

# Raspberry Piによる画像を用いた動作制御

木更津工業高等専門学校 電子制御工学科5年 大場翔太

## 1. 初期の機体からの構成の変更

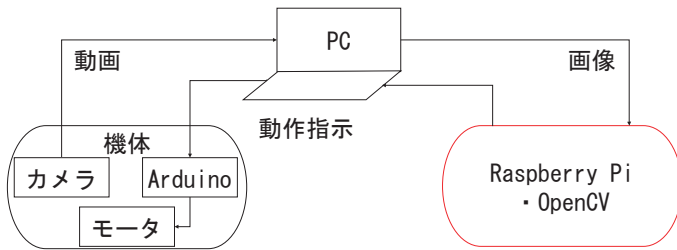


図1. 構成変更後

機体の初期構成でコントローラが担っていた動作信号の入力部を図1のようにRaspberry Piで置き換え、それに伴いRaspberry Piに搭載したOpenCVで処理して入力信号を決めるために画像を取り込む。大まかな流れは以下の通りである。

- (1) 水中ロボットから動画をPCへ送る。
- (2) 動画からRaspberry Piに画像を取り込む。
- (3) 取り込んだ画像をOpenCVで解析し、色と形の判別を行う。
- (4) 判別結果を元に進行方向とそれに伴うモータの出力用信号を決める。
- (5) 送られた信号からモータの動作を変更する。

## 2. OpenCVによる画像処理

今回使うOpenCVは、計算効率を優先しリアルタイムアプリケーションに重点を置いて設計されたコンピュータビジョンライブラリである。ライブラリには画像診断、セキュリティ、ユーザインタフェース、カメラキャリブレーション、ステレオビジョン、ロボット工学を含む、ビジョンの多くの領域にわたる500以上の関数が含まれています。CとC++で書かれており、PythonやRuby、Matlab、その他の言語用インタフェースも開発が進んでいる。<sup>1</sup>

今回使用する機能については以下に示す。

### 2.1 ビデオを読み込む

今回は水中ロボットからの動画を用いて制御するので、まずはビデオの読み込みが必要である。特に今回の状況下では、リアルタイムの動画を扱うので保存された状態の動画とは取り込み方が異なる。また、ある程度の長いビデオデータについて処理し続ける事が考えられるので、過剰な負荷を防ぐために一定フレーム毎に画像ファイルとして切り出すようにする。

### 2.2 画素を取得する

2.1で読み込んだ画像について、BGR値を取得する。水中では色の減衰率が異なるため、普段の色判別から調整して色を読み取る必要がある。また、事項で扱う輪郭検出のために二値画像への変換も必要となる。

### 2.3 輪郭を検出する

2.2で得た二値画像を使い、物体の輪郭を検出する。

### 2.4 輪郭を近似する

2.3で得た物体の輪郭をより少ない点で近似する。どれだけ精度で近似するかによって点の数が変わってくる。

### 2.5 物体の形を判別する。

2.4で得た点の数によって大まかな形を判別する。2.4で行う近似精度や、形の判別の際の点の数の閾値によって形の判別に誤差が出るので、実際に試しつつ値を変更する。

### 2.6 面積と重心を求める。

2.5で円形だと判別された図形について、面積と重心を、モーメントを出してから求めて、正面からのズレとある程度の距離を確認する。

### 2.7 モータへの出力信号を決める。

2.6で求めた左右上下方向のズレと距離からそれぞれのモータの出力を決める。

## 3. 参考文献

- 1 Gary Bradski & Adrian Kaehler 詳解OpenCV-コンピュータビジョンライブラリを使った画像処理・認識 1(2014)