コンピュータ理工学特別研究報告書

題目

シングルカメラステレオを用いた模型自動車の自動運転に関する研究

学生証番号　444233

氏名　大島樹瑞久

提出日　平成 30 年 1 月 24 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

本研究では，Raspberry Pi が制御する模型自動車に一つのカメラだけを使ったステレオカメラ（シングルカメラステレオ）による障害物検知機能を付加し，自動運転を行った．開発したシングルカメラステレオは，カメラの視野をミラーで左右に分割したミラーカメラシステムである．ステレオ視の原理によって被写体までの距離を測定し，障害物の判定に利用する．

シングルカメラステレオを用いた障害物検出は,入力画像を中央で分割し，距離画像の生成に必要な右目画像・左目画像のステレオ画像を抽出する.抽出したステレオ画像に対して歪みの除去と平行化を行なった後，ブロックマッチングによって距離画像を生成する．生成した距離画像からノイズを除去した後,あらかじめ設定した距離の閾値（80cm）によって 2 値化する．80cm よりも近い領域に存在する被写体を障害物の候補とする.二値化を行なった画像内に一定値以上の面積を持つ領域がある場合に障害物と判断する.最終的に障害物領域の中心座標と距離画像全体の画素平均値を抽出した．その結果，単色で特徴の無い大きな面積を持つ壁のようなもの以外は，障害物として検出することができた.

模型自動車の制御判定のために，距離画像の画像座標(x,y)と画素値(z)をデータとして収集し，3 次元点群を作成する．作成した点群に対して平面をフィッティングし，法線ベクトルを求める．得た法線ベクトルと模型自動車から平面に向かう方向ベクトルとのなす角を求め，障害物を避けるのに必要なステアリング角度を算出する.これにより，模型自動車の滑らかな制御を実現することができた.運転アルゴリズムは，障害物検知の結果を用いて，次のようにした.

(1)障害物があり，避けられる場合,右(左)にステアリングを動かし，回避する.

(2)障害物があり，避けられない場合，停止し，後退する.

(3)障害物がない場合，前進する.

また，壁に囲まれたコーナーなどで行き詰まらないように，後退を 3 回繰り返した時には，左にステアリングを切った状態で後退し，切り返しを図るようにした．この運転アルゴリズムを用いて14号館2階を一周させる実験を行なった．ただし,壁を障害物として検出できるようチェスボードを壁に付け,認識しやすいようにした.実験の結果,チェスボードを用いることでなんとか一周することができたが，これらの手助けなしでは周回することができなかった.

目次

1章 序論

自動運転技術の研究・開発・実用化が活発である．アメリカの NHTSA(米国運輸省道路交通安全局)や SAE(自動車技術者協会)は，自動運転のレベルを 6 段階に分けている．これは,自動操縦システムに自動車の運転を任せる度合いを,段階で示したものである．レベル 1～2 が運転支援にとどまるもの，レベル 3 以上が自律走行を行えるものとされている．運転支援とは，前方との車間距離に応じてブレーキを制御する自動ブレーキなど，ドライバーの運転を補佐する機能である．また自律走行とは，運転支援の機能が高度化し，あらゆるシーンでの運転が自動化され，ドライバーレスで走行できるもののことである．

自動運転の処理を認知・判断・操作の 3 つに分けることができる．認知とは,自車輌の周囲を確認するセンサ(カメラ・レーダー・LIDAR・ソナーなど)を用いて，歩行者や路面状態，他車両の位置などを検出し，周囲の現状を把握することである．判断とは，認知で得られた結果から次に何をするべきかを決定することである．操作とは，認知で把握した情報と判断の結果から最適のルートを計算し，ルートに沿った走行を行うことである．

本研究は,模型自動車を 1 つのカメラモジュールを使用した自作の制御システムを用いて自律走行させるものである．この研究の目的は，画像を用いた障害物検出や自動運転の基本技術を自分で作成することで，画像処理の技術･組み込みシステム技術を習得することである．そして，完成した制御システムを用いて本学の 14 号館 2 階の廊下を一周することが，本研究の最終目標である．

今回の研究では，実車の代わりに TAMIYA 製のラジコンカーに加工を加えたものを使用した．車載カメラと車載コンピュータを，Raspberry Pi とカメラモジュールで代用した．カメラモジュールは1台で2台分の視点を再現できるように加工を加えたものを用いた．このカメラシステムをシングルカメラステレオとよぶ.シングルカメラステレオから得る画像情報を元に障害物を検出し，自動的に回避するプログラムを開発した．

本論文では，2章で自動運転技術について，3章で模型自動車とシングルカメラステレオのシステム構成について説明する．次いで4章では障害物の検出方法．5章では運転アルゴリズムの開発と自動走行結果を説明し，6章で結論を述べる．

2 章　自動運転技術

自動車メーカーや IT 企業の目標は,2020 年に完全自動運転を実用化することである. この目標に向けた開発や実用化の発表が，様々な企業からなされている. 例えば，Google の自動運転車開発部門であるウェイモは，ハンドルやブレーキがない電気自動車を開発し，路上試験を行っていた．BMW は，2014 年に，自動運転プロトタイプによるドリフト走行をラスベガス・スピードウェイで行った．オンライン配車サービスで有名な Uber が Volvo と提携して, サンフランシスコにて自動運転タクシー「uberX」(図 2.1) の配車を 2016 年 12 月 14 日より開始した．このような自動運転技術の進歩は道路交通システムのパラダイムを変えるものとして，自動車産業だけでなくサービス産業や運輸産業からも注目を集めている．事実,自動運転技術は自動車以外の場所でも応用されている．一例を挙げると，日産が自社の自動運転技術を応用して開発した，行列待ちを楽にしてくれる自動運転チェア「ProPilot Chair」(図 2. 2)が話題を呼んだ.

図 2.1 Uber が発表した自動運転タクシー「uberX」

図 2.2 日産の自動運転技術を応用したイス「ProPilot Chair」

一方, 自動運転は確実なロードマップに基づいて実用化が進められているわけではない．Uber が提供を開始した「uberX」は, 赤信号をスルーしていく映像が撮影されたり,

車両が横転するほどの事故を起こしたりしている．自動運転技術は，手放しで命を預けられるレベルに達しているとまでは言えない．

序論でも述べたように，自動運転技術には法令面や技術面から定めたレベルが定義されている．表 2.1 に NHTSA の定義を示す．日本国内ではレベル 2 までが市販売されている．日産が製品化した「ProPilot」や，“ぶつからない車”というコピーで有名になったスバルの「アイサイト」がレベル 2 に相当する．レベル 2 までの自動化は人間をサポートするものであり，運転の責任は人間にある．一方，レベル 3 以上は基本的に人間が操作を行う必要がないため，運転の責任はシステム側にあることになる．そのため，政府が法令やインフラを整備し終えるまで実用化が難しいとされている．そのような中でも，Audi はレベル 3 の自動運転技術を搭載したモデル「A8」を 2018 年からドイツ国内で発売を予定している。

表 2.1 NHTSA による自動化のレベル定義

レベル	概要	安全運転に係る 監視・対応主体
レベル 0 手動運転	人間の運転者が全てを行う．	運転者
レベル 1 運転支援	ステアリング操作か加減速のいずれかを サポートする．	運転者
レベル 2 部分運転自動化	ステアリング操作と加減速の両方が連帯して 運転をサポートする．	運転者
レベル 3 条件付き運転自動化	特定の場所で全ての操作が自動化， 緊急時はドライバーが操作．	システム (一部運転者)
レベル 4 高度運転自動化	特定の場所で全ての操作が 完全自動化される．	システム
レベル 5 完全運転自動化	あらゆる状況においても操作が自動化． ハンドルもアクセルも不要．	システム

自動運転では，人間が行なっている認識・判断・操作をコンピュータが行う．そのためには，人間に代わって周囲環境を認識する装置が必要となる．そこで用いられるもの

が環境センサである.自動運転に用いられる代表的な環境センサは 4 つあり，それらを表 2.2 に示す.

カメラは解像度が高い画像を撮影するので,色・文字を区別することができ，路面のペイントや道路標示なども読み取ることが可能である．カメラセンサには単眼カメラとステレオカメラがある.単眼カメラは,一枚の画像を連続的に撮影・処理し，画像の変化から障害物の位置を検出するものである．ステレオカメラは 2 台のカメラから得る画像を重ね，被写体像の位置の違いを比較することにより，障害物の３次元的な位置や距離を検出するものである．単眼カメラと比較すると,ステレオカメラは障害物検出精度が良く，デメリットとしてコストが高い.

ミリ波レーダーは，ミリ波と呼ばれる波長が長い電波を照射し，物体から反射する電波の遅延時間を検出することで，物体までの距離と位置を把握するセンサである．他のセンサに比べると雨や霧・逆光などの影響を受けないため，視界が効かない夜間や悪天候に強い.ただし解像度が低いので，詳細な情報を得ることが難しい.

LIDAR(Light Detection and Ranging)は，赤外レーザー光をパルス状に照射し,物体から反射する光を計測するセンサである．ミリ波レーダーと同様に,電磁波の伝播遅延時間を計測する方式であるが，空間分解能が高い.しかし，雨や他の光の影響を受けやすい.

表 2.2 代表的な環境センサ

センサ		原理
カメラ	単眼カメラ	1台のカメラで撮影した映像を処理する
	ステレオカメラ	2台のカメラで撮影した映像を処理する
ミリ波レーダー		波長が1～10mmの電波を照射し，反射波を処理する
LIDAR		赤外レーザ光を照射し，反射光を処理する

自動運転レベルの 1～2 は，表 2.2 で示したセンサのうちどれかを使用すれば実用化できる．レベル 3 以上になると，複数のセンサを使用した複合的なシステムを開発しなければ，実用化が困難である.これらの中では,カメラセンサが最も多く搭載されている.

3 章　模型自動車とシングルカメラステレオのシステム構成

3.1　模型自動車の構成

今回の研究では，自作の組み込みシステムを使って，市販の模型自動車を制御することで自動運転の実験を行なった．組み込みコンピュータに，PICマイコンとRaspberry Piを組み合わせたものを使用した．模型自動車には，図3.1のタミヤ社製 RCカー，シャーシ MF-01X を用いた．このシャーシをRCカーとして動作させる場合，内蔵の電波受信機からステアリング(前輪の方向角度)を変える前輪用サーボモータとESC(Electric Speed Controller)に対してPWM(Pulse Width Modulation)信号を送信し，車体の動きをコントロールする．今回は，前輪用サーボモータと前進後退速度を制御するESCに送信する信号をPICマイコンから出力し制御を行う．また，Raspberry PiからPICマイコンに対し，I2Cインタフェースを介して指令を送ることで模型自動車を動作させる．

図 3.1 タミヤ社製 RC カー シャーシ MF-01X

サーボモータと ESC に送る信号は，PWM 制御のための周期的矩形波である．制御に用いる周期的矩形波は図 3.2 のようなものであり，パルス幅は 1.5msec を中心とした 1.1msec～1.9msec の範囲で制御を行う．ステアリングを操作する場合は，1.5msec のパルス幅で前輪の方向角度が中央になる．1.1msec あるいは 1.9msec のパルス幅の時，左右に最大角度になる．前進後退を行う場合，1.5msec でいわゆるニュートラル状態，1.5msec より短くすると前進，1.5msec よりも大きいと後退となる．なお，パルス電圧は 6.0V 程度の大きさである．

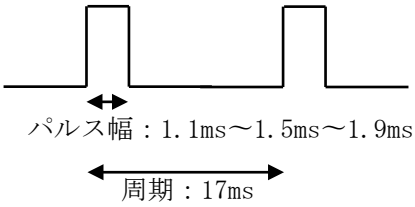


図 3.2 サーボモータと ESC に送信する PWM 信号

模型自動車の制御に必要な PWM 信号を発生させるために PIC マイコンを使用した．PIC マイコン・Raspberry Pi・RC カーの接続を図 3.3 に示す．PIC マイコンの型番は PIC 16F619 を使用した．PIC マイコンは I2C を介して Raspberry Pi と接続されており，Raspberry Pi から PIC マイコンにコマンドを送ることで適切な状態に設定する．

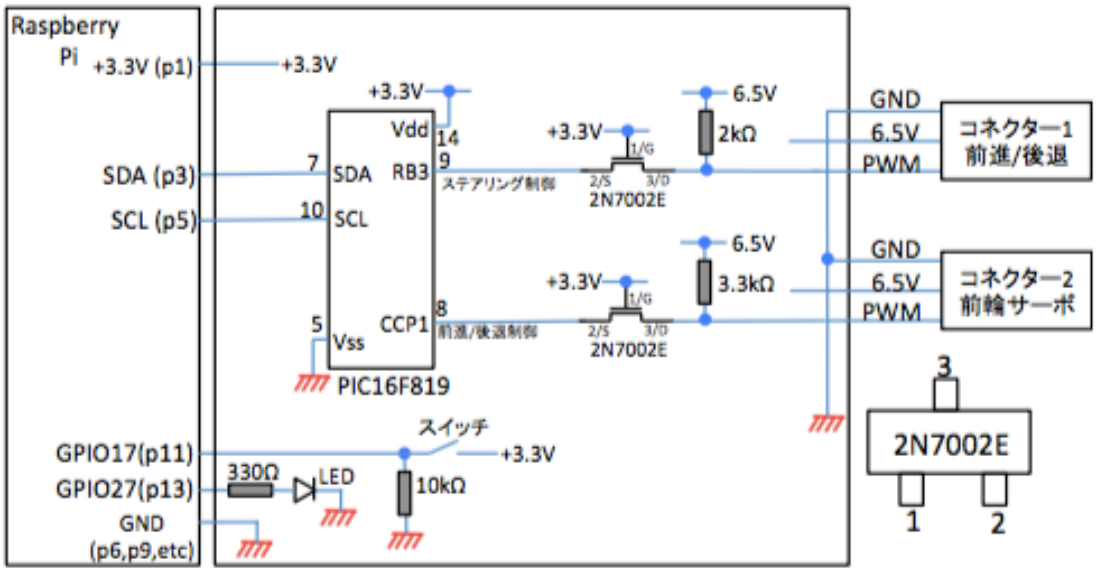


図 3.3 Raspberry Pi と RC カーの ESC とのインタフェース回路

3.2 シングルカメラステレオの構成

今回の研究では、模型自動車を制御する環境センサとしてステレオカメラを使うことにした．その主な理由を以下に示す．

(1) 障害物までの距離を，直接，計測することができる．単眼カメラでも距離計測は不可能ではないが，被写体（前方車両）の大きさを仮定するなど，不確実である．

(2) 遠距離を計測することに向いている．昨年の特別研究 II[5]で自作の LIDAR センサを開発した．これは測定可能な距離が短く，模型自動車の運転速度を上げることが難しかった．今回の研究では，この問題を解決することを目指した．

(3) システムが単純である．LIDAR やミリ波レーダーで電磁波の遅延時間を計測するものは高度なアナログ回路技術が必要である．それに比べるとステレオカメラはシステムが単純である．

次に，Raspberry Pi にステレオカメラを実装する方法を検討した．その候補を以下に示す．ここで，Pi カメラは Raspberry Pi 専用のカメラモジュールであり，USB カメラと比べて，高速に映像を取り込むことができる．

(1) ２台の USB カメラ，または，１台の Pi カメラと 1 台の USB カメラを接続する．

(2) Pi カメラを接続した 2 台の Raspberry Pi を利用する．

(3) Pi カメラにミラーで構成するアタッチメントを付加してシングルカメラステレオとする．

これらの中で Pi カメラを使ったシングルカメラステレオを採用した．その理由は，模型自動車のシステムが単純化でき，かつ，距離画像を生成する時間が最も高速になると考えたからである．

シングルカメラステレオは文献[4]などに先行事例がある．通常の単眼カメラに図3.4のような構成のミラーあるいはプリズム光学系を付加することで，1 枚の画像に右目画像・左目画像を写し込むステレオカメラである．今回は，図 3.4 の実カメラに Pi カメラを用いた．ミラーを使ったアタッチメントは木材とアルミ板などで自作した．

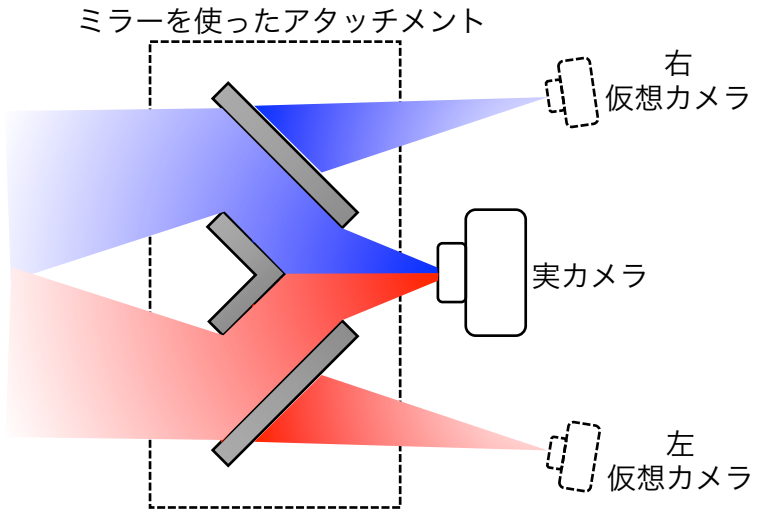


図 3.4 シングルカメラステレオの構成図

3.3 模型自動車システムの構成

模型自動車・Raspberry Pi・シングルカメラステレオを使った自動運転システムの全体を図 3.5 のように構成した．Raspberry Pi の OS は最も一般的な Rasbian を使い，自

動運転のプログラムはC/C++で開発した.画像処理ライブラリとしてOpenCVを使った.プログラムを開発する段階では，キーボード，マウス，ディスプレイを接続する．走行テストを行う場合は,MacBookとRaspberry Piを有線LANのケーブルで接続し,MacBookのターミナルからSSHで接続する．MacBookで簡単なプログラムの修正と，プログラムの起動を行う．MacBookなどを接続せず，スタンドアロンで動作させることも可能である.

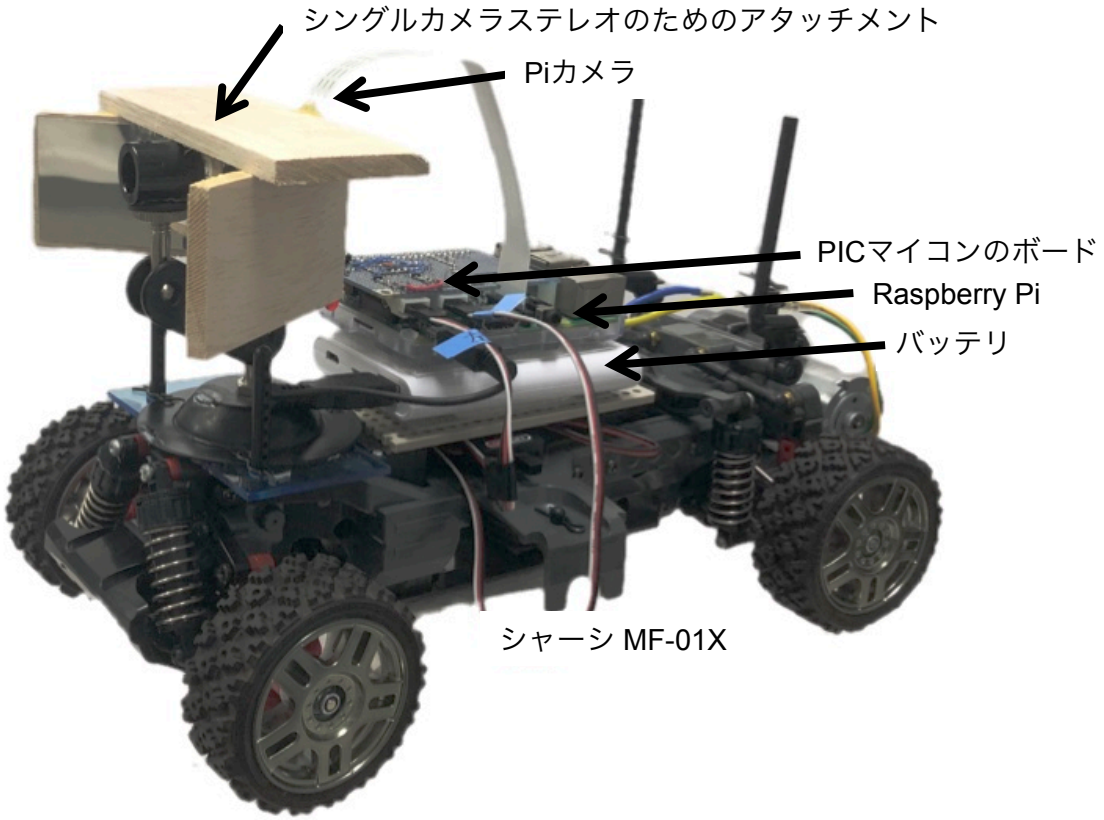


図 3.5 模型自動車システムの外観

4 章　障害物の検出方法

4.1　検出方法の概要

研究した障害物の検出方法を，図 4.2 を使って説明する. シングルカメラステレオで 640x480 画素の画像を撮影すると図 4.2 の入力画像のようになる. 入力画像全体を 4 分割し，下側の 2 箇所を画像として取り出す．2 枚の画像はステレオ画像になっているので，ステレオ処理を行うことで距離画像を生成する. 距離画像を処理することで障害物を検出する.

ここでステレオ画像処理について説明する. 同一の被写体を異なる位置に置いた 2 台のカメラで観察すると，カメラから被写体までの距離に応じて，被写体像の位置が変化する．ここで，特性が同じ 2 台のカメラを水平に並べて配置したステレオカメラ（標準ステレオとよぶ）の場合，画像上での被写体像の位置ずれが水平方向だけになる．この位置ずれ量を視差とよぶ. 視差は被写体までの距離に反比例する．ステレオマッチングは，標準ステレオで撮影した 2 枚の画像から，画素ごとに視差を計算するための画像処理のことである．通常，ブロックマッチングで実装する.

一方，一般的なステレオカメラは標準ステレオになっていない. 標準ステレオで設計し，組立を行った場合でも，厳密に言えば誤差が生じている．場合によっては，2 台のカメラの視線方向を注視したい一点で交差させる輻輳ステレオを構成することもある. 今回設計したシングルカメラステレオは，図 3.4 における仮想カメラの視線方向がカメラの前方 80cm 程度の距離で交わる輻輳ステレオになっている.

ステレオマッチングの処理に先立って，ステレオカメラの状態をあらかじめ測定することと，測定した数値（ステレオカメラのパラメータとよぶ）を用いて，元の 2 枚の画像を，標準ステレオで撮影した 2 枚の画像と等価な画像に幾何学的に変換する．最初にステレオカメラの状態を測定することをステレオカメラキャリブレーションという. ステレオカメラキャリブレーションで得た数値を使って，幾何学的な変換を加えることを平行化という．ステレオカメラキャリブレーションは事前に行う処理，平行化は入力画像に対して距離を求める度に行う処理である.

今回，行う処理の全体は次のようになる（図 4.3）

(1) あらかじめシングルステレオカメラをキャリブレーションする.
(2) キャリブレーションで得たパラメータを用いてステレオ画像を平行化する.
(3) ステレオマッチングを行い，距離画像を生成する.
(4) 距離画像から障害物の位置を推定する.
(5) 推定結果を基に模型自動車のステアリングとアクセルワークを制御する.

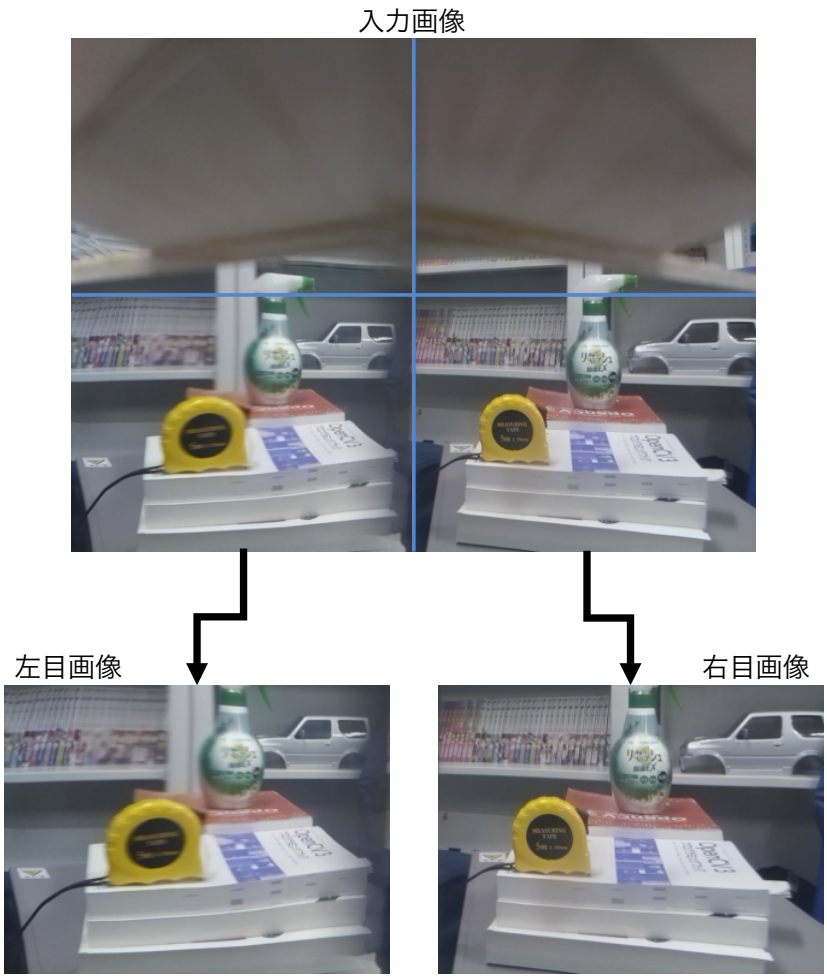


図 4.2 シングルカメラステレオで撮影した画像と使用する分割画像

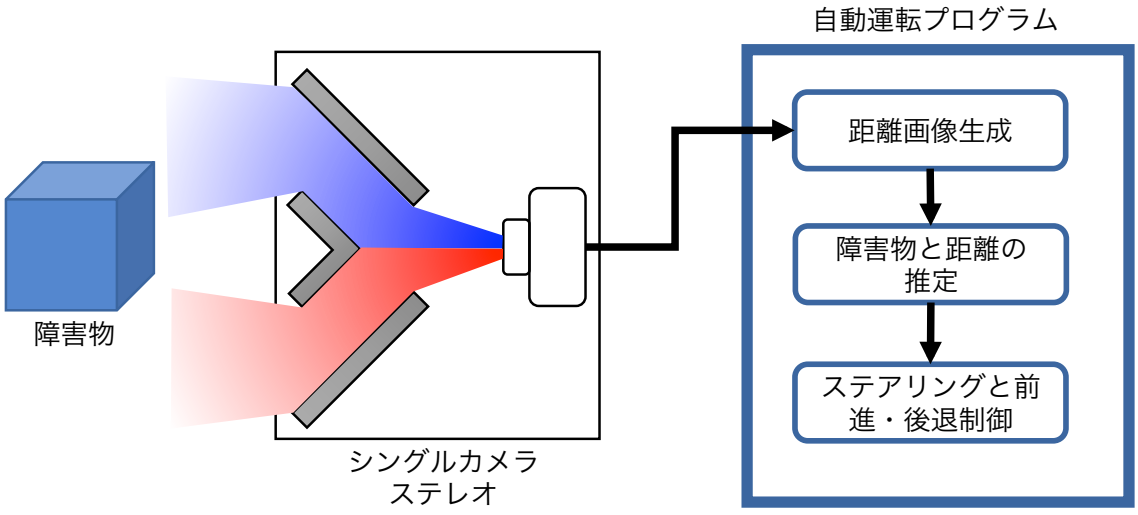


図 4.3 自動運転システムの処理の流れ

4.2　シングルカメラステレオのキャリブレーションと平行化

ステレオカメラのキャリブレーションでは，図 4.4 のような，配置と寸法が既知の平面パターン（チェスボード画像）をステレオカメラで 10 組程度撮影する．撮影する画像は，チェスボード画像を様々な角度から観察したものにする．今回の実験では図 4.4 のチェスボード画像を A3 サイズにプリントし，それを，シングルカメラステレオで撮影した．撮影した 9 枚の元画像から図 4.2 のように抽出した 9 組の右目画像・左目画像の組みを図 4.5 に示す．

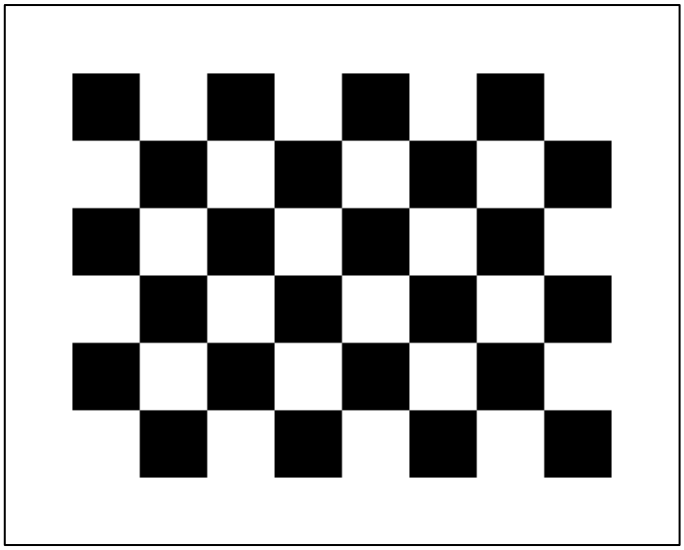


図 4.4 使用したチェスボード画像

OpenCV にステレオカメラをキャリブレーションする関数群があるので，それらを利用した．その結果，次のステレオカメラパラメータを得る．

(1) それぞれのカメラの焦点距離，画像中心，レンズ歪み係数などの内部パラメータ．

(2) 2 台のカメラ間の位置関係を示すステレオカメラの外部パラメータ．

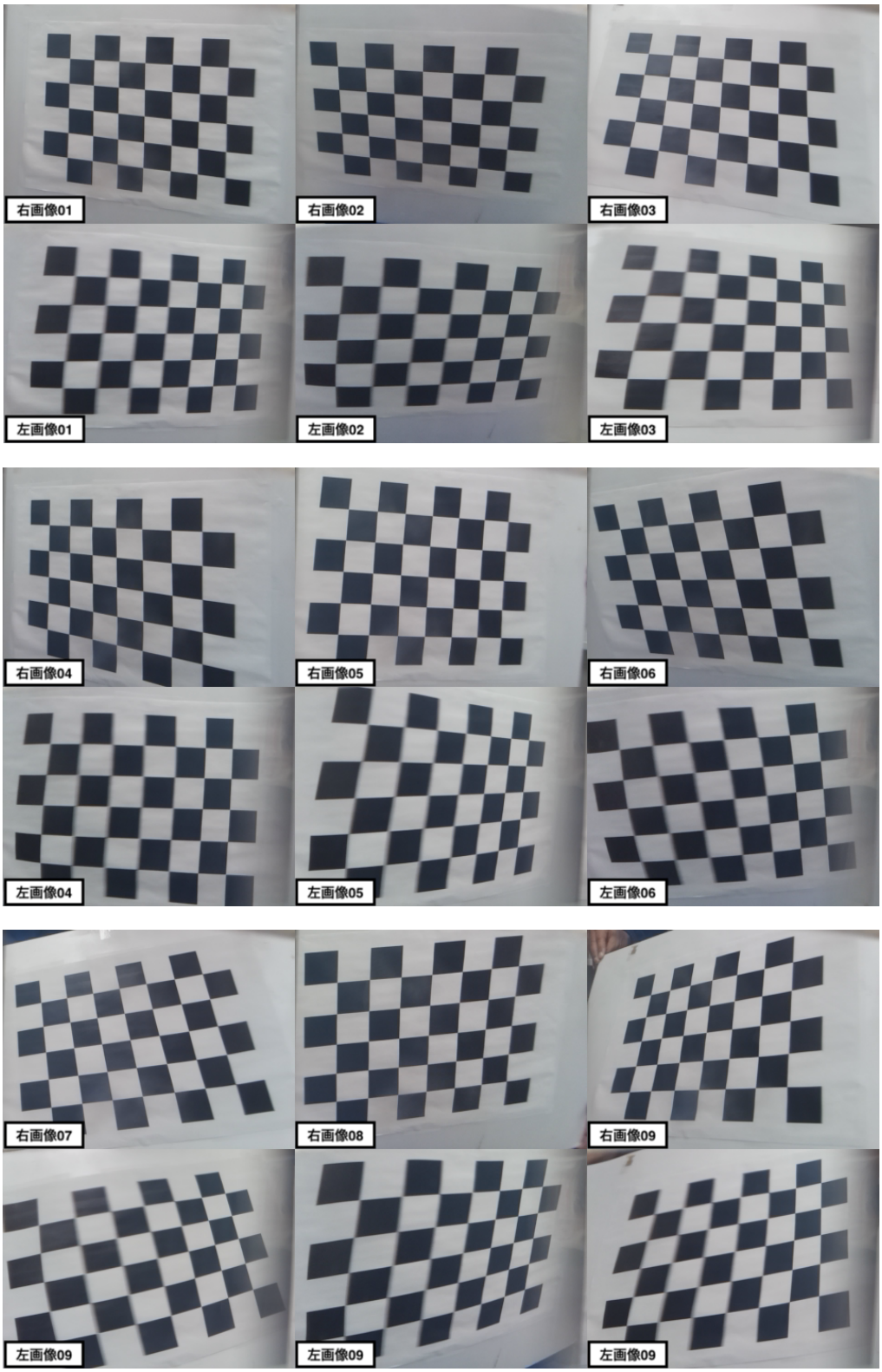


図 4.5 撮影した 9 組のチェスボード画像

次に，キャリブレーションで得たパラメータを利用して画像から歪みの除去と平行化を行う.平行化とは，2 つの画像間で対応点が同じ走査線に位置するように画像を幾何変換することである．これにより,対応点探索に用いるエピポーラ線が画像の横軸と平行になるので，高速に対応点探索が行えるようになる.

画像から歪み除去と平行化を行なったあと，距離画像を求めるために対応点探索を行う．対応点探索の方法には，大きく分けてローカル法とグローバル法の 2 種類がある．ローカル法は処理速度が速い．グローバル法は処理速度が遅く，精度が良い．OpenCV はリアルタイム応用を考えて高速性を重視しているため,ローカル法を中心に実装されている．ステレオマッチングに使う StereoBM と StereoSGBM の 2 つのクラスがある.StereoBM クラスは，ブロックマッチングを使用したクラスである.StereoSGBM クラスは,ブロックマッチングを採用しながらも,グローバルな情報を加味して，グローバル法並みの性能を得ている.今回は,リアルタイム性が重要であるから,StereoBM クラスを使用した.ここまでの処理の流れを画像で図式化したものを図 4.6 に示す.また，指定した範囲内の障害物検出の精度を上げるため，SADWindowSize を 21 に設定し，誤検出を減らした.

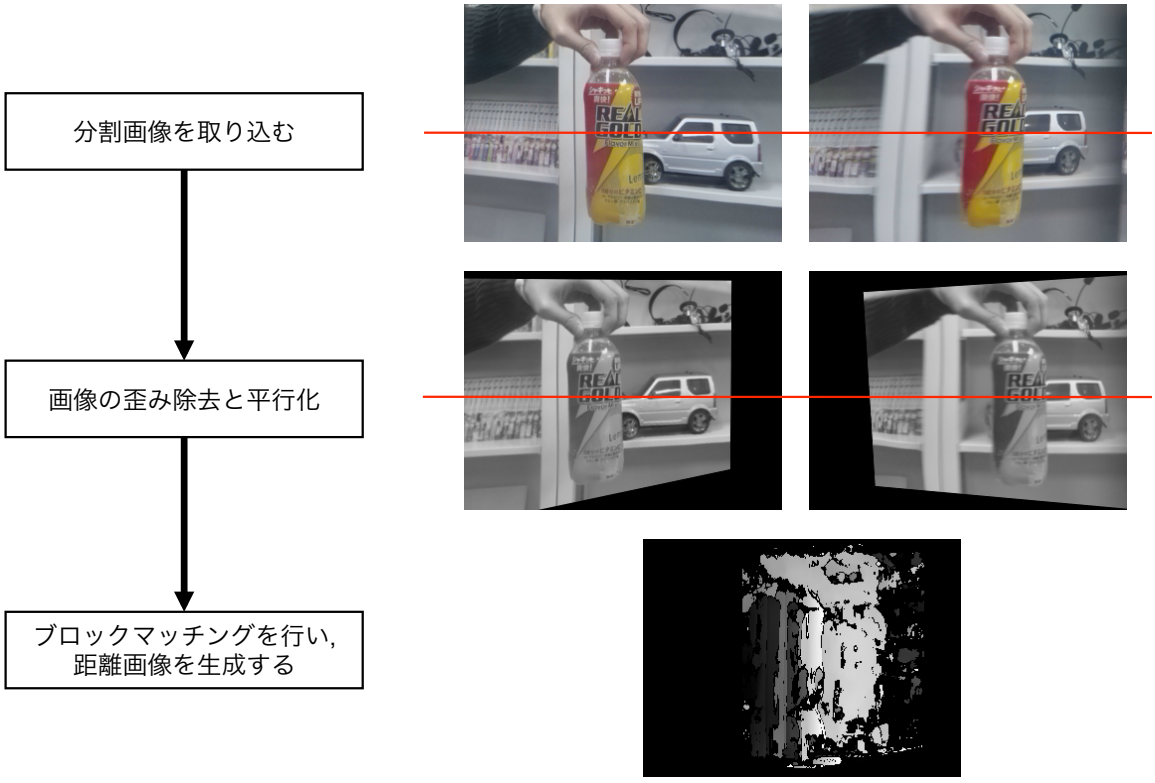


図 4.6 距離画像生成までの流れ

4.3 障害物の判定

4.2 の手法で生成した距離画像を用いて障害物の判定を行う.障害物であることを次のように判定した.

(1) 距離画像内の物体と模型自動車の実距離がある閾値より近いこと．かつ,

(2) その物体の大きさがある閾値よりも大きいこと.

距離画像の画素値から実距離を推定する方法を説明する.まずいくつかの物体について，距離画像の画素値とその物体までの実距離を測定した.それを図 4.7 の折線で示す.距離画像の画素値は視差を表し，理論的には，視差は距離に反比例する．この性質を利用して表 4.1 の折線を関数近似した(図 4.8，表 4.1).

障害物と判定する距離の閾値は，模型自動車の動作速度と，今回開発したシングルカメラステレオで安定的に測定できる距離の範囲を考慮して，試行錯誤的に決定した．その結果，模型自動車からの距離が 80cm 未満のものを障害物と判定した.

StereoBM が出力する距離画像を，80cm の距離を閾値として 2 値化することで，障害物の領域とそうでない領域に分離できる.しかし，そのままではごま塩ノイズや穴あきが多く，障害物判定に用いることができない.まず，ごま塩ノイズを除去するためにメディアンフィルタを使用した．メディアンフィルタは，ごま塩ノイズの除去によく使われる手法で，画素値の大きな変化による影響を受けづらいため，画像の鮮明さを劣化させずに除去が可能である．次に，穴あきの除去を行うために距離画像に対して 2 値化を行い，モルフォロジー変換を用いて穴あきを除去した．閾値は，先ほど決めた認識距離である 80cm 付近の画素値 80.0 とし，閾値以上の場合は画素値を 255.0 に変更するようにした．その結果，障害物が認識距離内にある場合，障害物と認識しノイズが除去された鮮明な画像で表示することができた.

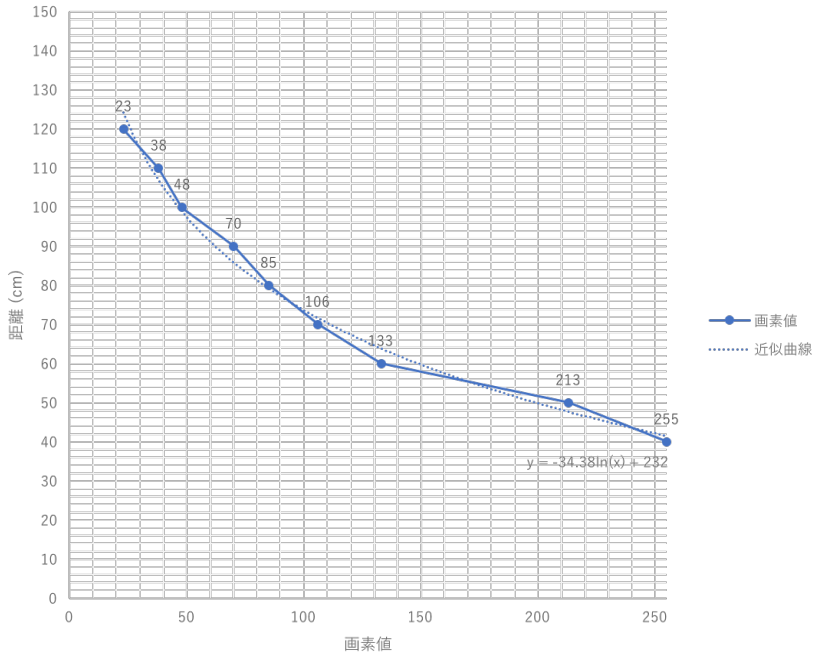


図 4.7 計測した距離と距離画像の画素値の関係

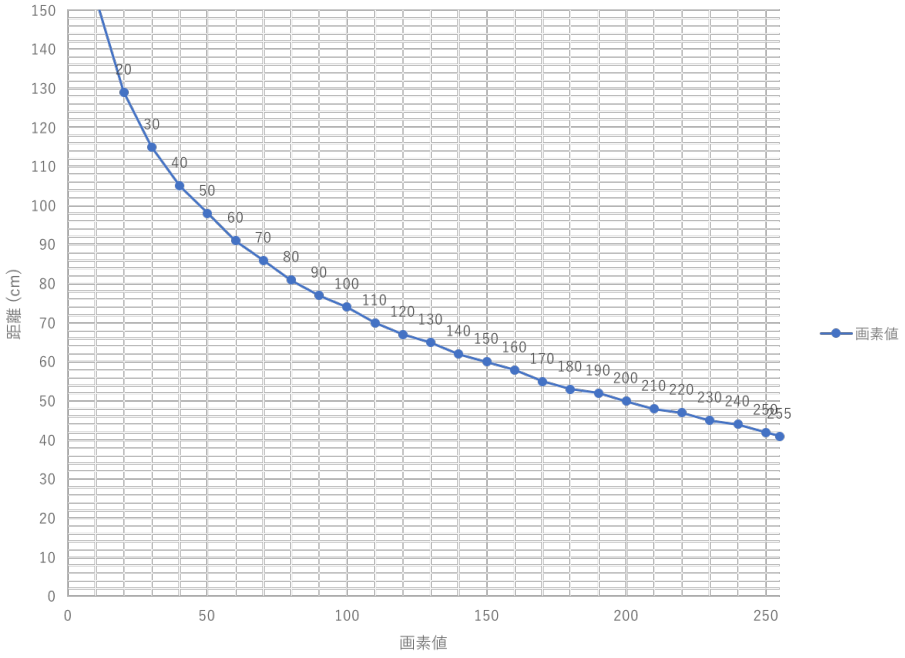


図 4.8 近似曲線に基づいた距離に対する画素値の計算上の値

表 4.1 画素値と距離の実測値と近似曲線を用いた画素値と距離の関係

実測値			近所曲線による値		
画素値	距離	色	画素値	距離	色
	150		0		
	140		10	153	
	130		20	129	
23	120		30	115	
38	110		40	105	
48	100		50	98	
70	90		60	91	
85	80		70	86	
106	70		80	81	
133	60		90	77	
213	50		100	74	
255	40		110	70	
	30		120	67	
	20		130	65	
	10		140	62	
	0		150	60	
			160	58	
			170	55	
			180	53	
			190	52	
			200	50	
			210	48	
			220	47	
			230	45	
			240	44	
			250	42	
			255	41	

障害物を判定する第２の条件のために，２値化した領域の大きさを求める.モルフォロジー変換を行った 2 値画像から領域を検出し,OpenCV の関数 cv::approxPolyDP を用いて多角形近似を行う.多角形近似した領域が一定以上の面積だった場合に障害物とし,領域の中心座標を障害物の位置とする(図4.9)．障害物の位置に応じてステアリングや前進後退の制御を行う.

制御に用いる位置情報やステアリング角は次の手順で求める.

(1) 距離画像の画像座標(x,y)とその画素値(z)をデータとした３次元点群を求める.

(2) ３次元点群に対して平面をフィッティングし，法線ベクトルを求める.

(3) 法線ベクトルと模型自動車から平面に向かうベクトルとの角度を求め，模型自動車に対し，平面の傾き角や向きを算出する(図 4.10,図 4.11).

(4) (3)で得られた傾き角などのデータから模型自動車のステアリングの角度を算出し,決定する.

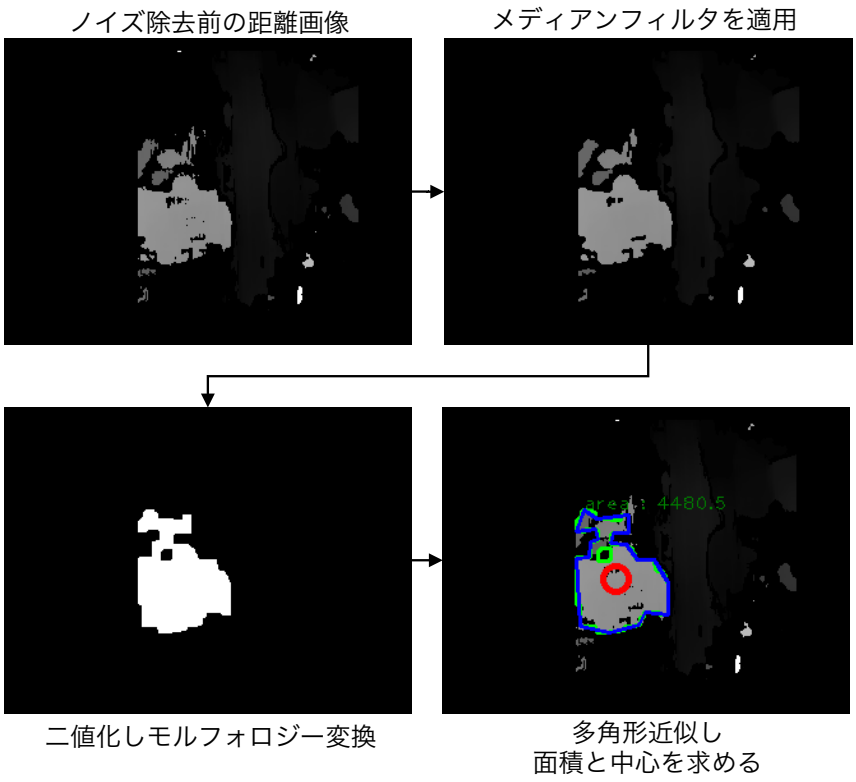


図 4.9 距離画像内の障害物に対するマーキング

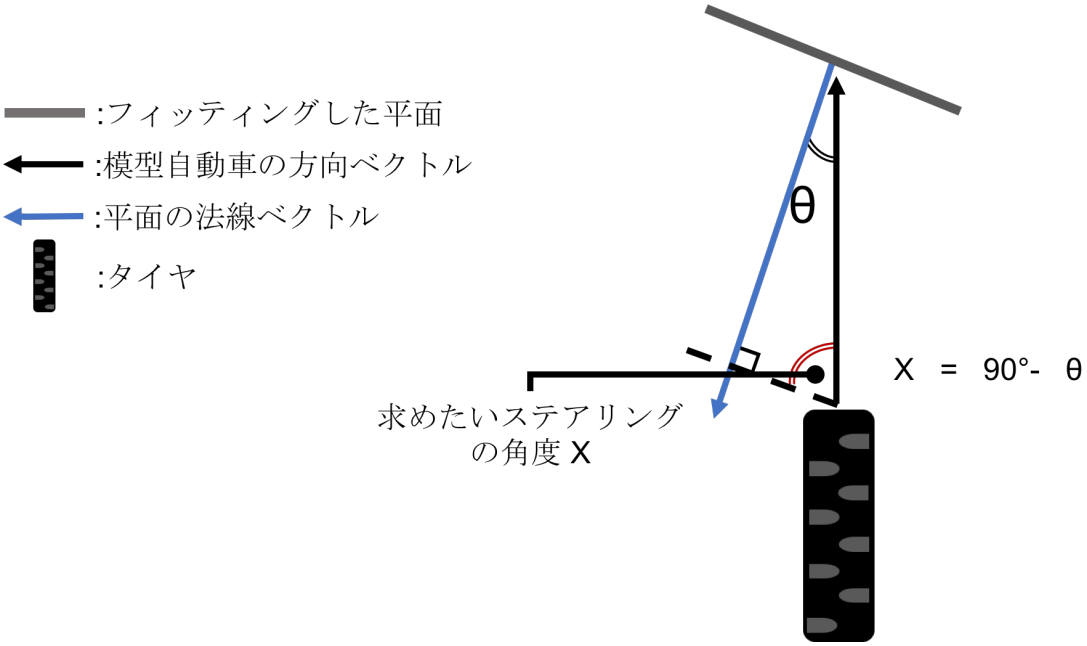


図 4.10 模型自動車のステアリング角の推定

:フィッティングした平面

:平面の法線ベクトル

θ_1 :正の角

θ_2 :負の角

θ_1ならば, 平面は左奥に傾いている.
θ_2ならば, 平面は右奥に傾いている.

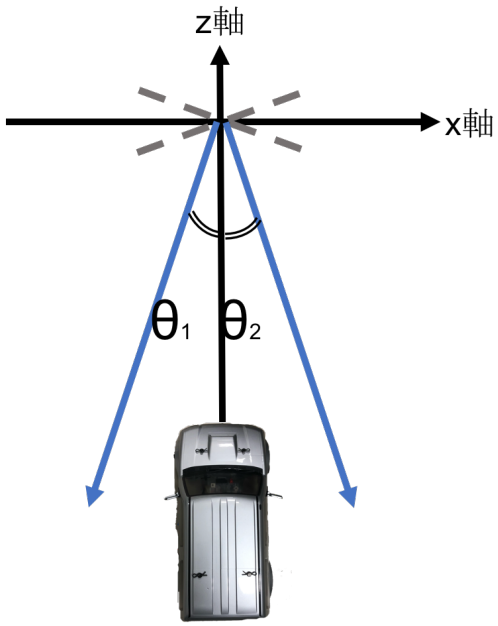


図 4.11 フィッティングした平面の向き推定

5 章　運転アルゴリズムの開発と自動走行

5.1　運転アルゴリズム

障害物検知の結果を用いて模型自動車の走行を制御する.前輪タイヤのステアリング角度を制御する数値は，中央を0，左折を行う場合は正の整数，右折を行う場合は負の整数を設定する．ステアリング角度に設定できる数値の範囲は -15〜15 である．また，前進後退を制御する数値は，ニュートラル状態を0，前進を行う場合は負の整数，後退を行う場合は正の整数を設定する．実際に前進後退する数値の範囲は-30〜-12，12〜30 であった．右折か左折する場合には，-12 あるいは 12 の数値では前進後退しなかった．絶対値が大きくなるほど高速に移動する.

基本的な制御ルールを表 5.1 のように設定した.

表 5.1 模型自動車の制御ルール

条件		ステアリング角度	制御内容
障害物がある	平面の角度 θ　　90°- θ	-11°　～　-70°	右折
		11°　～　70°	左折
	避けられない角度	-10°　～　10°	後退
障害物がない		ー	前進

ただ，この条件だけでは角や壁に対して直進した際に行き詰まってしまう可能性がある．解決方法として，後退の処理を 3 回連続で行った時にステアリングを左に切りながら後退することで切り返しを図ることにした.このアルゴリズムを利用して，障害物を回避しながら廊下を走行させる.

5.2　障害物検知の実験結果

提案手法を検証するため，模型自動車の走路上に障害物(図5.1)を設置し，自動運転させた．

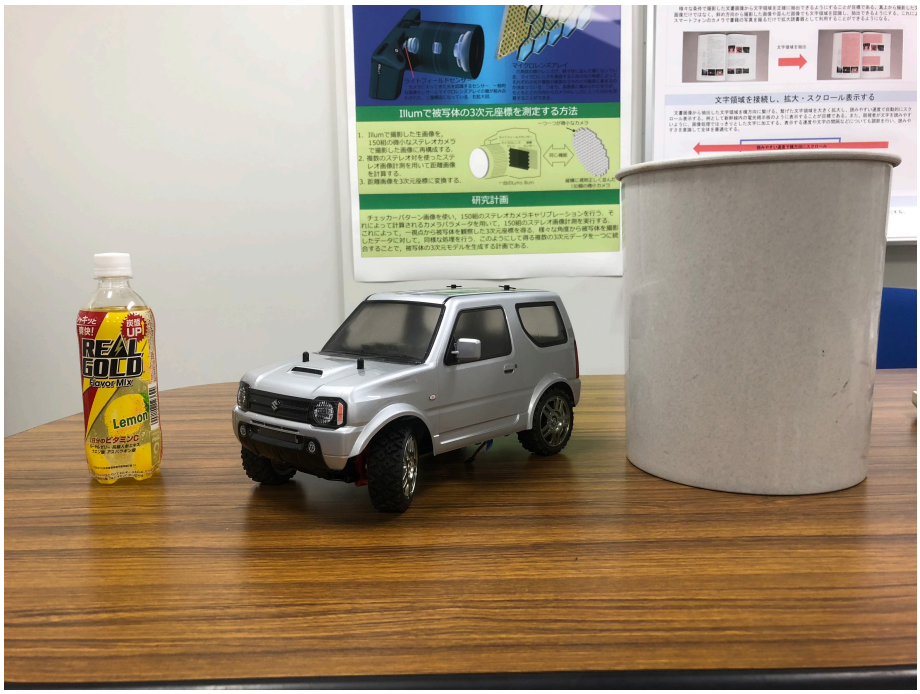


図 5.1 使用した障害物の一例

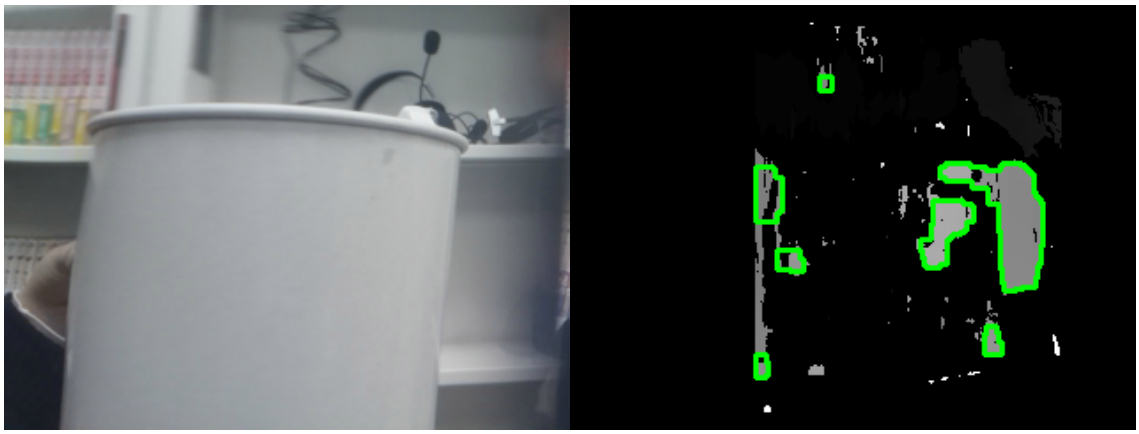
障害物には，色のメリハリが強いもの，色のメリハリは少ないが輪郭のはっきりしているもの，色も輪郭もはっきりしないものを選んだ．これらの障害物を模型自動車の進行方向に置き，自動制御によって避けることができることを検証する．障害物は1度に全てを設置するのではなく，1つずつ分けて確かめた．以後，その検証結果について記述する．

進行方向に模型自動車を設置した場合，障害物と判断し回避することができた．実際の検出画像（図5.2）を見ると，クリアパーツを使用している部分以外は，障害物の輪郭をマーキングできている．推定した中心座標(図5.2赤丸部)も障害物である模型自動車の中心を捉えられており，制御を行うには十分の精度と考えられる．クリアパーツ部には，向こう側の透過像と表面での反射像が重畳しており，ステレオマッチングが難しいことがわかる．また，車体表面のテクスチャが少ない部分も，ステレオマッチングに失敗している．

原画像の一つ　　障害物をマーキングした画像

図 5.2 障害物の検出(模型自動車)

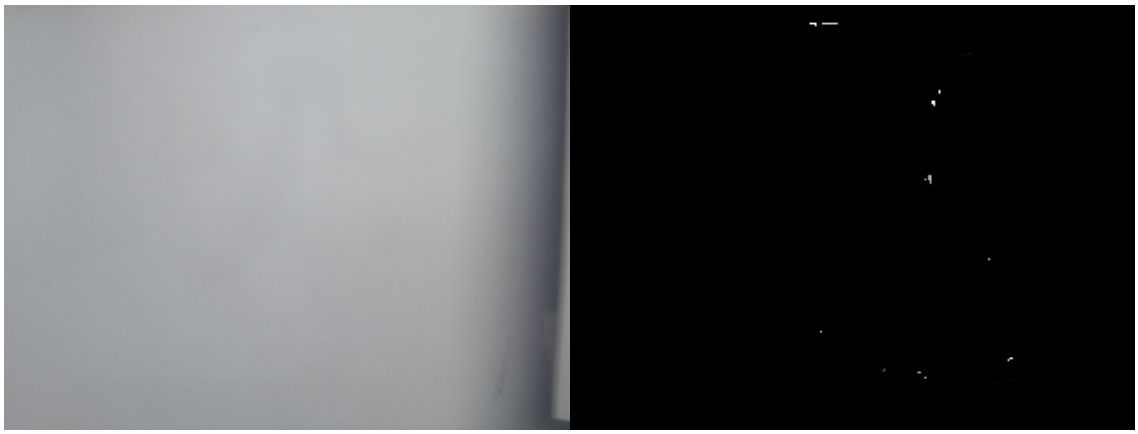
ゴミ箱を障害物にすると, 検出できず衝突してしまう場合が多かった. 検出時の画像を見ると, 障害物であるゴミ箱を正しく把握できていないことがわかった. 部分的には領域を検出しているが, 障害物と判断できるほどの面積ではないため障害物と判定されない. 原因は, 物体表面に濃淡変化が少ないことである.



原画像の一つ　　障害物をマーキングした画像

図 5.3 障害物の検出(ゴミ箱)

図 5.4 は壁に対して障害物検出を行なった時の画像である. 見てわかるように, 特徴になりうる要素がないため, 距離画像が正しく生成されず, その結果, 障害物の検出に失敗した. ゴミ箱の場合と同様, 特徴となる部分が少なかったために不正確な検出になってしまった. この結果は, 障害物を検出し回避を行いながら廊下を一周させるという実験目標に置いてカメラセンサのみでは達成が困難であることを示している.



原画像の一つ　　障害物をマーキングした画像

図 5.4 障害物の検出(壁)

5.3　自動走行に関する実験結果

本研究の目標にしたがって,14 号館 2 階の廊下を自動走行で一周させた．今回は，右回りに走行することを想定して実験を行なった.ただし,壁に直進していった場合，チェスボード画像を貼ったボードを立てかけ，擬似的に壁に特徴を与えた．これは，特徴のない壁を障害物として検出できるようにするために行った．図 5.5 にその時の動作の一例を模式的に示す.

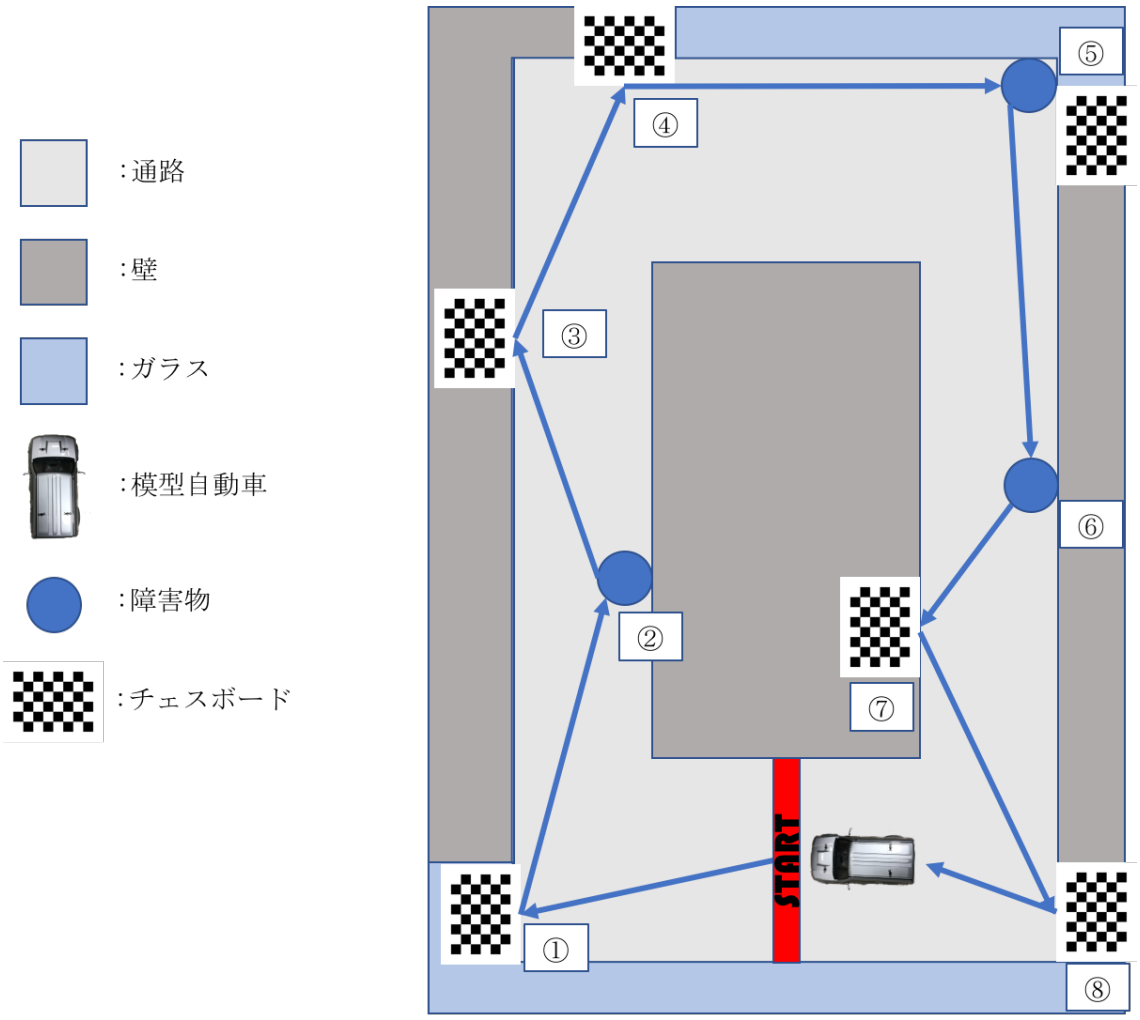


図 5.5 廊下を一周させた時の動作経路

図 5.5 内の番号に沿って実験結果の説明を行う．模型自動車がスタートし①に差し掛かった時，壁がガラスだったためチェスボードを使用し特徴を与えた．回避行動としては，切り返しを行った．②で障害物に差し掛かると，模型自動車はステアリングを左に切り回避した．③でまた壁に差し掛かったのでチェスボード画像を使用し，右へ回避した．④も同様にしてチェスボードを使用して切り返しによって回避した．⑤では，植木鉢があったが鉢植えに特徴があまりないことと，後ろの壁がガラスだったために障害物として検出できず，チェスボードを使用し，切り返しを行うことでここを回避した．⑥では，張り紙のされたゴミ箱を障害物として検出し，自動的に右に避けた．⑧は壁がなく道になっており，そのまま直進すると一周することができないので，チェスボードで擬似的に壁を作った．結果，模型自動車は右折しスタート地点に戻ってくることができた．

一周することはできたが，やはり単色の壁やガラス，ゴミ箱などを障害物として検出するのは難しく、こちらの手助けなしでは自動制御は成り立たなかったため，実験成功とは言えない.

5.4 考察

単色で特徴がない壁などを除いて物体を障害物として検出し，それらを回避することができた.物体表面に特徴がない障害物は，ステレオカメラ以外のセンサを併用するなどの必要がある．例えばLIDARを用いて壁を検出するであるとか，何らかの光パターンを投影してステレオ計測を補助するなどの方法が考えられる.今回の研究で，現在の自動運転技術は様々なセンサによって支えられており，またその技術がいかに高度であるかを実感した.

模型自動車の自動走行に関しては，無駄のないように障害物を避けることができなかった.人間が行う追い越し運転を自動的に行うには，障害物の横幅や奥行きを検出し，無駄の無い走行を行う必要がある.

6章　結論

仮想的に2台分のカメラ視点を再現するシングルカメラステレオ試作し，これを接続したRaspberry Piを模型自動車に搭載して自動運転を試みた．撮影した画像からリアルタイムで距離画像を生成し，その画像から障害物を検出する．検出した障害物の情報から,模型自動車を制御し自動走行させた．物体表面に濃淡変化がある物体は障害物として検出し，自動的に避けることができた．しかし，単一色の壁やゴミ箱を障害物として検出することができなかった．また，障害物検出する前に最低検出範囲よりも近づいてしまい，そのまま衝突してしまう場合があった．ここから得られる課題点は，カメラセンサ以外のセンサと併用して自動制御を行うことと，1処理の処理速度を上げることであると言える．

今回の研究のように模型自動車やRaspberry Piを用いることで，本来なら高い研究費・リスクや人員がかかる実験でも，ローコストで安全に行えることがわかった．これより，Raspberry Piは，ローコストで実験を行うことができ，これから小学校で必修科目になるプログラミングの授業や，組み込み等の基本技術を学ぶにあたって非常に利用価値の高いものであることがわかった．

参考文献

[1] 松ヶ谷和沖,「自動運転を支えるセンシング技術」, DENSO TECHNICAL REVIEW Vol.21，pp.13-21，2016 年.

[2] 青木啓二,「自動運転車の開発動向と技術課題」, 情報管理 Vol.60, No.4, pp.229-239, 2017 年.

[3] 実吉敬二,「ステレオカメラによる自動運転の可能性」，電子情報通信学会誌 Vol.100，No.11，pp.1309-1315，2017 年 11 月.

[4] W. Lovegrove and B. Brame, “Single-camera stereo vision for obstacle detection in mobile robots,” Proc. SPIE 6764, p. 67640T, 2007, doi:10.1117/12.732651

[5] 竹内貴哉,「Raspberry Pi を用いた模型自動車の制御　−光切断法による障害物検知と自動運転−」，コンピュータ理工学部特別研究報告書，2017 年 1 月.

謝辞

本論文を作成にあたり,丁寧な御指導を賜りました蚊野浩教授に感謝いたします.

付録　本研究で作成した自動運転のプログラム

	内容
pim_conste.cpp	シングルカメラステレオを装着したカメラモジュールから得られた画像に対し，画像分割を 1/4 サイズで行う．分割した画像の中から，距離画像生成可能な右画像と左画像に対し，歪み除去・平行化を行った後，ブロックマッチングを用いて距離画像を生成する．生成した距離画像から障害物検出・位置の推定を行い，その情報から模型自動車の前進後退・ステアリングを制御するプログラム．