

1 ソフトウェア

1.1 アルゴリズム

カメラがボールっぽい物体を捉えていたらそれに向かうベクトルを，なければ総移動距離から自己位置を推定し，予め決められたゴールへ向かうベクトルを採用．機体周囲に放射状に取り付けた測距センサから適当にベクトルを作って合成．機体正面に取り付けられた測距センサ三人衆から適当にベクトルを作って合成．ベクトルを適当に左右のモータへの速度指令値（正確には PWM 信号値）に変換して突っ込む．

1.2 画像認識

炎上ボールの認識には配布された Raspberry Pi NoIR Camera V2（以下，カメラモジュール）を使用する．画像撮影から最も近い（と思われる）ボールへのベクトルを出力する一連の手順を，簡単に以下に示す．

画像取得

カメラモジュールへのアクセスには既成ライブラリ [1] を利用した．撮影によりピクセル値の二次元配列（`cv::Mat`）が出力として得られる．画像サイズの大小が及ぼす処理時間への影響は極めて大きい，それが画像処理結果の精度に与える影響は未検証であるため，現在は適当な値が設定されている．

カラーモデル変換

得られた画像のカラーモデルを RGB から HSV に変更する．

2 値画像化

HSV 画像データに対し赤色マスクをかけて二値画像に変換する．赤色マスクの範囲は適当．

ノイズ除去

適当に膨張・縮小を繰り返してノイズを除去する．ちゃんとやってる処理じゃねえから現状ほとんど意味がねえ．

構造解析 2 値画像中の輪郭線を検出した後，それを矩形で囲む．囲んだ矩形を縦横比で解析し，ボールの縦横比に対して $\pm 20\%$ 以上の差があるものを除外する．残った矩形の重心点を求めて，適当にベクトルを作る．

参考文献

- [1] "RaspiCam: C++ API for using Raspberry camera with/without OpenCV" ,
" <https://www.uco.es/investiga/grupos/ava/node/40>" , 2017 年 5 月 31 日最終確認 .