卒業研究報告題目

視線情報と位置情報を組み合わせた

側方確認動作検知手法の提案

Aaaaaa

Aaaaa

指導教員　田島　孝治　准教授

岐阜工業高等専門学校　電気情報工学科

2013E23　髙木雅哉

平成30年(2018年) 2月xx日　提出

Abstract

目次

[第1章. はじめに 1](#_Toc504948662)

[1.1. 背景 1](#_Toc504948663)

[第2章. 関連技術 3](#_Toc504948664)

[2.1. 自動運転技術 3](#_Toc504948665)

[2.1.1. 先進安全自動車 3](#_Toc504948666)

[2.1.2. 自動運転レベル 3](#_Toc504948667)

[2.2. 視線入力装置 4](#_Toc504948668)

[2.2.1. 視線入力装置とは 4](#_Toc504948669)

[2.2.2. 基本原理 4](#_Toc504948670)

[第3章. 研究概要 5](#_Toc504948671)

[3.1. 目的 5](#_Toc504948672)

[3.2. 提案方式の概要 5](#_Toc504948673)

[第4章. システム概要 6](#_Toc504948674)

[4.1. 設計方針 6](#_Toc504948675)

[4.2. 右左折検知 7](#_Toc504948676)

[4.3. 側方確認動作検知 11](#_Toc504948677)

[第5章. 実装 14](#_Toc504948678)

[5.1. ソフトウェア 14](#_Toc504948679)

[第6章. 評価 15](#_Toc504948680)

[6.1. 実験方法 15](#_Toc504948681)

[6.2. 実験結果 15](#_Toc504948682)

[6.2.1. 側方確認動作の検知 15](#_Toc504948683)

[6.2.2. 右左折の検知 15](#_Toc504948684)

[6.2.3. 車線変更の検知 15](#_Toc504948685)

[6.3. 考察 15](#_Toc504948686)

[第7章. まとめ 15](#_Toc504948687)

[7.1. 研究の成果 15](#_Toc504948688)

[7.2. 今後の課題 15](#_Toc504948689)

[参考文献 16](#_Toc504948690)

# はじめに

## 背景

近年、運転者支援システムが充実してきている。これらのシステムによる支援の内容は、車線の逸脱、急発進防止、追突防止など多岐に渡る。さらに、高速道路での使用を前提としているなどの条件付きではあるが、ドライバーが行っていた操作をシステムに任せることができる技術も実用化されている。これらの現在実用化されている自動運転技術は、部分運転自動化と呼ばれている。

　自動運転技術では、車両の制御のみをシステムが受け持ち、危険を察知した場合にはドライバーの判断で危険を回避するための適切な行動を取る必要がある。そのため、運転支援システムに頼るのではなく、ドライバー自身が運転操作を行う意識を持つとともに、普段の運転と変わらない安全確認を行う必要がある [1]。ドライバーに求められる代表的な安全確認として、車線変更時と交差点進入時の安全確認があげられる。車線変更時は、変更しようとする側の後方を確認する。交差点進入時は、車両と横断歩道を渡ろうとする歩行者の有無を確認する。左折時には、車両の左側をすり抜けようとする自転車や原付に注意する必要がある。これらは側方確認または巻き込み確認と呼ばれ、運転中に車両の進行方向とは別の方向を見る必要があり、運転初心者には難易度が高く、忘れやすい安全確認である。

　現在実用化されている自動運転技術は、直線走行時にのみ使用できる。交差点内では、車両と歩行者、自転車等が混在しているため、交通状況を把握することが難しく利用が困難である。また交差点内では、信号無視や一時不停止などの交通規則の無視や、前方不注意や操作不適などのヒューマンエラーが多く、ある時刻の情報から、未来の行動を予測することが難しい。そのため、全区域で自動運転が可能な、完全自動運転実現への障壁となっている。

さらに交差点内では、ほかの区域に比べて多くの交通事故が発生している。交通事故総合分析センターがまとめた統計によると、信号ありの交差点内で起きた事故が15.5%、信号のない交差点内で起きた事故が23.9%、交差点付近で起きた事故が14.2%である。つまり交差点で起きる事故は全体の約54%を占めていることになる [2]。これは、交差点では注意しなければならない点が多いことが原因として挙げられる。

交差点内では、運転者支援システムによる十分な支援を受けることが難しいため、ドライバーに確実な安全確認を促す必要がある。そのために、ドライバーによる安全確認を検知する手法を確立する必要がある。

# 関連技術

## 自動運転技術

### 先進安全自動車

　先進安全自動車とは、ドライバーの安全運転を支援するシステムを搭載した自動車のことである [3]。様々な運転支援技術が普及しており、その一つとして、先行車との接近を自動で検知し衝突が予測される際には自動で減速をする「衝突被害軽減ブレーキ」がある。この技術は平成27年度で国産乗用車の45.5%に搭載されている [4]。この他にも、一定速度で走行しながら、先行車との車間距離を制御するACC(Adaptive Cruise Control)や、走行車線の中央付近を維持するように制御するLKAS(Lane Keeping Assist System)などが実用化されている。

### 自動運転レベル

　自動運転技術には様々な段階が想定されており、手動運転から完全自動運転までをシステムの介入度に応じて4段階から5段階に分類している。細かな名称や定義は国や団体によって異なっているが、大まかに表2-1のような分類となっている [3]。システムがすべての運転操作を行う、自動運転と呼ばれるのは、レベル3以上である。

　　　　　　　　　　　　　表2-1　自動運転レベル

|  |  |  |
| --- | --- | --- |
| レベル | 名称 | 内容 |
| 0 | 運転者支援なし | 従来の運転 |
| 1 | 運転者支援 | 特定の条件でACC、LKASなどが作動し、  運転操作を個別に支援する |
| 2 | 部分的自動運転 | ACCやLKASなどのシステムが同時に作動し、  車両の制御が行われる |
| 3 | 条件付き自動運転 | 車両の制御と危険察知をシステムが行い、  危険が見つかった場合の応答を運転者に委ねる |
| 4 | 完全自動運転 | 運転者が車両の制御を行うことはない |

## 視線入力装置

### 視線入力装置とは

　視線入力装置とはアイトラッカーとも呼ばれる、目の動きを取得するためのデバイスである。視線検出技術を用いてユーザーがどこを見ているかを推定し、これをコンピュータに入力する [5]。一般的な視線入力装置では、視線を検出するためのハードウェアに、視線情報を使用しコンピュータを操作するためのソフトウェアが同梱されている。

### 基本原理

　視線検出技術では、ユーザーの視線方向を測定するために目の動きを検出する。目の動きの検出は、目を動かしたときに動かない部分と動く部分を検出することから始まる。この動かない部分を基準点、動く部分を動点と呼ぶ。基準点と動点を検出した後、基準点に対する動点の位置に基づいて、視線方向を測定する。この基本的な原理は、どの視線入力装置でも変わらない [6]。

　基準点と動点の選び方には様々な種類があり、使用するハードウェアや目的によって使い分けられている。一般的な視線入力装置で使用されているのは角膜反射法と呼ばれる技術である。弱い赤外線を目に照射し、その反射光の角膜上の位置を基準点とし、瞳孔の位置を動点として視線方向を推定する。この方法では、赤外線の光源と赤外線カメラが必要になるため、装置の規模は比較的大きくなる。また、可視光カメラを使用しての視線検知を可能にする方法もある。基準点を目頭にし、動点を虹彩とする方法である。この方法では、光源を用意する必要がなく、通常の可視光カメラのみで視線検出が可能になるため装置の規模は比較的小さくなる [6]。

　この2つの方法にはそれぞれメリット・デメリットが存在し、前者の方法では、装置の規模が大きくなるというデメリットはあるものの、より高精度な視線検出を可能にする。後者の方法では、装置の規模が小さくなるというメリットはあるが、あまり精度が良くないという特徴がある。

# 研究概要

## 目的

本研究では、視線情報と位置情報を組み合わせて、運転中の側方確認動作をできるだけコストを抑えて検知することを目標とする。これは、初心者ドライバーが実際の運転中に安全確認ができているのかを確認するためである。GPSを使い、運転中の位置情報を取得し、交差点の右左折などの車両の挙動を検知する。また、運転中の視線を計測し、側方確認動作を検知する。

## 提案方式の概要

　本研究で提案する方式の概要を図3-1に示す。提案する方式では視線情報をローカルで利用できるWebカメラとWebGazer.jsによって取得する。またGPSロガーによって運転中の位置情報を取得する。視線情報から側方確認を、位置情報から交差点の右左折と車線変更を検知し、組み合わせることで実際の運転中に正しい安全確認ができているのかを確認する。

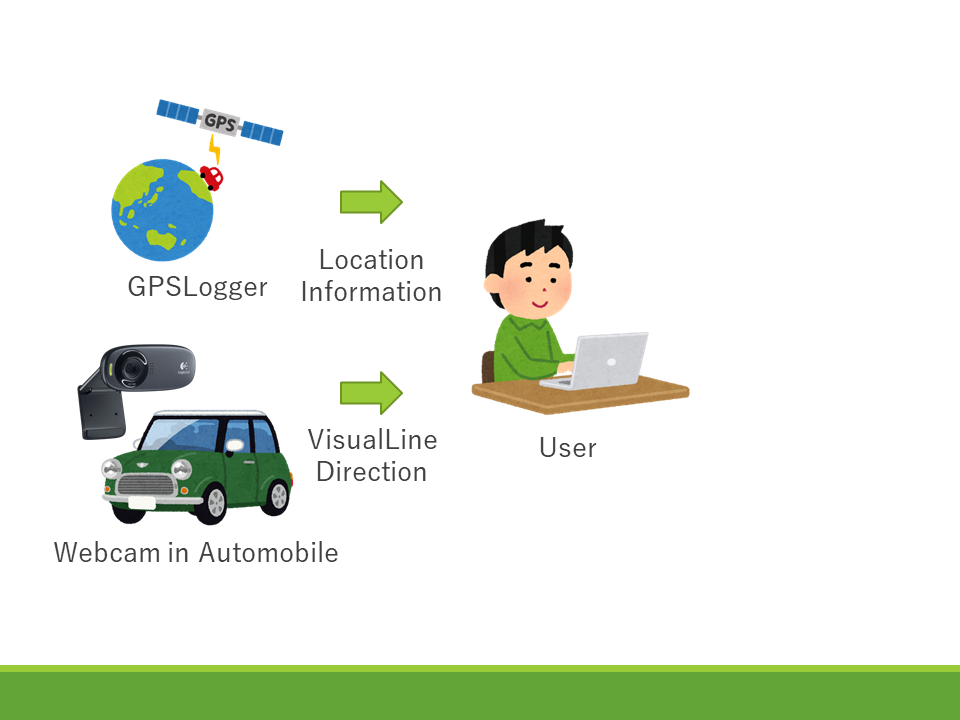


図3-1　システム概要

# システム概要

## 設計方針

本研究では、位置情報と視線情報を取得し、これらを組み合わせることで目的を達成することを目指す。位置情報はGlobalSat社のGD-100 GPS+Data Loggerを使用して取得する。視線情報はlogicool社のHD Webcam c270とWebGazer.jsを使用して取得する。図4-1に取得する位置情報の例を示す。

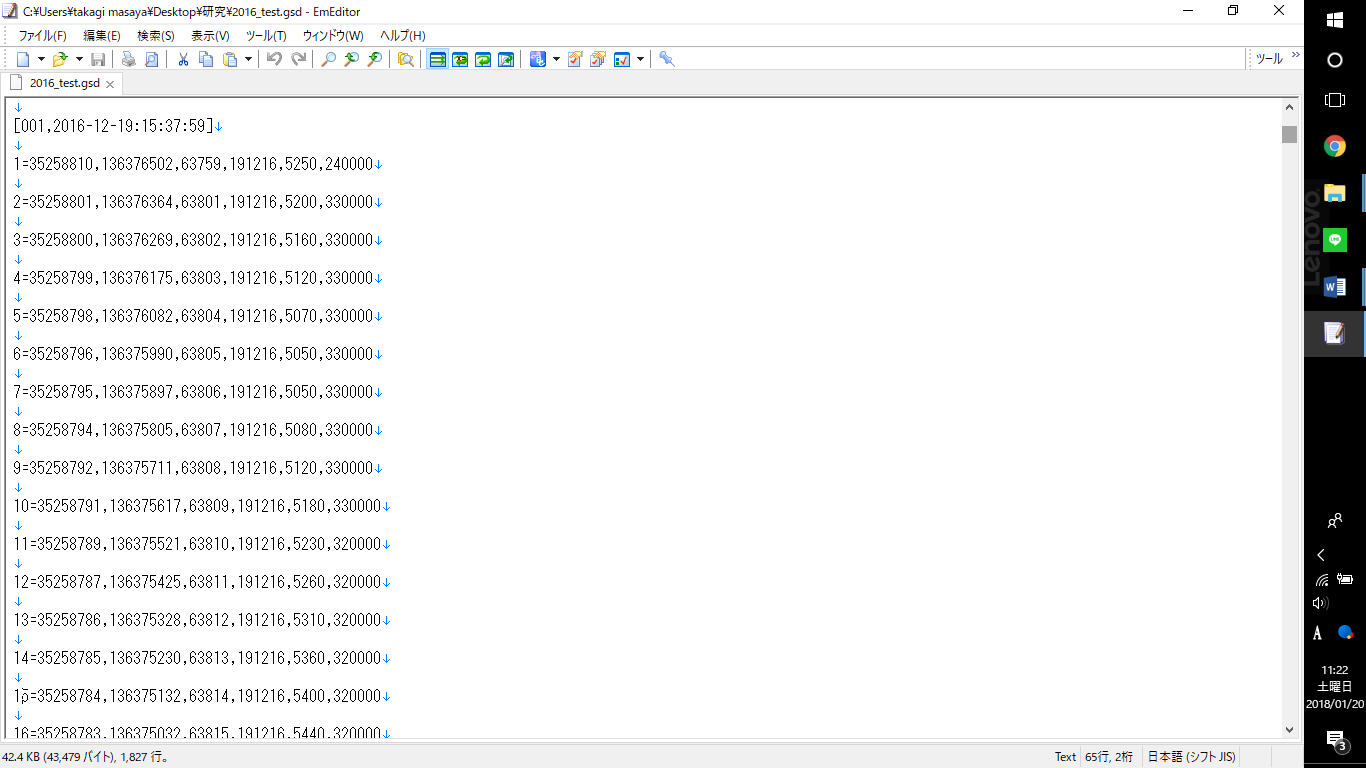


図4-1 ロガーで取得する位置情報の例

## 右左折検知

図4-2に位置情報から右左折を検知するまでの流れを示す。途中、計算した値はSinglePointDataクラスによって管理する。表4-1にSinglePointDataクラスによって管理される値を示す。

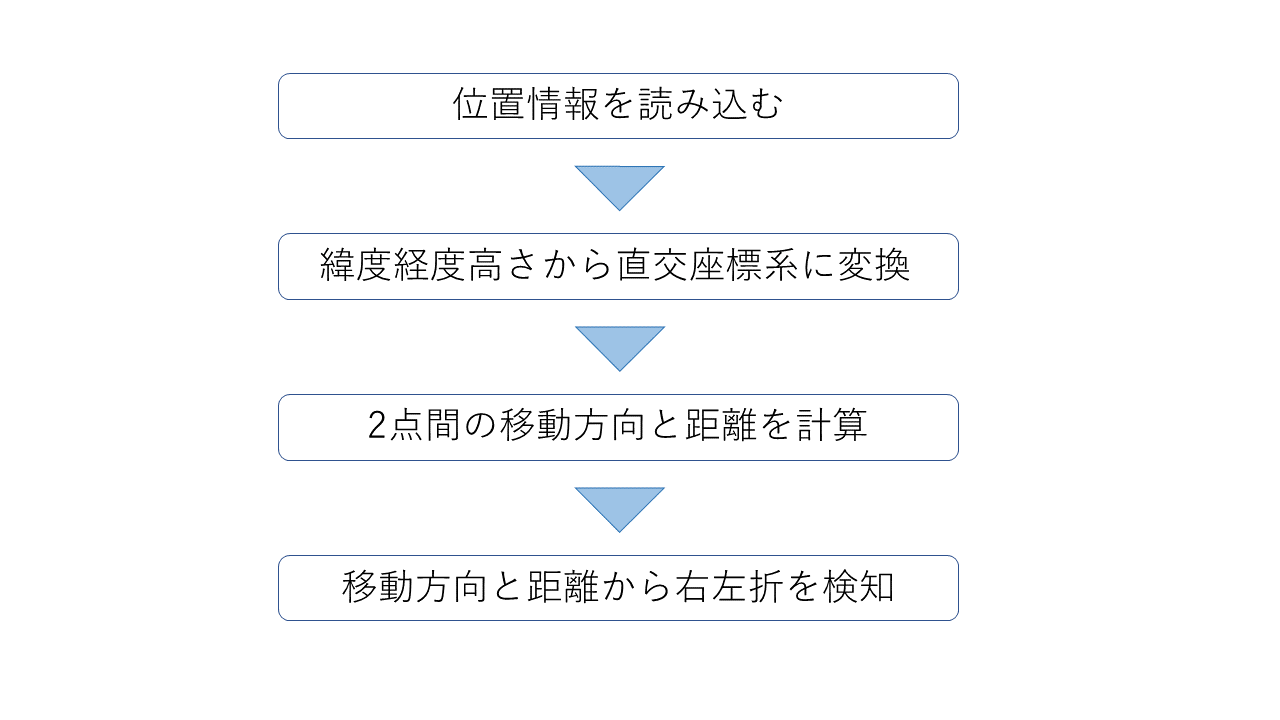


図4-2 右左折検知の流れ

|  |  |  |
| --- | --- | --- |
| メンバ名 | 型 | 値など |
| ID | int | データに割り振ったID |
| day | String | 日付 |
| time | String | 時間 |
| speed | double | 速度 |
| angle | double | 移動方向 |
| distance | double | 移動距離 |
| pos | GPSPosition | 位置情報 |

表4-1 SinglePointDataクラス

　緯度経度高さの値から直交座標への変換に使用した式を以下に示す。

ただし、

とする。

　車両が右左折する場合には、ある点から一定の距離内で、進行方向が大きく変化する。よって、ある2点間の進行方向と距離から右左折を判断することができる。

そのため、ある地点から、次の地点までの進行方向と距離を計算し、ある地点から一定の距離移動するまでにどれだけ進行方向が変化したかを確認する。

　右左折を開始した点を見つけるアルゴリズムと右左折終了と判断するアルゴリズムをそれぞれ図4-3、4-4に示す。初めに、位置情報に対して右左折開始を判断するアルゴリズムを適用する。開始点が見つかったら、右左折終了を判断するアルゴリズムを適用し、終了点を見つける。これをすべての位置情報に対して適用する。

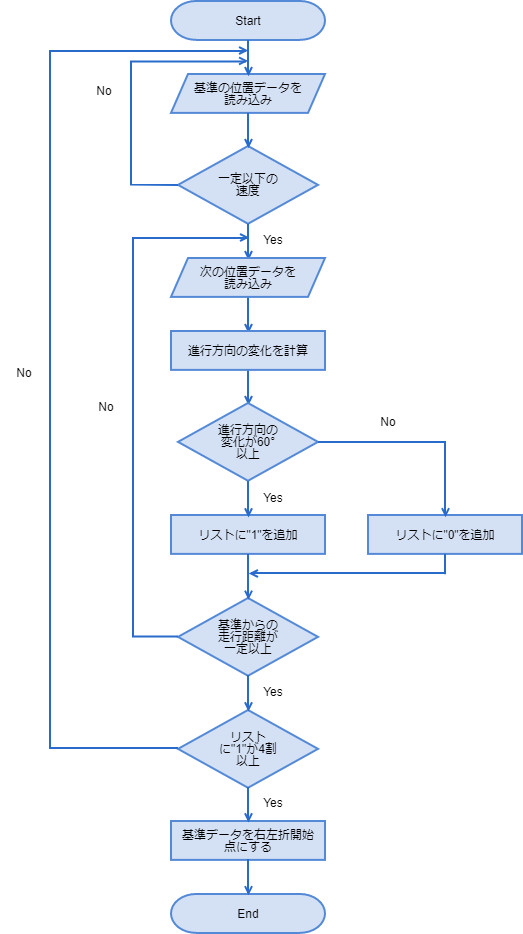


　　　　　　　　　　　　　　　　図4-3 右左折開始

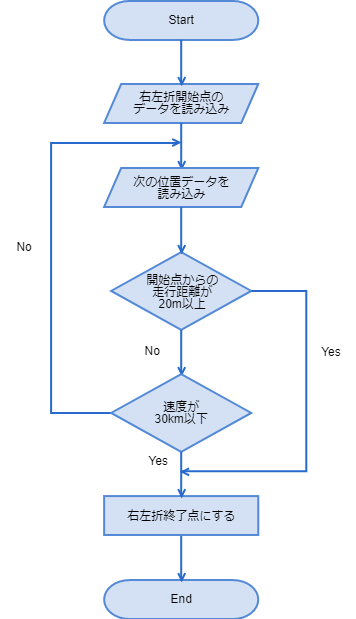


　　　　　　　　　　　　　　　　図4-4 右左折終了

## 側方確認動作検知

　ドライバーが側方確認を行う際には、確認する方に視線を向けて歩行者や車両の有無を確認する。これを検知するために、運転中の視線をWebカメラとWebGazer.jsによって取得する。視線情報はページ上の座標に割り当てられた状態で取得されるため、これを元に側方確認動作の検知を行う。

視線情報は、顔が検出できない場合取得できない。そのため側方確認の際、顔の向きが正面ではなくなり、視線情報を取得できなくなる。側方確認の前後には、確認する方向に視線が向いていると考え、視線情報の補完を行う。視線情報を補完するアルゴリズムを図4-5に示す。

　補完した視線情報から、側方確認動作の検知を行うアルゴリズムを図4-6に示す。視線情報に対してこのアルゴリズムを適用していき、データが無くなったら終了する。

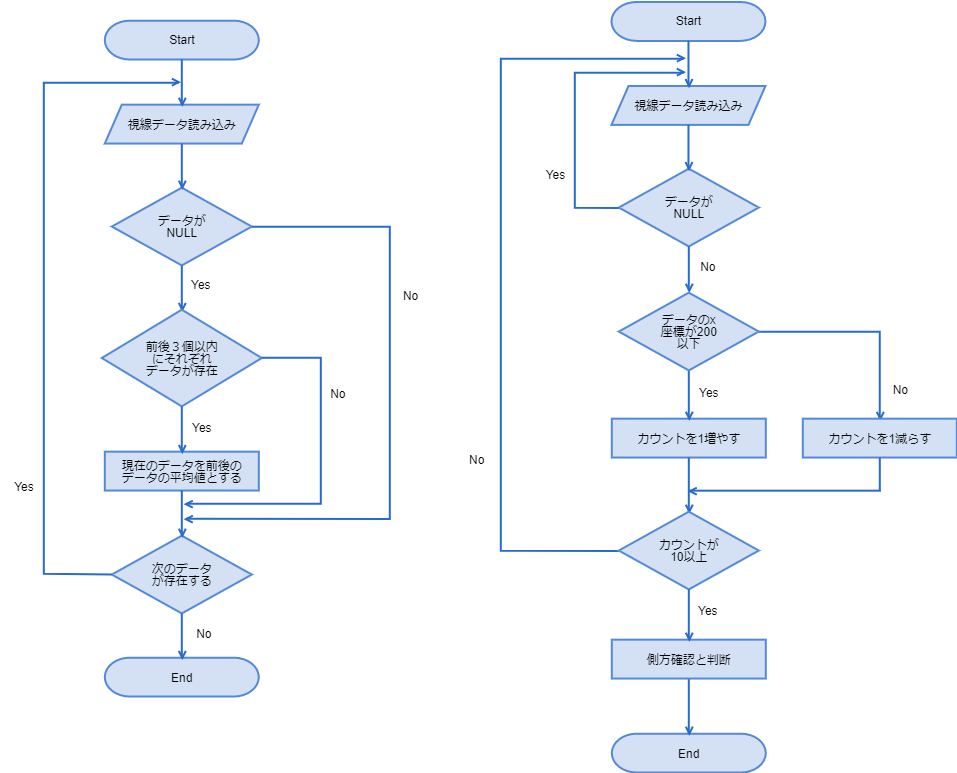


　　　　　　　　　　　図4-5　視線情報の補完

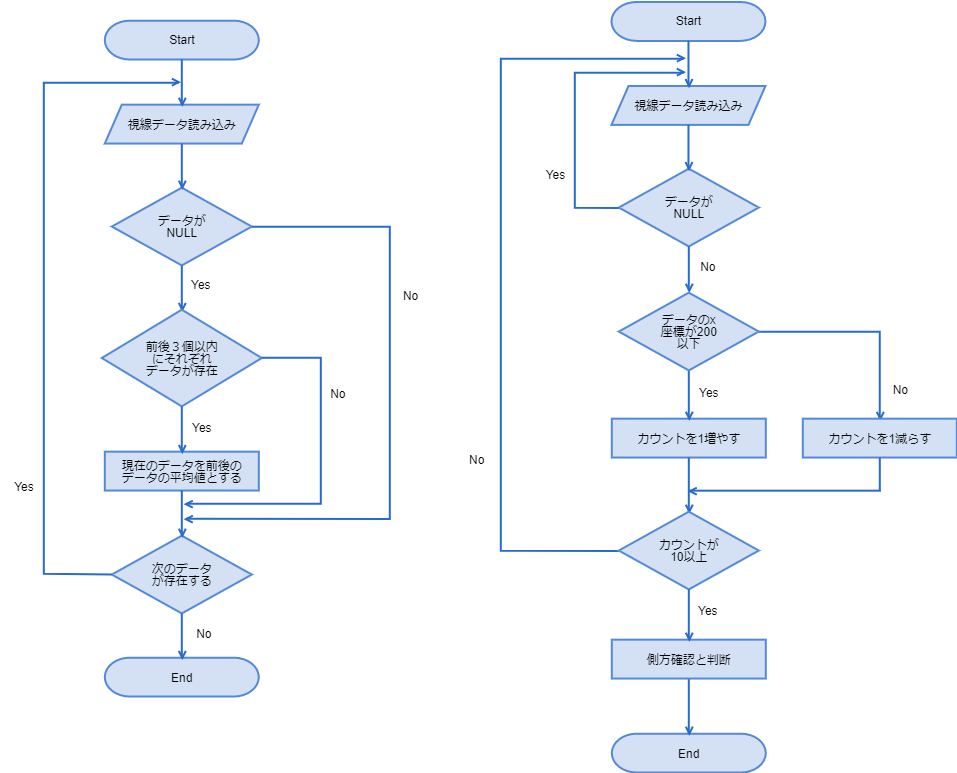


　　　　　　　　　　　図4-6　側方確認動作検知

## GoogleMapへの結果出力

　得られた結果をわかりやすく表示するため、Google Maps JavaScript APIIを使用して結果の表示を行う。得られた結果からHTML文を作成し、ファイルに書き出すことで結果を出力する。

　位置情報はポリラインによって表示する。左折と判断した部分を黄色で、右折と判断した部分を赤色で、それ以外を水色として色分けし、ひと目でどこが右左折と判断した部分なのかをわかるよう表示する。側方確認動作の検知結果は時間によって記録するため、そのままではGoogleMapに出力ができない。そのため、同時刻の位置情報を参照して、マーカーを立てることで表示する。

# 実装

## 視線情報測定ソフトウェア

### 開発・実行環境

　視線情報を測定するソフトウェアの開発環境を表5-1に示す。今回作成したソフトウェアは、ローカルで動作させることを目的としたが、Webgazer.jsを実行するためにローカルサーバを構築する必要があったため、XAMPPを使用してローカルサーバを構築した。

表5-1　視線情報測定ソフトウェアの開発環境

|  |  |
| --- | --- |
| 開発OS | Windows 10 |
| 開発言語 | HTML5, javascript |
| エディタ | Notepad++ |
| Webサーバソフトウェア | Apache |

### ソフトウェア画面

　図5-1に実装した測定画面を示す。

### ログの出力

　測定した結果は、時刻と紐付けして配列に書き込むことで記録する。測定終了した際に、記録しておい測定結果と時刻を文字列に書き出し、txtファイルとして出力する。このtxtファイルをダウンロードさせることでログを出力した。

## 位置・視線情報処理アプリケーション

### 開発・実装環境

　位置情報と視線情報を処理するアプリケーションの開発環境を表5-2に示す。

表5-2　位置・視線情報処理アプリケーションの開発環境

|  |  |
| --- | --- |
| 開発OS | Windows 10 |
| 開発言語 | JAVA |
| IDE | NetBeans IDE 8.2 |

### データ読み込み画面

　実装した、データを選択する画面を図5-xに示す。取得した位置情報と視線情報のファイルパスを入力することでデータを選択する。また直接ファイルパスを入力するだけではなく、

JFileChooserクラスを使用してファイルを選択できるようにした。

### GoogleMapへの結果出力

　読み込んだデータから得られた結果は、PrintWriterクラスを使用してHTML文をファイルに出力し、ブラウザで読み込むことで表示した。出力した結果を図5-xに示す。出力するHTML文は、最初と最後の部分が共通であり変化しないため、それぞれheader.txtとfooder.txtとして保存しておき、これらと結果部分を結合することで出力した。

# 評価

## 実験方法

## 実験結果

### 側方確認動作の検知

### 右左折の検知

## 考察

# まとめ

## 研究の成果

本

## 今後の課題

本

# 参考文献

[1] 自動運転レベルの定義を巡る動きと今後の対応（案）: 内閣官房IT総合戦略室

https://www.kantei.go.jp/jp/singi/it2/senmon\_bunka/detakatsuyokiban/dorokotsu\_dai1/siryou3.pdf

[2] 全自動車事故の発生状況 : 三井住友海上

http://www.ms-ins.com/special/rm\_car/accident-data/

[3] 自動運転技術に関わる国際ガイドラインの概要と課題 : 交通安全環境研究所

https://www.ntsel.go.jp/forum/2014files/1106\_1445.pdf

[4]「安全運転サポート車」の普及啓発に関する関係省庁副大臣等会議 : 経済産業省

http://www.meti.go.jp/report/whitepaper/data/pdf/20170404002\_2.pdf

[5] ローコスト視線入力装置による意思伝達環境の構築およびマニュアル作成 : 伊藤史人

http://alsjapan.org/wp-content/uploads/2017/06/%E4%BC%8A%E8%97%A4%E5%8F%B2%E4%BA%BA%E6%B0%8F\_%E5%A0%B1%E5%91%8A%E6%9B%B8%EF%BC%88%E5%85%A8%E6%96%87%EF%BC%89.pdf

[6]視線検出技術-基本原理 : 富士通研究所

http://www.fujitsu.com/jp/group/labs/resources/tech/techguide/list/eye-movements/p03.html