Technical support for `TrackUtil`

Fritz Francisco
fritz.francisco@hu-berlin.de

April 8, 2020

1 `TrackUtil`

TrackUtil was developed mainly by Paul Nührenberg at the University of Konstanz, Germany and provided as part of the supplemental information of the pre-print publication:

Francisco, Fritz, Paul Nuehrenberg, and Alex L. Jordan. "High-resolution animal tracking with integration of environmental information in aquatic systems." *bioRxiv*(2020).

TrackUtil is written in the common programming language Python and utilizes the machine vision library OpenCV. Its main purpose is to ease the process of annotating data-sets for machine learning approaches (specifically detection and image segmentation application Mask-RCNN). Further, output in form of trajectories obtained from predictions of the network can be refined using `TrackUtil` in order to create continuous tracks with individual identities and merge these across multiple videos/views, in which they are detected.

1.1 Installation

For ease of use we recommend setting up a virtual env using Anaconda/Miniconda and Python 3. A virtual environment basically consists of a container within your operating system, in which you then can install a specific version of python and all its dependencies, without the risk of disrupting your base system.

To do this follow the following steps:

1. Follow the "Regular Installation" instructions to obtain and install Miniconda here: https://docs.conda.io/
2. Once installed, create a `conda` environment using the supplied **trackutils.yml** [Linux],**trackutils_win64.yml** [Windows,Mac] file in the Anaconda Prompt (Windows) or a Terminal (Mac/Linux):

Command:

```
conda env create -f trackutils.yml
```

3. When in the Anaconda Prompt/Terminal, initiate the environment by running:

   Command:

   ```
   conda activate trackutils
   ```

4. Move to the directory containing **TrackUtil.py**:

   Command:

   ```
   cd path/to/TrackUtil/
   ```

5. Start TrackUtil by running:

   Command:

   ```
   python3 TrackUtil.py
   ```

2 Running `TrackUtil`

Once everything is set up you can either annotate or refine tracks using `TrackUtil`.

2.1 Start Application

Start `TrackUtil` by following the steps 3-5 described in 1.1

2.2 Annotating

Annotating refers to the process of manually labeling data in order for a machine learning process to learn from these, initial annotations. In this case the data comes in form of images, in which the objects are to be labeled as binary, image masks. These black and white masks (0,1), with the same dimensions as the initial image, define the outline of the objects (white,1) that should later be detected automatically by a trained network and discard the background, having no information (black,0).

1. After having started `TrackUtil` (see Installation 5) you can load a video by opening the dropdown menu **File → Load video**

2. Next we need to create a class to annotate. This is done by clicking on the **Annotator** drop-down menu and writing a designated class name in the **Add class** input field. The class is created by hitting the RETURN/ENTER key. You only need to create a class once per session but be sure to stay consistent with the naming in all possible ways (spelling,capitalization,no-spaces!) since this is vital for the further training steps to work. Therefore, a class "Guppy" should further always be referred to as "Guppy" and when labeling other videos should also be named "Guppy".
   Now you can navigate through the individual video frames and annotate objects in the images by drawing an outline, which is done by left click.

3. When finished with a video be sure to save annotaions and the process to a annotations file **AND** dataset file! This is done by navigating to the **File** → **Save annotations** or **File** → **Write to dataset** respectively. It is recommended that you also write to a global dataset file for each trial or experiment. This means you **File** → **Write to dataset** but create another file named destinctly different from the other (for example 'global_dataset.h5'). You should overwrite this file for every video (**File** → **Write to dataset** → **select the 'global_dataset.h5' file**) since this adds the annotations to it creating a single file for all videos.

2.3 Merging

Merging refers to the process by which trajectory snippets with individually assigned identities are combined througout a vieo to continuous tracks for each individual. These video-specific tracks and their corresponding identities can then further be matched across multiple videos/views (in a multi-camera set-up). This allows us to know the location of each tracked individual, when observed from each camera head.

2.3.1 Single file merging

1. After having started `TrackUtil` (see Installation 5) you can load a video by opening the dropdown menu **File** → **Load video**. Further you should load the corresponding trajectory file (.pkl) **File** → **Load tracks** and set the number of individual in the **Settings** menu.

2. You should now see the video with overlaid masks onto it. Further, by pulling up the handle at the bottom edge of the window you can see the visualization of the detection snippets. These can now be merged by going through the video and assigning them to each other, based on the individual identities they belong to.
   In order to do so you can access the keyboard shortcuts for easy of use by navigating to the drop-down menu **Tracks**.

3. Once a single video is fully merged, be sure to save the progress to a new file named **'merged_....pkl'** with **File** → **Save tracks**. You should also

save intermittently since the software seems to crash quite often. It is also good practice to never overwrite the original file and always keep it as backup!

4. You can easily return to a project that was saved before by using **File** → **Load Tracks** and selecting the corresponding tracks file.

2.3.2 Multi file merging

1. Following the same procedure as in 2.3.1 you can merge tracks and their corresponding identities with each other across multiple videos/views as well. To do so, navigate to **File** → **Connect**. This will open another `TrackUtil` window in which you can load another video and it's tracks. To make sure that identities are synchronized across both files navigate to the **Settings** menue and select **Fix identities in merge connected**

2. Again, try to save as often as possible (**File** → **Save tracks**) and ideally to a file created and named **'multimerged_view1_view2....pkl'** with `view1` and `view2` corresponding to the camera names that were merged.

3 Further notes

As some might be working on linux computers or may encounter a linux command line from time to time it may be useful to know some basic linux commands:

`ls` : ["list"] lists all files in the current directory

`cd` : ["change directory"] changes into the directory following the command. If no directory is specified it will change into the basal `home` directory

`cp` : ["copy"] copy specified files to designated directory. `cp` is followed by the filenames to be copied and then the designated directory

`mv` : ["move"] moves specified files to designated directory which does not replicate them. `mv` is followed by the filenames to be moved and then the designate directory

`pwd` : ["print working directory"] shows the current directory

4 Final Remarks

Using self-developed software can be challenging, since it is still very much a beta-version and continuously being updated as development moves on. Therefore is is also very important to make notes of certain aspects that do not work very well or instances when errors occur. It would be great if you can not whenever errors occur in order that we can update the software and make it more

user-friendly in the future.

If you have any questions please feel free to contact me and I'll try my best to help!