per frame as a
444 function of spatial resolution **(C)**, or temporal resolution **(D)**. **(E-F)** The
445 average time required to annotate a single frame as a function of spatial
446 resolution **(E)**, or temporal resolution **(F)**. In **C-F**, data for videos depicting a fly

447 walking straight, turning, grooming its forelegs, head, or abdomen are shown
448 in orange, purple, green, cyan, and red, respectively.

449

450 For each movie, body and leg snakes were manually initialized using
451 the first image frame. Segmentation was then automatically propagated
452 forward through the remainder of the image sequence. Whenever the
453 automated tracker made a mistake, the process was interrupted and the user
454 manually corrected the error. Automated tracking was then restarted from this
455 frame until the next mistake was observed. In all cases, automated body
456 tracking did not require manual intervention. Therefore, we only took note of
457 manual corrections in leg snake annotation.

458 To quantify FlyLimbTracker's performance across this range of spatial
459 and temporal resolutions, we calculated two normalized quantities. First, we
460 calculated the average number of manual corrections per node per frame
461 (Fig. 3C-D). To do this, we measured the total number of user interventions
462 while processing an image sequence and normalized this quantity by $T \times 6 \times 3$,
463 where T is the number of frames, each of which contains eighteen free
464 parameters: six legs with three editable control points each. As a second
465 metric we quantified the average time required to annotate a single image
466 frame (Fig. 3E-F). To do this, we recorded the total time required to annotate
467 an image sequence and divided this value by the total number of frames. This
468 normalized quantity combines both the computing time required for automated
469 annotation as well as the time required to manually correct annotation errors.

470 Overall, we observed that reducing spatial (Fig. 3A,C,E), or temporal

471 (Fig. 3B,D,F) resolution resulted in an increase in the number of manual
472 interventions (Fig. 3C-D) as well as a longer time required for annotation (Fig.
473 3E-F). While the numbers of corrections were similar for equivalent amounts
474 of down-sampling (up to 8-fold), annotation time was appreciably longer for
475 straight walking and turning. This reflects the importance of having
476 overlapping images in successive frames for automated tracking: a feature
477 that may be less common during locomotion where the position of a leg can
478 vary substantially within a walking cycle. Notably, in a number of other cases
479 (e.g., grooming), the annotation time per frame flattens across spatial and
480 temporal resolutions. This is probably due to the trade-off between automated
481 processing and manual correction times. Resolution strongly influences the
482 computing time required for automated tracking: smaller or fewer images can
483 be processed more quickly. However, as resolution decreases, user
484 interventions required to correct errors begin to dominate annotation time
485 required to annotate each frame.

486 *Visualization and analysis of leg segment tracking data*

487 FlyLimbTracker provides a user-friendly interface that allows body and
488 leg segment tracking data to be exported in a CSV file format, simplifying data
489 analysis and visualization. We illustrate three representations of body and leg
490 tracking data for annotated videos of the five behaviors previously described
491 (Supplementary Videos 6-10). First, within FlyLimbTracker itself, leg joint
492 and/or body trajectories can be displayed overlaid upon the final raw video
493 frame (Fig. 4A₁-E₁). This representation provides a way to project time-
494 varying data onto a static image and illustrates the symmetric or asymmetric
495 limb motions that control straight walking/grooming or turning, respectively.

496 Second, leg segment trajectory data can be exported and processed
497 externally (e.g., using Matlab or Python). These data can be rotated along
498 with the fly's frame of reference (Fig. 4A₂-E₂) for a direct comparison of leg
499 segment movements between distinct actions. A similar approach has been
500 used to visualize how neurogenetic perturbations influence claw movements
501 during locomotion [10], but can now be used to study the effects of these
502 manipulations on other previously inaccessible leg segments and behaviors
503 (e.g., grooming or reaching). In a third visualization, the speeds of each claw
504 can be plotted to provide an exceptionally detailed characterization of
505 locomotor gaits (Fig. 4A₃-B₃), or grooming movements in stationary animals
506 (Fig. 4C₃-E₃).

507

508 **Figure 4. Analysis and visualization of FlyLimbTracker leg tracking data.**

509 Visualizations of leg segment annotation results for videos of a fly **(A)** walking
510 straight, **(B)** turning, **(C)** grooming its forelegs, **(D)** grooming its head, or **(E)**
511 grooming its abdomen. **(A₁-E₁)** Leg segmentation results (red) and joint
512 positions (color-coded by frame number) are overlaid on the final frame of the
513 image sequence. **(A₂-E₂)** Leg segment trajectories are rotated and color-
514 coded by frame number. This permits alignment and comparison of leg
515 movements across different datasets. **(A₃-E₃)** The instantaneous speeds of
516 each leg tip (claw) are color-coded.

517

518 **Discussion**

519 Existing methods for tracking insect leg segments rely on sophisticated

520 optical equipment and/or laboriously-applied leg markers, often in tethered
521 animals [8-10]. While these approaches are extremely valuable, they may
522 potentially disrupt natural behaviors and cannot report the motions of multiple
523 joints in untethered animals. Here we have introduced a method that uses
524 computer-vision techniques to address these technical barriers. The software
525 implementation of this approach, FlyLimbTracker, permits semi-automated
526 tracking of body and leg segments in freely behaving *Drosophila*. Use of
527 FlyLimbTracker only requires a single high-resolution, high-speed camera and
528 does not require prior marking of leg segments. Additionally, it can be used
529 with video data across a range of spatial and temporal resolutions, permitting
530 a flexible blend of automated and manual annotation. Importantly, when
531 automation has difficulty segmenting low quality data, FlyLimbTracker
532 remains a powerful tool for manual leg tracking annotation since it uses easily
533 manipulated spline-snakes and provides an interface for user-friendly data
534 import and export.

535 The open-source nature of FlyLimbTracker can facilitate community-
536 driven improvement and customization of the algorithm. We can envision a
537 number of improvements moving forward. First, tracking currently requires
538 overlap of a fly's body between successive frames. This constraint places a
539 lower bound on video temporal resolution and could be improved by using, for
540 example, nearest-neighbor matching approaches like the Hungarian algorithm
541 [31] to link segmentation control points between successive frames. Second,
542 additional leg control points may be added to FlyLimbTracker to more
543 precisely annotate thorax-coxa-trochanter segments. Third, FlyLimbTracker's
544 requirement of user initialization, makes it only semi-automated and restricts

545 batch processing of multiple videos for high-throughput data analysis. This
546 may be overcome using additional prior information to automatically identify
547 and optimize body snakes. Fourth, FlyLimbTracker's snake-based approach
548 to tracking could easily be adapted for the study of other species (e.g., mice,
549 stick insects, and cockroaches) by modifying the shape of snake models.

550 **Acknowledgements**

551 We thank Cédric Vonesch, Michael Rusterholz, and Loic Perruchoud for early
552 contributions to the body tracking algorithm.

553 **Funding**

554 VU, RDG and MU were supported by the Swiss SystemsX.ch Initiative
555 (2008/005) and the Swiss National Science Foundation for the Promotion of
556 Scientific Research (200020-121763 and 200020-144355). PR was supported
557 by a Human Frontier Science Program Fellowship (LT000057/2009) and a
558 Swiss National Science Foundation Fellowship (P300P3_158511). RB
559 acknowledges support from European Research Council Starting Independent
560 Researcher and Consolidator Grants (205202 and 615094).

561

562 References

- 563 1. Olsen SR, Wilson RI. Cracking neural circuits in a tiny brain: new
564 approaches for understanding the neural circuitry of *Drosophila*. Trends
565 Neurosci. 2008;31: 512–520.
- 566 2. Noldus L, Spink AJ, Tegelenbosch R. Computerised video tracking,
567 movement analysis and behaviour recognition in insects. Computers
568 and Electronics in agriculture. 2002;35: 201–227.
- 569 3. Dankert H, Wang L, Hoopfer E, Anderson DJ, Perona P. Automated
570 monitoring and analysis of social behavior in *Drosophila*. Nature
571 Methods. 2009;6: 297.
- 572 4. Branson K, Robie AA, Bender JA, Perona P, Dickinson MH. High-
573 throughput ethomics in large groups of *Drosophila*. Nature Methods.
574 2009;6: 451–457.
- 575 5. Donelson N, Kim EZ, Slawson JB, Vecsey CG, Huber R, Griffith LC.
576 High-Resolution Positional Tracking for Long-Term Analysis of
577 *Drosophila* Sleep and Locomotion Using the “Tracker” Program. PLoS
578 One. 2012;7: e37250.
- 579 6. Pérez-Escudero A, Vicente-Page J, Hinz RC, Arganda S, de Polavieja
580 GG. idTracker: tracking individuals in a group by automatic identification
581 of unmarked animals. Nature Methods. 2014;11: 743–748.
- 582 7. Deng Y, Coen P, Sun M, Shaevitz JW. Efficient Multiple Object Tracking
583 Using Mutually Repulsive Active Membranes. PLoS One. 2013;8:
584 e65769.
- 585 8. Kain J, Stokes C, Gaudry Q, Song X, Foley J, Wilson RI, et al. Leg-
586 tracking and automated behavioural classification in *Drosophila*. Nature
587 Communications. 2013;4: 1910–1918.
- 588 9. Bender JA, Simpson EM, Ritzmann RE. Computer-Assisted 3D
589 Kinematic Analysis of All Leg Joints in Walking Insects. PLoS One.
590 2010;5: e13617.
- 591 10. Mendes CS, Bartos I, Akay T, Márka S, Mann RS. Quantification of gait
592 parameters in freely walking wild type and sensory deprived *Drosophila*
593 *melanogaster*. eLife. 2013;2.
- 594 11. Pick S, Strauss R. Goal-Driven Behavioral Adaptations in Gap-Climbing
595 *Drosophila*. Current Biology. 2005;15: 1473–1478.
- 596 12. Seeds AM, Ravbar P, Chung P, Hampel S, Midgley FM, Mensh BD, et
597 al. A suppression hierarchy among competing motor programs drives
598 sequential grooming in *Drosophila*. eLife. 2014;3.
- 599 13. Delgado-Gonzalo R, Uhlmann V, Schmitter D, Unser M. Snakes on a
600 Plane: A perfect snap for bioimage analysis. IEEE Signal Process Mag.

- 601 2015;32: 41–48.
- 602 14. Dénervaud N, Becker J. A chemostat array enables the spatio-temporal
603 analysis of the yeast proteome. 2013. pp. 15842–15847.
- 604 15. Schmitter D, Wachowicz P, Sage D, Chasapi A, Xenarios I, Simanis V,
605 et al. A 2D/3D image analysis system to track fluorescently labeled
606 structures in rod-shaped cells: application to measure spindle pole
607 asymmetry during mitosis. *Cell Division*. 2013;8.
- 608 16. de Chaumont F, Dallongeville S, Chenouard N, Hervé N, Pop S,
609 Provoost T, et al. Icy: an open bioimage informatics platform for
610 extended reproducible research. *Nature Methods*. 2012;9: 690–696.
- 611 17. de Chaumont F, Coura RD-S, Serreau P, Cressant A, Chabout J,
612 Granon S, et al. Computerized video analysis of social interactions in
613 mice. *Nature Methods*. 2012;9: 410–417.
- 614 18. Chenouard N, Buisson J, Bloch I, Bastin P, Olivo-Marin J-C. Curvelet
615 analysis of kymograph for tracking bi-directional particles in
616 fluorescence microscopy images. *IEEE 17th International Conference
617 on Image Processing*. 2010;;: 3657–3660.
- 618 19. Kass M, Witkin A, Terzopoulos D. Snakes: Active contour models.
619 *International journal of computer vision*. 1988;;: 321–331.
- 620 20. Delgado-Gonzalo R, Chenouard N, Unser M. Spline-Based Deforming
621 Ellipsoids for Interactive 3D Bioimage Segmentation. *IEEE Transactions
622 on Image Processing*. 2013;22: 3926–3940.
- 623 21. Brigger P, Hoeg J, Unser M. B-spline snakes: a flexible tool for
624 parametric contour detection. *IEEE Transactions on Image Processing*.
625 2000;9: 1484–1496.
- 626 22. Delgado-Gonzalo R, Thévenaz P, Unser M. Computer Aided Geometric
627 Design. *Computer Aided Geometric Design*. Elsevier B.V; 2012;29:
628 109–128.
- 629 23. Delgado-Gonzalo R, Thévenaz P, Seelamantula CS, Unser M. Snakes
630 With an Ellipse-Reproducing Property. *IEEE Transactions on Image
631 Processing*. 2012;21: 1258–1271.
- 632 24. Jacob M, Blu T, Unser M. Efficient Energies and Algorithms for
633 Parametric Snakes. *IEEE Transactions on Image Processing*. 2004;13:
634 1231–1244.
- 635 25. Delgado-Gonzalo R, Schmitter D, Uhlmann V, Unser M. Efficient Shape
636 Priors for Spline-Based Snakes. *IEEE Transactions on Image
637 Processing*. 2015;24: 3915–3926.
- 638 26. Press WH, Flannery BP, Teukolsky SA, Vetterling WT. *Numerical
639 recipes: the art of scientific computing*. Cambridge University Press;

- 640 1986.
- 641 27. Dijkstra EW. A note on two problems in connexion with graphs.
642 Numerische mathematik. 1959;: 269–271.
- 643 28. Jacob M, Unser M. Design of steerable filters for feature detection using
644 canny-like criteria. IEEE Transactions on Pattern Analysis and Machine
645 Intelligence. 2004;26: 1007–1019.
- 646 29. Felzenszwalb PF, Huttenlocher DP. Distance Transforms of Sampled
647 Functions. Theory of Computing. 2012;8: 415–428. doi:10.4086/toc
- 648 30. Sage D, Neumann FR, Hediger F, Gasser SM, Unser M. Automatic
649 tracking of individual fluorescence particles: application to the study of
650 chromosome dynamics. IEEE Transactions on Image Processing.
651 2005;14: 1372–1383.
- 652 31. Kuhn HW. The Hungarian method for the assignment problem. Naval
653 research logistics quarterly. 1955;: 83–97.
- 654
- 655

656 **Supporting Information**

657 **Supporting Video Legends**

658 Raw videos used for sensitivity analyses (Fig. 3) and visualization (Fig. 4):

659 Video 1 – A fly walking straight.

660 Video 2 – A fly turning.

661 Video 3 – A fly grooming its forelegs.

662 Video 4 – A fly grooming its head.

663 Video 5 – A fly grooming its abdomen.

664 Video 6 – A fly walking straight (video 1), annotated using FlyLimbTracker.

665 Video 7 – A fly turning (video 2), annotated using FlyLimbTracker.

666 Video 8 – A fly grooming its forelegs (video 3), annotated using

667 FlyLimbTracker.

668 Video 9 – A fly grooming its head (video 4), annotated using FlyLimbTracker.

669 Video 10 – A fly grooming its abdomen (video 5), annotated using

670 FlyLimbTracker.

671

672 **Appendix**

673 ***User interface***

674 FlyLimbTracker's interface can be used in either basic or advanced
675 mode. In the basic mode, only the name of the active image is visible. All
676 parameters are hidden and only default parameter values are used. When
677 switching to the advanced mode, all parameters become visible and can be
678 adjusted by the user. Parameters that can be adjusted in the interface include:

- 679 • Image parameters

- 680 ○ Channel: for multichannel images (e.g., bright-field and
681 fluorescence), this parameter selects the channel upon which
682 segmentation is performed. In most cases, the bright-field
683 channel should be selected.
- 684 ○ Smoothing: adjusts the width (standard deviation, in pixels) of a
685 smoothing filter used to preprocess the image sequence. Larger
686 values yield smoother images, but likely obscure details such as
687 the fly's legs. We recommend choosing a value approximately
688 equal to the average width (in pixels) of the fly legs.
- 689 ○ Subtract background: performs background subtraction on the
690 image sequence. The background model used is the median of
691 each pixel across the whole image sequence. In practice,
692 background subtraction is not desirable in datasets with a low
693 signal-to-noise ratio since a fly's legs typically have low contrast
694 and can be smoothed out by median filtering.
- 695 • **Body model parameters**
- 696 ○ Annotation method: switches between automated and manual
697 annotation of the body snake. Automated annotation is obtained
698 by automatically optimizing the body snake from its initial,
699 manually chosen position. Manual annotation relies exclusively
700 on user interactions.
- 701 ○ Energy trade-off: adapts the relative importance of data fidelity
702 (image-based) and regularization (shape-based) terms in the
703 body snake energy. A fully image-based snake would be
704 optimized using image information only, while a fully shape-

705 based snake would be optimized to retain a fly's shape
706 regardless of the underlying image data. For data with low
707 image quality the regularization term (shape-based) becomes
708 more important.

709 ○ Max iterations/immortal: tunes the maximum number of
710 iterations used to optimize the body snake. If *immortal* is
711 chosen, the snake keeps evolving until it achieves convergence.
712 Allowing the snake to be immortal usually yields better
713 segmentation results, but significantly increases computing time.
714 Conversely, a smaller number of iterations can estimate
715 segmentation quickly, but not necessarily as effectively. Usually,
716 4000-5000 iterations provide a good trade-off between
717 computing time and segmentation quality. However, this value
718 should be customized according to data quality.

719 ○ Freeze snake body: when ticked, locks the control points of the
720 fly body snake, which then appear as blue instead of red. In this
721 setting, individual points cannot be further edited. This feature is
722 useful when the fly body is properly initialized and edits are done
723 on the legs only, as it prevents displacing body control points
724 when trying to select a leg control point. However, it remains
725 possible to translate, move or rotate the entire fly body.

726 • Leg model parameters

727 ○ Annotation method: switches between automated and manual
728 segmentation of the fly's legs. Although body segmentation and
729 tracking is robust even for low resolution or low signal-to-noise

730 ratio data, leg tracking is much more sensitive. Therefore, the
731 user is given the option to restrict automation to body tracking.
732 In the manual segmentation setting, the legs are simply
733 propagated by translation along with body motion and can be
734 manually adjusted post-hoc for each frame. This allows
735 FlyLimbTracker to be a useful tool for annotating either low-
736 quality or high-quality data.

- 737 ○ DP trade-off: determines the relative importance of data fidelity
738 (bright) and regularization (straight) terms when performing
739 dynamic programming (DP) to initialize the leg snakes. The
740 algorithm tries to find the optimal path between a given leg
741 anchor and tip by optimizing the trade-off between image
742 intensity (bright) and straightness (straight). Relying on image
743 brightness alone typically yields irregular movements of the fly's
744 legs since the algorithm becomes very sensitive to image noise
745 (e.g., isolated pixels of high intensity). Conversely, relying on
746 straightness alone yields, in the most extreme case, a straight
747 line between the anchor and tip. Note that this parameter is only
748 used when initializing a leg. It does not influence tracking.
- 749 ○ Energy trade-off: determines the relative importance of data
750 fidelity (image-based) and regularization (sequence-based)
751 terms for the leg snakes. A purely image-based leg snake is
752 optimized using the image data only. This typically yields
753 suboptimal solutions that are sensitive to image noise.
754 Conversely, a fully sequence-based leg snake maximizes its

755 resemblance to the corresponding leg snake from previously
756 annotated frames and ignores image data. More importance
757 should be given to sequence-based energy for low quality data
758 when leg snake annotations are readily available.

759 ○ Tip propagation mode: determines the relative importance of
760 data fidelity (image-based) and regularization (sequence-based)
761 terms while tracking leg tips. We identify potential tips by
762 searching for candidate locations in a neighborhood
763 encompassing leg motions from previously annotated,
764 neighboring frames. The final tip position is chosen as a trade-
765 off between the position predicted by leg motion from previous
766 annotated frames (sequence-based), and tip candidates
767 identified by processing the current frame (image-based).

768 ○ Max iterations/immortal: tunes the maximum number of
769 iterations used to optimize the leg snakes in a manner similar to
770 how the same parameter is used to optimize the body snake.

771 In both basic and advanced modes, the upper part of the interface
772 contains several menu items (Analyze, Save/Load and Help):

773 • Analyze: extracts measurements from the current body
774 segmentation using Icy's ROI Statistics plugin (Publication Id: ICY-
775 W5T6J4).

776 • Save/Load: allows the user to export and save annotations to a
777 CSV file format (see Output section below). This can also be used
778 to reload previously saved CSV annotations.

779 • Help: contains information about the plugin version (About), and a
780 link to FlyLimbTracker's online documentation page
781 (Documentation (online)).

782 Finally, several action buttons are located on the lower part of the
783 interface. These are split into three sections.

784 • Fly shape editing: the left button enables movement of individual
785 control points. The middle and right buttons, respectively, enable
786 resizing and rotation of the body and leg snakes.

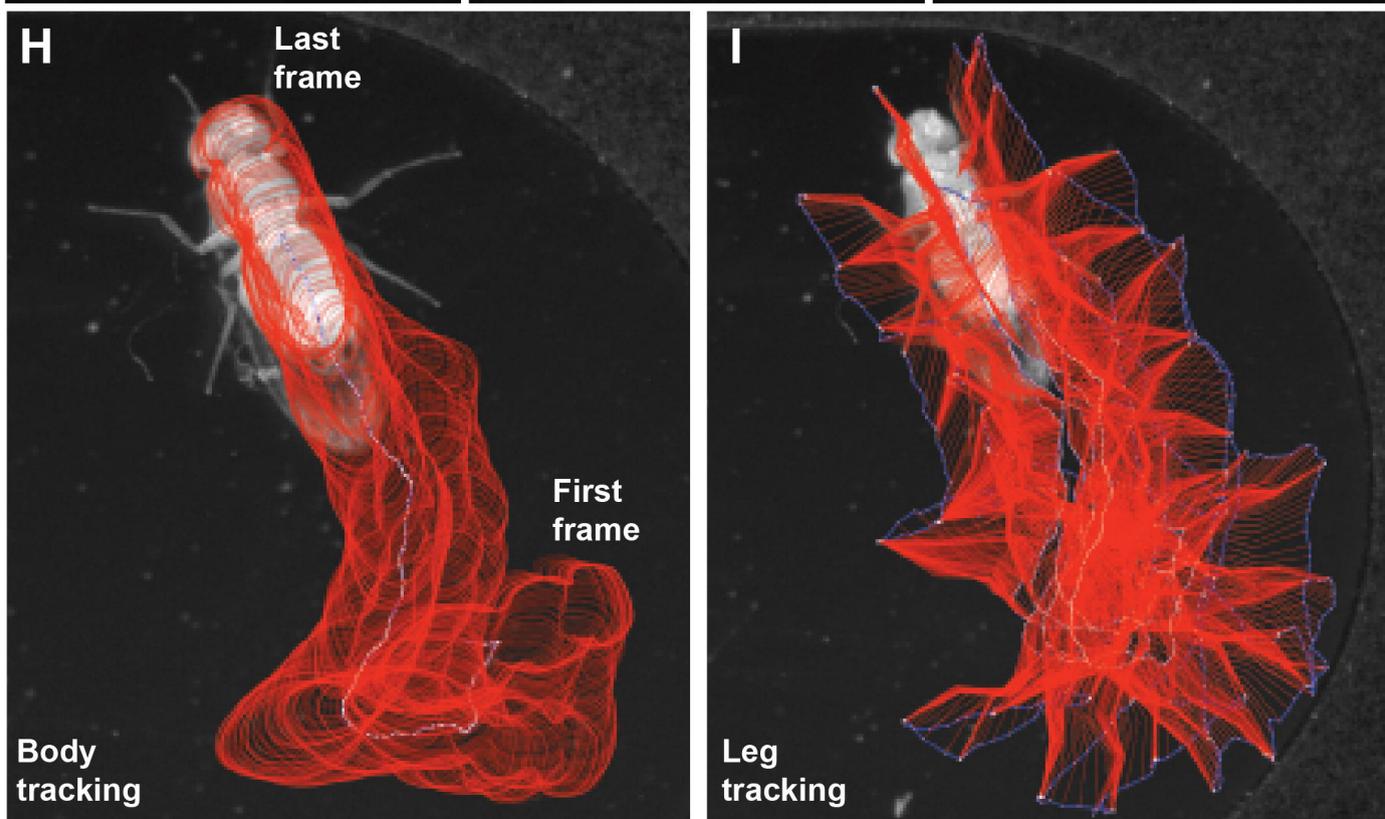
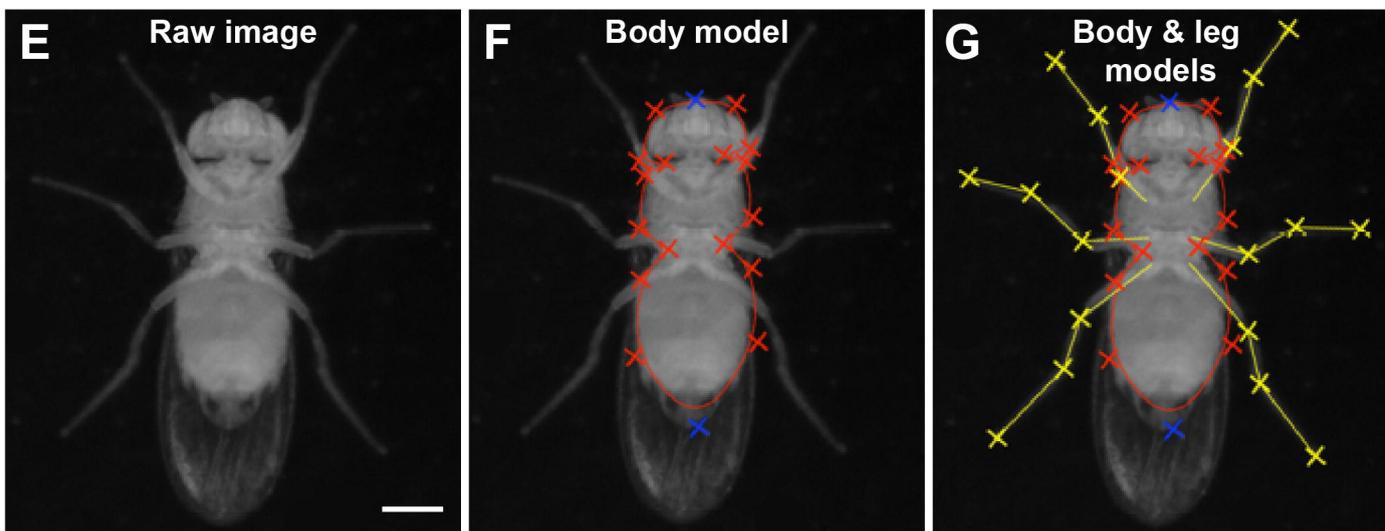
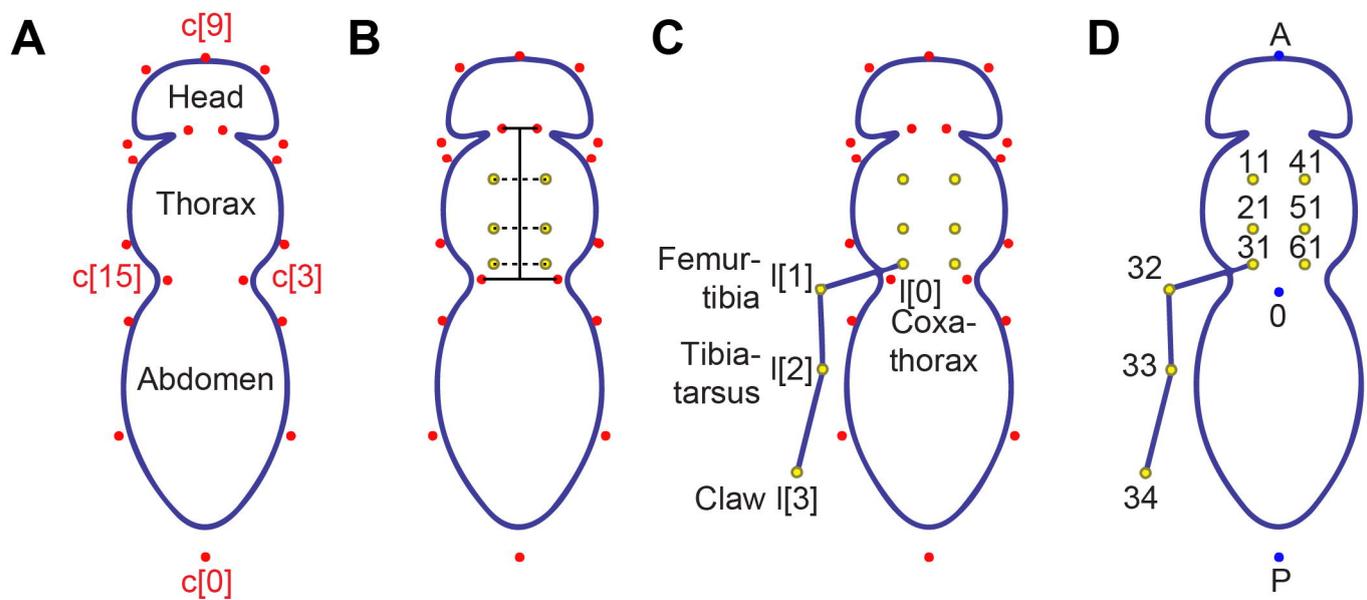
787 • Snake action: automatically optimizes the snake at its current
788 position (left button), or deletes it (right button). Note that both
789 actions are applied to the body snake and all leg snakes
790 simultaneously. If annotation methods for body or leg snakes are
791 set to *manual*, the corresponding snakes are left unmodified.

792 • Tracker action: performs backward (left button) or forward (center-
793 left button) tracking, interrupts tracking (center-right button), or
794 extracts/displays tracks (right button) using Icy's Track Manager
795 plugin (Publication Id: ICY-N9W5B7). The tracking algorithm is
796 implemented to allow backward and forward tracking, giving the
797 user flexibility to initialize tracking at any frame of the image
798 sequence. If any of the body or leg snakes are set to manual
799 annotation, the forward and backward tracking buttons will only
800 propagate current annotations to the next or previous frame,
801 respectively. If all snakes are set to automated annotation, tracking
802 will be performed in the selected direction until the end/beginning of

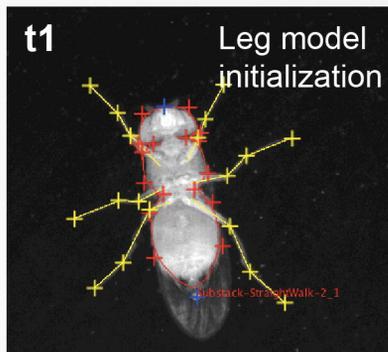
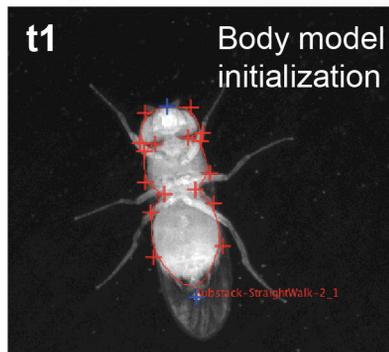
803 the image sequence is reached, unless it is manually halted using

804 the tracking interruption button.

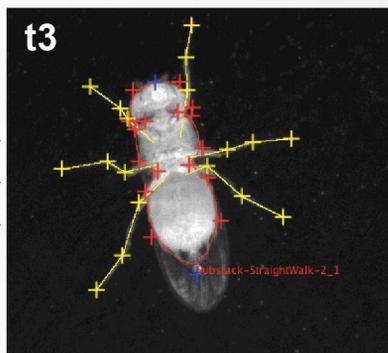
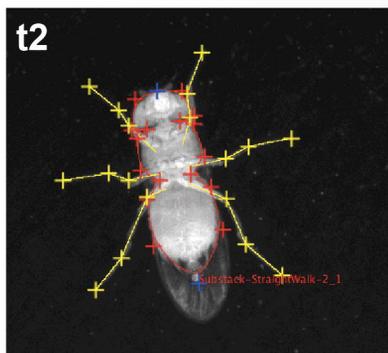
805



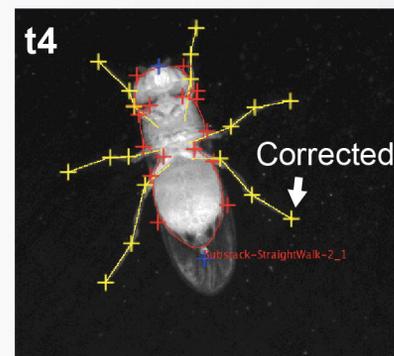
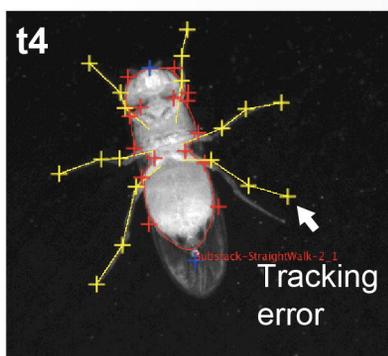
A Step 1: Manual initialization



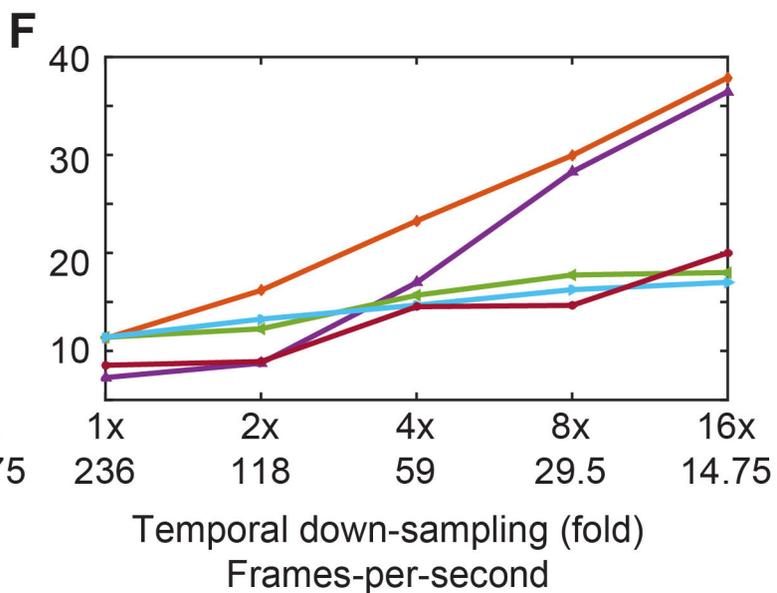
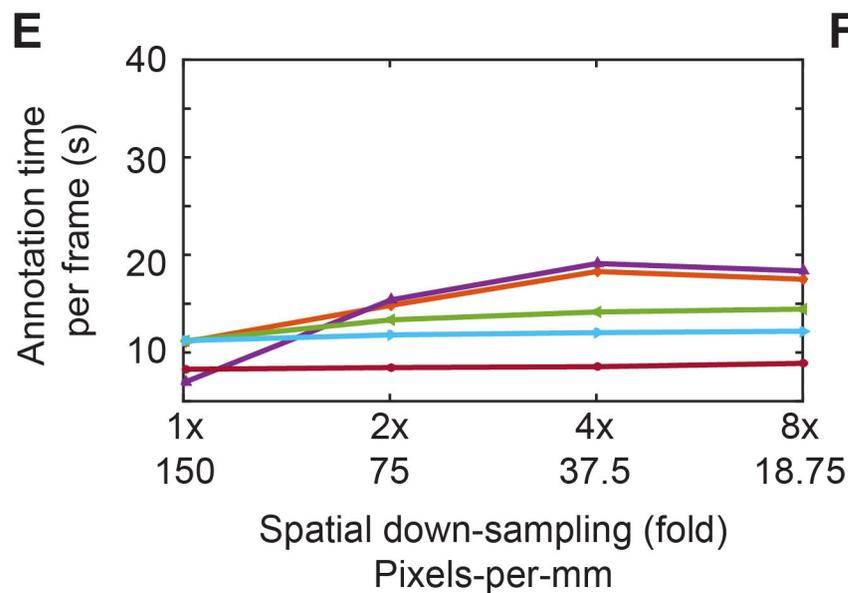
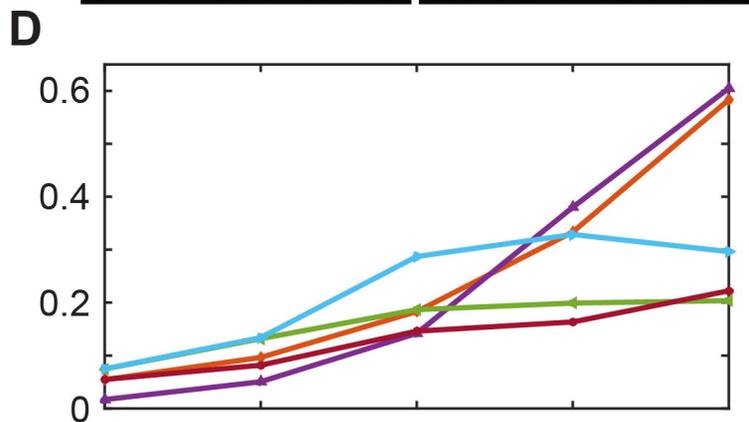
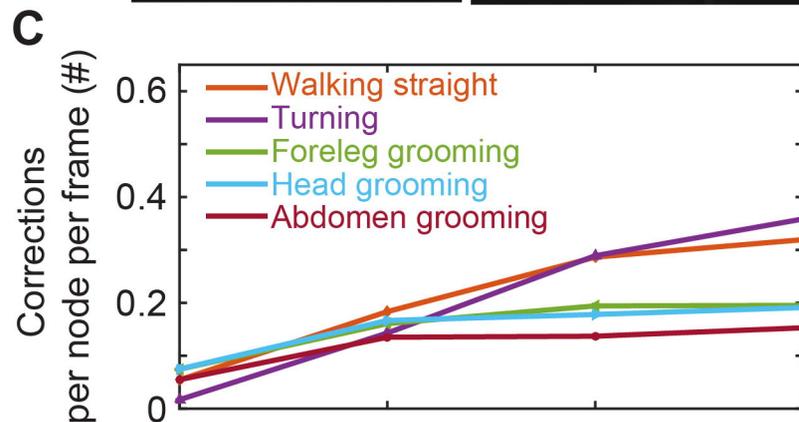
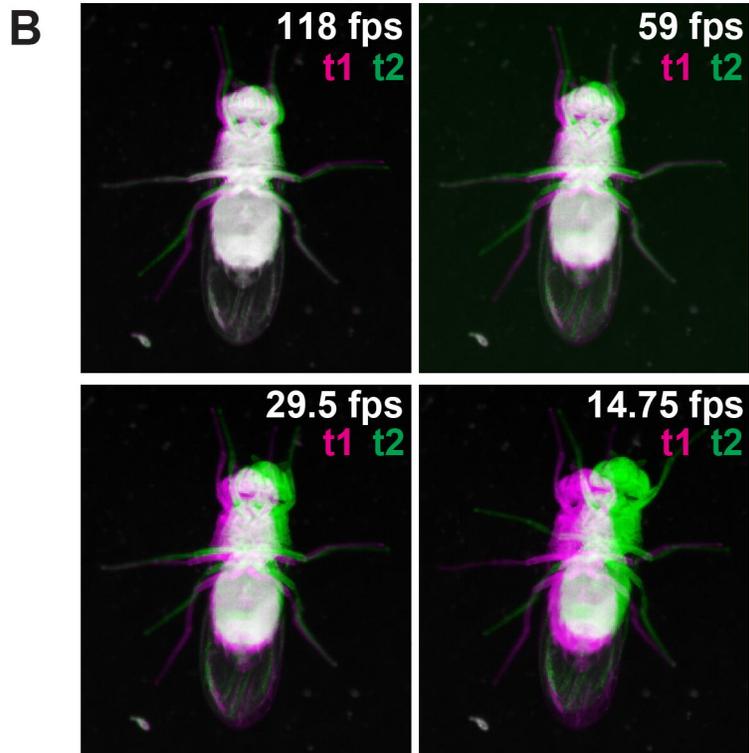
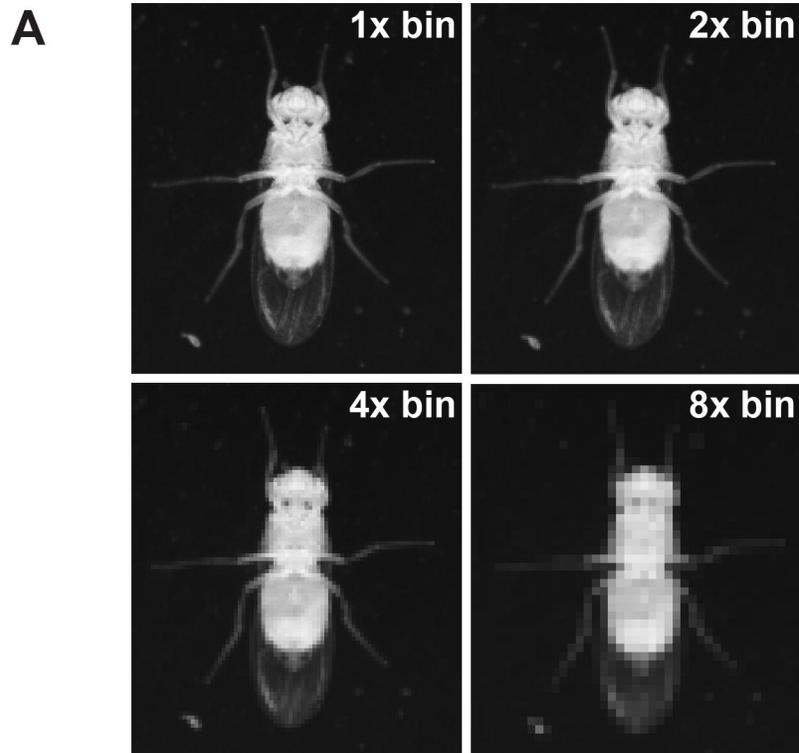
B Step 2: Automated tracking

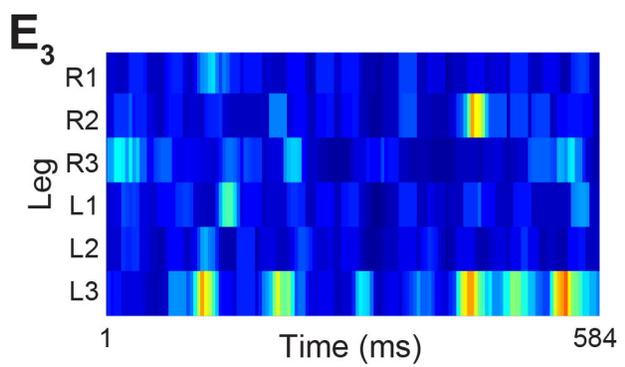
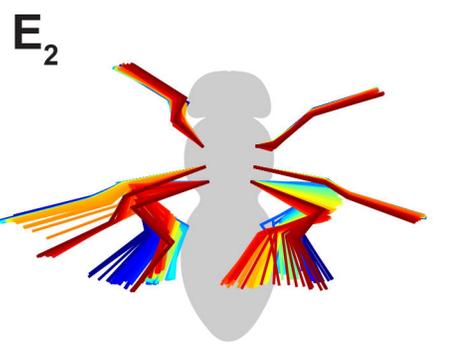
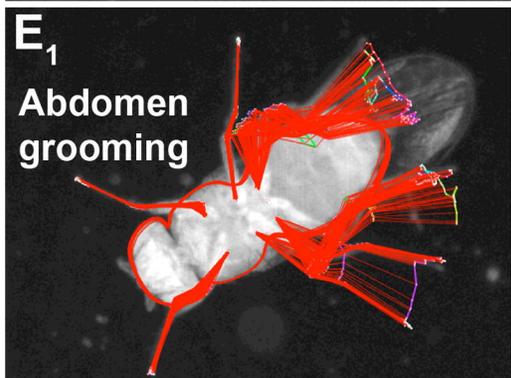
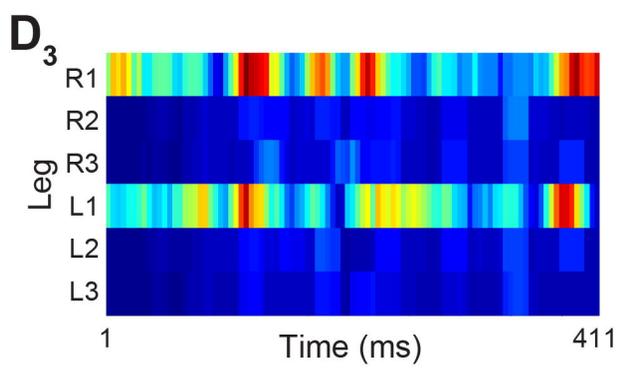
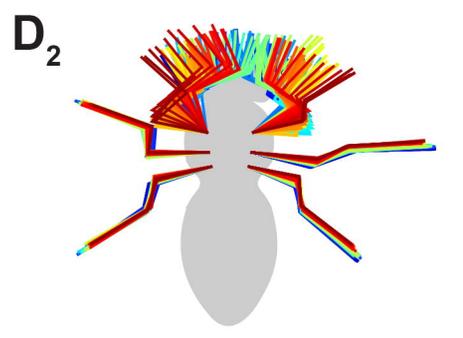
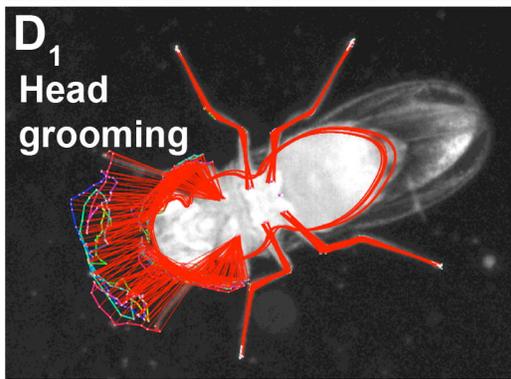
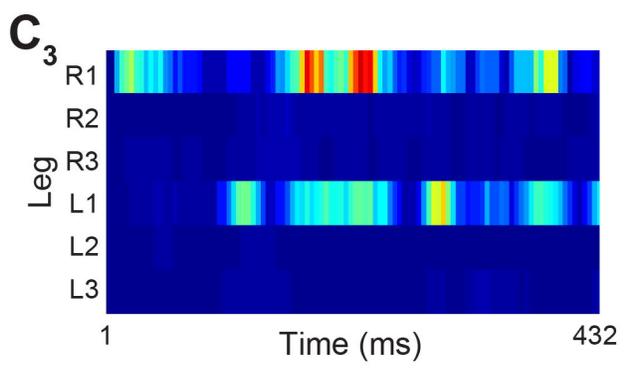
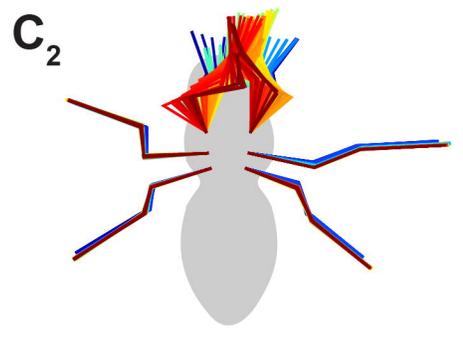
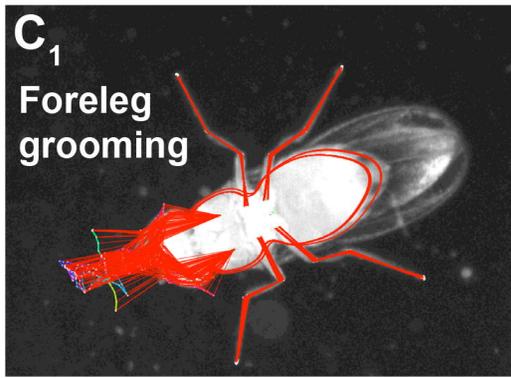
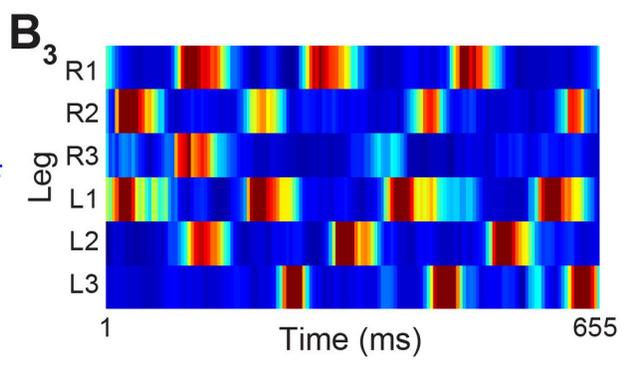
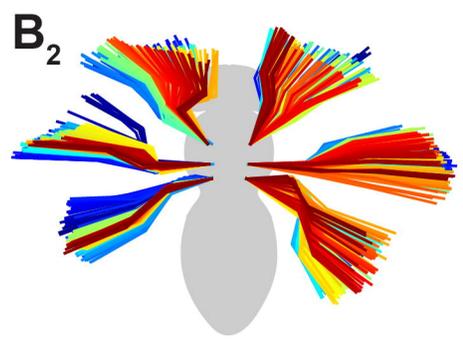
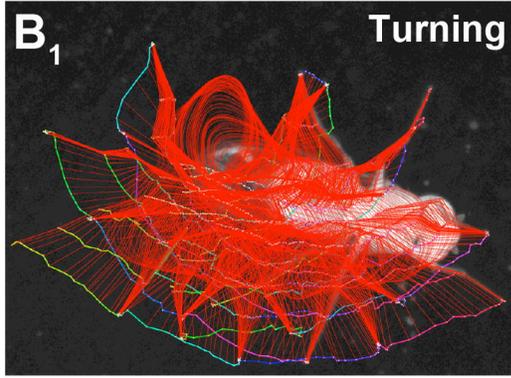
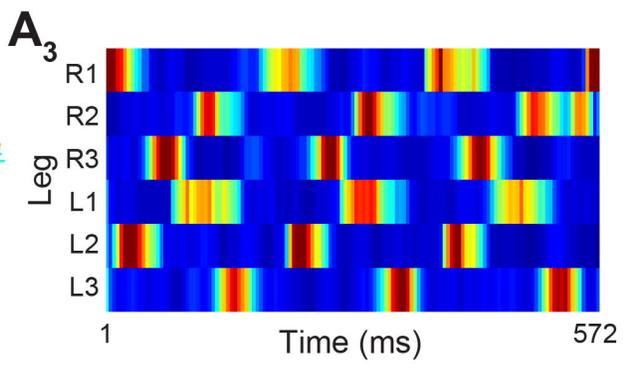
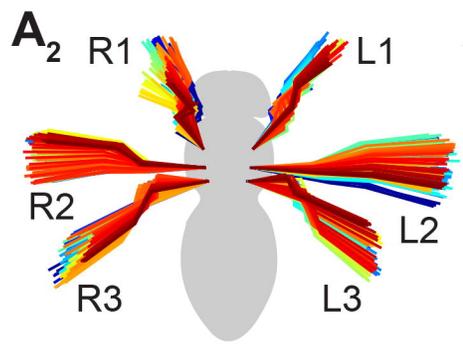
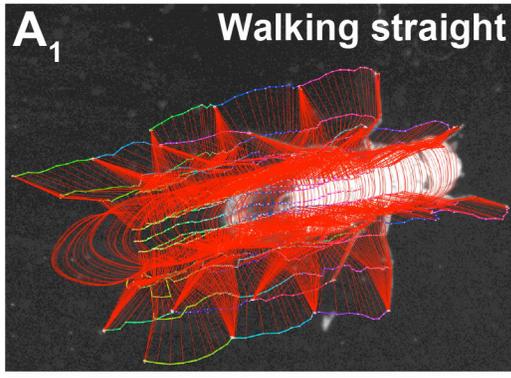


C Step 3: Manual error correction



Step 4:
Automated
tracking from
t5 onwards





Start End
Image sequence

0 66
Instantaneous claw speed (cm/s)