

# コンピュータ理工学特別研究報告書

## 題目

Raspberry Pi を用いた模型自動車の制御  
-車線検出とライントレース-

学生証番号 1245252

氏名 山崎 隆誠

提出日 平成 28 年 2 月 1 日

指導教員 蚊野 浩

京都産業大学  
コンピュータ理工学部

## 要約

自動運転は、路面状態や周囲の環境を正しく認識し、それに応じて自動車を適切に制御することで実現されている。このシステムを構成する要素の中で、カメラで自車の周囲を観察し、撮影した画像を処理することで周囲状況を認識する技術は重要な役割を果たす。特に、道路上にペイントされた車線の認識は自動運転を実現する上で重要な技術であり、車載カメラは車線認識に最も適した装置である。

本研究では、Raspberry Pi とカメラモジュール、模型自動車を組み合わせ、自動運転を模擬する車とした。模型自動車に搭載したカメラを使って前方を観察し、車線を自動的に検出する。そして、検出した車線を自動的にトレースするように模型自動車を制御した。実際の道路の車線はレーンの境界を示す破線でペイントされていることが多い。しかし、この実験では、レーン中央を示す一本の実線を車線とし、車線検出とライントレースの技術を基本から習得した。

本研究で開発した車線検出アルゴリズムは以下の手順で処理を進める。

- (1) 画面下部から車線候補点を抽出する。
- (2) 車線候補点からハフ変換によって車線候補線を抽出する。
- (3) 最も有力な車線候補線を、画面左端・右端・上端のいずれかに達するまで、エッジ点を探索しながら追跡する。

この基本となるアルゴリズムに、

- (a) 画面下端に車線以外のエッジが複数存在する場合、
- (b) 車線の湾曲が大きい場合、
- (c) 道路を想定した紙の表面で光が鏡面反射する場合、

の 3 つの誤作動を起こす原因に対策を施すことで、車線検出が誤作動する可能性を軽減した。

模型自動車の制御は、車線検出で得た情報に基づいて車線の状態を 5 つに分類し、状態遷移を繰り返すことで進行方向を制御した。しかし、状態が移り変わる境目付近に走路が検出された場合、頻繁に状態遷移を繰り返し、結果として不自然な動作になることがあった。

以上のように車線検出と模型自動車の制御を組み合わせることで、自動運転に利用するコンピュータ技術の基本を開発することができた。

## 目次

|                            |          |
|----------------------------|----------|
| 1 章 序論                     | ・ ・ ・ 1  |
| 2 章 Raspberry Pi を用いた模型自動車 | ・ ・ ・ 3  |
| 2.1 模型自動車のシステム構成           | ・ ・ ・ 3  |
| 2.2 ソフトウェアの開発環境            | ・ ・ ・ 7  |
| 3 章 車線の検出と模型自動車の制御         | ・ ・ ・ 8  |
| 3.1 車線検出の従来技術              | ・ ・ ・ 8  |
| 3.2 車線を検出するための提案手法         | ・ ・ ・ 8  |
| 3.3 模型自動車の制御手法             | ・ ・ ・ 12 |
| 4 章 実験結果と考察                | ・ ・ ・ 16 |
| 4.1 車線検出に関する実験結果           | ・ ・ ・ 16 |
| 4.2 模型自動車の制御に関する実験結果       | ・ ・ ・ 21 |
| 4.3 考察                     | ・ ・ ・ 23 |
| 5 章 結論                     | ・ ・ ・ 24 |
| 参考文献                       | ・ ・ ・ 25 |
| 謝辞                         | ・ ・ ・ 25 |
| 付録                         | ・ ・ ・ 26 |

## 1 章 序論

自動車はコンピュータ技術を取り込むことで高度化してきた。代表的なものはエンジンやステアリングの制御にマイコンを利用することである。また、カーナビは経路の案内情報を提供することに特化した専用コンピュータである。このように、コンピュータ技術は自動車の性能を改善することに貢献してきた。

これに対して、最近、実用化が進みつつある自動運転は、自動車産業や自動車社会に革新的な変化をもたらす可能性がある技術である。自動運転は、路面状態や周囲の車両・歩行者などの環境を認識し、認識結果に基づいて自動車を適切に制御することで実現できる。その要素技術は、環境を測定・観察するためのセンサ技術、測定・観察したデータから周囲環境を認識する情報処理技術、認識結果に基づいて自動車を適切に制御する技術、などである。これらの要素技術の中で、カメラを用いて自車の周囲を撮影し、その画像を処理することで周囲状況を認識する技術が大きな役割を果たしている。

自動運転に利用するカメラ的な装置には、可視画像を撮影する通常のカメラ、通常のカメラを2台組み合わせたステレオカメラ、赤外線カメラ、LIDAR とよばれる距離画像センサ、などがある。現在の自動運転技術は、これらの装置を組み合わせて周囲環境を測定・観察している。これらの中で、通常のカメラは装置が単純・安価でさまざまな用途に使える汎用性がある。現状の画像処理技術では、周囲環境を認識できる能力は限られているが、今後、技術が高度化することにより、利用範囲を広げると期待されている。本論文では、可視カメラを用いた単眼の車載カメラ技術を扱う。

自動運転における車載カメラの用途は次のものである。

- (1) 車線や走路の認識。
- (2) 周囲の車両や歩行者など障害物の認識。
- (3) 信号や道路標識の認識。

これらの中で、車線認識は車載カメラに最も適したものである。他のカメラ装置では実現が難しい。障害物の検出は車載カメラでも可能であるが、ステレオカメラと距離画像センサがより適している。信号や道路標識の認識も車載カメラを用いることが考えられるが、それだけで確実に認識することは困難である。これらは地図情報に組み込んでおくであるとか、近距離通信技術を用いて車両に伝えることが有力な方法である。

車線認識技術は、通常運転中の車線逸脱に対して警告を出す車線逸脱防止支



援システム (LDW: Lane Departure Warning) として実用化されている。これは、高速道路や車線変更が必要無い 1 車線道路における居眠り運転などを防止するためのシステムである。車両の前方を撮影するように設置した車載カメラで道路路面を撮影し、常時、車線を認識する。そして、車両が車線を越えた場合に警告を出す。現在の車線認識技術の利用は、このように運転者に対して情報を提供するだけのものである。自動運転においては、認識した車線情報に基づいて、ハンドル・ブレーキ・アクセルを制御するようになる。

画像処理による車線認識は、道路路面を撮影した画像から車線としてペイントされた画素集合を検出し、それら进行处理することで、レーン領域の境界を特定する処理である。実際の道路では、

- (1) 車線は破線としてペイントされる。
- (2) 車線には、かすれやゴミなどによるノイズが生じる。
- (3) 太陽光や照明の影響による見え方の違いが生じる。

などがあるため、正確・確実に車線を検出するには複雑で高度な処理が必要である。

本研究の目的は、車線検出とそれに基づく自動運転の基本的な技術を自作することによって、画像処理技術、組み込みシステム技術、自動運転に利用されるコンピュータ技術の基礎を取得することである。その中でも、安定して正確に車線を検出することができる画像処理技術を開発することが、本研究の主たる目標である。

今回の研究のために実車を利用することは容易でない。そこで、Raspberry Pi にカメラモジュールを接続したものを、車載カメラと車載コンピュータに見立てた。実車のモデルとして、TAMIYA 社のバギー工作セットに自作の加工を加えたものを利用した。地面として大きな紙面を使い、紙面に車線を模擬する線を描いた。そして、紙面に描いた車線を抽出し、車線を自動的にトレースするプログラムを開発した。実際の道路の車線はレーン境界を示す破線で描かれている。しかし、本研究では車線検出をより基本から学ぶため、レーン中央を示す実線を車線として用いた。

以下、2 章では Raspberry Pi を用いた模型自動車のシステム構成と、ソフトウェアの開発環境について述べる。3 章では車線検出と模型自動車の制御について述べる。4 章では車線検出と模型自動車の制御に関する結果と考察を述べる。最後に、5 章で本研究の結論を述べる。

## 2 章 Raspberry Pi を用いた模型自動車

この章では、Raspberry Pi を用いた模型自動車のシステム構成、ソフトウェアの開発環境について説明する。

### 2.1 模型自動車のシステム構成

この実験で使う模型自動車は表 1 の要素から構成される。

表 1 模型自動車の構成要素

|          |                                                            |
|----------|------------------------------------------------------------|
| コンピュータ   | Raspberry Pi 2, OS: Raspbian                               |
| カメラ      | Raspberry Pi 専用カメラモジュール                                    |
| モータ      | 前輪ステアリング用：サーボモータ<br>後輪駆動用：DC モータ                           |
| 台車, 車輪など | タミヤ楽しい工作シリーズ No. 112 バギー工作基本セット<br>ユニバーサルプレート              |
| 電源       | モバイルバッテリー (5V, 1A) (無線 LAN を使う場合, 1.5A)<br>電池ボックス (単三×4 本) |

Raspberry Pi (図 1) は名刺サイズの小型コンピュータで、ディスプレイやネットワークを接続することで、通常のパソコンとして使用可能である。また、汎用入出力端子 (GPIO) を使って回路や装置を接続することが可能で、専用の電気機械装置を容易に試作できる。

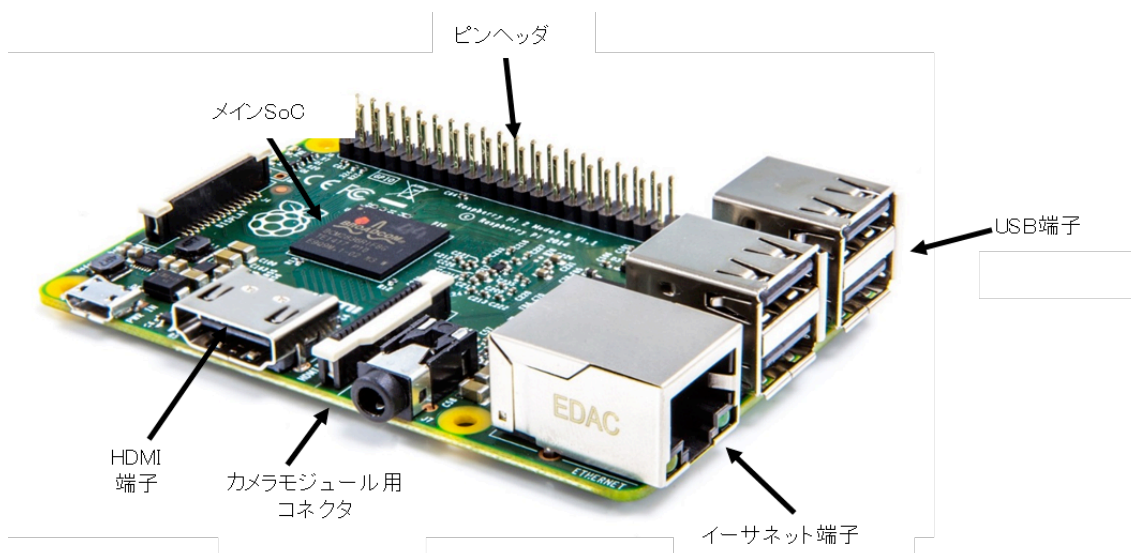


図 1 Raspberry Pi 2 の外観

カメラには Raspberry Pi 専用のカメラモジュールを用いる。これは、Raspberry Pi ボードの CSI コネクタに接続する CMOS カメラである。静止画は  $2592 \times 1944$  画素、動画は  $1920 \times 1080$  のフレームを 30FPS (Frame Per Second) で撮影できる。

サーボモータは回転軸の角度や速度が制御できるモータである。サーボモータに送る制御信号によって回転軸の角度を変化させることができるため、模型自動車のステアリングやロボットの手足の関節部分に使われる。図 2 に前輪ステアリング部の写真を示す。

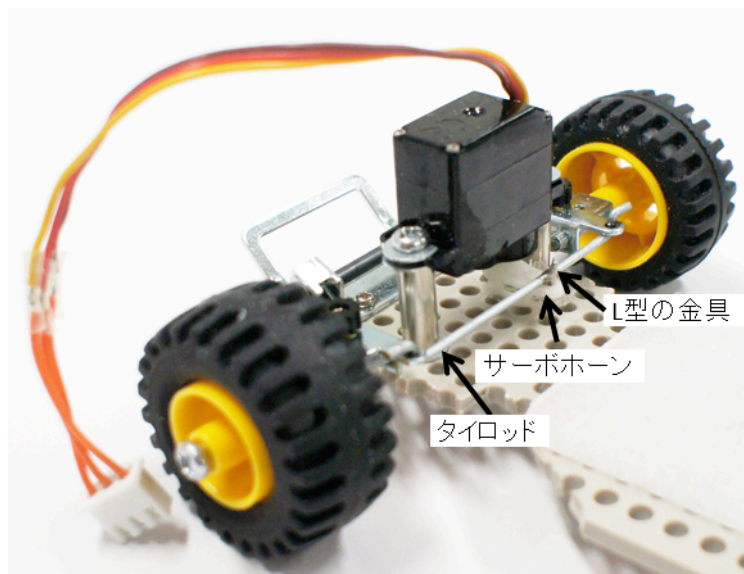


図 2 前輪ステアリング部

図 3 にサーボモータと DC モータを Raspberry Pi2 から制御して動作させるためのモータドライブ回路を示す。図 4 に模型自動車の全体を要素に分解した写真を示す。図 5 に模型自動車を構成するシステム要素の接続関係を示す。図 6 が模型自動車の外観である。

この模型自動車を、スタンドアロンで動作させる方法として、Raspberry Pi 起動直後に、ライントレースのアプリケーションを自動的に動作させる方法と、後述するリモートデスクトップを無線 LAN で利用する方法がある。リモートデスクトップを利用すると、ライントレースを行うアプリケーションの動作が少し遅くなり、無線 LAN のための電力を余分に必要とする。

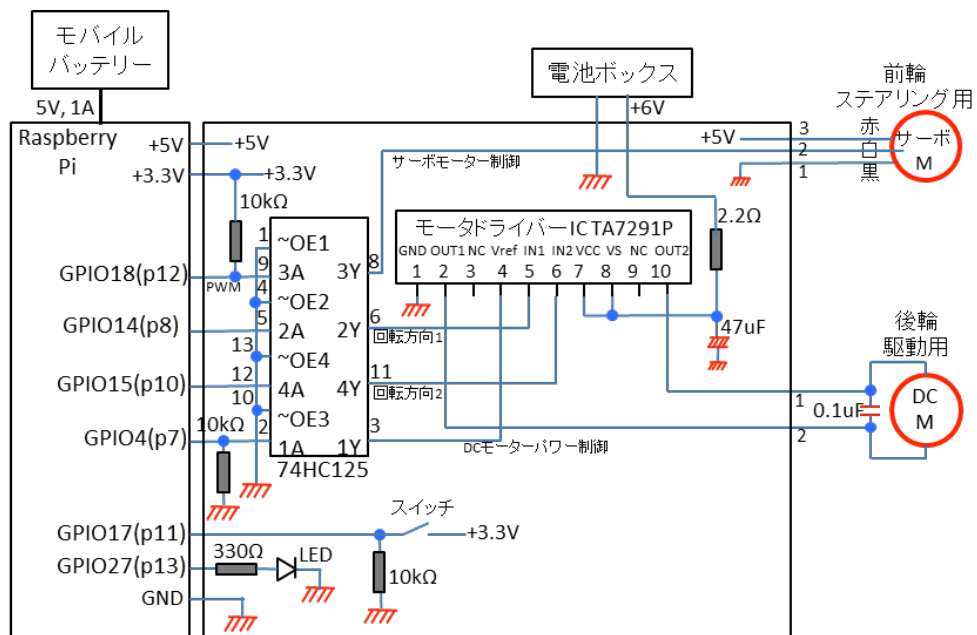


図3 モータドライブ回路

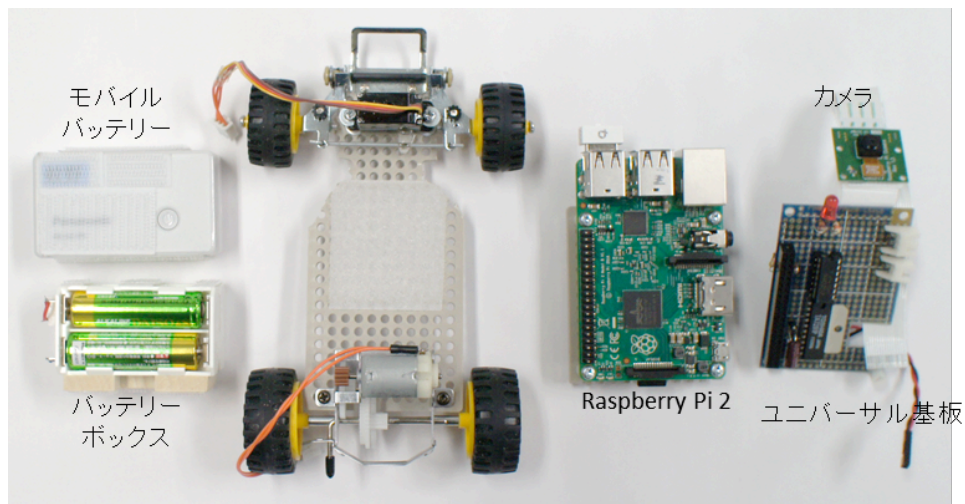


図4 コンポーネントに分解した様子

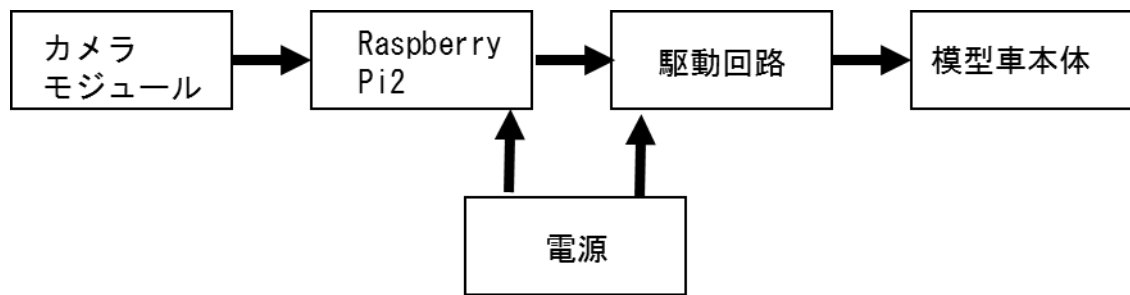


図5 模型自動車を構成する要素のブロック図

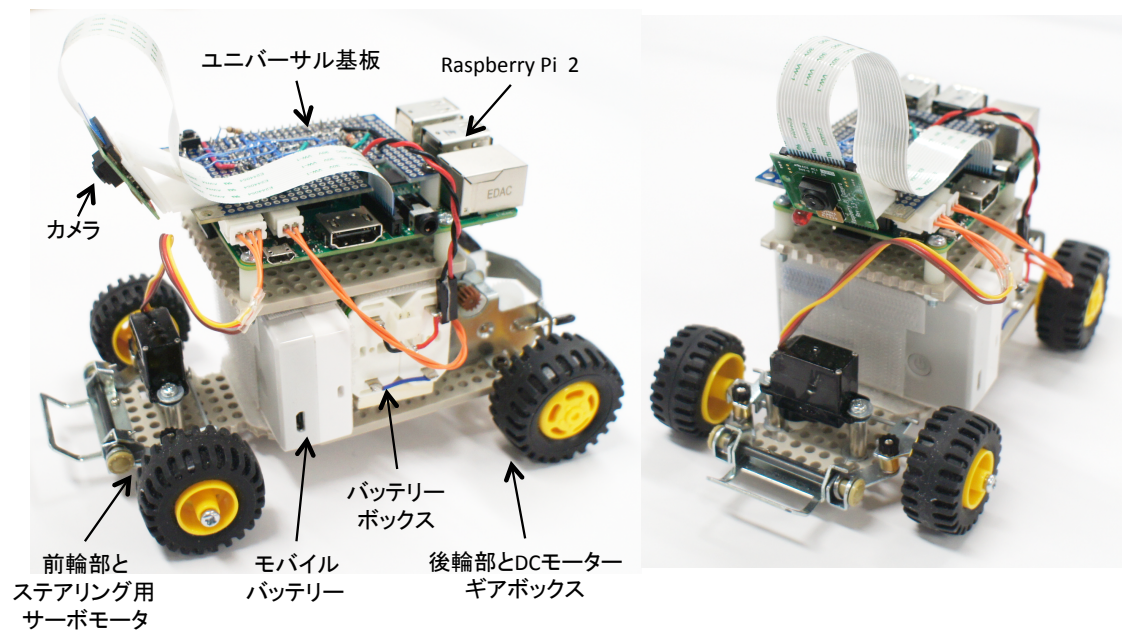


図6 模型自動車の外観と構成要素

## 2.2 ソフトウェアの開発環境

プログラムは次の環境で開発した。

- (1) プログラムはC言語を用いた。コンパイラには gcc を使用した。
- (2) カメラモジュールから映像を取り込むために、`raspicam_cv` ライブラリ[4]を使用した。
- (3) 画像処理のライブラリとして OpenCV を用いた。
- (4) DC モータとサーボモータは GPIO 端子から制御した。そのためのライブラリとして Wiring Pi[5]を用いた。
- (5) プログラムの開発は図 6 で示した模型自動車にディスプレイとマウス、キーボードを接続した状態で行った。また、Microsoft Remote Desktop というアプリケーションを使い、模型自動車を無線 LAN のクライアント、Mac を無線 LAN のサーバとし、Mac を端末とする環境でもプログラム開発を行った。後者の環境では、模型自動車を外部機器に接続するケーブルを取り除き、スタンドアロンで動作させることができる。ただし、動作は少し重くなる。

### 3 章 車線の検出と模型自動車の制御

#### 3.1 車線検出の従来技術

五十部・中村は，通常の道路面を撮影した画像に対して水平方向一次微分処理(Sobel フィルタ)を行い車線のエッジ候補点を抽出した．エッジ候補点にハフ変換を行い，車線候補線を抽出する．そこから車線ではない線を除外していくことで，車線検出を行った[1]．また，山田・田森・塩見は車線の特徴量として，空間フィルタによって得られたエッジを用い，ハフ変換によって車線候補点から車線候補線を抽出し，車線幅を指標に車線に該当しない線分を削除することで，車線検出を行った[2]．林，富沢，末廣，工藤は，入力画像を鳥瞰変換し，ノイズを取り除くためにメディアンフィルタをかけ平滑化した．その後，二重円型オペレータによって車線候補点を抽出し，最後にハフ変換を用いて車線検出を行った[3]．

このように車線検出の従来手法は，

- (1) 入力画像から画像処理演算で車線の候補点を抽出する．
  - (2) 車線候補点にハフ変換を施すことで車線候補線を抽出する．
  - (3) 車線候補線から車線の条件に該当しないものを除外する．
- という流れで構成される．

#### 3.2 車線を検出するための提案手法

本研究で開発した手法を説明する．カメラモジュールから  $320 \times 240$  画素の画像を取得する（図 7）．この画像からわかるように，本研究における車線は白い面に黒色で描かれた線である．また，通常の道路における車線はレーンの境界線であるが，本研究における車線はレーンの中央を示す線である．取得した画像の中で上側の部分には道路面以外の余分な情報を多く含んでいる．また，道路面の中で重要な部分は模型自動車に近い下側の領域である．そこで，図 8 のように画像処理を行う領域を制限した．以下の本文では，この制限した領域を処理対象の画像とする．





図7 取得した画像の例 (320×240 画素の画像)

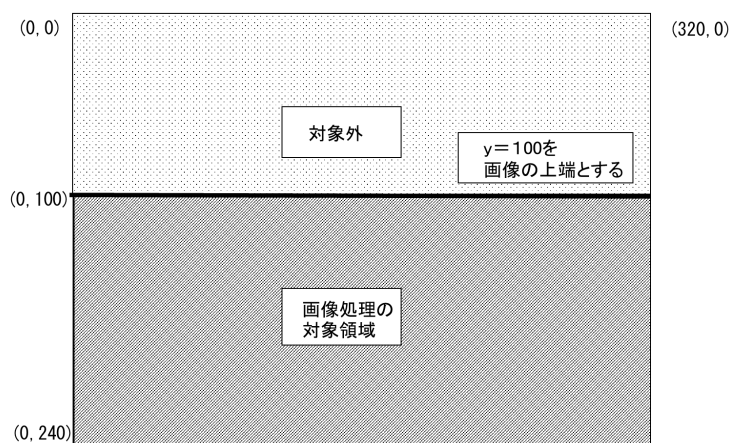


図8 処理対象領域の設定

基本的な処理の流れを示す。

- (1) カメラモジュールから得た画像(濃淡画像)に対して水平方向のソーベルフィルタをかける。縦方向のエッジだけを強調するのは、画面下端(模型自動車側)から上側に向かって、車線があると仮定したためである。
- (2) 縦方向のエッジを強調した画像の下端付近を二値化し、ハフ変換を施すことで、画面下端領域の線分を検出する。
- (3) ハフ変換で得た線分の上端から、画面上端方向に向かって左右2画素の範囲でエッジ探索を行い(図9)、エッジ点が存在した場合、それを線分と結ぶ。そして、エッジ探索の範囲を上へ1画素分移す。



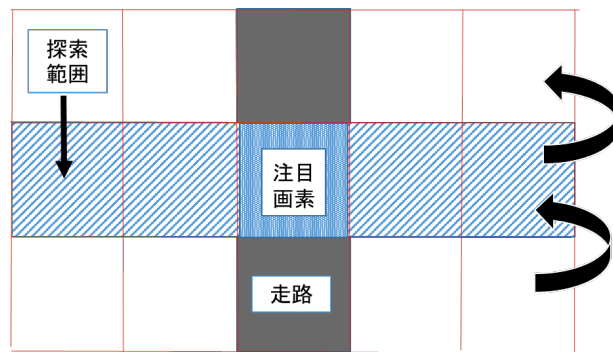


図9 エッジ探索

(4) (3)の処理を画面上端・左端・右端のいずれかに到達するまで繰り返す。以上が車線検出の基本的なアルゴリズムであるが、これだけでは、環境によっては正しく車線を検出できない場合がある。以下がそのような場合である。

- (a) 画面下端に複数の線分があり、車線以外の線分が最も顕著であるため、ハフ変換によって車線以外の線分を車線としてしまった場合 (図10)。

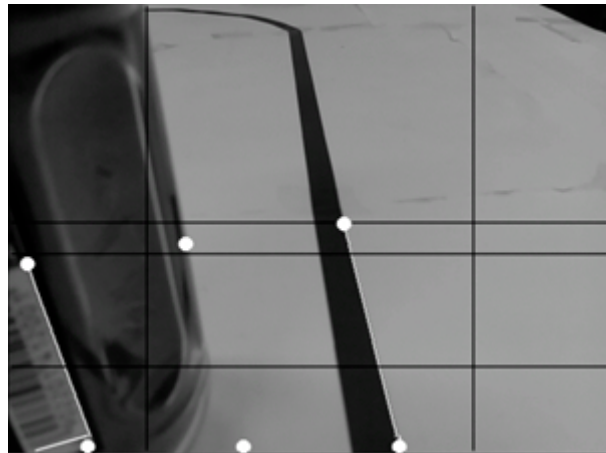


図10 車線検出に失敗する(a)の場合の例

- (b) エッジを探索する車線の湾曲が大きくなると、縦エッジが弱くなる。そのため、二値化したエッジが途切れてしまい、最後までエッジ探索を行えない場合 (図11)。

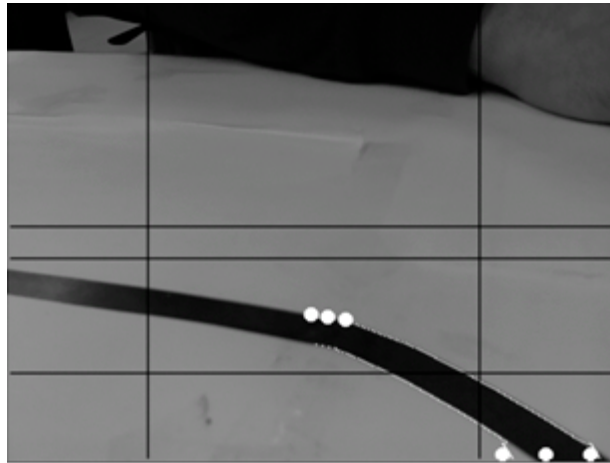


図 11 車線検出に失敗する (b) の場合の例

(c) 路面を想定した紙の表面で光が鏡面反射することが原因で起こるエッジと，車線のエッジを区別できなかった場合（図 12）．

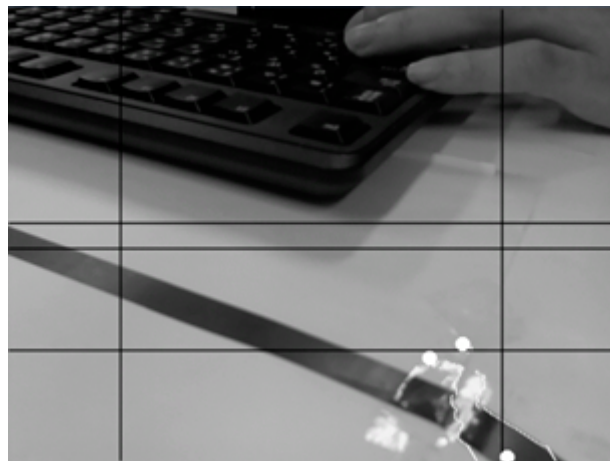


図 12 車線検出に失敗する (c) の場合の例

これらの問題を改善する手法を以下に示す．

ハフ変換によって得た画面下端に存在する複数の線分から，車線の線分である条件を判定することで，車線とそれ以外を区別する．このために，線分下端の点が，次に述べる車線の左端か右端のいずれであるかを調べる．左端は注目画素の左画素値が 255, 右画素値が 0 である．右端は注目画素の左画素値が 0, 右画素値が 255 である．線分下端の点の組み合わせの中で，左端と右端の幅が

もっともらしい組み合わせを車線の左端・右端とする。

車線の湾曲が大きくなると探索に失敗する問題に対しては、縦方向のエッジ成分だけでなく、横方向のエッジ成分を併用する。そのために、基本的なエッジ探索は縦方向のエッジ画像で行うが、画面上端・左端・右端のどれかに達する前に車線が途切れた時、エッジ探索に用いる画像を縦方向のエッジ画像から横方向のエッジ画像に切り替える。そして、エッジが途切れた位置から横方向のエッジを用いてエッジ探索を再開する。これを、画面上端・左端・右端のいずれかに達するまでエッジ探索を繰り返す。

道路を想定した紙の表面で光が鏡面反射し、かつ、鏡面反射によるエッジと車線のエッジが重なった時に、車線のエッジ以外のエッジを探索してしまう可能性がある。この問題に対しては、車線のエッジを探索する可能性を高めることで、誤ったエッジを探索する可能性を軽減する。このために、ハフ変換によって得た画面下端の線分が、左右どちらに傾いているかを調べる。左に傾いた線分である場合、車線は画面左端、もしくは画面上端に到達すると仮定する。線分が右に傾いている場合にも同様に、車線は画面右端、もしくは画面上端に到達すると仮定する。実際には、車線の湾曲が大きい場合、画面下端の線分の傾きと異なった方向に車線が到達することも考えられるが、今回はそれを除外した。線分が左に傾いている時、エッジ探索は注目画素と、その左画素にのみ行い、線分が右に傾いている時は、注目画素と右画素にのみエッジ探索を行う。このようにエッジ探索の範囲を限定することで、車線のエッジを適切に検出する可能性を上げた。

### 3.3 模型自動車の進路を制御する手法

模型自動車の進行方向を制御するために、エッジ探索で得た車線情報を用いて車線の状態を 5 つに分類し、状態遷移を繰り返すことで進行方向を制御する方法を開発した。前輪タイヤのステアリング角度を制御する数値は、中央が 0、向かって右を正の値、向かって左を負の値で表す。サーボモータの稼動域は数値で-13~13 の範囲である。

図 13~図 17 に、車線の探索が終了した時の状態とサーボモータに設定する数値、および前進・後退の関係を示す。このように、車輪の角度としては、直進・左・右の 3 通りだけを判断している。より細かい角度を設定することで、なめらかな走行が可能になると考えられるが、それを実現する制御則を見出すこと

は難しい．そのため，今回はこのように単純な制御則を開発した．

状態 1：車線の探索が図 14 の赤色領域で終了した場合

処理：ステアリング角度 0（直進），前進

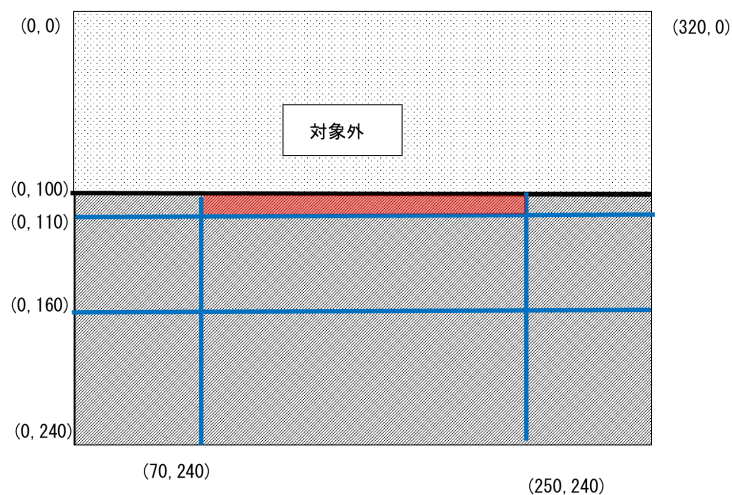


図 13 状態 1，直進

状態 2：車線の探索が図 15 の赤色領域で終了した場合

処理：ステアリング角度 13（右），前進

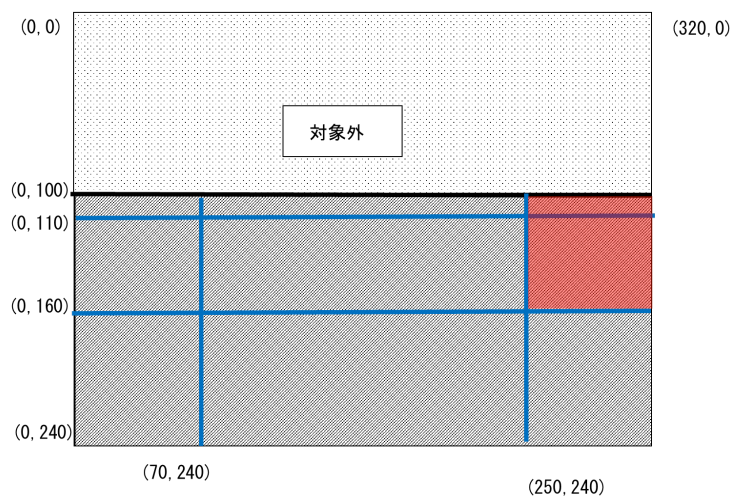


図 14 状態 2，右方向に前進

状態 3：車線の探索が図 16 の赤色領域で終了した場合

処理：ステアリング角度-13（左），後退

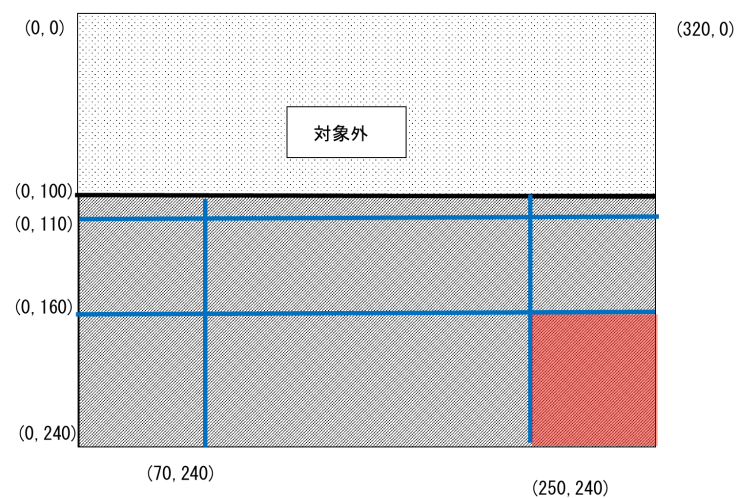


図 15 状態 3，左方向に後退

状態 4：車線の探索が図 17 の赤色領域で終了した場合

処理：ステアリング角度-13（左），前進

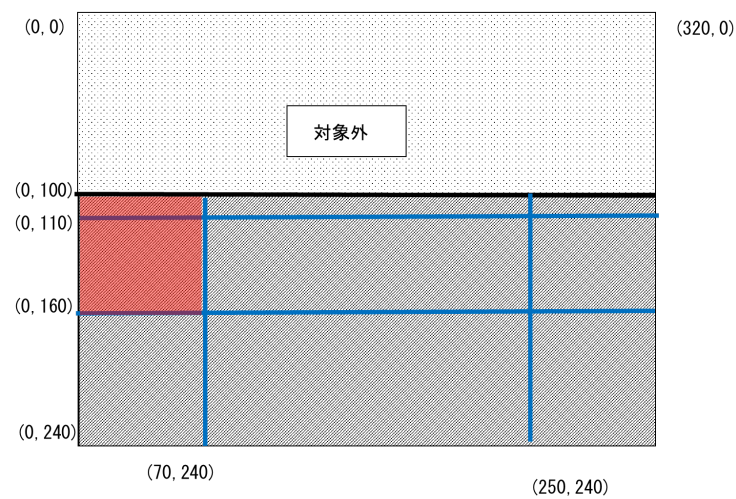


図 16 状態 4，左方向に前進

状態 5：車線の探索が図 18 の赤色領域で終了した場合

処理：ステアリング角度 13（右），後退

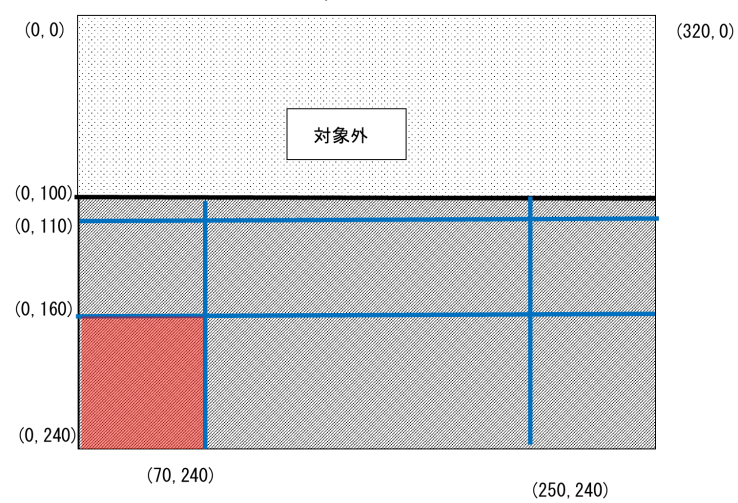


図 17 状態 5，右方向に後退



## 4 章 実験結果と考察

### 4.1 車線検出に関する実験結果

路面を想定した紙面に黒色で印刷した実線を車線に見立て、実験を行った。

図 18 は、3 章に示した基本となるエッジ探索プログラムの実行結果である。図 18 が示すように、画像下部に車線だけ見える場合であれば、その他の場所に車線以外の物体が存在しても、車線を正しく検出することができた。

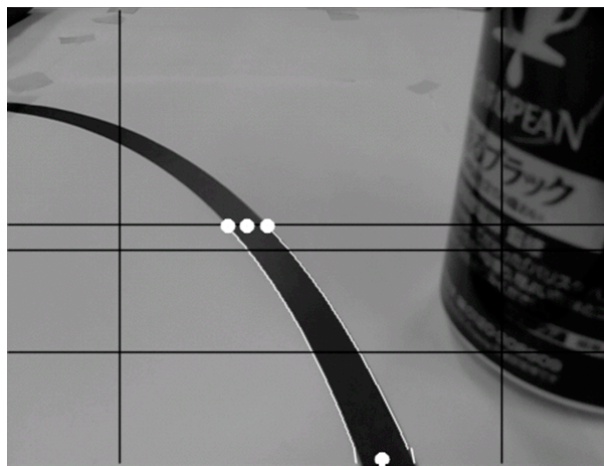


図 18 基本となるエッジ探索の結果

3 章で示したように、このアルゴリズムだけでは、環境によっては正しく車線検出できない問題があった。以下がその問題と、提案する手法を実行した結果である。

#### (1) 画面下端に車線以外のエッジが存在する場合

図 19 が車線検出に失敗した例である。図 20 がその時のエッジ画像を二値化したものである。図 21 は提案手法を用いたプログラムの実行例である。図が示すように、画面下部の領域内に障害物が存在する場合でも、車線を適切に検出することができた。ただし、現在の手法では、あらゆる物体と車線を区別できるわけではない。例えばペンや棒状の物体で、その幅が車線と類似している場合には正しく区別できない。

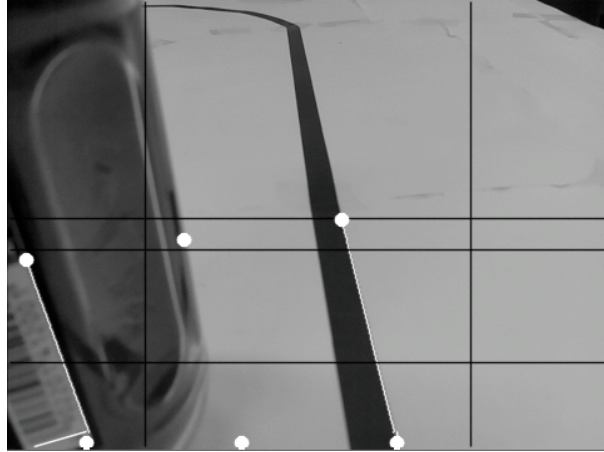


図 19 車線検出に失敗した例



図 20 車線検出に失敗した時のエッジ画像(二値画像)

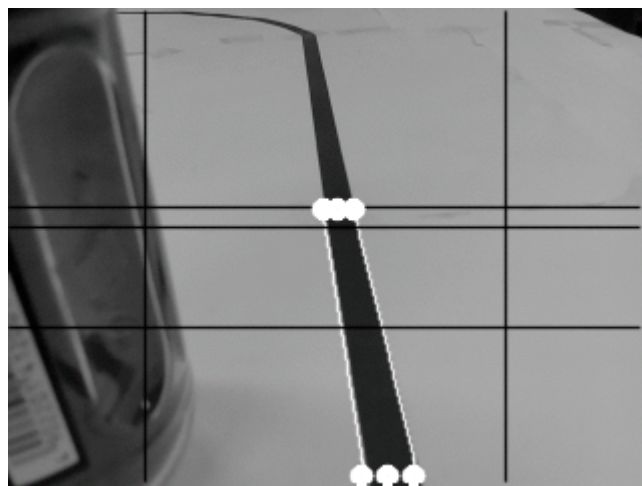


図 21 提案手法を用いたプログラムの実行結果



(2) 車線の湾曲が大きい場合.

図 22 が, 車線の湾曲が大きいいため, 追跡に失敗した例である. 図 23 は縦方向のエッジを抽出した画像(二値画像), 図 24 は横方向のエッジを抽出した画像(二値画像)である. 図 23 で車線のエッジが途切れている座標から(図 23 赤丸), エッジ探索に用いる画像を図 24 に切り替え, エッジを再探索した. その結果, 図 25 に示すように車線を検出することができた.

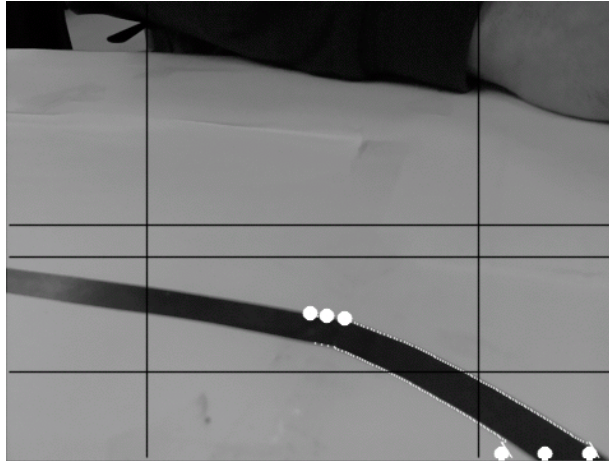


図 22 車線が横に伸びたため, 車線検出に失敗した例



図 23 縦方向のエッジを抽出した画像(二値画像)

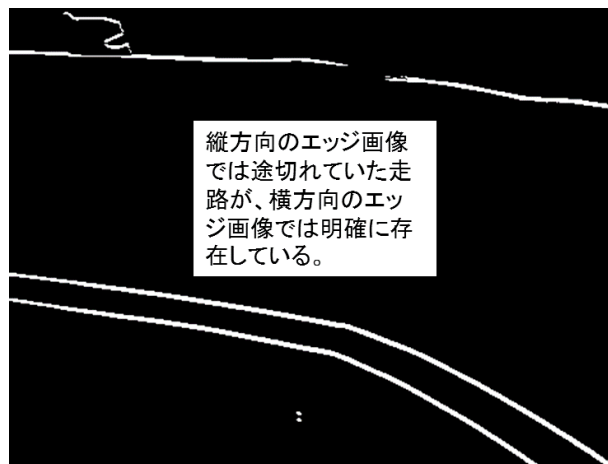


図 24 横方向のエッジを抽出した画像(二値画像)

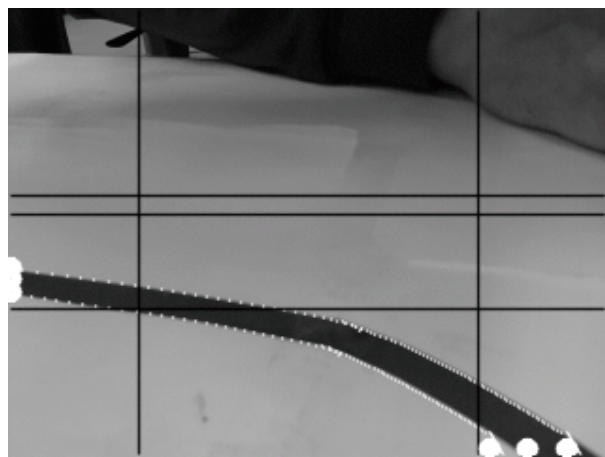


図 25 提案手法を用いたプログラムの実行結果

(3)道路を想定した紙の表面で光が鏡面反射する場合.

図 26 は鏡面反射のエッジと車線のエッジが重なったため、車線検出に失敗した例である．図 27 と図 28 は提案手法を用いたプログラムを実行したときの結果である．図 27 の場合は、車線を検出することができたが、図 28 場合では正しく車線検出することができなかった．このように、提案手法で必ずしも成功するわけではないが、成功する可能性が増す．



図 26 鏡面反射が原因で車線検出に失敗した例



図 27 提案手法を用いたプログラムの実行結果 1 (成功)



図 28 提案手法を用いたプログラムの実行結果 2 (失敗)

画像の二値化は sobel フィルタを適用後、画面全体に大津の方法を用いた可変閾値で行っている。二値化の範囲を限定すれば、車線と紙面で起こる鏡面反射のエッジを区別できるのではないかと考えた。図 29 は二値化の範囲を限定した時の実行結果である。二値化の範囲を限定しても、車線と鏡面反射によるエッジを区別することができず、期待していた結果を得られなかった。



図 29 二値化の範囲を限定した場合のエッジ抽出結果

#### 4.2 模型自動車の制御に関する実験結果

車線検出で得た情報に基づいて、車線の状態を 5 つに分類し、状態遷移を繰り返すことで、模型自動車の進行方向を制御した。このプログラムを用いて、作成した周回コースのラインをトレースすることができた。

図 30 は周回コースのコーナを、切り返ししながらトレースする制御の例である。最初、直線をまっすぐ走行している場合、状態 1 にある。右折コーナに近づくと、ラインが右に曲がるので状態 2 に遷移する。図のように模型自動車では曲がり切れないカーブの場合、車線の終端はさらに湾曲するので、状態 3 に遷移する。ここでステアリングを切り返して後退すると状態 2 に戻ることができる。そこで、再度、右方向に前進する。これを繰り返すと、最終的には状態 1 にまで戻ることができるので、直進する。

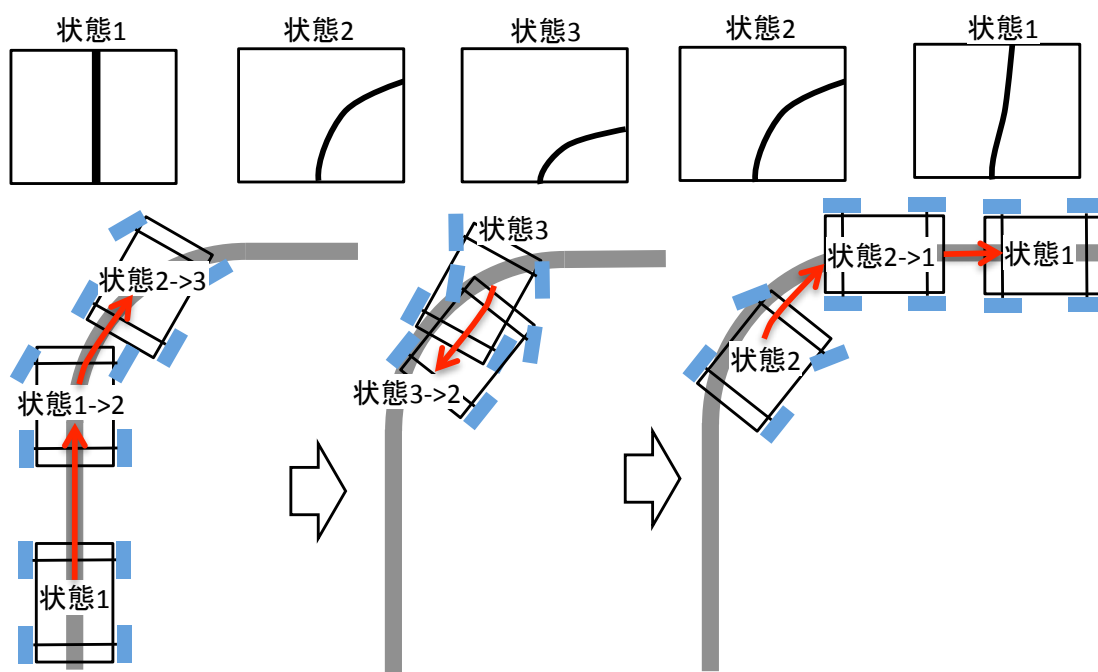


図 30 模型自動車の制御例

この方法の問題として、状態遷移を繰り返すことだけで制御しているため、状態が移り変わる境目付近(図 31 の赤線)に走路が検出された場合、頻繁に状態遷移を繰り返し、結果として不自然な動作になることがある。また、現在の車線検出のプログラムは、画面手前に車線が存在していることを前提にしているため、模型自動車がコース上に位置していない場合の動きは定義されていない。従って、カメラが車線を捉え続けなければならないという問題点もある。

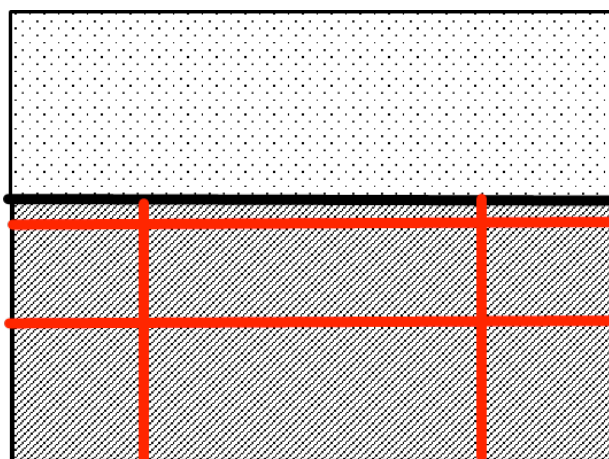


図 31 5つの状態の境目

### 4.3 考察

道路面をとらえた一枚の静止画であっても，その画像の中の車線を正しく検出することは容易ではない．映像を使って，リアルタイムに適切な車線検出を行い続けることは更に困難である．しかし，自動運転には，適切な車線検出は必要な情報である．想定される多くの状況の中で，車線の誤認識が発生する原因の対策を行い，誤認識する可能性を可能な限り減らさなければならない．自動車の自動運転実現のために重要なことは，誤認識の発生確率を減少させ，より完全な精度に近い車線検出システムを実現することである．

模型車の制御に関しては，検出した車線情報を元に状態遷移を繰り返すことで，ライントレースすることができた．しかし，単純な状態遷移の繰り返しのみで制御しているため，車線が状態の移り変わる境目付近で検出された場合に，状態遷移を頻繁に繰り返す．これは人間が自動車を制御する場合には起こり得ない動作であり，極めて不自然な動作である．その結果，人間が自動車を運転する時のような滑らかさは実現できなかった．

## 5 章 結論

カメラモジュールを接続した Raspberry Pi2 を模型自動車に搭載し、カメラから得た画像に処理を施すことで、車線を検出した。検出した車線情報を元に、自動車に見立てた模型自動車を、作成したコース上で自動制御することができた。また、画面下端に車線以外のエッジが存在する場合、車線の湾曲が大きい場合、紙面上で光の鏡面反射によるエッジが車線付近に存在する場合の 3 つのパターンに対して、車線検出が誤認識する可能性を軽減することができた。

実際の道路で自動車の自動運転を実現するためには、カメラを用いて自動車の周囲を撮影し、それを処理することで周囲状況を認識する技術がほぼ完全なものではなくてはならない。安定して動作する範囲が限定されていては事故に繋がりがねない。運転者が自動車の制御に関する権限を全てシステムに委譲するためには、自動運転のシステムが完璧で、あらゆる場合でも安定して動作しなければならない。本研究を通して、数あるコンピュータの利用技術の中でも、自動車のように人間の命を扱うようなものについては絶対的な信頼のおけるシステムが必要不可欠であることを、改めて痛感した。

また、本研究で行ったように、実際には高コストな組み込み機器であっても、Raspberry pi を用いることで、低コストで模擬実験を行えることが分かった。Raspberry Pi によって様々な用途の模擬実験を低コストで行うことができるため、Raspberry Pi は組み込み機器の基本技術を学ぶ上で非常に有効な手段である。



## 参考文献

- [1] 五十部宏幸, 中村靖, 「単眼車載カメラ画像を利用した自動車運転支援」, 広島工業大学紀要研究編 第43巻 pp. 317~322, 2009年.
- [2] 山田直人, 田森信行, 塩見彰睦, 「適応エッジ保存平滑化を用いた白線検出手法」, 電子情報通信学会論文誌. D, 情報・システム. 2, パターン処理. 2005-08-01, 88(8):1421-1431. 11p, 2005年.
- [3] 林直宏, 富沢哲雄, 末廣尚士, 工藤俊亮, 「二重円型オペレータによるロバストな白線検出手法(移動ロボットのための視覚)」, ロボティクス・メカトロニクス講演概要集 Vol. 2012, p. "2P1-J11(1)"-"2P1-J11(4)"
- [4] [https://github.com/robidouille/robidouille/tree/master/raspicam\\_cv](https://github.com/robidouille/robidouille/tree/master/raspicam_cv)
- [5] <http://wiringpi.com>

## 謝辞

本研究を進めるにあたり, 様々な指導を賜りました担当教授の蚊野浩先生に深く感謝申し上げます.

## 付録 本研究で作成したプログラムの説明

|            | 機能                                              |
|------------|-------------------------------------------------|
| autorun. c | カメラモジュールから得た画像の中から車線を検出し，その情報を元に模型自動車を制御するプログラム |