

gakka10_VB(2016年度版)

What is this ?

本学にて作成している「ヴァーチャルボーリング」というソフトウェアにおける、カメラモジュールです。主にボールの検出とモーションの追跡を行います。

ファイル構成

ball.py

ボールについてのクラスを定義しています。

field

- Ball.diameter

ボールの直径の情報を保持しています。

motion.py

カメラで解析したボールの動きについての情報を持つデータ構造としてのクラスを定義しています。

field

- Motion.position

ボールの x 軸方向についての初期位置を保持しています。

- `Motion.velocity`

ボールの x 速度、y 速度を格納したタプルを保持しています。

camera.py

二次元的に物体の運動を観測する用の `NormalCamera` クラスと、三次元的に物体の運動を観測する用の `StereoCamera` クラスが用意されています。

ダックタイピングにより、`NormalCamera` クラスと `StereoCamera` クラスには互換性が備わっています。現状、`StereoCamera` クラスは未実装です。

method

- `init(self, camera_id)`

カメラ ID を入力して、そのカメラにて映像を表示するためのオブジェクトを生成します。 `StereoCamera` クラスの場合、このメソッドの中でカメラキャリブレーションを行なうことになるでしょう。

- `detectBallProperty(self)`

画面に表示される、検出できた円がひとつのみ場合に Enter キーを押すと、その円を運動を追いたいボールであるとして、`Ball` オブジェクトを返します。半径の値をコンソール上に表示しますので、値が安定してから Enter キーを押すと動作が安定するでしょう。

Esc キーを押すとプログラムを終了させます。

- `detectBallMotion(self, ball, waitTime = 0.5, resolution = 32)`

ボールの運動の解析した情報を返します。具体的には、ボールの直径と

同程度のピクセル幅を持った物体の座標を検知し、その初期座標と平均の速度を返します。 `waitTime` はボールを検出しなくなっからの待機時間を、 `resolution` は `y` 軸方向の分解能を意味します。

Rキーを押すといったん解析をリセットします。具体的には、いったん適当な値 `-1` を返してから、再度本メソッドを呼びだしています。 Escキーを押すとプログラムを終了させます。

注意 ゲームモジュールは中央を原点とする左手座標系—`x`軸は左、`y`軸は上、`z`軸は手前がそれぞれ正—であり、カメラモジュールでは左上を原点とする直交座標系—`x`軸は右、`y`軸は下がそれぞれ正—であるため、これら座標系の変換を行なう必要があります。なお、カメラモジュールでいう `x`軸、`y`軸はそれぞれゲームモジュールの`x`軸、`z`軸に対応しています。詳しくは、本メソッドが返す `Motion` クラスの引数についての処理を参照してください。

constant.py

プログラム内に登場する諸定数を定義しています。

list

- `MODE_3D`

True にすると `NormalCamera` の代わりに `StereoCamera` オブジェクトを生成します。

- `CAMERA_FPS`

カメラの FPS 情報を保持しています。30から変更する必要はないと思います。

- `CAMERA_WIDTH`

カメラの横幅を設定しています。

- CAMERA_HEIGHT

カメラの縦幅を設定しています。

- FILE_PATH

Unity モジュールとデータをやりとりするためのファイルのパスを設定しています。

- LANE_WIDTH

実際にボールを投げるレーンの横幅を設定しています（単位は m）。

main.py

プログラム実行用のファイルです。 これまでに紹介した諸々を用いて、ボールの検出、モーションの追跡、追跡データのゲームモジュールへの送信を行います。

よくわからないままプログラムの改変に望む人は、基本的にこのファイルは編集しない方がよろしいでしょう。

TODO

- C++への移行（パフォーマンスに問題が出る場合）
- StereoCamera クラスの実装

依存しているライブラリ

- OpenCV 3.1