

11 動体検知コントロールカーの作成

14 番 菅原僚大

指導教員 飯坂ちひろ

1. 目的

Raspberry Pi を利用して無線 LAN で通信を行うコントロールカーを完成させる。

卒業研究を通してネットワークや Linux、ハードウェア制御などに関する知識を深める。

2. 研究概要

2.1 研究内容

テーマ発表時にはリモート接続で PC 画面から手動制御を行う物を計画していた。OpenCV を利用して物体との距離が測れることを知り、手動制御だけでなく動体を検知して自動制御も行える物の作成に切り替えることにした。

2.2 開発環境

表 1 開発環境

使用機器	Raspberry Pi
使用 OS	Raspbian wheezy
開発言語	C++
主な使用ライブラリ	OpenCV, raspicam

2.3 OpenCV について

OpenCV はコンピュータで画像や動画の処理を行うために必要な様々な機能が実装されたオープンソース・ライブラリである。

OpenCV を利用することにより顔認識や色認識、動体検知といった処理を行うことができる。

また、画像に文字や図形の描画も行えるため認識した物の数や値などを表示させることも可能である。

今回の研究では物体との距離を測定し、その値によるモータの制御と値の表示を行うことにした。

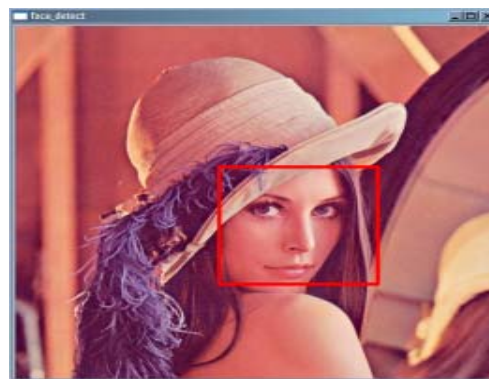


図 1 OpenCV による顔認識の処理例

2.4 カメラモジュールについて

カメラモジュールは Raspberry Pi 専用に開発された Raspberry Pi Camera Board を使用する。2592 × 1944 の解像度にまで対応し、動画や静止画の撮影が可能になっている。



図 2 カメラテスト

カメラモジュールは Raspbian のターミナル上からコマンドによって撮影が可能になっているが OpenCV でリアルタイムに処理することができない。

そのため、OpenCV で処理ができるように最適化されたライブラリ raspicam を使用して画像処理を可能にしている。

2.5 画像処理の手順

OpenCV を利用した画像処理は以下の手順で行う。

- ① カメラモジュールから取得した画像データを格納する配列を準備し、画像を取得する。
- ② 配列の内容を1つずつ参照していくため、処理時間がかかる。そのため必要に応じて画像データのサイズを縮小する。
- ③ 縮小した画像データにマスク処理を施し、色認識や顔認識がしやすいように最適化を行う。
- ④ 最適化した画像をもとに参照を行い、顔や色を抽出する。
- ⑤ 検出した部分を表示するために座標を計算し文字や図形として元画像に足す。
- ⑥ 表示画像として作成した画像データをウィンドウで表示する。

またプログラムは以下になる。

```
while(true){
    Camera.grab();                //カメラ設定
    Camera.retrieve(CapImage);    //画像の取得
    putText(CapImage,"raspberr",Point(150,200),FONT_HERSHEY_SIMPLEX,1.0,Scalar(0,0,255),1,CV_AA);
                                   //画像に文字を表示
    imshow("CaptureTest",CapImage); //画面に表示
    waitKey(100);                //表示仕様上必要
}
```



図3 画像処理テスト

今後は、センサから取得した数値や物体との距離を計算し表示させる機能を実装させる予定である。

3. 進捗状況

はじめに無線 LAN の設定を行い、SSH 制御を有効にして遠隔での制御を可能とした。

カメラの動作確認を行い処理速度を調べ、CPU への負荷と最適なフレームレートを調べた。

次に UI を作成するためにカメラ映像の表示方法について調べ、プログラムが分散しないように OpenCV を使用することにした。

OpenCV のパッケージと C++ プログラムをコンパイルするためのパッケージ Cmake を導入し、処理速度と CPU への負荷を調べた。

調べた結果から画像サイズを 360×240 に決定し、画像処理の最適化と必要な機能のアルゴリズムについて調べた。

その後、処理速度と CPU への負荷が最適化されるようにプログラムの改善を始めた。

4. 今後の作業計画

今後の作業は以下のように計画している。

表2 作業計画

12月～ 1月上旬	画像処理の最適化, センサ制御の実装
1月中旬	本体の作成, 走行テスト
1月下旬	制御の細かい仕様変更
2月	全体の処理の最適化

5. 参考文献

- ・CQ 社出版 インターフェース 2014 年 1 月号
- ・CQ 社出版 ラズベリーパイで I/O