

博士学位論文

埼玉工大の自動運転機能後付けバスと
その安全性向上に関する研究

Retrofitted Autonomous Bus and its Safety Improvement

埼玉工業大学大学院 工学研究科
博士後期課程 システム工学専攻

王 治

目次

第1章	序論	6
1.1	自動運転.....	6
1.2	各章の内容.....	8
第一部	本プロジェクトの概要	9
第2章	自動運転車の発展	10
2.1	DARPA グランドチャレンジ.....	10
2.2	自動運転レベルの定義.....	13
2.3	自動運転実証実験.....	14
2.4	自動運転事故.....	16
2.4.1	Tesla.....	16
2.4.2	Uber.....	17
2.4.3	名古屋大学.....	17
2.4.4	群馬大学.....	18
2.5	本文で取り組む安全対策.....	19
第3章	自動運転車に支える技術	21
3.1	自動運転車のハードウェア.....	21
3.1.1	LIDAR.....	21
3.1.2	RADAR.....	22
3.1.3	CAMERA.....	24
3.1.4	GNSS.....	26
3.1.5	IMU.....	27
3.2	自動運転車のソフトウェア.....	28
3.2.1	OS, ROS and Autoware.....	28
3.2.2	HD Map.....	29
3.2.3	Localization.....	33
3.2.4	Perception.....	36

3.2.5	Prediction	37
3.2.6	Planning	38
3.2.7	Control	39
第4章	埼玉工業大学の自動運転車.....	42
4.1	ROBOCAR.....	42
4.2	自動運転マイクロバス.....	43
4.2.1	ジョイスティック式自動車	43
4.2.2	ジョイスティック式自動車運転装置	43
4.2.3	埼玉工業大学の自動運転マイクロバス	46
4.2.4	自動運転実証試験による発見した問題	49
4.2.5	各問題の対策方法	50
第二部	本論文で新しい提案.....	52
第5章	車載カメラの視認性を高めるヘイズ除去.....	53
5.1	従来ヘイズ除去手法の問題所在.....	53
5.2	ヘイズ除去に関する研究背景.....	54
5.2.1	ダークチャンネルに基づくヘイズ除去原理	54
5.2.2	ダークチャンネル手法の拡張	56
5.3	提案手法.....	57
5.3.1	グレースケール画像	57
5.3.2	パラメータ ω の推定.....	57
5.3.3	全局大気光の改良	57
5.4	実験の流れ.....	58
5.5	実験結果.....	59
5.5.1	ヘイズ除去を行った画像の目視による評価	59
5.5.2	検出精度と IOU の比較	59
5.5.3	PSNR と SSIM の比較.....	61
5.5.4	処理時間の調査	61

5.6	本研究に残した問題.....	62
第6章	自動運転実験車両用のドライブレコーダー.....	63
6.1	自動運転実験車両に必要性.....	63
6.1.1	従来の自動運転事故と事故対応.....	63
6.1.2	従来の実証実験中データ記録の問題所在.....	64
6.2	ドライブレコーダーの研究背景.....	64
6.2.1	従来のドライブレコーダーの研究.....	64
6.2.2	Autonomous-Car Drive Recorder.....	65
6.3	ACDRの構成.....	66
6.3.1	ACDRの全体システム図.....	66
6.3.2	ACDRの録画面.....	66
6.3.3	オペレーターの操作画面.....	67
6.4	ACDRの有用性.....	73
6.4.1	ACDRの応用例.....	73
6.4.2	ACDRの利点.....	73
6.5	本研究に残した問題.....	74
第7章	その他の安全確保機能.....	75
7.1	蛇行を抑えるモデル予測操舵制御.....	75
7.1.1	モデル予測操舵モデル.....	75
7.1.2	MPCによる経路追従性の調査.....	76
7.1.3	MPCによるハンドルをぶれる問題の改善.....	77
7.2	ローライザー切り替え機構.....	79
7.2.1	NDTとGNSSによる自己位置推定の比較実験.....	79
7.2.2	実証実験中ローライザーの応用.....	80
7.3	定時点灯信号についてサイクル情報の利用.....	81
7.4	フェールセーフ機能.....	83
第8章	結論.....	84

第9章 参考文献.....	85
謝辭	93

第1章 序論

1.1 自動運転

100 万年前、人間は手を解放し、両足で歩くことを学んだ。4000 年前、人間は馬車を作って、人を乗るまたは荷物を運搬する。150 年前、人間はガソリン自動車を発明した、人間はこれまで以上に速く、遠くに行くことが実現した。現在、人間は運転をより簡単かつ安全にするため、自動運転車の研究が活発に行っている。

近年、自動停止機能など安全運転支援機能ついた観光バスの新車の販売は徐々に始まりつつあるが、バス車両自身のライフサイクルが乗用車に比べて長い（新車の代金が高く、古い車両を使い続ける傾向がある）ので、乗用車の安全運転支援機能のように普及は進んではいない。バスドライバーの運転士不足は深刻であり、運転しやすい車両で労働環境を良くして新規の運転手を増やし、離職につながる高齢バスドライバーの事故を減らす自動運転／安全運転支援の追加機能の開発が重要な課題となっている。この課題に対応するため埼玉工業大学では、既存のバスに自動運転／安全運転支援機能を後付する仕組みを東京農工大学の和田准教授と株式会社ミクニライフ & オートが開発した「ジョイ・カー」をベースに開発している。このシステムは任意の古いバス車両（オートマ）に後付けできるので、この後付け自動運転／安瀬運転支援機能が既存のバスの操作性と安全性を上げることができれば、ドライバー不足の問題の解決し運転免許返納後の高齢者の移動の便や地域公共交通の維持に役立つのではないかと考えている。

自動運転技術の進歩により、自動運転実証実験が世界の各地で展開されている。2019 年、64 社がカリフォルニア州[1][2]の自動運転道路使用許可を取得し、463 万キロメートルを走行した。その中に、走行距離が一番多くの Waymo 社は 153 台の自動運転車でカリフォルニア州の 230 万キロメートルを走行し、21273 キロメートルごとの自動走行に一回手動介入を行っていることと自転車やバイクの誤認識が減らしたことなどが報告された。日本国内では、日本政府の主導による東京臨海部の一般道と高速道における自動運転の実証実験や、山間地域における道の駅などを拠点とした自動運転、或は空港制限区域内における自動運転実証実験が日本各地で展開されている。その一方で、自動運転実証実験数の増加と共に、事故の発生も必然的に増えるであろう。

埼玉工業大学ではこの開発のために延べ2,415名の体験試乗者を乗せて計652kmの公道（本番走行のみテスト自動走行想定3000km）を自動走行した。申請者はこの実証実験全てにおいて助手席でシステムオペレーターとして自動運転システムの開発に参画した。様々な土地における自動運転実証実験で遭遇した経験をもとに自動運転バスの安全性を向上する検討・提案を行い、3件の国内学会発、4件の国際会議、2編の学術論文を公表した。本博士論文は、これらの発表をベースに、以下の2部構成からなっている。

最初に、車載カメラの視認性を高めるヘイズ除去手法を提案した。悪天候の下に、特にヘイズ天気に対して、カメラから撮った写真にはヘイズが存在していて、写真が全体的に白っぽくなって、障害物の検出率が低くなる。ヘイズ除去は画像処理の分野にノイズ除去と理解できる。提案したヘイズ除去手法はDark Channel Prior[3]手法をベースとして進化させた。提案手法はYOLO (You Only Look Once) [4]という障害物検出アルゴリズムの前処理として利用する。ヘイズ除去後とヘイズ除去前に比べて、障害物の検出率が高くなることを確認した。更に、目視の判断やPSNRとSSIMに基づき画質評価を行い、以前の手法より良い結果を取った。

最後に、埼玉工業大学自動運転バスが参加していた自動運転実証実験に基づいて、走行中に様々な危険状況に対して様々な対策方法（信号のサイクル情報の利用手法、ローカライザーの切り替え機構の開発、フェールセーフ機能を自動運転バスに追加したこと）を含めて、走行中に車両の内部データなど重要な情報を監視しながら、自動運転実験車両用のドライブレコーダーを提案した。自動運転実証実験ではヒヤリハット（や事故）を完全に避けることは難しいが、同じ原因のヒヤリハット（や事故）に合うことが2度とないようにするためにはヒヤリハット（や事故）の原因を正確に分析する必要がある。自動運転のPCはLinux/ROSで動いていてLinux/ROSには標準のログにRosbagがあるが、Linuxがリアルタイムではないのでこのログを再生・分析するだけではヒヤリハット（や事故）が再現できるとは限らず、ヒヤリハットや事故の原因究明が十分にできるとは限らない。さらに、自動運転の記録ではテイクオーバーリクエストに対する運転手の動作やオペレーター画面の記録も重要である。そこで、車両の全方位カメラ、鳥瞰図とオペレーターの操作画面及び運転席の映像を記録する自動運転実験車両用のドライブレコーダーを開発した。

1.2 各章の内容

本文の第 2 章では自動運転車の歴史と近年各企業の自動運転車開発競争を紹介した。その後、自動運転実証実験が必要な理由と自動運転実証実験の事故の例を説明した上、本文で取り組む安全対策について紹介した。

第 3 章では世界中に自動運転車が搭載しているセンサー及び自動運転車が利用されている技術を紹介した。そこで、各センサーと技術の問題点を述べた。

第 4 章では埼玉工業大学の自動運転マイクロバスの仕組みと埼玉工大自動運転システムを構成しているジョイスティック式自動車運転装置を紹介した。そこで、埼玉工大自動運転バスが実証実験走行中に発見した問題点と解決する方法を記述した。

第 5 章では車載カメラの視認性を高めるヘイズ除去手法を提案した。悪天候の下に、特にヘイズ天気に対して、カメラから撮った写真にはヘイズが存在していて、写真が全体的に白っぽくなって、障害物の検出率が低くなる。ヘイズ除去は画像処理の分野にノイズ除去と理解できる。提案したヘイズ除去手法は Dark Channel Prior 手法をベースとして進化させた。提案手法は YOLO (You Only Look Once) という障害物検出アルゴリズムの前処理として利用する。ヘイズ除去後とヘイズ除去前に比べて、障害物の検出率が高くなることを確認した。更に、目視の判断や PSNR と SSIM に基づき画質評価を行い、以前の手法より良い結果を取った。

第 6 章では埼玉工業大学自動運転バスが参加していた自動運転実証実験に基づいて、走行中に様々な危険状況に対して様々な対策方法を含めて、走行中に車両の内部データなど重要な情報を監視しながら、自動運転実験車両用のドライブレコーダーを提案した。自動運転実験車両用のドライブレコーダーは車両の全方位カメラ、鳥瞰図とオペレーターの操作画面及び運転席の映像を記録する。該当のドライブレコーダーは従来の自動運転実証実験が存在していた問題を解決した。事故が発生しても、事故を再現できるようになった、事故調査をスムーズに進めることが利点である。

第 7 章は様々な安全走行確保機能を紹介した。中速走行際に蛇行を抑えるモデル予測操舵制御、定時点灯信号のサイクル情報の利用手法、ローカライザー切り替え機構、フェールセーフ機能を述べた。

第 8 章には本文のまとめと今後の課題を述べた。

第一部

本プロジェクトの概要

第2章 自動運転車の発展

2.1 DARPA グランドチャレンジ

自動運転車はロボットカー、無人運転車などとも呼ばれていて、人間の操作を介入しなくても自動的にハンドルを回したり、アクセルブレーキを踏んだりことができる。自動運転技術は多年にわたって研究が行われてい s る。特に近年、CPU、GPGPU、SSD やメモリなどコンピューターハードウェアの進化と低価格のため、自動運転技術に沢山の可能性が与えられた。

アメリカ国防総省の先進研究プロジェクト推進機関(DARPA)は自動運転技術発展のため、グランドチャレンジを開催し、世界初の長距離無人自動車の競技となった [5][6]。2004 年開催された第一回目のグランドチャレンジの走行総距離は 240km である。一番目はカーネギーメロン大学のサンドストームであり(図 2-1)、11.78 キロメートルしか走行しなかった。第一回目のグランドチャレンジには完走した車両はなかった。



図 2-1. カーネギーメロン大学のサンドストーム (引用元[6])

2005 年開催された第二回目のグランドチャレンジの走行総距離は 212km であり、完走した車両は 5 台がある。スタンフォード大学の Stanford Racing Team の自律走行車スタンレー(図 2-2)が第一位となって、スタンレーはフォルクスワーゲン TouaregR5(ディーゼルエンジン)をベースとして、改造した。当時の車両ではレーザー測距装置(laser range finder)四つ、レーダー(Radar)、6DOF(degrees of freedom)

の慣性計測装置、単眼カメラ、ステレオカメラと GPS 情報を 10Hz で取り扱って、Pentium M を搭載しているパソコン合計 7 台でデータを処理する [7] [8]。



図 2-2. スタンフォード大学のスタンレー (引用元[7])

2007 年では第三回目グランドチャレンジが開催された。コースは市街地を想定した全長 96km で、6 時間以内に完走することを求められた、6 チームが完走した。カーネギーメロン大学のタータンレーシングチームの自律走行車 BOSS (図 2-3) が第一位となって、車両では 64 ラインのライダー、カメラとレーダーなど装置を搭載した [9] [10]。スタンレーを皮切りとして、ライダーが自動運転車の不可欠な部分になった。



図 2-3. カーネギーメロン大学の BOSS (引用元[10])

DARPA グランドチャレンジ開催以来、色々な自動運転実証実験や競技 (the European Land-Robot Trial、the Intelligent Vehicle Future Challenge、the Autonomous Vehicle Competition、the Hyundai Autonomous Challenge、the VisLab Intercontinental Autonomous Challenge、the Grand Cooperative Driving Challenge) は世界各地で行った。したがって、大手企業や大学による自動運転の研究を促進していた[11]。大学では主にカーネギーメロン大学、スタンフォード大学やマサチューセッツ工科大学などがある。Google[12]、Cruise[13]、Uber[14]、Baidu[15]、Tesla[16]、Nvidia[17]、Mobileye[18]、Pony.ai[19]、Toyota[20]など企業では主に乗用車を開発している。一方、TuSimple(図森未来)[21]ではトラックの自動運転の開発を注目している。図2-4では各企業の自動運転車を表示している。

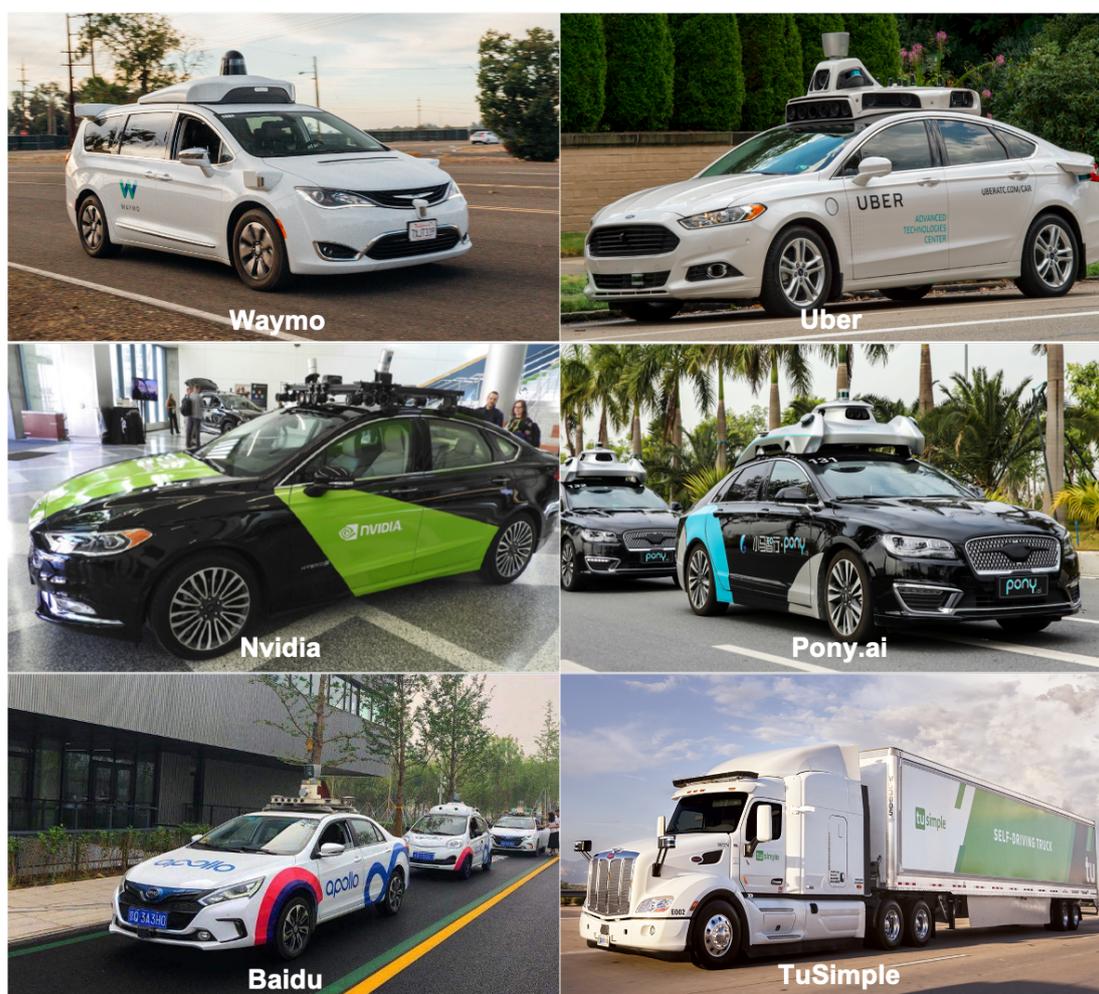


図2-4. 各企業の自動運転車 (引用元[12][14][17][19][15][21])

2.2 自動運転レベルの定義

2014年、SAE(Society of Automotive Engineers, 米国自動車技術協会)では6段階の自動運転レベルを定義した。しかし、分類や用語の不明確のため、修正などが行われ、2016年からJ3016発表された、以下のように定義されている[22]。

レベル0：運転自動化無し

レベル1：運転支援

レベル2：部分的運転自動化

レベル3：条件付き運転自動化

レベル4：高度運転自動化

レベル5：完全運転自動化

近年、市販されている自動車が自動運転レベル2相当の運転支援システムはいくつか装備している。

トヨタ自動車：

- 高度運転支援システム：先行車両と無線通信しながら追従走行と全車速域で道路の白線などをセンサーで検出する。

- セーフティセンス：進路上の先行車をレーザーレーダーと単眼カメラで検出し、衝突予測を行う。白線認識、オートマチックハイビームなどもある。

本田技研工業：

- ホンダセンシング：衝突軽減ブレーキ、誤発進抑制機能、車線維持支援システムなど。

日産自動車：

- プロパイロット：自動車専用道路のみで、同一車線内ハンズオフ(手放し運転)が可能なナビ連動ルート走行を実現したシステム。GPSが受信できないトンネル内、対面通行、合流や急カーブとの状況下は利用できない。

メルセデスベンツ：

- ディストロニックプラス：レーダーセンサーにより先行車を認識し、速度に応じて車間距離を適切にキープシステム。

以上の運転支援システムを概にすると、以下になる。

- 車線保持機能：逸脱しそうな場合に警告表示やステアリング操作を介入する。

- クルーズコントロール機能：レーダーや超音波センサーで先行車両までの車間距離を保持して、自動的に追従する。

- 衝突予測機能：ミリ波レーダー及びカメラを用いて、障害物に衝突の恐れがある場合、警告を出して、ステアリングまたはブレーキ操作を行う。

更に、これらより高度な自動運転機能を装備している車両もあり、アウディのA8やテスラなど車両では一定の条件を満たす場合、自動運転が可能と宣言された。

2.3 自動運転実証実験

市販車両と違って、自動運転実験車両(図2-5)は主にライダー(LIDAR、Light Detection and Ranging)センサーを搭載して、Autoware[23]やApollo[24]など自動運転オープンソースを利用して、自動運転の研究を行っている。日本において、多くの会社や大学はAutowareを利用して、AutowareはLinuxとROS(Robot Operating System)[25]をベースとして、開発されている。ライダー、レーダー、カメラ、GNSS(Global Navigation Satellite System)などの環境センサーを利用して、自車位置や周囲物体を認識しながら、自動走行を行う。



図2-5. 日本の自動運転実験車両 (右上 Tier IV 車両の引用元[26])

自動運転を実現するためには、次のような自動運転技術が欠かせない。

- 高精度地図(High Definition map)
- 自己位置推定(Localization)
- パーセプション (Perception)
- 予測(Prediction)
- プランニング(Planning)
- コントロール(Control)

自動運転技術を開発するため、上記された技術を検証する必要がある、自動運転実証実験は日本各地で行っている。自動運転実証実験に基づいて、実験中に様々な問題点を発見し、解決する。そこで、実証実験に参加した車両がだんだん進化になる。次の図2-6のように自動運転開発サイクルをなっている。

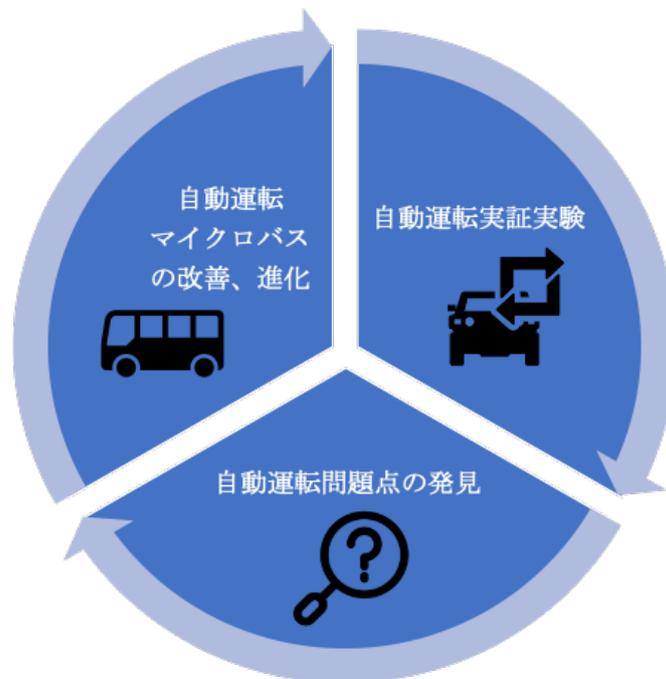


図2-6. 自動運転実証実験による開発サイクル

日本で行われた実証実験は色々な種類があって(図2-7)、車両の操舵と制動の正確さを確認、改善するため、一番多くのは閉鎖空間の実証実験、空港や公園などよく行われている。

次は他の交通参加者と障害物への対応を検証するため、交通量が少ない、山間地域における道の駅などを拠点とした自動運転実証実験はよく行われている。

続いて、より多くなる他の交通参加者と障害物への対応を検証するためには、交

通量が多い公道の実証実験はお台場、新宿など行ったことがある。

最後には、遠隔操作室を備え、色々なサービスを検証することを目的としたレベル4の公道実験である。

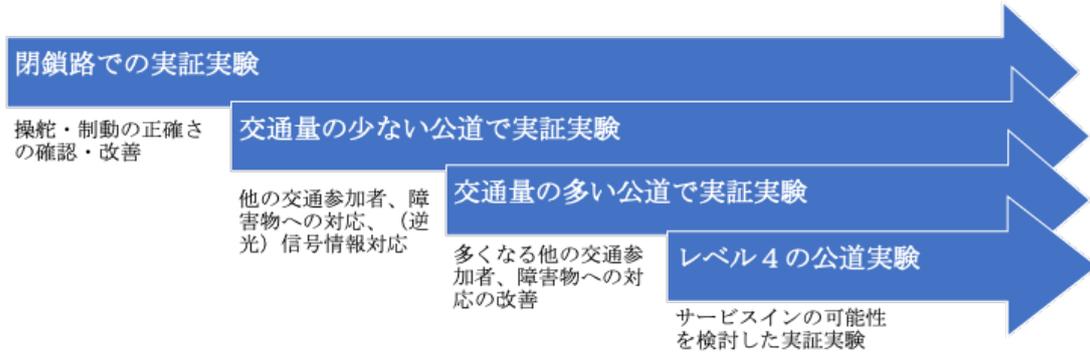


図2-7. 日本で行われた自動運転実証実験の種類

2.4 自動運転事故

2.4.1 Tesla

2016年1月20日、中国の京港高速道路で特斯拉 (Tesla Model S) が前方の道路清掃車に追突事故が発生した[27]。事故が発生する前に、車両が Autopilot の運転モードになって、事故現場ではブレーキを動いた実績がなかった (図2-8)。



図2-8. 中国の京港高速で発生した特斯拉事故(引用元[27])

2016年8月2日、中国北京の一般道路で特斯拉事故が発生した。前方の車両が路駐を見ながら、右に避けた。後方にある Autopilot 運転中の特斯拉が前方の路駐を認識できなかった、そのままにクラッシュした。

二つの事故原因は主に車載カメラ Mobileye は前方車両が認識できなかった。ま

た道路清掃車のシャーシが一般車両より高く、RADAR の検出範囲を超えた可能性もある。同年、中国のテスラオフィシャルサイトは自動運転から自動補助運転に変更した。

2.4.2 Uber

2018年3月20日の夜10頃、アリゾナ州で世界初のレベル4の自動運転事故が発生した[28]。自動運転中のUberは自転車を押しながら車道を渡っていた49歳の歩行者 Elaine Herzberg を時速約64キロではねて死亡させた。事故の原因は障害物が認識できなかった(図2-9)。



図2-9. アリゾナ州で発生したレベル4の自動運転事故(引用元[28])

日本の国土交通省の指示により、LiDARにたよる自動運転には「急制動」が多いものが多いである。それは、LiDARからスキャンした点群の種類が分類できない時、全て固い停止障害物として扱われることが原因である。しかし、この機能をオンにすれば、車両が走行中に急ブレーキが多くて、乗り心地がすごく悪いので、そのため、この機能をオフにした。その結果、AIの判定が、本来人が歩くところでないところを人が歩いて、自転車を引いていたので、人と自転車の該当する点群の塊への対応がオフにされていて、停止するような指令が出なかった。

2.4.3 名古屋大学

2019年8月26日の午後2時20分頃、名古屋大学所有の低速自動運転車は時速

約 14 キロで市道を自動運転走行中に後ろから追い越してきた車両と接触事故が発生した。事故の原因調査を行った、原因は自動運転車両の位置と方位検知機能が進行すべき方位を誤検知した。そして、誤った急ハンドルが発生した。図 2-9 は事故が発生する際に、LIDAR による自己位置推定された様子を表示した。衝突の 1.4 秒前に LIDAR 情報と点群地図をマッチングしている。衝突の 1.3 秒前、LIDAR による推定された自車の前進方向が左に約 56 度をずれた。その結果、コンピューターは車が左に向かうことを認識していて、それを正しい方向を修正すると、ハンドルを右に 56 度を回ることになる。

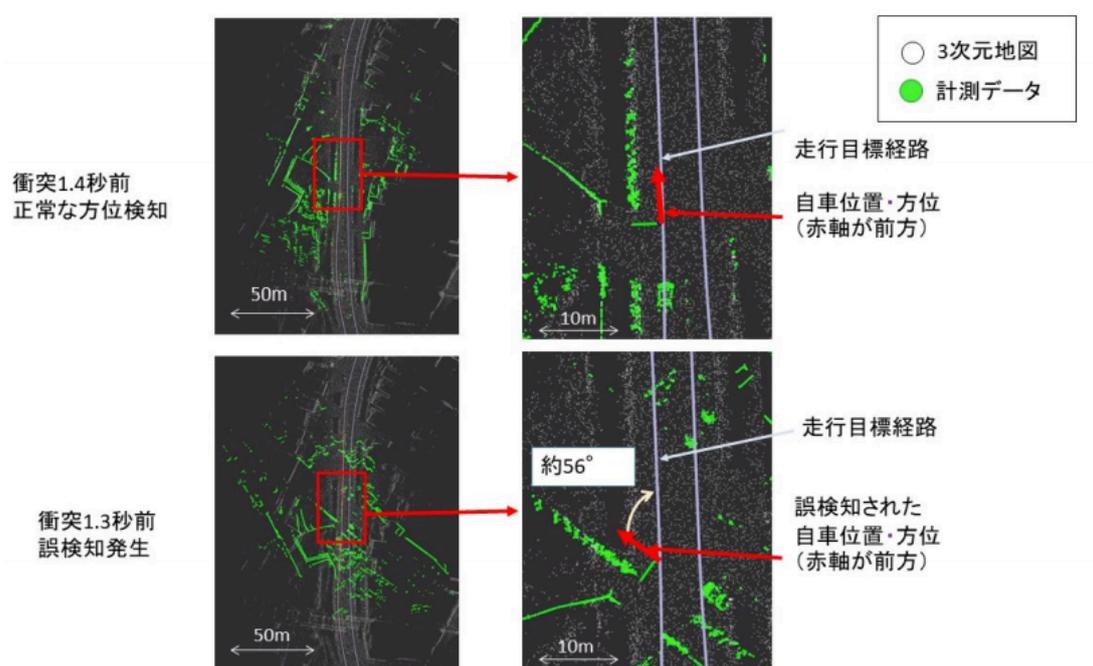


図 2-9. 事故が発生する際に推定された自転車位置 (引用元[29])

2.4.4 群馬大学

2017年9月1日の午前1時10分頃、群馬県桐生市内の県道で実証実験中の自動運転車は自己位置推定が失敗し、時速20キロ程度でガードレールに衝突事故が発生した(図2-10) [30]。事故が発生する際に、車内には運転席と助手席、後部座席に職員3人が乗り、桐生市天神町の緩やかな左カーブ手前で一時停止した後、自動運転モードに切り替え発車したところ、ハンドルが想定外に左に切られ衝突した。事故の直前に、人間の運転に近づけるためハンドル動作をスムーズにするプログラム変更を行っており、群馬大学は「プログラムミスの可能性がある」ことを事故原因に推定し

た。



図 2-1 0. 群馬大学自動運転事故現場（引用元[30]）

2019年9月25日午前2時20分頃、群馬大学は大分県大分市で自動運転実証実験中に、自己位置推定が失敗し、歩道の縁石と接触事故が発生した。事故が発生する際に、自動運転バスが交差点で左折中に、歩道の縁石に乗り上げた。群馬大学は自動走行システムの問題ではなく、運転手の人為的なミスが原因を公表した。

2.5 本文で取り組む安全対策

現段階の自動運転車は様々なセンサーや技術の支え、一定の条件下に、人間は車のハンドル、アクセル、ブレーキを触らなくても車が自動的に走れる。しかし、自動運転車による交通事故はゼロではなくて、自動運転車を作るという人間の本来の意図に反して、自動運転車に乗るには、人間の手動運転よりもストレスになる場合がある。本文では従来の自動運転事故を考えて、埼玉工業大学の自動運転バスを用いて、自動運転実証実験に参加し、実験中に発見した問題点を解決する。

事故やヒヤリハットの原因は主に障害物の誤認識と位置推定失敗、信号色の認識ができないことや中速走行中の車両に蛇行が発生すること。そのため、ジョイカー[31]をベースとして、構成された埼玉工大の自動運転バスが実証実験中に安全走行のため、様々な手法を提案した（図 2-1 1）。

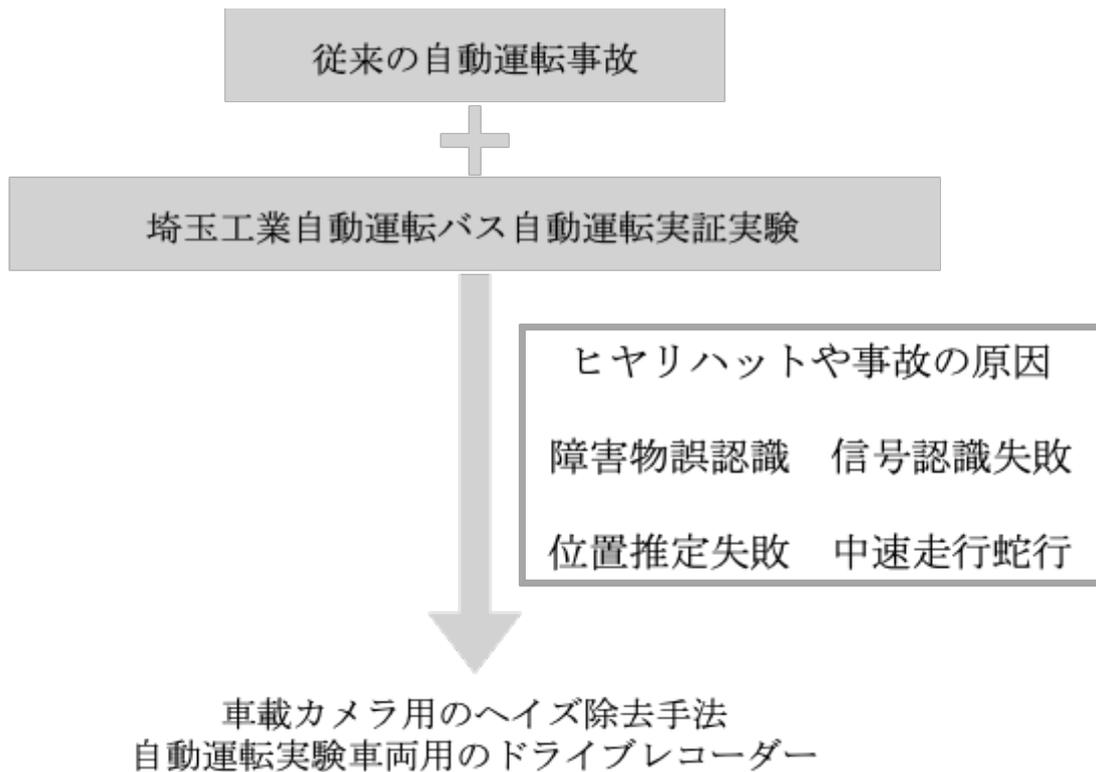


図 2-1 1. 本文で取り組む安全対策

悪天候の下における車載カメラ画像の障害物検知能力の改善するため、高速ヘイズ除去手法を提案した[32][33][34]。ヘイズ除去後の画像の視認性、物体検出能力などをこれまで以上に向上させることが確認できた。ヒヤリハットや事故の原因を究明するために、車両の内部データや各環境センサーの誤差情報をリアルタイムで流れる必要があるため、ユーザーインターフェイスを提案した[35]。提案したユーザーインターフェイスを自動運転実験車両用のドライブレコーダー[36]の一部の構成として車両の内部データを記録する。従来の市販されているドライブレコーダーの基本機能を保つ上、オペレーターやドライバーの誤動作や車両の内部データを記録することができる。事故が発生する際に、事故分析の信頼性や説得力が高い。

第3章 自動運転車に支える技術

自動運転とは高度複雑なシステム、それを実現するため、ソフトウェアとハードウェア両者と共に依存している。近年、自動運転車が話題になる理由は、技術の進歩によるものだけではなく、それもハードウェアの発展によるものである。

3.1 自動運転車のハードウェア

3.1.1 LIDAR

LIDAR(Light Detection and Ranging)、日本語はライダーと呼ばれて、自動運転車の目であることを自動運転業界に認識され、車両周辺の障害物をスキャンし、数値表層モデル(Digital Surface Model, DSM)に基づいた三次元データを流れて、高精度な3D点群を取得する。現在市販されている多くのLIDARは905nm波長前後のレーザー光を生成し、対象物体に照射し、反射光を捉えて、障害物までの距離を算出し、対象物体表面の反射率(反射強度)も捉えることもできる。以下(図3-1)は各企業販売しているLIDAR製品、回転式のLIDAR(VLS-128, RS-RubyとPandar40Mがあつて、FOVは360度)だけではなく、固定式のLIDARもある(VelarrayとRS-Lidar-M1のFOVは120度、VelabitのFOVは60度)[37][38][39]。

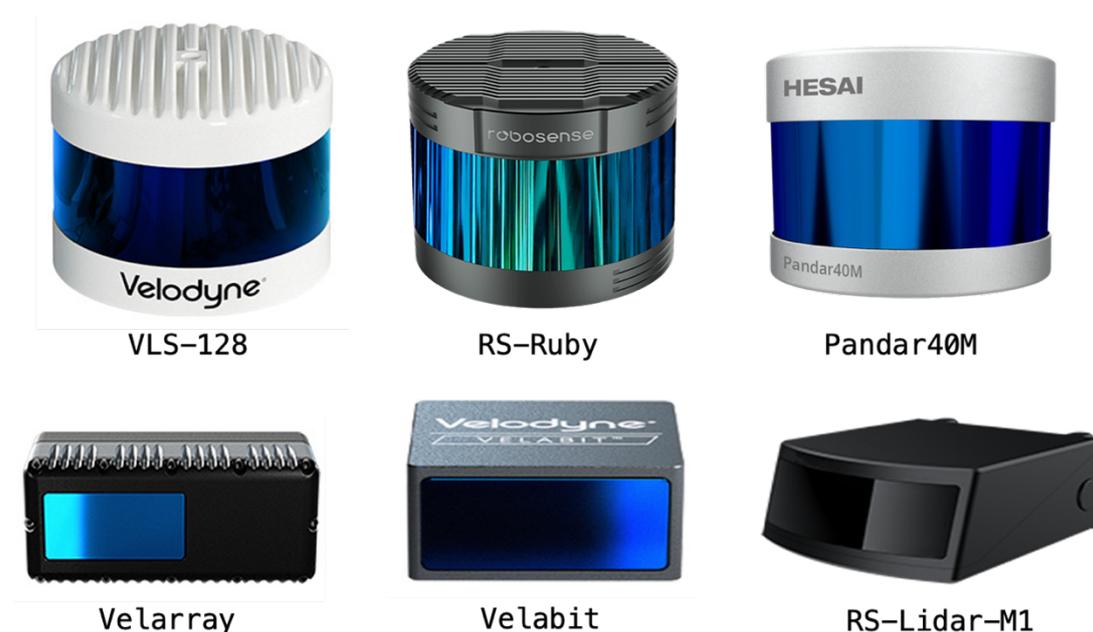


図3-1. 市販されている様々なLIDAR (引用元[37][38][39])

数値表層モデル(DSM)とは、地表面とその上にある地物表面の標高からなる三次

元データで、建物や樹木の高さを含んでいる[40]。センサーから流れているデータは地物の空間位置情報とレーザーの強度情報を持っている。DSM 画像は図 3-2 のように表示している。

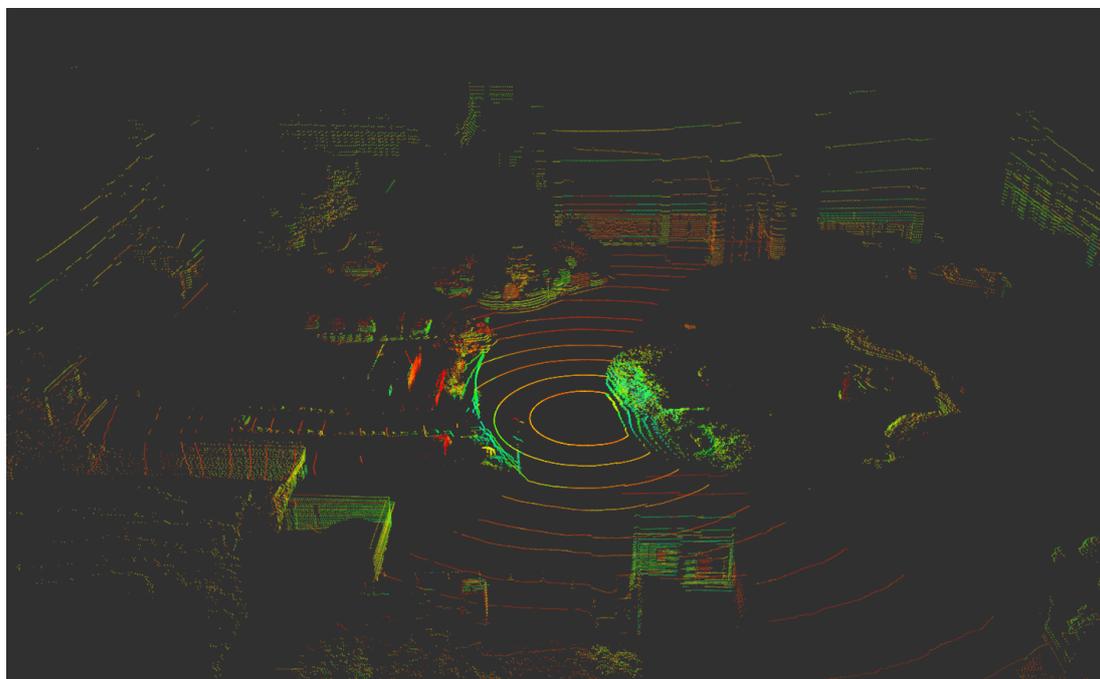


図 3-2. 数値表層モデル(DSM)3次元画像

しかし、LIDAR の一つ欠点は測定距離が遠くほど、測定精度が低くなる。例えば、Velodyne 社の VLP-32C について測定距離は 200 メートル、50 メートル以内の場合測定精度は±5cm、50 メートルから 200 メートルまでの場合測定精度は±10cm である。もう一つの欠点は、905nm 波長の電波は近赤外線(780nm~2526nm 範囲内の電磁波)で、空気中で伝播する時、浮遊しているほこりや水滴の影響を受けやすいため、雨、雪、霧や砂嵐などの悪天候下ではうまく機能しない、検出距離が数十メートルに短縮される。

3. 1. 2 RADAR

RADAR(Radio Detection and Ranging)、日本語はレーダーと呼ばれて、自動運転技術を支える一つ重要な技術である。動作原理は LIDAR と同じ、違うのは使っている電波の波長である。RADAR はよくミリ波レーダーと呼び、波長通常は 4~12mm、波長が長いので、電波が物体に接触したとき、光波の吸収が少ない(減衰が少ない)、雪や霧を通して、はるかに遠くの目標を探知することができる。しかし、欠点として多く

のレーダーの視野角度は 10 度しかない、複数のレーダーを併用することが必要である。例えば、我々の自動運転実験車両(プリウス)では Delphi 社の ESR2.5 と SRR2 を併用している。視野範囲は以下図 3-3 で表示している [41][42]。

- Delphi ESR2.5 (・メインビーム : 174m・ミッド・ビーム : 60m・FOV : ± 10 度、 ± 45 度)
- Delphi SRR2 (・ビーム : 30m・FOV : 80 度)

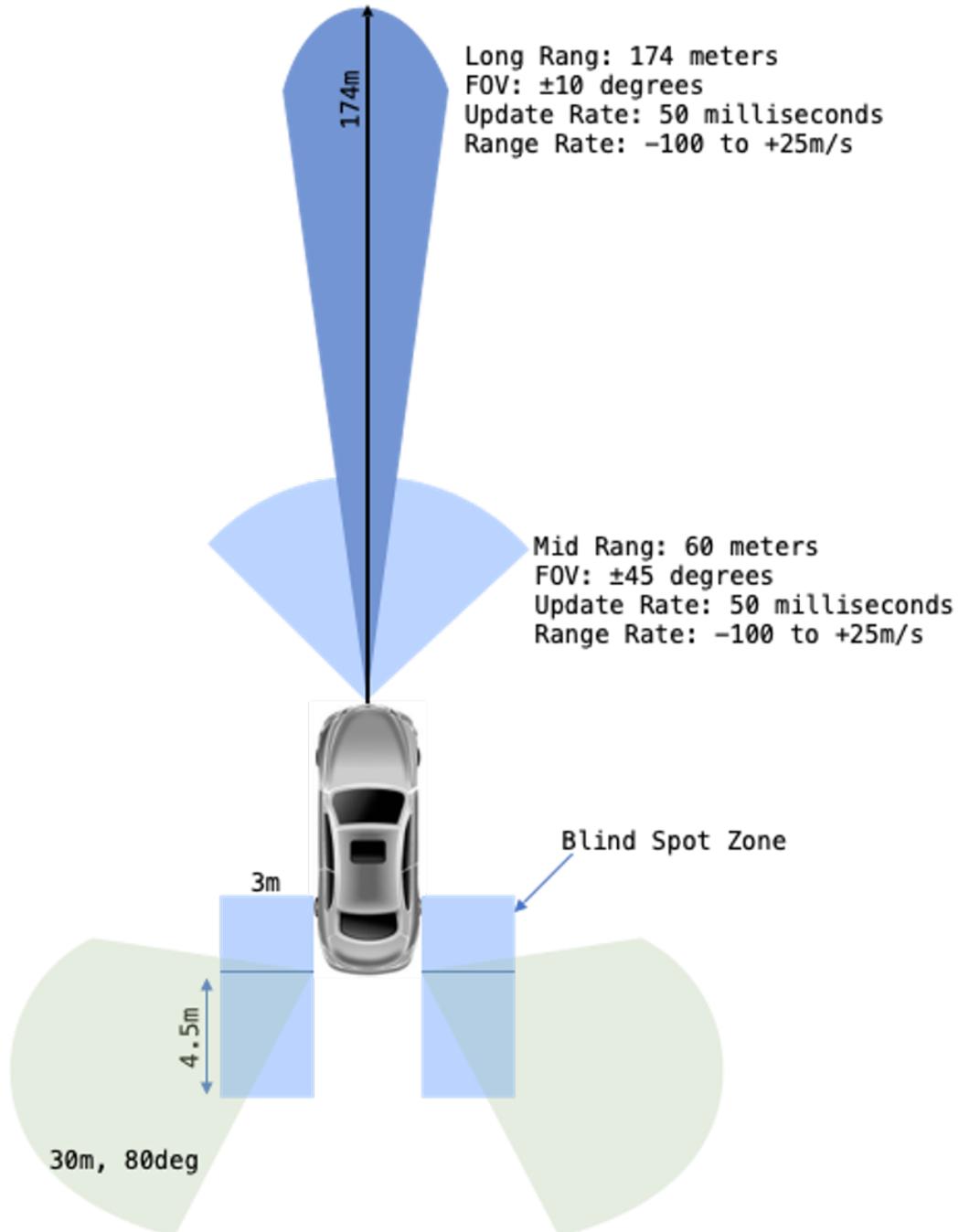


図 3-3. Delphi ESR2.5 と Delphi SRR2 を装備している車の視野範囲

その一方、レーダーとして重要な機能は、ドップラー効果(Doppler Effect) [43]に基づいて、対象の速度情報が取得できる。目標はレーダー近づくとき、反射信号の頻度は発射信号の頻度より高い；目標はレーダーに離れると、反射信号の頻度は発射信号の頻度より低い。頻度の改変値に基づいて、目標との相対速度が算出できる。

3. 1. 3 CAMERA

自動運転車において、Camera は車載カメラを意味していて、LIDAR や RADAR の不足点を補足するために搭載する。車載カメラの利点は対象物の色彩情報を捉え、主に信号識別や障害物検出を利用している。しかし、カメラの動作環境は天候の変化により弱い。代表的な問題所在は車載カメラの画像は霧など悪天候の影響を受けることで、コントラストが低下して、不鮮明になり障害物を正しく認識できなくなる。西日や逆光の場合、カメラで映した映像では明るい部分が白く飛んでしまったり、暗い部分が黒く潰れてしまったりすることがある。図 3-4 は単眼 USB カメラ [44] を表示した、各規格のレンズを付けることが可能である。



図 3-4. 単眼 USB カメラ (引用元[44])

車載カメラは単眼 RGB カメラだけではない。近年、自動運転車に対する障害物を検出のため、遠赤外線カメラ (Far Infrared Rays, FIR) の利用も多くなる。遠赤外線カメラは物体から放射される遠赤外線を映像化できることから、光源に左右されることなく、夜間の遠方撮影や画像認識も可能である。図 3-5 はイスラエルのベンチャー企業 AdaSky の遠赤外線カメラ画像、逆光や夜間でも車両前方の映像がはっきり見える[45]。

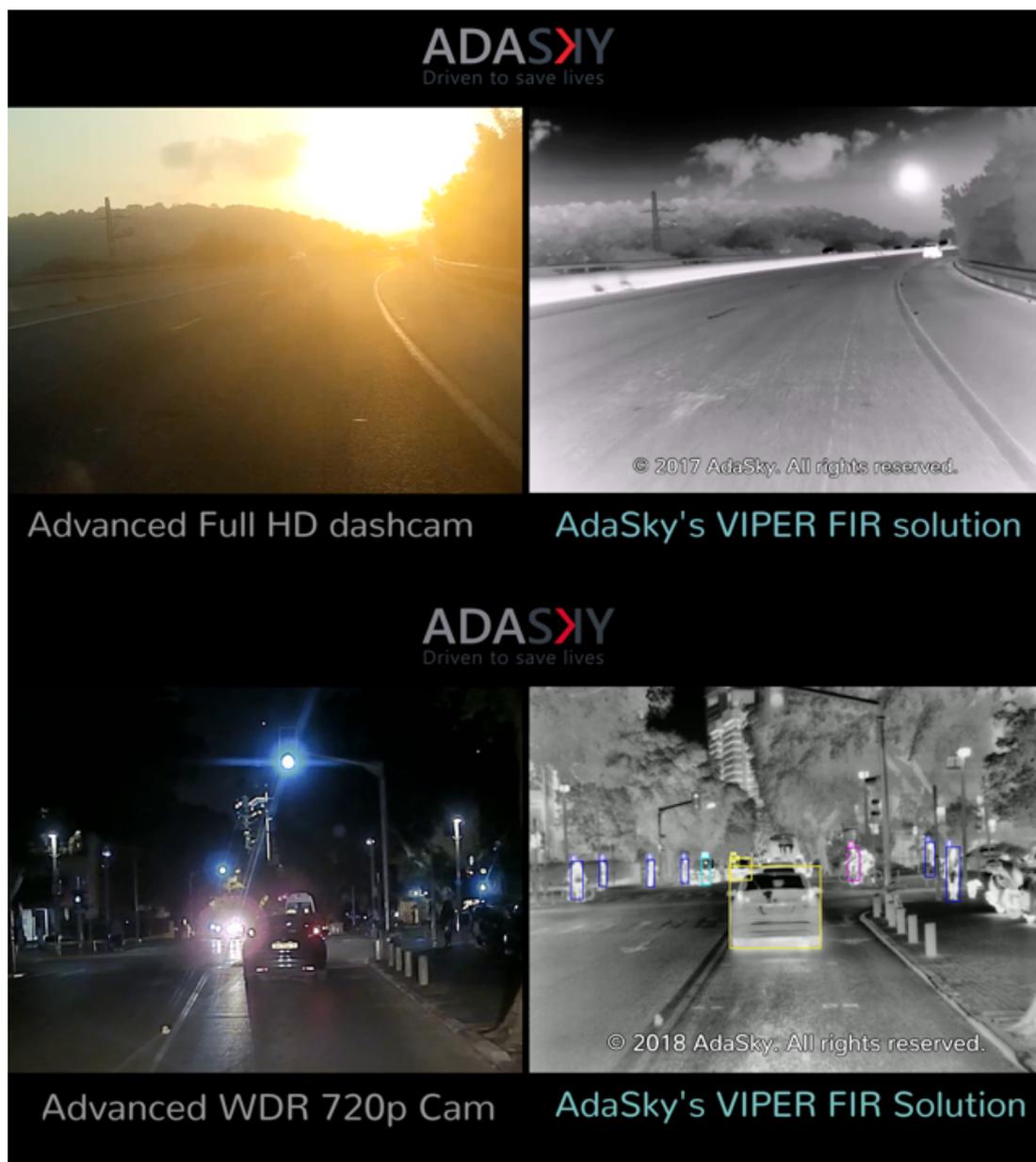


図 3-5. 逆光や夜間の遠赤外線画像 (引用元[45])

高度運転支援システム(ADAS)において、単眼カメラ Mobileye も多数自動運転車に搭載されている。独自のプロセッサチップ(EyeQ チップ)を搭載し、画像処理アルゴリズムを用いて、障害物、白線や車間距離などを検知する[46]。近年、Mobileye 欺いて攻撃の目的を達成することの研究が行われた。図 3-6 ではドローンを利用して、時速 90km の道路標識を投射し、Mobileye を識別させる。極端な状況を考えると、高速走行中の自動運転車前方に歩行者などを投射して、自動運転車に急停止することも可能だと考える[47][48]。

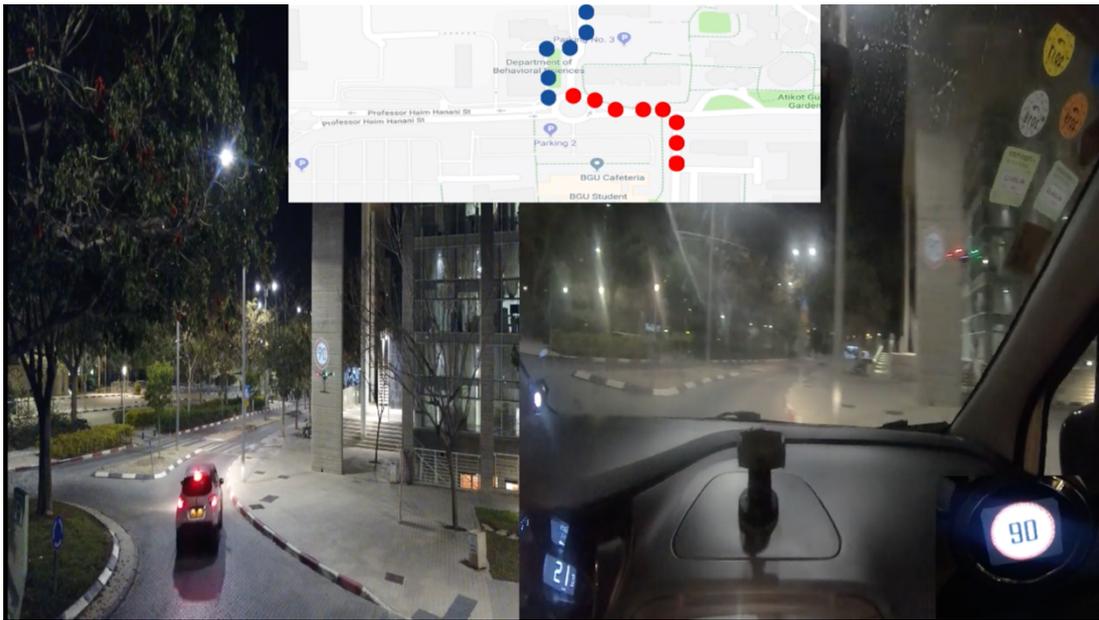


図 3-6. ドローンで道路標識を投射し、Mobileye に識別させる (引用元[47])

3. 1. 4 GNSS

GNSS(Global Navigation Satellite System)、全地球航法衛星システムと呼ばれて、人工衛星から発射される信号を用いて位置測定、航法、時刻配信を行うシステム[49]。GNSS は GPS(米国)、Galileo(欧州連合)、GLONASS(ロシア連邦)、BDS(中国)、QZSS(日本)など各国の測位衛星の総称である。

一般に、スマホやカーナビで使われている GPS には、数メートルの測位誤差がある。これは、GPS 衛星までの距離が長いため、宇宙空間や大気圏内で発生する様々な物理現象により、電波の到達時間揺らぎが発生する。現在、自動運転車の測位システムではよく GNSS-RTK(Realtime Kinetic)技術を使い、センチメートルの精度で自己位置推定を行う。RTK が高精度な測位データを得られる理由は、正確な位置がわかっている基準局の受信データを使う[50][51]。図 3-7 は GNSS-RTK の受信仕組みを示す。

基準局と移動局が同じ信号を受信し、移動局へ受信データをインターネット経由で伝えて、差分計算することで高精度の測定を行う。そのため、基準局に近い場所なら高い精度が得られ、基準局から離れるに従い精度が落ちていく[52]。

しかし、GNSS による測位精度は、周囲の環境に大きく影響される。トンネル、森林や高層ビルが多い市街地など電波が届きにくい場所において、電波が何回反射し、測位精度は数メートルに飛んでしまう。もう二つ測位誤差の要因は、衛星から放送される衛星位置情報が足りないこと、電離層や対流圏による伝播遅延が常に変化していることの二つである[53]。

伝播遅延とは測位信号(マイクロ波)を地表付近の電離層や対流圏に遭遇し、曲折率の関係で速度や経路が変化し、伝播が遅れて、距離測定に影響を及ぼす。

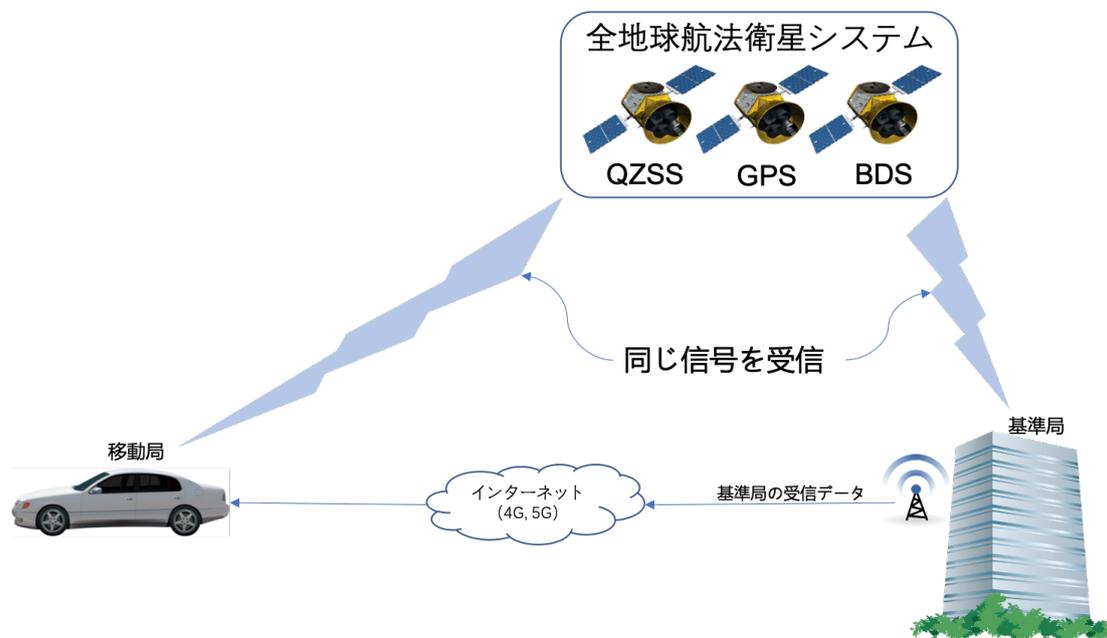


図 3-7. GNSS-RTK の受信仕組み

3.1.5 IMU

IMU(Inertial Measurement Unit)は慣性計測装置と呼ばれている。例として、6DOF の IMU は 3 軸のジャイロと 3 方向の加速度計によって(図 3-8)、3 次元の角速度と加速度が求められる[54]。

自動運転車の場合において、走行中の車両は振動を発生し、出力したセンサー情報(LIDAR、GNSS など)の精度が低下となる。その誤差を軽減するため、IMU データを利用して、劣化したセンサー情報を補正することができる。代表的な技術は GNSS と

IMU 装置を搭載し、移動体の位置と姿勢を算出する。そのため、衛星測位信号が走行中に途切れた場合(トンネル内の走行)でも、何秒間で正確な位置と姿勢測位情報を継続的に提供することができる [55] [56]。

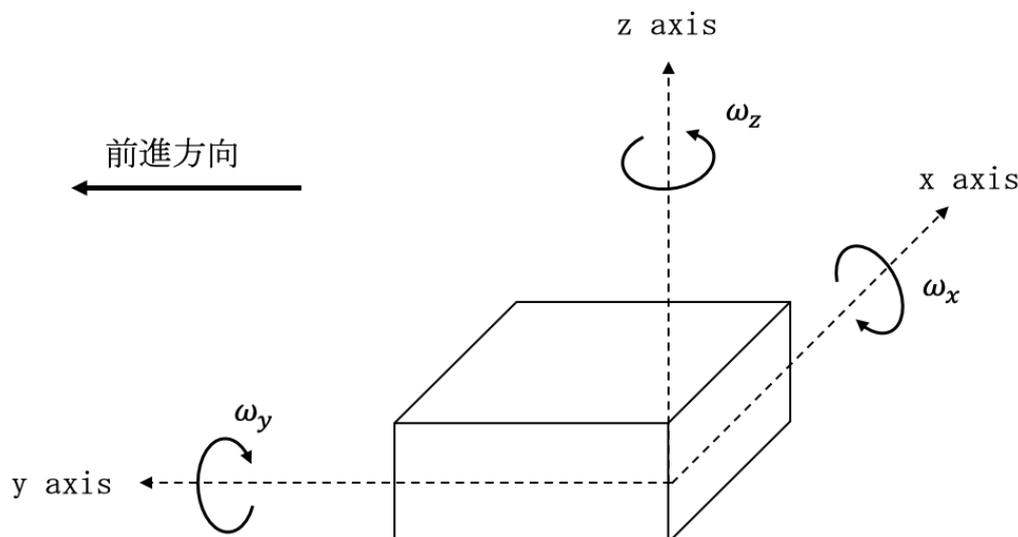


図 3-8. 6 自由度の IMU

3.2 自動運転車のソフトウェア

3.2.1 OS, ROS and Autoware

OS (Operating System) はオペレーティングシステムと呼ばれて、コンピュータのシステムソフトウェアである [57]。主な汎用 OS は Windows、Linux、macOS、Android、iOS などがある。我々は研究のため、全てのプログラムは Autoware に基づいて改造する。Autoware は Linux システムしかサポートしてないから、OS は Linux システムを使っている。Linux システム一つの欠点は実時間処理が保証されていない。ただし、独自の Linux カーネルを開発または改造すれば、独自のソフトウェアに対して、実時間処理が保証できる (Baidu は Apollo の実時間処理を保証するため、独自の Linux カーネルを開発した [58])。その一方、自動運転車企業では、よくリアルタイムオペレーティングシステム (RTOS) [59] を使っている。その名前のとおり、リアルタイム性を持つことが特徴である。RTOS の主要な機能である資源管理において、時間資源の優先度に基づく配分と実行時間の予測可能性を提供することに特化している。

Autoware か Apollo か、どのオープンソースでも ROS (Robot Operating System) に依存する。ROS はロボットのため設計されたソフトウェアプラットフォーム、OS で

はなく、ミドルウェアである[60]。ロボット開発における豊富なライブラリやツールを提供し、様々なロボットやセンサーにも対応し、開発言語はC++とPythonをサポートしている。自動運転システムにおけるハードウェア、ソフトウェア、OSとROSの依存関係は下の図3-9で表示する(写真：TierIV Academy[26])。

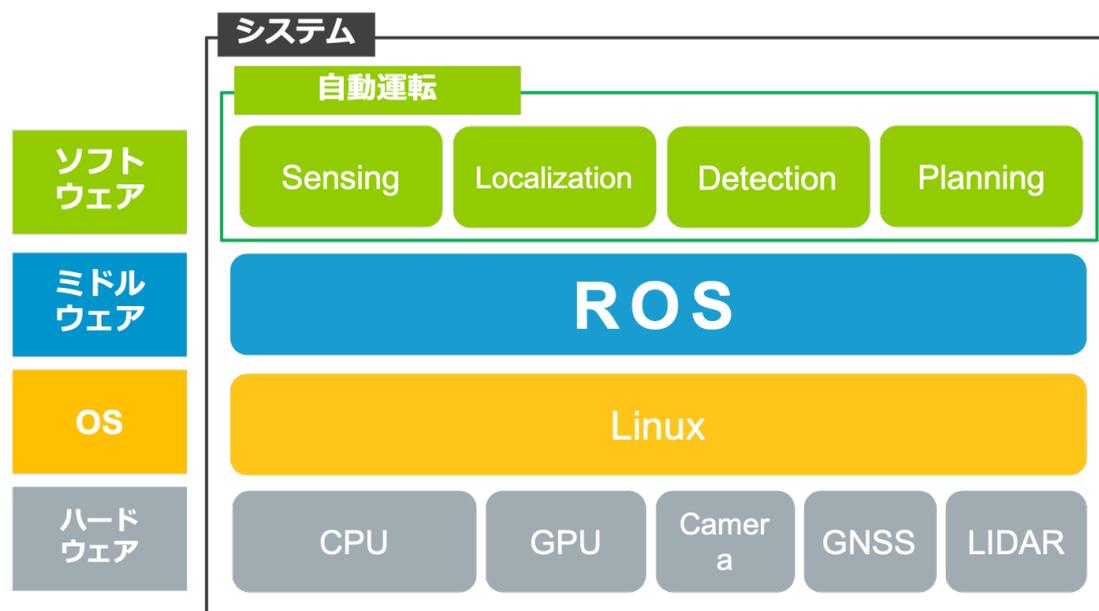


図3-9. ハードウェア、ソフトウェア、OSとROSの依存関係(引用元[26])

3.2.2 HD Map

HD map(High Definition map)は高精度地図であり、自動運転の土台であることを業界に認識している。自動運転企業において、高精度地図の品質によって、企業自身の技術レベルを判断することができる。高精度地図は一般的な車載ナビ地図と違い、3次元の点群データ、車線情報、道路標識、信号機の位置と色情報、更に道路の渋滞情報などをもち、主に車両の自己位置推定、パーセプションやプランニングのため利用される(図3-10)[61][62][63]。

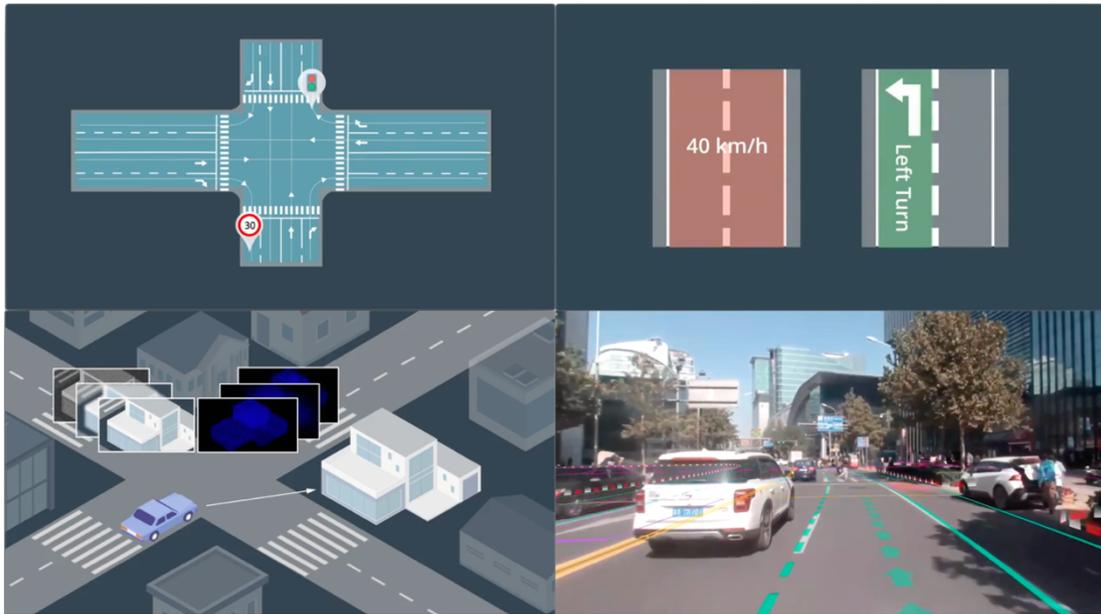


図 3-1 0. 自動運転車へ高精度地図の応用 (引用元[61])

3次元点群データに基づいて、LIDAR 情報と合わせ、自己位置を算出する。夜間、悪天候または信号機までの距離が遠く、信号認識ができない場合、高精度地図から、信号機の色状態を提供する。車線情報について、車が正確な道路の中心線を識別することを役立つになるはず、なるべく中心線に沿って走行する。また、速度制限や歩道区域などに入る前、車速を減らすことを事前に計画する。車両前方は障害物がある時、車線変更が必要となって、高精度地図は車両走行経路の選択範囲を減少することが役立つになる。

高精度地図を作るのは非常に時間や費用をかかり、自動運転車重大な難題である。一般的に、LIDAR、GNSS、Camera、IMU、Odometry の情報が必要で、完成した高精度地図は以下(図 3-1 1)のようになる[64][65][66]。

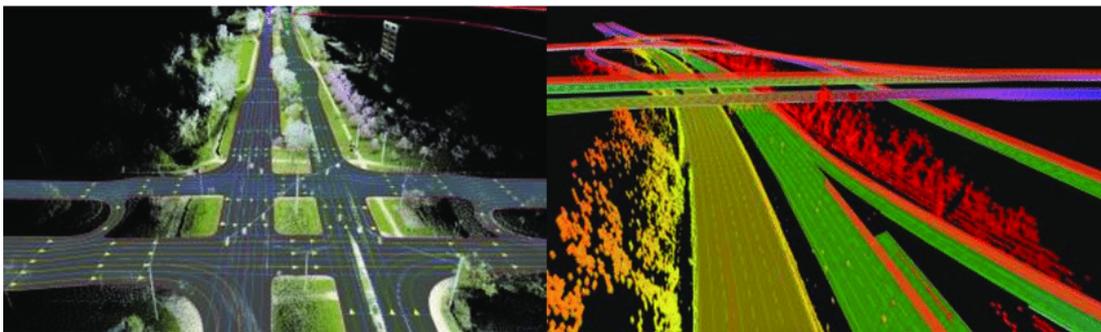


図 3-1 1. 高精度地図の例 (引用元[65][66])

我々は Autoware を基づき、自動運転研究を行うため、簡易版の高精度地図を使っている。点群情報、信号座標と車線の位置情報を含んでいる埼玉工業大学の正門の

地図データは以下で示す(図 3-1 2)。



図 3-1 2. 埼玉工業大学正門の 3 次元地図

3 次元地図を作るため、LIDAR 情報を利用する。Autoware では NDT (Normal Distribution Transform) アルゴリズムを使用する [67]。探索空間内にある点群データを格子状 (ボクセル) に分割し、各ボクセル内の点群データを正規分布し、前回のスキャンデータを参照しながら、二つの点群データをマッチングする。点群データを収束するまで、非常に時間かかることが欠点である。NDT mapping を用いて、埼玉工業大学正門の地図を作成する様子は図 3-1 3 で示す。

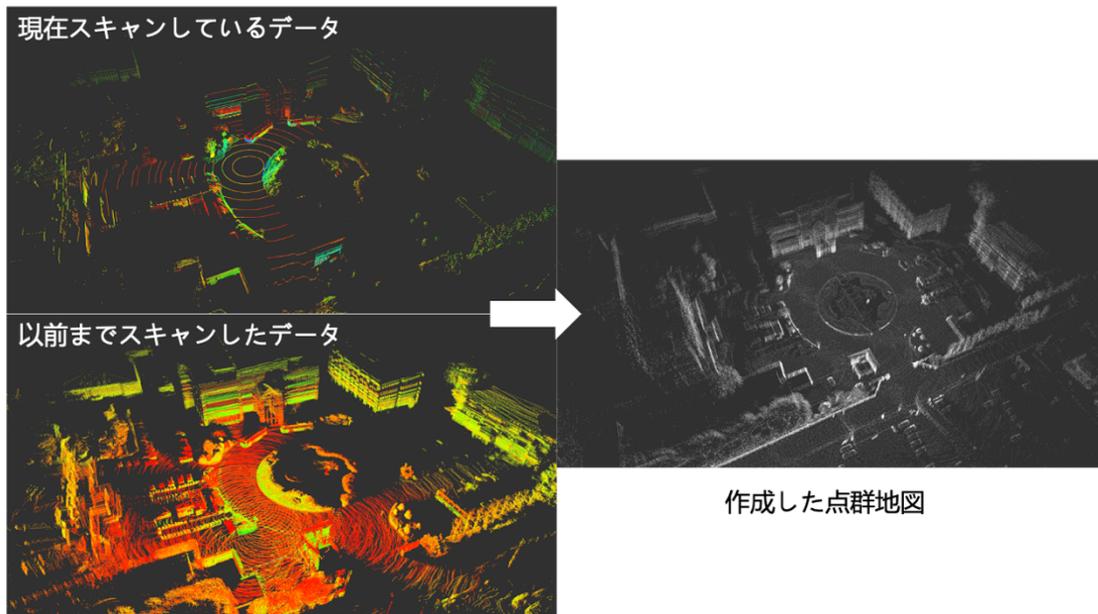


図 3-1 3. 地図を作成する様子

現在、地図生成主な手法は SLAM(Simultaneous Localization and Mapping)である。同時に自己位置推定と地図生成を行う手法、代表的なのは Zhang ら提出した LOAM である[68]。LIDAR の生データを Point Cloud Registration 処理(二つの Point Cloud を整列させる空間変換を見つけるプロセス)を行う。LIDAR Odometry は 10Hz で LIDAR の動きを推定しながら、点群データの座標変換を行う。LIDAR Mapping は 1 Hz で 3 次元地図を作成する。Transform Integration は 10Hz で 3 次元地図の最適化を行う。ブロック図 3-1 4 は以下のように表示する。

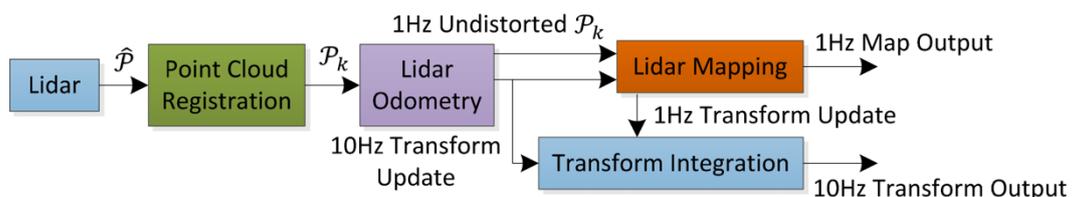


図 3-1 4. LOAM 手法のブロック図 (引用元[69])

LIDAR で 3 次元データを収集する時、周辺の環境は変わる可能性が高いと考える。風が吹いて、木の枝や草はいつも動いて、スキャンしたデータはいつも違う。LIDAR に対して、この動きはノイズとして、生成した三次元地図の精度が悪い。この問題を解決するため、地面、樹木や草など不固定物を除くことが必要である。LeGO-LOAM[69][70]は LIDAR の動きを推定する前に、セグメンテーション(Segmentation)と特徴点抽出(Feature Extraction)を行う。LeGO-LOAM のブロック図 3-1 5 は以下の

ように表示する。

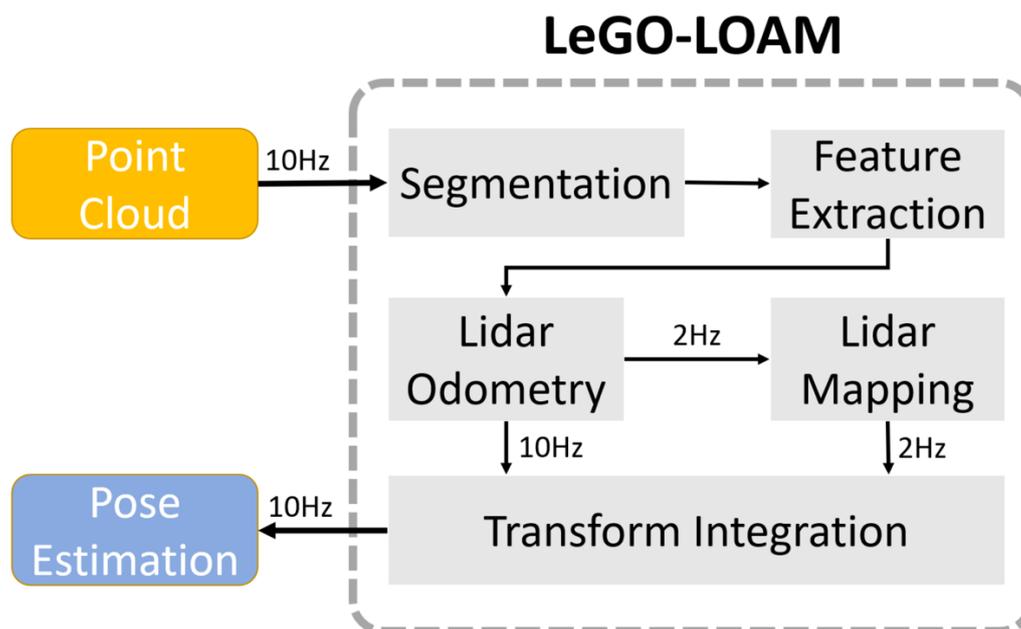


図 3-1 5. LeGO-LOAM 手法のブロック図 (引用元[70])

3. 2. 3 Localization

Localization とは自動運転車の自己位置を推定すること。車がどこにいるかを正確に知ることが必要、XY 座標だけではなく、車の前進方向も知ることが必要である。一般的な自己位置推定方法は GNSS-RTK、LIDAR、慣性航法装置 (Inertial navigation System, INS) または視覚 (Visual localization) に依存し、それぞれに独自の長所と短所がある。

GNSS-RTK は前節のように記述された。RTK を利用して、位置推定の精度は通常 10cm 以内。ただし、トンネルや建物が多い市街地など電波が届きにくい場所において、電波が何回反射し、位置推定の精度は数十メートルに飛んでしまう。もう一つの欠点は、GNSS の更新頻度が遅い (10Hz)、時速 60km の車両は 1.667m を走った。

LIDAR の自己位置推定は高精度地図に基づいて、LIDAR 情報と点群データをマッチングして、自己位置を算出する。主に二つ点群データについて平行移動と回転行列を求め、誤差を最小化にする。代表的な手法は ICP (Iterative Closest Point) [71] [72] [73] や NDT マッチング [74]、またはヒストグラムフィルタ (Histogram filter) [75] などがある。ただし、点群データをマッチングするのは計算量が多くことは欠点になる。近年、GPU などを利用の上、実時間処理が可能になった。NDT マッ

チングによる自己位置推定の様子は以下の図 3-1 6 で表示する。(白い点は 3 次元地図データ、色付きの点は LIDAR のスキャンデータ。)

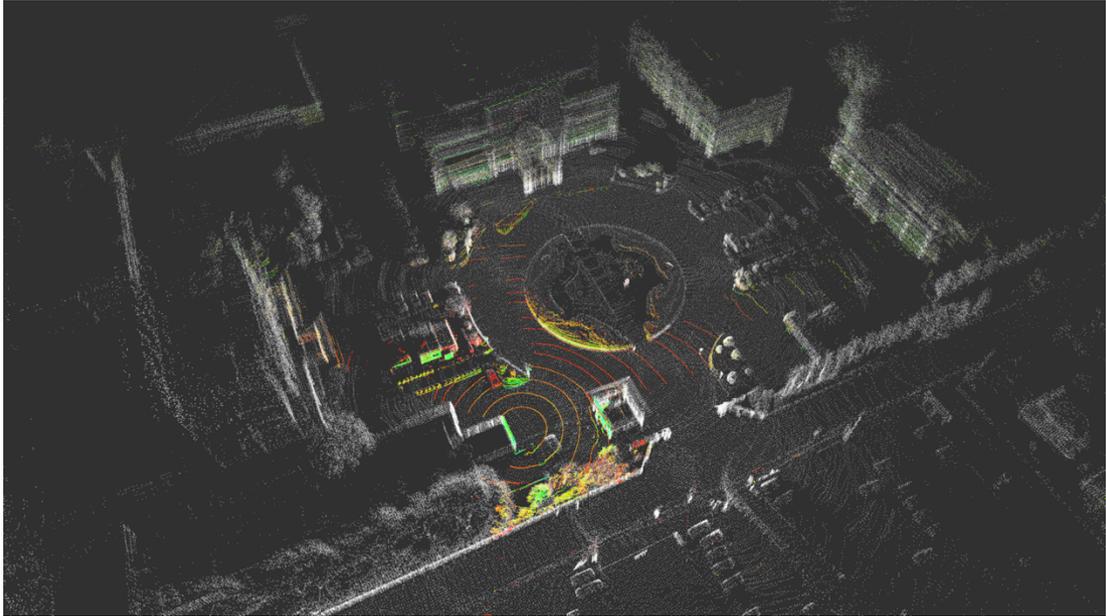


図 3-1 6 . LIDAR で自己位置を推定する様子

慣性航法装置による自己位置推定はよく補助手段として使われている。INS は車両の加速度情報 a を提供することが可能となる。

車両の初期位置 S_0 、初期速度 v_0 と走行時間 t によって、自己位置 S は式 3-1 で計算する。

$$S = S_0 + v_0 t + \frac{1}{2} a t^2 \quad (3-1)$$

車両の加速度情報(測量値)は世界座標系には Gyroscope(ジャイロ) [76] に依存する。Gyroscope の Spin axis と Rotor を固定されて、三つの外部ジンバル(Gimbal)を回転させて、相対値を測定し、物体の角度、角速度あるいは角加速度を検出する(図 3-1 7)。

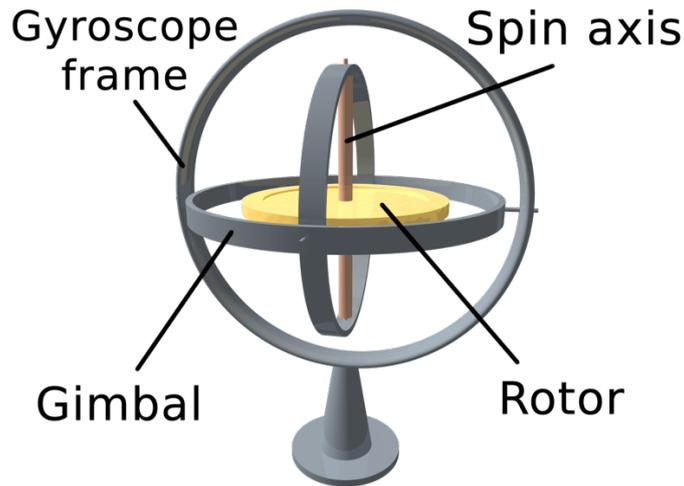


図3-17. 3D ジャイロ (引用元[76])

現在、市販されている IMU はほぼ INS と Gyroscope を内蔵されている。IMU の更新頻度が 1000Hz まででき、実時間の位置情報を提供することは可能である。ただし、IMU の測量誤差は時間と共に増加することが最大の欠点で、そのため短時間だけの位置推定は可能である。この問題を解決するため、GNSS-RTK 測位と組み合わせることが必要である。IMU は GNSS の低い更新頻度を補い、GNSS も IMU の運動誤差を修正する。

単純に視覚による自己位置推定[77]はかなり難しく、常に他の方法(GNSS-RTK)と併用し、より良い位置推定結果を得ることができる。図3-18を例として、RTK から車両の位置は両車線の真ん中に推定された。カメラから識別した車線は赤い車線で表示し、現実の道路車線と合わせて、正しい位置推定を行った。

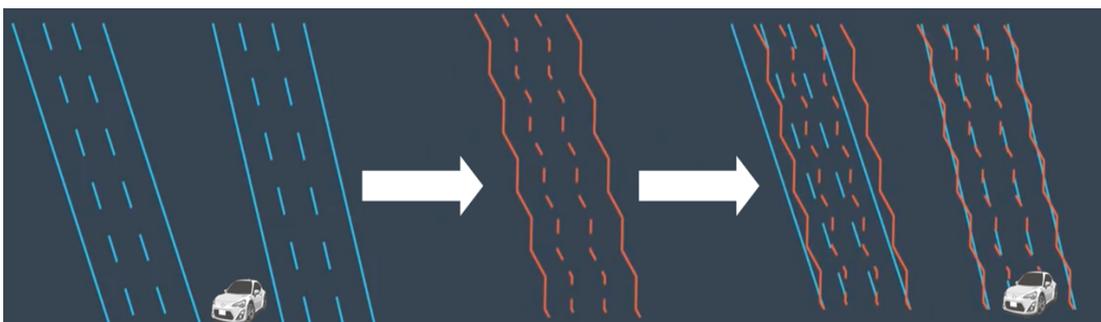


図3-18. 視覚による自己位置推定の例 (引用元[61])

それぞれのセンサーをフュージョンにより、それらの欠点を補うことが現在主流なソリューションになっている。代表的な手法はルドルフ・カルマン提案したカルマンフィルター(Kalman Filter)[78]である。カルマンフィルターは、情報が不明確な動的システムである限り、システムが次に何をするかについて、根拠に基づいた推測

を行うことができる。カルマンフィルタは継続的に変化するシステムに適して、メモリの消費量が少ない(一つ前の状態を保持するだけ)という利点があり、非常に高速であるため、リアルタイム問題や組み込みシステムに最適である[79]。

3.2.4 Perception

Perception は日本語で感知といい、自動運転車に対しては世界を感知すること。人間自身は色々なセンサーがあって、目で見ると、耳で聞く、手で触る。自動運転車も色々なセンサーを搭載し、周辺の環境を感知する。近年この分野はコンピュータビジョンと呼ばれて、感知による四つのコアタスクは検出(Detection)、分類(Classification)、追従(Tracking)及びセグメンテーション(Segmentation)。

1. 検出：視野範囲内物体の位置を見つける。
2. 分類：見つけた物体を明確に識別する。
3. 追従：時間と共に移動物体(車、歩行者など)を観察する。
4. セグメンテーション：カメラ画像の各ピクセルを意味分類に一致させる。

現在、自動運転において、主流となっている感知技術は人工知能の深層学習を利用した方が多い。カメラは画像処理のため、YOLO[80][4]、Faster RCNN[81][82]などの物体認識(図3-19)手法があり、それ以外、特に悪天候の場合、カメラの視認性を高めるため、ヘイズ除去の手法も提案された[32][33][34]。画像セグメンテーションによる SegNet[83][84]なども提案された(図3-20)。

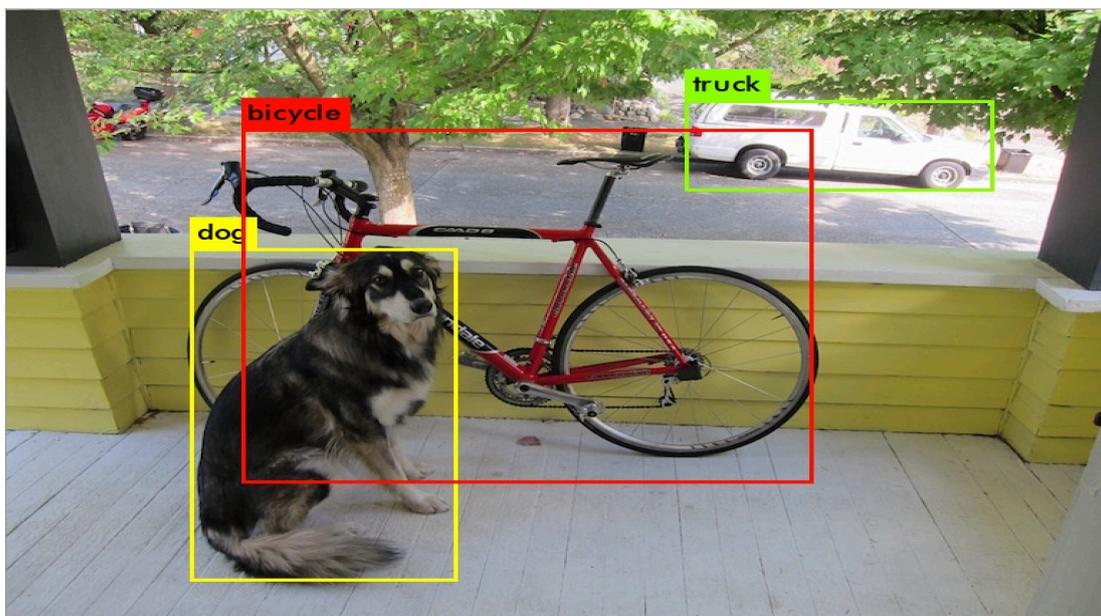


図3-19. カメラによる物体認識(引用元[4])

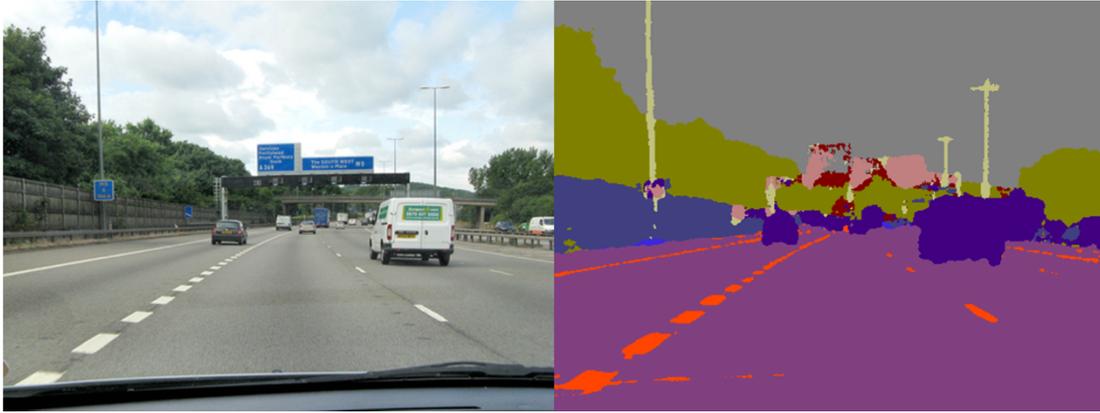


図3-20. 画像セグメンテーション(引用元[84])

近年、LIDAR の価格が安くなると共に、LIDAR において障害物検出手法が主流となっている[85][86][87]、図3-21の中では車を認識して、Rviz上に青いボックスで表示された。

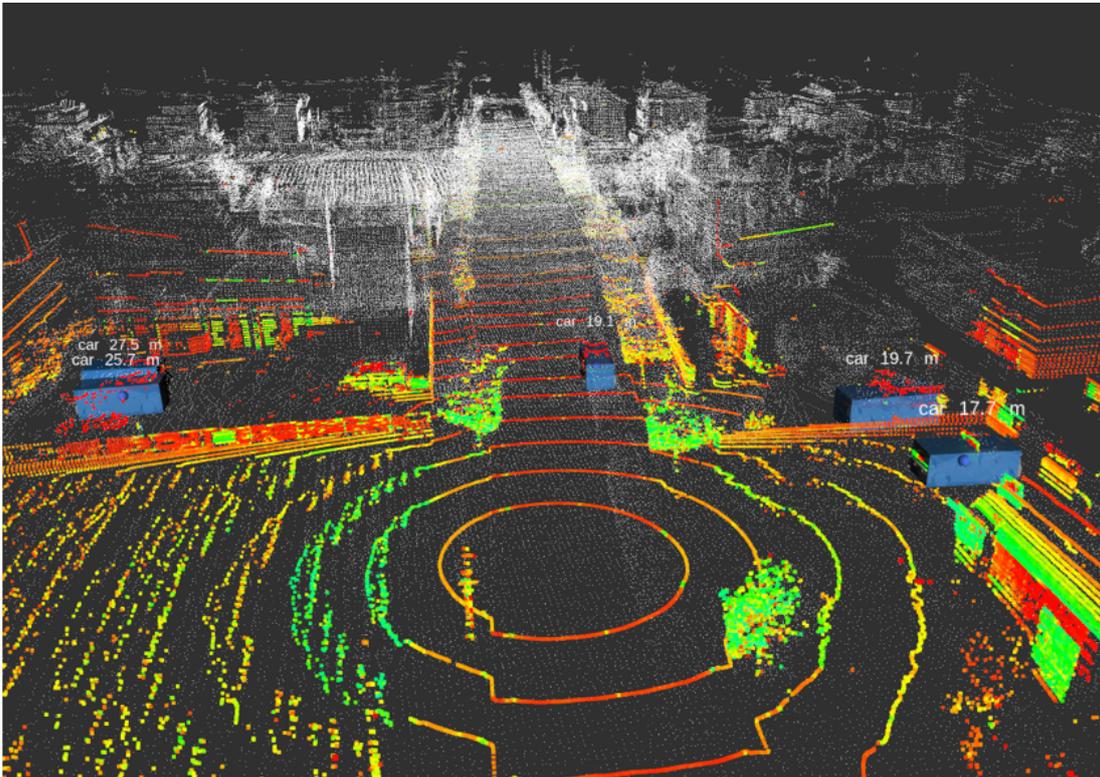


図3-21. LIDARによる障害物認識(Point Pillars)

3.2.5 Prediction

Predictionとは自動運転車を周りの移動物体(車、自転車、歩行者など)の行動を分析しながら、走行経路を予測すること。走行中の車は、検出された全ての物体に対する行動予測を行う必要があり、予測経路を生成する。また、一定の時間ごとに、新

しく行動予測を行って、再計算の上に予測した経路を更新することが必要である。予測された経路は、自動運転車のプランニング段階に必要な情報提供する。

予測するには二つの方法があり、モデルに基づいた予測 (Model-based prediction) [88]とデータで駆動する予測(Data-driven prediction) [89]である。モデルに基づいた予測の利点は、既存の物理知識、交通法則や常識などを組み合わせて、直感的な方法で理解しやすい。データで駆動する予測の利点は機械学習に基づき、トレーニングデータが多いほど、より良い予測結果が得られる。

図 3-2 2 では前方の車両(青い車)を追い越す際に、経路を予測する例を表示した。赤い線は元々の走行経路であり、三つの予測経路を算出した。

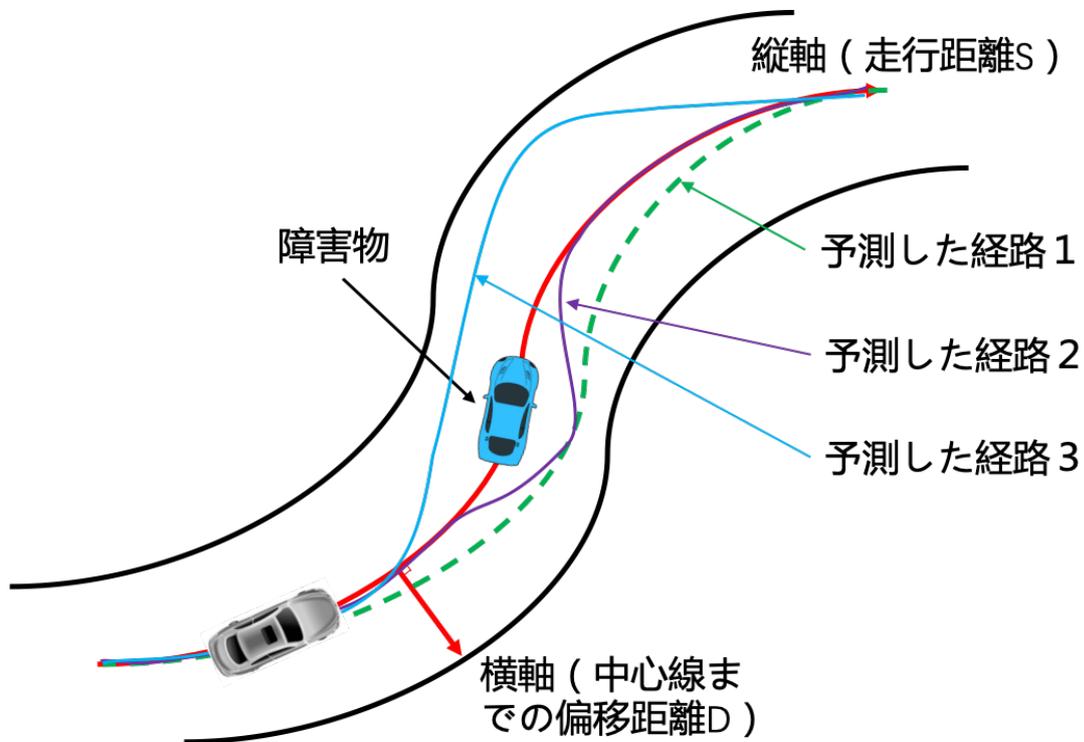


図 3-2 2. 車両を追い越す際に予測した経路

3. 2. 6 Planning

Planning とは高精度地図、自己位置推定や予測した経路を組み合わせ、車両の走行経路を生成すること。プランニングには経路計画(Road Navigation)と軌道生成(Trajectory Generation)二つの部分で構成されている。

経路計画の目標は、地図上で A から B への最適なルートを見つける。代表的なルートを探る方法はスタンフォード大学から提案し、2007 年の DARPA グランドチ

ランレンジで応用された Hybrid A* [90] 手法である。

軌道生成の目標は見つけた最適なルートを走行する際に、障害物により進路が遮られた場合、障害物を回避できる軌道を生成すること。そのため、コスト関数を用いて、候補ルートを探査し、コストが最も低いルートを選択する。コスト関数は以下の要素を考慮する必要がある：

- 車線中心に対する偏移距離
- 障害物までの距離
- 車速や曲率の変化
- 車両へのストレス

代表的な軌道生成手法は Lattice Planner [91] があり、3次元の問題を二つの二次元の問題を分解する。Frenet [92] 座標系における縦軸 (ST、Shifting and Time) と横軸 (SL、Shifting and Lateral) それぞれの軌道を生成し、組み合わせる手法 (図 3-23)。

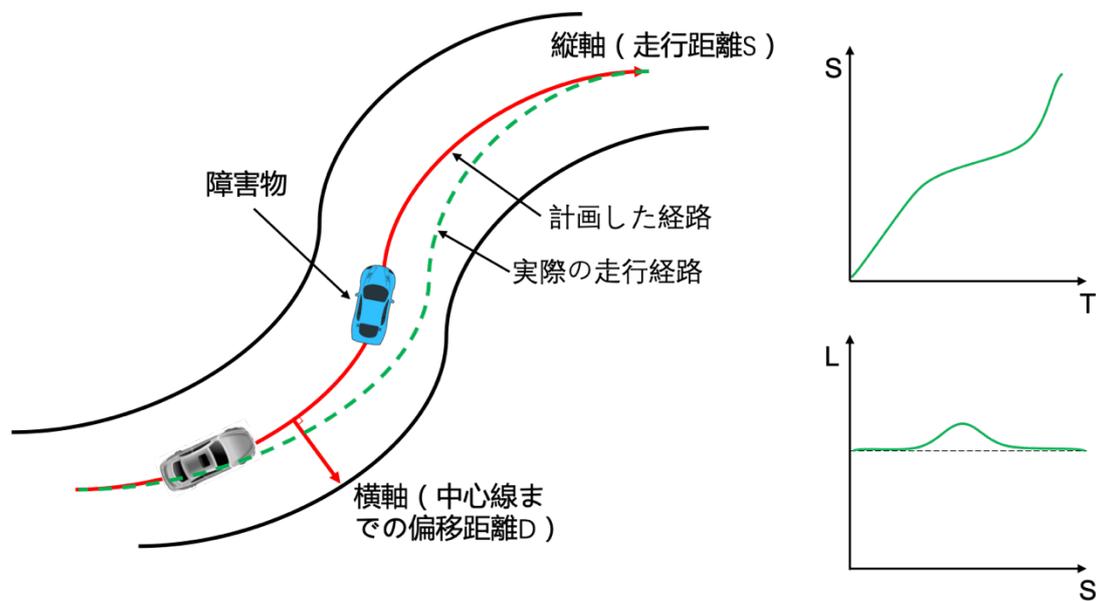


図 3-23. Frenet 座標系における Lattice Planner で軌道生成

3.2.7 Control

Control とはステアリング (Steering)、アクセル (Acceleration) とブレーキ (Brake) を入力し、車両を前進させること。通常、コントローラーは一連の位置情報、速度情報やステアリング情報を含んでいるウェイポイント (Waypoint) を介して走行予定の経路を受け取り、それぞれのウェイポイントを入力し、走行予定の経路をスム

ーズに通過させることである。

コントローラーは次のような特徴を持っている必要がある(図3-24)。

- 高精度：路面が濡れている場合や路面が急な場合でも、走行予定の経路から逸脱しないように車両をコントロールする。
- 実現可能性：ゲームの中に、車両は簡単に90度回転できて、車速も時速200キロまでスピードアップすることができる。ただし、現実世界では実現できない。
- 安定性：乗り心地は非常に重要であり、急な加減速や急ハンドルなど非スムーズな運転を避けることが必要である。

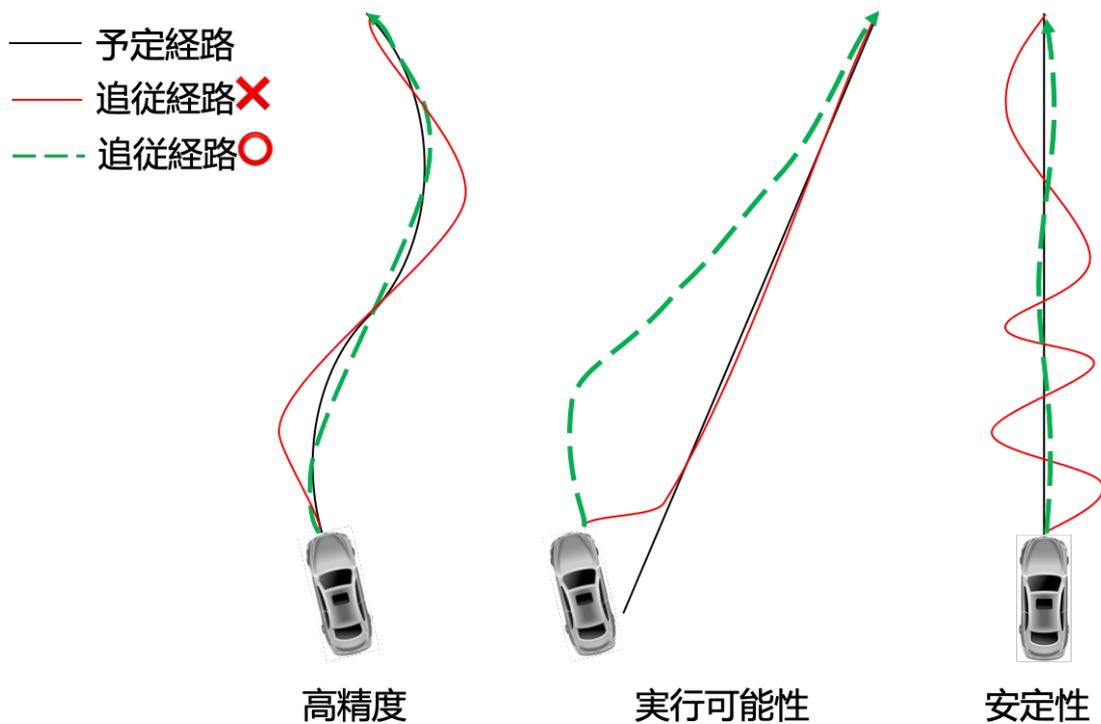


図3-24. コントローラーに必要な三つの特徴

自動車業界に最も使われている制御手法はPID(Proportional-Integral-Derivative) [93]制御となっており、フィードバック制御の一種である。PID制御最大の利点は非常に簡単であり、多くの場合、制御効果が良い。ただし、PID制御は単なる線形コントローラーであり、非常に複雑なシステムには適していない。自動運転車の場合、ステアリングとアクセルを制御するには、異なるPIDコントローラーが必要である。そのため、多くの解決策[15][94][95]は、速度制御(Longitudinal Controller)にはPIDコントローラー、ステアリング制御(Lateral Controller)にはLQR(Linear-quadratic regulator)コントローラーを使用する。PIDコントローラーもう一つの欠

点はリアルタイムの測定誤差に依存しており、測定が遅延を発生すると PID 制御は役に立たなくなる。

学术界において、自動運転車の制御手法 MPC (Model Predictive Control) [96] 制御が汎用されている。非常に複雑なコントローラーであり、数学に依存度が高くて、パラメーターの調整は非常に時間かかる。基本的には、三つのステップに分かれる。

- ステップ 1 : 車両モデルを構築する。
- ステップ 2 : 一定時間の制御入力による計算や最適化を行う。
- ステップ 3 : 最初の制御コマンドを実行して、プロセスを繰り返す。

第4章 埼玉工業大学の自動運転車

4.1 RoboCar

RoboCar[97]とは、プログラム制御により走る、曲がる、止まるなどの制御が可能な自動運転を開発するための車両プラットフォームであり、自動運転の開発ツールとして、ZMP 会社から販売されている（図4-1）。



図4-1. ZMP が販売している様々な RoboCar (引用元[97])

埼玉工業大学の RoboCar(図4-2)はプリウスとベースとして、自動運転の研究を行っている。車両はLIDAR、RADAR、GNSS、Mobileye などセンサーを搭載し、Autowareを利用して、自動走行が可能となっている。



図4-2. RoboCar による開発した自動運転車

4.2 自動運転マイクロバス

4.2.1 ジョイスティック式自動車

日本にジョイスティック式自動車が生産された前に、英国 Steering Development 社の Space Drive と米国 Ahnafield 社の JoyVan が存在している[31] (図4-3)。



図4-3. 米国 Ahnafield 社の JoyVan (引用元[31])

車両に標準装備された油圧ポンプと油圧アクチュエータによる駆動システムを基本としている。特に、ステアリングホイールの駆動は、車両に備えられている油圧式のパワーステアリング用の油圧モーターを強化することで実現している。しかしながら、日本の場合、直接車両の油圧ポンプと油圧パワーステアリングを改造すると、改造された車両の安全性は確認できなくて、車検が通られない。そのため、埼玉工業大学の自動運転バスは既存のジョイスティック式自動車運転装置を利用する。

4.2.2 ジョイスティック式自動車運転装置

図4-4はジョイスティック式自動車運転装置の概念図を表示している。ジョイスティックを操作する信号に従い、ハンドル、およびペダルの接続された電動モータを駆動することで、それぞれの運転装置を作動させる。

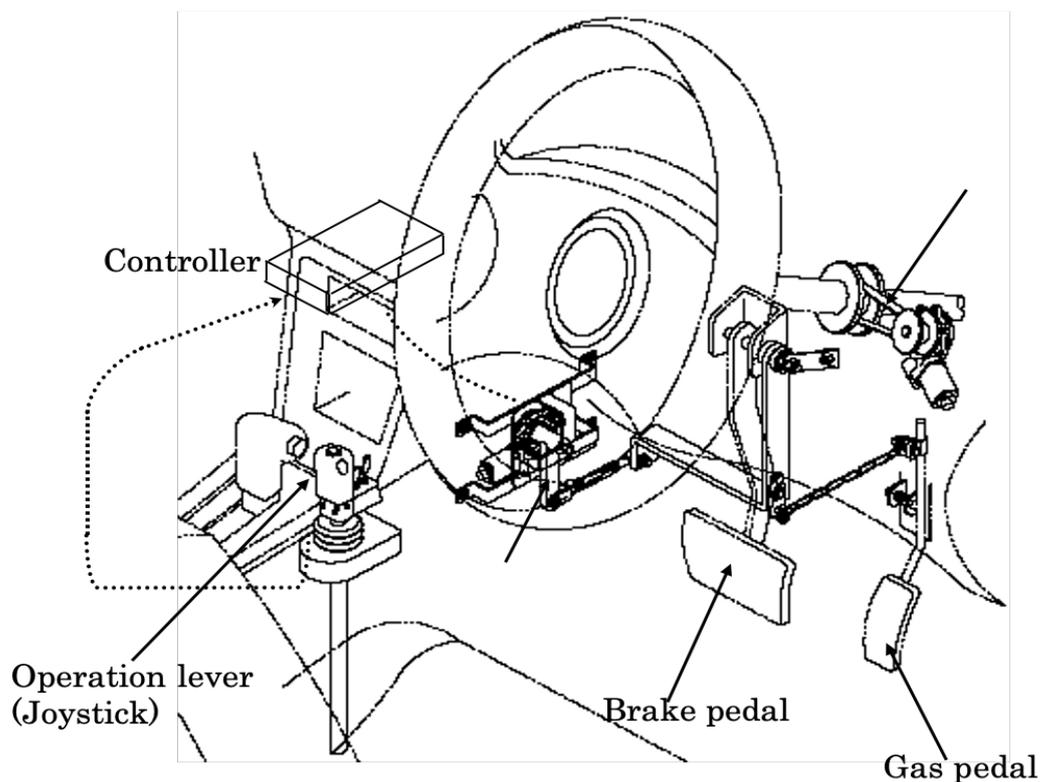


図4-4. ジョイスティック式自動車運転装置 (引用元[31])

電動モータの動力は、歯車や電磁クラッチ、センサ、リンクなどから構成される駆動ユニットをハンドル駆動用、ペダル駆動用にそれぞれ1つずつ搭載する (図4-5)。

図4-5はモーター制御システムを表示する。電動モータはマイクロコンピュータを備えるコントロール基板による制御される。ジョイスティックから与えられる信号 θ^* を角度指令として認識し、それに比例した変位 θ をハンドルやペダルで実現するようフィードバック制御が実行される。

制御システムの安全性を確保するために、各種センサーの断線検出とモーターの過電流、電源電圧などを監視しながら、システムの運用を行う。

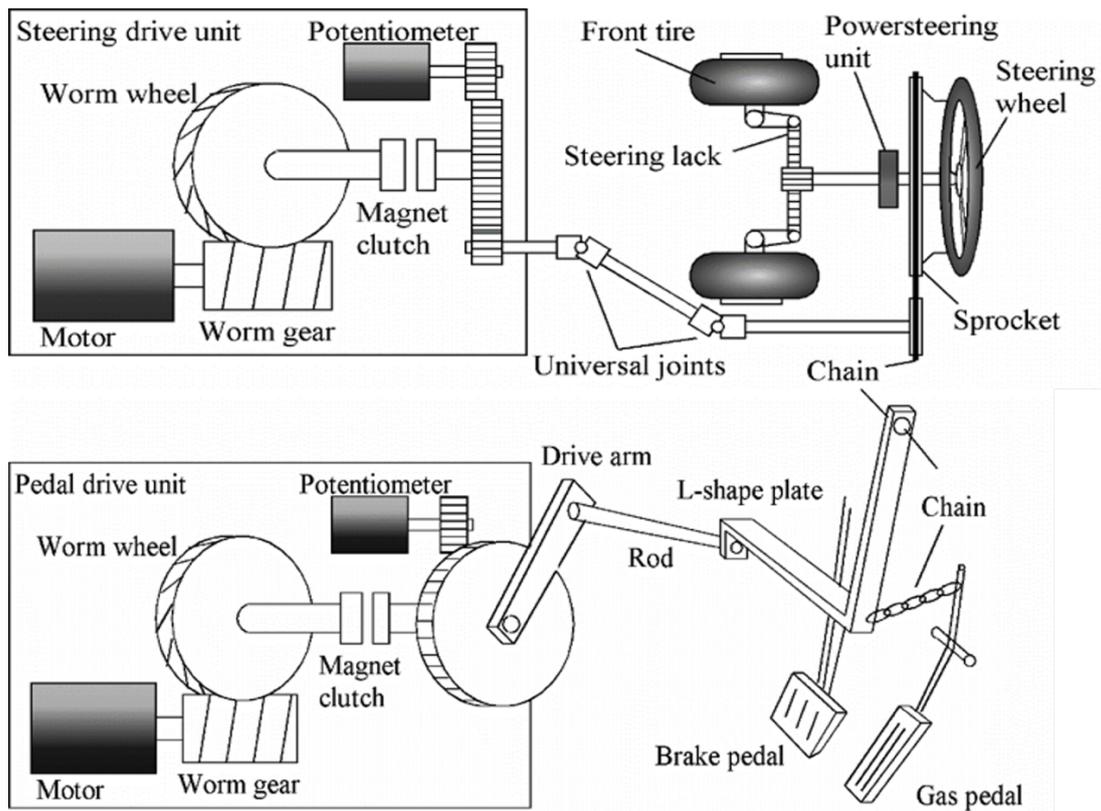


図 4-4. モーター駆動システム (引用元[31])

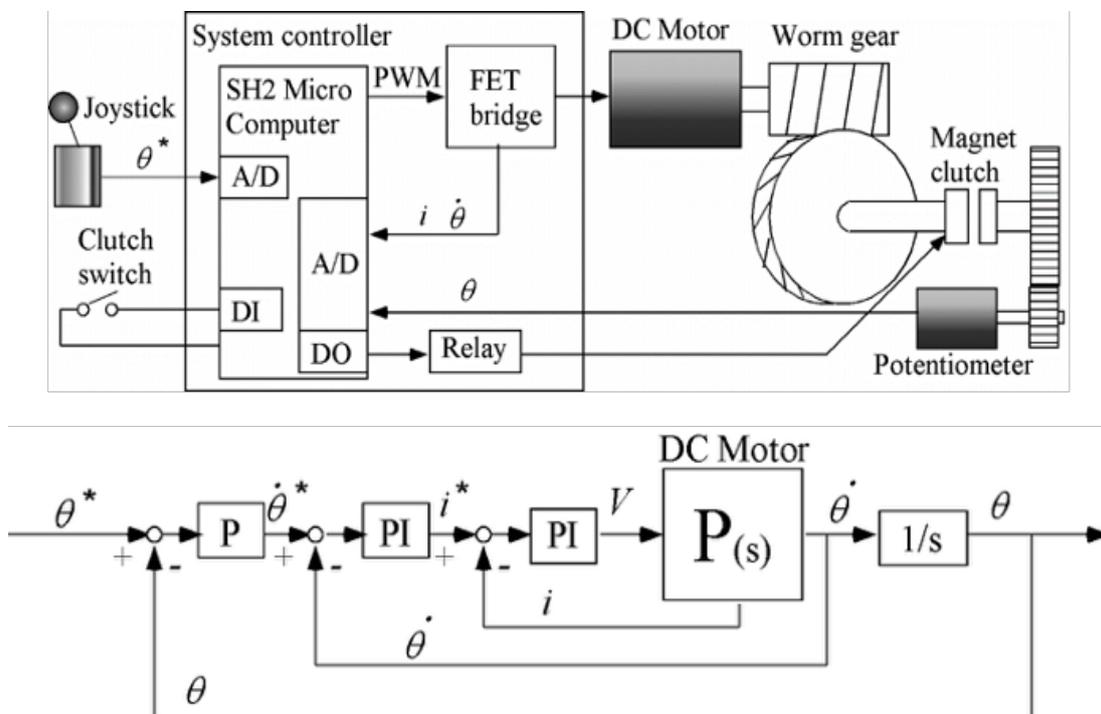


図 4-5. モーター制御システム (引用元[31])

自動運転マイクロバスを構成しているシステムは図 4-6 で示す。

二つのジョイスシステム基本機能を統合する車両運動制御装置 (Vehicle Motion

Controller) を開発した。VMC と Autoware PC の情報交換は CAN 通信によって行う。さらに、VMC にはシフトレバーの動作制御やウィンカー、ライト、ホーンなどの車両走行時に操作が必要な装置の ON・OFF 制御と車速、エンジン回転数などの車両走行を管理する機能を搭載した。

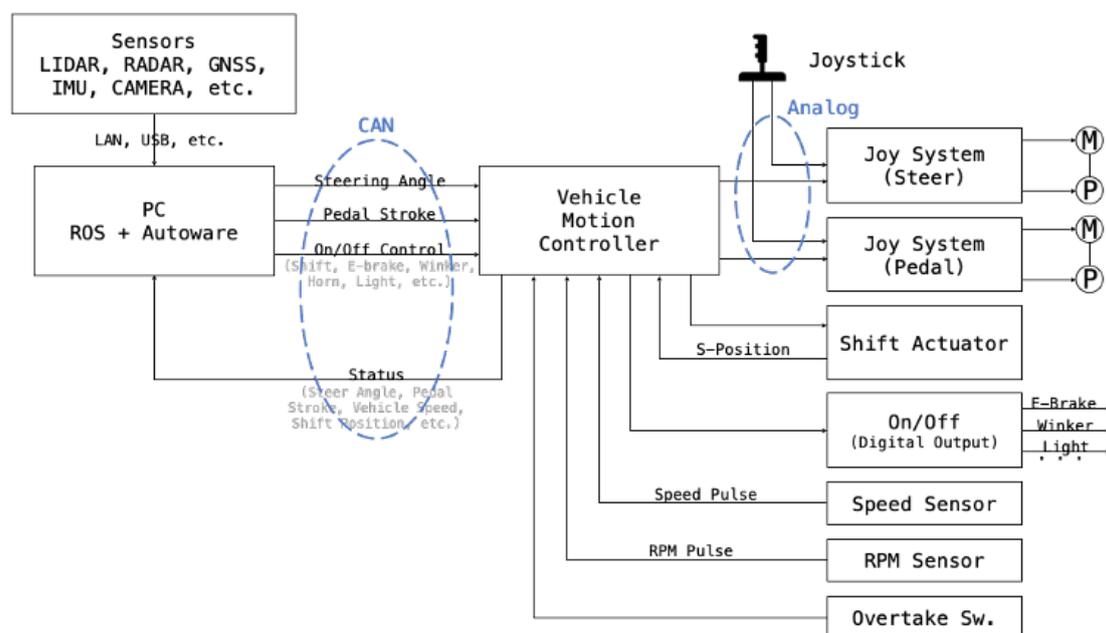


図 4-6. 自動運転マイクロバスの車両制御システム (引用元[31])

4.2.3 埼玉工業大学の自動運転マイクロバス

埼玉工業大学の自動運転マイクロバスはジョイスティック式自動車運転装置 [31] を応用し、様々なセンサーを搭載し、バスの自動運転システムを構築している。埼玉工業大学の自動運転システムはどんな車両でも後付けができることは特徴であり、日本の車検でも問題なく通られる。高度な安全運転支援機能の後付けにより、バスの運転に自信がない方でも、バスによる送迎ができるようなバスができれば、運転手不足、赤字路線など地域の公共交通問題を解決できます。

埼玉工業大学自動運転マイクロバスは車両の販売を目指して、研究が行われている。車体の大きさを考え、LIDAR は 9 個付いていて、信号識別のため、単眼カメラは 3 個 (レンズは 15mm、20mm、30mm) 付いている。埼玉工業大学以外に、販売を目的としていない先進モビリティの自動運転バス [98] (GNSS と磁気マーカーを利用し、最大時速 70 キロ) と群馬大学の自動運転バス [99] (GNSS と LIDAR を利用し、最大時速 25 キロ) がある。各自動運転バスは図 4-7 で示す。



図4-7. 日本の自動運転バス（引用元[98][99]）

図4-8と図4-9は実装した運転席の様子である。ジョイスティック式運転装置を搭載した後も運転者は通常と同様に運転席に座り、手動運転を行うことや自動運転の際には緊急時に備えて車両や操作装置の様子を監視することが可能である。

この車両はオートマチックトランスミッション車両であり、アクセルとブレーキ2つのペダルで運転を行う。また、シフトレバーは自動運転時に制御装置によりレバーの位置を変更することが可能である。

アクセルとブレーキそれぞれの上には、フットスイッチが備えられている。緊急時に人間がペダル操作を行うことで、自動的に運転モードが自動から手動に切り替わる仕組みになっている（テイクオーバー機能）。また、ハンドルにもテイクオーバー機能のためのスイッチが備えられている。運転者はハンドル、ペダル、それぞれ独立にテイクオーバーを行うことができる。



図 4-8. ジョイスティック式システムを実装した運転席

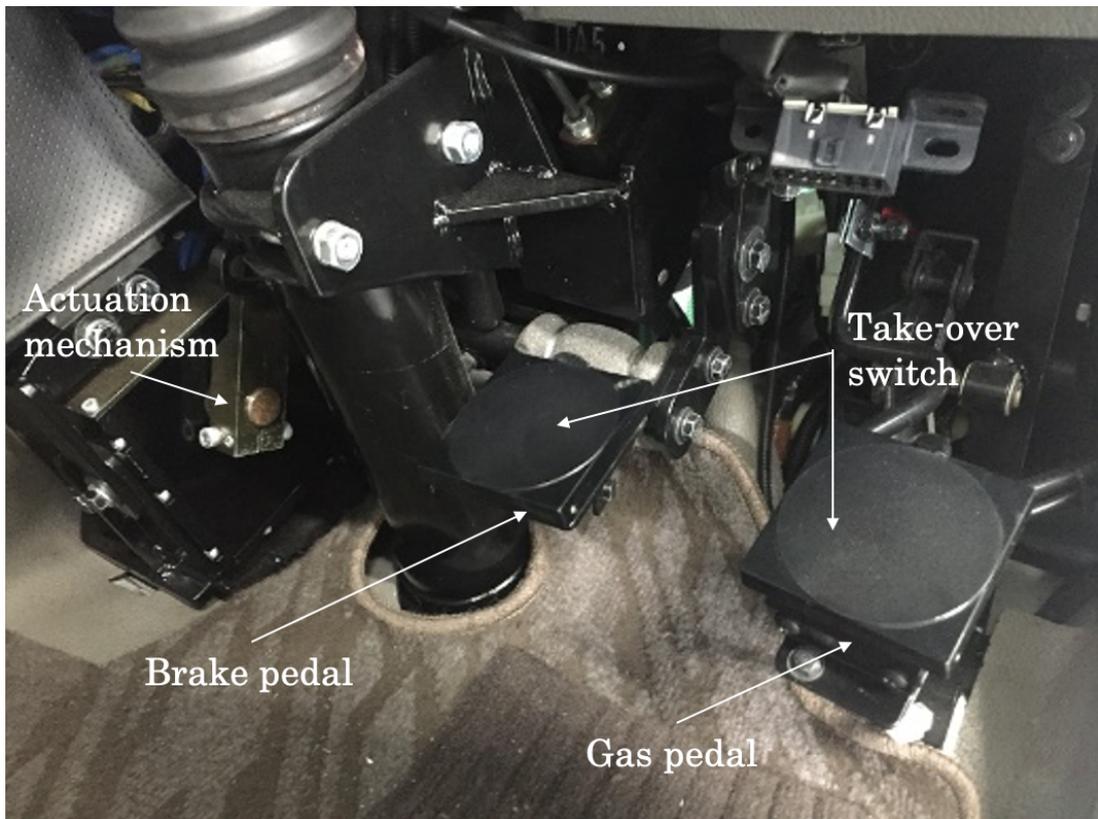


図 4-9. ジョイスティック式システムを実装した運転席

自動運転バスの運転席全体の様子には図 4-10 で表示している。運転席はドラ

イバー専用のモニターがあって、そこで、ドライバーが車両の進行経路を確認することが便利である。インジケーターの指示を見ると、車両の運転モードなど車両状況を確認することができる。また、自動運転復旧ボタンはステアリングとアクセルブレーキそれぞれ独立に分けて、手動モードで走行中の車両にも、止まらずにすぐ自動運転モードに入られる。



図 4-1 0. 自動運転バス運転席全体の様子

一般的なバスを運転すると、車体は普通車より大きく、中型や大型免許を持っている人でも、バスの運転には自信がない方が多い。埼玉工大の自動運転システムはどんな車両でも後付けことができる。そのため、埼玉工大の自動運転システムを付けば、高度な自動運転機能を利用でき、バスなど大型車の運転が簡単になる。

実際の自動運転実証実験中に埼玉工大の自動運転バスが 90 度のカーブを曲がる時に、人間の運転よりうまい。

4.2.4 自動運転実証試験による発見した問題

埼玉工大の自動運転バスは 8 か月で、自動運転レベル 3 の走行距離は 650 キロを超えた。日本各地の実証実験を参加し、走行環境が違い、走行中に様々な問題を発見した。

- 自動運転バス開発セレモニー試走実験
- 埼玉工業大学周辺の公道実証実験
- 坂戸市「坂戸・夏よさこい」の公道実証実験

- 本庄早稲田駅周辺の公道実証実験
- パシフィコ横浜真空展の閉鎖区間実証実験
- 加須市騎西「銀杏祭」の公道実証実験
- 兵庫県科学公園西播磨 Spring8 の公道実証実験
- 愛知県日間賀島の公道実証実験
- さいたまスーパーアリーナの公道実証実験
- 横須賀 YRP スカモビ 2020 の公道実証実験
- 豊田市交通安全センターの閉鎖区間実証実験

上記の実証実験中に発見した代表的な問題は以下となる。

- 埼玉工業大学周辺の公道実証実験：
 - 赤信号の認識が遅くて、停止線を超える問題
 - 逆光の下に信号機の色が認識できない問題
 - 轍や横風のため中速走行する（50km/h）際に蛇行する問題
- 兵庫県西播磨 Spring8 の自動運転実証実験
 - 悪天候に、車載カメラの障害物を誤認識する可能性がある問題
 - 前方車両が走行経路中にある場合、急ブレーキがある問題
 - バンク角が逆の道路で中速走行する（50km/h）際に蛇行する問題
- 愛知県日間賀島の自動運転実証実験
 - 走行コースの中に GNSS の測位誤差は大きいけど NDT の測位誤差が小さい場所と GNSS の測位誤差は小さいけど NDT の測位誤差が大きい場所が混在していた問題

4.2.5 各問題の対策方法

これらの問題を解決するため、それぞれに対策方法を提案した。

信号認識が遅い、停止線で止まらない問題について：

1. 定時点灯の信号について、信号色のサイクル情報の利用手法を Autoware に実装する。更に、信号色のサイクル情報を通信インターフェイスで監視する。
2. 太陽の位置情報を計算し、信号と太陽が同じ方向の場合、テイクオーバーリクエストを出して自動運転モードを切る。
3. 安全に止まれる距離の最小値を割っても信号の識別ができなければテイクオーバーリクエストを出して自動運転モードを切る。同上、停止線までの距離を通信イ

ンターフェイスで監視する。

蛇行の問題について：

1. MPC コントローラーを利用の上、計画経路にあるウェイポイントとの偏移距離や角度をインターフェイスで監視する。

GNSS-RTK と NDT の測位誤差が大きい所の混在する問題について：

1. ローカルライザーの切り替え機構を利用する。更に、指令速度、現在速度、ストローク値など車両情報と GNSS-RTK、NDT の測位誤差をインターフェイスで監視する。

障害物の誤認識、トラッキング精度の問題について：

1. 車載カメラの障害物検出能力を向上するため、高速なヘイズ除去手法を利用する。

2. トラッキング情報を利用する。そこで、Point Pillars や Mobileye を併用し、相対速度を用い、対象物の速度情報を算出する。

以上の対策方法を利用し、事故をならないように機能して、問題を解決した。筆者らは事故を起さる可能性を考え、自動運転ドライブレコーダーを提案した。

ヘイズ除去手法、通信インターフェイスと自動運転ドライブレコーダーは5、6、7章で説明する。その後、論文の最後には信号のサイクル情報、MPC コントローラーとローカルライザーの切り替え機構を説明する。

第二部

本論文で新しい提案

第5章 車載カメラの視認性を高めるヘイズ除去

5.1 従来ヘイズ除去手法の問題所在

ヘイズ除去とは、悪天候においてカメラで撮った写真が霧、煙霧から生じるノイズ画像から、ノイズを除去すると人や風景がはっきり見えるようになる研究分野である。自動運転車の発展と共に、この障害物検出の研究分野が進められている。問題の所在は車載カメラの画像はヘイズの影響を受けることで、コントラストが低下し、不鮮明になり、障害物を正しく認識できなくなることで交通事故に繋がる可能性が高く危険である。そこで車載カメラのヘイズ影響下の障害物検知能力改善にするため、高速なヘイズ除去手法を提案した[32][33][34]。

従来のヘイズ除去の研究について、画像中のヘイズは画像を撮影した景観の3D的な奥行情報に依存し、ヘイズが薄くても奥行情報があれば画像中では濃く映る。この現象を考慮した、異なる天候で撮影された2枚の画像を利用して奥行を求める手法が提案された[100][101][102]。Kopfら[103]は、撮影場所の奥行の3D情報が既知であるものとした手法を開発した、しかし奥行の3D情報が既知であると仮定することは難しい。近年画像1枚からヘイズを除去する手法が数多く発表されている。Heらはヘイズのない屋外画像において、空以外の領域では、各ピクセルの近くにRGBの値のどれかがほぼゼロとなるピクセルがあることに気づき、そのことに基づいてDark Channel Prior(DCP)法を提案した[3]。GibsonはDCP法を改良しメジアンフィルタを利用することで高速化する手法を開発した[104]。Liuらはヘイズ除去の量を自動的に調整できる方法を提案した[105]。なお、DehazeNet[106]、MSCNN[107]、AOD-Net[108]のように深層学習に基づいてヘイズ除去をする手法が提案されているが、最速なAOD-NetであってもGPUなしでは遅い問題がある[109]。

従来の問題点を挙げると、入力情報が異なる天候の2枚画像や異なる撮影角度の2枚画像や3D情報などが必要であり、リアルタイム性が求められる自動運転車に適用することは難しい。また、DCP[3]に基づいて改良された手法を行うとコントラストが低下し、障害物認識率が下がることもあり、従来のヘイズ除去手法はそのままでは利用できない。障害物認識アルゴリズムは霧、煙霧などはあまり考慮されていないため、悪天候での障害物検出率が低下する恐れがある。これらを踏まえ、この論文の目的は自動運転の安全性を高めるため、自動運転に対して適切なヘイズ除去手法を提案

する。筆者らはヘイズが障害物認識率を下がることを一つの要素と考えている。ヘイズ除去を行うことで障害物の検出精度が上げることが可能である。本手法を車載カメラに導入すると、自動運転車の障害物検出能力が上がり、走行の安全性を確保できる。

5.2 ヘイズ除去に関する研究背景

5.2.1 ダークチャンネルに基づくヘイズ除去原理

5.2.1.1 ヘイズ画像の物理モデル

画像のヘイズを記述するため、Narasimhan の大気光モデル[110]が利用される。

$$I(x) = J(x)t(x) + A(1 - t(x)) \quad (5-1)$$

ここで、 $I(x)$ は画素位置 x における観察された RGB 強度、 $J(x)$ はヘイズ無し画像の強度であり、 A は全局大気光、 $t(x)$ は透過図と呼ばれる。全局大気光と透過図が求めれば式(1)を解くことによってヘイズ無し画像を得られる。

5.2.1.2 透過図 $t(x)$ の推定

Heらは、カラー画像 $I(x)$ のダークチャンネルを以下のように定義した[3]。

$$I^{DC}(x) = \min_{y \in \Omega(x)} \left(\min_{c \in \{R,G,B\}} I^c(y) \right) \quad (5-2)$$

ここで、 $I^c(y)$ は座標 y におけるチャンネルのピクセル値を表し、 $\Omega(x)$ は x 周りのマスク領域を表す。ダークチャンネルはヘイズの量を表した図であり、ヘイズのない画像のピクセル値ではほぼゼロである。 $J(x)$ はヘイズのない画像であるので、式(4-3)が得られる。

$$J^{DC}(x) = \min_{y \in \Omega(x)} \left(\min_{c \in \{R,G,B\}} J^c(y) \right) \approx 0 \quad (5-3)$$

Heはこの式(4-3)が成立することを前提条件として、Dark Channel Prior (DCP)と呼んでいる。式(4-1)はDark Channelでも成立する式なので式(4-1)と式(4-3)から式(4-4)が得られる。

$$I^{DC}(x) \approx A(1 - t(x)) \quad (5-4)$$

これを解き透過関数として以下の式を得る。

$$t(x) \approx 1 - \frac{I^{DC}(x)}{A} \quad (5-5)$$

5.2.1.3 全局大気光**A**の推定

He らはヘイズ領域の中で一番明るい領域を全局大気光とした。ダークチャンネル画像の中でピクセル値の上位 0.1% を選択し、この選択したピクセル座標における入力画像のピクセル値の中で最も強度が高い画素 A_0 を大気光としている。

$$A = A_0 \quad (5-6)$$

5.2.1.4 ヘイズ除去のパラメータ ω

He らは、ヘイズ除去の多少をコントロールするため、式(4-7)にパラメータ ω ($0 \leq \omega \leq 1$) を導入した。

$$t(x) \approx 1 - \omega \frac{I^{DC}(x)}{A} \quad (5-7)$$

このパラメータ ω は走行環境によって数値が違い、手動で決めることが必要になる。

5.2.1.5 復元画像

式(4-6) (4-7) を式(4-1) に代入し、復元画像 $J(x)$ は以下のように表す。

$$J(x) = \frac{I(x) - A_0}{t(x)} + A_0 \quad (5-8)$$

ゼロ除算を避けるために、He らは小さな正数 t_0 を導入した。

$$J(x) = \frac{I(x) - A_0}{\max(t(x), t_0)} + A_0 \quad (5-9)$$

提案した手法では He らと同じ $t_0 = 0.1$ とした。

5.2.1.6 復元画像の後処理

復元画像内のエッジ付近の歪みを改善するために、He らは、透過図にソフトマッチング (Soft matching) とバイラテラルフィルタ (Bilateral filter) を適用した。この処理は、復元画像を改善できるが、非常に時間がかかる。

5.2.2 ダークチャンネル手法の拡張

5.2.2.1 Median Dark Channel Prior

He ら [3] によって開発されたダークチャンネル手法の計算速度を高めるために、Gibson ら [104] は、ソフトマッチング処理を省略し、最小値フィルタ (式(4-2)に示すように) の代わりにメジアンフィルタ (Median filter) を利用した。Gibson らは、ヘイズのない画像のメジアンダークチャンネル (Median Dark Channel) のピクセル値がほぼゼロに近いという前提条件 Median Dark Channel Prior (MDCP) を置くこととした。

$$J^{MDC} = \underset{y \in \Omega(x)}{\text{med}} \left(\underset{c \in \{R, G, B\}}{\min} J^c(y) \right) \approx 0 \quad (5-10)$$

透過図 $t(x)$ は：

$$t(x) \approx 1 - \frac{J^{MDC}(x)}{A} \quad (5-11)$$

メジアンフィルタを使用し、ソフトマッチング処理を省略することにより高速な動作速度を有するという二つの利点がある。

5.2.2.2 パラメータ ω の推定

Liu ら [105] は、式(4-12)においてパラメータ ω を自動的に調整する方法を提案した。

$$\omega = pm \quad (5-12)$$

ここで、 m はすべてのピクセル値の平均であり、 p は実験的に決められた値を利用している。

5.3 提案手法

5.3.1 グレースケール画像

カラー画像は夕日等、照明環境の変化の影響を受けやすいのでグレースケール画像に基づいてアルゴリズムを開発することを目指した。予備実験では、Gibsonの方法はグレースケールの画像でも適度に効果的であることが分かった。そこで、提案手法では、次の式(4-13)で透過図を定めることとした。

$$t(x) = 1 - \omega \frac{\text{med}_{y \in \Omega(x)} Y(y)}{A_0} \quad (5-13)$$

5.3.2 パラメータ ω の推定

ω は次の形で仮定する：

$$\omega = \min(pm(m + 0.5), 0.95) \quad (5-14)$$

m はグレースケール画像最大強度値と最小強度値の平均値であり、これはすべてのピクセルの平均値を計算するよりも時間を節約する。予備実験では p 値が $p \in [1.2, 1.5]$ に属するとき良好な結果が得られたので、この実験ではすべての実験において $p = 1.3$ を使用した。

5.3.3 全局大気光の改良

Heらの手法[3]に基づき、全局大気光の計算方法の方法によるある程度はうまくいくのだが、車のヘッドライト、白い建物、または何らかの白いオブジェクトに対して選択する可能性が高かった。これが復元画像の誤差の原因である。

この問題を解決するため、筆者はヘイズ領域に限定された上、全局大気光を計算することを提案した。

マスクサイズ(Mask Size)を 7×7 に設定し、図5-1-1の画像のダークチャンネル画像を求め、ダークチャンネル画像の中の上位10%の輝度で2値化した画像を図5-1-2に示す。右側の画像には細かなノイズが残し、そこでモルフォロジー(Morphology)変換を利用してノイズを減らすこととした(図5-1-3)。予備実験によってモルフォロジー演算のカーネルサイズ(Kernel Size)は 15×15 とした。

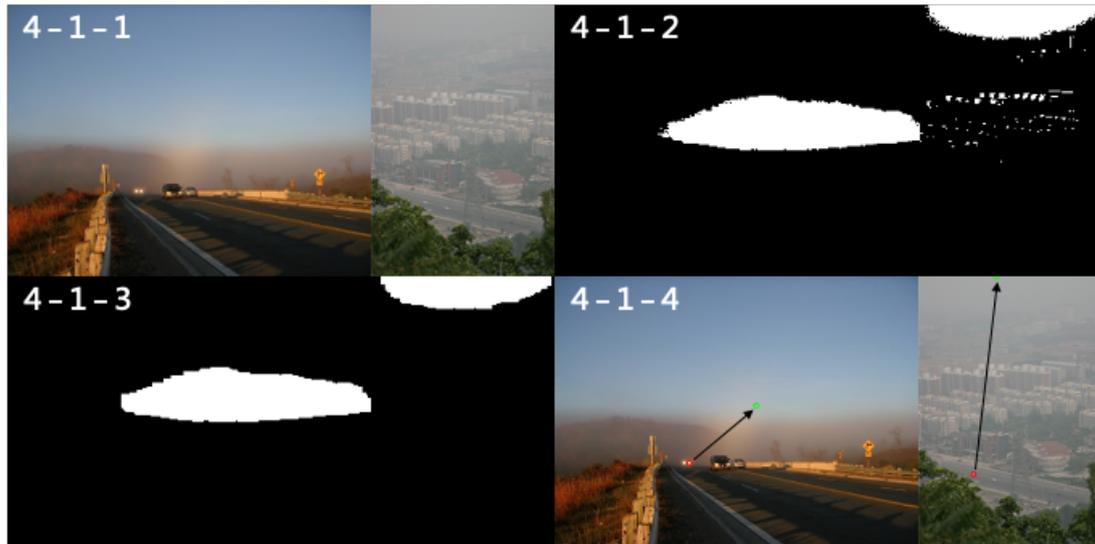


図 5-1. (5-1-1)元画像 (5-1-2)ヘイズ領域 (5-1-3)モルフォロジー変換後のヘイズ領域 (5-1-4)全局大気光の代表ピクセル(提案手法)

図 5-1-4 を参照し、赤い円で囲まれたピクセルは以前の手法で選択したものである。左の画像において車のヘッドライトが囲まれている。右の画像において白い車が選択されている。緑色の円は提案手法であり、ヘイズ領域の一番明るい場所が選択されていて、大気光が正しく推定されていることがわかる。

5.4 実験の流れ

実験には下記の画像を利用する。

- 大学周辺で霧が出た際に撮影した 30 枚の画像
- [3]や[104]や[105]で利用されているベンチマーク画像
- インターネットで集めた 300 枚のヘイズ画像

上記画像を利用して下記の 4 つの評価実験を行う。

- ヘイズ除去を行った画像の目視による評価
- YOLO による自動車の検出結果のソフトマックス関数 (Softmax function) [111]の確率と intersection over union (IOU) [112][113]の比較
 - ピーク信号対雑音比 (PSNR, Peak signal-to-noise ratio) [114]と構造類似性 (SSIM, Structural Similarity Index Measurement) [115]の比較
 - 処理時間の比較

5.5 実験結果

5.5.1 ヘイズ除去を行った画像の目視による評価

ベンチマーク画像でヘイズ除去を行った例を図5-2に示す。



図5-2. ベンチマーク画像の実験結果

提案手法はヘイズが除去できていることが確認できて、画像の中に遠方の車ははっきり見えるようになった。

5.5.2 検出精度と IOU の比較

ヘイズ環境下の車載カメラ画像からの障害物検知能力の向上が研究の目的であるので、Softmax 関数の確率と IOU を用いて各手法の性能を評価する。ここで物体検出には YOLO[4]を利用することし、検出閾値を 0.5 とした。プログラム上では "yolov3.cfg" 設定ファイルと "yolov3.weights" モデルファイルを使用した。

我々は最初に各手法でヘイズ除去作業を行い、次に自動車の検出精度を評価した。Softmax 確率は高いほうが良い。検出結果の一例を図5-3に示す。

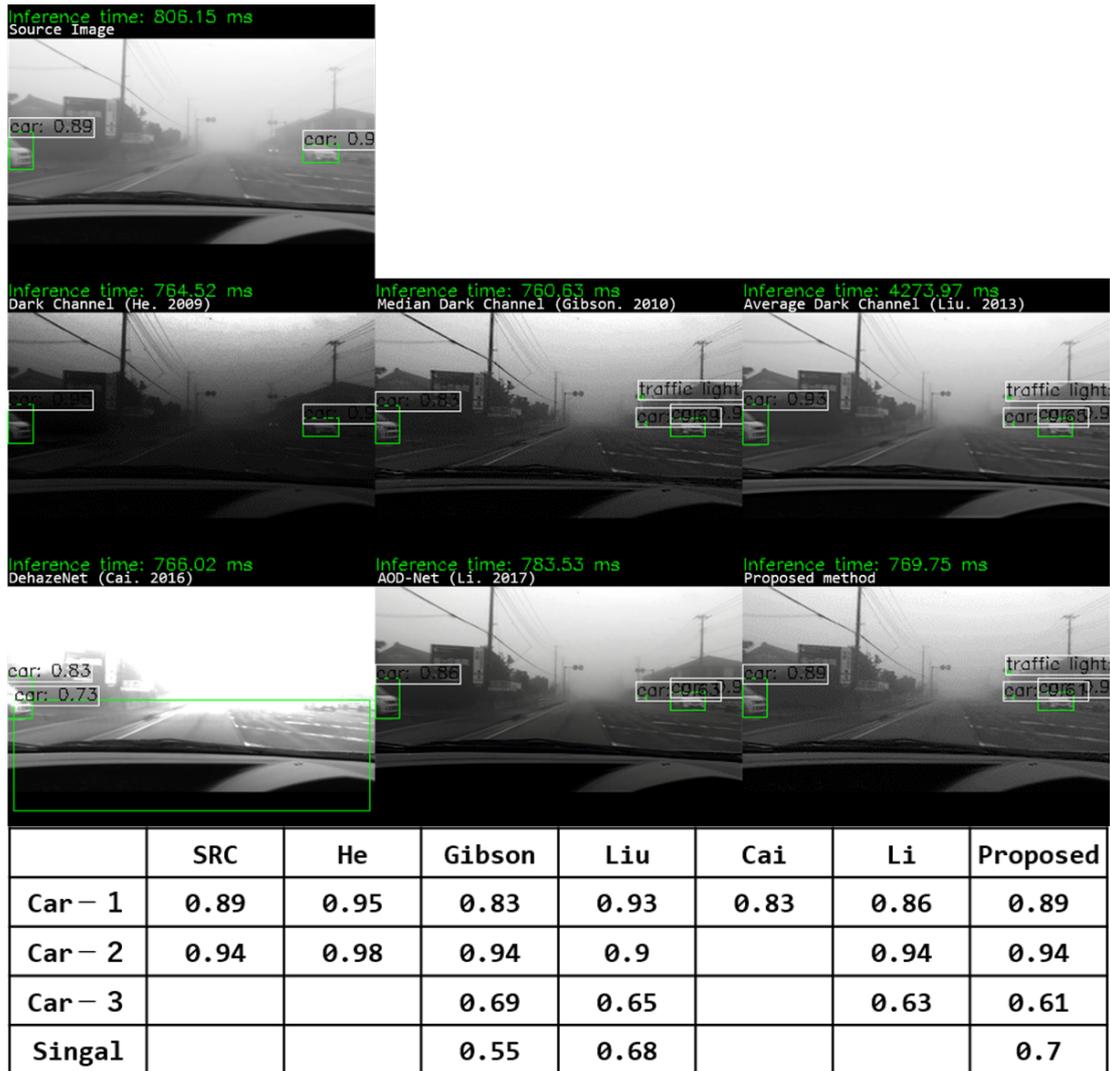


図 5-3. YOLO に基づく物体の Softmax 関数の確率(検出率)

図 5-3 によって、提案手法は Gibson や Liu の方法に匹敵することが分かる。30 枚のベンチマーク画像を利用して IOU を計算した。Softmax 関数の確率の閾値を 0.5 から 0.9 に設定した際に検出された車両の数の変化を表 5-1 に示し、IOU の閾値を 0.1 から 0.9 に変化させた際に検出された車両数の変化を表 5-2 に示した。

表 5-1. Softmax 関数の確率の閾値(Confidence)変化に伴う検出した車両数の変化

Detected	Source	He	Gibson	Liu	Cai	Li	Proposed
conf>0.5	60	48	63	65	37	63	63
conf>0.6	51	45	56	58	34	60	56
conf>0.7	42	38	48	46	27	42	48
conf>0.8	34	34	35	34	16	35	32
conf>0.9	23	28	21	24	10	26	25

表 5-2. IOU の閾値変化に伴う検出した車両数の変化

Detected Cars	Source	He	Gibson	Liu	Cai	Li	Proposed
IOU>0.0	60	48	63	65	37	63	63
IOU>0.1	60	48	62	65	37	63	63
IOU>0.2	60	48	62	65	37	63	63
IOU>0.3	60	48	62	65	37	63	63
IOU>0.4	59	48	61	64	37	61	61
IOU>0.5	59	48	60	62	36	61	61
IOU>0.6	57	47	54	59	35	58	57
IOU>0.7	52	44	46	52	31	52	50
IOU>0.8	33	30	34	37	20	36	35
IOU>0.9	9	7	9	10	9	8	10

表 5-3 に Softmax 関数の確率と IOU の閾値を 0.7 に設定したときの結果を示す。提案手法は 41 台車両が検出され、最良の結果が得られたことが確認できた。

表 5-3. 信頼係数と IOU の閾値による、検出された車両の数

	Source	He	Gibson	Liu	Cai	Li	Proposed
Detected	38	36	39	40	23	37	41

5.5.3 PSNR と SSIM の比較

表 5-4 は PSNR と SSIM の 30 枚画像における平均値を示している。PSNR と SSIM は値が高いほど性能が良い。提案手法の性能が一番高いことがわかる。

表 5-4. PSNR と SSIM の平均値

	He	Gibson	Liu	Cai	Li	Proposed
PSNR	10.55	14.81	18.84	13.18	16.14	19.46
SSIM	0.61	0.79	0.9	0.56	0.83	0.92

5.5.4 処理時間の調査

インターネットでダウンロードした 300 枚のヘイズ画像を用いて実行時間を調

査した。プログラムは OpenCV4.0.0-pre を利用し、VS2015 で作成した。プログラムの処理時間とピクセル数は線形関係が存在していることが分かった。実時間処理を目標とすれば 24FPS 以上が必要、画像 1 枚の処理時間は 0.041 秒以下で処理する必要があるが、この時処理できるピクセル数は約 45 万 (600×800) であった。

5.6 本研究に残した問題

以上の実験による、次の結論を引き出すことができる。

グレースケール画像だけで計算コストの低いことが利点とし、既存のヘイズ除去手法と比肩できるヘイズ除去手法が開発した。

開発した手法が YOLO による障害物検知の性能向上に役立つことを明らかにした。車載カメラに適用されると、自動運転車の走行安全性を高めることが確認できた。

ヘイズのないグレースケール画像で式(10)は必ずしも成立しない。しかし、この前提条件で求めた透過図を用いてヘイズを除去できることが分かったわけだが理由が必ずしも明確ではない。効果の有無を数多くの画像で検討し、この理由を分析するのは今後の課題とする。

また、第 3 章が紹介した遠赤外線カメラに基づいたヘイズ除去は単眼 RGB カメラより良い結果を取得した(図 5-4)。左側は普通の RGB カメラ画像、右側は遠赤外線カメラ画像、遠赤外線カメラ画像からヘイズが除去した RGB カメラ画像に復元することは一つの挑戦と考える。



図 5-4. ヘイズの下に遠赤外線カメラの映像

第6章 自動運転実験車両用のドライブレコーダー

6.1 自動運転実験車両に必要性

自動運転車を安全走行するため、様々な技術やセンサーを研究されている。事故を発生しないように車両の走行状態を監視したり、フェールセーフ機能を追加したり、伴走車両から走行環境を確保する方法などがある。筆者は自動運転実験車両に対して、従来の自動運転事故を分析しながら、自動運転実験車両ドライブレコーダーを提案した。実証実験中に事故が起きても、きちんと事故調査や分析ができるように工夫した。事故が発生した際に、事故の原因が明らかにならず公衆の疑問が解消できない場合には、自動運転反対の声が大きくなってしまい、自動運転の実用化は完全に否定されかねない。そのため、自動運転実験車両用のドライブレコーダーは事故調査の一つ手段として、とても重要な課題になる。

6.1.1 従来の自動運転事故と事故対応

前節 2.2.4 で述べた Uber は 2018 年 3 月 20 日、アリゾナ州で自動運転事故が発生した。当時の車両は車内と車外のカメラ映像の記録装置を搭載していた(図 6-1)。しかし、単に車内と車外カメラ映像を頼り、事故原因を調査するために一般の乗用車は十分だが、自動運転車は難しいことである。具体的に人為的なミスか、コンピューターのミスかがわからない。



図 6-1. Uber の車内と車外のカメラ映像 (引用元[116])

前節 2.4.3 で述べた名古屋大学は 2019 年低速自動運転中に事故が発生した。名古屋大学は低速自動運転車両事故報告書を公表した。しかし、報告書には事故原因はログファイルにより自車位置推定失敗との推測が考えていた、他の技術の欠陥は確認できなかった。更に、コンピュータの計算負荷により遅延が生じる可能性があることを考えているが、ただし、証拠は出せない。

自動運転実証実験ではヒヤリハットや事故を完全に避けることは難しいが、同じ原因のヒヤリハットや事故に遭うことが2度とないようにするため、ヒヤリハットや事故の原因を正確に分析することが必要である。

6.1.2 従来の実証実験中データ記録の問題所在

我々の乗用車（プリウス）とマイクロバス（リエッセ）形態の自動運転車両は、共に自動運転用のオープンソース Autoware を独自に改造して利用しており、4年間の自動運転実証実験の経験に基づいて、現時点で市販されているドライブレコーダーを進化させ、自動運転専用のドライブレコーダーを提案したいと考える。

日本の場合、従来の実証実験では Rosbag (ROS 標準の車速、自己位置、映像情報などのメッセージデータを保存するためのログファイル) で記録しながら、オペレーターは Rviz 画面を監視して、実験を行う。ただし、Linux システムには実時間処理が保証されてない、タスクの実行時間、実行順位が固定されなくて、記録した情報が失われる可能性があり、これらのログファイルのみに依存し、事故の再現性はない。万が一事故が発生した場合、Rosbag だけを頼りに全てが分析できるとは限らないので事故対応は期待できない。我々は色々な情報を調べてみたが、現時点では自動運転専用のドライブレコーダー情報が全くないことが分かった。そこで、従来ドライブレコーダーを踏まえて、自動運転レベル 2、3 での走行時には、自動運転実験車両用のドライブレコーダーは車両前方の映像だけではなく、車内でのオペレーターの操作画面とドライバーの動作を記録できる機能を持たせることを提案する。またレベル 4 での走行時には、別の記録（遠隔操作室の運転手の動作、音声など）をとっておく必要がある。事故があったとき、オペレーターやドライバーによる何らかの誤操作の有無を確認できることが非常に重要である。同時に、車速やステアリング、センサー類の誤差情報も映像として記録する。

6.2 ドライブレコーダーの研究背景

6.2.1 従来ドライブレコーダーの研究

ドライブレコーダーは主に自動車事故が発生した時の状況を記録する装置である(図6-2)。一般に、映像記録型ドライブレコーダーの研究では、ニアミスの解析と予防などの研究が行われ[117]、事故を回避する為に、車の死角における人の飛び

出しなどといった危険予測の研究が行われている[118]。より優れた機能をドライブレコーダーに導入すると、自動運転車の安全性を向上できる。とはいうものの、近年の日本各地での自動運転実証実験中に起こった多く事故の原因は、プログラムによる自己位置推定が失敗したことによるものであり、人や車などとのニアミスで発生した事故は少ない。事故の発生にあたっては、従来の乗用車の場合と異なり、自動運転車内部のデータを記録しておくことが重要になる。



図6-2. 市販されているドライブレコーダー（引用元[119][120]）

[121][122]では、コンピューター化された車における、車速、ドライバーの視線または白線距離などの車載データ記録装置の設置が提案された。[123]では自動運転車の事件データ記録装置の設置も提案した。ただし、それらはまだ想定上のシステムであり、主に車両間の情報共有のために設計されている段階である。Chen は 360 度カメラ付き事件データ記録装置を用いた盗難防止警報システムを提案した[124]。

6.2.2 Autonomous-Car Drive Recorder

埼玉工業大学の自動運転バスは Autonomous-Car Drive Recorder (ACDR) を搭載している。自動運転実験車両用のドライブレコーダーは全ての重要な車載データを記録し、車両の鳥瞰図、ドライバー席の映像（ハンドルとペダル）も同時に記録する。

更に、自動運転実験車両において、ドライブレコーダーに電源供給の確保も重要

な課題である。自動運転車の電気設備は普通の乗用車よりも数倍多く、熱を抑える為に独自の冷却装置を付けることと各センサーのシグナル線分け、ノイズを避けることも前提として、全ての電源はヒューズボックスから流れる。自動運転用のドライブレコーダーは車両の常時電源と繋ぐことで、重大な事故が発生しても、常時電源が入ったままの状態になるので、ドライブレコーダーは正常に動作する。常時電源はサブバッテリーから給電するので、エンジンをかけなくても、電源は半日以上保つことができる。

6.3 ACDR の構成

6.3.1 ACDR の全体システム図

自動運転実験車両用のドライブレコーダー (Autonomous Car Driver Recorder) [36]の構成は図 6-3 のように 3 画面で表示される。オペレーターの操作画面、ドライバー席の映像と車両の鳥瞰図を録画し、最後にモニターへ出力する。

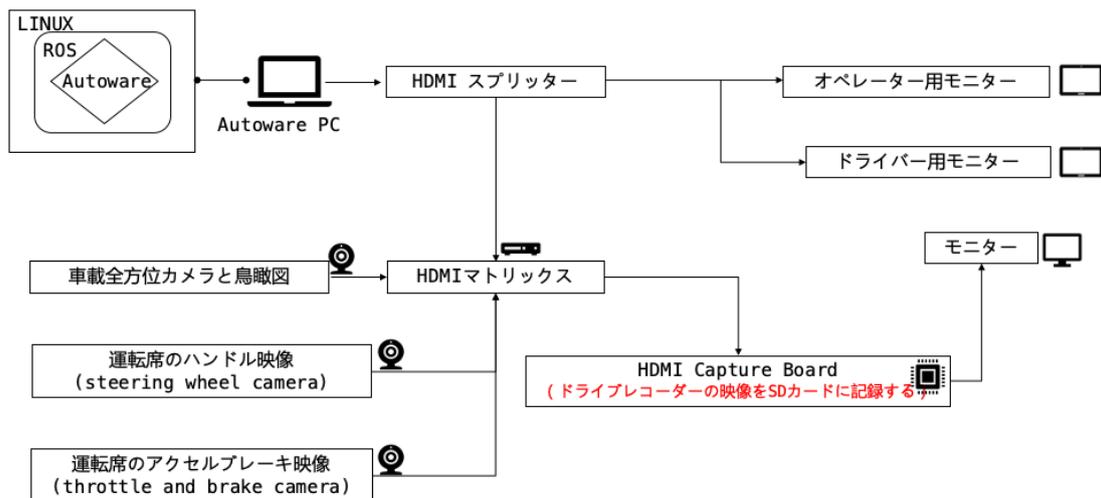


図 6-3. ACDR の全体システム図

6.3.2 ACDR の録画面面

ACDR の録画面面は図 6-4 で示す。左側はオペレーターの操作画面、Rviz (ROS visualization) を用いたカメラ映像、信号映像と三次元地図及びライダー情報などを表示し、横にあるユーザーインターフェイスは車両の運転モードと車速、ステアリング、各機器の健康状態などを表示している。右側は全方位カメラと運転席のハンドルとアクセルブレーキ映像を表示している。

全方位カメラと鳥瞰図の映像を利用すると、は事故が起きた時、当時の走行環境は目視で判断することは可能である。運転席の映像はドライバーの誤操作を確認でき、また事故が発生する前に何かを対応しているかを記録する。オペレーターの操作画面を記録すると、ユーザーインターフェースによる各機器の健康状態や車両の内部データを映像の形で記録することが可能となり、また事故に関わる誤操作の有無を確認することができる。

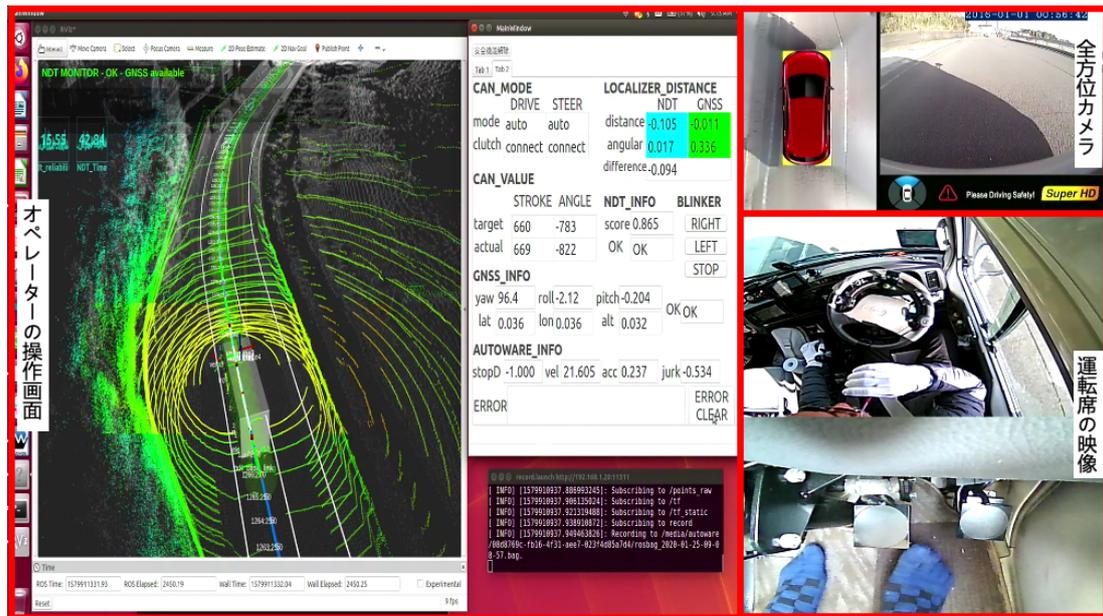


図 6-4. ACDR の録画面面

6.3.3 オペレーターの操作画面

オペレーターの操作画面は大きく分けて 2 つとする。一つは Rviz 画面で、主に車両周囲のセンサー情報を表示し、障害物と交通信号の認識を確認するためのものである (図 6-5)。図の中に、白い点は事前に読み込んだ点群地図、緑の円は LIDAR の線であり、青い線は計画経路、計画経路の上に、それぞれのウェイポイントが目標速度とウェイポイント番号を表示している。もう一つは自動運転車のユーザーインターフェース画面、車両の CAN 情報を表示し、自動運転モードがマニュアルモードかオートモードか、センサー類の誤差は安全範囲内にあるかどうかを確認するためのものである。そして、ターミナルで必要な車両内部データを表示しながら、ログファイルに保存する。

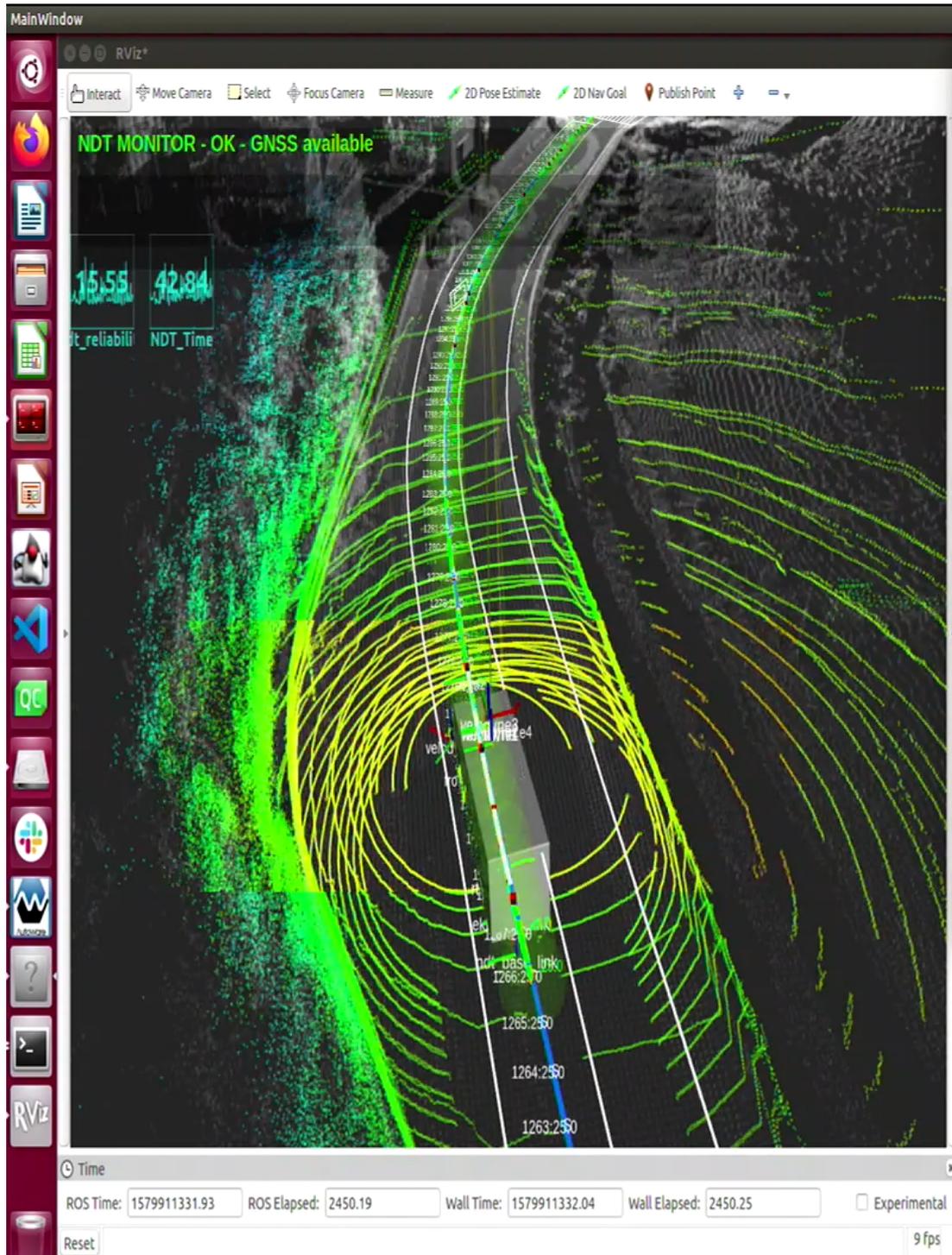


図6-5. オペレーターの操作画面にある Rviz 画面

自動運転に関する研究は長年間に行った、大きく分けて感知、フュージョン、ディスプレイジョン、コントロール四つの部分がある。障害物検出、自己位置推定、経路追従や車速とハンドルのコントロール、それぞれが重要な役割を果たす。以上の課題を解決するためには、車両との安定な通信を確立することは前提であり、自動運転車との

通信も重要な課題である。自動運転車開発の初期段階では、Sadigh ら[125]は自動車と人間の相互作用の問題に対する解決策を提案した。解決策はドライバーに十分かつ重要な車両情報を表示する。Debernard ら[126]は様々な自動運転レベルに、ドライバーまたはオペレーターが正確な状況認識を確立の上、車両運転を関与できるようにするため、適切なインターフェイスが必要である。筆者らはこれらの研究に基づいて、CANバスまたはセンサーから取得されたデータを単に表示されるだけでなく、オペレーターに向けに設計されたインターフェイスが自動運転の試走を簡単かつ効率的になる。

筆者らは改造した乗用車プリウス（5人乗り）とマイクロバス（24人乗り）に基づいて、自動運転の研究を行っている。従来のユーザーインターフェイスの研究に基づいて、改良した（図6-6）。提案したユーザーインターフェイスは走行中の車両に様々な内部データ、車両の走行モードを監視することが可能である。

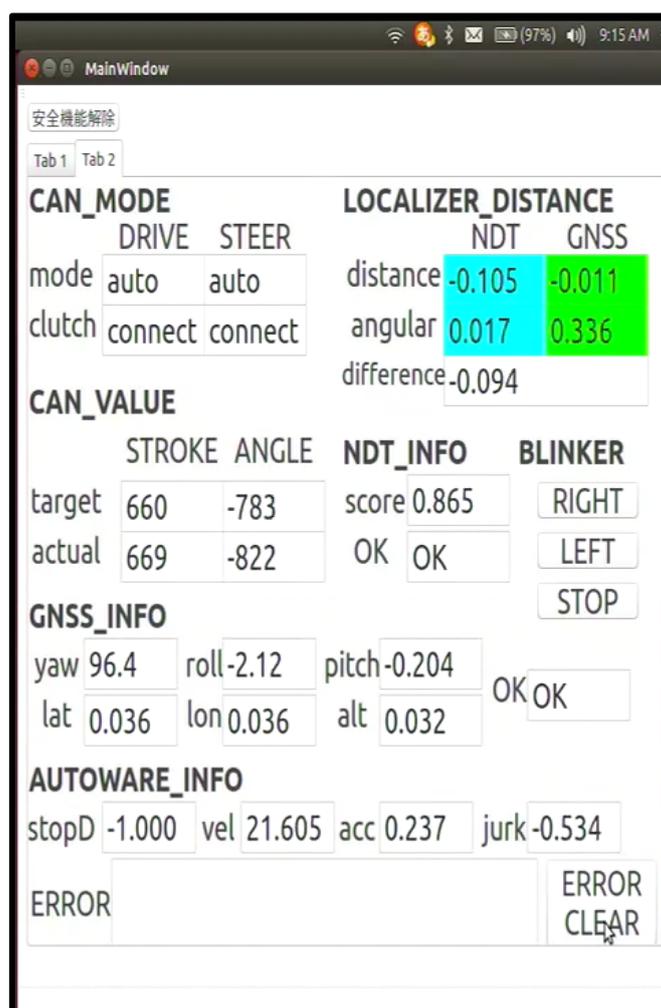


図6-6. オペレーターの操作画面にあるユーザーインターフェイス

6.3.3.1 車両のCANデータ

自動運転車両はKvaser[127]を利用し、Autoware PCと車両のマイコンボードと繋ぎ、通信を行う。CANデータに基づいて、車両実際の速度やステアリング、ウインカーなど情報を取得できて、Autoware PCに流れる。リアルタイムで車両の走行状態を監視するため、これらのCAN情報などを表示することが必要である。KvaserはLinux専用のドライバーが提供されていて、またROSをサポートすることが利点であり、次の図6-7で表示している。

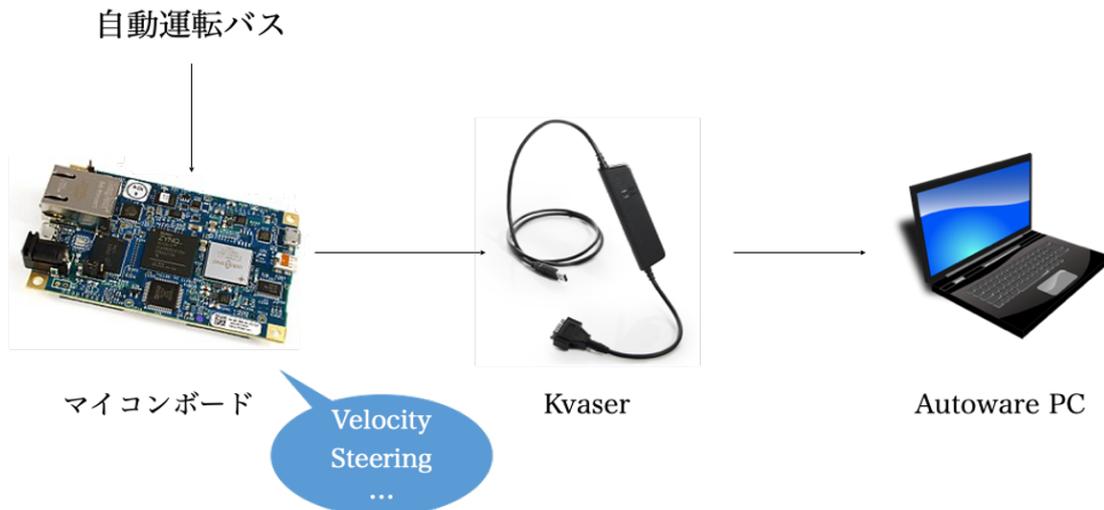


図6-7. 車両のCANデータを取得するため通信の流れ図

6.3.3.2 車両の運転モード

自動運転車は自動運転モードと手動運転モードがあり、また自動モードに対して、アクセルブレーキとステアリングを独立に設定することができる。一方、運転モードはいつでも相互的にテークオーバーすることができる(図6-8)。オペレーターとドライバーは、走行中の車両に運転モードをリアルタイムで認識するため、これらの情報を表示することが必要である。

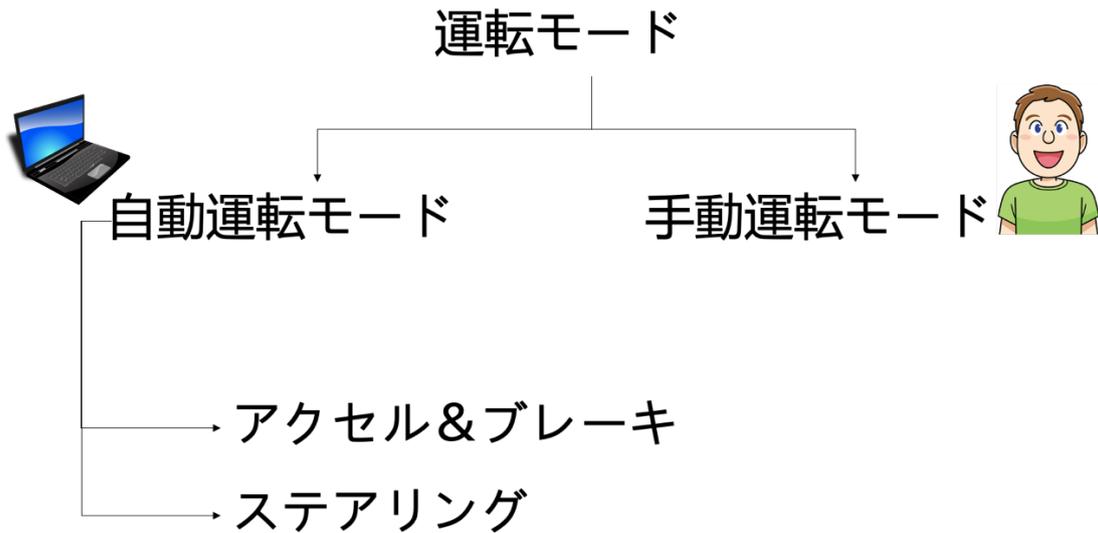


図 6-8. 自動運転車の運転モード

6.3.3.3 予定経路からのオフセット距離

自動運転はスムーズに予定経路を追跡できる必要がある。しかし、走行中にセンサーの振動が発生し、測量精度が低下になる。またデータ量が多くて、コンピュータの処理による遅延が発生した可能性がある。そのため、車両中心が予定経路に沿って走れることが難問である。安全走行を確保するため、車両中心から最も近いウェイポイントまでのユークリッド距離を計算することが必要であり、計算されたユークリッド距離が一定の閾値より大きい場合、自動運転モードから手動運転モードに切り替える。将来、計算されたユークリッド距離をオフセットとして使用しハンドルの角度を修正つもりである。

ユークリッド距離の計算は式(6-1)で示す。

$$distance = \sqrt{(Vehicle.x - Waypoint.x)^2 + (Vehicle.y - Waypoint.y)^2} \quad (6-1)$$

6.3.3.4 自車位置の測位誤差

自動運転車は常に様々なセンサーが装備されていて、車両の振動や悪天候など多くの要因によって影響を受けることがよくある。自己位置を推定するには GNSS-RTK または LIDAR を使用する。

GNSS-RTK による自己位置を推定する時、GNSS の標準偏差が自己位置推定の精度が決められる。標準偏差が極大の場合には、推定された自己位置は数十センチずれる可能性があつて、自動運転モードから手動運転モードに切り替えることが必要である。標準偏差は式(6-2)で示す。

$$\text{Localization error} = \begin{cases} \text{standard deviation of latitude} \\ \text{standard deviation of longitude} \\ \text{standard deviation of altitude} \end{cases} \quad (6-2)$$

LIDAR による自己位置を推定する時、NDT マッチング手法による自己位置を算出する。自己位置推定の精度は NDT スコアによる判断する。NDT スコアが一定の閾値を超える場合、LIDAR 情報と地図の点群情報を合わせることができなく、自己位置情報が失って、危険のため、自動運転モードから手動運転モードに切り替えることが必要である。NDT スコア[67]は式(5-17)で示す。

$$\text{NDT score}(p) = \sum_i \exp\left(\frac{-(x'_i - q_i)^t \Sigma_i^{-1}(x'_i - q_i)}{2}\right) \quad (6-3)$$

ここで、 $p = (t_x, t_y, \theta)^T$ は入力点群の座標系から見た参照点群の座標系の相対座標、 $T(p)$ は入力点群座標系から参照点群座標系への変換行列、 x_i は入力点群中の点 i の座標、 x'_i は入力点群を参照点群の座標系に変換した後の入力点群中の点 i の座標、 Σ_i, q_i は変換後の入力点群 x'_i の共分散と平均座標である。

すべての自己位置情報はリアルタイムで表示し、推定された精度が十分に高い場合は背景が水色で表示され、精度が低すぎる場合は背景が赤色で表示される。

上記の情報以外に、車両の yaw、roll、pitch また一時停止線までの停止距離、ウェイポイントの番号なども表示している。更に、信号のサイクル情報により算出された定時点灯信号の色情報、ローカライザーの切り替え機構により各ローカライザーの動作状況、フェールセーフ機能が動作する際に、エラー情報などを監視することが必要である。これらの安全確保機能は第 7 章で説明する。

6.4 ACDR の有用性

6.4.1 ACDR の応用例

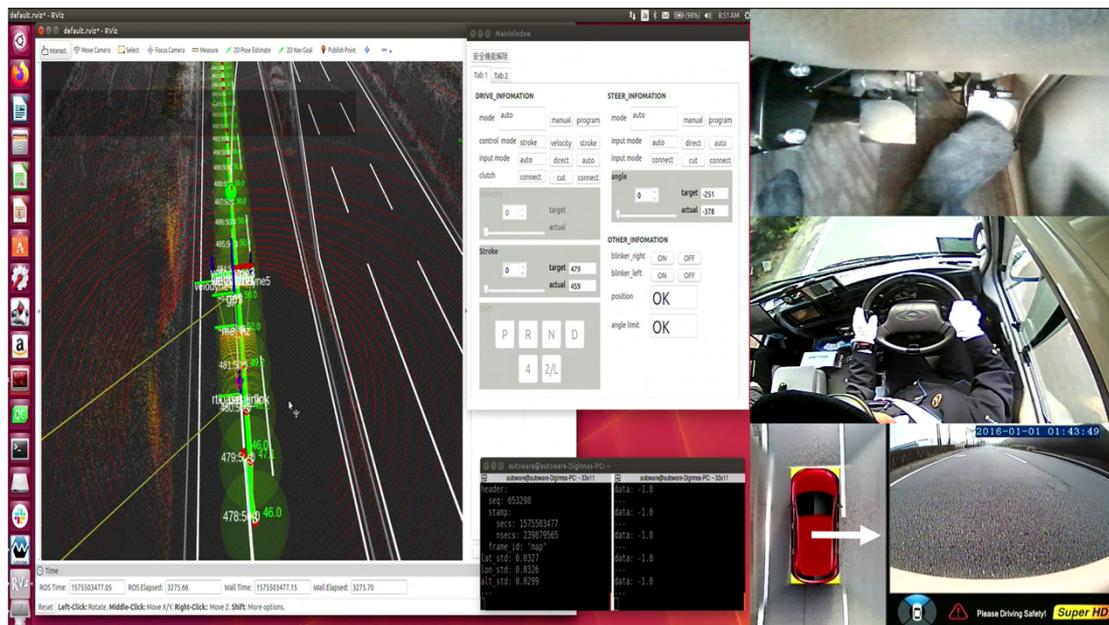


図 6-10. ドライブレコーダーの応用例

西播磨で自動運転実証実験は GNSS による位置を推定し、車両は左車線で走行していた (図 6-10)。GNSS の測位誤差が小さいだが、経路追従の誘導誤差がすごく大きいので、車両が完全に右車線に入った。しかし、単純に Rviz を見て、全ての情報は正常の状態、危険とは思えない。事故が起きた場合、ACDR を付けないと、事故の原因がわからない。また車両が中速走行中、蛇行が発生する際に、ドライブレコーダーの鳥瞰図から見ると、わかりやすい。

6.4.2 ACDR の利点

ACDR は市販のドライブレコーダーの機能を保つ上、オペレーターの操作画面とドライバー席の映像と車両の鳥瞰図を記録することで、自動運転実験車両用のドライブレコーダーが完成する。

ACDR の利点としては以下の 5 つがあげられる：

- 多角度から同時に車外と車内の状況を監視又は録画する。
- 随時録画されて、すぐに走行した様子を再生できる。
- 車両周囲の死角が縮小されるため、ある程度の安全性が確保される。
- Autoware PC と分離されている、CPU の稼働率などへの影響はない。

- 周囲のカメラ画像と車速、ステアリングやセンサー情報なども含めて映像で記録する。

6.5 本研究に残した問題

我々は自動運転実証実験中の事故対応ができるようにするため、自動運転用のドライブレコーダーを提案した。ACDR は Rosbag の記録にみられる Linux システムのリアルタイム処理が保証されていないという欠点を補っており、事故が発生した場合、オペレーター又はドライバーによる何らかの誤操作の有無を記録できて、車両周囲のカメラ映像と車速情報などを含めた事故の再現性が期待できる。

本研究に提案した自動運転実験車両用のドライブレコーダーを構成するために、HDMI 信号は何度も入出力されていて、HDMI 機械の通信不良や熱暴走が発生したことがある。今後は、複数の HDMI 機械を一つに統合の上、車両走行による振動に強いことを考える必要になる。

第7章 その他の安全確保機能

7.1 蛇行を抑えるモデル予測操舵制御

Autoware の経路追従の中に、pure_pursuit と mpc_follower と実装されている。以前の自動運転実験車両ロボカーは pure_pursuit を利用し、自動走行を行う。埼玉工大の自動運転バスは pure_pursuit を利用すると、中速走行の時、カントが逆のカーブや横風で、指定経路から大きく流れるのを避けるために経路追従性を上げるよう近くを見て走る設定にすると蛇行することがある。pure_pursuit は経路追従性を上げようとして近くを見るようにすると蛇行する。蛇行を抑えるように遠くを見るようにすると経路追従性が悪くなる。経路追従性を蛇行のバランスを調整すると、非常に難しい問題である。この問題を解決するために、モデル予測操舵制御を利用した。

7.1.1 モデル予測操舵モデル

モデル予測操舵制御 (Model Predictive Control) は前節 3.2.7 で述べた。ここで、ステアリングの制御について紹介する (図 7-1)。最初は Autoware から経路の CSV ファイルを読み込んだ後、CSV ファイルにあるそれぞれのウェイポイントの座標を三次元多項式にフィットする (図中の Reference Trajectory)。次は時刻 k の車両位置に基づいて、生成した三次元多項式を利用し、 $k+p$ 時刻までの車両ハンドル状態を予測する (図中の Predicted Output)。最後は時刻 k の車両ハンドル状態と予測した車両のハンドル状態の差に応じて、ステアリングを調整する (図中の Predicted Control Input)。以上の処理を時間的に繰り返して行い、モデル予測制御となる。

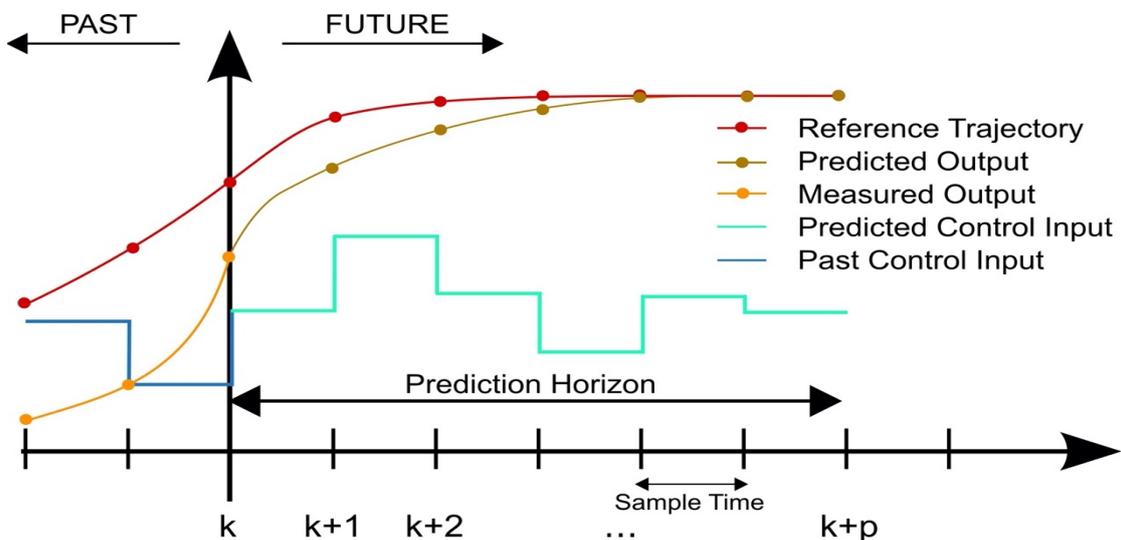
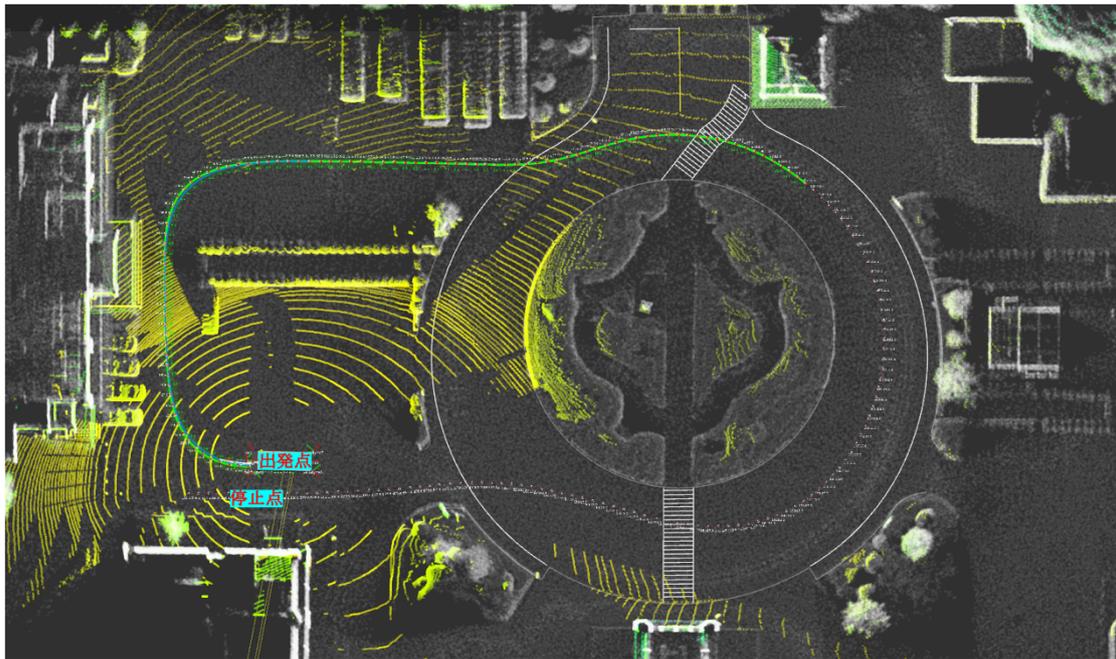


図 7-1. MPC コントローラーの動作原理(引用元[96])

7.1.2 MPCによる経路追従性の調査

モデル予測操舵制御の経路追従性を検証するため、自車位置から計画経路までの偏移距離を計測する。本来の MPC は自転車モデルに基づいて、予測区間(prediction horizon)は 70 ステップで設定され、1 ステップは 0.1 秒で処理していた。7 秒先の制御入力パターンの中で予測経路と計画経路の偏差の積算が一番小さくなる制御パターンを求める最適化問題を解く。制御パターンの最初の制御入力だけを利用し、車両を動かす。

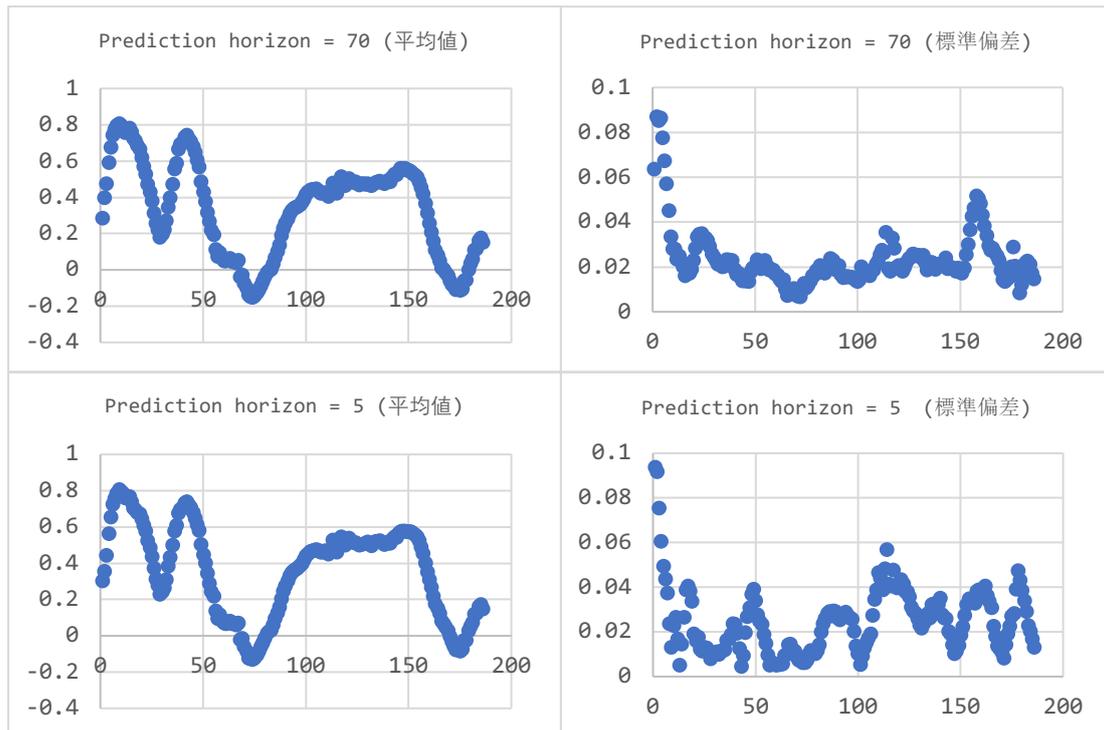
今回は MPC の予測区間 prediction horizon を 70 ステップ(7 秒先の経路予測)と 5 ステップ(0.5 秒先の経路予測)で設定した上、図 7-2 の示すような経路を各 5 回走行した。車両が自動運転中に、5 回ごとの走行実験に対して、それぞれのウェイポイントから自車位置(GNSS-RTK による推定した自己位置)までの距離を記録し、平均値と標準偏差を計算する。



7-2. 埼玉工大の校内における自動走行経路

実験結果は散布図で平均値と標準偏差を示す(図 7-3)。

平均値のグラフを比較して、予測区間が 5 ステップ、70 ステップでも、平均値がほとんど変わらない。標準偏差のグラフを比較して、予測区間 prediction horizon は 5 ステップより、70 ステップの場合は標準偏差が小さいことを確認した。



7-3. 自転車位置から計画経路までの距離の平均値と標準偏差

7.1.3 MPC によるハンドルをぶれる問題の改善

モデル予測制御に基づいたステアリングの制御を行う際に、曲率半径が小さいカーブで走行する時に、ハンドルのぶれる問題はよく発生する。曲率半径が大きいカーブで走行する時、ハンドルのぶれる問題はほぼ発生しない。その問題を解決するために、以下の実験を行った。走行実験は岡部駅のロータリーを利用し、曲率が小さい走行経路を作成した（図7-4に示す）。自動運転を行い、計画経路を追従しながら、それぞれのウェイポイントに対するハンドルの動作角度を記録する。モデル予測制御の Prediction Horizon を 5, 10, 30, 70 に設定して、4回走行実験を行った。

実験結果をそれぞれのウェイポイントに対するステアリングの値の平均値を計算し、図7-5で示す。図の中に、全ての曲線において最初の1番目のウェイポイントに対するハンドルのぶれる原因はハンドルの角度は 0 から経路を追従するように右に回した。Prediction horizon (P.H.) は 5 の時、ハンドルの動作角度が一番滑らかである。P.H. は 10 と 30 の時、ハンドルは 1 回ぶれた。P.H. は 70 の時、ハンドルは 2 回ぶれた。更に、P.H. は 10 の時、車両の自転車位置の最初値がその他の 3 回より右に寄っています。

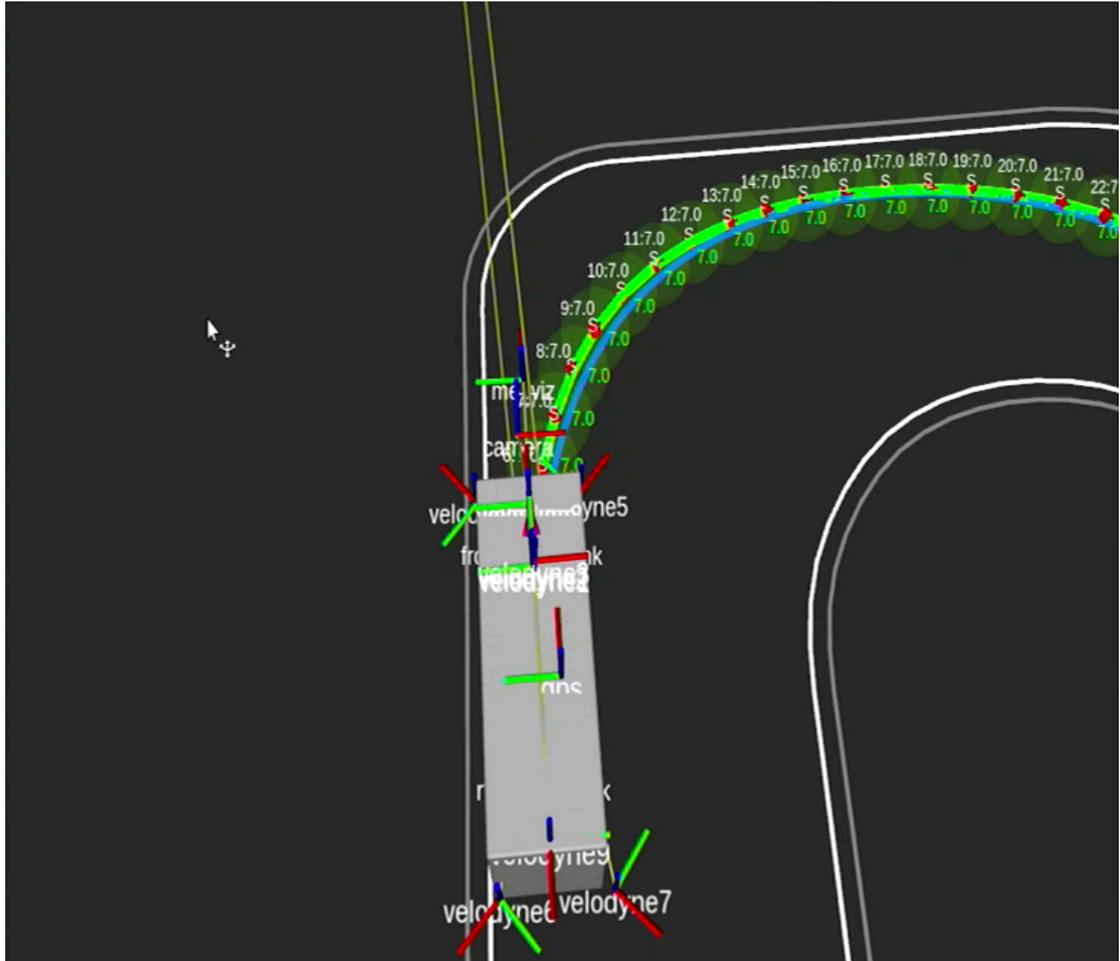


図 7-4. 岡部駅のロータリーで作成した曲率半径が小さい走行経路

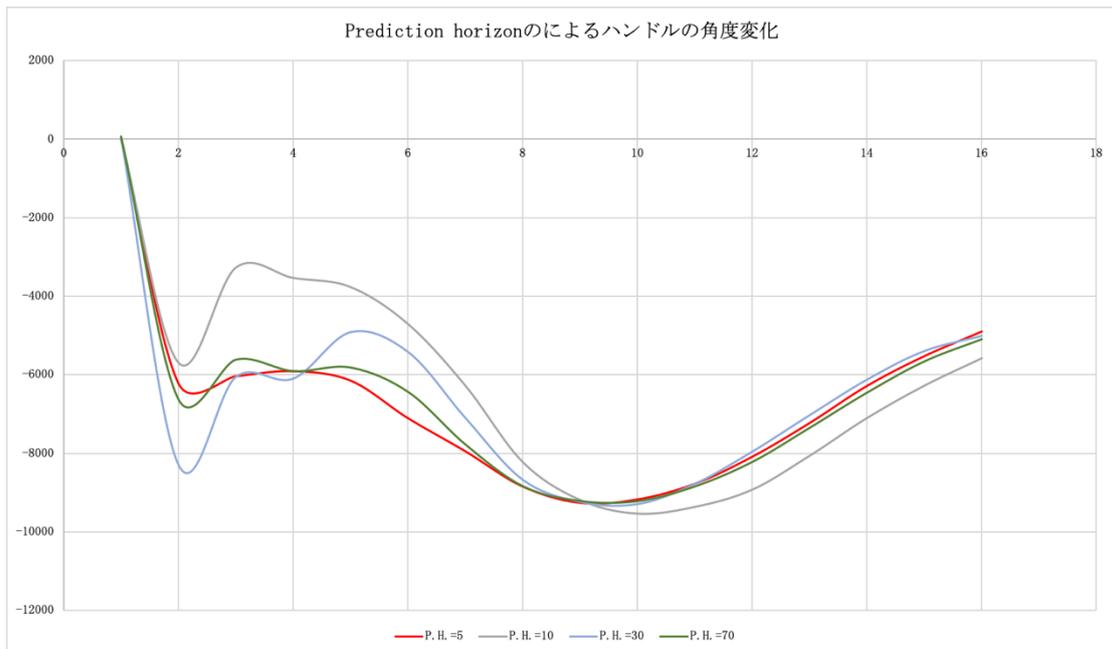


図 7-5. Prediction horizon の変化による実際のハンドルの動作角度

これらの実験データに基づいて、モデル予測制御の改善手法を提案した。車両は曲率半径が小さいカーブを通過する際に、必ず低速で走行する。特にバスの場合、速度を出しすぎると、車体が転倒する可能性が高い。これらの原因を考えて、Prediction horizon の値を走行速度による変化したことを提案した。低速（時速 10 キロ以下）で走行する場合、P.H. を 5 に設定する。一般の速度（時速 40 キロ）で走行する時、経路追従性を保つために、P.H. を 70 に設定する。

7.2 ローカライザ切り替え機構

ローカライザ切り替え機構とは GNSS と NDT による位置推定手法の切り替え機能であり、同時に GNSS と NDT による推定された自車位置と計画経路までの距離や角度を示す。自動運転実証実験の走行コースは多くの場合、GNSS の測位誤差は大きい NDT の測位誤差が小さい場所と GNSS の測位誤差は小さい NDT の測位誤差が大きい場所が混在していた、そこで、様々なローカライザを併用することが必要である。

7.2.1 NDT と GNSS による自己位置推定の比較実験

ローカライザ切り替え機構を検証するために、岡部駅のロータリー付近で自動運転実験を行った。図 7-5 は自動運転実験の計画経路を表示している。開始点から A 点までと C 点から終点までは建物が少ない走行区間であり、A 点から B 点を経由し、C 点までは建物が多いた走行区間である。

NDT と GNSS による位置を推定の上、各 5 回の自動走行を行って、計画経路までの平均距離は次の図 7-6 で示す。

開始点から A 点までと C 点から終点までの走行区間について、建物が少ない、NDT マッチングが失敗し、自己位置推定ができなかった。逆に、建物が少ないので、GNSS の測位信号の反射が少なく、GNSS-RTK を利用して、測位誤差は 3 cm 前後で安定に通過した。

A 点から B 点を経由して、C 点までの走行区間について、建物が多くて、NDT マッチングが安定な状態で、走行できた。更に、NDT で走行した時、計画経路までのずれた距離は GNSS による小さくて、走行安全性は GNSS による高いことを確認した。



図 7-5. 岡部駅のロータリー前の自動走行経路

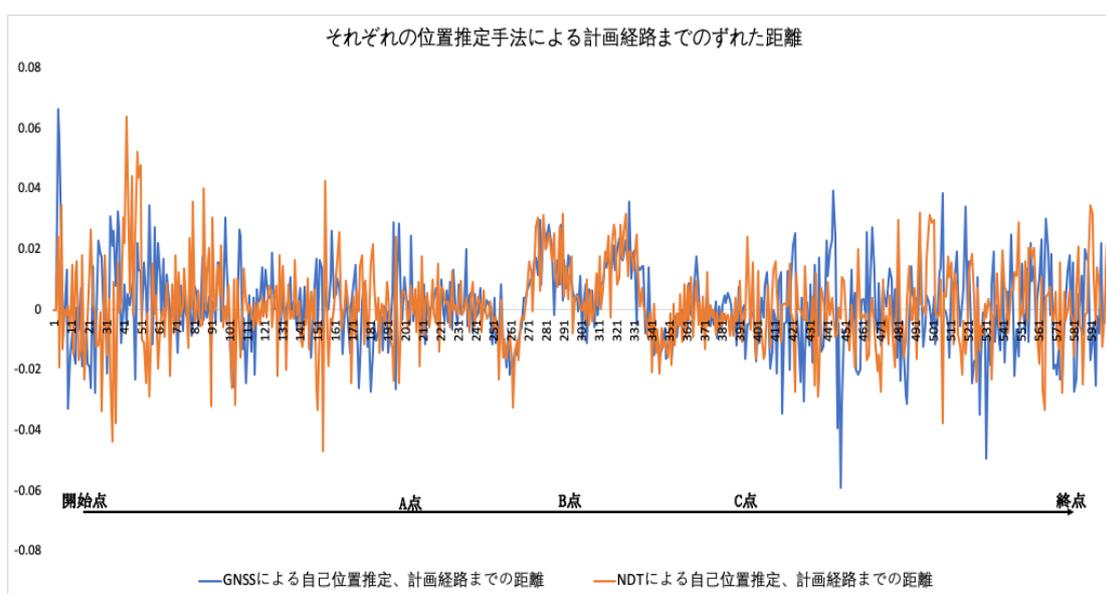


図 7-6. それぞれの位置推定手法による計画経路までのずれた距離

7.2.2 実証実験中ローカライザーの応用

図 7-7 は日間賀島の実証実験中に、ローカライザー切り替え機構の動作の様子である。樹木が多い場所を通過する際に NDT による自己位置推定を行い、通過した後は GNSS-RTK による自己位置推定を行う。緑色の背景は利用最中のローカライザーである。青色の背景は利用していないローカライザーだが、推定された自車位置と計画経路までの距離や角度が安全範囲内を意味する。推定された自車位置と計画までの距離や角度は安全範囲を超える場合、背景は赤色に変わる。

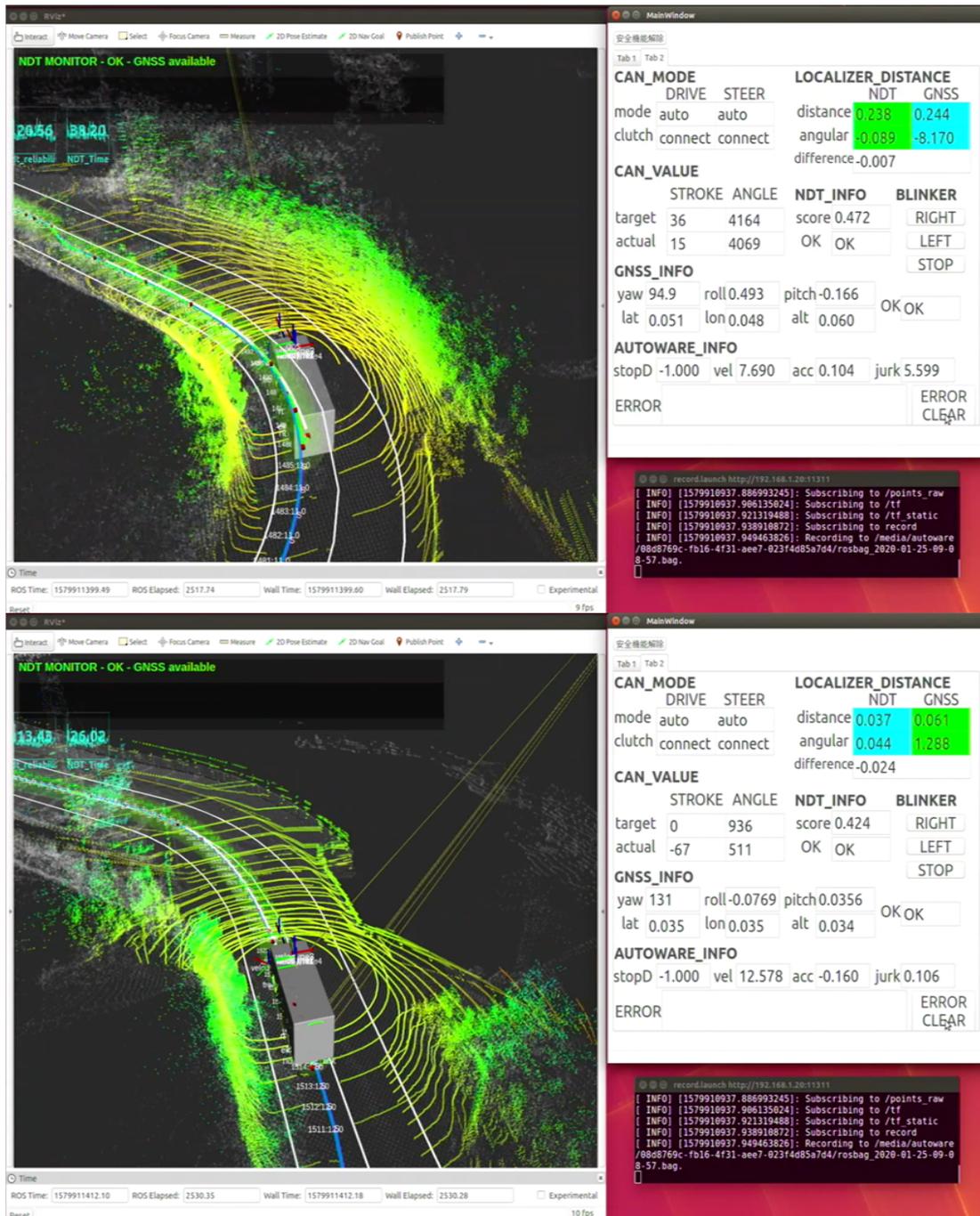


図 7-7. ローカルライザー切り替え機構を利用した様子

7.3 定時点灯信号についてサイクル情報の利用

信号のサイクル情報を利用手法は定時点灯の信号に対する開発していた。信号機の色情報を事前に調査し、世界時間に合わせて、繰り返しで流れてくる。信号のサイクル情報を利用すると、逆光また貨物車両に立ち往生して信号を識別できない場合でも信号情報を算出することが可能である。更に、次の場合図 7-4 の示すように、車両のブレーキ制御はスムーズに行うことができる。

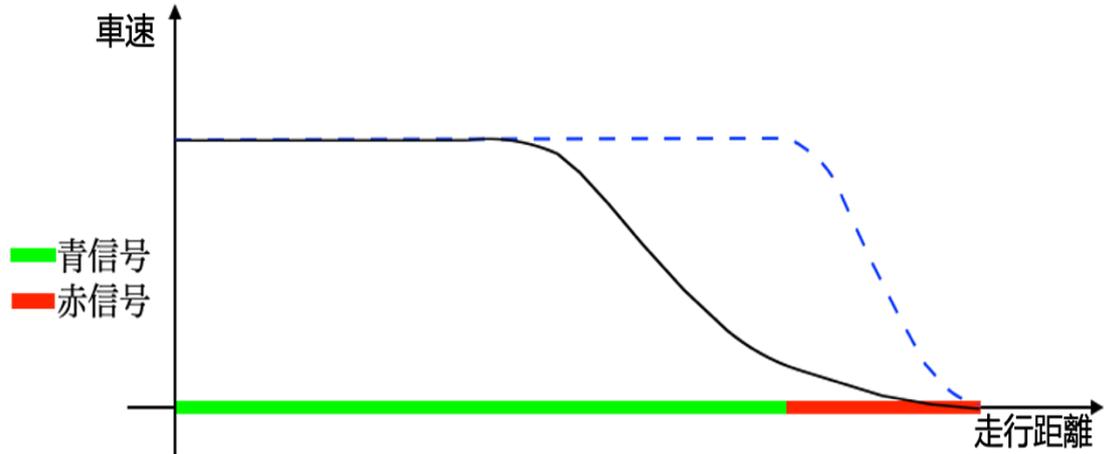


図 7-4. 信号のサイクル情報を利用の上ブレーキ制御を行う

図の中に、青い信号だが、そろそろ赤信号に変わることが信号のサイクル情報による算出されていた、現在の車速による一時停止線でスムーズ、急減速をしないように止められると、赤信号に変わる前にブレーキ制御を行うことが必要である。点線は信号が赤に変わる際に、ブレーキ制御を行い、停止線で止まるまで車速の変化であり、実線は信号のサイクル情報を利用し、現在の車速によるブレーキ制御を行い、停止線で止まるまで車速の変化である。図中の実線の勾配は点線より小さくて、スムーズに止まれることを確認した。

7.4 フェールセーフ機能

筆者らは埼玉工大の自動運転バスを安全走行、事故回避するために、以下で示したフェールセーフ機能を追加した（表 7-1）。フェールセーフ機能は車両が自動走行中に、危険が発生しそうな場合、ドライバーにテイクオーバーリクエストを出して、自動運転モードを切って、手動介入を行う。

表 7-1. フェールセーフ機能が動作する場合

1.	操舵角の角速度が閾値を超えた状態で同じ方向に動き続けた
2.	車速に基づいた設定されたステアリングの操舵角が閾値を超えた
3.	GNSS の測位誤差が閾値を超えた
4.	NDT マッチングスコアが閾値を超えた
5.	信号が赤の時に安全に停車できる走行範囲にて信号色を識別できなかった
6.	信号は貨物車両が立ち往生して見えなかった、停電で付いていなかった
7.	車両後輪中心から走行経路までの偏移距離が閾値を超えた
8.	道路白線を超えた
9.	...

フェールセーフ機能が動作する際に、動作の原因はユーザーインターフェースでエラー情報として表示する。更に、視覚の注意を導くため、聴覚顕著性方向モデル [128]により、フェールセーフ機能が動作する際に、警告音を流して、オペレーターとドライバーに通知する。フェールセーフ機能は実証実験中に様々な情報を監視し、走行安全性を向上した。

第8章 結論

本文は埼玉工業大学の自動運転バスが参加していた自動運転実証実験による、従来の自動運転事故を分析しながら、安全走行を確保する手法を提案した。従来の自動運転事故の原因と自動運転実証実験中にあるヒヤリハットの原因をまとめて、それぞれに解決方法を以下のように纏めた。

1. 車載カメラの視認性を高めるためにヘイズ除去手法を提案した。特に悪天候の自動走行において、ヘイズ除去処理は車載カメラの障害物検出率を高めて、ある程度の走行安全性を向上した。
2. 自動運転実証実験を基づいて、走行中に発生した危険状況に対して様々な対策方法を含めて、走行中に車両の内部データなど重要な情報を監視しながら、自動運転実験車両用のドライブレコーダーを提案した。
3. 様々な安全確保機能は自動運転バスの中速走行する際に蛇行の問題、定時点灯信号に対する信号が識別できない問題、走行コースにより測位誤差は大きい所が混在していた問題を解決した。更に、フェールセーフ機能は安全走行を確保する。

以上の提案手法は埼玉工業大学の自動運転バスに実装されていて、車両の走行安全性を高めることが確認できた。

今後の課題は図7-1で示すような起伏が激しい道路と長いトンネルの走行環境において事故位置推定の問題である。道路の起伏が激しい場合、LIDAR スキャンマッチングによる算出された自己位置が不安定のため、自動走行の危険性が高い。LIDAR スキャンマッチングの安定性を高める方法は問題となる。長いトンネルの場合、LIDAR からトンネルの壁までの距離が同じく、自己位置を算出することが難しい。そこで、オドメトリとIMU情報を用い、位置推定方法が課題となる。



図7-1. LIDARによる走行が厳しい環境

第9章 参考文献

- [1] “California DMV.” [Online]. Available: <https://www.dmv.ca.gov/portal/dmv>. [Accessed: 29-Apr-2020].
- [2] “2019 Autonomous Vehicle Disengagement Reports.” [Online]. Available: https://www.dmv.ca.gov/portal/dmv/detail/vr/autonomous/disengagement_report_2019. [Accessed: 29-Apr-2020].
- [3] K. He, J. Sun, and X. Tang, “Single image haze removal using dark channel prior,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 12, pp. 2341-2353, 2011, doi: 10.1109/TPAMI.2010.168.
- [4] J. Redmon and A. Farhadi, “YOLOv3: An Incremental Improvement,” 2018, doi: 10.1109/CVPR.2017.690.
- [5] “自動運転車 - Wikipedia.” [Online]. Available: <https://ja.wikipedia.org/wiki/自動運転車>. [Accessed: 09-Apr-2020].
- [6] “DARPAグランド・チャレンジ - Wikipedia.” [Online]. Available: <https://ja.wikipedia.org/wiki/DARPAグランド・チャレンジ>. [Accessed: 09-Apr-2020].
- [7] “Stanford Racing.” [Online]. Available: <https://cs.stanford.edu/group/roadrunner/old/index.html>. [Accessed: 09-Apr-2020].
- [8] J. G. Mooney and E. N. Johnson, “A Comparison of Automatic Nap-of-the-earth Guidance Strategies for Helicopters,” *J. F. Robot.*, vol. 33, no. 1, pp. 1-17, 2014, doi: 10.1002/rob.
- [9] “Tartan Racing @ Carnegie Mellon.” [Online]. Available: <http://www.tartanracing.org/>. [Accessed: 09-Apr-2020].
- [10] C. Urmson *et al.*, “Tartan Racing: A Multi-Modal Approach to the DARPA Urban Challenge,” *Defense*, vol. 94, no. 4, pp. 386-387, 2007, doi: 10.1002/rob.20251.
- [11] C. Badue *et al.*, “Self-Driving Cars: A Survey,” 2019.
- [12] “Waymo.” [Online]. Available: <https://waymo.com/>. [Accessed: 09-Apr-2020].
- [13] “Cruise.” [Online]. Available: <https://www.getcruise.com/technology/>. [Accessed: 29-Apr-2020].
- [14] “Uber - Wikipedia.” [Online]. Available: <https://ja.wikipedia.org/wiki/Uber>. [Accessed: 09-Apr-2020].
- [15] “Apollo.” [Online]. Available: <http://apollo.auto/>. [Accessed: 09-Apr-2020].
- [16] “Autopilot.” [Online]. Available: <https://www.tesla.com/jp/autopilot>. [Accessed: 11-Apr-2020].
- [17] “自動運転車のテクノロジーとソリューション - NVIDIA Automotive.” [Online]. Available: <https://www.nvidia.com/ja-jp/self-driving-cars/>. [Accessed: 09-Apr-2020].
- [18] “Mobileye | Autonomous Driving & ADAS (Advanced Driver Assistance

- Systems).” [Online]. Available: <https://www.mobileye.com/>. [Accessed: 09-Apr-2020].
- [19] “Pony.ai.” [Online]. Available: <https://pony.ai/en/index.html>. [Accessed: 09-Apr-2020].
- [20] “トヨタ自動車株式会社 公式企業サイト.” [Online]. Available: https://global.toyota.jp/?padid=not_tjptop_menu_global.toyota. [Accessed: 11-Apr-2020].
- [21] “TuSimple.” [Online]. Available: <https://www.tusimple.com/>. [Accessed: 09-Apr-2020].
- [22] “テクニカルペーパー 自動車用運転自動化システムのレベル 分類及び定義 Taxonomy and definitions for terms related to driving automation systems for On-Road Motor Vehicles.”
- [23] “Autoware.AI.” [Online]. Available: <https://www.autoware.ai/>. [Accessed: 09-Apr-2020].
- [24] “ApolloAuto/apollo: An open autonomous driving platform.” [Online]. Available: <https://github.com/ApolloAuto/apollo>. [Accessed: 12-Apr-2020].
- [25] “ROS Wiki.” [Online]. Available: <http://wiki.ros.org/>. [Accessed: 09-Apr-2020].
- [26] “株式会社ティアフォー.” [Online]. Available: <https://tier4.jp/>. [Accessed: 14-Apr-2020].
- [27] “Tesla事故_CCTV.” [Online]. Available: <http://m.news.cctv.com/2016/09/14/ARTI080nLahX0ezuWjLstquq160914.shtml>. [Accessed: 29-Jun-2020].
- [28] “Uber事故解析.” [Online]. Available: https://www.sohu.com/a/225986599_765855. [Accessed: 29-Jun-2020].
- [29] 名古屋大学 低速自動運転車両事故 検証委員会, “名古屋大学低速自動運転車両事故報告書,” Aug. 2019.
- [30] “群馬大学の自動運転車が物損事故.” [Online]. Available: <https://www.iza.ne.jp/kiji/life/photos/170902/lif17090211270007-p2.html>. [Accessed: 29-Jun-2020].
- [31] M. Wada, F. Kameda, and Y. Saito, “A Study on Steering Control for a Joystick Car Drive System.”
- [32] Z. Wang, D. Watabe, and J. Cao, “Improving visibility of a fast dehazing method,” in *World Automation Congress Proceedings*, 2016, vol. 2016-Octob, doi: 10.1109/WAC.2016.7582960.
- [33] Z. WANG, D. WATABE, and J. CAO, “Real-Time Grayscale Dehazing Scheme For Car Vision,” *Int. Symp. Affect. Sci. Eng.*, vol. ISASE2018, no. 0, pp. 1-6, 2018, doi: 10.5057/isase.2018-c000012.
- [34] Z. WANG and D. WATABE, “Research on Haze Removal for Autonomous Car,” *Trans. Japan Soc. Kansei Eng.*, vol. 18, no. 6, pp. 417-421, 2019, doi: 10.5057/jjske.tjske-d-19-00004.
- [35] Z. WANG, H. SAI, K. OGIWARA, D. WATABE, Y. SAITO, and M. WADA, “An Operator Interface for Autonomous Vehicles,” *Int. Symp. Affect. Sci.*

- Eng.*, vol. ISASE2020, pp. 1-4, 2020, doi: 10.5057/isase.2020-C000014.
- [36] Z. Wang, D. Watabe, H. Sai, Y. Saito, and M. Wada, “ACDR: Autonomous-Car Drive Recorder,” *J. Robot. Mechatronics*, vol. 32, no. 3, pp. 634-637, Jun. 2020, doi: 10.20965/jrm.2020.p0634.
- [37] “Velodyne Lidar.” [Online]. Available: <https://velodynelidar.com/>. [Accessed: 09-Apr-2020].
- [38] “RoboSense LiDAR.” [Online]. Available: <https://www.robosense.ai/>. [Accessed: 09-Apr-2020].
- [39] “HESAI.” [Online]. Available: <https://www.hesaitech.com/en>. [Accessed: 09-Apr-2020].
- [40] “測量に関するミニ知識 | 国土地理院.” [Online]. Available: <https://www.gsi.go.jp/chubu/minichishiki12.html>. [Accessed: 12-Apr-2020].
- [41] “Delphi Electronically Scanning Radar,” 2009.
- [42] “Delphi SRR2 - Rear and Side Detection System.” .
- [43] “ドップラー・レーダー - Wikipedia.” [Online]. Available: <https://ja.wikipedia.org/wiki/ドップラー・レーダー>. [Accessed: 12-Apr-2020].
- [44] “Machine Vision - Area Scan Cameras | FLIR Systems.” [Online]. Available: <https://www.flir.com/iis/machine-vision/>. [Accessed: 30-Apr-2020].
- [45] “ADASKY | Driven to Save Lives.” [Online]. Available: <https://www.adasky.com/>. [Accessed: 09-Apr-2020].
- [46] S. Shalev-Shwartz, S. Shammah, and A. Shashua, “On a Formal Model of Safe and Scalable Self-driving Cars,” 2017.
- [47] D. Nassi, R. Ben-Netanel, Y. Elovici, and B. Nassi, “MobilBye: Attacking ADAS with Camera Spoofing.”
- [48] “MobilBye - Injecting a traffic sign to advanced driver assistance systems (ADAS) - YouTube.” [Online]. Available: <https://www.youtube.com/watch?v=PP-qTdRugEI&feature=youtu.be>. [Accessed: 10-Apr-2020].
- [49] “衛星測位システム - Wikipedia.” [Online]. Available: <https://ja.wikipedia.org/wiki/衛星測位システム>. [Accessed: 11-Apr-2020].
- [50] “搬送波位相測定値による精密測位の理論及び解析処理.” [Online]. Available: http://gpspp.sakura.ne.jp/tutorial/html/gps_symp_2005_2.htm. [Accessed: 11-Apr-2020].
- [51] “トランジスタ技術2019年10月号.” [Online]. Available: <https://toragi.cqpub.co.jp/tabid/889/Default.aspx>. [Accessed: 11-Apr-2020].
- [52] “基準局掲示板.” [Online]. Available: <http://rtk.silentsystem.jp/>. [Accessed: 11-Apr-2020].
- [53] “GPS気象学-GPS測量における大気遅延量の処理.” [Online]. Available: http://www.geod.jpn.org/web-text/part3_2005/tsuji/tsuji-2.html. [Accessed: 11-Apr-2020].
- [54] “慣性計測装置 - Wikipedia.” [Online]. Available: <https://ja.wikipedia.org/wiki/慣性計測装置>. [Accessed: 11-Apr-2020].
- [55] “慣性計測ユニット (IMU) | アナログ・デバイスズ.” [Online].

- Available: https://www.analog.com/jp/landing-pages/003/sensor_pv_jp/sensor_home_jp/imu.html#. [Accessed: 11-Apr-2020].
- [56] A. Solution, “SPAN® IMU-ISA-100C.” pp. 3-4, 1993.
- [57] “オペレーティングシステム - Wikipedia.” [Online]. Available: <https://ja.wikipedia.org/wiki/オペレーティングシステム>. [Accessed: 12-Apr-2020].
- [58] “GitHub - ApolloAuto/apollo-kernel: Collections of Apollo Kernels.” [Online]. Available: <https://github.com/ApolloAuto/apollo-kernel>. [Accessed: 12-Apr-2020].
- [59] “リアルタイムオペレーティングシステム - Wikipedia.” [Online]. Available: <https://ja.wikipedia.org/wiki/リアルタイムオペレーティングシステム>. [Accessed: 12-Apr-2020].
- [60] “Robot Operating System - Wikipedia.” [Online]. Available: https://ja.wikipedia.org/wiki/Robot_Operating_System. [Accessed: 12-Apr-2020].
- [61] “Apollo - Udacity.” [Online]. Available: <https://classroom.udacity.com/courses/ud0419>. [Accessed: 14-Apr-2020].
- [62] “DeepMap Blog.” [Online]. Available: <https://medium.com/deepmap-blog>. [Accessed: 14-Apr-2020].
- [63] F. Poggenhans *et al.*, “Lanelet2: A high-definition map framework for the future of automated driving.”
- [64] K. Jo, C. Kim, and M. Sunwoo, “Simultaneous localization and map change update for the high definition map-based autonomous driving car,” *Sensors (Switzerland)*, vol. 18, no. 9, p. 3145, Sep. 2018, doi: 10.3390/s18093145.
- [65] “HD Maps for Autonomous Driving and Driver Assistance | HERE.” [Online]. Available: <https://www.here.com/products/automotive/hd-maps>. [Accessed: 13-Apr-2020].
- [66] “TomTom | Home.” [Online]. Available: https://www.tomtom.com/ja_jp/. [Accessed: 14-Apr-2020].
- [67] E. Takeuchi and T. Tsubouchi, “A Fast Scan Matching in 3-D Space using 3D Normal Distributions Transform for Mobile Robotic Mapping,” *Transform*, pp. 2-7, 2006.
- [68] J. Zhang and S. Singh, “Low-drift and real-time lidar odometry and mapping,” *Auton. Robots*, vol. 41, no. 2, pp. 401-416, 2017, doi: 10.1007/s10514-016-9548-2.
- [69] T. Shan and B. Englot, “LeGO-LOAM: Lightweight and Ground-Optimized Lidar Odometry and Mapping on Variable Terrain,” *IEEE Int. Conf. Intell. Robot. Syst.*, pp. 4758-4765, 2018, doi: 10.1109/IRoS.2018.8594299.
- [70] “GitHub - RobustFieldAutonomyLab/LeGO-LOAM: LeGO-LOAM: Lightweight and Ground-Optimized Lidar Odometry and Mapping on Variable Terrain.” [Online]. Available: <https://github.com/RobustFieldAutonomyLab/LeGO-LOAM>. [Accessed: 14-Apr-2020].

- [71] N. D. McKay and P. J. Besl, "A method for registration of 3-D shapes," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 14, no. 2, pp. 239-256, 1992.
- [72] J. Salvi, C. Matabosch, D. Fofi, and J. Forest, "A review of recent range image registration methods with accuracy evaluation," *Image Vis. Comput.*, vol. 25, no. 5, pp. 578-596, 2007, doi: 10.1016/j.imavis.2006.05.012.
- [73] M. Poreba and F. Goulette, "A robust linear feature-based procedure for automated registration of point clouds," *Sensors (Switzerland)*, vol. 15, no. 1, pp. 1435-1457, 2015, doi: 10.3390/s150101435.
- [74] P. Biber, "The Normal Distributions Transform: A New Approach to Laser Scan Matching," *IEEE Int. Conf. Intell. Robot. Syst.*, vol. 3, no. October, pp. 2743-2748, 2003, doi: 10.1109/iros.2003.1249285.
- [75] "Robot Localization II: The Histogram Filter - deepideas.net." [Online]. Available: <https://www.deepideas.net/robot-localization-histogram-filter/>. [Accessed: 15-Apr-2020].
- [76] "Gyroscope - Wikipedia." [Online]. Available: <https://en.wikipedia.org/wiki/Gyroscope>. [Accessed: 15-Apr-2020].
- [77] P. E. Sarlin, C. Cadena, R. Siegwart, and M. Dymczyk, "From coarse to fine: Robust hierarchical localization at large scale," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2019, vol. 2019-June, pp. 12708-12717, doi: 10.1109/CVPR.2019.01300.
- [78] "カルマンフィルター - Wikipedia." [Online]. Available: <https://ja.wikipedia.org/wiki/カルマンフィルター>. [Accessed: 17-Apr-2020].
- [79] "How a Kalman filter works, in pictures | Bzarg." [Online]. Available: <https://www.bzarg.com/p/how-a-kalman-filter-works-in-pictures/>. [Accessed: 17-Apr-2020].
- [80] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2016, vol. 2016-Decem, pp. 779-788, doi: 10.1109/CVPR.2016.91.
- [81] R. Girshick, "Fast R-CNN," *arXiv*, 2015.
- [82] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," pp. 1-10, 2015.
- [83] "SegNet." [Online]. Available: <https://mi.eng.cam.ac.uk/projects/segnet/>. [Accessed: 18-Apr-2020].
- [84] V. Badrinarayanan, A. Handa, and R. Cipolla, "SegNet: A Deep Convolutional Encoder-Decoder Architecture for Robust Semantic Pixel-Wise Labelling," May 2015.
- [85] A. H. Lang, S. Vora, H. Caesar, L. Zhou, J. Yang, and O. Beijbom, "PointPillars: Fast Encoders for Object Detection from Point Clouds," *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 2019-June, pp. 12689-12697, Dec. 2018.

- [86] A. Asvadi, L. Garrote, C. Premebida, P. Peixoto, and U. J. Nunes, “DepthCN: Vehicle Detection Using 3D-LIDAR and ConvNet.”
- [87] M. Herzog and K. Dietmayer, “Training a Fast Object Detector for LiDAR Range Images Using Labeled Data from Sensors with Higher Resolution,” doi: 10.1109/ITSC.2019.8917011.
- [88] M. Tomizuka, “Model based prediction, preview and robust controls in motion control systems,” in *Proceedings of 4th IEEE International Workshop on Advanced Motion Control - AMC '96 - MIE*, 1996, vol. 1, pp. 1-6, doi: 10.1109/AMC.1996.509370.
- [89] J. Bosch and H. H. Olsson, “Data - Driven Continuous Evolution of Smart Systems,” pp. 28-34, 2016.
- [90] D. Dolgov, S. Thrun, M. Montemerlo, and J. Diebel, “Practical search techniques in path planning for autonomous driving,” in *International Symposium on Combinatorial Search, SoCS 2008*, 2008.
- [91] T. M. Howard, C. J. Green, A. Kelly, and D. Ferguson, “State Space Sampling of Feasible Motions for High-Performance Mobile Robot Navigation in Complex Environments,” doi: 10.1002/rob.20244.
- [92] B. R. Iyer and C. V. Vishveshwara, “Frenet-Serret description of gyroscopic precession,” *Phys. Rev. D*, vol. 48, no. 12, pp. 5706-5720, Oct. 1993, doi: 10.1103/PhysRevD.48.5706.
- [93] “PID制御 - Wikipedia.” [Online]. Available: <https://ja.wikipedia.org/wiki/PID制御>. [Accessed: 19-Apr-2020].
- [94] R Rajamani, *Vehicle Dynamics and Control*. New York: Springer-Verlag, 2006.
- [95] “Apollo - overview of the control module, CSDN blog.” [Online]. Available: <https://blog.csdn.net/u013914471/article/details/82775091>. [Accessed: 19-Apr-2020].
- [96] “Model predictive control - Wikipedia.” [Online]. Available: https://en.wikipedia.org/wiki/Model_predictive_control. [Accessed: 04-Jun-2020].
- [97] “一人乗りロボ・物流ロボ・宅配ロボ・警備ロボ・無人フォーク・車両&バスの自動運転のZMP.” [Online]. Available: <https://www.zmp.co.jp/>. [Accessed: 23-Apr-2020].
- [98] “先進モビリティ株式会社 | 先進モビリティ、それは自動運転を軸とするスマートな移動手段を実現する社会。” [Online]. Available: <https://www.as-mobi.com/>. [Accessed: 29-Jun-2020].
- [99] “自動運転営業バス、群馬県前橋市に登場。” [Online]. Available: <https://jidounten-lab.com/w-autonomous-bus-maebashi-gunma-level4>. [Accessed: 29-Jun-2020].
- [100] S. G. Narasimhan and S. K. Nayar, “Contrast restoration of weather degraded images,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25, no. 6, pp. 713-724, 2003, doi: 10.1109/TPAMI.2003.1201821.
- [101] Y. Y. Schechner, S. G. Narasimhan, and S. K. Nayar, “Instant dehazing of images using polarization,” in *Proceedings of the 2001 IEEE*

Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001, 2001, vol. 1, pp. I-325-I-332, doi: 10.1109/CVPR.2001.990493.

[102] S. Shwartz, E. Namer, and Y. Y. Schechner, “Blind Haze Separation,” in *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Volume 2 (CVPR' 06)*, 2006, vol. 2, pp. 1984-1991, doi: 10.1109/CVPR.2006.71.

[103] J. Kopf *et al.*, “Deep photo: model-based photograph enhancement and viewing,” *ACM Trans. Graph.*, vol. 27, no. 5, pp. 1-10, 2008, doi: 10.1145/1409060.1409069.

[104] K. Gibson, D. Võ, and T. Nguyen, “An investigation in dehazing compressed images and video,” *MTS/IEEE Seattle, OCEANS 2010*. 2010, doi: 10.1109/OCEANS.2010.5664479.

[105] Q. Liu, M. Chen, and D. Zhou, “Fast haze removal from a single image,” *2013 25th Chinese Control Decis. Conf. CCDC 2013*, pp. 3780-3785, 2013, doi: 10.1109/CCDC.2013.6561607.

[106] B. Cai, X. Xu, K. Jia, C. Qing, and D. Tao, “DehazeNet: An end-to-end system for single image haze removal,” *IEEE Trans. Image Process.*, vol. 25, no. 11, pp. 5187-5198, 2016, doi: 10.1109/TIP.2016.2598681.

[107] Y. H. Lai, Y. L. Chen, C. J. Chiou, and C. T. Hsu, “Single-image dehazing via optimal transmission map under scene priors,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 25, no. 1, pp. 1-14, 2015, doi: 10.1109/TCSVT.2014.2329381.

[108] B. Li, X. Peng, Z. Wang, J. Xu, and D. Feng, “AOD-Net: All-in-One Dehazing Network,” *Proc. IEEE Int. Conf. Comput. Vis.*, vol. 2017-Octob, pp. 4780-4788, 2017, doi: 10.1109/ICCV.2017.511.

[109] B. Li *et al.*, “Benchmarking Single Image Dehazing and Beyond,” vol. 14, no. 8, pp. 1-12, 2017.

[110] S. G. Narasimhan and S. K. Nayar, “Vision and the atmosphere,” *Int. J. Comput. Vis.*, vol. 48, no. 3, pp. 233-254, Jul. 2002, doi: 10.1023/A:1016328200723.

[111] L. Boltzmann, “Studien über das Gleichgewicht der lebendigen Kraft zwischen bewegten materiellen Punkten,” *Wiener Berichte*, vol. 58, pp. 517-560, 1868.

[112] P. Jaccard, “Étude comparative de la distribution florale dans une portion des Alpes et des Jura,” *Bull. la Société vaudoise des Sci. Nat.*, vol. 37, pp. 547-579, 1901.

[113] P. Jaccard, “THE DISTRIBUTION OF THE FLORA IN THE ALPINE ZONE.,” *New Phytol.*, vol. 11, no. 2, pp. 37-50, 1912, doi: 10.1111/j.1469-8137.1912.tb05611.x.

[114] “ピーク信号対雑音比 - Wikipedia.” [Online]. Available: <https://ja.wikipedia.org/wiki/ピーク信号対雑音比>. [Accessed: 21-Apr-2020].

[115] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, “Image Quality Assessment: From Error Visibility to Structural Similarity,” 2004.

[116] “死亡事故を起こした自動運転車の車載カメラ映像 - YouTube.”

[Online]. Available: <https://www.youtube.com/watch?v=06upo5t0GuQ>. [Accessed: 29-Jun-2020].

[117] 定雄堀野 *et al.*, “映像記録型ドライブレコーダーを用いたタクシー事故・ニアミス解析と予防安全,” pp. 276-277, 2009, doi: 10.14874/JERGO.45SPL.0.276.0.

[118] R. Matsumi, P. Raksincharoensak, and M. Nagai, “Study on autonomous intelligent drive system based on potential field with hazard anticipation,” *J. Robot. Mechatronics*, vol. 27, no. 1, pp. 5-11, Feb. 2015, doi: 10.20965/jrm.2015.p0005.

[119] “ドライブレコーダー セルスター工業株式会社.” [Online]. Available: <https://www.cellstar.co.jp/products/recorder/>. [Accessed: 29-Jun-2020].

[120] “d’ Action 360 - ダクシオン 360 - | CARMATE.” [Online]. Available: <https://daction.carmate.jp/>. [Accessed: 29-Jun-2020].

[121] A. Pérez, M. I. García, M. Nieto, J. L. Pedraza, S. Rodríguez, and J. Zamorano, “Argos: An advanced in-vehicle data recorder on a massively sensorized vehicle for car driver behavior experimentation,” *IEEE Trans. Intell. Transp. Syst.*, vol. 11, no. 2, pp. 463-473, Jun. 2010, doi: 10.1109/TITS.2010.2046323.

[122] O. Musicant, T. Lotan, and T. Toledo, “Safety correlation and Implications of an In- In - vehicle Data Recorder on Driver Behavior,” in *Transportation Research Board Annual Meeting*, 2007, no. January.

[123] V. K. Veitas and S. Delaere, “In-vehicle data recording, storage and access management in autonomous vehicles,” May 2018.

[124] S.-Y. Chen, “Vehicle event data recorder and anti-theft alarm system with 360 degree panograph function.” 17-Jan-2012.

[125] K. Driggs-Campbell, V. Shia, and R. Bajcsy, “Decisions for autonomous vehicles: Integrating sensors, communication, and control,” in *HiCoNS 2014 - Proceedings of the 3rd International Conference on High Confidence Networked Systems (Part of CPS Week)*, 2014, pp. 59-60, doi: 10.1145/2566468.2576851.

[126] S. Debernard, C. Chauvin, R. Pokam, and S. Langlois, “Designing Human-Machine Interface for Autonomous Vehicles,” *IFAC-PapersOnLine*, vol. 49, no. 19, pp. 609-614, Jan. 2016, doi: 10.1016/j.ifacol.2016.10.629.

[127] “Kvaser | Advanced CAN Solutions.” [Online]. Available: <https://www.kvaser.com/>. [Accessed: 25-Apr-2020].

[128] E. M. Kaya and M. Elhilali, “Modelling auditory attention,” *Philosophical Transactions of the Royal Society B: Biological Sciences*, vol. 372, no. 1714. 2017, doi: 10.1098/rstb.2016.0101.

謝辞

本論文は筆者が埼玉工業大学大学院工学研究科システム工学専攻博士後期課程に在籍中の研究成果をまとめものである。同専攻教授渡部先生には指導教官として本研究の実施の機会を与えて戴き、その遂行に当たって終始、ご指導を戴いた。ここに深謝の意を表す。

本研究は埼玉工業大学自動運転開発チームの崔英泰には研究遂行にあたり日頃より有益なご討論後助言を戴いた。ここに感謝の意を表す。

最後に、日本での留学中に与えてくれた生活上の支援に対して、両親に深謝したいと思う。