# Eel Tracker

## User Manual for version 1.0

James Campbell

April 6, 2017

# Contents

# Chapter 1

# Keyboard Shortcuts for Manual Corrections

| | |
|---|---|
| Z, X | Back, forward tracked frame |
| A | Redefine head and tail point (midline will be automatically detected again) |
| S | Manually define midline points (Press enter to set, in between points will be interpolated) |
| O | Save corrections in a new file |

# Chapter 2

# User Guide

## 2.1 Quick Start

The following program has three parts:

1. Video Analysis
2. Video Post-Processing
3. Tail beat analysis

Here we'll provide a walk through for all three sections which will allow you to analyze tail beat frequencies from videos in a semi-automated process. The first two parts are done in matlab, resulting in a `csv` file holding the coordinates of the eel's midline relative to the swimming vector. The final tail beat analysis will be done in R.

### 2.1.1 MATLAB Compiler Runtime

The easiest way to deploy the MATLAB parts of the analysis is by using MATLAB Compiler Runtime (MCR). This allows you to run compiled MATLAB scripts without holding a paid license. You can download MCR here: https://www.mathworks.com/products/compiler/mcr.html. This binary was compiled with MATLAB 2013a (32-bit, version 8.1, windows), so you will need to download that same version of MCR. Note that you must download the same architecture type that the binary was compiled on, so even if you are using a 64-bit computer, you still need to download the 32-bit version of the program to run this. Also note that the MCR installation is invisible. Once complete, there are no items added to your start menu and the MCR will automatically open when you double click the compiled binary.

### 2.1.2 Get R packages

For the tail-beat analysis, we've saved the code you need in two packages hosted on github under the user RTbecard. These packages are not on the CRAN, but you can find instructions for installation of the page of each repository.

1. ezPSD (https://github.com/RTbecard/LUAnimBehav)
   This will be used for the frequency analysis

2. LUAnimBehav (https://github.com/RTbecard/ezPSD)
   This is a general purpose package for animal behavior where we've stored the functions for automating the tail beat analysis.

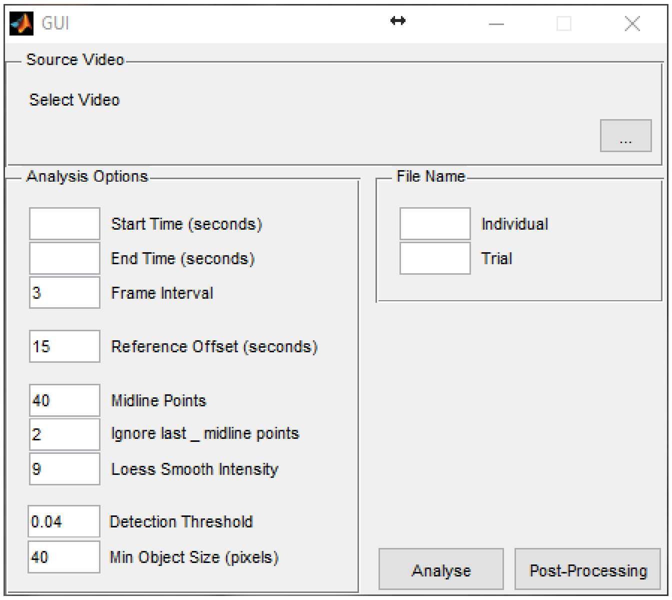
### 2.1.3 Get MATLAB code/binary

You can get the code for the MATLAB program here: https://github.com/RTbecard/EelTracker. If you want to use a full version of matlab, you start the program by running the script GUI.m. If you are interested in the MCR version, you only need to download the exe file located in the folder CompiledBinaries.

### 2.1.4 Video Analysis

Here we'll walk through the video analysis.

1. Start by starting the MATLAB program (double-click the exe if you are using MCR).



Start Time (seconds)

End Time (seconds)

3 Frame Interval

15 Reference Offset (seconds)

40 Midline Points

2 Ignore last _ midline points

9 Loess Smooth Intensity

0.04 Detection Threshold

40 Min Object Size (pixels)

Individual

Trial

2. When open you will see a set of options for analysis. While most are self-explanatory, we will quickly explain some of the options here:

    - Frame Interval: You can use this to process only every nth frame of the video.
    - Reference Offset: This is used for automatically defining the reference images for the background subtraction which is used for detection movement in the video. To determine what value this should be, watch your video and move to the starting frame of your analysis. Now, draw a box around the eel's location. Next play the video and time how long it takes for the eel to completely leave this bounding box you drew. Setting this value too low will result in detection errors, as the eel tracker uses frames n seconds before and after the current frame to create a reference image for motion detection.
    - Midline Points: The number of points to automatically draw along the midline. Note these points are equally spaced along the major axis of the eel.
    - Ignore last _ midline points: If you are analyzing a fish with a large soft tail, it can be useful to ignore the motion detected by the tail, as this will extend beyond the midline and cause problems for the spline estimation. Points will still be drawn for

   the ignored points, but they will be ignored when calculating a fitted spline for the midline.

   - Loess Smooth Intensity: This value determines the intensity of smoothing. Smoothing helps avoid errors caused by outlier points. If set too high, important features of the midline may be missed. If set too low, the midline may not resemble a smooth curve, and outlier points will have a greater effect o the estimated midline.
   - Detection Threshold: Higher values will make the motion analysis less sensitive.
   - Min Object Size: All detected objects with a pixel size smaller than this will be ignored by the analysis.

3. Select a video and press analyze to begin the auto midline detection. You will see an output like the image below where you have to mark the eel's location:

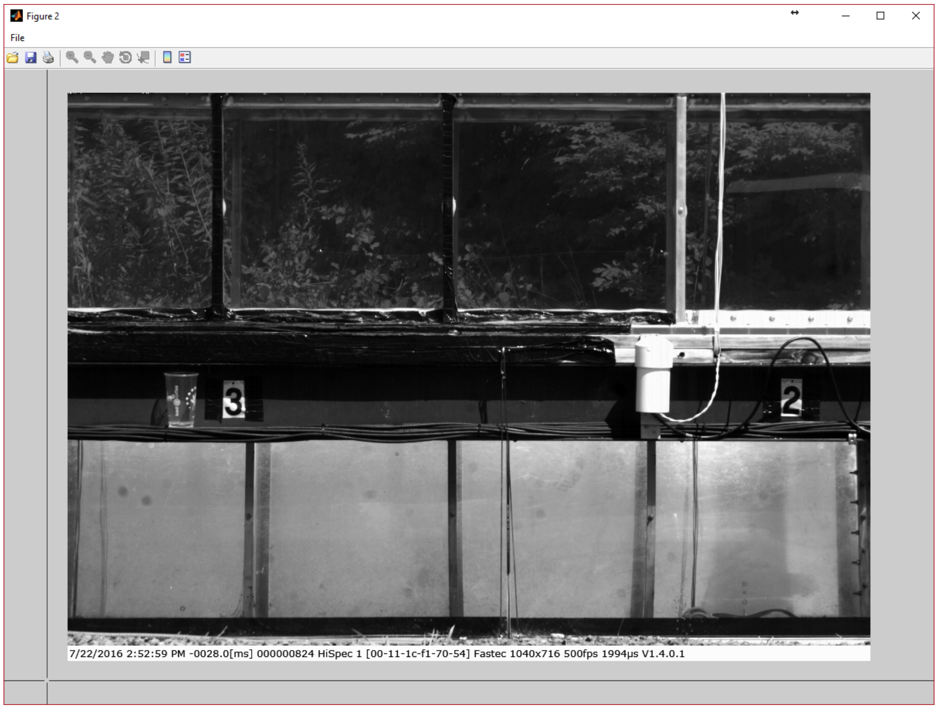


Figure 2
File
3
2
7/22/2016 2:52:59 PM -0028.0[ms] 000000824 HiSpec 1 [00-11-1c-f1-70-54] Fastec 1040x716 500fps 1994µs V1.4.0.1

4. You create a bounding box by clicking twice, first the upper left, the the upper right. This will be used as the tracking box. The tracking

box will continuously follow the eel through the video,and ignore all information outside that area. Make sure to make this box a little larger than the eel, so if the eel stretches, it won't end up exceeding the width of the box. Also, make sure the bounding box is always within the video frame. After you click twice, the tracking will begin.

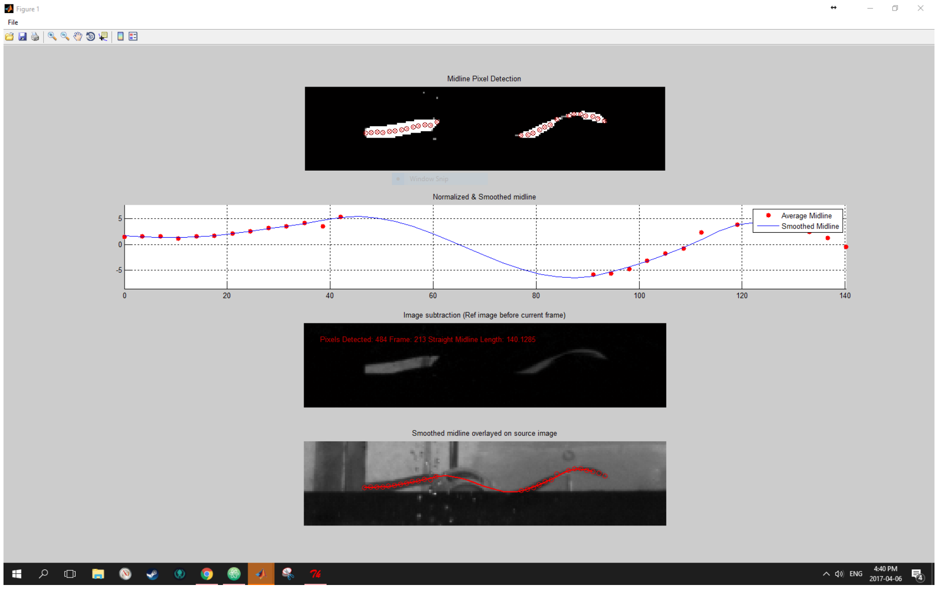


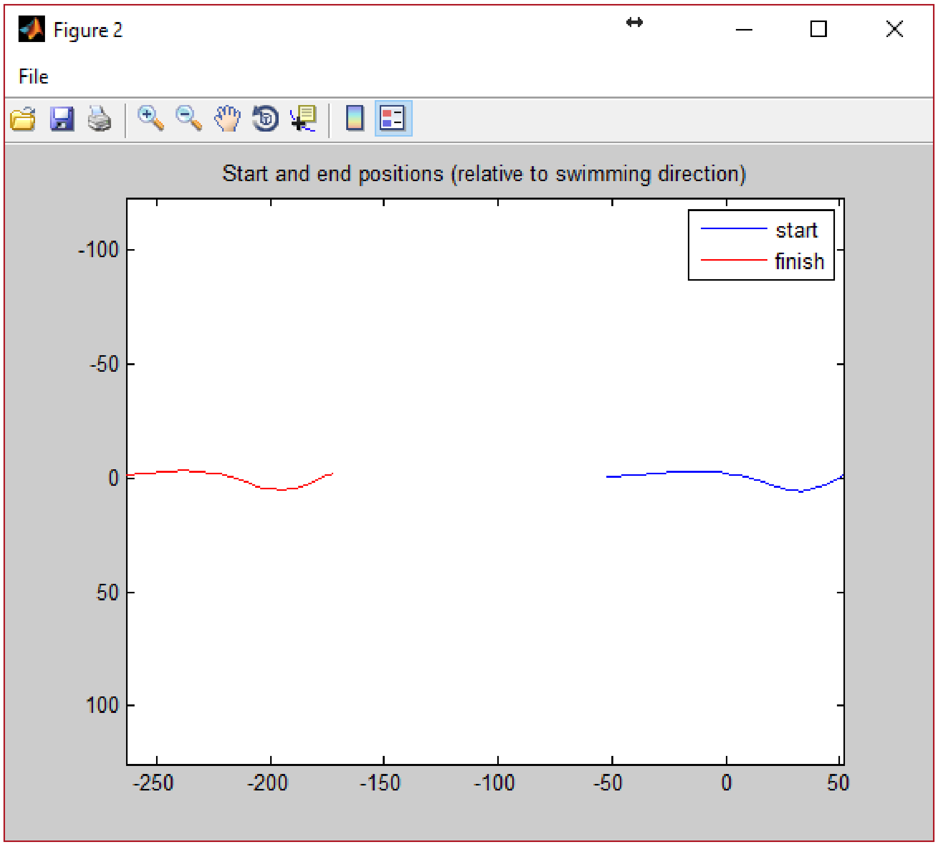
Figure 1
File
Midline Pixel Detection
Normalized & Smoothed midline
Average Midline
Smoothed Midline
5
0
-5
0
20
40
60
80
100
120
140
Image subtraction (Ref image before current frame)
Smoothed midline overlayed on source image
ENG
4:40 PM
2017-04-06

Pixels Detected: 484 Frame: 213 Straight Midline Length: 140.1285

5. When the tracking begins, you will see four plots which continuously track the eel. From top to bottom, the plots are described here:

   - Midline Point detection: Results from the object detection, showing detected midline points. White pixels are detected, grey are ignored detected pixels, and black is no detections.
   - This is a plot of the fitted midline curve. This curve should give a pretty good approximation of the midline. This plot is useful for getting feedback about the Loess Smooth Intensity parameter.
   - This is the result from the image subtraction. This can be used to see if you made any errors in setting your reference frame offset parameter. You should just see a gray shape approximating the detected eel here.
   - Plotted fitted midline. This is a final check to make sure that your midline correctly aligns with the original video.

6. When the analysis completes, you'll see the starting and ending positions of your detected eel, and your results will be saved as a `csv` file in the current folder.



### 2.1.5 Video Post Processing

If there are small obstructions in the video file or errors in the midline detection for a small number of frames, you can correct these by choosing the `Post Processing` option. You will be prompted to select the results file corresponding to the selected video, and you once again must define the bounding box of the eel. In this mode you can use `z` and `x` to jump to previous/next frames and you can use `a` to redefine the head and tail and `s` to manually correct segments of the midline.

When manually correcting midline segments, you will see green lines overlayed on the video frame. Every time you click, the point you marked will automatically be snapped to the closest green line until you finish by pressing enter. Any spaced between points you click will be automatically interpolated with a cubic spline, so you can define large areas of the midline with relatively few clicks. When you press enter, the points you marked will be used to redefine the midline once again. When you are finished press `o` and your corrections will be saved in a new `csv` file

### 2.1.6 Tail beat analysis

Here we'll give a brief explanation on how the analysis works, followed by an example analysis.

The R functions provided will do the following actions in processing your data from the MATLAB program:

1. **Reposition midline points**. In the MATLAB program, the midline points we're equally spaced along the major axis of the eel. While this is conveniant for tracking, it more usefull in the analysis if these points are equally spaced along the length of the eels midline. To do this, the R function will linearly interpolate 1000 points along the eels midline, based on the provided data. The function will then select the points from this high resolution interpolation resulting in a set number of points along the midline which are quite equally spaced.

2. **Convert to real time and spatial units**. Using the parameters specified, the units of frames and pixels will be converted to seconds and cm.

3. **PSD Analysis**. For each equally spaced midline point, a Power Spectral Density analysis will be done, resulting in a frequency and amplitude measure for each point on the eels body. Due to the small number of samples in the time series, we've opted for a multitapered spectrum using Slepian sequences. This is similar to a conventional Welches PSD, but instead of a series of partially overlapping window sequences with a constant window function, each of which covers a small portion of the time series, Thompsons multitapered approach creates a series of orthogonal window segments (Slepian sequences). Each slepian sequence covers the entire time series, and each of these windows has a unique window function which is orthogonal to the

rest. The benefit of this approach is that due to each window segment containing all the time series data, as opposed to just a small portion like in Welches method, a much higher frequency resolution is given for the results. To see instructions for downloading and running the package, follow the instructions on the following github repo: https://github.com/RTbecard/LUAnimBehav