

## ROSを用いた移動ロボットの自動運転

システム科学技術学部電子情報システム学科

1年 高井 航太

指導教員 システム科学技術学部電子情報システム学科

准教授 猿田 和樹

助教 寺田 裕樹

指導補助 システム科学技術研究科 電子情報システム学専攻

1年 池田 光汰

1年 富樫 大介

### 1. 目的

近年、高齢者の不注意事故が多発しており、大きな社会問題となっている。このような背景もあり自動運転技術による自動車が求められつつあり、そう遠くない将来自動運転車が世に誕生するといわれている。

そこで、本研究では移動ロボットキットを使用し、自動運転についての基本的な仕組みについて学び、自動運転技術の課題やその解決方法を明らかにすることを目的とする。

### 2. 使用機器

本研究で使用する移動ロボットには、Dexter Industries 社より発売された GoPiGo Beginner Starter Kit (以下、GoPiGo3) である。このキットには以下が含まれている。

- GoPiGo3 ボード (制御用基板)
- シャーシ (フレーム、ホイール、ハードウェア)
- モーター
- エンコーダ
- 電源バッテリーパックとケーブル (バッテリー別売)
- マイコン (Raspberry Pi Foundation 製の Raspberry Pi 3)

また、GoPiGo3 には Web カメラを接続して動画像を取得し、車両モデルを検知する。Web カメラには Logicool 社製の HD Webcam C270 を使用する。

### 3. 研究内容

- ① GoPiGo3 を用いて移動ロボットを製作する。
- ② 制御プログラム (前進・後退・停止・転回) を作成・実装し、GoPiGo3 の走行を制御する。
- ③ Web カメラより取得する画像から、車両を検知できるプログラムを作成し GoPiGo3 に実装し、車両検知実験を行う。なお、車両検知には、画像処理ライブラリ OpenCV を用いる。OpenCV は動画像処理に必要な機能を実現するライブラリである。

## 4. 研究結果

### 4.1. 移動ロボットの製作

本研究で移動ロボットとして用いる GoPiGo3 を製作した. 図 1 に GoPiGo3 の組み立て後の外観を示す. また, GoPiGo3 に接続した Web カメラを図 2 に示す. Web カメラは車両検知に使用する.

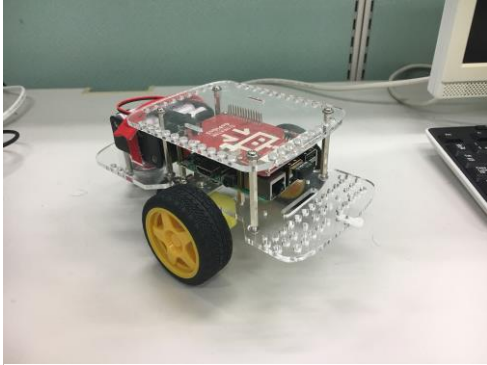


図 1 GoPiGo3



図 2 Web カメラ

### 4.2. 走行制御

GoPiGo3 の走行制御プログラムを作成・実装した. プログラムのフローチャートを図 3 に示す.

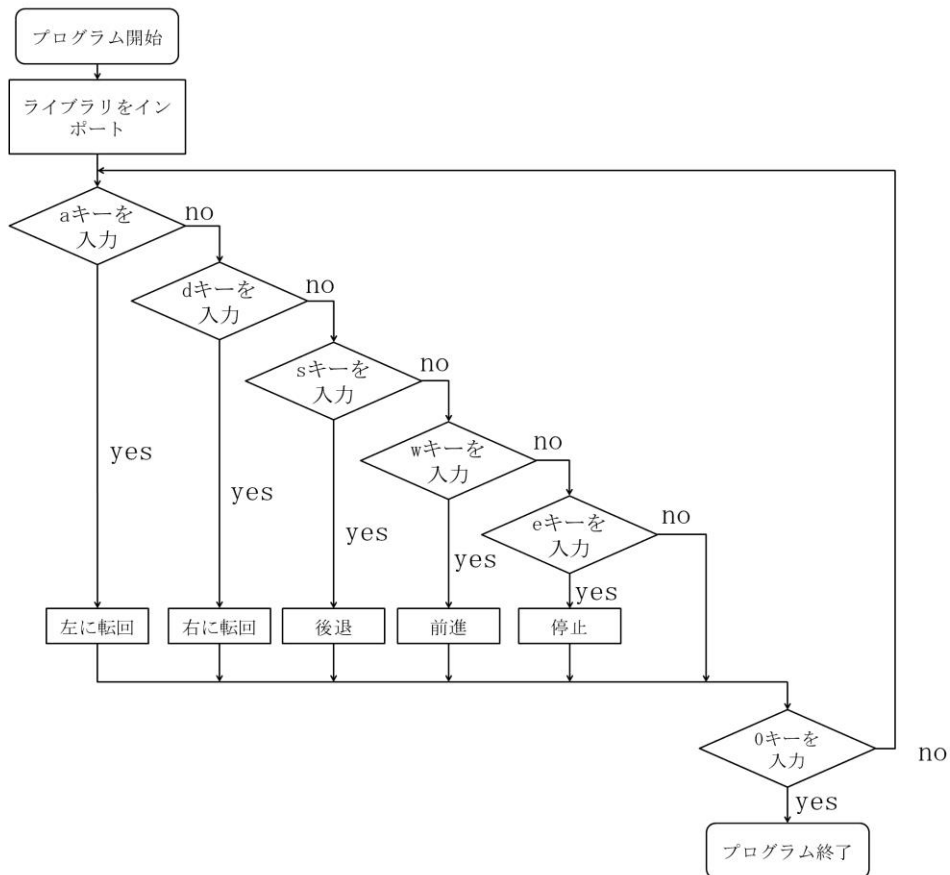


図 3 走行制御プログラムのフローチャート

このプログラムでは、GoPiGo3 をキー入力により制御する。最初に easygopigo3 というライブラリをインポートする。その後、a・s・d・w・eを入力するとそれぞれのキーに対応した動作をする。さらに0キーを入力することでプログラム全体が終了する。

実験により、前進・後退・停止・左右の転回の走行制御が可能であることを確認した。

#### 4.3. 車両検知実験

Web カメラから取得した画像に対して、OpenCV を用いて車両を検知するプログラムを作成・実装した。実装したプログラムでは文献[1]で作成された識別器とデータベースを基に車両モデルを検知する。検知プログラムのフローチャートを図4に示す。

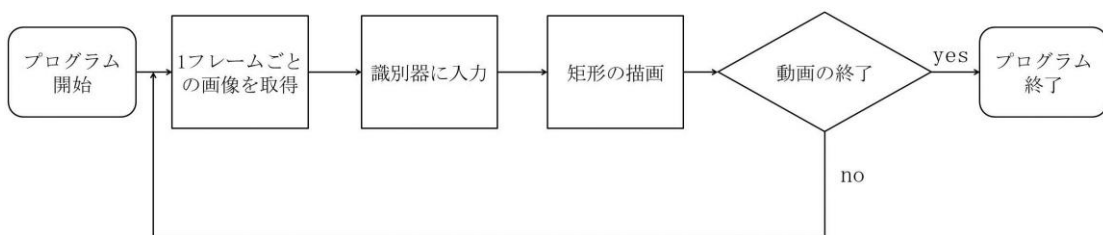


図4 車両検知プログラムのフローチャート

実装したプログラムを用いて車両検知実験を行った。実験には実際の車両ではなく、モデルカーを使用した。実験結果の一例を図5に示す。図中の赤枠が車両として検知された領域である。

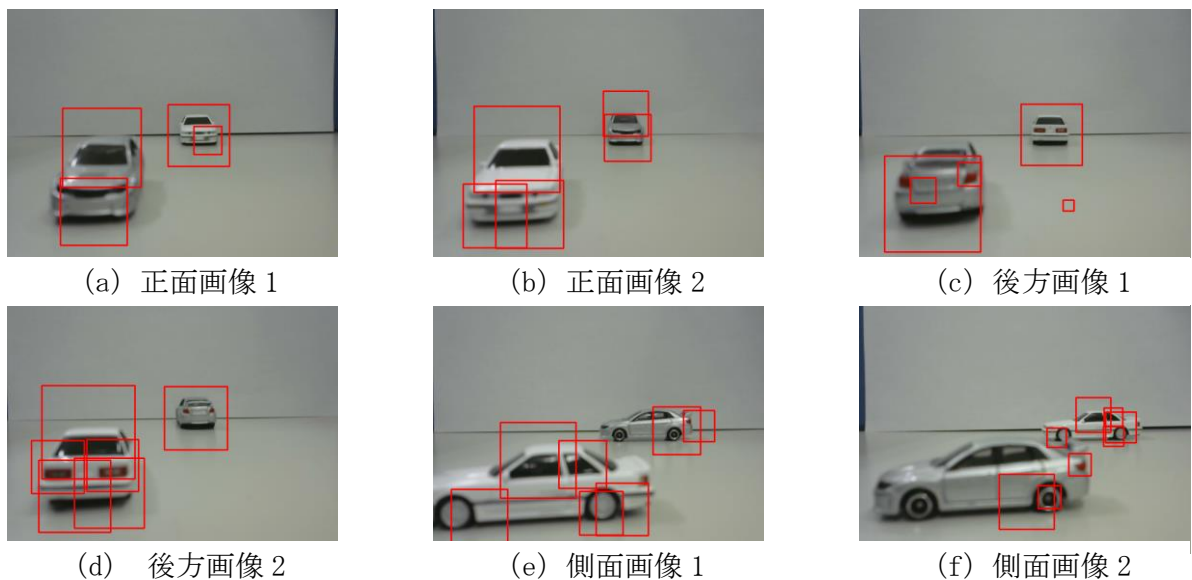


図5 車両検知結果

図5 (a) ~ (d) に示すように、車両の正面および後方から撮影した画像に対しては良好な検知結果が得られた。しかし、図5 (e) および (f) のように、側面から撮影した場合は未検知や誤検知が比較的多く発生した。

さらに、車両が近いほど複数の領域で車両として検知され、遠いほど少ない領域で正確に検知できること、車両は白より灰色のほうが検知しやすいことが傾向として見られた。

## 5. 考察

撮影方向の違いや車両までの距離により検知性能が異なる原因については、識別器の学習サンプルの影響が大きいと考えられる。また、車両の色の違いに対しては、背景と同系色であるかどうかの原因ではないかと考えられる。

一方で、実装したプログラムでは車両検知における遅延が大きく、GoPiGo3 をリアルタイムで自動制御するのが困難であった。遅延の原因は、接続した Web カメラの解像度に対してマイコンのスペックが不足していると考えられる。また、図 6 のような大きな領域も検知対象として処理していることも負荷の原因と挙げられる。

本研究では、自動運転制御の実現には至らなかったが、上記の課題を解決することで、GoPiGo3 が車両を検知したら迂回して走行する等の、自動運転が可能になると考えられる。

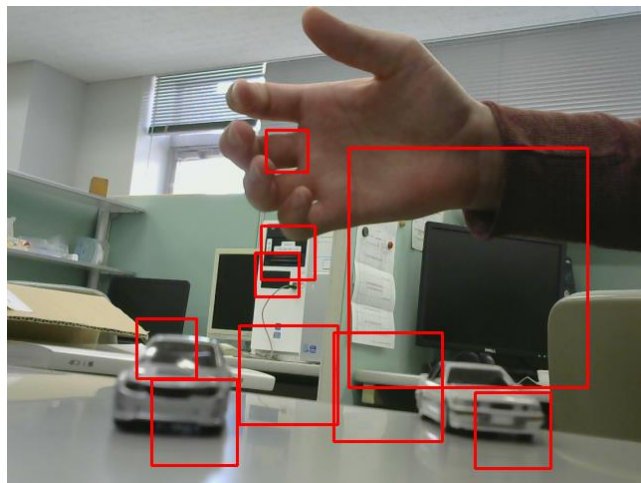


図 6 誤検知の例

## 6. まとめ

本研究では、GoPiGo3 を用いて移動ロボットを製作し、基本的な走行制御、および自動車を検知するプログラムの実装した。また、実験により走行制御と車両検知が可能であることを確認できた。

今後の課題として、リアルタイムで車両検知が可能なプログラムを実装し、検知結果に基づき車両を制御するプログラムを組み込み、自動運転を実現することが挙げられる。

## 参考文献

- [1] Andrews Sobral, “Vehicle Detection, Tracking and Counting on Behance”, <https://www.behance.net/gallery/4057777/Vehicle-Detection-Tracking-and-Counting>, 2018/3/17
- [2] GitHub, “vehicle\_detection\_harcascades”, [https://github.com/andrewssobral/vehicle\\_detection\\_harcascades](https://github.com/andrewssobral/vehicle_detection_harcascades), 2018/3/17