Pyper Documentation

Release 1.0.0

Charly Rousseau

Jun 15, 2017

Contents

CHAPTER 1

Introduction

This program allows the tracking of specimen (e.g. a mouse, a rat ...) in an open field maze from pre-recorded videos or from a live feed. The live stream of frames can be generated from a usb camera or from the camera (normal or NoIR) of the Raspberry Pi. On the Raspberry Pi, a subclass of the standard PiCamera object is used to speed-up video acquisition and online processing.

The combination of recording and online position tracking allows the definition of complex behavioural experiment as the program can send a TTL pulse to an other computer upon detecting the mouse (for example: in a certain region of space).

The modules can be used interactively from the python interpreter or through the provided interfaces. This program provides both a command line interface and a graphical user interface. For the CLI, the defaults are saved to a user specific preference file.

Some basic analysis of the extracted position data is also available. Two classes are also supplied for viewing of the recorded videos or transcoding.

Please see the instructions below for installation and usage.

CHAPTER 2

Example

Fig. 2.1: An example of the tracking software in action.
The yellow square at the bottom right lights up when the mouse enters the predefined Region Of Interest (the yellow circle). This behaviour can easily be overwritten by the user by specifying a different function.

CHAPTER 3

Documentation

Installation

Dependencies

You will need at least:

- opencv 2.4.9.1
- numpy
- skimage
- python-progressbar

Recommended modules

These modules are required only for some of Pyper's modules and won't be necessary if you do not use these optional features (e.g. the GUI).

Command line

- configobj

Acquisition

- picamera (for the raspberry pi)

Analysis

- scipy

GUI

- pyqt5
- PyOpenGL

Installation on linux (assuming a debian based distribution)

Note: Please note that the GUI will currently not work on the raspberry pi. This is due to a limitation of the video driver supplied by Broadcom. This program uses QtQuick which in turn uses hardware acceleration to minimise the load on the CPU. The current driver does not support this feature. This issue might be solved in the future using EGLFS. Also, the new UP project by Aaeon should not have this issue as it uses an Intel GPU.

Important: If you use an Ubuntu distribution, Ubuntu removed the PyQt5 bindings for python 2.7 in 14.04 but reintroduced them afterwards. Please ensure that you use a version for which the *python-pyqt5* is available and not only *python3-pyqt5*. You can check this using:

```
apt-cache search python-pyqt5
```

To install the dependencies, run:

```
sudo apt-get update
sudo apt-get install python-opencv python-skimage python-progressbar python-configobj␣
↪python-scipy git
```

Then download the motion tracking program using:

```
git clone https://github.com/SainsburyWellcomeCentre/Pyper.git pyperMotionTracking
```

If you want to use the command line interface, copy the motionTracking.conf file from the src folder to your home folder preceded by a dot. Assuming pyperMotionTracking is in your home folder, type the following command:

```
cp ~/pyperMotionTracking/src/motionTracking.conf ~/.motionTracking.conf
```

The following will install the additional dependencies for the GUI:

```
sudo apt-get install python-pyqt5 python-opengl python-pyqt5.qtopengl python-pyqt5.
↪qtquick qml-module-qtquick-controls
```

Finally, for the raspberry-pi camera:

```
sudo apt-get install python-picamera
```

Remember to activate the camera in raspi-config

```
sudo raspi-config
```

Then select camera -> activate

Installation on MacOSX (tested on Mavericks)

Installation instructions by Christian Niedworok.

Installing Homebrew:

Homebrew is a package manager that allows to install a lot of standard open source software on mac that wouldn't be available otherwise. One of them is OpenCV.

Important: You will need XCode to install Homebrew

If you have the OSX 10.10 you can install Xcode from the app store, otherwise you need to go to https://developer.apple.com/xcode/, sign in with your apple account (you may have to register as a developer to do this) and download an earlier version. The last version that runs on OSX 10.9 is Xcode 6.2.

Note: After installation of Xcode make sure you start it, since it will finalize the install upon its first launch. Be advised that downloading and installing Xcode can take considerable time (>30 minutes).

Then, you can install homebrew.

```
ruby -e "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/master/
↪install)"
```

the installer will run and probably tell you it will change some user rights. For example: *"The following directories will be made group writable: /user/local/lib"*. It will also probably ask you to confirm with enter and prompt for your admin password.

Now we have to make sure homebrew software is visible to the system. Open a new terminal **window**, and in there, type:

```
echo $PATH
```

and check whether you can see both of the following in the output: "/usr/local/sbin" and "/usr/local/bin"

if "/usr/local/bin" is missing, run the following:

```
echo 'export PATH="$PATH:/usr/local/bin"' >> ~/.bash_profile
```

if "/usr/local/sbin" is missing, do the same but replace /usr/local/bin by /usr/local/sbin

Now open another new terminal window, close the other (old) terminals, run the command in the "important" box below and get ready to install openCV and python.

Important: Homebrew will potentially install additional versions of software you might already have on your system. This software will be installed to /usr/local/. To prevent these versions from clashing, run the following command whenever you are working on the terminal and want to use homebrew or a software that has been installed using homebrew.This will ensure that - during the currently open terminal session - the homebrew versions have precedence over any other potentially installed versions.

```
export PATH="/usr/local/bin:$PATH"
```

Installing openCV with python:

Please note that there is a default python on the mac that should not be modified. Unfortunately for us though, it is quite an old version. So we'll install a new one and use/modify that one.

Note: Be aware that the installation with homebrew may take some time and will use processor resources as it will need to compile software.

```
brew tap homebrew/science
brew install --with-ffmpeg opencv # Option to have codecs support
brew install python
```

The following will set up python for package downloads and create an alias called brewPython that will run the python you just installed.

```
mkdir -p ~/Library/Python/2.7/lib/python/site-packages
echo 'import site; site.addsitedir("/usr/local/lib/python2.7/site-packages")' >> ~/
↪Library/Python/2.7/lib/python/site-packages/homebrew.pth
echo 'alias brewPython="/usr/local/bin/python"' >> ~/.bash_profile
```

If you want to use this version of python from your standard mac "Applications" folder, run:

```
brew linkapps python
```

The following will now install python dependencies for the motion tracking software:

```
sudo  -E /usr/local/bin/pip install numpy scipy scikit-image python-dateutil
sudo  -E /usr/local/bin/pip install pyparsing matplotlib image
sudo  -E /usr/local/bin/pip install PyOpenGL progressbar
```

Installing the GUI:

The Graphical User Interface relies on a graphical library called QT (initially developed by Nokia). To use the GUI, you will need to install this library and its python bindings.

Caution: QT5 with homebrew requires OS X Lion or newer

To install QT via homebrew first open a terminal, ensure proxies and $PATH are set (see above), then copy this:

```
brew install qt5
brew install PyQt5 --with-python # Installs the bindings for python 2.7 which is␣
↪necessary for openCV 2
```

Getting the program

Finally download the motion tracking program using:

```
git clone https://github.com/SainsburyWellcomeCentre/Pyper.git pyperMotionTracking
```

If you want to use the command line interface, copy the motionTracking.conf file from the src folder to your home folder preceded by a dot. Assuming pyperMotionTracking is in your home folder, type the following command:

```
cp ~/pyperMotionTracking/src/motionTracking.conf ~/.motionTracking.conf
```

At the end if the program doesn't start, try running:

```
brew update
brew upgrade
brew doctor
```

This should let you know if there are any issues with your homebrew installation. It might be that homebrew is asking you to link some libraries. If so follow the instructions on screen. Ensure that /usr/loca/lib is writable.

Installation on Windows

Instructions by Andrew Erskine

To install python you can use a science oriented python distribution. Please make sure you download python 2.7 Then to install the dependencies, you can follow the *pip* commands of the MacOS instructions. E.g.:

```
pip install numpy scipy scikit-image python-dateutil pyparsing matplotlib image␣
↪PyOpenGL progressbar
```

The core of the program works fine. You just have to install openCV and link it with your version of python:

- Download OPENCV for Windows: http://opencv.org/downloads.html
- Extract the file (automatic) (doesn't have to be Python folder)
- Go to the folder where you extracted OpenCV and find opencv\build\python\<yourversion (e.g. 2.7)>\<yoursystem (e.g. 64-bit)>\cv2.pyd
- Copy the cv2.pyd file and put it in C:\<PythonFolder (e.g. Python27)>\Lib\site-packages\
- Open a python console and check it worked:

```
>> import cv2
>> print cv2.__version__
```

Finally download the motion tracking program using:

```
git clone https://github.com/SainsburyWellcomeCentre/Pyper.git pyperMotionTracking
```

If you want to use the command line interface, copy the motionTracking.conf file from the src folder to your home folder preceded by a dot.

The GUI however should work but has not been tested because the python bindings for QT5 are not provided for python 2.7 on windows. If you would like to use the GUI, you will have to compile pyqt5 for python 2.7. This has not been tested here.

Quick start

Starting the program

To get a quick feel for the software, some example files have been included in the resources folder of the program. Start by opening the program, in a terminal:

Change your working directory to that of the source of the program (subfolder src). This must fit your installation folder. For example if user rcajal downloaded on the desktop:

```
cd ~/Desktop/pyperMotionTracking/src
```

Then actually launch the program:

```
python tracking_gui.py
```

Note: You may want to use brewPython if you followed the mac instructions or a full path on windows

The graphical interface should now be started.

Preview

Open a file using the file menu or your platform shortcut and select the *teleporter.h264* file in the resources subfolder of the program (i.e. where you downloaded it with git). You can now go to the *Preview* tab and navigate through the file.

- Select a reference frame before the specimen enters the arena (also avoid the very first image).
- Select a start frame (after the reference) where the mouse has entered the arena.
- Select the end of the tracked portion or leave it as is to select the end of the recording by default.

Tracking

You can now navigate to the *Track* tab. Press the *start tracking* arrow (*play* button) and you should see the video begin to play. At the beginning, the video should play as the source. Note that the resolution is better than in the preview because the preview is downscaled. Once the video playback reaches the point that you selected for the beginning of the tracking, the mouse should become outlined in red and leave a green line as a trail delineating its trajectory. You can then select a different image type from the drop down menu at the bottom of the window to get a feel for the processing going on in the background.

Analysis

Once the end of the recording is reached, you can proceed to the *Analyse* tab and click **Update**. The list of coordinates will now appear in the table. Coordinates of (-1, -1) indicate default coordinates that are selected if:

- The frame is before the beginning of the tracked segment of the recording.
- The specimen could not be found in the arena with the specified parameters.

If you now click angles and distances two graphs should appear indicating the change of direction at each frame and the distance made at each frame respecively. To save the coordinates, select **Save** and provide a destination path.

Recording

If a camera is plugged in to your computer (or it is a laptop with a built-in webcam), you can try the recording mode.

- Go to the *Record* tab and select a destination file path. Using the default .avi extension should work in most cases. If the output is empty at the end of your recording, try one of the other file formats instead.
- Select **Ref** = 5 and **Start** = 6.
- Now stand in front of the camera and try to be as still as possible. Press the record button. The video should start to appead.
- Move slightly to either side, the program should start tracking your movements.

You can now go back to the *Analyse* tab and tick the *Recording* box. Now click **update** again and do the plots again. These should now show the new data from the recording.

Usage

This software can be used in different ways. You can use the graphical interface, the command line interface (from the terminal) or import the modules from a python interpreter.

Using the Graphical interface:

To use the Graphical interface, start a terminal window and then:

Change your working directory to that of the source of the program (subfolder src). This must fit your installation folder. For example if user rcajal downloaded on the desktop:

```
cd ~/Desktop/pyperMotionTracking/src
```

Then actually launch the program:

```
python tracking_gui.py
```

When using the graphical interface, hovering over buttons and other elements should display tooltips explaining what these do.

You are greeted by the *welcome* tab. To navigate between functionalities, please select a corresponding tab on the left. The *Preview* and *Track* tabs require a video to analyse to be loaded. To do so, use the *file* menu or the standard open shortcut on your platform. The record tab will pop up without a camera attached but will require one to select the path to save the video. The *analysis* tab requires to have performed the tracking or recording step before use.

Please note never input empty parameters in the number fields as these are checked for valid numbers.

The *Preview* tab

The preview tab enables you to have a look at a downscaled version of your video to select your start and end point for the analysis as well as the frame that will serve as a reference. Currently, the reference frame must appear before the data frames. This may change in the future and also a dialog may be added to load a picture to serve as a reference. You can navigate in your video using the controls on the left of the progress bar at the bottom. Then, when on the desired frame select **Ref**, **Start** or **End** accordingly. An end of -1 corresponds to the end of the file. It is advised to select a frame that is not the first one (e.g. 5 onwards) for the **Ref** as the camera may take a few frames to adjust some parameters and the video may also alter the very first frames.

The *Track* Tab

This tab is more advanced. It allows you to Track a specimen in the open field from a pre recorded video. The **Ref**, **Start** and **End** parameters set in the *preview* tab apply to the *Track* tab.

The parameters below (except the type of frame displayed) will only be taken into account the next time the **track** button is pressed.

You should then set a threshold **Thrsh** for the brightness of you sample in the difference image. You can get an idea for this parameter by using the *diff* option from the drop down menu at the bottom. The **min** and **max** parameters refer the minimum and maximum areas of the sample respectively. These are experessed in pixels^2. The **Mvmt** parameter refers to the maximum displacement **in either dimension** of the specimen between two consecutive frames.

In addition, you can supply a number of frames to average for the reference (starting from the **Ref** frame). If detection is difficult, specifying several reference frames and setting the **Sds** parameter will enable you to use variation in the current frame relative to the standard deviation of the averages.

Finaly, the **Clear** parameter will clear object touching the border of the image, The **Extract** parameter is used if the arena is white on a dark background, this option will automatically detect it as an ROI. Finaly, the **Norm** parameter removes the slight variations in global brightness between frames. it will normalise each frame to the brightness of the reference.

The default detection parameter should be appropriate for the example videos.

The drop down menu allows you to select between the type of image (level of processing) you want to display. Please note that this option only applies to the portion of the recording that is being actively tracked, the ignored portion will just be displayed as the source (i.e. *raw*).

Finaly, you can define a region of interest using the yellow circle icon. When the mouse enters this ROI, the program will trigger a callback method. The default method draws a square at the bottom right of the image but this behaviour can be altered by overwritting the callback method in a subclass of the GuiTraker class (see API).

If you cannot detect your sample successfuly, please refer to the troubleshooting section.

The *Record* Tab

This tab essentialy reproduces the behaviour of the *Track* tab but for videos that are being recorded through a USB or FireWire camera for example. Before starting you must supply a destination path to save the video. The extension will determine the format. The available formats will dependend on the codecs available to ffmpeg in you installation. In tests, best restults were obtained with .avi and .mpg

The *Calibration* tab

This menu computes the parameters of the lens used to acquire the videos and once calculated, these parameters will be used automatically to undistort the images in the *Track* and the *Record* tabs. To use this functionnality, you must provide a folder containing a series (e.g. 10) of images acquired with the same lens and parameters as the video and containing a reference chessboard. A printable template can be found at http://docs.opencv.org/2.4/_downloads/pattern.png which will work with the default parameters. Once the folder is selected, press **Calibrate** and wait for the calibration to finish. Once done, the controls will become available and allow you to browse through the images used for calibration, the images with the features drawn and the undistorted images. You can also save the camera matrix for reference. In the future, it should become possible to load a calibration file from the *Track* and *Record* tabs.

The *Analyse* tab

This tab provides simple analysis and graphing features as well as the ability to save the list of coordinates.

First select the checkbox that corresponds to the tab you want to analyse. Then, click **Update** and you can finaly save your coordinates and plot graphs of distance made by the specimen between frames and change of directions at each frame. To save the graphs, right click on them and a menu will promt you for a destination path.

Using the Command Line Interface:

API (For programmers)

Only public methods and the constructor are documented here

The core

The graphical interface

The command line interface

The analysis

Utilities

Troubleshooting

The buttons are grayed out I can't use them

- The video must be loaded or processed for some functions to become available

The detection doesn't give me any result

- Please check that your sample is in the image during the frame range you are tracking
- broaden your size restriction and lower your threshold
- use the diff option and check what the camera is seeing as the diff.
- Select your threshold based on the previous test.
- See 'The diff is noisy and there is a lot of background' below for further troubleshooting.

The diff is noisy and there is a lot of background

- As the camera will setup/settle at the beginning of the recording and the video format may use compression (filtering) in time, using the very first frame as reference is usually inadvisable, considering using frame >= 5 instead for best results.
- If the images are noisy, you can consider using more than n=1 for the number of reference frames. In that case, the frames will be averaged and the signal n*SD above average used for tracking.

Please consider using high SD numbers in the case where background is detected.

I performed the tracking but no coordinates appear in the 'Analyse' tab

- Make sure that you select the proper mode (tracking or recording) via the UI tickboxes and select update.

I changed some parameters but the tracking is not affected

- The trackign can only run in a non interactive way. You need to click the stop button and play again after changing parameters (ref, threshold...) except for the type of frame displayed.

I see two ROIs

- This means you have drawn a new ROI after the tracking. One ROI belongs to the UI, the other to the image. If you now press track again, the image ROI will now match the UI ROI.

CHAPTER 4

Indices and tables

- genindex
- modindex
- search