

Whisker Tracking Command Line Reference

Contents

- [Concepts](#)
 - [Command line interface](#)
 - [Interchangeable parts](#)
 - [Tracing](#)
 - [Measurement](#)
 - [Classification](#)
- [Reference](#)
 - [trace](#)
 - [measure](#)
 - [classify](#)
 - [reclassify](#)
 - [whisker_convert](#)
 - [measurements_convert](#)
 - [report](#)

Concepts

Command line interface

The programs described below are run from your operating system's command line; they are **not** Matlab functions. Typically, each program reads a file as input and outputs the results to another file. These output files may be read by Python or Matlab functions provided by this package.

More information on the command line for your operating system may be found:

- [Windows command prompt FAQ](#)
- [OS X Terminal](#)
- [An introduction to the Linux terminal](#)

Interchangeable parts

There are four command line program to trace and track whiskers in a movie. Each of these programs represents a different step in the tracking procedure. By keeping these steps separated, rather than glomming everything into a single command, one can swap out any of these steps with a customized tool.

The most useful example of this is the classify step. This step uses a few heuristic rules to give a first guess as to whether a traced object is part of a whisker or not. These rules might change from experiment to experiment. By replacing the `classify` program with your own Matlab script, for example, one can customize this step but still take advantage of other parts of the [Whisker Tracking](#) package.

Tracing

Videos are automatically analyzed to estimate backbone positions (with subpixel precision) for segments of whiskers, facial hairs, and other dark line-line objects. We try to take an approach favoring **false positives**; it is better to trace too many things than to fail to detect the presence of a whisker.

Each frame of a video is treated independently. Subsequent steps will try to assign labels to traced segments identifying which segments belong to which whisker.

Measurement

Rather than compare the point-by-point representation of the backbone of a traced segment, a set of features descriptive of the shape are used for comparison.

Table 1: Features

feature	description
time	The index of the frame in the video on which the traced segment was found.
length	Length of the traced path in pixels.
tracing score	Tracing involves finding the optimal correlation between a parameterized filter and the image at a point. This optimal value is a score. This is the median of these scores evaluated at points along a traced segment.
angle at follicle	Degrees of deflection away from the animal's left-right axis.
mean curvature	The average curvature over the traced path in units of <code>1/pixel</code> .

follicle position	(x,y) point in pixels.
tip position	(x,y) point in pixels.

Classification

Two steps are used to identify which segments belong to which whiskers throughout a video uses two steps. A first guess is made that separates segments belonging to whiskers from other traced segments. This first guess is used to generate statistics based on measured features. These statistics are used in a second pass, together with the idea that whiskers maintain an order along the face, in order to refine the guess and assign specific identities for different imaged whiskers.

Reference

Note

These programs require [special files](#) to be present in the current working directory.

trace

Performs fully automated detection and tracing of all whiskers.

Usage

```
trace <video> <destination.whiskers>
```

Parameters

<video>

The path to the video file. The video must be one of the following formats:

- [StreamPix SEQ](#)
- 8-bit greyscale [TIFF](#)
- Any format decodable with [FFMPEG](#).

<destination.whiskers>

The path to the file to which results will be saved.

Example

```
trace path/to/data/movie.mp4 path/to/data/result.whiskers
```

measure

Measures several properties of the traced segments output from [#trace](#). Outputs a table with an entry for each traced segment and fields for the features listed in the [features table](#).

Usage

```
measure --help
measure --face <side> <source.whiskers> <destination.measurements>
measure --face <x> <y> <axis> <source.whiskers> <destination.measurements>
```

Parameters

-h, --help

Displays a help message.

```
--face <side>, --face <x> <y> <axis>
```

<side> may be left, right, top, or bottom.

Indicates the side of the image along which the face is oriented. The value effects how follicle and tip position are estimated.

<side> is used for videos recorded in such a way that the mouse face lies on one side of the image with the whiskers protruding into the field. Using <side> may not work well if the face is not confined to one side of the image or when the whisker pad is located near a corner of the image.

When <side> doesn't work well, use <x> and <y> should be the approximate coordinates of the center of the whisker pad. <axis> should be x or h if the face (the anterior-posterior axis) lies mostly in along the image's horizontal axis. Otherwise, <axis> should be y or v to indicate the vertical axis.

```
<source.whiskers>
```

The source file containing the traced whisker shapes.

```
<destination.measurements>
```

The destination file containing the table of computed features. This file will also be used to keep track of whisker identities in later steps.

Examples

```
measure --face left path/to/data/result.whiskers path/to/data/result.measurements
```

```
measure --face 30 420 x path/to/data/result.whiskers path/to/data/result.measurements
```

classify

Uses a length threshold to separate whisker segments from other traced segments (e.g. facial hairs). Looks for frames with the expected number of long segments and provides a preliminary label according to their order on the face.

Usage

```
classify --help
classify <source.measurements> <destination.measurements> <side>          --px2mm <double> -n <int> [--
limit<low>:<high>] [--follicle <int>]
classify <source.measurements> <destination.measurements> <x> <y> <axis> --px2mm <double> -n <int> [--
limit<low>:<high>] [--follicle <int>]
```

Parameters

```
-h, --help
```

Displays a help message.

```
<side>, <x> <y> <axis>
```

<side> may be left, right, top, or bottom.

Indicates the side of the image along which the face is oriented. The value effects how follicle and tip position are estimated.

<side> is used for videos recorded in such a way that the mouse face lies on one side of the image with the whiskers protruding into the field. Using <side> may not work well if the face is not confined to one side of the image or when the whisker pad is located near a corner of the image.

When <side> doesn't work well, use <x> and <y> should be the approximate coordinates of the center of the whisker pad. <axis> should be x or h if the face (the anterior-posterior axis) lies mostly in along the image's horizontal axis. Otherwise, <axis> should be y or v to indicate the vertical axis.

```
<source.measurements>
```

The source file containing measured features of traced segments.

```
<destination.measurements>
```

The destination file. This may be the same as <source.measurements>. Contains the table of measurements in <source.measurements> with updated assigned identities.

```
--px2mm <double>
```

The pixel size in millimeters. Should be in units of mm/px.

`-n <int>`

The number of segments longer than the length threshold expected in most frames. Typically, this is the number of imaged whiskers.

If set to -1, `#classify` will choose the minimum length threshold that maximizes the number of frames with the same number of long segments.

`--limit<low>:<high>`

Optional.

`#classify` adaptively finds the a minimum length threshold that yields the most number of frames with long whiskers. Use `<low>` and `<high>` to bound the length threshold. These should have in units of mm.

`--follicle <int>`

Optional.

Only count follicles that lie on one side of the line specified by this threshold (in pixels). The direction of the line points along the x or y axis depending which is closer to the orientation of the mouse's face.

Example

Run `classify` for a video with the face on the left of the image with a pixel size of 40 μm . Label a segment as a whisker if the segment is longer than a threshold between 1 mm and 50.0 mm and if the follicle is found on the left of the vertical line: `x=100 px`. Automatically try to determine the number of whiskers.

```
classify source.measurements destination.measurements left --px2mm 0.04 -n -1 --limit1.0:50.0 --follicle 100
```

reclassify

Refines an initial guess of segment identities.

Usage

```
reclassify --help
reclassify -n <int> <source.measurements> <destination.measurements>
```

Parameters

`-h, --help`

Displays a help message.

`<source.measurements>`

The source file containing measured features of traced segments with initial guess's for segment identities.

`<destination.measurements>`

The destination file. This may be the same as `<source.measurements>`. Contains the table of measurements in `<source.measurements>` with updated assigned identities.

`-n <int>`

The number of whiskers imaged. Typically, this is the same number used for `#classify -n <int>`.

If set to -1, `#reclassify` will use the guess provided by `<source.measurements>`.

Example

```
reclassify -n 3 source.measurements destination.measurements
```

whisker_convert

Converts between different formats available for measurements files. This utility is mainly used for updating to newer file formats.

It can also be used to convert files to a "human readable" text format.

Usage

```
whisker_convert -h
whisker_convert <source.whiskers> <destination.whiskers> <format>
```

Parameters

`-h, -help`

Displays a help message.

`<source.whiskers>`

The source whiskers file.

`<destination.whiskers>`

The destination file. This may be the same as `<source.whiskers>`, in which case the original will be overwritten.

`<format>`

The destination format to use. To get a list of available formats see:

```
whisker_convert -h
```

Warning

The `whiskpoly1` format is a lossy format and may not retain interesting aspects of traced whiskers such as the angle near the follicle. Don't use it.

Example

```
whisker_convert source.measurements destination.measurements whiskbin1
```

measurements_convert

Converts between different formats available for measurements files. This utility is mainly used for updating to newer file formats and converting to "human readable" text.

Usage

```
measurements_convert -h
measurements_convert --<source.measurements> <destination.measurements> <format>
```

Parameters

`-h, -help`

Displays a help message.

`<source.measurements>`

The source measurements file.

`<destination.measurements>`

The destination file. This may be the same as `<source.measurements>` in which case the original will be overwritten.

`<format>`

The destination format to use. To get a list of available formats see:

```
measurements_convert -h
```

Example

```
measurements_convert source.measurements destination.measurements v1
```

report

Compares two `.measurements` files and prints a report of differences in identity assignments between whiskers.

Usage

```
report (-diff|-hist) <A.measurements> <B.measurements>
```

Parameters

`-diff`

Lists the frames where differences were discovered.

`-hist`

Histograms the number of differences per frame.

`<A.measurements>`, `<B.measurements>`

The two measurements files to compare. The same set of whiskers should be labelled in both. The labels used to identify trajectories do not have to be the same; `report` will find the most likely correspondence given the curve features recorded in the `.measurements` files.