

Patra2

ゾウリムシの追跡プログラム

- ソース: <https://github.com/jnishii/Patra2>
- <https://github.com/bcl-group/motion-capture> からの派生
- 上記motion-captureは[OpenCVを使ったモーション テンプレート解析](#)を参考に作成

環境構築

1. Python環境のインストール

```
$ pyenv install 3.10.2
$ pyenv rehash
$ pyenv global 3.10.2 # インストールしたPython 3.10.2 を利用するための設定
```

2. 確認

```
$ pyenv versions
  system
* 3.10.2 (set by /usr/local/var/pyenv/version)
$ which python
/usr/local/var/pyenv/version/shims/python
$ python --version
Python 3.10.2
```

3. Poetryのインストール

PoetryはPythonのライブラリ管理ツール。これを使ってPatra2に必要なPythonライブラリを管理する。

```
$ curl -sSL https://install.python-poetry.org | python3 -
```

4. Patraのダウンロード

```
git clone git@github.com:jnishii/Patra2.git
```

5. 必要なライブラリのインストール

```
$ cd Patra2
$ poetry install
```

Patra2の使い方

使用上の注意

- 解析したいファイルは patara2/Data/ の下に置く。サブフォルダを作っても良い。
- 出力ファイルは patra2/outputs/ の下に作られるサブフォルダ(ファイル名と同じ名前)に保存される。

解析の順番

- キャリブレーション (calibration.py)
- ゾウリムシの追跡 & 行動解析 (Patra2.py)
- 速さのフーリエ解析 (Speed_fourier.py)

キャリブレーションの取り方 (calibration.py)

1. マイクロメータの動画ファイルを解析ファイルと同様に460×640ピクセルとし、10mm×10mmのマイクロメータが収まるようにを撮影しておくこと！！
2. calibration.pyを起動し、キャリブレーション用ファイルを読み込む

```
poetry run python calibration.py <マイクロメータの動画ファイル名>
```

実行例

```
poetry run python calibration.py Data/chiba/1111/calibration.mov
```

3. 5mm × 5mm(1目盛り50μm) のマイクロメータの目盛り線を、縦目盛りの5mmの線上 → 縦目盛りの0mmの線上 → 横目盛りの0mmの線上 → 横目盛りの5mmの線上 の順にクリック。
4. クリック後にEnterキーで終了
5. ターミナルに以下のような出力が出る。これをゾウリムシの追跡のときに使うので記録しておく。単位はμm/pixel

```
X_scale: 15.1515151515
Y_scale: 15.5279503106
```

ゾウリムシの追跡 & 行動解析 (Patra2.py)

起動方法

起動方法その1

Patra2のディレクトリ内で以下を実行

```
$ poetry run python Patra2.py 解析したい動画ファイル名 X_scaleの値 Y_scaleの値
```

例

```
$ poetry run python Patra2.py sample/Ba_dance_3.mov 17.1613401 16.8496433
```

起動方法その2

run.shの中に、上記コマンドが書いてある。エディタで修正して以下を実行

```
$ ./run.sh
```

実行方法

1. 追跡したいゾウリムシをマウスクリック
2. 取り消したいものは、近くをクリック
3. Enterキーで追跡開始。指定したいいずれかのゾウリムシが画面から出るか、動画の終わりまで追跡
 - 途中停止はsキー。追跡強制終了はESC。
4. 終了後、グラフが表示される。確認したら Command-q で終了

出力データ

outputs/の下に解析した動画ファイル名のフォルダができる。

- ローパス（バターワース）処理： 直流成分は1/2, 約0-7.4Hzまではほぼ100%透過, 減衰域は7.4925Hz-9.99Hz
- 軌道データファイル Para<id>_<filename>.csv
 - データ列:
time,para_id,X,Y,X_lpf,Y_lpf,Vx,Vy,Vx_lpf,Vy_lpf,V,V_lpf,V_lpf2
 - time: 時間
 - para_id: ゾウリムシのID
 - X, Y: 位置情報(μm)
 - X_lpf, Y_lpf: X, Yをローパス処理したもの
 - Vx, Vy: X, Yの差分で求めた速度(μm/s)
 - V: Vx, Vyから求めた速さ
 - Vx_lpf, Vy_lpf: X_lpf, Y_lpfから求めた速度
 - V_lpf: Vx_lpf, Vy_lpfから求めた速度
 - V_lpf2: V_lpfをローパス処理したもの(注意: 稀に負の値になることがある)。
- 周波数分析データファイル Para<id>_<filename>_freq.csv
 - データ列: ,X_freq,X_amp,Y_freq,Y_amp,V_lpf2_freq,V_lpf2_amp
 - X_freq,X_amp,Y_freq,Y_amp: X, Y軌道の周波数分析結果
 - V_lpf2_freq,V_lpf2_amp: V_lpf2の周波数分析結果
- 周波数ピーク分析データファイル
 - 上記周波数分析データファイルからピーク周波数を抽出したもの

- Para<id>_<filename>_X_peak_freq.csv
- Para<id>_<filename>_Y_peak_freq.csv
- Para<id>_<filename>_V_peak_freq.csv

そのほか

- 全てのゾウリムシが画面から出るまで追跡できるようにプログラムを改造することも出来るが、ちょっと時間が必要