Readme for FLLIT

Contents

1 Overview of the FLLIT Program

The Feature Learning-based LImb segmentation and Tracking (FLLIT) program is compiled on MATLAB R2016a and runs on a Linux OS (eg. Ubuntu).

The program processes input data consisting of 512 pixels $\times$ 512 pixels video image sequences of the sample animal, e.g. *Drosophila* fruitfly or spiders. Currently, input supports TIFF and PNG files. The video should be in made in a single channel (grayscale), and taken at 250 frames per second or higher speeds. Currently, TIFF and PNG files are supported. The field of view should be held steady without movement throughout the video. The task is to:

1. Identify (at a pixel level) the legs of the sample animal via the 'Segmentation' module;
2. Track the tip positions of the legs via the 'Tracking' module;
3. Report tracking results consisting of:
   - Centroid position
   - Angles of rotation (relative to the y-axis)
   - Leg trajectory in arena-centered frame of reference
   - Leg trajectory in body-centred frame of reference

   These results can be further processed with the 'Data Processing' module and visualized via the 'Make Video' module.

1.1 Set up of FLLIT

As illustrated in Figure 1.1, the FLLIT program directory consists of the following folders and executables (as well as readme and license files):

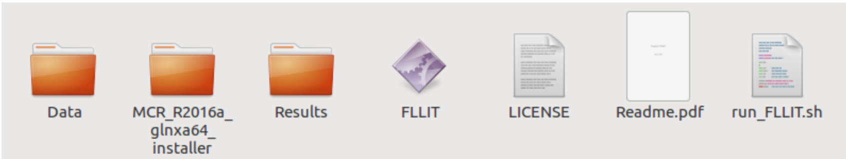


Figure 1.1: Folder Structure and Executables

- MCR_R2016a_glnxa64_installer folder: Download the MATLAB runtime libraries and extract into this folder.
- Data folder: All image datasets to be analyzed are placed in the 'Data' folder.
- Results folder: After analysis with FLLIT, all results can be retrieved from the 'Results' folder.
- FLLIT and run_FLLIT.sh: These are the executables for the FLLIT program.

1.1.1 Installation of required MATLAB runtime libraries

Download MATLAB runtime libraries and extract to the MCR_R2016a_glnxa64_installer folder. To initiate installation, open a terminal in the FLLIT directory and issue the following command:

```
sudo bash ./MCR_R2016a_glnxa64_installer/install
```

By default the library will be installed to the following location:

```
/usr/local/MATLAB/MATLAB_Runtime/v901
```

Please take note of the installation directory as this will be needed for executing the program. Typical installation time on a desktop computer takes about 10 minutes.

1.2 Executing the program

To run the FLLIT program, first open a terminal in the FLLIT directory. To ensure that executable rights are accorded to the application, the following commands should be issued:

```
chmod +x run_FLLIT.sh FLLIT
```

The FLLIT program can then be executed with the following command:

```
bash run_FLLIT.sh <deployedMCRroot>
```

Here, the <deployedMCRroot> is to be replaced by the root location where the MATLAB runtime libraries have been installed. For example, as described in section 1.1.1, the default location of the <deployedMCRroot> in Ubuntu is

```
/usr/local/MATLAB/MATLAB_Runtime/v901
```

The interface of the FLLIT program is shown in Figure 1.2. To begin, the user selects the image folder containing the image frames of the source video.

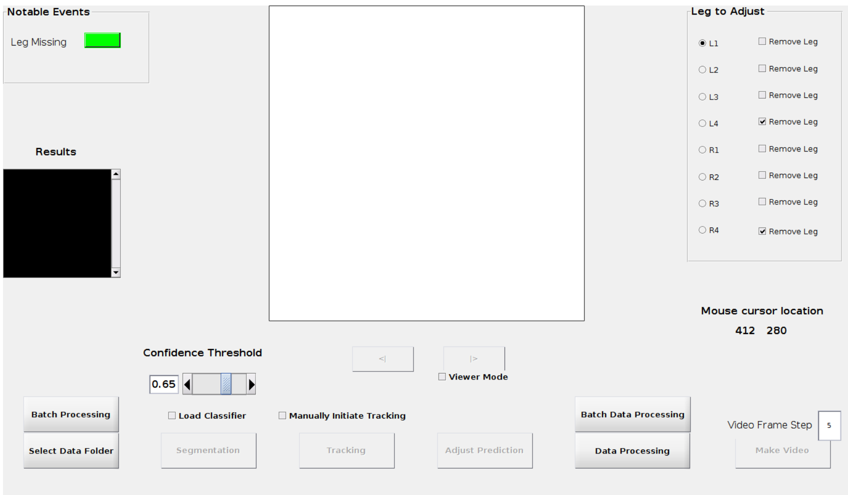


Figure 1.2: Graphic user interface of the FLLIT program

1.3 Program Workflow

The workflow pipeline is depicted in Figure 1.3.

Segmentation is first run to identify legs. Tracking is then run to determine the body centroid trajectory, rotation angle and individual trajectories of each leg. Subsequently, the tracking results can be retrieved and visualized in the form of a short movie, and analysed using the Data Processing functions.

The entire workflow is automated. Expected run time for a demo dataset of 1000 video frames on a typical desktop computer is about 70 minutes. Multiple videos can be processed at a time by selecting the Batch processing option.

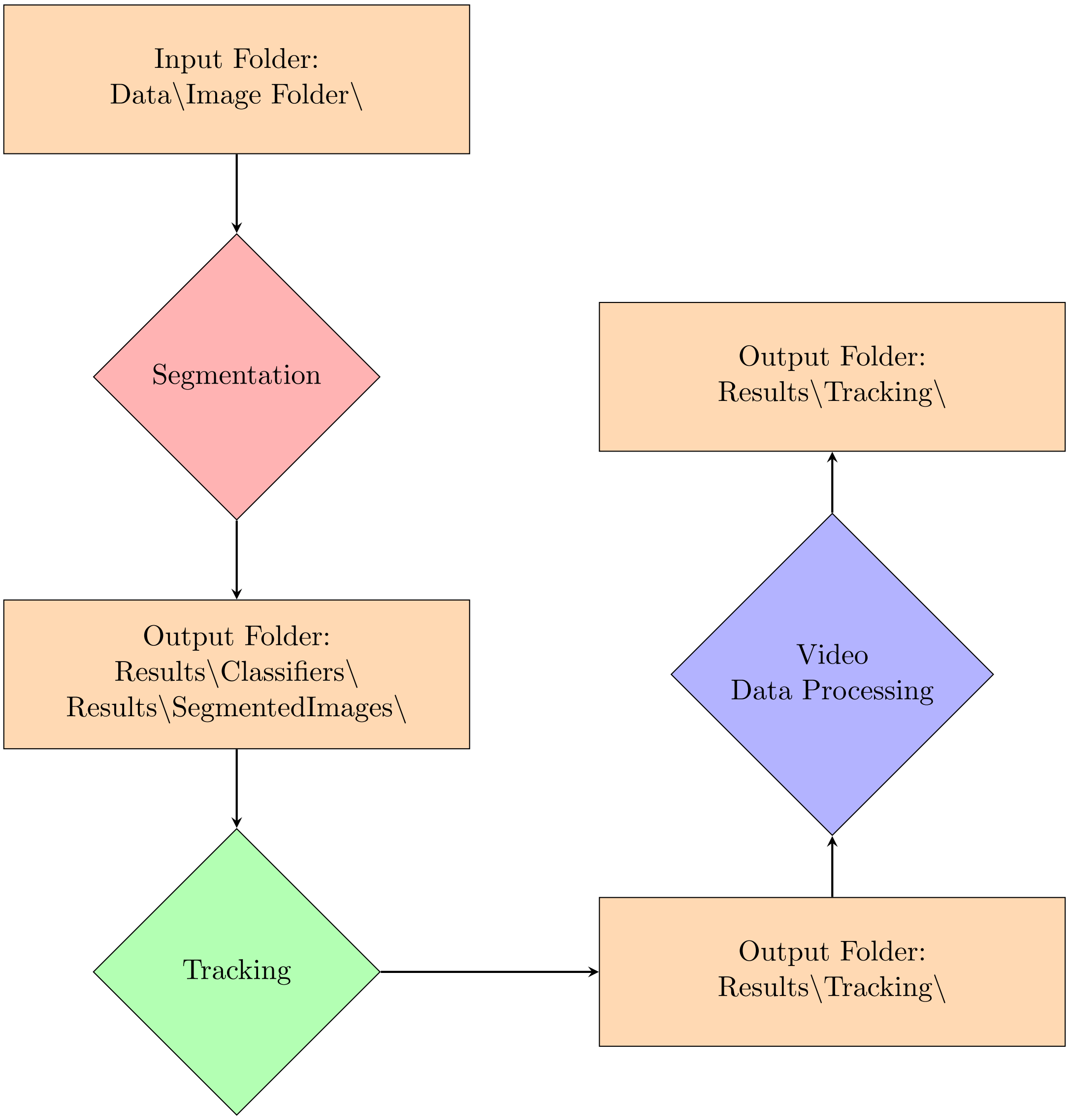


Figure 1.3: Flow chart of program

1.3.1 Segmentation

Segmentation is first carried out to identify the legs of the sample animal and is the most time-consuming portion of the analysis procedure. As illustrated in Figure 1.4, when the ‘Segmentation’ button is clicked, the program will proceed to extract the sample silhouette by automated background subtraction, and also generate training samples on a subset of the images. These training samples are used to train a classifier. After a classifier is learned, it is applied to novel images. The classifier assigns a confidence score to each pixel, ranging from 0 (least likely to belong to a leg) to 1 (most likely to belong to a leg). The default confidence score threshold is set at 0.65 and this can be adjusted by moving the slider bar or typing the desired threshold into the box.

The user also has the option to load an existing background image (which would be beneficial for videos where the sample moves slowly or not at all). This can be done by creating a background directory folder in the data folder and placing the background image (.png format Figure 1.5a) inside. There is also the option to load an existing classifier.

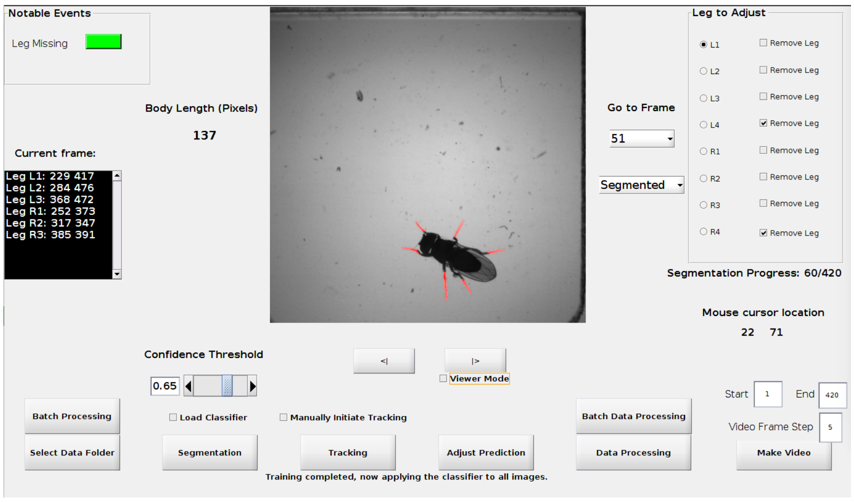
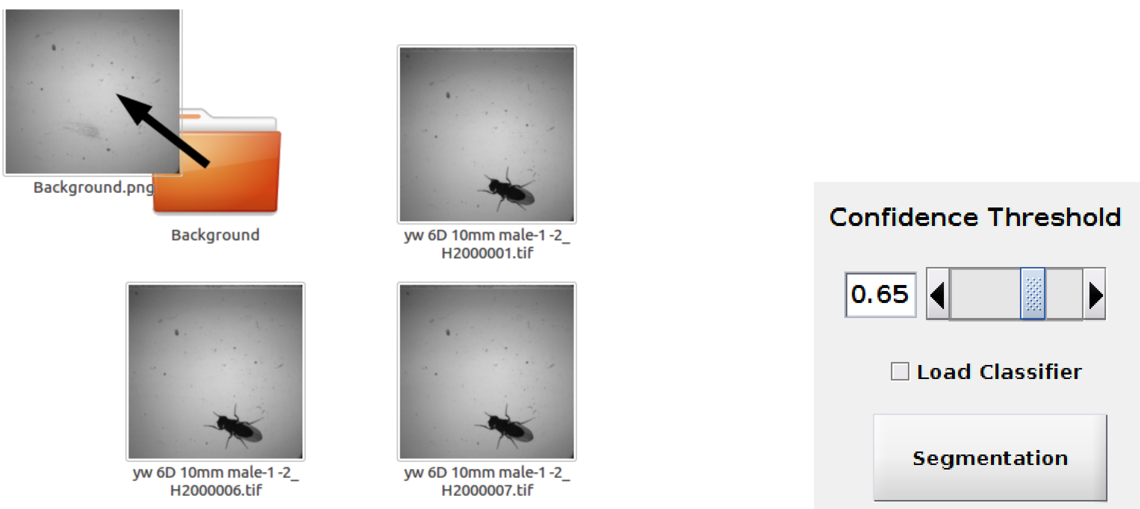


Figure 1.4: An example of a segmented image



(a) Using a prepared background image (b) Load classifier

Figure 1.5: User options for segmentation

1.3.2 Tracking

Tracking is initiated by clicking on the tracking button after segmentation has been completed (Figure 1.6). Tracking will automatically initiate on the first frame where all legs can be confidently assigned and the legs will be assigned the labels L1−R3, corresponding to left fore leg, left middle leg, left hind leg and right fore leg, right middle leg and right hind leg respectively. The leg tips are circled out and their corresponding pixel coordinates are shown on the text display box to the left of the user interface.

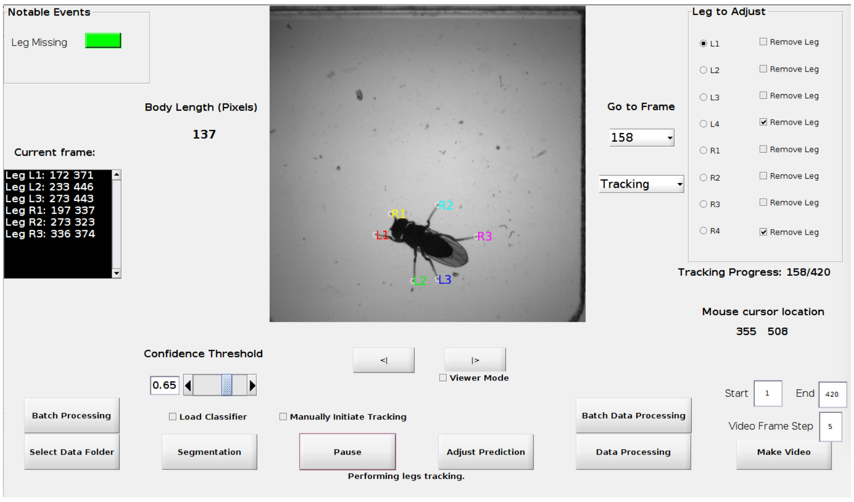


Figure 1.6: Tracking

Once tracking has started, the body length of the animal (anterior-most position on head to posterior-most position on wings) will also be automatically shown. During tracking, the green button next to the Legs Missing indicator will turn red if a leg is not identified. To correct an error, click 'Pause' followed by 'Adjust Prediction'. The user can then select a leg from the 'Select Leg for Labelling' panel and double click on the image to annotate or correct the tip position. The tip position coordinates of a leg can also be directly changed from the text display box on the left. Finally, the user can save corrections or discard changes to exit the 'Adjust Prediction' mode.

Navigating to the previous and subsequent frames can be done by clicking on the $<\|$ and $\|>$ buttons or via the 'left' and 'right' keys from the keyboard. The next frame $\|>$ button will perform the tracking algorithm on the subsequent frame. The 'Viewer Mode' box can be checked if the user wishes to browse through existing tracking data without running the tracking module. There is also a dropdown bar which allows selection of the frame of interest.

There is also the option to remove or include more legs during the tracking process. For example, decreasing the number of tracked legs allows analysis of datasets involving amputated flies, and inclusion of up to 8 legs allows for tracking of spider legs.

In some scenarios, the user may wish to manually initiate tracking from a particular frame, eg. as required for spider leg tracking. This can be done by checking the 'Manually Initiate Tracking' box. Clicking on the 'Tracking' button will turn the button text to 'Initial'. From here, the user needs to activate the 'Adjust Prediction' mode and label all the legs accordingly. Clicking on the tracking button again will then allow tracking to proceed normally from the user-initiated frame.

1.3.3 Video

Once tracking is complete, the tracking results are automatically saved to the Results folder as .csv and .mat files. Tracking results can also visualized with the 'Make Video' function (Figure 1.7).

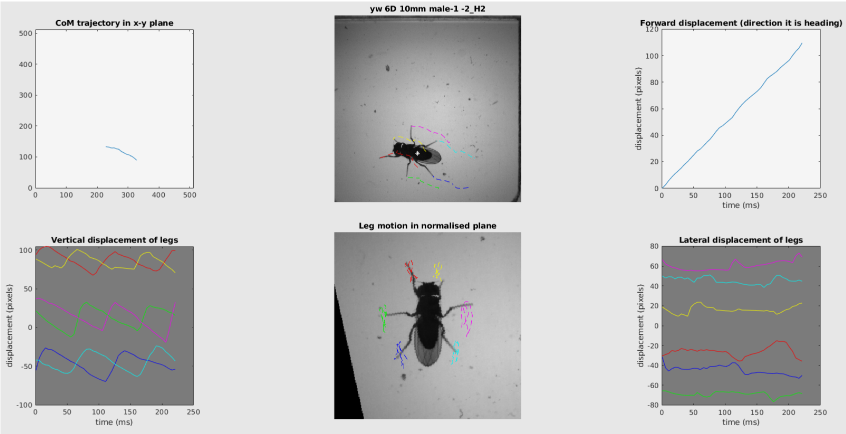


Figure 1.7: Sample video

1.3.4 Data Processing

The raw tracking results can be processed on clicking the 'Data Processing' button. The user can input the Frame Skip per Second (FPS) and image size (mm) of the dataset. The automatic data analysis will output the following parameters and plots:

Body motion

1. Body length: Length of the sample animal estimated in each frame (anterior-most position on head to posterior-most position on wings).
   Output: Data saved to bodylength.csv.

2. Body velocity: instantaneous velocity of the body (centroid) in the sample animal.
   Output: data saved to BodyVelocity.csv, plot saved as BodyVelocity.pdf.

3. Turning points of the body trajectory: to locate the turning points, the trajectory is reduced to a piecewise-linear curve using the DouglasPeucker algorithm, following which a turning event is identified as involving an angle $> 50\,\mathrm{deg}$ between two neighbouring linear segments constituting the simplified trajectory.
   Output: plot saved as BodyTrajectory.pdf.

Individual stride parameters

A stride event is identified as the movement of a leg with respect to the arena, lasting a minimum of 15ms.

1. Stride duration: the duration of a stride event.

2. Stride period: the duration from one stride event to the next.

3. Stride displacement: displacement of the leg tip during a stride event.

4. Stride path covered: total path covered by the leg tip during a stride event.

5. Anterior extreme position [1]: landing position (in the body-centred frame of reference) of a leg tip at the end of a stride event.

6. Posterior extreme position [1]: take-off position (in the body-centred frame of reference) of a leg tip at the start of a stride event.

7. Stride amplitude: displacement along the direction of motion for a stride event.

8. Stance linearity [1]: defined as the deviation of the stride path from a curve smoothed over (at 20ms intervals) the corresponding anterior and posterior extreme positions of the stride.

9. Stride stretch: distance of the leg tip position from the body centre in the middle of a stride event.

Output: All these parameters are saved to StrideParameters.csv.

Leg motion

1. Leg speed: the instantaneous speed of each leg with respect to the arena is obtained.
   Output: data saved to LegSpeed.csv, plot saved as Gait.pdf.

2. Gait Index [1]: this measures the type of gait coordination exhibited by the (six-legged) sample animal during its motion. A gait index of 1 corresponds to a tripod gait, -1 corresponds to a tetrapod gait while 0 constitutes an non-canonical gait. In our implementation, the gait index is obtained by a moving average over a 120 ms window.
Output: data saved to GaitIndex.csv, plot saved as GaitIndex.pdf.

3. Movement percentage: percentage of the time that a leg is in motion.
Output: data saved to LegParameters.csv.

4. Mean stride period.
Output: data saved to LegParameters.csv.

5. Total path covered: this measures the total trajectory covered by each leg in the entire video sequence.
Output: data saved to LegParameters.csv.

6. Mean AEP/PEP: the average posterior and anterior extreme positions of a leg.
Output: data saved to LegParameters.csv.

7. Footprint regularity: measured as the standard deviations of the posterior and anterior extreme positions of a leg.
Output: data saved to LegParameters.csv.

8. Leg trajectory domain area: defined as the area of the minimal convex hull that contains the entire leg trajectory in the body-centred frame of reference.
Output: data saved to LegParameters.csv, plot saved as LegDomain.pdf.

9. Leg trajectory domain area / path covered: the ratio of the domain area to path covered for a leg. This gives an intuitive measure of leg spread in its range of movement.
Output: data saved to LegParameters.csv.

10. Length and width of the leg trajectory domain: obtained via the intersection of the principal component vectors with the convex hull.
Output: data saved to LegParameters.csv.

11. Footprint deviation with respect to the trajectory principal component: measures the angle of deviation of the vector connecting the mean anterior and posterior extreme positions from the primary trajectory principal component vector. An example of how the 2 vectors can significantly deviate from each other is illustrated in figure 1.8.
Output: data saved to LegDomain.csv.

12. Foot-dragging: a foot-dragging event is identified to be occurring when a leg is moving with respect to the arena yet is relatively stationary with respect to the body (lasting for at least 15ms, similar to the threshold duration for a stride event). In the case of such an event, the leg and frames involved are recorded.
Output: data saved to FootDragging.csv.

13. Middle legs spread: measures the distance between the tips of the middle legs. This is evaluated in terms of two metrics: 1) the shortest path covered from tip to tip through the middle legs and body; 2) the horizontal spread distance.
Output: data saved to MiddleLegsSpread.csv.

14. Leg domain overlap: the overlap area across any two leg domains. The leg domains usually have no overlap in wildtype flies, whereas in the case of mutants, there might be significant overlaps due to different walking patterns.
Output: data saved to LegDomainOverlaps.csv.

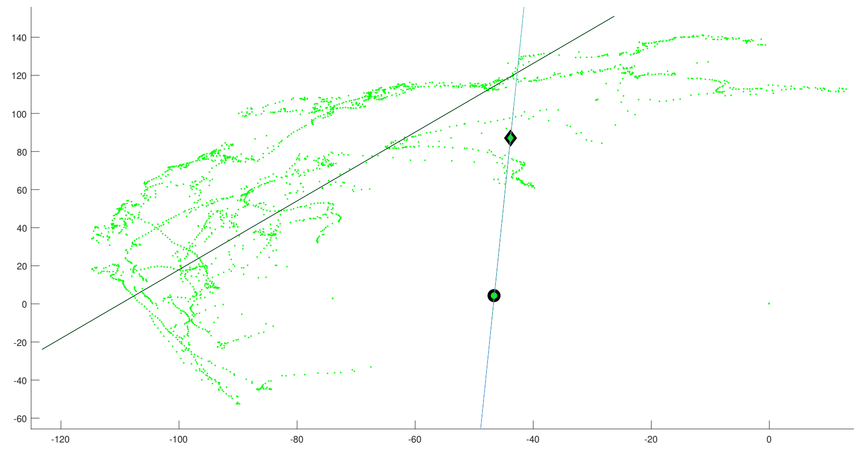
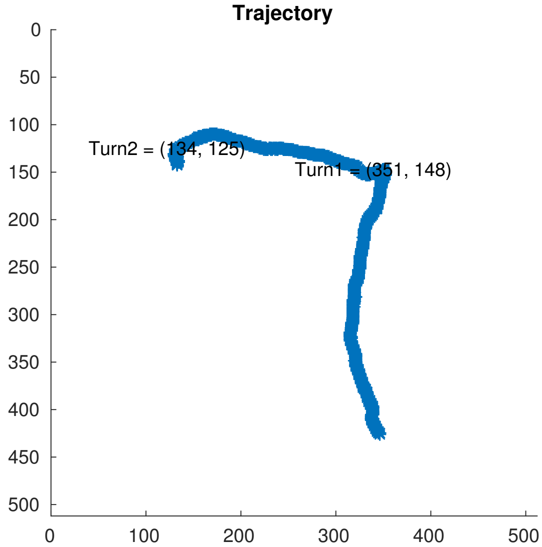
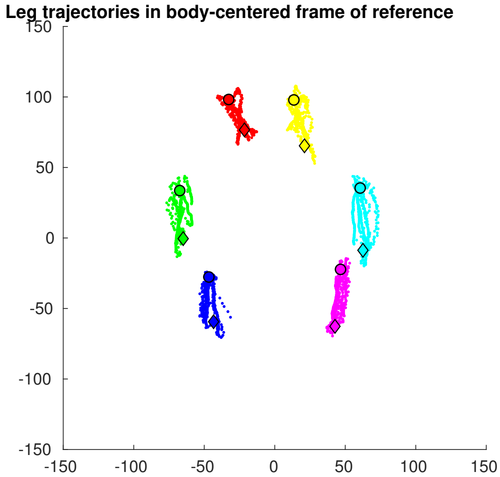


Figure 1.8: An example of a leg trajectory in a drosophila with spinocerebellar ataxia demonstrating significant offsets between the trajectory and the footprint positions. The black line is the primary principal axis of the trajectory while the blue line connects the mean anterior/posterior extreme positions (indicated with the circle/diamond markers respectively). In this particular example, the angular deviation between the two vectors is 153 deg.

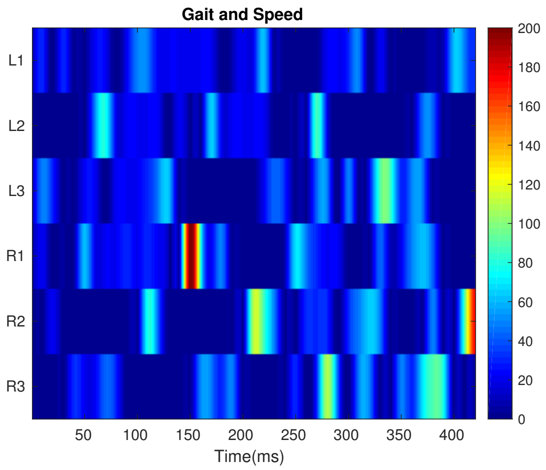
Sample plots are shown in figure 1.9.



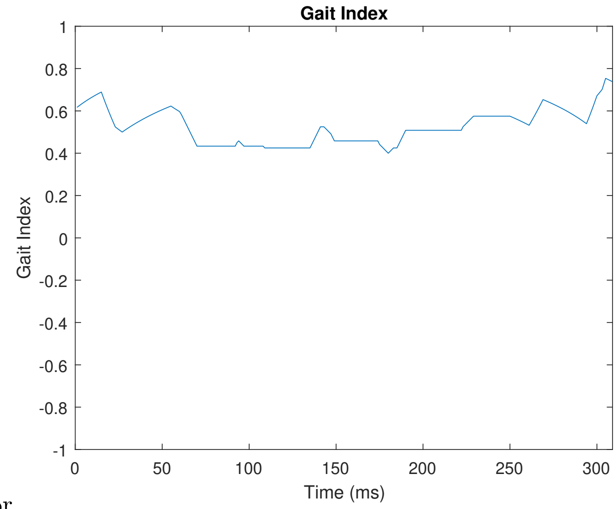
(a) Trajectory with turning points marked out. The units here are pixels.



(b) Leg trajectory domains. The circle/diamond marker denotes the mean landing/take-off position.



(c) Gait plot for a sample drosophila dataset. Color intensities denote leg speeds in units of mm/s.



(d) Gait index plot

Figure 1.9: Sample output

Bibliography

[1] César S Mendes, Imre Bartos, Turgay Akay, Szabolcs Márka, and Richard S Mann. Quantification of gait parameters in freely walking wild type and sensory deprived drosophila melanogaster. *elife*, 2:e00231, 2013.