TRACKTOR v0.1 - tutorial

Last updated: 04.2018

If you find errors in this tutorial and/or identify links that are no longer working or outdated information, please notify Vivek Hari Sridhar by [email](mailto:vsridhar@orn.mpg.de) or by reporting an issue on [GitHub](https://github.com/vivekhsridhar/tracktor). These issues will be fixed as soon as possible.

# About

Tracktor is an OpenCV based object tracking software. The software is able to perform single-object tracking in noisy environments or multi-object tracking in uniform environments while maintaining individual identities.

Tracktor is command based (i.e. there is not graphical user interface [GUI]) but anyone with basic coding/scripting skills such as [R](https://www.r-project.org/) users can readily use it. The tool is aimed at teaching biologists the basics of computer vision while solving relatively easy tracking problems.

# Requirements

Tracktor works on all operating systems (Windows, Linux and Mac). Below is a step by step guide to install everything required to run Tracktor. Please note that most of the lines of code you will have to run for installation have to be run in the Terminal ([Mac](https://www.macworld.co.uk/how-to/mac-software/how-use-terminal-on-mac-3608274/)/[Linux](https://www.howtogeek.com/140679/beginner-geek-how-to-start-using-the-linux-terminal/)) or the [Command Prompt](https://www.digitalcitizen.life/7-ways-launch-command-prompt-windows-7-windows-8) (Windows), which is readily available in your computer.

# Installation

Follow the steps outlined below. We recommend to install opencv and run Tracktor within a virtual environment using miniconda. This makes it 1) easier to install and 2) potentially prevents many troubles if your system already uses python (Mac/Linux) or that you already have another python version installed.

*1) Install miniconda*

Miniconda is an open source distribution of Python that aims to simplify package management and deployment. Python is a programming language and requires an interpreter (i.e. software) to run. I recommend using Python 3 since the code for Tracktor was designed in this version. Download and install miniconda with Python3 from here (please note that Anaconda, a more elaborate version of miniconda, will also work in the same manner):

<https://conda.io/miniconda.html>

Important : tick the box that says "Add Python X.X to PATH” when the installer launches otherwise you will have issues running pip (see step 2 below).

*2) create and activate a virtual environment in miniconda*

Creating a virtual environment will allow to install everything you need in this separate “box”, thus preventing to make any changes in your main system. Once miniconda is installed, you can create a virtual environment with the command in the terminal/command prompt:

conda create --name myenv

(replace “myenv” with any name you want to give to this environment)

You will be asked to confirm (proceed ([y]/n)?), type “y”.

That’s it, your virtual environment has been created. Now you can activate it (=work from within this environment) anytime you want with the command:

source activate myenv (Mac/Linux)

or

activate myenv (Windows)

(replace “myenv” with the name of your environment)

**All the following steps in this tutorial should be performed from within the virtual environment (= after activating it).**

At all times, you can check whether you are working in the virtual environment or not by simply looking at the terminal/command prompt: if you are in the virtual environment, its name will be displayed in parentheses before all lines of code. If you want to deactivate the virtual environment, run:

source deactivate (Mac/Linux)

or

deactivate (Windows)

*3) Install the necessary Python Packages*

[Pip](https://pip.pypa.io/en/stable/) can be used to install all packages. Pip is typically already installed with Python, but if you are having difficulty running it, information is available [here](https://pip.pypa.io/en/stable/). To install a package, simply run “pip3 install packagename” in the Terminal or Command Prompt. Install the following packages (e.g. “pip3 install numpy”):

* [numpy](http://www.numpy.org/)
* [pandas](https://pandas.pydata.org/)
* [scipy](http://www.scipy.org/)
* [scikit-learn](http://scikit-learn.org/stable/)
* [Jupyter notebook](http://jupyter.org/install)\*

\*Jupyter notebook is optional. However, it is highly recommended it for interactive coding. All example notebooks are written as jupyter notebooks, and it also makes it generally easier to run code.

*4) Install OpenCV*

[OpenCV](https://en.wikipedia.org/wiki/OpenCV) (Open Source Computer Vision) is a library of programming functions focused on real-time computer vision. OpenCV is freeware that works across all platforms but installing it on your computer might be the trickiest part of getting Tracktor to run on your machine. Fortunately miniconda has an easy-to-use installation of OpenCV. Run:

conda install -c menpo opencv3

If you get no error message, OpenCV is now installed in your virtual environment. You can check if the installation worked by typing:

python

import cv2

If these two lines give no error: congratulations, the installation was successful! You can **exit python for now, using the exit() command**.

It is possible that the installation does not work on your machine as described. If you struggle with this part, please check online for solutions, or get in touch with a IT specialist, a computer savvy colleague, or us to help you out. As mentioned, this is the trickiest part of getting Tracktor up and running, but as soon as OpenCV is properly installed, you will be able to track videos. More information: <https://anaconda.org/menpo/opencv3>

*5) Download Tracktor*

Go on Tracktor’s GitHub page <https://github.com/vivekhsridhar/tracktor> and click the green “clone or download” button. Download the .zip file to your choice location on your computer, and unzip the folder.

Tracktor is basically just a set of lines of code, there is no “installation” required as for most software, and there is no GUI (Graphical User Interface). You will need to enter the various parameters (e.g. name of video, location of the video, etc.) directly into the code, but we will guide you through it for an easy start. For your understanding, here’s a short description of the content of Tracktor. All the files or folders labelled here as “non-relevant” are not necessary to understand how to work with Tracktor.

[.ipynb\_checkpoints](https://github.com/vivekhsridhar/tracktor/tree/master/.ipynb_checkpoints) non-relevant

[\_\_pycache\_\_](https://github.com/vivekhsridhar/tracktor/tree/master/__pycache__) non-relevant

[logo](https://github.com/vivekhsridhar/tracktor/tree/master/logo) non-relevant

[output](https://github.com/vivekhsridhar/tracktor/tree/master/output) non-relevant

[videos](https://github.com/vivekhsridhar/tracktor/tree/master/videos) This is the folder where the example videos are stored

[.DS\_Store](https://github.com/vivekhsridhar/tracktor/blob/master/.DS_Store) non-relevant

[LICENSE](https://github.com/vivekhsridhar/tracktor/blob/master/LICENSE) non-relevant

[README.md](https://github.com/vivekhsridhar/tracktor/blob/master/README.md) This Tutorial

[logo-small.jpg](https://github.com/vivekhsridhar/tracktor/blob/master/logo-small.jpg) non-relevant (Tracktor’s logo)

[single\_fish.ipynb](https://github.com/vivekhsridhar/tracktor/blob/master/single_fish.ipynb) Example. Jupyter notebook. This code was created to track a single fish in a noisy environment.

[spider\_track.ipynb](https://github.com/vivekhsridhar/tracktor/blob/master/spider_track.ipynb) Example. Jupyter notebook. This code was created to track 2 spiders of very different sizes in a controlled environment, maintaining identities.

[track\_termites.ipynb](https://github.com/vivekhsridhar/tracktor/blob/master/track_termites.ipynb) Example. Jupyter notebook. This code was created to track up to 8 termites in a controlled environment, maintaining identities.

[tracktor.py](https://github.com/vivekhsridhar/tracktor/blob/master/tracktor.py) This is the actual program. All the functions used in tracktor are defined here. It will not run by itself, but all 3 examples above use the functions available in this file. For unexperienced users, I recommend not touching this file, but rather pick one of the jupyter notebooks available above and change the parameters to fit your actual tracking problem.

# Running Tracktor

*6) Open a Jupyter notebook*

We recommend running Tracktor through Jupyter. For this, first make sure your virtual environment is activated (see section 2). Then start Jupyter by typing:

jupyter notebook

This will open a window in your browser (e.g. Chrome, Firefox, etc.), displaying your folders. We will first describe here how to run one of the example notebooks we provide in Tracktor. Navigate to the tracktor folder and open one of the example jupyter notbook (e.g. “single\_fish.ipynb”). Jupyter will ask you to select the kernel you want this notebook to run in. It’s important that you select your virtual environment because that is where your installation of OpenCV is. If you are not asked directly when the notebook opens, you can change it later by clicking on the top menu on “kernel > change kernel” and selecting your virtual environment.

*6) Running the example code*

All the code has been commented for ease of understanding. You will find explanations for the various functions used, how to change the parameters and what each parameter does, directly in the code. In the jupyter notebooks, you can access the description of any function by placing the cursor in the parentheses after a function and pressing **shift tab**. This will display the comments associated with this function from the main code. Please note that the descriptions we provide are only for the Tracktor functions (all functions starting with “tr.” in the jupyter notebooks), and that the descriptions will be very minimal for the all the other functions (i.e. the ones coming directly from python or OpenCV).

Alternatively, you can also find all of the detailed comments in the main code for Tracktor: tracktor.py

The example jupyter notebooks are organized in 3 sections (called “cells”). The first cell simply imports all the packages of functions necessary to run the code. If you have an error there, it is very likely that one of the packages hasn’t been installed properly. If this happens, please go back to sections 3-4 and check that everything has been installed correctly.

The second cell deals with all the input parameters. For instance, this is where you can choose how many targets to track, their maximum and minimum sizes, the name of the input video, etc. When you use Tracktor on your own data, these are the parameters you will have to tune to fit your problem, by trial and error.

The third section is the actual tracking. The first line is the path to the video so you need to replace this path with the location of the video on your computer. That’s it! You can now run the entire code and it should track the example video of your choice (fish, termites, spiders).

Once the tracking is done, Tracktor will output 2 files: one video with the tracking overlaid and one .csv file with the XY coordinates of the result of tracking.

*7) Running Tracktor on your own videos*

Once you have familiarized yourself with Tracktor through our examples, you can start to modify the example codes for your own purpose. Play around with the different parameters in the second cell to find the ones that will produce the best tracking for your problem. If you encounter issues, or cryptic error messages, we recommend to search in Google for answers. A large community is using OpenCV-based solutions for tracking and Stack Overflow is full of answers. If your problems persist, don’t hesitate to get in touch with us by [email](mailto:vsridhar@orn.mpg.de) or by reporting an issue on [GitHub](https://github.com/vivekhsridhar/tracktor), we will do our best to help you getting started with Tracktor.