

公立はこだて未来大学 **2024年度 システム情報科学実習**
グループ報告書
Future University Hakodate 2024 Systems Information Science Practice
Group Report

プロジェクト名
Make Brain Project
Project Name
Make Brain Project

グループ名
視覚を持つ AI カー
Group Name
AI car with vision

プロジェクト番号
/ Project No.
21-C

プロジェクトリーダー / **Project Leader**
狩野政宗 Masamune Karino

グループリーダー / **Group Leader**
畠山出海 Izumi Hatakeyama

グループメンバー / **Group Member**
畠山出海 Izumi Hatakeyama
藤巻侑暉 Yuki Fujimaki
今枝和暉 Niki Imaeda
幸才琉紀 Ryuki Kosai
佐々木優吏 Yuri Sasaki

指導教員
香取勇一 栗川知己 加納剛史
Advisor
Yuichi Katori Tomoki Kurikawa Takeshi Kano

提出日
2025年1月99日
Date of Submission
January 99, 2025

概要

近年、人工知能や自動運転技術の研究が進展し、交通安全向上への期待が高まっている。交通事故は深刻な社会問題である。日本における過去10年間のデータを通じて、交通事故件数は減少傾向の後再び増加傾向が見られ、死亡者数は年によって変動している。令和3(2021)年から令和5(2023)年の間で、交通事故件数は39,000件から37,000件へ減少し、令和5年には死亡者数が前年よりも68人増加して3,573人となっている[1]。

本研究では、Raspberry Pi5を搭載したラジコンカーを用いて、画像認識と機械学習を活用した「自律走行システム」の開発に取り組んだ。自律走行は、交通事故の主な原因となるヒューマンエラーの低減や安全性の向上に寄与する技術として期待されている。本研究の目的は、人間の脳が担う運転操作を人工知能で再現し、交通規則を遵守しながら目的地までの最適な経路を自ら選択・走行できる自律運転の実現を目指すものである。

本研究で制作したシステムでは、格子状に構成された走行ルートを幅優先探索アルゴリズムを用いて解析し、最適な道順を自動で選択する機能を実装した。さらに、カメラによる画像認識を活用し、停止線や一時停止、進入禁止標識といった交通規制を正確に認識して走行制御を行った。

実験にはコストや安全性の観点からラジコンカーを簡易モデルとして採用し、自律走行技術を検証した。具体的な取り組みとして、ハードウェアチームはラジコンカーの組み立て・改造やカメラモジュールの設置、走行コースの構築・作成を担当した。一方、ソフトウェアチームはPythonを用いた画像認識アルゴリズムの開発や幅優先探索アルゴリズムの実装、カスケード分類器を活用した交通標識認識に取り組んだ。

キーワード 自動運転, 画像認識, 機械学習

(※文責: 佐々木優吏)

Abstract

In recent years, research into artificial intelligence and automated driving technology has advanced, raising expectations for improving traffic safety. Traffic accidents are a serious social problem. Data for the past 10 years in Japan show a downward trend in the number of traffic accidents followed by an upward trend again, and the number of fatalities has fluctuated from year to year. Between 2021 and 2023, the number of traffic accidents decreased from 39,000 to 37,000, while the number of fatalities increased by 68 to 3,573 in 2023.

In this research, an “autonomous driving system” was developed using a radio-controlled car equipped with a Raspberry Pi5, utilizing image recognition and machine learning.

Autonomous driving is expected to be a technology that contributes to reducing human error, the main cause of traffic accidents, and improving safety. The objective of this research is to realize an autonomous driving system that can reproduce the driving operations of the human brain using artificial intelligence, and can select and drive the optimal route to the destination by itself while complying with traffic regulations.

The system created in this research analyzed a grid of driving routes using a width-first search algorithm and implemented a function that automatically selects the optimal route. Furthermore, the system utilized camera-based image recognition to accurately recognize traffic regulations such as stop lines, temporary stops, and no-entry signs for driving control. For cost and safety reasons, a radio-controlled car was used as a simple model for the experiment to efficiently verify the autonomous driving technology. Specifically, the hardware team was responsible for assembling and modifying the radio-controlled car, installing the camera module, and building and creating the driving course. The software team, on the other hand, developed an image recognition algorithm using Python, implemented a width-first search algorithm, and worked on traffic sign recognition using a cascade classifier.

Keywords: autonomous driving, image recognition, machine learning

(※文責:佐々木優吏)

1.1 背景.....	4
1.2 研究動機.....	4
1.3 目的および重要性.....	5
第2章 関連研究.....	5
2.1 関連研究.....	5
2.2 必要なスキル.....	6
2.3 実装手法.....	7
2.3.1 白線認識.....	7
2.3.2 標識認識.....	7
2.3.3 目的地までの自律走行.....	7
第3章 プロジェクト学習の最終的な目標.....	7
3.1 ライン (停止線, 白線) 認識機能の実装.....	7
3.2 標識認識機能の実装.....	7
3.3 目的地までの自律走行機能の実装.....	7
第4章 目的を達成するための手法, 方法.....	8
4.1 考案したアイデア.....	8
4.1.1 白線 (今回は黒線だが, 以後白線と記述) 検出システム.....	8
4.1.1.1 輝度値差による検出.....	8
4.1.1.2 オブジェクト判定.....	10
4.1.2 標識検出システム.....	11
4.1.3 ルート検出システム.....	11
4.2 新しい解決方法や手段.....	12
第5章 結果.....	13
5.1 主要な結果.....	13
5.2 各チーム毎に得られた結果.....	13
5.2.1 ソフトウェア.....	13
5.2.2 ハードウェア.....	13
第6章 考察.....	14
参考文献.....	15

第1章 はじめに

1.1 背景

現在, 世界中で自動運転技術の研究開発が急速に進んでいる. 交通事故の状況は年々変化し, 深刻な社会問題となっている中で, 自動運転車は将来のモビリティ社会において重要な役割を果たすと考えられている.

過去10年間, 交通事故の件数は2013年から2023年にかけて, 約50万件から約31万件と全体的に減少傾向にある. しかし, 2023年には交通事故発生件数が7,091件 (前年比2.4%増), 死者数が68人 (前年比2.6%増), 負傷者数が8,994人 (前年比2.5%増) であった[1][2]. このように, 交通事故は減少傾向が続く一方で, 年によっては増加の兆候も見られ, 持続的な対策が求められている. さらに, 高齢ドライバーに関する課題も深刻である. 2023年のデータによれば, 交通事故死者数のうち54.7%が高齢者で占められており, 高齢化社会の進行に伴い, この傾向が続くことが懸念されている. 特に, 日本では高齢ドライバーが関与する事故が多く, 交差点での衝突や追突事故, 右左折時の衝突といった重大なケースが報告されている.

このような課題において, 人工知能技術は問題解決の鍵として注目されている. 人工知能技術を活用することで, 自動運転車は周囲の環境を的確に認識し, 人間のように状況判断を行い, 適切な行動を取ることが可能になる. 具体的には, 画像認識を用いた交通標識の検出や物体認識, 経路の最適化など, AIのアルゴリズムが自動運転の精度と安全性を大きく向上させることができると考える.

(※文責: 佐々木優吏)

1.2 研究動機

過去のプロジェクト学習に自動運転車を作成した事例[3]があったため, 報告書からその詳細を調査した. その結果, その自動運転車はカメラで取得した路面の形状を車体制御で維持することでサーキットを周回走行できるように機械学習を行うというものだった. 我々はここに発展できる余地があるという結論に至り, この自動運転システムの目的をサーキット周回から実際の道路走行を可能にする方向に改変しようと考えた.

(※文責: 畠山出海)

1.3 目的および重要性

自動運転技術は交通事故の根本的な予防と安全性向上における重要な解決策として期待されている. 本研究「視覚を持つAIカー」は, 機械学習と画像認識技術を活用し, 交通事故のリスクを低減するためのアプローチを探求することを主な目的とする.

本研究の背景から得られる自動運転の課題は以下の3点に集約される:

1. ヒューマンエラーの削減 交通事故の主要因である安全不確認, 脇見運転, 操作ミスなどのヒューマンエラーを技術的に克服することを目指す. AIによる高精度な状況認識と瞬時の判断により, 人間の運転に伴う不確実性を最小限に抑える技術の開発を目指す.
2. 運転困難者の移動支援 高齢者や運転に制約のある人々の移動の自由を確保するため, 自動運転技術による支援システムを研究する. 加齢による認知機能低下や身体機能の衰えを補完し, 安全で快適な移動環境を技術的に実現する方法を探求する.
3. 自動運転技術の実践的検証 raspberry pi5とラジコンカーを用いた小型自動運転システムの開発を通じて, 画像認識, 機械学習, リアルタイム走行制御の実践的な技術検証を行う. 具体的には, カメラによる状況認識, 幅優先探索, 標識認識, 白線・停止線認識などの技術的課題に取り組む.

これらの課題を解決するため、本研究では以下の技術的アプローチを採用する：

- 画像認識技術の開発
- 機械学習による状況判断アルゴリズムの実装
- 自動ブレーキおよび走行制御システムの設計

本研究は、単なる技術開発にとどまらず、交通安全に対する人間の脳で判断に近い論理的な自律走行アルゴリズムの実装を解決策として提示し、より安全で包括的な交通環境の実現に貢献することを最終的な目標としている。将来的には、本研究の成果が自動運転技術の発展に寄与し、交通事故の根本的な予防につながることを期待している。

(※文責：佐々木優吏)

第2章 関連研究

2.1 関連研究

本研究では、視覚認識を備えたAIカーの開発を目指した「Make Brain Project」(公立はこだて未来大学2021年度システム情報科学実習報告書)を先行研究として参照する[3]。このプロジェクトは、交通事故の主因であるヒューマンエラーを削減することを目的とし、Raspberry Pi 4を搭載したラジコンカーを用いて機械学習と画像認識技術を応用した自動運転システムを構築した。具体的には、AIカーに白線認識、障害物認識、道路標識認識の機能を実装することを目標とし、カメラモジュールによる画像データの取得とPythonによるプログラミングを中心に開発が進められた。白線認識では、PID制御を用いて走行時の安定性を確保し、障害物検知には超音波センサーを活用した。また、道路標識認識にはカスケード分類器を用い、停止標識を対象としたモデルの開発が行われたが、学習データの収集や分類器の精度向上に課題が残り、対応可能な標識は限定的であった。最終的に、白線に沿った自走、停止標識の認識、障害物の検知といった基本的な動作を実現するAIカーが完成したが、特定の環境での走行に限定され、より複雑なコースや多様な状況への適応は今後の課題として示された。

この成果を踏まえ、本研究「視覚を持つAIカー」は、自動運転技術の実現に向けた一歩として、ラジコンカーをベースにした小型自動運転車の開発を目指す。この研究では、Pythonを用いた画像認識技術や機械学習を活用し、車両が自律的に走行経路を判断し、交通ルールを認識できるシステムを構築する。特に、車線追従や障害物回避、停止標識の認識といった基本的な機能の実現に注力し、人間の運転ミスを補完する技術を追求する。

(※文責：佐々木優吏)

2.2 参考にしたスキル

1. Opencvによる画像処理 (白線認識)

白線間でのラジコンのサーボを制御するため逆透視変換を用いる。逆透視変換は遠近法によって台形として写る領域を長方形に変換して表示する。

2. 標識認識

例年のプロジェクトで使用されたカスケード分類器を使用する。カスケード分類器は処理速度が速く、ディープラーニングと比べ、少ない訓練データで認識を行うことができる。

(※文責：藤巻侑暉)

2.3 実装手法

2.3.1 白線認識

Alcarlは道路の白線を認識し、その情報を基に車両の位置を適切に調整することが求められる。そのため本プロジェクトでは画像処理技術を使用して白線を検出する。具体的には、Canny法や、Hough変換、逆透視投影変換を利用する。これにより、車両は正確に車線内を走行する。

2.3.2 標識認識

Alcarlは、交通標識を正しく認識し、それに従って運転する能力が必要である。カメラを用いて画像から交通標識を検出し、認識する。標識認識には、カスケード分類器やディープラーニングを活用することで標識情報を正確に認識し、適切な運転操作を実行することができる。本プロジェクトではカスケード分類器の作成を行い、標識認識を行う。例年のプロジェクトでは10枚の正解画像を使用してカスケード分類器を作成していたが、本研究では、約1100枚の正解画像を用意してカスケード分類器を作成した。

2.3.3 目的地までの自律走行

ヒューマンエラーの削減や運転困難者の移動支援を行うため出発地点から目的地までの自律走行を行えると、コースを模したグリッドマップをプログラムコードで再現し、スタート地点、目的地、通行止め地点を指定したうえで幅優先探索を行うことでスタート地点から目的地までの自律走行を行う。

(※文責：藤巻侑暉)

第3章 プロジェクト学習の最終的な目標

1 ライン (停止線, 白線) 認識機能の実装

画像から道路の白線を検出し、その情報を基にサーボの角度とモーターの回転する速さを調整する機能を実装する。これにより自律的な姿勢制御が可能となる。

2 標識認識機能の実装

カメラからリアルタイムで入力される標識を検出し、認識するシステムを実装する。この機能のためにカスケード分類器を利用する。

3 目的地までの自律走行機能の実装

コード上で作成したコースに近似させたマップ情報を用いて、幅優先探索を行い目的地までの行程における右左折の挙動を記録する。

(※文責：藤巻侑暉)

第4章 目的を達成するための手法, 方法

4.1 システム面での問題解決手法

4.1.1 白線 (今回は黒線だが, 以後白線と記述) 検出システム

走行している路面について搭載したカメラからフレームとして情報を入手し, それを処理する形をとった. この時の処理に際して後述するシステムに適しているという理由から, 取得した画像データを透視変換を用いて白線が平行に並ぶように画像を加工した. (図4.1) 今回, 路面に用いる色を白と黒に限定し, 線を黒, その他路面を白と設定した. このようにすることで透視変換を行った際に得られるフレームを処理しやすくなる.

透視変換の後は加工した画像から白線を検出する処理を行う必要があるが, この時の処理に関して2通りのアプローチを設けた.

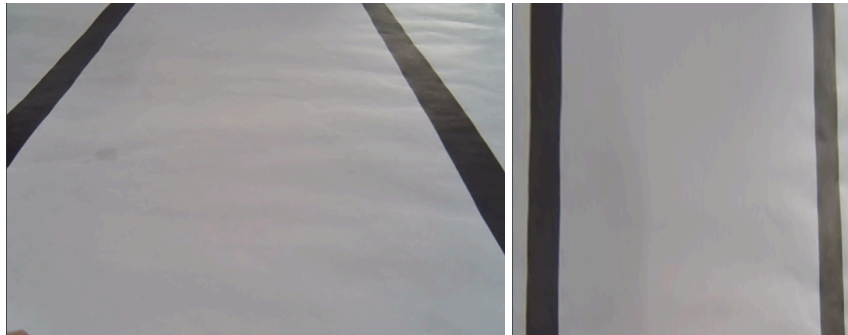


図4.1 カメラで取得したフレーム (左) と透視変換後 (右)

4.1.1.1 輝度値差による検出

この処理では, 画像内をx座標で区切り領域を設定する. フレームに対して1画素ごとに縦に分割, それぞれの行の画素群に対して輝度値の平均値をリスト化した. この時, 取得した平均値のリストには必ず数値が乱高下している部分が発生する (図4.2). その部分を境界と定め, 輝度値が急上昇する境界から急降下する境界までの領域を走るべき路面の領域と判断する処理を行った (図4.3).

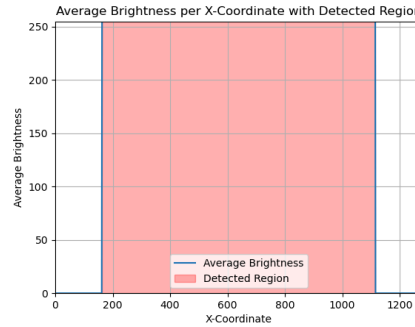


図4.2 各x座標における輝度値平均のグラフ 元データはテスト用画像

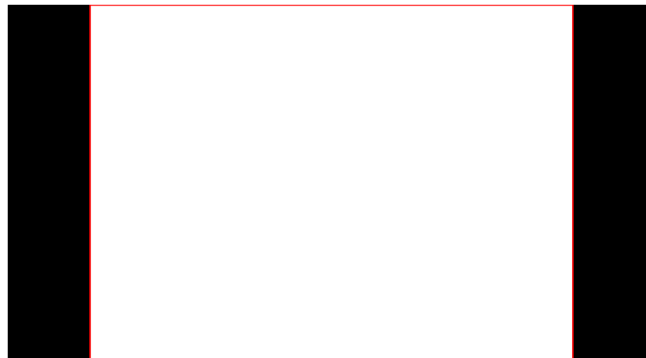


図4.3 路面の領域を図示した画像

さらに, この処理を車体が白線に対して傾いている場合も有効にするべく処理の追加を行っている. 車体が白線に対して傾いている場合, 透視変換後のフレームに白線は傾いたものが1本しか映らない. この時は境界が1つになるため路面の領域が白線に対して両側に検出できてしまう.

これを防止するために輝度値を用いた白線の傾き検出が必要になったので, 境界のx座標における上側半分と下側半分の輝度値の平均を比較するシステムを構築した. 傾いている場合, 赤線で示した境界 (図4.4.1) において必ず上下での輝度値差が生まれる. このことを利用し, 上下での輝度値差が下に偏っていた場合は左に, 上に偏っていた場合は右に車体が傾いていると定義し, 併せてその輝度値差の数値によって傾きの度合いも見ることによってサーボモーターを適切に動かすことができるようにした. 輝度値の変化や比較の基準には実験で求めた閾値を設定した.

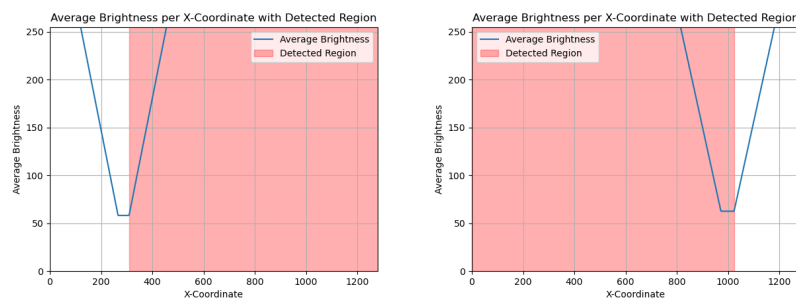


図4.4.1 車体が左に傾いている想定 of 各x座標における輝度値平均のグラフ (左), 車体が右に傾いている想定 of 各x座標における輝度値平均のグラフ (右)

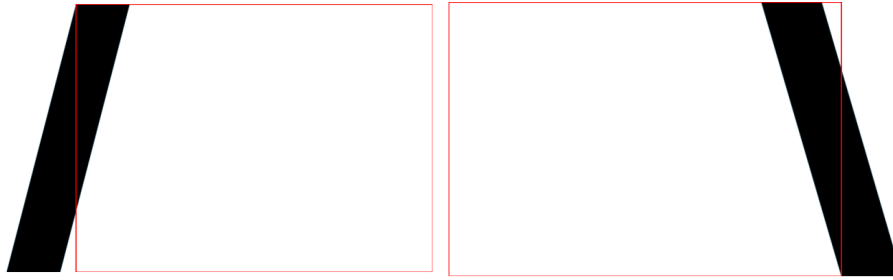


図4.4.2 車体が左に傾いている想定領域図示 (左), 右に傾いている想定領域図示 (右)

この手法をとるメリットとして、ノイズに対して強い耐性を持つことがある。輝度値の大小関係を用いるという特性上、多少のノイズでは判定が覆ることがない。そのため、正常に機能している限り動作の安定性が保証される。

4.1.1.2 オブジェクト判定

この処理では、画像内の白線をオブジェクトと認識することでその輪郭を描画、フレームに対する輪郭線の傾きを測定することで車体の姿勢を制御する (図4.5)。輪郭の検出にはOpenCVの標準機能を用いた。

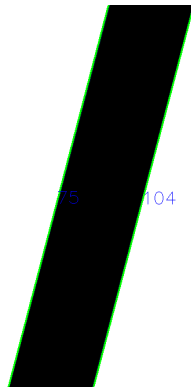
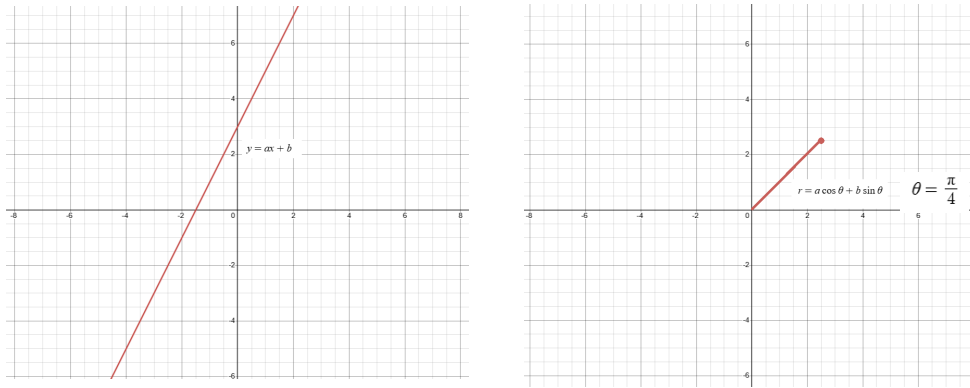


図4.5 白線から輪郭を検出、角度を描画した画像

この標準機能ではハフ変換を用いた直線検出の応用を行っている。ハフ変換は画像処理やコンピュータビジョンで直線や円などの形状を見つけるためのアルゴリズムであり、主にエッジ検出を終えた画像から特定の形を効率よく探すのに使われる。ハフ変換では各点で作る可能なすべての直線を計算して、それをパラメータで表現する。通常、直線は $y=ax+b$ という傾きと切片で表されるが、この方法は垂直に近い線を扱うときに不安定になる。そのため、 $p=x\cos\theta+y\sin\theta$ という極座標系を使うのが一般的で、直線を距離と角度で表現する (図4.6)。画像のエッジ上にある各点について、可能性のあるすべての p と θ を計算し、それをパラメータ空間に記録する。多くの点が交差する場所が見つかった場合、それが画像内の直線に対応していると判断される。これを用いて画像内に存在する直線を検出するのだが、繰り返していると検出した直線で囲まれた領域が発生する。その領域をオブジェクトとして認識するのがこの標準機能の概要となる。

輪郭線の傾きの基準はフレームの横軸である。その傾きを用いてサーボモーターを動かす角度を決定する。

この手法を用いるメリットとしては、計算量が少ないため他の処理の邪魔をすることなく動作する点がある。用いているのがパッケージの標準機能のみであるため、過剰な計算処理が不要となりスムーズな動作が可能となる。



(図4.6) 傾きと切片で表される直線 (左) 極座標で表される直線 (右)

以上の2つのアプローチを考案したが、今回のrasberrypi5のCPUで1つ目の輝度値差による検出処理を行ったときかなりの処理落ちが発生、検出のリアルタイム性が失われていた。さらにデバッグ時にウィンドウ上での表示が追いつかなくなっていたことにより動作不具合の原因特定が難しく、修正に時間を要することになってしまった。そのため今回は、動作保証がありデバッグがしやすい2つ目のオブジェクト判定を採用した。

4.1.2 標識検出システム

標識検出に際しては、カスケード分類器を用いたオブジェクト検出システムを採用した。

カスケード分類器とは、機械学習を用いて画像内の特定のオブジェクトを検出するための手法である。この分類器は物体の特徴を捉えた特徴量をもとに、画像内の特定領域が対象物を含むかどうかを判断する。ここでの特徴量とは、画素ピクセルの明暗パターンや様々な形状パターンを複合的に判断するHaar特徴量であり、この特徴量が一致するパターンが入力画像内にあるかどうかを探索するのがこの分類器である。特徴としてかなり大まかな探索が可能であることが挙げられ、対象のオブジェクトの特徴量だけを参照しているためオブジェクトが歪んでいたとしても検出することが可能であり、その代わりに誤検出の可能性がある。

今回は「止まれ」と、「進入禁止」の標識に関してそれぞれカスケード分類器を作成、搭載したカメラで検出作業を行い(図4.7)、検出次第割り込み処理で動作を行うようにプログラムを行った。

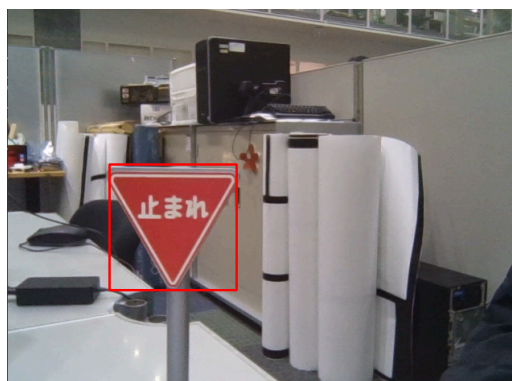


図4.7 「止まれ」標識の検出

4.1.3 ルート検出システム

自動走行にする都合上、地図情報についての処理もシステムで行う必要があった。そのため、地図上の現在地と目的地を入力することで最短ルートを探検、その走行過程における車体の右

左折の挙動を記録するシステムを考案した。現在地を入力を行う必要があるのは、今回はGPSを用いていないため情報取得の方法が存在しないからである。

今回用いたのはグリッドマップによる探索である。グリッドマップとは格子状の地図であり、ノードとエッジに属性を付与できる点から汎用性が高い地図の形式となっている(図4.8)。この属性付与ができる特性を使えば、単純な格子状でない地図にも対応可能である。

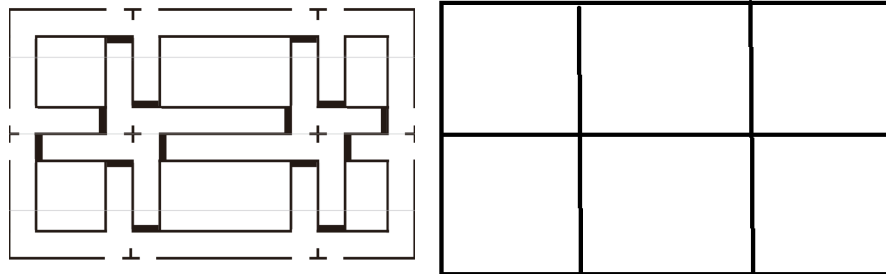


図4.8 今回使用したマップ (左), マップをグリッドマップに近似したもの (右)

探索に用いたのは重み付きの幅優先探索である。エッジにマップ上の距離で重みを付与し、最終的な重みが最小になるルートを実際の分岐点から順に探索する。探索の過程では、分岐点のたびに挙動についての命令を保存する必要がある。そのため方角を基準として進行方向を管理し、分岐点で行うのが直進か、右左折かをリストに保存する形で実装した。例を挙げると、現在の進行方向が東向きであり、分岐の後の進行方向が南向きであるとき、分岐点では右折をする必要がある。

ここで一つ工夫したのが、一方通行にもかかわらず後退する必要がある場合の探索についてである。グリッドマップの属性付与では特定のエッジを片側のみ通行不可能に設定することも可能であり、今回のマップ作成でこれを利用した形状のものが候補に挙がっていた。このプロセスでは通行不可能なエッジを通行するルートが検出された場合に代替の回避ルートの探索を行うが、今回はこれを再帰的な探索で実装することに成功した。プログラムの起動時、まず最初に始発点と終着点の間の探索でルーティングを行うが、片側が通行不可になっているエッジが候補になる場合がある(図4.9)。このとき、この衝突を検知した時点の現在地のノードと次に行くべきノードの間で同様の探索を行う。隣り合うノードのため最短は真っ直ぐ後退するルートだが、一方通行の設定がなされているためそこは一時的に障害物と定義、回避ルートの探索に入る。探索が終了次第、元の探索を継続する。この機能があることで、走行中に急に道路状況が変わったり終着点を変更した時にも対応が可能となる。

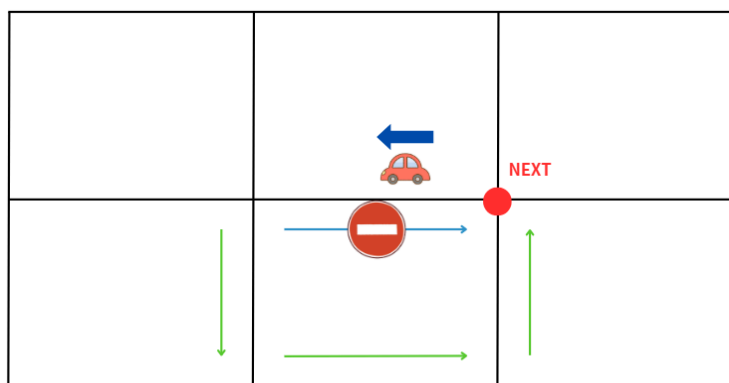


図4.9 後退時の回避探索

(※文責: 畠山出海)

4.2 ハードウェア面での問題解決手法

本研究では、白線を正確に認識するためカメラには道路以外の背景が極力映らないように調整する必要があった。また、標識を認識するにはカメラが少し左方向を向くことが求められた。これらの目的を達成するため、それぞれ異なる役割を担う2台のカメラを用いることとし、それぞれの適切な配置を検討した。これらのカメラを限られたスペース内に設置する必要があり、市販の製品では設置要件を満たせなかったため、自作することとした。初めに、必要な設計を紙に図示し、Adobe Illustratorを用いて詳細な設計図を作成した。その後、レーザーカッターを用いて材料を切り出し、組み立てを行った。当初、白線を認識するカメラは標識を認識するカメラの直上に配置していたが、視野範囲が狭く十分な性能が得られなかったため、より高い位置に再配置した。一方、標識を認識するカメラについては左方向に向ける必要があったが、最適な傾斜角が明確ではなかったため、角度を調整可能な設計を採用した。このような設計により、柔軟かつ効率的なカメラ配置が実現できた。

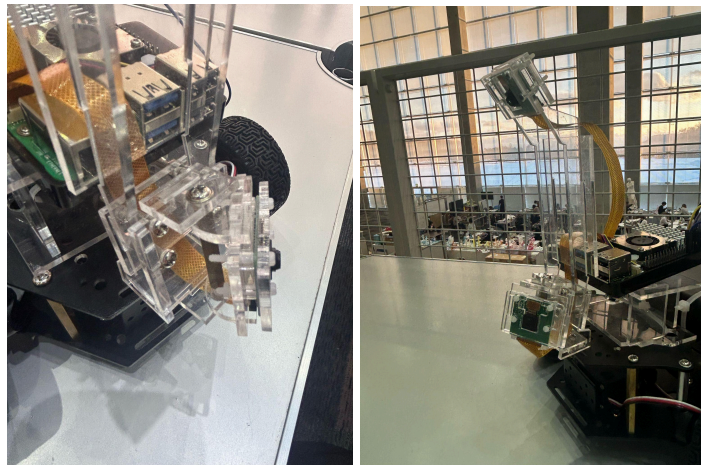


図 4.10 カメラの土台

(※文責: 今枝和暉)

第5章 結果

5.1 主要な結果

作成したマップ上で目的地までライン(白線, 停止線)を認識して走行し, 目的地に到着時自動で停止させることができた。止まれ標識の認識も高い精度で行えた。具体的には, 標識の半分を覆っても正確に止まれ標識を認識し, 白黒画像の止まれ標識も正確に認識した。

(※文責: 藤巻侑暉)

5.2 各チーム毎に得られた結果

5.2.1 ソフトウェア

ソフトウェアグループは、止まれの標識を認識するカスケード分類器の作成、入力画像からライン(停止線、白線)の輪郭と角度を用いたAIカーの安定した走行と停止、幅優先探索を用いた目的地までの自律走行を目標としたが、これらの目標は概ね達成することができた。

5.2.2 ハードウェア

ハードウェアグループでは、実機の組み立てをはじめ、ラジコン車両の走行を可能にするためのコース設計および作成、カメラによる認識を目的とした標識の製作、さらにカメラ本体の設置や位置調整のための土台の作成を主な目標として活動を行った。

これらの目標は概ね達成されたものの、走行に必要な電力を供給可能な走行に必要な電力を供給可能なバッテリーが国内で入手困難であった。さらに、左右のタイヤが正確に同じ方向を向かない状態が生じた。その結果、直線走行において困難を伴う場面があった。そこで、コンセントから直接電力を供給する方法を採用することや、ねじ止めの微調整を何度も行うことにより達成することができた。

(※文責:幸才琉紀)

第6章 考察

良かった点は2つある。1つ目は高精度の標識認識をするカスケード分類器を作成できたことだ。本研究で作成したカスケード分類器は標識の半分を覆っても正確に止まれ標識を認識し、白黒画像の止まれ標識も正確に認識することができた。これにより、経年劣化した標識や雪で覆われた標識、視界の悪い天候時の標識認識を行えると考察できる。

2つ目は道路に似せたマップ上で出発地点から目的地までの安定した走行を行えたことだ。この目的地までの自律走行と高精度の標識認識はヒューマンエラーの削減と運転困難者の移動支援に貢献できると推測される。

課題は大きく分けて3つある。1つ目は出力されるカメラ映像に遅延があったことだ。計算量が多くなったことやWi-Fiの電波が届きづらい場所で実験を行ったことが原因である。映像の遅延によって実験時に問題点の確認をリアルタイムで行えなかったため、制御に時間をかけてしまったりした。

2つ目はAlcarがラインの中心でなく、端を走行する問題があることだ。この問題の原因として、逆透視変換を用いてラインの角度をリアルタイムで計算するアルゴリズムが挙げられる。この処理では、カメラ映像にラインが1本しか映っていない場合と、2本映っている場合とで、サーボモータの制御が適切に行えなかった。実験中に見られた現象として一方のラインがカメラの視野外に出ると、残った1本のラインの角度だけを基準にサーボが制御される結果、車体が片方のラインに近い位置で並行して走行する状態となった。カメラの画角を広げてラインが常に2本映るよう調整する方法も検討したが、その場合、交差点や対向車線のラインが映り込むため、制御が困難になる問題が新たに生じた。

3つ目はモバイルバッテリーによる電力供給ができなかったことだ。コードが重くなってしまったことで実行に必要な電力分をモバイルバッテリーで供給できなかったことによると推測される。

1つ目と3つ目の課題はコードの重さが主な原因であるため、不要な演算を消したり、デバッグ用にカメラ映像を録画することで解決でき、2つ目の課題は前述した通り、カメラ映像にラインが1本しか映っていない場合と、2本映っている場合とで、サーボモータの制御が正しく行えていないことが原因であったため、この制御を正しく行うことで解決できると考察する。

参考文献

- [1] 警視庁. (2023). 各種交通事故発生状況 (令和5年度). 警視庁ウェブサイト. Retrieved from https://www.keishicho.metro.tokyo.lg.jp/about_mpd/jokyo_tokei/tokei_jokyo/vta.html

- [2] 一般社団法人日本自動車工業会. (2023). 交通安全への取り組み. Retrieved from https://www.jama.or.jp/operation/safety/traffic_safety/index.html
- [3] 脳を作るプロジェクト 視覚を持つAIカー. (2021). 公立はこだて未来大学 システム情報科学実習 グループ報告書. https://www.fun.ac.jp/wp-content/uploads/2023/04/document22_A.pdf
- [4] 北山直洋. PythonによるOpenCV4 : 画像処理プログラミング+Webアプリ入門. カットシステム. 2021.
- [5] 吉田顕一. Raspberry Pi + AI 電子工作超入門 実践編. 株式会社ソーテック社. 2021.
- [6] 土田米一. Raspberry Piユーザーガイド 第2版. Eben Upton・Gareth Halfacree. 2014.
- [7] 桑井博之, 富沢聡, 永田雅人. 実践OpenCV 2.4 for Python: 映像処理&解析. カットシステム. 2014.