

赤外線センサとカメラセンサを用いた車線維持と追従走行

2013SE238 渡辺悠介 2013SE263 吉田翔

指導教員：大石泰章

1 はじめに

近年の自動車産業では、安全性や利便性を追求し自動運転を可能にしてきている。例として、自動追従、自動ブレーキ、オートライト等があり研究が進められている。2018年には「高速道路での複数レーンにおける自動車線変更の技術の実用化」、2020年までに「交差点を含む一般道での自動運転技術の導入」等が計画されている[1]。ところで、実際の運転時は他車に追従することが多く、自動車事故の中に追従事故が最も大きな割合(約30%)を占めている[2]。したがって、自動追従は実現すべき重要な技術である。また、わき見運転や漫然運転による車線のはみ出しも交通事故に直結する。そのため、車線維持も欠かせない技術である。

本研究ではZumoという自動車模型[3]にArduinoというマイクロコンピュータと画像認識カメラ(Pixy)[4]を搭載し、車線維持と追従走行の実現を目指す。ここで安全性と精度をいかに高めるかが課題となる。安全性を高めるためには、赤外線センサを利用することで想定した道路白線を越えないようにすることが必要であり、精度を上げるためには、画像認識カメラセンサで認識した物体との車間距離を適切な距離に保つこと、そして停止したときの行きすぎ量を減らすことが必要である。

2 使用する自動車模型

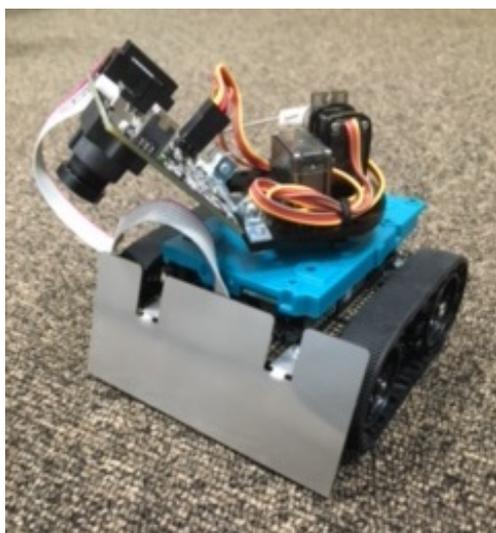


図1 カメラつき自動車模型

本研究では自動車模型(Zumo)にマイクロコンピュータ(Arduino)と画像認識カメラ(Pixy)を搭載し、制御を行う。使用する自動車模型を図1に示す。Pixyとはレンズ視野が水平75度・垂直47度のカメラで、最大7色の色を

記憶させることが可能である[5]。対象物体はPixyの中において長方形で認識されるので、対象物体の大きさと形が変わらなければ長方形の面積から対象物体とPixyの距離を求めることが可能である。本研究のPixyはpan/tiltキットを搭載している。pan/tiltキットとは、Pixyを横方向及び縦方向に機敏に動かすことができるキットである。これによりPixyの物体感知能力が向上する。また、Zumoには赤外線センサが搭載されている。このセンサはZumoの車体の裏に6つ内蔵されており、それぞれが床の色の明るさを数値化して読み取る。これを応用して、床に書かれた道路白線を越えないようにすることができる。また、床の色に応じて減速したり、一時停止することも可能である。

3 車線維持

車線維持を実現するため道路を想定したコースで自動車模型を走行させ、道路白線を越えないようにするプログラムを作成し実装する。Zumoに内蔵されている赤外線センサは0から2000までの値を出力し、この値の大小で色を識別する。本研究では6つあるセンサの値を読み取ることで、車線維持を行う[6]。図2は6つのセンサを示す。6つのセンサを左から順にaからfと呼ぶことにし、以下センサa、センサbのように記す。

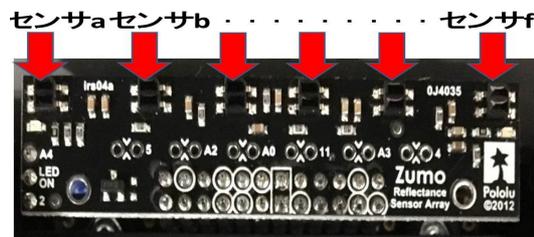


図2 赤外線センサ

3.1 ON/OFF 制御を用いた車線維持

路面が黒で、境界に白線を引いたコース上に自動車模型を走行させる。また、コース上には路面を赤くした領域と黄色くした領域があり、赤い領域では一時停止、黄色い領域で減速走行を行うことにする。それ以外の場合では10[cm/s]で前進させる。自動車模型に搭載されているセンサが左右の白線を感知することで車線維持を行う。右のセンサfが白線を感知した時はその白線に沿うように左に曲がり、同様に左のセンサaが白線を感知した時はその白線に沿うように右に曲がるようにする。aとfのセンサは、同じ色に対しても出力値に差があるのでそれを反映した命令をする必要がある。例えばセンサfの場合、

292 以下の出力が「白」を意味するが、センサ a の場合、328 以下の出力が「白」を意味する。センサ制御のプログラムにおけるセンサ出力と自動車模型の動作との対応関係を表 1 に示す。表中の○はセンサが白色を認識したことを示し、●はセンサが黒色を認識したことを示す。

表 1 センサ出力とそれに対応する動作

センサ a	センサ f	動作
●	●	前進
○	●	右曲がり
●	○	左曲がり

また、センサ b とセンサ e で黄色の値を認識したときは 5[cm/s] で減速走行を行う。センサ c とセンサ d で赤色を認識したときは一時停止を行う。一つのセンサで複数の色を認識させることも可能だが、誤作動を減らすため、黄色の認識をセンサ b, e で、赤色の認識をセンサ c, d で担当させ役割を分担させる。黄色の認識においてはセンサ b, e の出力の平均をとり、赤色の認識においてもセンサ c, d の出力の平均をとることで、精度の向上を図る。以上により、白線に沿って走行するとともに、黄色の領域では減速走行、赤色の領域では一時停止を行うと考えられる。

3.2 実験結果

以上をプログラム化して自動車模型に実装したところ、左右どちらかのセンサが白線を認識した際、白線に沿うように走行を行い、黄色い領域では減速走行を行い、赤い領域では一時停止を行うことを確認した。これにより車線維持に成功したといえる。

4 追従走行

4.1 追従走行の実現

カメラ (Pixy) を使って追従走行を実現する [7]。Pixy にはあらかじめ認識させたい色を覚えさせる。今回は赤色のボールを認識させる。Pixy の認識するボールの面積を常に一定に保つことで、自動車模型がボールとの距離を一定に保ちながら走行するようにする。また、Pixy には pan/tilt キットが搭載されており、Pixy 自身も 2 次元的に動くことができる。ただし、動くことのできる範囲はカメラを中心に横方向 (pan) は 180 度、縦方向 (tilt) は 150 度である。

追従走行のフローチャートを図 3 に示す。図 3 中の「カメラを物体の中心に動かす」はカメラの中心と物体の中心との誤差を零にするべく、PD 制御によってカメラの向きを変えることを示す。また、「追従制御」はカメラとの距離が 20[cm] 以外であれば 20[cm] に収束させることを示す。「追従制御」では、まず Pixy が読み取ったボールの横幅と縦幅の画素数を掛け面積を求める。20[cm] の距離に対応する面積とその求めた面積との差に適当なゲイン

をかけた値を a とし、これを左右の車輪の速度の平均とする。また、ボールに向かって走行するよう左右の車輪の速度に差をつける必要がある。カメラはボールの方向を向いているので、カメラの向きと Zumo の正面方向との差に適切なゲインを掛けた値を b とする。求めた a と b に基づき、左車輪の速度を $a + b$ で、右車輪の速度を $a - b$ で定める。

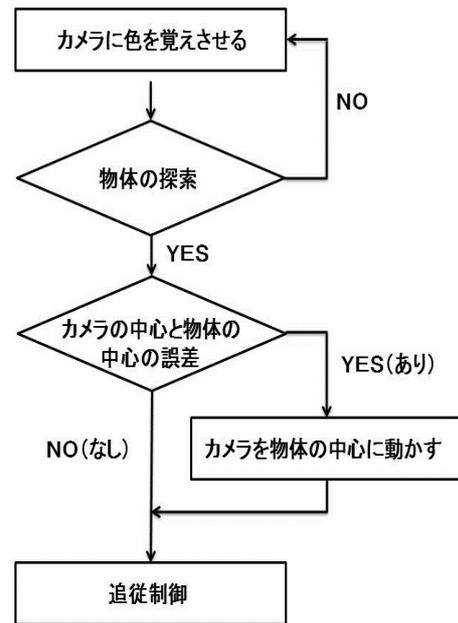


図 3 追従走行のフローチャート

4.2 距離と面積の関係

Pixy は対象物を認識する際に二次元の画像を用いる。奥行きを感じることができないため対象物との距離を面積の大きさに判断し処理を行う。ボールとの距離を測定するために距離と面積の関係を求める。図 4 は、ピンホールカメラモデルである [8]。

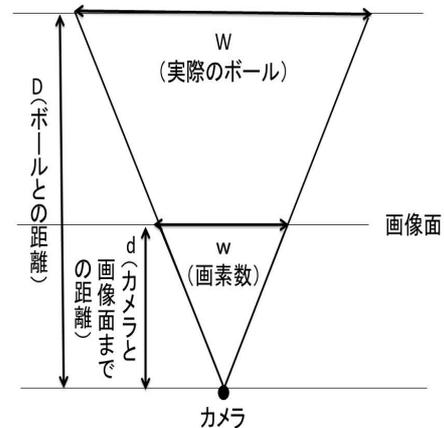


図 4 読み取る画素と距離のモデル

図中の W (実際のボールの大きさ) と d (カメラと画像面までの距離) は一定である。また、カメラと W で表される三角形とカメラと w で表される三角形は互いに相似の関係であるため、(1) のような比例式が得られる：

$$W : D = w : d. \quad (1)$$

これより次が従う：

$$Dw = Wd. \quad (2)$$

W と d は一定であるから、 D と w が反比例の関係にあることがわかる。

表 2 ボールまでの距離と面積の関係

距離 [cm]	横幅	縦幅	面積 (横幅×縦幅)
5	238	188	44744
10	120	120	14400
15	81	81	6561
20	61	61	3721
25	49	49	2401

表 2 は赤い色のボールを実際に Pixy に写したときのボールまでの距離と画素数の関係を表している。この結果からも距離 D が小さくなれば、画素数 w が大きくなることがわかる。ここで式 (1) と $D = 10$ [cm], $w = 120$ [ピクセル] を用いると次式が得られる：

$$Dw = 120 \times 10. \quad (3)$$

これより

$$D = \frac{1200}{w} \quad (4)$$

である。

式 (4) を使うことにより、横幅の画素数 w から、ボールと車体の車間距離 D を算出することが可能である。なお、この定数値 1200 は用いる実験機によって値が多少異なる。

4.3 ボール静止時の実験結果

赤いボールを Pixy に認識させ、そのボールに追従する制御を行う。追従走行はボールの中心とカメラとの距離を 20[cm] に収束させることで行う。図 5 は自動車模型から約 30[cm] 離れた場所にボールを静止させ、Pixy がボールを認識してからボールとの距離が 20[cm] に収束するまでの距離の変化を示したグラフである。距離は式 (4) を使ってボールの横幅の画素数から算出した。約 1.6[s] で物体を認識し、約 2.0[s] で目標距離 20[cm] に収束した。行き過ぎ量に着目しても約 1[cm] に収まっている。なお、図 5 ではボールとの距離の初期値は約 29[cm] と読めるが、配置は手動で行ったため、誤差が生じたものと考えられる。

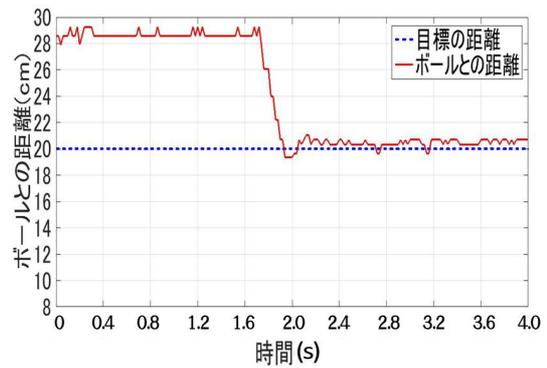


図 5 ボールを 30[cm] の距離に置いた場合

図 6 も同様に自動車模型から約 10[cm] 離れた場所にボールを静止させた場合のボールとの距離の変化のグラフである。約 1.2[s] で物体を認識し、約 1.8[s] で目標の距離に収束していることがわかる。また、行き過ぎ量に着目しても約 1[cm] に収まっている。なお、図 6 も初期値は約 9.5[cm] と読めるが、配置は手動で行ったため、誤差が生じたものと考えられる。

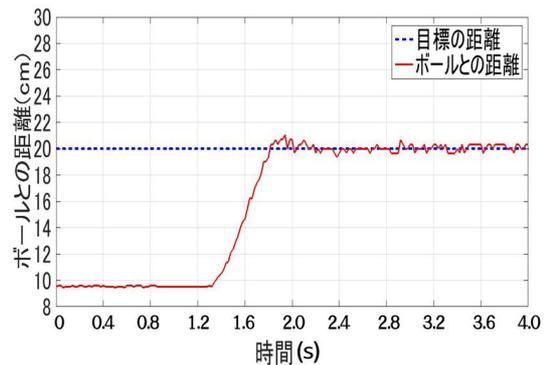


図 6 ボールを 10[cm] の距離に置いた場合

4.4 ボール運動時の実験結果

別の Zumo に赤いボールを乗せて走行させ、これに追従させる実験を行う。まず、Pixy に赤いボールを認識させ、ボールとの距離が安定した状態からスタートさせた。先行するボールつき Zumo(以下、先行車)をまっすぐ走行させ、その後ろを制御則を実装したカメラつき Zumo(以下、実験車)が衝突しないように追従することを目標とする。なお、今回の実験では 4.2, 4.3 節の実験とは別の Pixy つき Zumo の自動車模型を用いたため、式 (4) の定数 1200 の値が変わり、1275 を定数として用いることで画素数から距離の算出を行っている。図 7 は、先行車を 10[cm/s] の速度で約 4[s] 間直進させた後停止させ、それを実験車が追従して収束するまでのボールとの距離の変化のグラフである。10[cm/s] で先行車を直進させた場合、実験車が動いてから停止するまでに車間距離が最大 22[cm]、最小 21[cm] となっている。先行車が停止してからの距離の変

化に注目しても、衝突することなく目標距離の 20[cm] に近い値のまま収束をした。

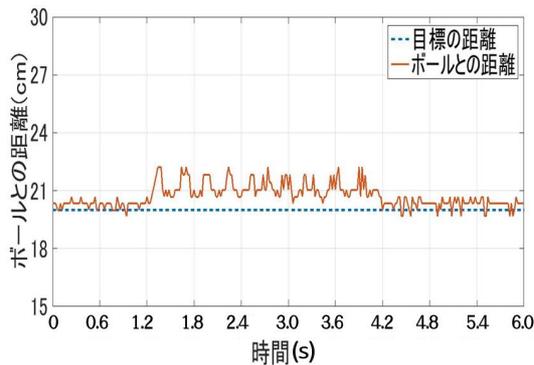


図 7 10[cm/s] の速度の先行車に追従させた場合

図 8 には先行車を 60[cm/s] という先の実験よりも速い速度で約 1.0[s] 間直進させて停止させたとき、実験車がそれを追従して収束するまでのボールとの距離の変化のグラフである。60[cm/s] で先行車を直進させた場合、実験車が動いてから停止するまでに車間距離が最大約 47.5[cm]、最小約 19[cm] となっている。ボールつき Zumo が停止してからの距離の変化に注目してみると、衝突することなく目標距離の 20[cm] に近い値に収束している。先行車が速い速度で直進しているときも、ボールとの距離がこの程度に収まっているのは性能がいい追従といえる。

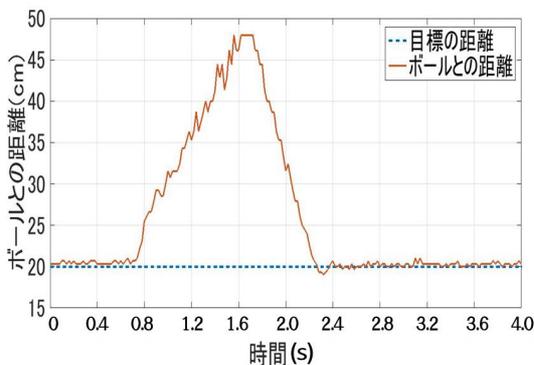


図 8 60[cm/s] の速度の先行車に追従させた場合

4.5 赤外線センサを用いた追従走行

3章で行った赤外線センサを用いて、車線維持と追従走行を組み合わせたプログラムを作成し実行した。その結果、追従走行中であっても白線を検出した際、車線維持を行い白線を越えることはなかった。また、黄色の領域では、カメラはボールを捕らえつつも、先行車に追従することなく減速走行を行った。また、赤色の領域では、一時停止を行った後、再び先行車に追従した。

5 おわりに

本研究では、道路を想定したコース上で自動車模型を走行させ、白線を越えないように車線維持を行う制御、そして、赤いボールを車に搭載し、その後ろを自動車模型が走行する自動追従走行制御の2つを行った。車線維持制御では、まず自動車模型 (Zumo) の赤外線センサの値を分析し、ON/OFF 制御によってプログラムを作成した。また、赤い領域と黄色い領域を加えることで一時停止や減速走行を可能にした。自動追従走行では、自動車模型 (Zumo) に搭載されたカメラ (Pixy) を用い、赤いボールを前の車に搭載し、これに追従する制御を行った。その際、ボールと Zumo との距離を 20[cm] に保つことで衝突しない追従を可能にした。

参考文献

- [1] 河口まなぶ：『ナビを使って目的地まで自動運転する日産の最新型実験車両公開』
<http://bylines.news.yahoo.co.jp/kawaguchimanabu/20151023-00050754/>
- [2] 交通統計 H13 年版, 交通事故総合分析センター (2002)
- [3] Pololu Robotics Electronics
<http://www.pololu.com/product/2510>
- [4] CMUcam5 Pixy Quick start
<http://www.charmedlabs.com/pixystart>
- [5] Pixy CMUcam5 イメージセンサー
<http://www.robotshop.com/jp/ja/pixy-cmucam5-image-sensor.html>
- [6] 鈴木美朗士：『Arduino でロボット工作をたのしもう!』, 秀和システム, 東京, 2014.
- [7] GitBook-Arduino docs
<http://www.gitbook.com/book/fagkura/arduino-docs/details>
- [8] 出口光一郎：『ロボットビジョンの基礎』, コロナ社, 東京, 2000.