

平成29年度採択

戦略的基盤技術高度化・連携支援事業

戦略的基盤技術高度化支援事業

「積雪寒冷地域の交通弱者移動支援のための雪道走行を可能とする
自動運転技術の開発」

研究開発成果等報告書

令和2年5月

担当局 北海道経済産業局

補助事業者 株式会社ウィッツ、国立大学法人北海道大学

目次

1 研究開発の概要	- 5 -
1-1 研究開発の背景・研究目的及び目標	- 5 -
1-2 研究体制	- 19 -
1-3 成果概要	- 20 -
1-4 当該研究開発の連絡窓口	- 22 -
第2章 具体的な実施内容と成果	- 23 -
2-1. 【サブテーマ1-1】積雪環境で機能する自己位置推定技術開発	- 23 -
2-1-1. 雪道対応 SLAM 技術開発	- 23 -
2-1-1-1 (1) 初年度の目標	- 23 -
2-1-1-1 (2) 具体的な内容	- 23 -
2-1-1-1 (3) 2年度の目標	- 32 -
2-1-1-1 (4) 具体的な内容	- 32 -
2-1-1-2 (5) 3年度の目標	- 40 -
2-1-1-1 (6) 具体的な内容	- 40 -
2-1-2. 雪道対応セマンティック・セグメンテーション技術の開発	- 56 -
2-1-2-1 (1) 初年度の目標	- 56 -
2-1-2-2 (2) 具体的な内容	- 56 -
2-1-2-3 (3) 2年度の目標	- 72 -
2-1-2-4 (4) 具体的な内容	- 72 -
2-1-2-5 (5) 3年度の目標	- 82 -
2-1-2-6 (6) 具体的な内容	- 82 -
2-1-3. 雪道対応センサ・フュージョン技術の開発	- 89 -
2-1-3-1 (1) 初年度の目標	- 89 -
2-1-3-2 (2) 具体的な内容	- 89 -
2-1-3-3 (3) 2年度の目標	- 92 -
2-1-3-4 (4) 具体的な内容	- 92 -
2-1-3-5 (5) 3年度の目標	- 103 -
2-1-3-6 (6) 具体的な内容	- 103 -
2-1-4. 実証実験	- 112 -
2-1-4-1 (1) 初年度の目標	- 112 -
2-1-4-2 (2) 具体的な内容	- 112 -
2-1-4-3 (3) 2年度の目標	- 124 -
2-1-4-4 (4) 具体的な内容	- 125 -
2-1-4-5 (5) 3年度の目標	- 128 -
2-1-4-6 (6) 具体的な内容	- 130 -
2-1-5. 成果	- 133 -
2-1-5-1. 雪道対応 SLAM 技術開発	- 133 -
2-1-5-2. 雪道対応セマンティック・セグメンテーション技術の開発	- 133 -

2-1-5-3. 雪道対応センサ・フュージョン技術の開発	134
2-1-5-4. 実証実験	135
2-2. 【サブテーマ2】積雪環境仮想シミュレータ開発	136
2-2-1. 積雪環境仮想シミュレータの仕様策定	136
2-2-1-1. シミュレーション環境の調査	136
2-2-1-2. シミュレータ構成の検討	138
2-2-1-3. センサモデルの検討	140
2-2-1-4. テストコースの検討	143
2-2-2-1. シミュレータ環境の構築	144
2-2-2-2. テストコースの実装	144
2-2-2-3. センサモデルの実装	145
2-2-3. 積雪環境仮想シミュレータの動作検証	149
2-2-3-1. 動作検証項目の作成	149
2-2-3-2. 動作検証の実施	151
2-2-4. センサモデルの追加開発	152
2-2-4-1. LiDAR Velodyne HDL-32e	152
2-2-4-2. 全方位マルチカメラ Ladybug	154
2-2-4-3. 赤外線サーモグラフィ optris PI450	157
2-2-5. シミュレーションの改良	163
2-2-5-1. 知能化PFとの結合	163
2-2-5-2. 仮想環境改良	168
2-2-5-3. 積雪環境シミュレータに求める性能まとめ	176
2-2-5-3-1. シミュレーション性能	178
2-2-5-3-2. シミュレーション環境	180
2-2-5-3-3. テストシナリオ	181
2-2-6. 芦別コースの追加	183
2-2-7. シナリオ実行機能の追加	200
2-2-8. シミュレータ改善	204
2-2-8-1. 雪壁	204
2-2-8-2. カメラセンサ改善策検討	210
2-2-9. 成果	216
2-3. 【サブテーマ3】グローバル経路計画との連携技術開発	219
2-3-1. 初年度の取り組み	219
2-3-1-1. 初年度の目標	219
2-3-1-2. 具体的な内容	219
2-3-1-2-1. グローバル経路計画連携機能の仕様策定	219
2-3-1-2-2. グローバル経路計画連携機能の開発	223
2-3-1-2-3. 実証実験	228
2-3-2. 2年度の取り組み	231
2-3-2-1. 2年度の目標	231

2-3-2-2. 具体的な内容	- 231 -
2-3-2-2-(1) Waypoint の速度・進入角度の設定機能の開発	- 231 -
2-3-2-2-(2) Waypoint を補正する技術の開発	- 232 -
2-3-2-2-(3) 雪道走行での速度調整機能の検討	- 235 -
2-3-2-2-(4) 実証実験	- 237 -
2-3-3. 3年度の取り組み	- 240 -
2-3-3-1. 3年度の目標	- 240 -
2-3-3-2. 具体的な内容	- 240 -
2-3-3-2-(1) 移動支援アプリの開発	- 241 -
2-3-3-2-(2) 車両連携機能の拡張	- 242 -
2-3-3-2-(3) 雪道走行での速度調整機能の開発	- 243 -
2-3-3-2-(4) 実証実験	- 244 -
2-3-4. 成果	- 247 -
3章 全体総括	- 249 -
3-1. 複数年の研究開発成果	- 249 -
3-2. 研究開発後の課題と事業化について	- 250 -

1 研究開発の概要

1-1 研究開発の背景・研究目的及び目標

○研究開発の概要

本研究は、自動運転に欠かせない周囲状況・自己位置把握技術を、積雪路面にも対応させるものであり、自動運転車の高性能化・高機能化に資するものである。

また自動運転車は、その導入による交通事故の低減効果が期待されていることから、本研究は、冬季積雪環境での自動車交通の安全確保にも資する研究である。さらに、冬季積雪環境における新たな移動・輸送サービスが可能となることから、自動車を活用したサービス製品の高度化にも寄与する研究である。

合わせて本研究では、積雪環境に対応した仮想シミュレータを開発する。これは、自動運転技術の性能評価を効率化するものであり、自動運転技術の研究開発における生産性向上に資するものである。

○研究開発の背景および目的

本事業は、自動車の自動運転技術をさらに革新し、積雪寒冷地域でも利用可能なものとするための研究開発事業である。

現在、自動運転の技術開発は世界的に競争が激化しているものの、多数の利用者が見込める都市部での舗装路や高速道路の環境を主たる走行対象として想定しており、北海道を含む寒冷地域の特徴である冬季の積雪環境は想定対象から外れている。そのため、積雪寒冷地域での自動運転の実用化は、現時点では目途が立っていない（自動運転技術の積雪寒冷地未対応課題）。

そこで本研究では、現在の自動運転技術を積雪環境に適用した場合に生じる課題の一つを解決するための新技術確立し、積雪寒冷地域での自動運転車による革新的モビリティの実現に資することを目的とする（図 1-1(1)）。

本研究は、新技術の試作実装および実車評価のために、ヤマハ発動機が長年にわたって研究開発を続けてきた自動運転車をベースとして活用する。これにより、短期間のうちに、低コストで製品化可能な技術獲得を成功させる。

○従来技術での課題

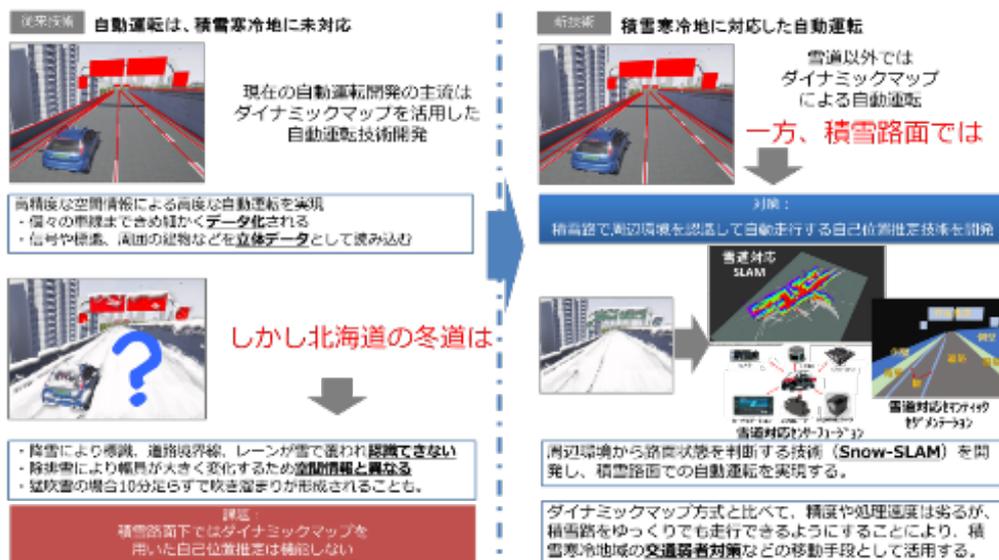


図 1-1(1). 従来技術の課題および新技術による課題解決

○新技術を実現するために解決すべき研究課題

(二) 情報処理に係る技術に関する事項

1 情報処理に係る技術において達成すべき高度化目標

(2) 当該技術の将来の展望

周囲状況の認識技術および車両自己位置の推定技術を高度化して、積雪環境に対応させることにより、積雪寒冷地域での自動運転車の実用化に向けた第一歩を踏み出すことができる。

(4) 川下分野特有の事項

3) その他分野に関する事項

b. 自動車分野に関する事項

① 川下製造業者等の特有の課題及びニーズ

- ア. 当該技術の活用による自動車の高性能化・高機能化
- イ. 自動車製造に関する各種プロセスの生産性向上
- ウ. 交通システムとの接続に向けた自動車の情報化の推進
- エ. 電気自動車等を含めたエネルギーシステム、サービスの実現

本研究では、自動車メーカーが先送りしている積雪寒冷地域での自動運転に伴う技術課題を解決することで、自動運転技術の高性能化・高機能化に貢献する。積雪環境を模擬するシミュレータを開発することで、自動運転技術開発に要する評価時間を短縮し、生産性向上に寄与する。合わせて、遠隔からの経路設定や進行制御のためのインターフェースを整備することで、多様なサービスアプリケーションシステムとの接続を可能とする。これらにより、北海道など冬季の交通困難が課題となる地域における、非積雪地域と同等の革新的な移動・輸送サービスの実現に寄与する。

【自動運転技術開発の動向】

1939年にニューヨークの万国博覧会で提唱され話題を呼んだ自動運転車のコンセプトは、その後の長年におよぶ広範な技術開発の積み重ねを経て、今日いよいよ実用化の段階を迎えている。自動運転車は、事故や渋滞など現代社会が抱える交通問題を解決し、さらには、新たな移動・輸送サービスを可能とすることから、より安全で快適、便利な社会への変革をもたらす技術として、世界中で熱い注目を集めるテーマとなっている。

我が国では安倍首相が「2020年までに運転手が乗車しない自動車によって、地域の人手不足や交通弱者を解消すべき」と発言され、高速道路における無人トラック走行や、無人バス、無人タクシーの社会実験を実施するとの方針が示されるなど、国際競争力維持の観点からも、自動運転技術は我が国が他国に先駆けて実現しなければならない喫緊かつ重要な課題として位置づけられている。これを受けて改訂された官民ITS構想・ロードマップ2016では、2030年までの「世界一安全で円滑な道路交通社会」の実現をゴールに掲げつつ、2019年までの自動運転レベル2（人間の運転者の監視を前提とする準自動運転）の市場化、そして、2025年までの自動運転レベル4（運転者の監視を必要としない完全自動運転）の市場展開開始という段階的なロードマップを打ち出している。この計画を達成するために克服すべき課題は、センサの高精度化・低コスト化、自動認識・判断を担うソフトウェア技術の高度化、高精度立体地図情報の活用など広範にわたるため、自動車メーカーは、IT企業とも連携を強化し、莫大な研究費を投じている。

今日、実用化が見込まれている自動運転技術は、ロボティクス分野の自律移動技術を応用したもので、人間の運転者に代わって自動的にハンドルやブレーキを操作するために、「認知／判断／操作」という運転中に繰り返されるタスクを機械に行わせる技術である。そのために、自動運転車では、以下の処理が繰り返し実行されている：

- 各種のセンサ群から得た情報をもとに車両の自己位置と周囲の状況を把握し(周囲状況・自己位置把握)、

- 目的地に安全かつ円滑に至るために走行すべき経路を決定し(走行経路計画)、
- 決定した経路に追従するように車速や舵角を制御する(車両挙動制御)。

このうち、「周囲状況・自己位置把握」を実現する上で鍵となる技術と目されているのが、「ダイナミックマップ」とよばれる高精度空間情報(立体地図)を活用した自己位置推定技術である。このダイナミックマップ方式は、カメラやレーザーセンサでとらえた人工構造物(道路境界、走行車線の区画線および中心線、交通標識および交通信号など)の位置や形状に関する情報を、記憶している立体地図情報と比較してマッチングさせて自己位置を推定することを動作原理としている。

ダイナミックマップ方式の利点は、高精度な自己位置把握を短時間で行うことができ、周囲状況(交通信号や交通標識、走行可能レーン、横断歩道など)の特定がしやすくなることから、高速な自動運転走行が可能となることである。このことから、都市部での自動運転を円滑かつ安全に実現するためには不可欠の技術とされ、内閣府ではダイナミックマップに関する戦略的イノベーション創造プログラム(SIP)が開始されるなど、官民挙げて立体地図情報データベース技術開発や、デジタルインフラとしての仕様の国際標準化(ISO 14296)などが進められている。

【積雪寒冷地域における課題】

上述のように、ダイナミックマップ方式による周囲状況・自己位置把握は、自動運転にとって極めて重要な技術ではあるが、特に積雪寒冷地域においては、原理的にうまく動作しない懸念がある。なぜなら、積雪によって人工構造物が隠れてしまった場合、センサによる認識ができなくなり、立体地図情報とのマッチングが不可能となるためである。また、雪道では除排雪作業によって道路幅員が大きく変化することや、いわゆる雪塊や吹き溜まりが出現することがあるため(図 1-1(2))、立体地図上では走行可能域であったとしても、実際の積雪環境下ではそうでない可能性がある。したがって、積雪寒冷地域における冬季の自動運転を実現するには、ダイナミックマップ方式に



完全に依存することはできず、それを補完できる周囲状況・自己位置把握の技術開発が避けられない課題となる。

図 1-1(2). 雪道上の吹き溜まり(左)と雪塊(右)

【課題対策方針およびベース車両活用】

上述の課題を克服するために、本研究では、積雪環境でも機能する、周囲状況・自己位置把握の技術開発を実施する。事前に記憶された立体地図情報に依存しない手法として、走行中の車両が自ら周囲地図を作成する技術(一般に SLAM と呼ばれる)が進化しているため、本研究ではこの手法をベースとして改良を施し、積雪路面でも対応できる技術を獲得する(図 1-1(3))。

実用化に向けた新技術の性能評価のためには、実車両による実証実験が不可欠であるが、本研究グループでは、ヤマハ発動機が研究開発を進めている無人自動走行車両(Unmanned Ground Vehicle)を利用する。これは、オフロード用の 4 輪駆動車に自律的自動運転技術を搭載したもので、主として農業分野や災害対策などでの利用に向けて研究開発が進められているものである。

オフロード環境は、人工構造物に依存した位置推定が困難であるという点で積雪環境に類似しているため、本研究による

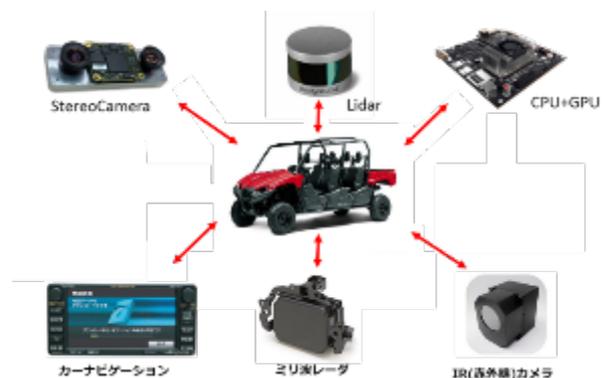


図 1-1(3) 研究開発のベースとする自動運転車と機能拡張

新機能実現のためのベース技術として活用し、これに積雪対応のためのセンサ追加や機能拡張を施した車両を用いて、実証実験を進めていく(図 3)。これにより、新機能の実証実験の準備にかかるコストと期間を大幅に短縮し、本事業期間中の実車評価を可能とする。

②研究開発の具体的な内容

本研究では、積雪環境での自動運転に適用可能な周辺状況・自己位置推定機能（これを、「Snow-SLAM 方式」と名付ける）を実現するために、以下の 3 つのサブテーマにより研究開発を行う：

- サブテーマ 【1. 積雪環境で機能する自己位置推定技術開発】、
- サブテーマ 【2. 積雪環境の仮想シミュレーション技術開発】、および、
- サブテーマ 【3. グローバル経路計画との連携技術開発】。

本研究において、「Snow-SLAM 方式」実証実験用に開発する自動運転車のシステム構成を、図 1-1(4)に示す。本研究の研究機関であるヤマハ発動機が開発を進めている自動運転車（ヤマハ発動機が北米向けに発売しているオフロード 4 輪バギー『Viking IV』の改造車）をベースとして、サブテーマ【1. 積雪環境で機能する自己位置推定技術開発】では、既存の【B】自己位置推定機能、【C】周辺環境認識機能、【F】環境センサに対して雪道対応を行い、

Snow-SLAM 方式を実現する。また、サブテーマ【3. グローバル経路計画との

連携機能開発】では、新たに【G】グローバル経路計画連携機能を開発し、カーナビ等によるグローバル経路計画と、既存の【E】ローカル経路計画を連携させる。

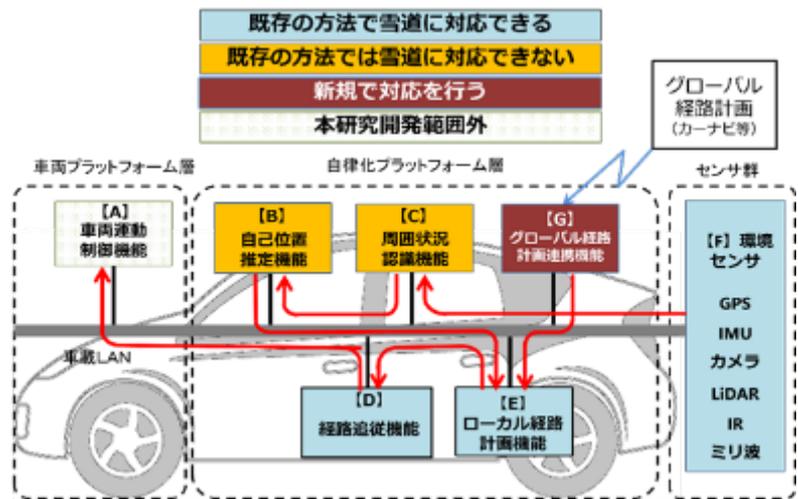


図 1-1(4). 自動運転車のシステム構成と今回の研究開発部位

各サブテーマの内容を、以下に説明する。

【1. 積雪環境で機能する自己位置推定技術開発】

積雪環境におけるダイナミックマップ方式の課題を克服するためには、あらかじめ作成された立体地図情報を用いることなく周辺状況と自己位置を把握できる技術が必要である。これを実現するために、本研究では、SLAM、セマンティック・セグメンテーション、および、センサ・フュージョンという 3 つの技術を雪道に対応させ、組合せて用いることで「Snow-SLAM」を実現する。そのため、以下の 4 つのタスクを設定する。

【1-1】雪道対応 SLAM 技術の開発

SLAM(スラム; Simultaneously Localization And Mapping)は、自己位置推定と周辺地図生成を確率的にフィードバックし合うことで、高精度で安定性の高い周辺状況・自己位置把握を実現する手法であり、主に自律移動ロボットに対して近年、急速に適用が進んでいる技術である。SLAM は、ダイナミックマップ方式とは異なり、あらかじめ作成された地図データを必要としない。その代わりに、車両（またはロボット）が走行中に獲得したセンシング情報に基づいて、自ら周囲の立体地図を作成して自己位置を推定する方式である。ただし、従来の SLAM は、人工建造物のエッジを抽出し、それを特徴点として、地図作成および位置推定を行っている。そのため、積雪環境では（ダイナミッ

クマップ方式と同様に) 特徴点の抽出が難しく十分な精度を得られないと考えられる。そこで本研究では、現在様々な分野で応用が進んでいる深層学習を適用し、その強力な汎化能力によって積雪環境からの特徴点の自動抽出を実現する。これにより、積雪環境に対応可能な SLAM を実現する。

【1-2】雪道対応セマンティック・セグメンテーション技術の開発

セマンティック・セグメンテーションは、画像上の各画素がどの物体に属するのかを推定するための画像処理技術である。ダイナミックマップ方式では、立体地図情報によって周囲状況の把握(交通信号と交通標識の区別等)が可能であるが、SLAM ではそれができない。そのため、カメラで取得した周囲画像にセマンティック・セグメンテーションを施すことによって、路面状態を認識し、走行可能領域や障害物を判別することを目指す。しかしながら、この技術も積雪環境における自動運転に応用された例は未だ無いため、有効性の実証を行う。

【1-3】雪道対応センサ・フュージョン技術の開発

センサ・フュージョンは、複数のセンサ、例えばミリ波レーダやカメラなどから得た多くのデータを統合的に処理することで、単一のセンサでは得られない高度な認識を実現する手法である。道路状況を検知するセンサは、主にマイクロ波、ミリ波、光波を利用したものに分類されるが、雪による影響の受けやすさ、分解能、探知距離、指向性、サイズ、コストなどが様々であるため、これにセンサ・フュージョン技術を適用して、積雪路面の状況を的確にとらえることができる組合せを検討する。その際、製品化においては低価格なセンサを利用できることが重要となるため、精度とコストのバランスを見極める。

【1-4】実証実験

上記の【1-1】～【1-3】を実現するために必要となる実験を、積雪環境にて実施する。ヤマハ発動機の自動運転車両に「Snow-SLAM」システムを試作実装して、様々な積雪状況での各種データ収集を行う。そして収集したデータに基づく各技術の最適化、および、自動運転機能の性能評価を行う。さらに、センサ類のコストダウンの方策を検討する。

その際、積雪環境には様々な要因(外気温、風速、日照、路面温度等)が影響するため、様々な状況での実証実験が必要となる。そのためには北海道における開発と実証実験が不可欠である。

研究実施機関：北海道大学、ヤマハ発動機、アイシン・コムクルーズ、エイ・ダブリュ・ソフトウェア、ヴィッツ、アーク・システム・ソリューションズ、アトリエ

実現可能性：本研究でベース技術として活用する自動運転車には、ヤマハ発動機が名古屋大学と実施した共同研究成果が使われているが、その開発成果はアイシン・コムクルーズがかかわったものである。この車両に搭載されている周囲状況・自己位置推定技術は SLAM によるもので、すでに果樹園等での実証実験を重ねており、不整地や山野路などのオフロード環境における走行可能領域の認識機能や障害物自動回避機能は実用レベルの性能にある。また北海道大学は、農業・林業用ロボット向けのロボスタな SLAM 技術の研究を実施しており、屋外自律運転に関する高度な知見を保有している。したがって、本研究チームにおいてはすでに自動運転のための基本技術は有しており、ここに北海道企業が加わることで、積雪による影響を十分に考慮した自動運転技術開発が実現できる。

開発成果物：積雪環境でも機能する周囲状況・自己位置推定機能(「Snow-SLAM」)

【2. 積雪環境の仮想シミュレータ開発】

今日の自動運転システムの信頼性・安全性検証では、仮想環境を用いたシミュレーションが欠かせないものとなっている。この理由としては、実車による屋外（特に公道）での自動運転試験走行はリスクが高いことに加えて、屋外環境では準備できる試験環境パターンが制限されることや、実車走行試験には膨大な時間と費用がかかることが挙げられる。

完全な自動運転を実現するために必要な試験走行距離は、150 億キロメートル（一般的な国内自動車メーカー）とも言われており、これを実車試験だけで行うのは困難である。実際、Google 社は、所有するデータセンターを活用して、1 日に約 480 万キロメートルにもおよぶ距離をシミュレーションにより「自動走行」させて、自動運転機能の性能を検証することができるとしている。

そこで本研究でも、冬季積雪環境の仮想シミュレータを新たに開発し、「Snow-SLAM」方式による自動運転の性能検証を、効率的に実施する。特に本研究の実施期間は最長 3 年であるため、冬季環境での実車試験は実質的に 2 回までしか行えない見込みである（最終年度の試験は確認程度の内容になることが予見されるため）。したがって、実車による性能評価のみでは不十分となる可能性が高く、仮想シミュレータによる評価が不可欠である。

研究実施機関：ヴィッツ、アーク・システム・ソリューションズ、アトリエ

実現可能性：本研究グループのヴィッツおよびアーク・システム・ソリューションズは、国内最大手自動車メーカーに自動運転仮想シミュレータを開発・提供しており、この分野で数少ない技術保有企業である。この仮想シミュレータは、降雨、太陽位置、反射、他車両の影や反射などを模擬する機能を備えている（図 1-1(5)）。このシミュレーション技術をベースとすることで、冬季積雪環境の仮想シミュレータを短期間に実現できる。



図 1-1 (5). 自動運転仮想シミュレータ画面

開発成果物：冬季積雪環境の仮想空間シミュレータ

【3. グローバル経路計画との連携機能開発】

上記のサブテーマ 1 で開発する「Snow-SLAM」方式は、車両の周囲状況を把握するものであり、その結果に基づいて、交通信号や交通標識の指示を遵守したり、他車両や障害物への衝突を回避したりするための経路生成が行われる（図 2 の【E】 ローカル経路計画機能）。これに加えて、自動運転車の実運用上は、利用者（もしくはサービス事業者）が所望の目的地を設定した上で、その目的地までの経路計画（グローバル経路計画）が可能である必要がある。

グローバル経路計画は、いわば既存の自動車のカーナビ製品が提供する機能であるが、自動運転車の場合、目的地および経路の設定は、運転者（または乗員）が行うだけでなく、サービス事業者が遠隔から自動的に行うことが想定される。したがって、柔軟なアプリケーションを可能とするためにも、自動運転車は、様々なグローバル経路計画を可能とするしくみを備えるべきである。

そこで、本研究では、グローバル経路計画情報を自動運転システムに取り込むための標準インターフェースを整備するとともに、その情報を Snow-SLAM によるローカル経路計画に反映させる機能を開発する（図 2 の【G】 カーナビ連携機能）。まずは、様々は利用形態を考慮して、必要となる標準インターフェース仕様を設計する。その後、既

存のカーナビ製品との連携を実現する。さらに、交通弱者への移動支援用アプリケーション開発を試行し、連携によるサービスの実証実験を行う。

なお本研究のベースとして活用するヤマハ発動機の自動運転車はオフロード用途のため、市街地走行を想定しておらず、カーナビゲーション等による経路計画を踏まえた走行経路指示には対応していない。そこで、本研究において新たに本連携機能を開発し、積雪環境下の市街地での自動運転サービスを実現する。

研究実施機関：アーク・システム・ソリューションズ、エイ・ダブリュ・ソフトウェア、ヤマハ発動機、北海道大学、ヴィッツ、アトリエ

実現可能性：本研究グループのエイ・ダブリュ・ソフトウェアは国内最大手自動車メーカ向けのカーナビゲーションシステムを開発するメーカであり、カーナビゲーションに関する高度な技術力を有している。

本サブテーマによるグローバル経路計画との連携機能は、カーナビ製品等による経路情報を自動運転車側の経路計画に反映させるものである。そのため、カーナビ製品等の経路探索結果を進路方向指示情報に変換して、車載ネットワーク上に送信させるなどの仕組みが必要となるが、これはエイ・ダブリュ・ソフトウェアの技術と知見に基づき、短期間に実現可能である。

③当初の研究開発の高度化目標及び技術的目標値

○高度化目標

(二) 情報処理に係る技術に関する事項

1 情報処理に係る技術において達成すべき高度化目標

(4) 川下分野特有の事項

3) その他の分野に関する事項

b. 自動車分野に関する事項

②高度化目標

ア. 自動車の高性能化・高機能化に関する技術の高度化

イ. 自動車の安全確保・信頼性向上に関する技術の高度化

ウ. 自動車の研究・開発・製造等の各種プロセスの生産性向上を実現するための当該技術の高度化

エ. 周辺の機器・システムとの接続に必要な技術の高度化

本研究は、自動運転に欠かせない周囲状況・自己位置把握技術を、積雪路面にも対応させるものであり、自動運転車の高性能化・高機能化に資するものである。また自動運転車は、その導入による交通事故の低減効果が期待されていることから、本研究は、冬季積雪環境での自動車交通の安全確保にも資する研究である。さらに、冬季積雪環境における新たな移動・輸送サービスが可能となることから、自動車を活用したサービス製品の高度化にも寄与する研究である。

合わせて本研究では、積雪環境に対応した仮想シミュレータを開発する。これは、自動運転技術の性能評価を効率化するものであり、自動運転技術の研究開発における生産性向上に資するものである。さらに、遠隔からの経路設定や進行制御のためのインターフェースを整備することで、多様なサービスアプリケーションシステムとの接続を可能とする。

【1. 積雪環境で機能する自己位置推定技術開発】

ダイナミックマップ方式を利用しない

「Snow-SLAM」方式により自己位置推定技術を向上させ、表1の自動運転性能目標を達成する。

低人口密度地域の場合、病院や商店、学校等の基礎的サービス施設への道路距離はおよそ30km以内と想定されることから、まずは、連続走行距離の目標値を30kmとする。

なお天候条件に関して、降雪状態では、吹雪等で視界がなくなる場合があり、本研究による技術の対応範囲外となるため、考慮対象から除外する。

表 1-1 (1). 「Snow-SLAM」方式による自動運転の性能目標

性能項目	目標値
連続走行距離	30km 以上
最高車速	時速 40km
路面	積雪 10cm の圧雪路
障害物	雪塊、吹き溜まりへの衝突回避率が 100% (ただし 1km あたり約 1 個の障害物を想定する)
天候	晴れまたは曇り

【2. 積雪環境の仮想シミュレータ開発】

上記サブテーマ1で開発する「Snow-SLAM」方式の性能目標を検証可能な自動運転シミュレータを構築するため、表2の環境を模倣できるシミュレーション技術を開発する。

なお表1-1(1)の性能目標は、本研究後に段階的に引き上げていくことが想定されるため、仮想シミュレータに対しては、本研究においてあらかじめ余裕をもたせた性能目標を掲げる。

表 1-1 (2). 積雪環境シミュレータの性能目標

性能項目	目標値
連続走行距離	50km 以上
最高車速	時速 50km
路面	積雪 20cm の圧雪路
障害物	1km あたり 1 個以上の障害物
天候	晴れまたは曇り

【3. グローバル経路計画との連携技術開発】

上記のサブテーマ1で開発する「Snow-SLAM」方式では、自車の周囲の経路算出のみが可能である(ローカル経路計画機能)ため、本サブテーマでグローバル経路計画との連携機能を開発し、一般道路での自動運転を実現する。

そのために、本連携機能では、waypoint(ウェイポイント)と呼ばれる途中通過点を外部から設定するためのオープンなAPIを設計して公開する。通過点の指定方法は、2種類以上(カーナビ製品によるものと、乗員の口頭によるオーバーライド等)を用意して、多様なサービスアプリケーションを実現可能なものとする。

④当初の目的及び目標に対しての実施結果

本事業では、3つのサブテーマごとに個別の実務を実施するためのタスクを設けて本事業を進めた。タスクは各サブテーマの年度目標を実現するための活動単位と位置づけ、3年間の研究開発期間で、のべ94タスクとなった。タスクの進捗は定期的に開催された「研究開発委員会」でタスクの実施内容や進捗状況、課題等が報告され、タスクに問題があれば都度アドバイザーより知見に基づいた解決方法などを教示いただきながら、最終的にはプロジェクトリーダー・サブリーダーの判断で対応方針を決定し問題解決図った。開発中に進捗遅れが発生したタスクにおいても、取組み内容や解決手法等を見直すなど対応し、最終年度までにはすべての活動について実施を完了した。以下に本事業で実施したタスクと、それぞれの実施結果を記す。

・実施タスクと実施結果

【サブテーマ1 2017年度】

No	タスク	タスク内容	タスク実施結果
1) 雪道対応 SLAM 技術の開発			
1	1-1) 既存の自律化プラットフォームの解析	既存の自律化プラットフォームの解析を実施する。	既存の自律化プラットフォーム（北大デモ）を解析完了
2	1-2) 雪道対応自律化プラットフォーム構想の検討	雪道に対応するための自律化プラットフォームの改良点を検討する。但し、2年度以降も継続して検討する。	来年度以降の自律化プラットフォームベースとなるシステム構成を検討完了 ⇒ROS の Navigation Stack をメインに構成する（来年度以降も継続）
3	1-3) SLAM 方式調査	初年度に検討する SLAM 方式を選定する。	初年度は、Lidar-SLAM 及び ODB-SLAM を選定とする
4	1-4) 雪道 SLAM 検証	雪道のデータを SLAM に取り込み、有用性を評価する。	雪道のデータを Lidar-SLAM 及び ODB-SLAM に取り込み初年度としての評価完了（来年度以降も継続）
5	1-5) 雪道対応自律化プラットフォームの試作	雪道に対応するための自律化プラットフォームのベースとなるシステムを構築する。	来年度以降の自律化プラットフォームベースとなるシステムを構築、ベースとなる車両プラットフォームとの I/F を検証済み（来年度以降も継続）
2) 雪道対応セマンティック・セグメンテーション技術の開発			
6	2-1) 識別クラス調査	識別クラスを調査し定義する。	識別クラスは 18 クラスで定義した。
7	2-2) フレームワークと実施環境の調査	フレームワークと実施環境の調査を行う。	機械学習用ライブラリ "Tensorflow" とニューラルネットワークライブラリ "Keras" の環境を構築し、サンプルプログラムの動作まで実施した。
8	2-3) データセット作成調査 (Label Me 調査)	データセット作成方法を調査する。	LabelMe の Web 版をローカル環境で構築できることがわかったので、ローカル環境で構築した。
9	2-4) データセット作成	データセットを作成する。	雪道動画に対して LabelMe を使用してデータセットを作成した。
10	2-5) 他の手法調査	他の手法調査を調査する	SegNet と FCN を調査し動作させた。
3) 雪道対応センサ・フュージョン技術の開発			
11	3-1) 初年度用センサの選定	初年度にデータを収集するセンサを選定する。	Lidar、GPS、ミリ波、赤外線、全方位カメラの選定（来年度以降も継続）
12	3-2) ROS 上でのデータ収集環境構築	ROS 上にデータを収集できる環境を構築する。	上記システムのデータを収集できる環境を構築
13	3-3) 収集データの検証とセンサ有用性評価	収集したデータが雪道対応することにより有用なデータか検証し、センサの取捨選択を行う。	位置情報は、GPS からより高精度な QZSS（準天頂衛星システム）も比較評価。 また、全方位カメラも高解像度（752×480⇒8K（7680×4320））を検討（来年度以降も継続）
4) 実証実験			
14	4-1) 実験ルール作成と備品準備	実証実験を安全に実施するための実験ルール作成と備品の準備を行う。	実証実験を安全に実施するための実験ルール作成
15	4-2) 実証実験計画作成	各回の実験計画を立案する	各回の実証実験計画書を作成
16	4-3) 実証実験報告書作成	各回の実験の報告書を作成する	各回の実証実験計画書を作成

【サブテーマ1 2018年度】

No	タスク	タスク内容	タスク実施結果
1) 雪道対応 SLAM 技術の開発			
17	1-1) LiDAR-SLAM 評価及び雪道環境認識	LiDAR-SLAM を実装し、実験車両での評価を実施する。さらに性能改善を検討する。	Cartographer (3D) による LiDAR-SLAM を採用。自己位置推定と環境地図作成に活用。障害物検知にも活用。

No	タスク	タスク内容	タスク実施結果
18	1-2) Visual-SLAM 評価 及び雪道環境認識	Visual-SLAM を実装し、実験車両での評価を実施する。さらに性能改善を検討する。	実装を完了して評価を実施。結果的に ORB 特徴量が少なく活用が困難であると判断。本プロジェクトでは他の手法で認識する手法を採用した。
19	1-3) サーモカメラ評価及び雪道環境認識	サーモカメラの採用を検討し、実験車両での評価を実施する。	サーモカメラにより轍（わだち）を検出し走行可能路検出機能を実現した
2) 雪道対応セマンティック・セグメンテーション技術の開発			
20	2-1) ROS 上でのセマンティック・セグメンテーション実現と実装	ROS 上でセマンティック・セグメンテーションを実現する。また知能化プラットフォーム渡すデータの形式について検討する。	ROS 上でセマンティック・セグメンテーションを実現し、知能化プラットフォームとの連携を実現した
21	2-2) 認識速度の向上検討	セマンティック・セグメンテーション認識速度向上を検討する（1 枚/秒→10 枚/秒）。	ICNet により、Full HD 画像で 25FPS 程度の速度が出ることを確認した
22	2-3) データセットの作り方再検討	たくさんのデータセットを作り上げるために反転/ホワイトノイズ等のデータの加工について検討をする。	反転/ホワイトノイズ等のデータの加工を実施した
23	2-4) データセット作成及び検証	データセットを作成する（800×18）枚	苫東の実地データを用いた学習セット、及び動画を使わずに作成した学習セット（動画を使うとどうしても似た画像になってしまうため）を作成した
3) 雪道対応センサ・フュージョン技術の開発			
24	3-1) 知能化プラットフォーム育成 (GPS 対応)	知能化プラットフォームを育成し、GPS 及び IMU にて自立走行する。	GPS 及び IMU にて自立走行を実現した
25	3-2) 知能化プラットフォーム育成 (SLAM or SS 対応)	知能化プラットフォームを育成し、GPS+SLAM 及び GPS+セマンティック・セグメンテーションで自立走行する。	GPS+Cartographer (3D)、GPS+セマンティック・セグメンテーションで自立走行を実現した
26	3-3) 知能化プラットフォーム育成 (fusion 検討)	知能化プラットフォームを育成し、センサ・フュージョンにより自立走行する。	セマンティック・セグメンテーション+LiDAR での道らしさ検出、サーモカメラ++LiDAR による轍検出を実現した
27	3-4) 知能化プラットフォーム育成 (性能評価と最適化)	雪道（苫東）での自立走行を実現し、性能評価及び最適化を実施する。	雪道（苫東）周回路コースでの自立走行を実現した
28	3-5) サブテーマ②（仮想シミュレータ）との連携	サブテーマ②（仮想シミュレータ）との連携を実現する。	シミュレータとの連携を実現し、実験車両を使う前に事前にシミュレータによる動作確認を実施した
29	3-6) サブテーマ③（グローバル経路計画）との連携	サブテーマ③（グローバル経路計画）との連携を実現する。	Waypoint の連携を実現した
4) 実証実験			
30	4-1) 実験ルール改善（安全考慮）	昨年度の実績より、自律走行を想定した実験ルールに改善する。	自律走行を伴う実証実験時の安全ルールを追加し改善を完了
31	4-2) 計画と報告書の作成	各回の実験計画の立案及び報告書を作成する。	各回作成を完了

【サブテーマ1 2019年度】

No	タスク	タスク内容	タスク実施結果
1) 雪道対応 SLAM 技術の開発			
32	1-1) LiDAR-SLAM 生成マップの活用設計と評価	LiDAR-SLAM で生成した環境地図の 2 次利用を検討し実装する。 また、現在使用している cartographer 以外にも Hector SLAM を検証する	一度取得したコストマップをグローバルマップへ対応可能とし、夏データ等がある場合は、より安定した走行が可能となりました。（完了）
33	1-2) 2 次データのコストマップ反映の最適化	2 次元データのコストマップへ反映の最適化を検討し、実装する。	点群補正（点群のないところを埋める処理）を実装（完了）

No	タスク	タスク内容	タスク実施結果
34	1-3) 自律走行制御改善 (IMU)	自律走行制御を改善する。ふらつきの補正を実施する	自己位置の異常値を除去するフィルタを実装 (完了)
35	1-4) 雪道での最終検証	すべてのシステムを結合して、雪道での最終検証を実施する	芦別・士別 (2/17-20) にて最終走行試験実施 (完了)
2) 雪道対応セマンティック・セグメンテーション技術の開発			
36	2-1) サーモカメラによる SS の検討	サーモカメラを用いてセマンティック・セグメンテーションを行う際の課題出し及び識別クラスの定義を検討する	サーモカメラによる輻検出を実装(完了)
37	2-2) サーモカメラによる SS 検証 (夏データ) と実装	サーモカメラの画像データでセマンティック・セグメンテーションを実現するために教師データの作成を行い、検証する	輻検出は、白黒画像に 2 値化し、直線を検出する処理で対応(完了)
38	2-3) サーモカメラによる SS 検証 (冬データ)	雪道でのサーモカメラによるセマンティック・セグメンテーションの検証を行う。	芦別・士別 (2/17-20) にて最終走行試験実施 (完了)
3) 雪道対応センサ・フュージョン技術の開発			
39	3-1) 調停システムの検討・設計	各センサーからの情報を調停する形式を検討する。	各センサーと調停システムは ID マップ (独自のコストマップ) を I/F とする(完了)
40	3-2) 調停システム実装	ID マップの重ね合わせ及びコストマップのコードを実装する。	実装済み(完了)
41	3-3) 調停システム評価と最適化	各センサーからの ID マップ繋ぎこみとコスト値の調整をする。	芦別・士別 (2/17-20) にて最終走行試験実施 (完了)
4) 実証実験			
42	4-1) 計画と報告書の作成	各回の実験計画の立案及び報告書を作成する。	各実証実験の計画書と報告書作成実施。芦別・士別 (2/17-20) の報告書をまとめて完了

【サブテーマ2 2017年度】

No	タスク	タスク内容	タスク実施結果
43	1-1) シミュレーション環境の調査	候補とするシミュレーション環境を調査し、使用する環境を選定する。	完了。 シミュレーション環境は UE4 で実装する、但し Gazebo を使用する検討も続行。
44	1-2) シミュレータ構成の検討	シミュレータと ROS の接続、入出力データについて検討する。	完了。
45	1-3) センサモデルの検討	作成するセンサモデルの種類、仕様を検討する。	完了。
46	1-4) テストコースの検討	積雪環境 (テストコース) の仕様を検討する	完了。同時にテストコース作成手法も検討したが、データが揃わなかった為、検討のみとなった。
47	2-1) シミュレータ環境の構築	シミュレーション環境を実装する。	完了。また、テスト用に ROS 上のプログラムを作成。
48	2-2) テストコースの実装	積雪環境 (テストコース) を実装する	完了。計画では実証実験コースを実際に走行し、センサで取得したデータを基に作成する予定であったが、作成方法を変更して実施した。
49	2-2) センサモデルの実装	センサモデルを実装する	完了。今年度に作成を行う予定のセンサモデルは全て実装済み。
50	3-1) 動作検証項目の作成	動作検証項目を検討し、仕様書を作成する	完了。
51	3-2) 動作検証の実施	動作検証を実施し、報告書を作成する	完了。

【サブテーマ2 2018年度】

No	タスク	タスク内容	タスク実施結果
1) センサーの追加開発			
52	1-1) LiDAR Velodyne HDL-32e	2017年度に試作したシミュレーション環境では、当初使用していた「VLP-16」をモデル化。新たに「HDL-32e」モデルを追加する。	シミュレータ側 (UE4) のモデル作成、及び ROS 側の入出力部を開発した。オープンループ状態で、rviz 上で動作確認を実施し完了した。
53	1-2) 全方位マルチカメラ Ladybug	全方位マルチカメラをモデル化する。	シミュレータ側 (UE4) のモデル作成、及び ROS 側の入出力部を開発した。オープンループ状態で、rviz 上で動作確認を実施し完了した。
54	1-3) 赤外線サーモグラフィ optris PI450	赤外線サーモグラフィをモデル化する。	シミュレータ側 (UE4) のモデル作成、及び ROS 側の入出力部を開発した。rviz 上で動作確認を実施し、完了した。
2) シミュレーションの改良			
55	2-1) 知能化 PF との結合①	知能化 PF との結合に向けた準備として、シミュレーション用の ROS のノードを作成 (CAN 通信、センサ)。基本的には実機用のノードとの入れ替えだけでシミュレーションで利用できる様にする。	CAN、LiDAR、ミリ波、GPS、IMU のシミュレーション用ノードの作成を完了。オープンループでの確認を実施し、完了した。
56	2-2) 知能化 PF との結合②	知能化 PF の入手後に、CAN、各種センサのドライバに当たる部分をシミュレーション用のノードに入れ替える。I/F 整合しない部分があれば修正する。	サブテーマ①より知能化 PF を入手し、2-1) で作成した環境への組み込みを実施。クローズドループでのシミュレーションにて動作確認を完了した。マニュアルも作成済み。
57	2-3) 仮想環境改良 ① 道路幅の拡張 ② 障害物の設置 ③ コース追加 ④ 道路脇の雪壁 (仕様のみ) ⑤ 芦別コース追加 (検討)	シミュレータの改良を行う。適宜シミュレータ使用中に見つかった問題があれば優先度の高いものから対応する。 ① 2017年度に実証実験コースをモデル化したが道路幅が実際よりも狭い。実際に道路幅を測り合わせる。 ② 障害物を設置できる様にする。 ③ 10月デモ披露コースをシミュレーションでも実施したい為、コースを追加する。 ④ 昨年、苫東周回路の実証実験の際、除雪車を入れたが、その時雪壁ができていた。再現する。 ⑤ サブテーマ①より、芦別のコースが欲しいとの要望有り。今年度は検討まで行う。	① 一見狭く見えたが実環境に近い。対応不要と判断し完了とした ② 障害物設置の仕様を作成、シミュレータへの実装を行い完了した ③ 10月のデモ披露が中止となった為、対応不要となり終了。 ④ シミュレータ仕様を策定し完了。 ⑤ 芦別のオーバルコース仕様を策定して完了
58	2-4) 積雪環境シミュレータに求める性能まとめ	積雪シミュレータに求める性能を、いくつかの視点からまとめる。	下記に関してシミュレータとして必要な性能を整理してまとめた。 ・認識系 (センサ種類、ノイズ等) ・挙動系 (摩擦、抵抗、雪粒子) ・連続試験の実施方法 (実行時の変動パラメータ、記録データ等)

【サブテーマ2 2019年度】

No	タスク	タスク内容	タスク実施結果
1) 芦別コースの追加			

No	タスク	タスク内容	タスク実施結果
1) 芦別コースの追加			
59	1-1) 芦別コース仕様検討	新規に作成する芦別コースの仕様を検討する。	芦別コースの仕様を配置すべきオブジェクト、積雪時の状態等を検討した。積雪時にコース上に設置されている杭も再現。
60	1-2) 実装 (芦別コースモデル)	芦別コースのモデルを作成する。また、シミュレータに組み込み設定画面に追加する。	芦別コースのモデルを作成した。オーバルだけでなく、多目的コースも使用する走行ルートが6月以降に決まったが、その点についても反映した。ただし、シミュレータ単体での自動走行は実装しない。
61	1-3) 動作検証	芦別コースにバイキングを走らせ、動作検証を実施する。 ・作成したモデルの位置、大きさ ・各種センサの反応 など。	完了。 但し、入手した図面と実際のコースの形状が異なっていた事がわかったが、時間の関係上、入手した図面のままで実装済み。
2) シナリオ実行の追加			
62	2-1) シナリオ実行の仕様検討	シナリオ機能実現方法の検討を実施する。また、シナリオ実行した時に走行結果を記録する方法、記録が必要なデータの検討も実施する。	シナリオ機能実現方法の検討を実施した。UE4 の blueprint にてシナリオ実行を実現する。
63	2-2) シナリオ内容の検討	今回実装するシナリオ (ユースケース) を検討する。ひとまず、重要そうな数パターン分の検討を実施する。	今回は、1周走行する毎にコース上の障害物の位置を配置しなおすシナリオとした。30周分を用意した。
64	2-3) 実装 (シミュレータ側)	UE4 に対し、シナリオ実行の機能を実装する。	1周毎に異なる障害物の配置となるシナリオを実装した。
65	2-4) 実装 (ROS 側)	シナリオ実行時に、ROS 側で実施すべき処理を実装する。 ・解析に必要なデータの保存 ・1つのシナリオ終了後の初期化 など。	シミュレータ側から、rosviz ファイルの保存をコントロールする方法で実装を行った。指示するのは、①記録開始、②記録終了の2種類として実装を完了した
66	2-4) 動作検証	シナリオ実行が意図した通りに動作しているかを動作検証する。	シナリオが意図したとおりに切り替わる事を確認し動作検証を完了した
3) シミュレータ改善			
67	3-1-1) 雪壁仕様検討	除雪した状態を再現する。道路の圧雪状態、道路わきの雪壁をどの様に実現するか検討する。雪壁の高さは3段階程度を想定する。	検討した結果、今回対応するのは、ラウンドコースと芦別コースのみと決定し仕様検討を完了した。
68	3-1-2) 雪壁実装	除雪した状態を再現する機能を UE4 側に実装する。	苦東コース、及び芦別コースでの実装を完了した。
69	3-1-3) 雪壁動作検証	除雪した状態が意図した通りに実装できているかを動作検証する。 ・各種センサの反応 など。	Lidar、RGB カメラ、サーモにおけるシミュレータ動作検証を完了した。
70	3-2) カメラセンサ改善策検討	複数カメラ系センサの同時出力に対応する	実現方法を検討した結果、①RGB カメラシミュレーションとサーモカメラシミュレーションの同期実行、②Ethernet へ変更することで同時出力を実現した。

【サブテーマ3 2017年度】

No	タスク	タスク内容	タスク実施結果
----	-----	-------	---------

No	タスク	タスク内容	タスク実施結果
71	1-1) グローバル経路計画システムの調査	調査を実施し報告書を作成する	オープンソースの Map アプリ(サービス)の機能の比較調査完了。 ROS の通信連携機能の調査完了。
72	1-2) グローバル経路計画連携方法の検討	システム仕様書	連携方法を策定し、システム仕様書の作成完了。
73	1-3) WayPoint 設定 API・I/F 仕様策定	I/F 仕様書	グローバル経路連携に提供する API 仕様書の作成完了。
74	2-1) グローバル経路計画連携機能の設計	詳細設計書	グローバル経路連携機能の詳細設計書の作成完了。
75	2-2) グローバル経路計画連携機能の実装	ソースコード	グローバル経路連携機能の実装を完了
76	2-3) グローバル経路計画連携機能の単体テスト	評価仕様書、評価結果報告書	単体試験項目の作成、実施完了。
77	2-4) グローバル経路計画連携機能のシステム結合	評価仕様書、評価結果報告書	未実施(サブテーマ①との IF 部分が未確定のため)
78	3-1) 実証実験	実証実験報告書	実証実験 3 回目の結果報告書を作成する。

【サブテーマ3 2018年度】

No	タスク	タスク内容	タスク実施結果
79	1-1) グローバルプランナー開発	Waypoint 情報から ROS の MAP 座標への変換処理を開発する。	グローバルプランナーの実装完了。 TurtolBot のシミュレータで検証済み。
80	1-2) RosBag のマッピングデータ作成・適用	RosBag から MAP を作成して、OpenStreetMap の地点情報に適用する。作成した MAP が使用に適していない場合は 6 月に再度収集して作成する。	昨年度の RosBag からマッピングデータを試作済。 7 月収集データから直線路のデータを作成。周回路は GPS がずれているため利用不可と判断した。
81	1-3) グローバル経路から収集可能なデータの調査	OpenStreetMap から収集可能なデータの精査及び、使用可能な情報を調査する。	調査完了。サブテーマ①で使用したいパラメータ(道路幅、車線情報等)は OpenStreetMap で設定可能だが、ほとんど使用されていないため Waypoint へ情報の追加は不可と判断した。
82	2-1) 既存の自動走行の経路制御処理の解析	既存システム(Autoware, YAMAHA の ROS ソース)から経路制御処理の解析を行う。	ROS の経路生成方法の調査完了。その他の経路補正手法(クロソイド曲線、スプライン補間等)の調査完了。報告済み。
83	2-2) グローバルプランナー調整機能の仕様検討	グローバルプランナー調整機能の仕様を検討する。	経路補正アルゴリズムとして Spline 関数を用いて、Waypoint に沿ったスムーズな曲線を算出し、各地点が一定以上の角度とならないよう補正地点を追加する方針とした。
84	2-3) グローバルプランナー調整機能の実装	グローバルプランナー調整機能を開発する。	Waypoint から Spline 関数による補正地点を算出する処理の実装を完了。 各地点の距離から補正地点の挿入/非挿入の判定を行う処理を追加対応した。
85	2-4) 雪道走行での速度調整の対応	グローバルプランナー調整機能で、雪道走行における Waypoint の速度調整について検討、対応する。	雪道走行の速度調整に対応し完了
86	3-1) 実証実験	グローバル経路計画機能とローカル経路計画機能の連携に関する実証実験を行う。	サブテーマ①と I/F 仕様について合意し、連携確認は実施済み。 Waypoint の補正処理についてサブテーマ①、②の連携環境で検証を実施し完了。

【サブテーマ3 2019年度】

No	タスク	タスク内容	タスク実施結果
87	1-1) グローバル連携機能サーバ環境構築	移動支援アプリからの通信による地図表示およびWaypoint情報を取得するための外部サーバを構築する。	OpenStreetMap、ルート検索APIのサーバ環境は構築済み。 外部から独自データを適用した地図を取得・表示することが可能となった。
88	1-2) グローバル連携機能ミドルウェアのライブラリ化	クライアント用(アプリ側)ミドルウェアをAndroidアプリから参照可能なライブラリを作成する。 Android Javaから使用可能なインタフェースを作成する。	JavaScriptをAndroidJavaクラスでラップし、Androidアプリから利用可能なライブラリの作成を完了した
89	1-3) 移動支援アプリ作成	外部サーバと通信して地図の表示、ルート検索とWaypointの取得、および車両との通信を行うアプリケーションを作成する。	1-2のライブラリを内包したAndroidアプリケーションの作成を完了した
90	2-1) 車両連携拡張機能の検討	車両情報をユーザに通知するためのUIを追加する。	UIに表示する車両ステータスの情報仕様を策定しサブテーマ①と連携して下記の機能追加を実現した。 ・走行中/停車中ステータス表示 ・ステアリング角度表示 ・各センサーの稼働状況表示
91	2-2) 車両連携拡張機能の設計	検討内容を元に車両連携に必要な機能を設計する。	車両ステータスの情報をアプリケーション上に表示するためのUI設計、ソフトウェア変更を行い、車両連携拡張機能を実現した。
92	2-3) 車両連携拡張機能の実装	検討内容を元に車両連携に必要な機能を実装する。	移動支援アプリのUIに車両ステータスを表示する項目、ユーザにセンサー稼働状況を通知するダイアログ表示機能を実現した。また、実証実験用に、ルート検索した結果のWaypointに対して車両の走行速度を手動で設定変更する機能を実現した。
93	2-4) 車両連携拡張機能の動作検証	車両連携のユーザ通知機能の動作検証を実施する	知能化PFと移動支援アプリの初回接続時にセンサー全ての情報を表示し、以降センサーの状態が変化した際にダイアログでユーザ通知するよう実装。知能化PFとの結合確認を実施して対応を完了。
94	3-1) 実証実験	実車にて移動支援アプリ(車両連携拡張機能を含む)と知能化PFとの実証検証を行う。	実証実験場(苫東、芦別)にて移動支援アプリによる経路情報に従い車両が自動走行することを実証した。

1-2 研究体制

(研究組織・管理体制、研究者氏名、協力者)

本事業の研究体制および役割は図1-2の通り。定期開催の研究開発委員会にはオブザーバとして北海道経済産業局を迎え、進捗状況や課題を報告するなど本事業を適正かつ確実に実施する体制とした。

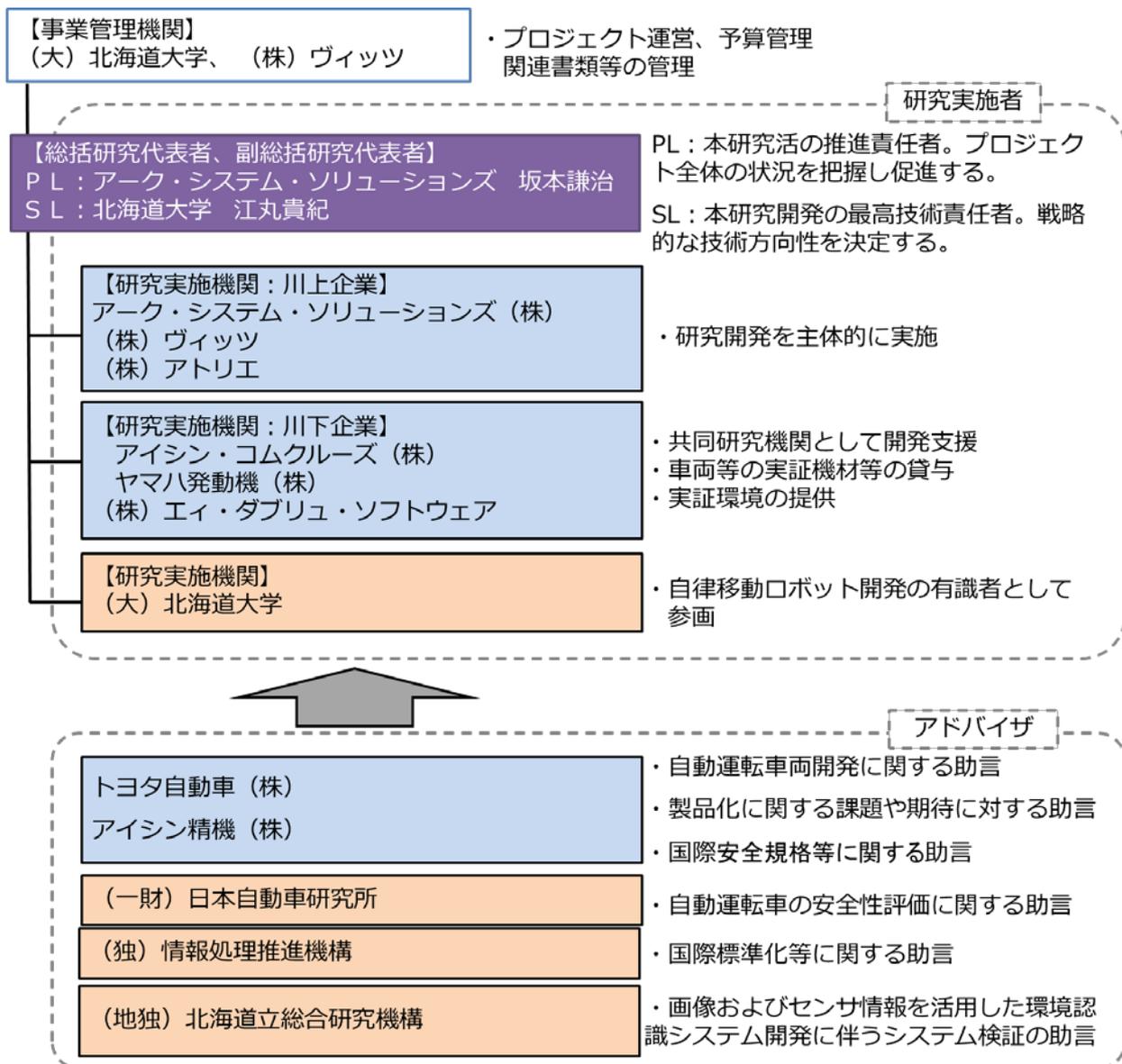


図 1-2. 本事業体制図と役割

1-3 成果概要

【1. 積雪環境で機能する自己位置推定技術開発】

ダイナミックマップ方式を利用しない「Snow-SLAM」方式により自己位置推定技術を向上させ表 1-3(1)の自動運転性能目標を達成した。なお天候条件に関して、降雪状態では、吹雪等で視界がなくなる場合があり、本研究による技術の対応範囲外としていたが、現実には冬季に実証実験時の降雪は免れない状況であったため、降雪によるセンサーへのノイズを除去する機能(積雪対応機能)を実現した。



図 1-3(1). 距離 30km 自動走行試験時の画像

表 1-3(1). 「Snow-SLAM」方式による自動運転の性能目標と成果

性能項目	目標値	成果
連続走行距離	30km 以上	30km 以上の連続距離の自動走行を達成した。途中激しい降雪があり中断したものの合計で 60km 以上の距離を自動走行した。
最高車速	時速 40km	設計速度 40km/h で機能開発を完了した。冬季実証実験では安全上の理由で最高速度を 30km/h に制限した（路面状況とタイヤ性能を考慮）ため、積雪路で時速 40km/h の実走行を検証することができなかった。なお、シミュレータ上では 40km/h での走行検証を実施済みである。
路面	積雪 10cm の圧雪路	積雪 10cm の圧雪路での走行を実現した。性能目標検証は圧雪路で実施したものである。
障害物	雪塊、吹き溜まりへの衝突回避率が 100%（ただし 1km あたり約 1 個の障害物を想定する）	人工的に積み上げた雪を、1km あたり 1 つ設置して実証した結果、100%の障害物回避を達成した。
天候	晴れまたは曇り	晴れまたは曇りの状況で、連続走行距離、路面、障害物の各目標を達成した。

【2. 積雪環境の仮想シミュレータ開発】

上記サブテーマ1で開発する「Snow-SLAM」方式の性能目標が検証可能な自動運転シミュレータを構築するため、表 1-3(2)の環境を模倣できるシミュレーション技術を開発した。なお表 1-3(1)の性能目標は、本研究後に段階的に引き上げていくことを想定しており、仮想シミュレータに対しては、本研究においてあらかじめ余裕をもたせた性能目標を掲げた。いずれの性能目標について本研究の目標値を達成した。



図 1-3(2). 仮想シミュレータ（LIDAR 模擬時）

表 1-3(2). 積雪環境シミュレータの性能目標

性能項目	目標値	成果
連続走行距離	50km 以上	連続走行距離無制限を実現
最高車速	時速 50km	時速 50km/h 以上を実現
路面	積雪 20cm の圧雪路	積雪 0cm、20cm、30cm、50cm を実現
障害物	1km あたり 1 個以上の障害物	障害物の設定個数無制限を実現
天候	晴れまたは曇り	晴れ、曇り、降雪に対応

【3. グローバル経路計画との連携技術開発】

上記のサブテーマ1で開発した「Snow-SLAM」方式では、自車の周囲の経路算出のみが可能である（ローカル経路計画機能）ため、本サブテーマでグローバル経路計画との連携機能を開発し、一般道での自動運転を実現した。本連携機能では、waypoint（ウェイポイント）と呼ばれる途中通過点を外部から設定するための API を開発した結果、



図 1-3(3). 冬季の一般道を走行する自動走行車両

一般公開することが可能となった。通過点の指定方法は、独自の Android アプリで出発地と到着地を地図上でタップする方法である。この操作によりシステムが経路を検索(ナビ)し自動で waypoint を設定することが可能となった。これによりサブテーマ1で開発した「Snow-SLAM」方式と連携し、一般道での自動走行を実現した。一方で乗員の口頭によるオーバーライド機能については、アプリ上でAndroidの音声認識を利用すれば実現可能ではあったものの、他のタスクを優先としたため本研究期間で対応することができなかった。

1-4 当該研究開発の連絡窓口

総括研究代表者(PL)

氏名:坂本謙治

事業者名・所属部署名:アーク・システム・ソリューションズ株式会社

所属役職:代表取締役

Tel:011-207-6460 Fax:020-4622-5064

E-mail:sakamoto_kenji@arcsys-sol.co.jp

副総括研究代表者(SL)

氏名:江丸 貴紀

事業者名・所属部署名:北海道大学 大学院工学研究院

機械・宇宙航空工学部門

所属役職:大学院工学研究院 准教授

Tel:011-706-6409

E-mail:emaru@eng.hokudai.ac.jp

第2章 具体的な実施内容と成果

2-1. 【サブテーマ1-1】積雪環境で機能する自己位置推定技術開発

積雪環境におけるダイナミックマップ方式の課題を克服するためには、あらかじめ作成された立体地図情報を用いることなく周囲状況と自己位置を把握できる技術が必要である。これを実現するために、本研究では、SLAM、セマンティック・セグメンテーション、および、センサ・フュージョンという3つの技術を雪道に対応させ、組合せて用いることで

「Snow-SLAM」を実現する。本サブテーマは、「Snow-SLAM」を実現するため4つのサブサブテーマを設定して開発を進めた。初年度は、「積雪環境に対応した自動走行車の試作、積雪環境での各種データ収集」を目標に、積雪環境での各種データ収集のためのセンサ選定、データ収集ための環境構築、データ収集のための実験計画などを実施し、積雪環境下での各種データをデータ収集することできた。収集したデータは、SLAM やセマンティック・セグメンテーションに取り込み、有用性を評価した。また、自動走行車に関しては、本プロジェクトで使用するヤマハ Viking VI を制御するための知能化プラットフォームのベースを試作し、来年度以降の自動走行におけるプラットフォームを作成した。

2年度は、「収集データに基づく各技術の最適化、および、自動運転機能の性能評価」を目標に、積雪環境での各種収集データを、SLAM やセマンティック・セグメンテーションに取り込み、有用性を評価した。また、自動走行車(ヤマハ Viking VI)に実際に搭載し、自動運転機能の性能評価を実施した。

最終年度は、「センサ類のコストダウン検討」を目標に、自動運転性能目標を達成のためのセンサ・フュージョンの最適化に取り組み、そのなかでセンサ類のコストダウンの検討を実施した。

2-1-1. 雪道対応 SLAM 技術開発

2-1-1-1(1) 初年度の目標

雪道対応 SLAM 技術開発について 29 年度は、以下目標に向けて活動を実施した。

・29 年度の目標

1. SLAM 方式調査
2. 雪道 SLAM 検証
3. 既存の知能化プラットフォームの解析
4. 雪道対応知能化プラットフォーム構想の検討
5. 雪道対応知能化プラットフォームの実装

29 年度に検討する SLAM 方式の選定後、選定した SLAM に雪道のデータを取り込み、有用性を評価する。尚、取り込む雪道のデータは「3-1-4.」の実証実験にて取得したデータを用いた。並行し、29 年度は雪道 SLAM 技術開発のプラットフォームの構想および、実装を開始した。具体的には、まず既存の知能化プラットフォームの解析を実施し、システム構成を把握後、雪道対応知能化プラットフォーム構想の検討を実施する。その後、検討結果に基づき雪道対応知能化プラットフォームの実装を行った。

2-1-1-1(2) 具体的な内容

【1. SLAM 方式調査(29 年度実施)】

初期の SLAM は、検出された特徴点(ランドマーク)の位置情報をすべてカルマンフィルタの状態量に組み込み、ロボット

への制御入力と観測されたランドマークの位置、そして IMU (慣性センサ) 等の値を統合し自己位置推定を行う。これはランドマークを観測するたびにカルマンフィルタの状態量が増え続けることを意味し、移動を続ければ計算コストが次第に増大しリアルタイムでの実行がこんなになるという問題点があった。この初期世代の SLAM には以下のような方法がある。

EKF-SLAM: 拡張カルマンフィルタを使用

Fast-SLAM: 自己位置推定にパーティクルフィルタ、ランドマーク処理に EKF を用いる事で計算を工夫

ランドマーク観測のセンサとしては大きくカメラと LRF (LiDAR 等) があり、カメラを用いるものを Visual SLAM、LRF の中でも特に LiDAR を用いた SLAM を LiDAR-SLAM と呼ぶ。画像情報を使うと特徴点のマッチングを行いやすいというメリットがあるが、デプス (Depth, 奥行き情報) を推定するための計算が別途必要であり、LRF では特徴点マッチングが難しい代わりに Depth が直接観測できる。そこで、まず LiDAR-SLAM と Visual SLAM から代表的な手法を1つずつ選択し、積雪環境において得られたデータに適用することで各手法の特徴を明らかにすることを目的とした。

Visual SLAM の調査結果についてまずまとめる。初期の Visual SLAM の集大成として単眼カメラを用いたリアルタイムで実行可能な Mono-SLAM が 2003 年に発表され、以降、単眼 Visual-SLAM の流れが生まれた。Visual SLAM の変遷を以下にまとめる。

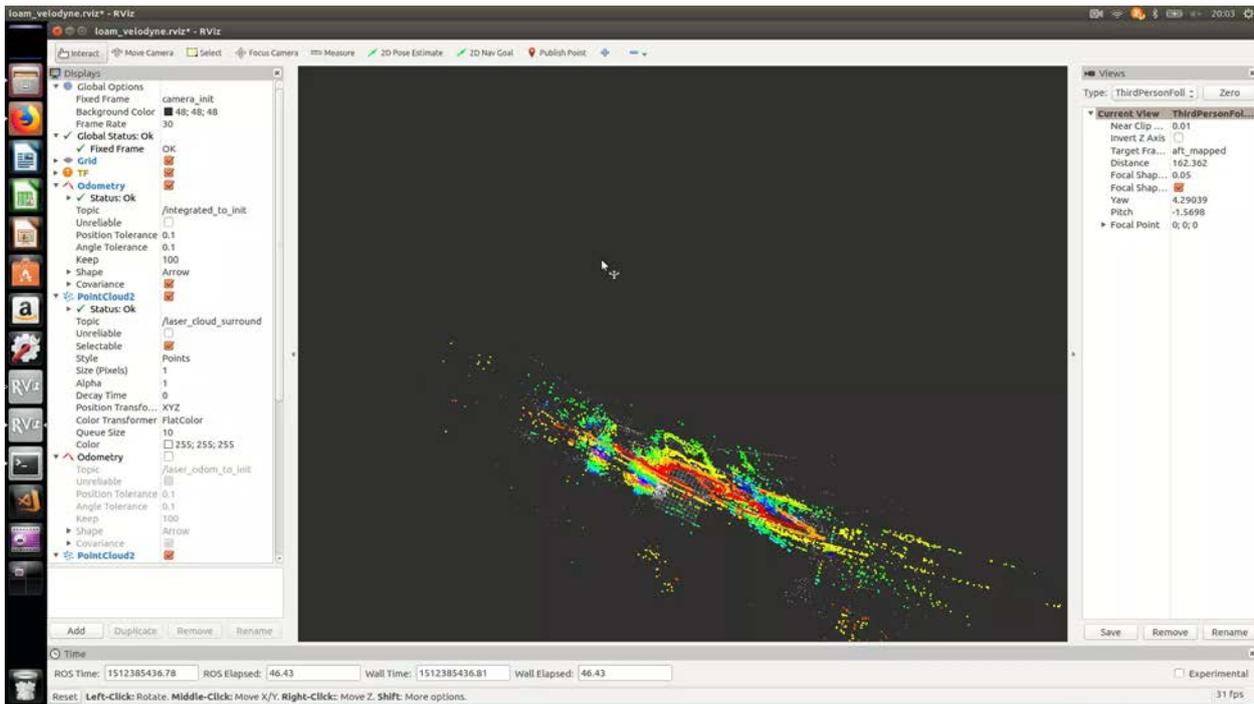
1. PTAM: parallel tracking and mapping (2007 年に提案)。EKF の計算コストの問題を解決し、マーカーレス AR の実現に貢献した。Pokemon Go にも同様の技術を使っている。特徴的な点に着目する特徴量ベースの手法の一種であり、動画のすべてのフレームをマップに反映させることをあきらめ、連続するフレームの特徴的な部分に対応点として取ることを毎フレーム行い、見え方が大きく変わったときのフレーム (= キーフレーム) のみに関してマップに反映する。PTAM はこの自己位置推定とマップ推定を並列・独立して行うことが最大の特徴である。自己位置推定は連続するフレームの対応点を取って毎フレーム行い、視界が大きく変わったフレームのみに関して三角法でマップに投影する。
2. DTAM: 2011 年。フレームの特徴的な点を取るのではなく、フレーム内の全てのピクセル輝度について対応をとる。輝度を直接用いるため direct な手法と呼ばれる。広範囲なマップを扱うことができず机の上程度の範囲しか SLAM ができない。
3. SVO: Semi-direct Visual Odometry。これも direct な手法。特徴点を一部使用することから完全に direct なわけではなく、また、loop close 検出も持たないため、DTAM と同様広範囲の SLAM は不可能である。
4. LSD-SLAM: Large Scale Direct SLAM (2014 年)。これも direct だが、loop close 検出を持つ。これによって大規模で実行できる初めての direct な SLAM となり、当時の最新の特徴量ベースの SLAM の精度を超えていた。
5. ORB-SLAM: 2015 年。PTAM 以降発展した様々な特徴ベースの SLAM の周辺技術。また、局所的にキーフレームを多数取り込むことで精度とロバスト性を向上させ、再び特徴ベースの SLAM が direct な SLAM の精度を凌駕した。
6. DSO: Direct Sparse Odometry (2017 年 3 月) LSD-SLAM の流れを汲む。カメラごとに最適化することで ORB と同程度の精度達成。Direct な手法の1つである。

また、別の流れとして SfM (Structure from Motion) がある。これは時間的にバラバラな画像群を繋ぎ合わせて 3 次元地図を作り出す手法である。地図を作るのに時間的な制約はない。すなわちリアルタイム性を犠牲にして精度を追求する手法であるため、リアルタイム性が必須である自動運転の用途には向かない。

これらの調査を踏まえ、初年度は Visual SLAM として特徴量ベースの SLAM に着目し、ORB-SLAM の適用を試みた。また、LiDAR-SLAM として現在最も一般的に用いられている Loam Velodyne を適用することにした。

【2. 雪道 SLAM 検証(29 年度実施)】

以下に示す図は、Loam_velodyne による SLAM の結果である。録画してフレーム落ちしている可能性もあるが、実証実験時に確認した際にはフレーム落ちは確認できていない。最後に生成した点群もおかしなことに外接弧の形になっている。実際走っている実証実験のフィールドには多少の凸凹はあるが、なぜきれいなマップができないかという点については考察が必要である。もっとも大きな要因として、LiDAR-SLAM を適用するために十分な形状的特徴量が環境に内在しないのではないかという点が考えられる。なお、Ford 社が実施した積雪環境における実験の知見として、雪道の難点の一つは粉雪に邪魔されて LiDAR が環境の点群に影響するとあったため、この点については来年度以降検討が必要であろう。



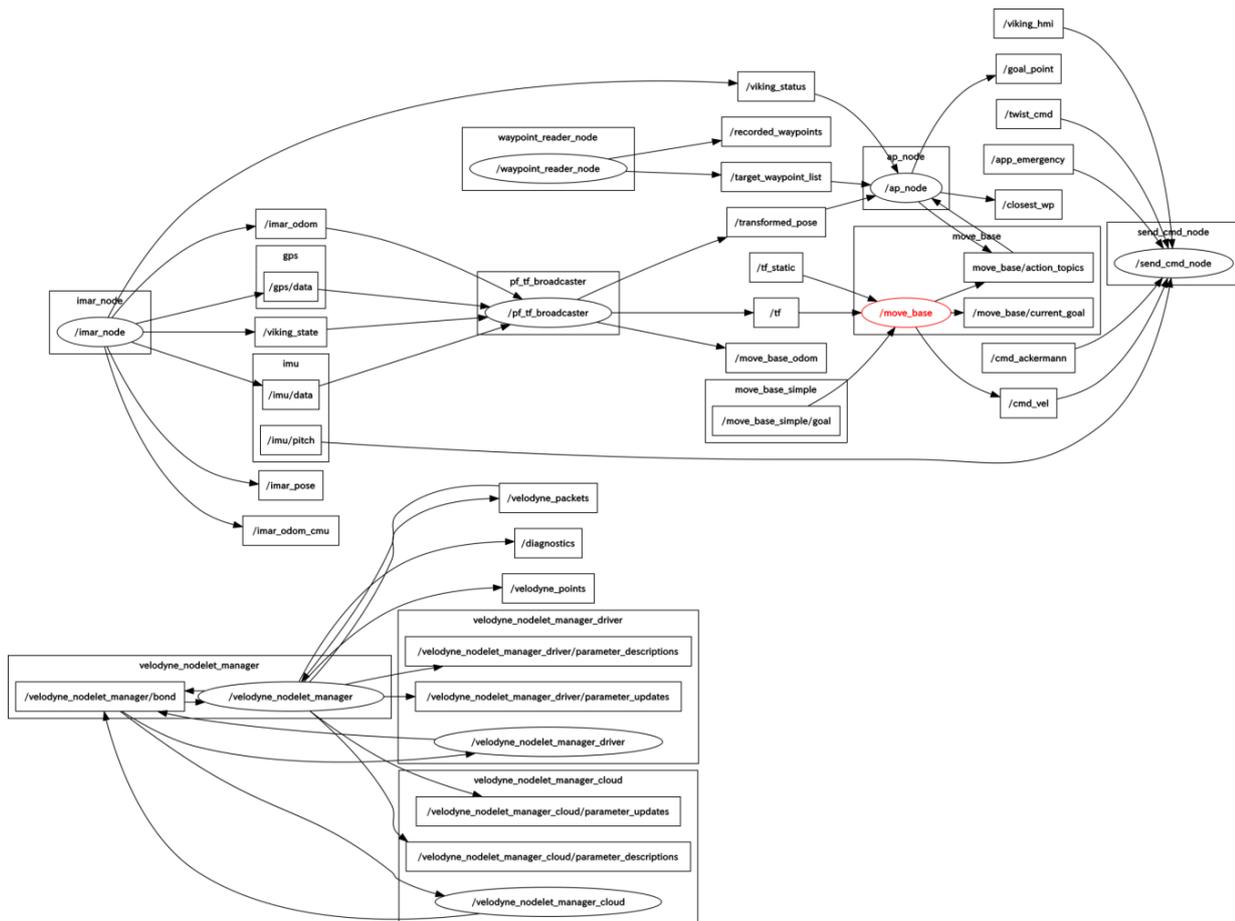
【3. 既存の知能化プラットフォームの解析(29 年度実施)】

本研究でベースとして活用する自動走行車(ヤマハ Viking VI)に搭載されている自律化プラットフォームのコードの一部をサンプルコードとして提供いただいている。既存の自律化プラットフォームの解析を以下手順で実施する。

1. サンプルコードをベースに必要なノートとトピックをダミーで構成する。
2. ROS のツール「rqt_graph」にてシステム構成を可視化する。
3. 可視化したシステム構成より、ノード・トピックの関係を理解する。

システム構成を可視化する ROS のツール「rqt_graph」を実行し、解析の足掛かりとする。尚、「rqt_graph」を使用するためにはシステムが動作している必要がある為、最初の手順としてソースのビルドを実施する。

次に「rqt_graph」によりシステム構成を可視化し各ノード・トピックからデータの流れを把握した上で、ソースコードの解析を進めていく。以下に解析結果として「rqt_graph」にて可視化したシステム構成を示す。



処理の大きな流れとしては以下の通り。

1. 車両に搭載した GPS から現在位置を取得する。
2. テキストから WayPoint を取得する。
3. 現在位置と WayPoint を比較し、テキストに記載された WayPoint を順々に Navigation Stack(※)が速度・ヨーレートを車両運動制御機能に CAN 送信する。

※ROS が用意している「ロボットの自動運転を実現する」枠組み。

現在位置、センサ情報、ゴール値を入力することで、自動運転が可能。

ゴール位置を指定すると周辺状況をセンサ情報から判断し、

現在位置からゴール位置への経路を生成し動作する。

各ノードの処理内容は以下の通り。

ノード	処理
immar_node	車両情報、imu データ、GPS データ等を Publish する。
pf_tf_broadcaster	immar_node からのデータを Subscribe し、GPS データを変換する。
waypoint_reader_node	テキストファイルから waypoint を読み込み、Publish する。

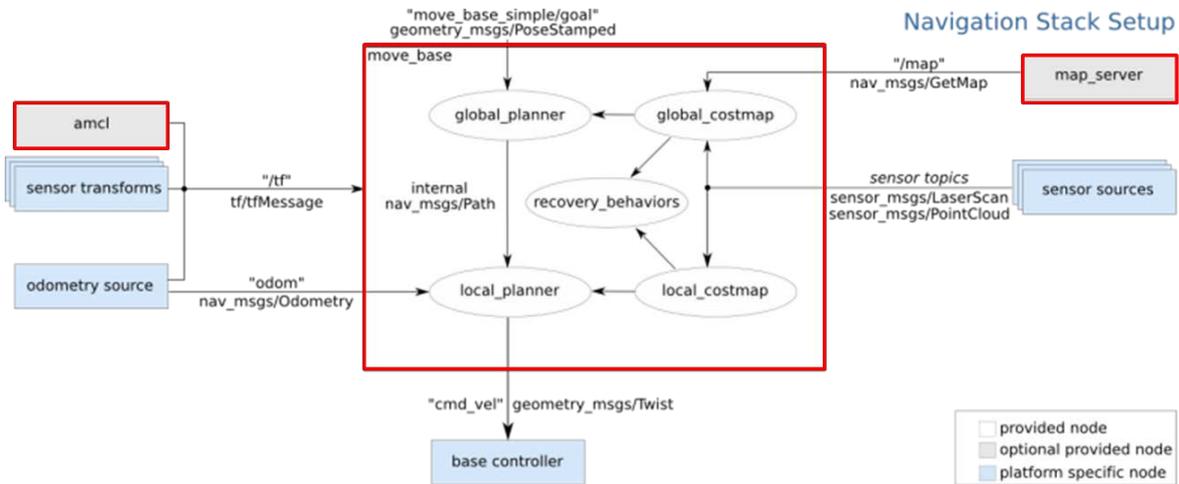
ap_node	waypoint、GPS データを Subscribe し、move_base に渡すゴール値を生成する。
move_base	Navigation Stack のナビゲーション機能を持つ。センサ情報、ゴール値、地図データから、CAN 送信を行う send_cmd_node に渡す速度とヨーレートを Publish する。
send_cmd_node	車両に CAN 送信を行うノード。速度、ヨーレート、車両制御モードを CAN 送信する。

【4. 雪道対応智能化プラットフォーム構想の検討(29 年度実施)】

雪道対応自律化プラットフォーム構想の検討を以下手順で実施する。

1. Navigation Stack をメインとした構成の検討
2. 「1. 」に SLAM を使用した構成の検討
3. 「2. 」の結果より、雪道対応自律化プラットフォーム構想を検討する。

まず、Navigation Stack をメインとしたプラットフォームを検討する。Navigation Stack は以下図の赤枠で囲った、「move_base」「amcl」「map_server」の 3 つのソフトウェアで構成されている。



move_base は「コストマップ管理機能」、「経路計画機能」を持つ。

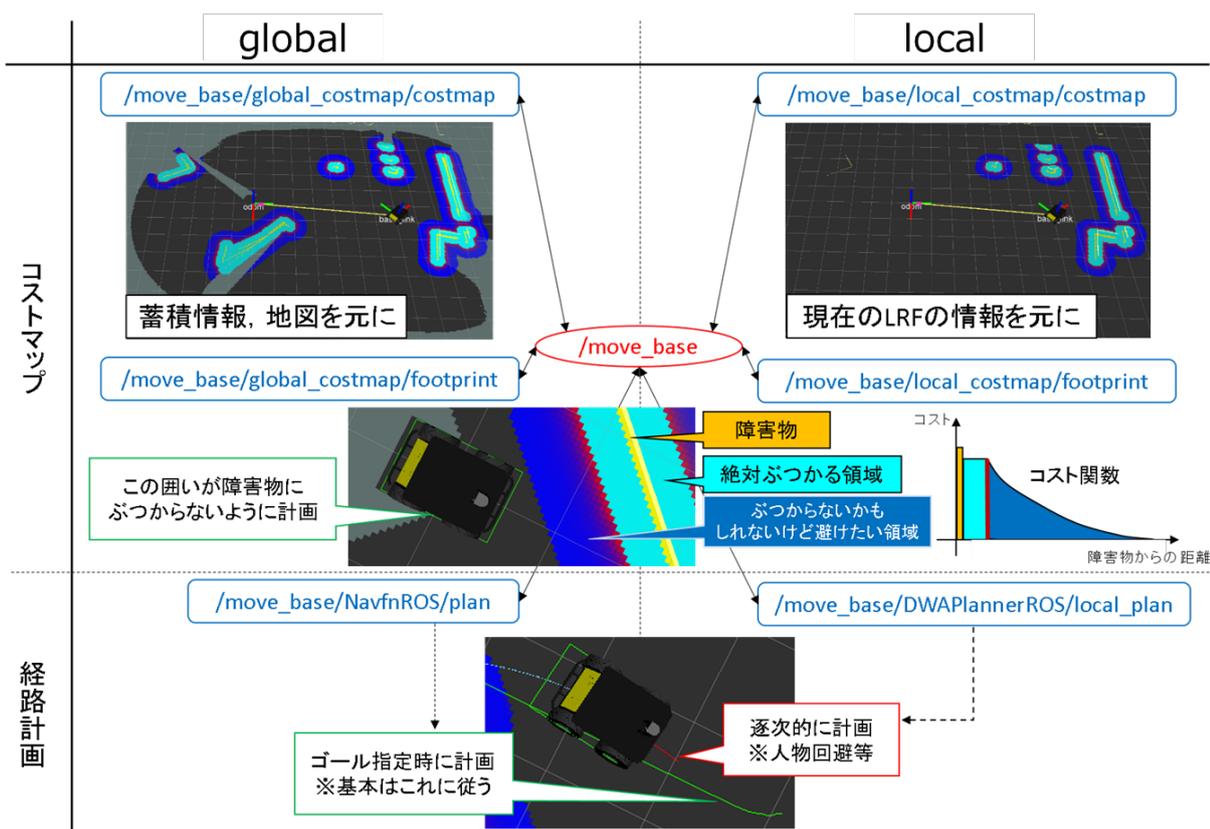
「コストマップ管理機能」は地図データやセンサによって検出した障害物に近いほど大きくなるコスト情報を管理する機能である。「global」と「local」の 2 種類あり、役割はそれぞれ以下の通りである。

名称	役割
global	事前に得られた地図データを元に予め振られたコスト情報。 地図データは「map_server」から静的に得られるデータである。
local	現在得られたセンサ情報をもとに、事前に得られた情報からは存在しない障害物に割り振ったコスト情報。 人など、動くものはこの local でコスト情報を割り振る。

「経路計画機能」はコスト情報を使って障害物を回避しながら経路計画を行う機能である。こちらもコストマップ管理機能と同じく「global」と「local」の2種類あり、役割はそれぞれ以下の通りである。

名称	役割
global	事前に得られたコスト情報をもとにゴール位置までの経路計画を行う。 スタート地点からゴールまでに障害物があった場合、その障害物を回避する経路計画を作成することが可能。
local	global 経路計画に沿って短い距離を逐次経路計画を行い、センサから得られた情報より予期せぬ障害物(人など)が経路計画に現れた場合、障害物を回避する経路計画を作成した後、再び global 経路に復帰する。

move_base をまとめると以下図となる。

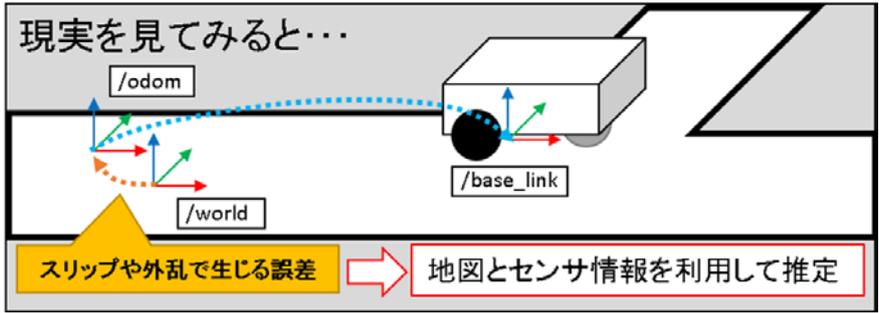


global は蓄積された地図データから得られた情報を元に障害物を回避する経路計画を作成する。

local は現在の LRF 情報を元に逐次短い距離の経路計画を作成する。

amcl(adaptive Monte Carlo localization)「自己位置推定機能」を持つ。

ロボットは車輪の回転量で自己位置を持つことができるが、スリップや外乱等で誤差が生じる場合がある。その誤差を地図データとセンサ情報から得られた周辺情報を駆使して補正を行う。

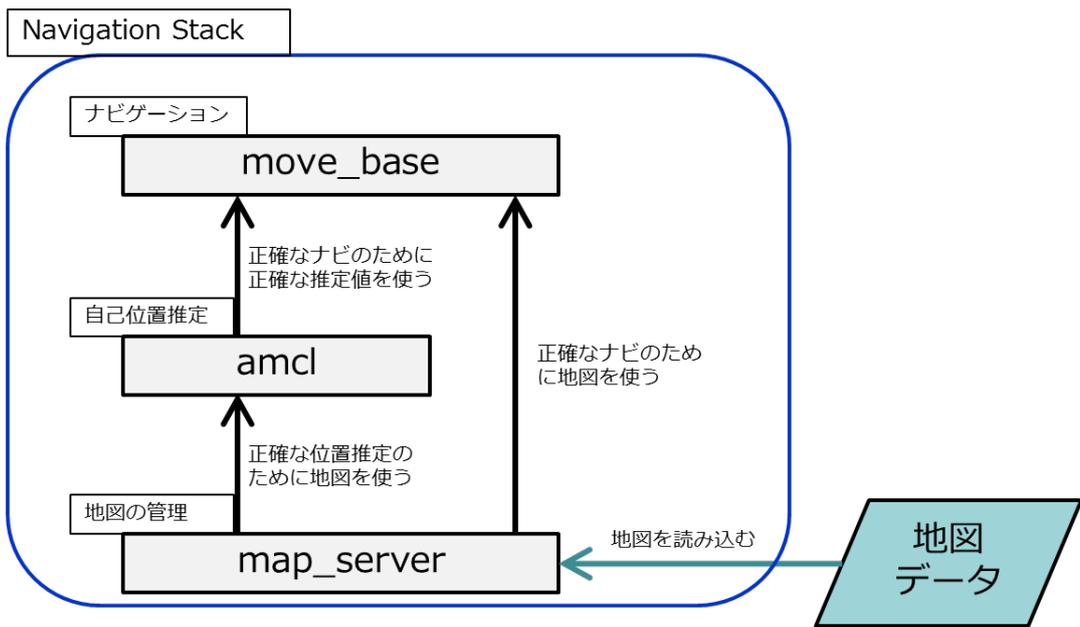


map_server は「地図機能」を持つ。
 地図データが持っているコスト情報を Navigation Stack 内で使用出来るようになるのが役割となる。

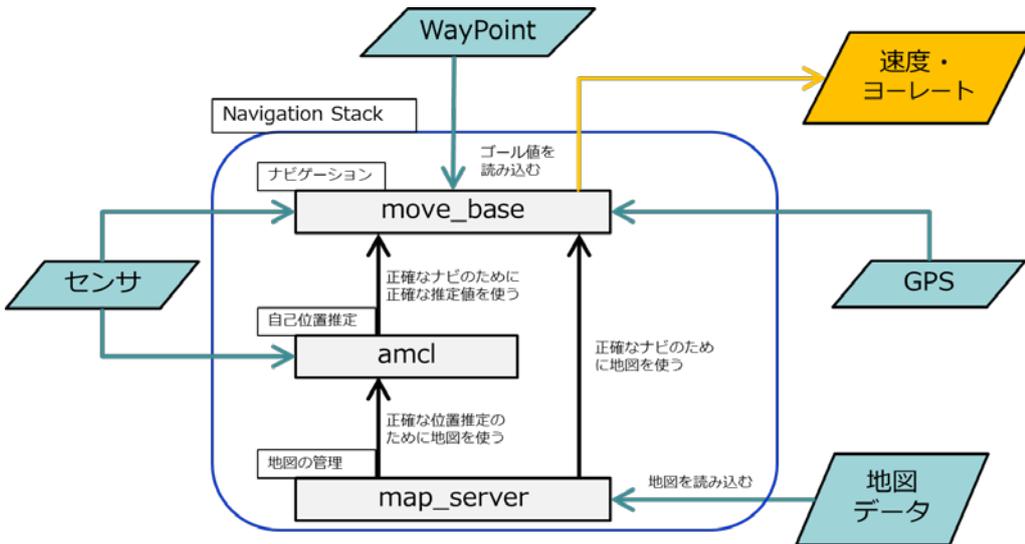
Navigation Stack の各ソフトウェアの名称及び役割は以下の通り。

名称	処理
move_base	Navigation Stack のナビゲーション機能を持つ。センサ情報、ゴール値、地図データから、速度とヨーレートを Publish する。
amcl	Navigation Stack の自己位置推定機能を持つ。センサ情報、地図データから、自己位置推定を行った位置データを Publish する。
map_server	Navigation Stack の地図管理機能を持つ。予め読み込んだ地図データを Publish する。

Navigation Stack の簡略図は以下の通り。



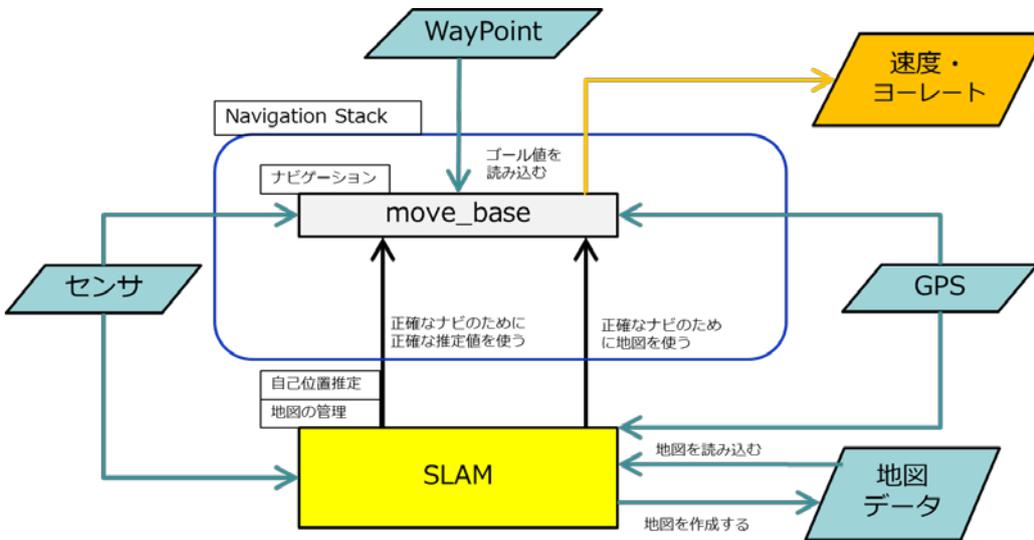
先程の図に、Navigation Stack を中心とした構成は以下図となる。



上記図に、SLAMを組み込む構成を検討する。

これまでの解析結果より、Navigation Stack 内の「amcl」「map_server」はそれぞれ「自己位置推定」と「地図管理」を機能として持つ。

つまり、「amcl」「map_server」の部分を SLAM に置き換えることが可能である。置き換えた図を以下に示す。

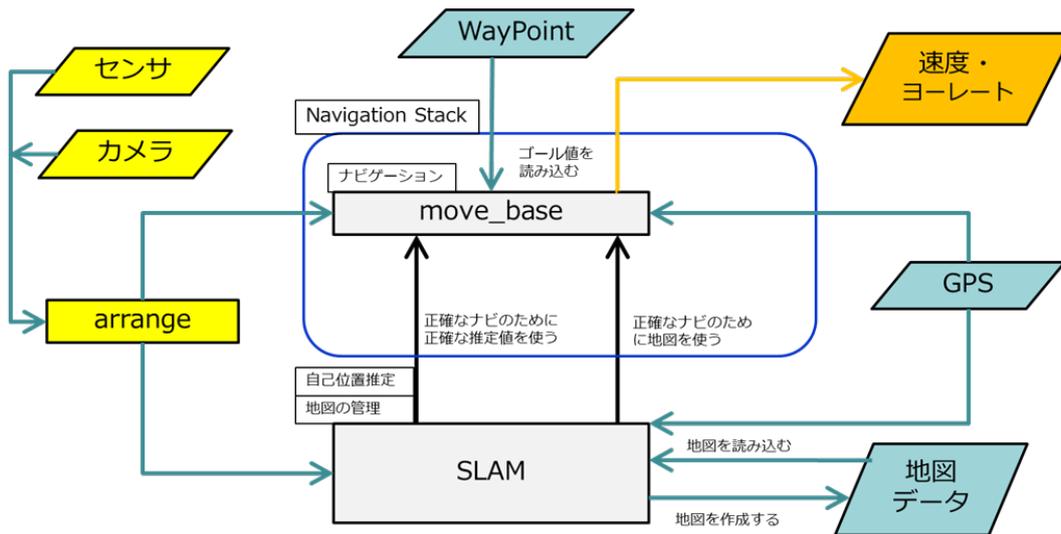


SLAM に置き換えることにより地図の作成も可能となり、SLAM を使用しない場合と比較して高精度で安定性の高い周辺状況・自己位置把握を実現することが可能となる。

最後に、雪道対応自律化プラットフォーム構想を検討する。

先程の構成では積雪環境において、特徴点の抽出が難しく十分な精度を得られない。

そこでカメラで取得した周囲画像にセマンティック・セグメンテーションを施したデータを組み合わせた構想として、以下構成を検討した。



今後、センサ・カメラのいずれかの有益な情報を使って自動運転を行う、または組み合わせた情報を使って自動運転を行う、のいずれかが考えられるが、来期は以下論文を調査し検討を進める予定。

https://www.ri.cmu.edu/wp-content/uploads/2017/11/Season-Invariant_Semantic_Segmentation_with_A_Deep_Multimodal_Network.pdf

⇒2D 画像と 3 次元データを組み合わせて、照明・天候・季節による外観変化の影響を受けないセマンティック・セグメンテーションについて記載。

https://www.ri.cmu.edu/pub_files/2015/6/FSR_2015_submission_50.pdf

⇒ビジュアルオドメトリのためのコンテキスト依存スイッチング戦略の学習について記載。

https://www.ri.cmu.edu/pub_files/2015/9/voxnet_maturana_scherer_iros15.pdf

⇒未調査(来期継続)。

https://www.ri.cmu.edu/pub_files/2015/5/ICRA15_1722_FI.pdf

⇒未調査(来期継続)。

https://www.ri.cmu.edu/pub_files/2014/7/aditya_tr.pdf

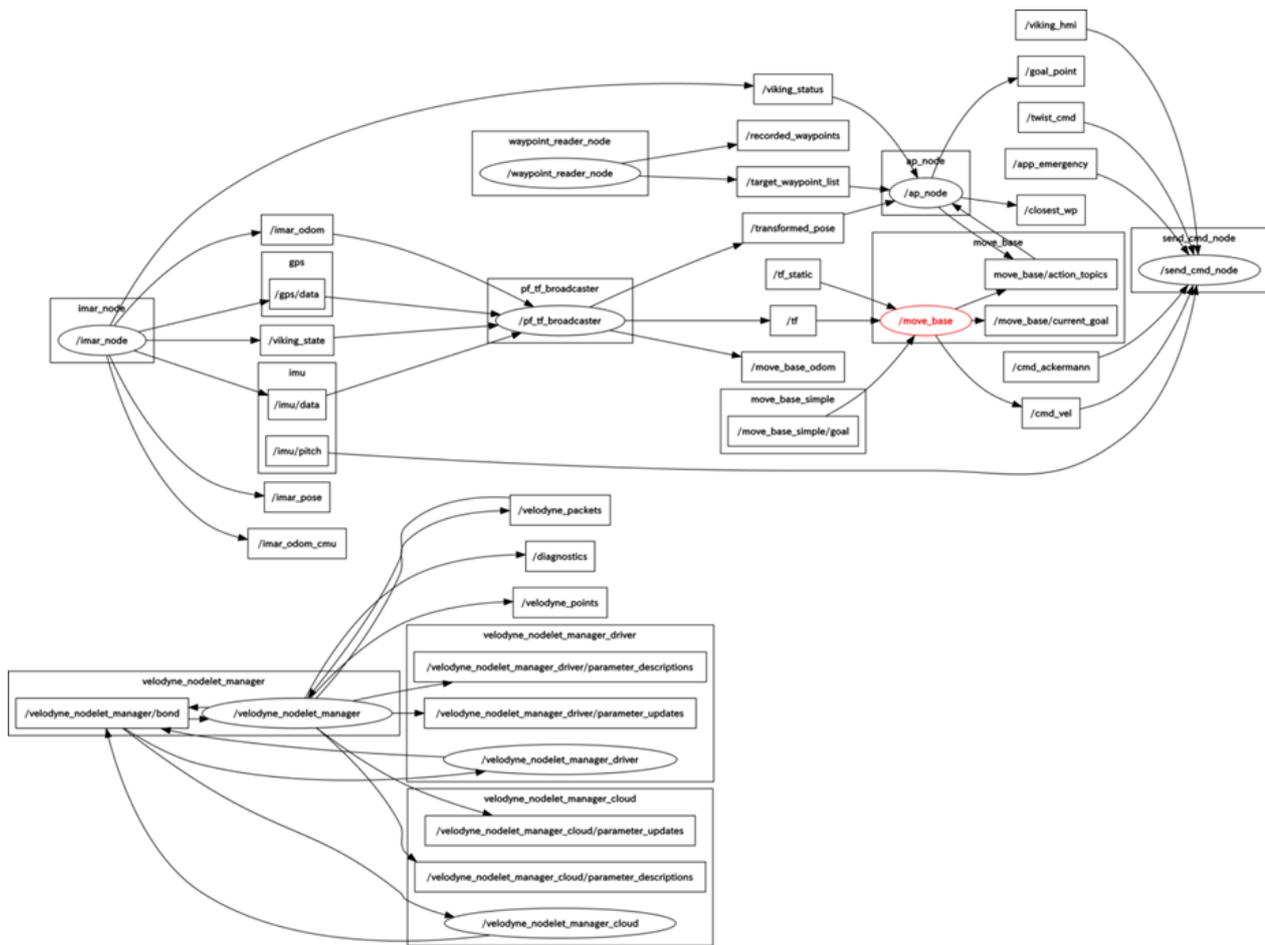
⇒未調査(来期継続)。

【5. 雪道対応知能化プラットフォームの実装(29 年度実施)】

雪道に対応するための自律化プラットフォームのベースとなるシステムを構築する。

Navigation Stack から速度・ヨーレートを出力するまでの試作を行った。

来期以降は「【2. 雪道対応知能化プラットフォーム構想の検討(29 年度実施)】」で検討したプラットフォーム構想に基づき、実装する予定である。



2-1-1-(3) 2年度の目標

雪道対応 SLAM 技術開発について 30 年度は、以下目標に向けて活動を実施した

・30 年度の目標

1. LiDAR-SLAM 評価及び雪道環境認識
2. Visual-SLAM 評価及び雪道環境認識
3. サーモカメラ評価及び雪道環境認識

2-1-1-(4) 具体的な内容

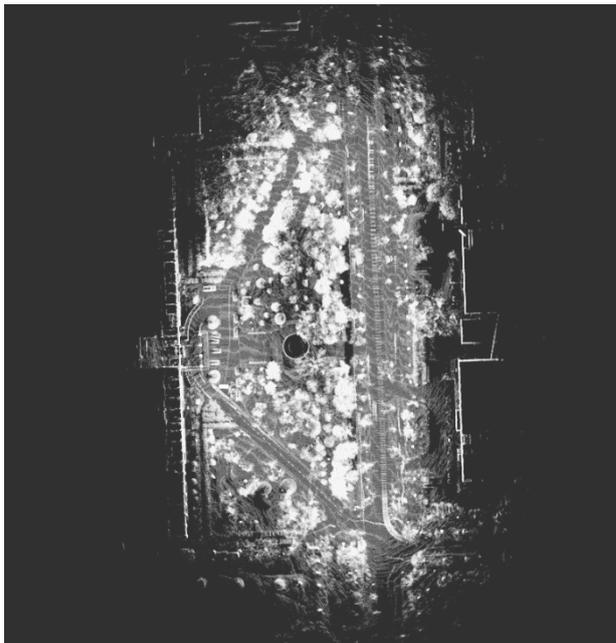
【1. LiDAR-SLAM 評価及び雪道環境認識】

SLAM を実現するためには三角測量・TOF 法などの原理に基づくレーザー測距システム LiDAR を実装した手法のほか、RGB-D カメラ (kinect や RealSense など)、RGB ステレオカメラ、あるいは単眼カメラを利用した方法など数多く提案されている。本プロジェクトでは、最も基本的な手法として LiDAR-SLAM を検討した。LiDAR-SLAM として代表的なのは、Gmapping, Hector-SLAM, Cartographer の 3 つである。これらの諸減について以下にまとめる。

諸元	Gmapping	Hector	Cartographer
----	----------	--------	--------------

アルゴリズム・ベース	RBPF(Rao-Blackwell→ Particle Filter)	Scan-Matching+拡張 Kalman filter	Graph-base
Loop-Closing	なし	なし	あり
DOF	3DOF(Odom+Lidar)	3DOF(Lidar)／ 6DOF(IMU+Lidar)	3DOF(Lidar)／ 6DOF(IMU+Lidar)
メリット	屋内環境、メジャー	凸凹な環境、odom 情報いらない	屋内環境向け、odom 情報いらない
デメリット	odom 情報が必須、凸凹な環境、広域に不向き	Lidar フレーム更新頻度、精度に要求高い	CPU にかかる計算負荷が大きい

これらの手法について、作成した地図の正確さ・計算コスト・積雪環境における適用可能性という3つの基準で評価を行った。その結果、Cartographer は計算量の少なさ、リアルタイムで正確な地図を生成可能であるという点で最も優れていた。しかしながら、Cartographer にもいくつかの欠点がある。まず、指定できるパラメータが膨大であり、その調整が煩雑である点である。また、Cartographer は計算コストを考慮したアルゴリズムとなっており、引用した文献では GPU を利用可能なことになっているが、公開されているソースコードではまだ利用することができない。そのため、GPUによって最適化することができていない状態である。実際、周りが何も無い(特徴点が疎な環境)ではシステムがフリーズする傾向にあった。



また、より最近のアルゴリズムとして NDT scan matching についても実験を行った。NDT scan matching は Autoware という自動運転のユーザーインターフェースから利用することができる。上図は北海道大学工学部周辺の環境において 3次元地図を作製した例であるが、今まで説明した LiDAR-SLAM よりも正確な地図が作成できている。しかしながら、この例では地図の作成に 3 時間以上 (DELL 社 Alienware, CPU: i7-7820HK, GPU: GeForce GTX1080 with Max-Q Design, Memory: 32GB) かかっており、リアルタイム性が必要とされるアプリケーションには不適である。事前

の点群地図を作成するためには最適であると考えられるが、自動運転システムのナビゲーションまで行おうとすると不適である。

【2. Visual-SLAM 評価及び雪道環境認識】

積雪環境においては、カメラベースの SLAM (Visual-SLAM) の適用は難しい。これは、積雪環境では色特徴量が疎であるため、SLAM を行うための十分な特徴量を得ることができないためである。本プロジェクトでは Visual-SLAM の代表的な手法である ORB-SLAM2 を適用してみたが、望ましい結果を得ることはできなかった。

【3. サーモカメラ評価及び雪道環境認識】

3-1. 各センサの性能評価

初年度に取得した、雪道走行時におけるセンシング結果のデータ検証を実施するために、今回使用するセンサの特徴を以下の表に整理した。

LiDAR は、水平全方位 360 度、かつ、長距離に周囲の障害物、および freespace を 3 次元的に把握できるが、豪雨、豪雪、霧などの悪天候時に検出性能が低下する。

サーモカメラは、環境中の温度分布をセンシングできるが、逆光などの影響を受けやすい。

ミリ波は、雨、霧、逆光などの影響を受けにくい、電波の反射率の低い物体の検出が難しい。

各センサの特徴

No.	対象	製品名	性能	強み	弱み
1	LiDAR	HDL-32E [Velodyne 製]	<ul style="list-style-type: none"> 測定距離約 100m 水平全方位 360 度 測定精度約±2cm 32 個のレーザー送受信センサ 	<ul style="list-style-type: none"> 水平全方位 360 度、かつ、長距離に周囲の障害物、およびフリースペースを 3 次的に把握できる 電波の反射率が低い物体も検出できる (段ボール箱/木材/発泡スチロールなど) 	<ul style="list-style-type: none"> 高価である[約 580 万円] 透過しない 豪雨、豪雪、霧などの悪天候時に検出性能が低下
2	サーモカメラ	PI450 [Optris 製]	<ul style="list-style-type: none"> 非接触で二次元の温度測定 (-20°C~150°C) 測定精度±2°C または ±2% 解像度 382 x 288 ピクセル 	<ul style="list-style-type: none"> 環境中の温度分布をセンシングできる 	<ul style="list-style-type: none"> 逆光などの影響を受けやすい 物体の表面温度に依存する (裏側を透過して見えない)
3	ミリ波	IWR1642 [TI 製]	<ul style="list-style-type: none"> 測定距離約 84m アンテナの視野角は±60° 	<ul style="list-style-type: none"> 安価である[約 3 万円] 雨、霧、逆光などの影響を受けにくい 透過する 	<ul style="list-style-type: none"> 電波の反射率の低い物体の検出が難しい (段ボール箱/発泡スチロールなど) 物体の識別は困難

※各センサ仕様、データシートをもとに作成

今回使用する各センサの仕様を、以下の図に示す。

仕 様	
商品コード	HDL-32e
センサータイプ	32 個のレーザー + 検出器 (送受信センサー)
測定範囲・測定視野	水平 360° 全方位 垂直 41.3° (+10.67° ~ -30.67°)
測定距離	約 100m (1m ~ 100m)
測定スピード	5 ~ 20Hz
測定ポイント数	約 700,000 ポイント / 秒
測定精度	±2cm (1σ@25m)
距離測定方式	LiDAR TOF 方式
レーザークラス	Class 1 Eye Safe
レーザー波長	903nm
電源	12V @ 2Amps
動作電圧	9 - 32VDC
重量	約 1.2kg
寸法	高さ 144.2mm x 直径 85.3mm
耐衝撃	500m/秒 2 振幅、11 ミリ秒間
耐振動	5 Hz ~ 2000Hz、 3 Grms
耐環境	IP67 対応
環境温度	動作時: -10°C ~ 50°C 保管時: -30°C ~ 90°C
出力インタフェース	イーサネット 100Mbps (IP アドレス変更可)
データ	UDP パケット (距離、回転角度データ)
位置・オリエンテーション	MEMS 加速度センサー 六軸モーション補正用ジャイロセンサー
GPS	GPS タイム・シンクロナイズ機能 (GPS レシーバーを含む)

LiDAR はパルス状に発光するレーザーを照射して散乱光を検出し、その反射時間 (発光後に反射光を検出するまでの時間) から距離を算出する技術です。



LiDAR(HDL-32E)の仕様

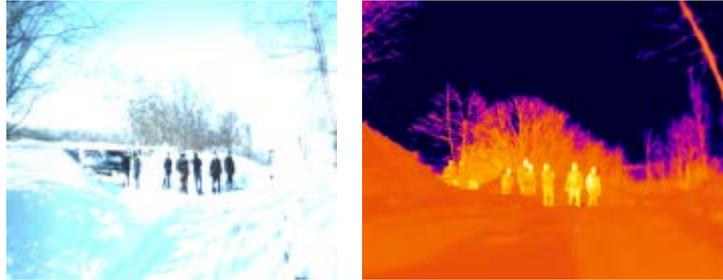
- 赤外線サーモカメラでは、対象物から出ている赤外線放射エネルギーを検出し、見かけの温度に変換して、温度分布を画像表示することが可能です。



- 非接触で二次元の温度測定
-20°C~1500°Cを可視化

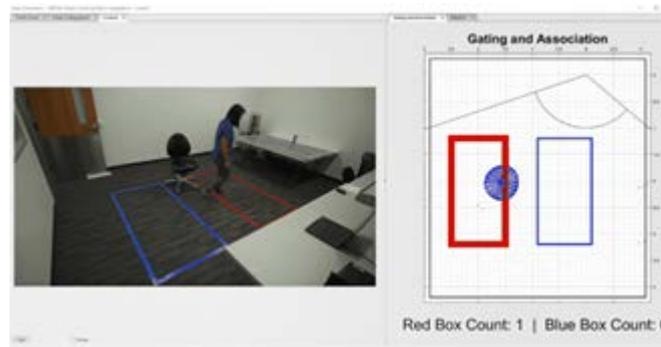
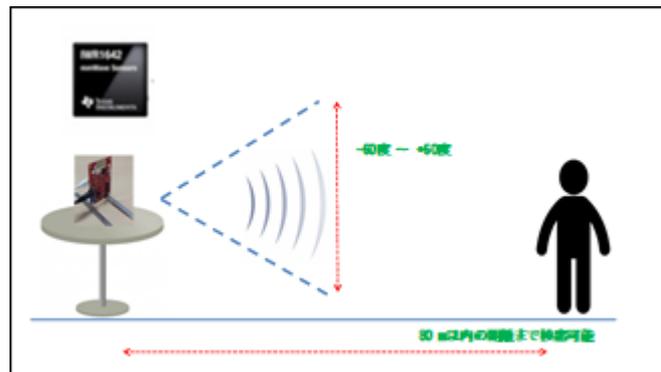
- 解像度は 382 × 288、
フレームレートは 80Hz

- 測定波長域は、
赤外線 7.5 ~ 13 μm



サーモカメラ(PI450)の仕様

- ミリ波とは1波長の長さがミリオーダー(1~10mm)の周波数のことです。
- 一般的には、直進性が強いいため、自動運転時の前方検出用に利用されるケースが多数です(76GHz)。
※後方や側方検出用は 24GHz
- 最大 84m の距離にある物体を検出
- アンテナの視野角は ±60°



ミリ波レーダー(WR1642)の仕様

センシング結果の検証

初年度に取得した、雪道走行時におけるセンシング結果のデータ検証結果を以下に記載する。

LiDAR、および、サーモカメラは、わだち、樹木、人、吹き溜まりを検出出来た。

ミリ波レーダは、樹木、人を検出出来たが、わだち、吹き溜まりを検出出来なかった。

対象別比較結果

No.	対象	対象ファイル	カメラ(画像)	LiDAR	サーモカメラ	ミリ波
1	車道 (わだち)	/Data/bagfiles /2_28/Round/ 2018-02-28-14- 53-31.bag Bag Time: 1519797263		 検出可能	 検出可能	 検出不可能
2	樹木	/Data/bagfiles /2_28/Round/ 2018-02-28-14- 53-31.bag Bag Time: 1519797376		 検出可能	 検出可能	 検出可能
3	人	/Data/bagfiles /2_28/Round/ 2018-02-28-14- 53-31.bag Bag Time: 1519797378		 検出可能	 検出可能	 検出可能
4	雪の 吹き溜まり	/Data/bagfiles /2_28/Round/ 2018-02-28-14- 53-31.bag Bag Time: 1519797242		 検出可能	 検出可能	 検出不可能

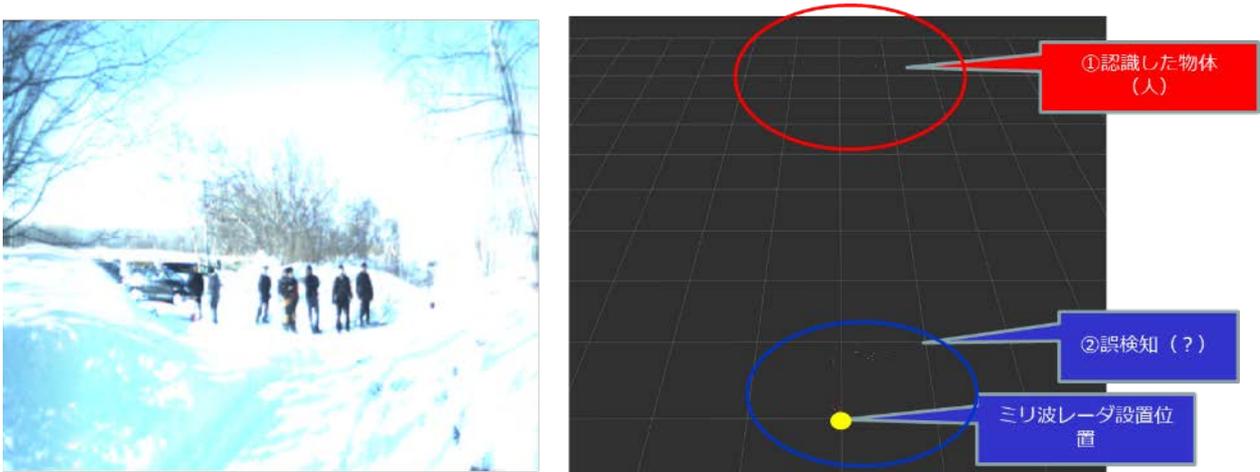
※2018/2/28 苦東周回コースにて取得したデータをもとに解析

ミリ波レーダの活用

2018/2/28 苦東周回コースの取得したミリ波データを解析した結果を以下に示す。

カメラ画像上の人をミリ波で検出できることが判明した。

また、ミリ波レーダ設置位置から半径1メートル以内に検出した点群データがあるが、障害物は存在しないため、誤検知と思われる(ノイズ、雑音と判断)。



2018/2/28 苦東周回コースの取得データ(ミリ波)解析結果

ミリ波レーダについて、性能評価を実施した結果を以下に示す。

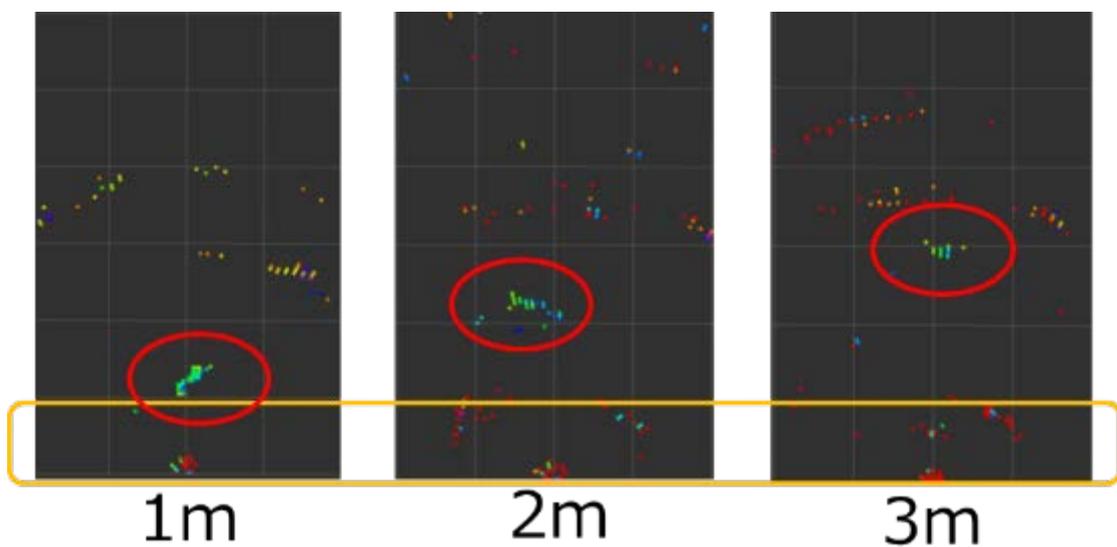
ミリ波レーダ設置位置から設置位置半径 1メートル以内のノイズ除去を行う場合、以下方法①または方法②の案が考えられる。

方法①: 設置位置半径 1メートル以内の障害物検知結果を間引く(黄枠内)。

方法②: 生データから特徴量の変化に注目して、対象物を検出する。

※参考「ミリ波レーダーを用いた物標識別の検討(日本無線技報)」

著: 富木 洋一、時枝 幸伸、西山 拓真、菅原 博樹



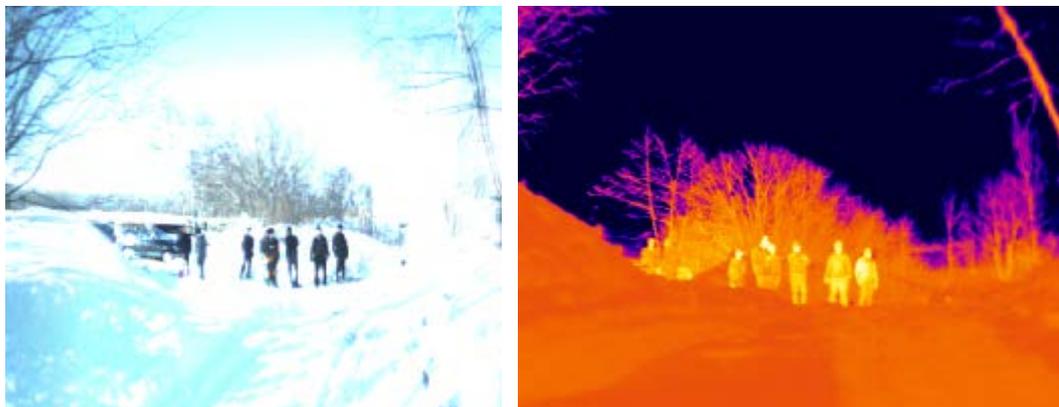
ミリ波レーダの性能評価結果

サーモカメラの活用

2018/2/28 苦東周回コースのカメラ画像とサーモカメラ画像を以下に示す。

人の境界線は RGB 画像データよりも鮮明であるのに対して、温度差の少ない道路と雪の吹き溜まりの境界線は不鮮明である。

このため、雪道上の SLAM に使用するためには、オブジェクト間の温度差が重要であると考えられる。
また、サーモカメラにてわだちの検出が可能であるため、知能化プラットフォームでの制御を行う際の経路指定に活用できると考える。以上のことより、引き続きサーモカメラの活用についてはセマンティックセグメンテーションに含めて研究することとした。



2018/2/28 苫東周回コースのカメラ画像とサーモ画像

2-1-2-(5)3年度の目標

雪道対応 SLAM 技術開発について令和元年度は、以下目標に向けて活動を実施する。

・令和元年度の目標

1. LiDAR-SLAM 生成マップの活用の設計と評価
2. 2次データのコストマップ反映の最適化
3. 自律走行制御改善(IMU)
4. 雪道での最終検証

2-1-1-(6)具体的な内容

【1. LiDAR-SLAM 生成マップの活用の設計と評価】

我々が実験フィールドとして主に活用している苫東のような環境(樹木が多い林道のような環境)では SLAM を成功させるために必要な特徴量マッチングが大変困難である。そこで、北海道大学内においてデータを取得することにより、様々な SLAM を適用して評価を行った。データを取得するために、以下のように乗用車にセンサを搭載するための治具を作成した。搭載したセンサは LiDAR, サーモカメラ, RGB カメラ, みちびき受信機, IMU である。



センサを搭載した実験車両(左)および搭載したセンサ

一式(右)

今回使用したセンサは以下の通りである

LiDAR: Ouster 社 OS-1-64

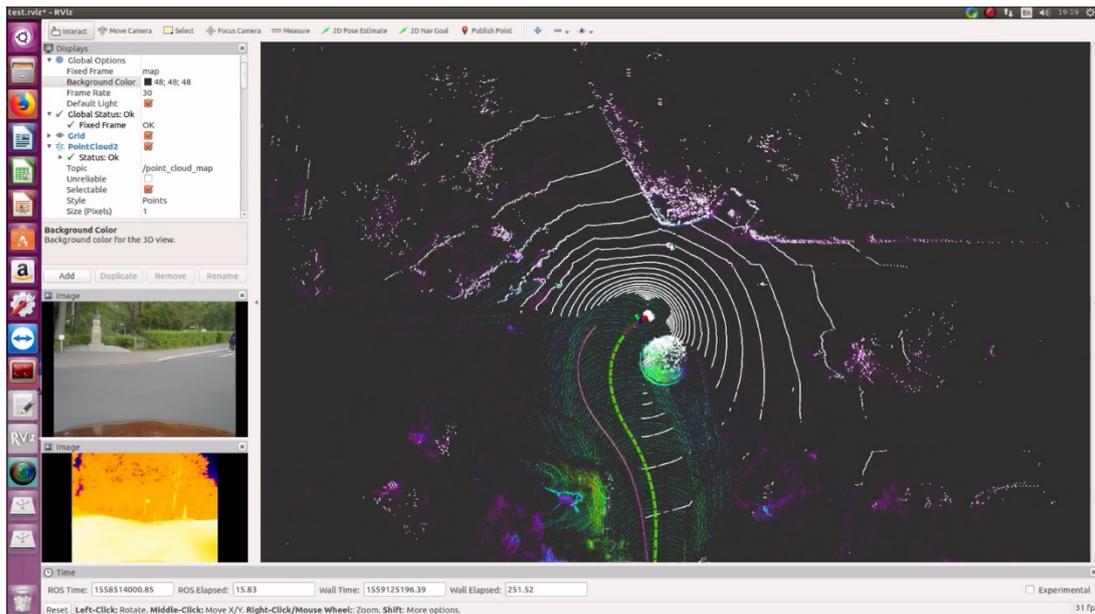
サーモカメラ: Optris 社 PI-640

RGB カメラ: USB2.0 カメラモジュール ELP-USBFHD06H-L100

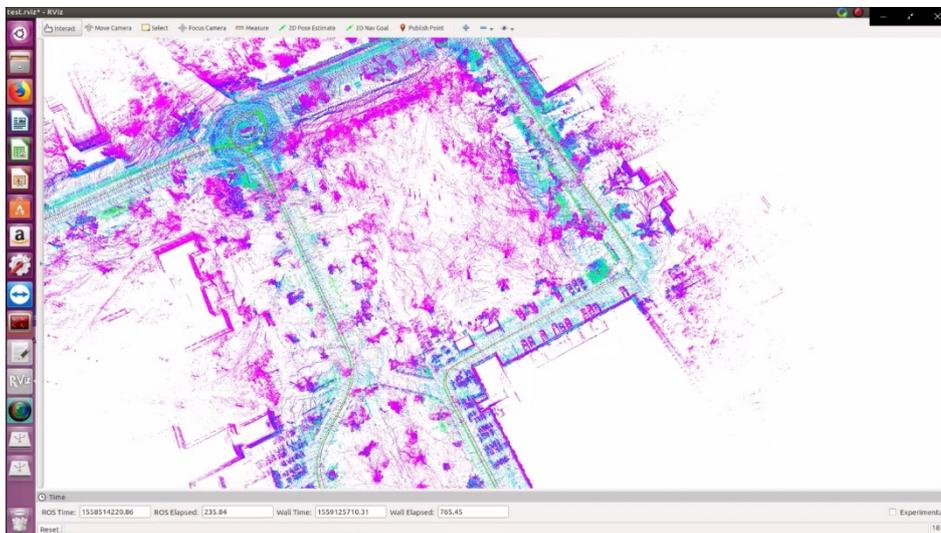
みちびき受信機: 三菱電機社製 AQLOC-Light

IMU: Xsens Mti-30-2A8G4

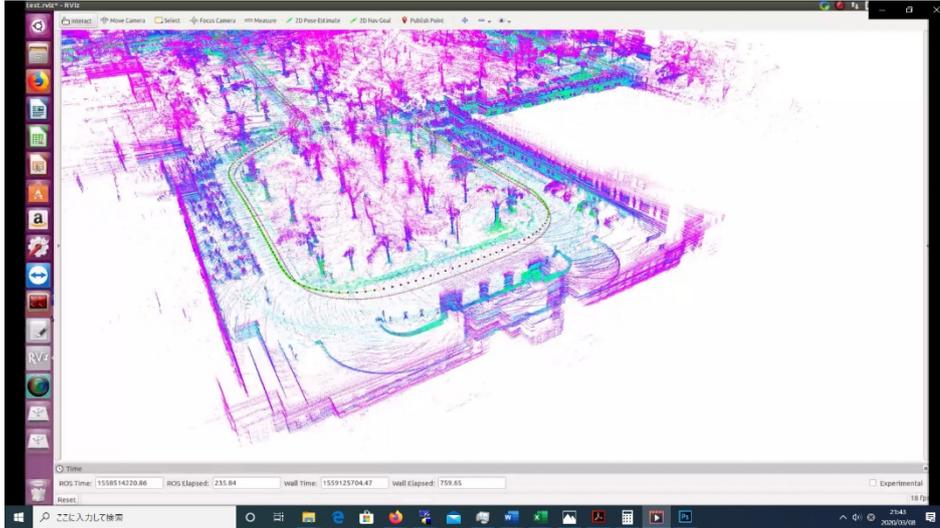
これらのセンサから得られた情報を利用して、Graph-SLAM に基づいた LiDAR 特徴量を利用した SLAM を適用することによって、3次元地図の生成および自己位置同定を試みた。



LIDAR, サーモカメラ, RGB カメラによって取得されたデータを可視化したもの



SLAM を適用することによって得られた 3 次元地図①



SLAM を適用することによって得られた 3 次元地図②

図に示す通り、LiDAR 特徴量が得られる環境においては高精度な 3D マップが得られることを確認した。さらにこれらの手法を雪道環境に適用するために、以下のような環境におけるデータ取得を行った。



本来は3車線存在するが、右側の車線が積雪によって車線が消失

図に示す環境

は、写真上部の青色看板を見るとわかる通り本来3車線存在する環境である。しかしながら、一番右側の車線が積

雪によって消失しており、もし夏季期間に取得した3次元高精度マップが存在していたとしても、その情報を利用して自己位置同定を行うことが非常に困難な環境である。このような環境でのSLAMを試みるため、様々な環境でのデータ取得を試みた。計画では2019年12月よりデータの取得を実施する予定であったが、12月～1月にかけて記録的な雪の少なさのために実験が進まず、2月中旬以降、ようやくデータの取得を開始することができた。現在、様々なSLAMの手法を適用しながらデータの解析を実施中である。

問題点としては、3次元SLAMの実行には膨大なデータ処理ならびに計算量が必要となるため、リアルタイムで実行することは非常に困難である点が挙げられる。例えば今年度に導入したOuster OS-1-64は1秒間に得られる点群の数がおよそ100万点であり、1分間では6,000万点の点群データが得られることになる。もちろん得られる点群の数が多ほど詳細な環境データが得られることを意味しておりメリットではあるが、その処理をリアルタイムで実行することが非常に困難になるというデメリットも抱える。計算機の処理能力の向上や搭載可能なメモリ量の増大というハードウェア能力の向上によって解決出来る可能性もあるが、同時にセンサによって得られるデータ量も増大するため、根本的には大量のデータから処理に必要なデータのみを抽出して処理するようなアルゴリズムが必要となる。この1つのアプローチが本提案で行っている2次元マップの利用であり、以降、2次元データのコストマップ反映の最適化について述べる。

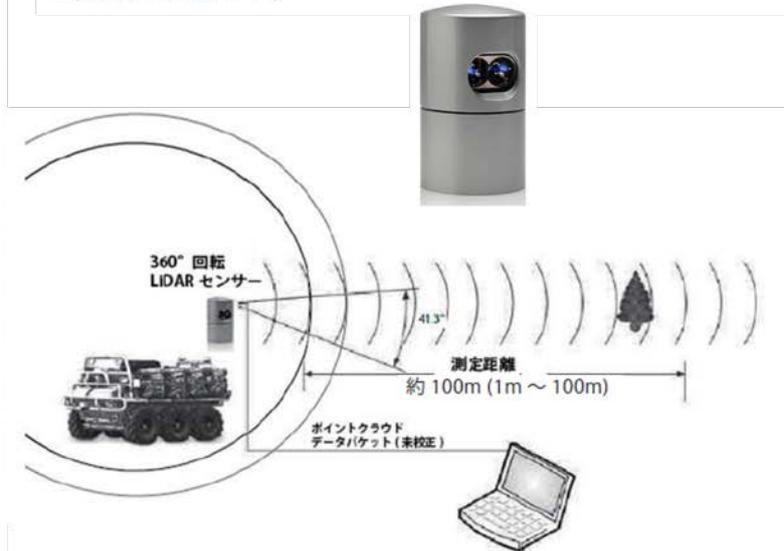
【2次元データのコストマップ反映の最適化】

3D-LIDARにより周辺環境を検知し、障害物などの情報をコストマップとして生成し、調停システムに渡すことで、各種センサからの情報と合わせて、走行可能箇所を特定する。

今回使用する各センサの仕様を、以下の図に示す。

仕 様	
商品コード	HDL-32e
センサータイプ	32 個のレーザー + 検出器 (送受信センサー)
測定範囲・測定視野	水平 360° 全方位 垂直 41.3° (+10.67° ~ -30.67°)
測定距離	約 100m (1m ~ 100m)
測定スピード	5 ~ 20Hz
測定ポイント数	約 700,000 ポイント / 秒
測定精度	±2cm (1σ@25m)
距離測定方式	LiDAR TOF 方式
レーザークラス	Class 1 Eye Safe
レーザー波長	903nm
電源	12V @ 2Amps
動作電圧	9 - 32VDC
重量	約 1.2kg
寸法	高さ 144.2mm x 直径 85.3mm
耐衝撃	500m/秒 2 振幅, 11 ミリ秒間
耐振動	5 Hz ~ 2000Hz, 3 Grms
耐環境	IP67 対応
環境温度	動作時: -10°C ~ 50°C 保管時: -30°C ~ 90°C
出力インタフェース	イーサネット 100Mbps (IP アドレス変更可)
データ	UDP パケット (距離、回転角度データ)
位置・オリエンテーション	MEMS 加速度センサー 六軸モーション補正用ジャイロセンサー
GPS	GPS タイム・シンクロナイズ機能 (GPS レシーバーを含む)

LiDAR はパルス状に発光するレーザーを照射して散乱光を検出し、その反射時間 (発光後に反射光を検出するまでの時間) から距離を算出する技術です。



LiDAR(HDL-32E)の仕様

LiDAR で測定した対象物の位置は、レーザーの照射角度 (ϕ , θ) と対象物までの距離 (r) からなる極座標として取得することができる。

車輛の制御のためには、車両周辺の対象物の位置を車両からの相対位置として求める必要がある。

このため、極座標系として取得したデータを直交座標系に変換して、使用する。

この直交座標に変換したデータの塊を PointCloud と呼ぶ。また、PointCloud 内の 1 つの測定座標を測距点と呼ぶ。

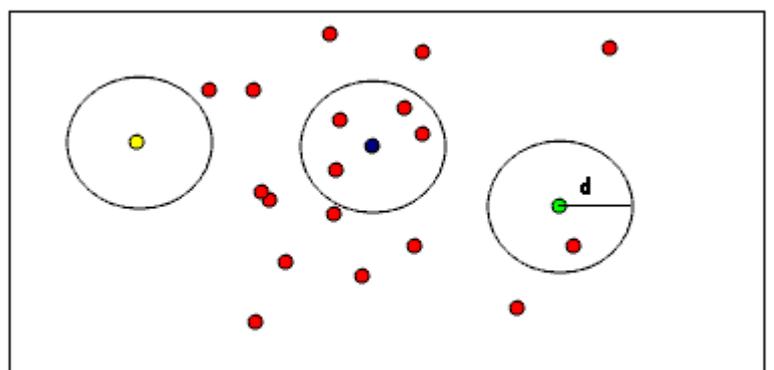
降雪の影響

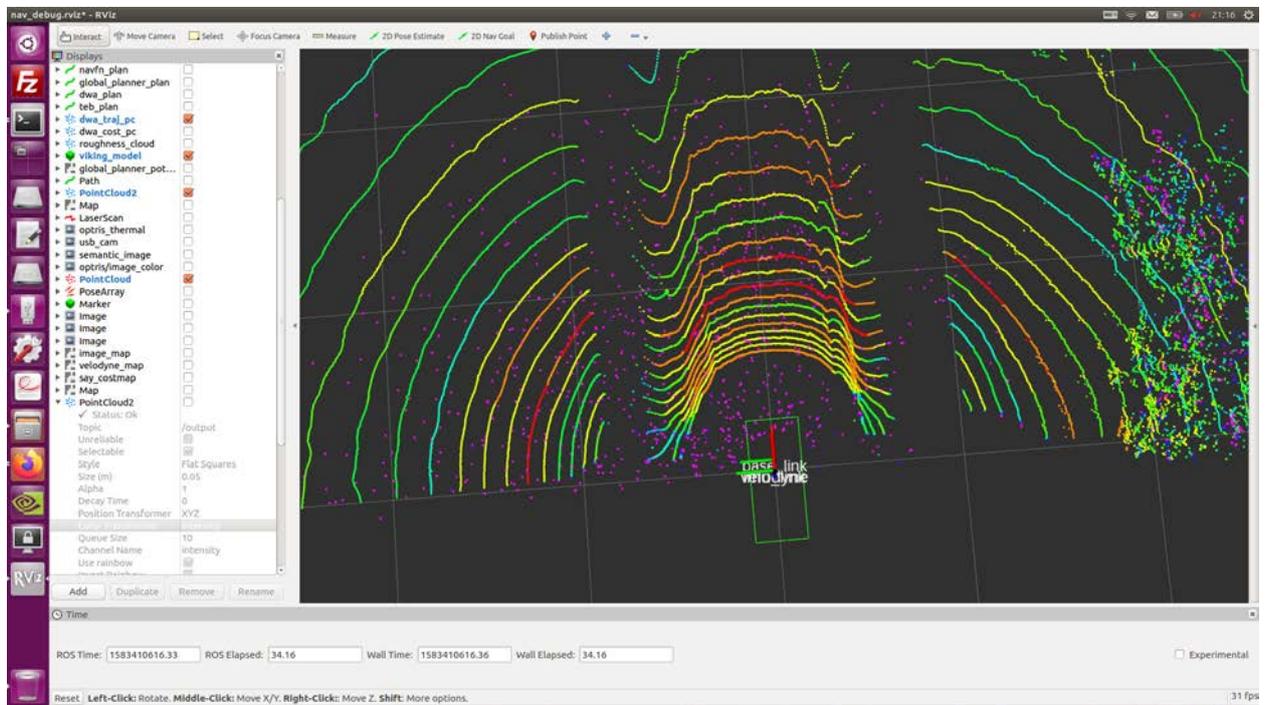
LiDAR はレーザー光の反射を利用して、対象物までの距離を測定するため、降雪時は、空中を落下する雪を測定してしまう。

LiDAR では、対象物が何かを検知することができないため、PointCloud としては、空間上に対象物が存在することになる (実際には雪であるため、車両の走行には何も支障がないが、PointCloud としては、衝突する対象になってしまう)。

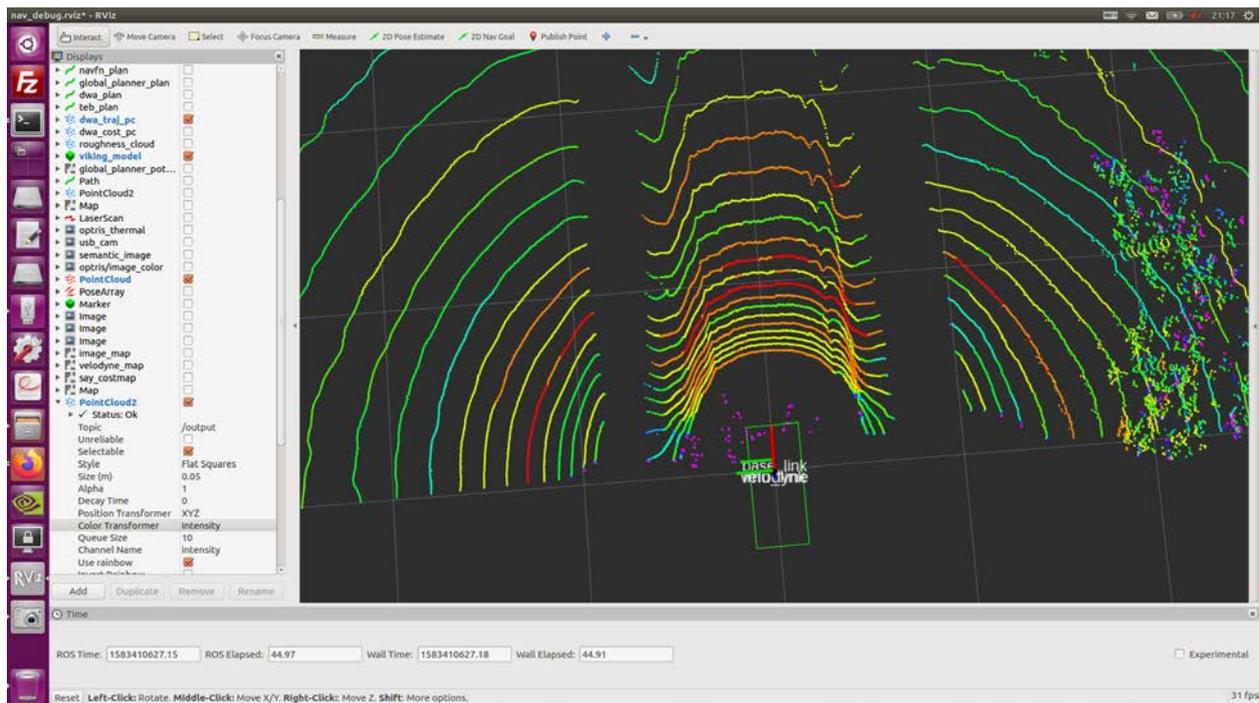
降雪による影響を排除するため、「Radius Outlier Removal フィルタ」を実装した。

それぞれの点群の半径 d の中に、点群が指定された個数以下ならば削除するフィルタである。例えば右図の画像で指定された点群が 1 ならば、半径 d 内に点群が存在しない黄色の点が削除される。指定された点群が 2 の場合、黄色に加えて緑の点群も削除される。





フィルタ適用前の降雪時の PointCloud の状況



フィルタ適用後の降雪時の PointCloud の状況

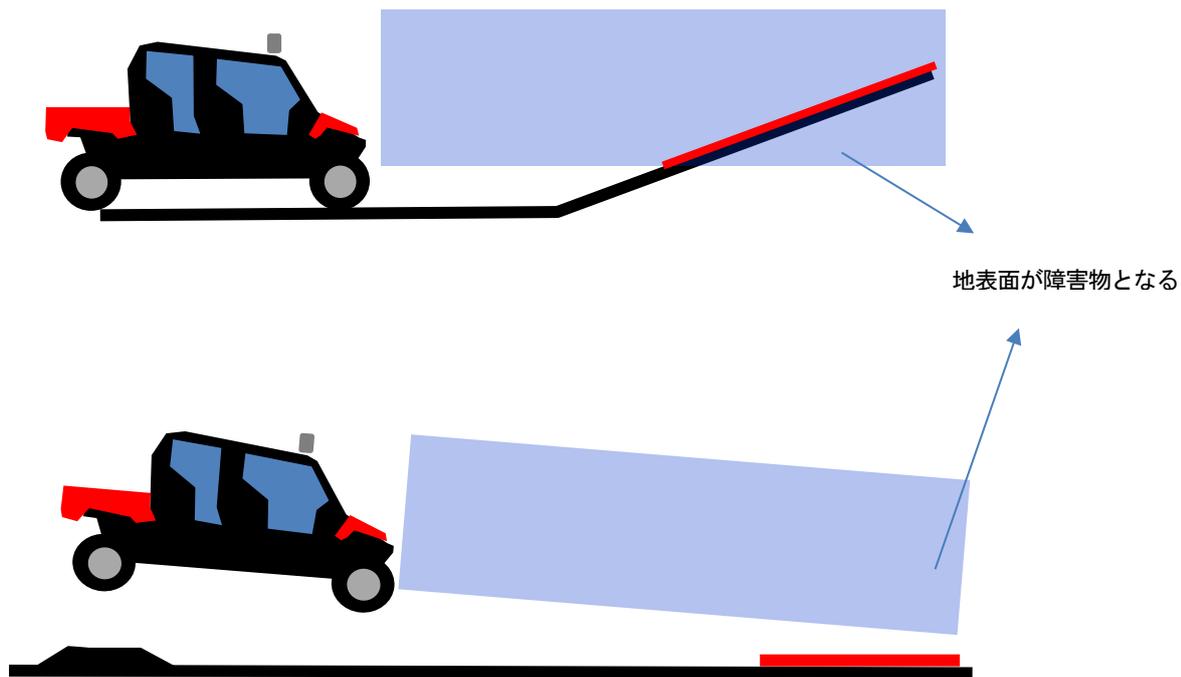
地形による地表面の誤検知

PointCloud 内の測距点のうち、地表面から一定の高さ以上のものを障害物として認識するようにすると、処理が単純で、効果があるが、これは、地表面が平滑な場合にのみ利用することが可能な方法である。

この方法で障害物を検出しようとすると、車両の姿勢が変化した場合や、上り坂で、地表面を障害物として検出してしまう。



このエリア内にある
PointCloud を障害物と

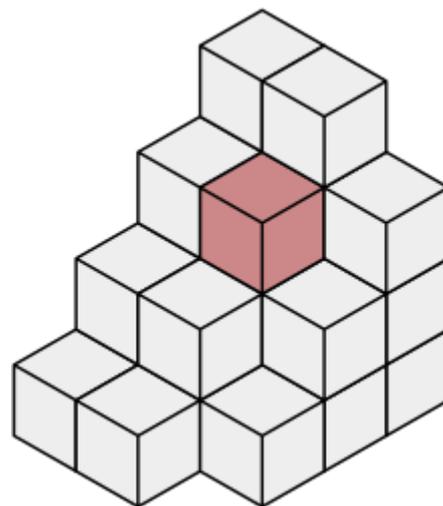


Voxel による障害物検出方式

空間を複数の立方体に分割して、空間内に測距点が存在するかを検出する方法で、空間上の立方体を Voxel と呼ぶ。

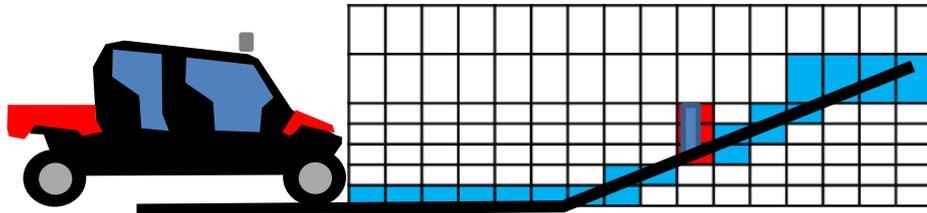
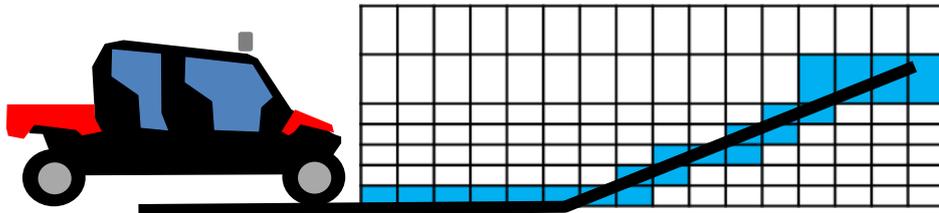
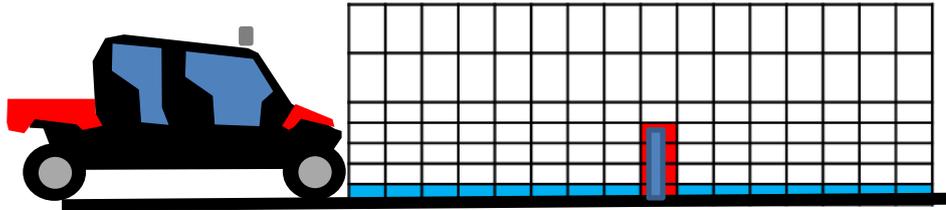
今回、高さにより、 $50\text{cm}(W) \times 50\text{cm}(D) \times 20\text{cm}(H)$ と $50\text{cm}(W) \times 50\text{cm}(D) \times 50\text{cm}(H)$ の2種類の Voxel を使用して、障害物を検出する方式を採用した。

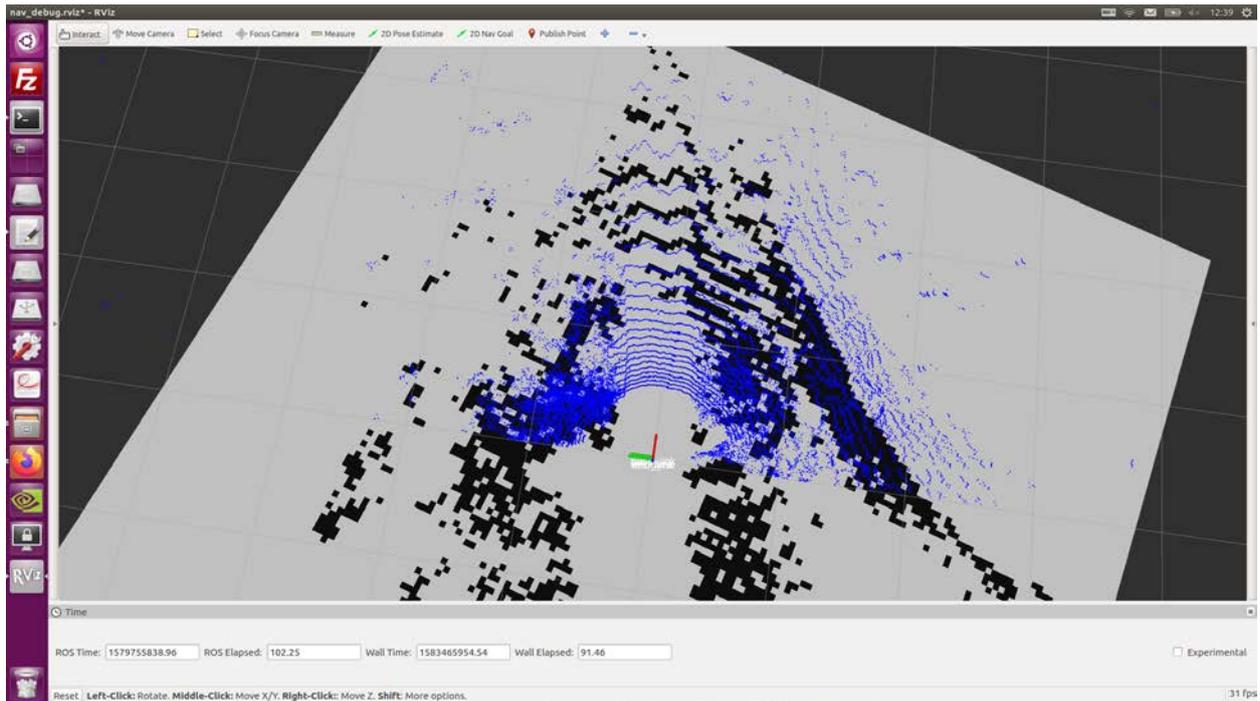
地表面から 100cm の部分は、 $50\text{cm}(W) \times 50\text{cm}(D) \times 20\text{cm}(H)$ の Voxel を使用し、地表面から $100\text{cm} \sim 200\text{cm}$ の部分には、 $50\text{cm}(W) \times 50\text{cm}(D) \times 50\text{cm}(H)$ の Voxel を使用した。



測距点が存在する Voxel が3段以上連続して存在するときに、障害物と認識するものとして、その Voxel に対応するコストマップ上のメッシュをコストありとすることにする。

これにより、車両の姿勢変化による地表面の誤検知、および、上り坂の誤検知が発生しなくなり、安定した走行が可能となった。





坂道で道路を障害物と誤検知した時の状況

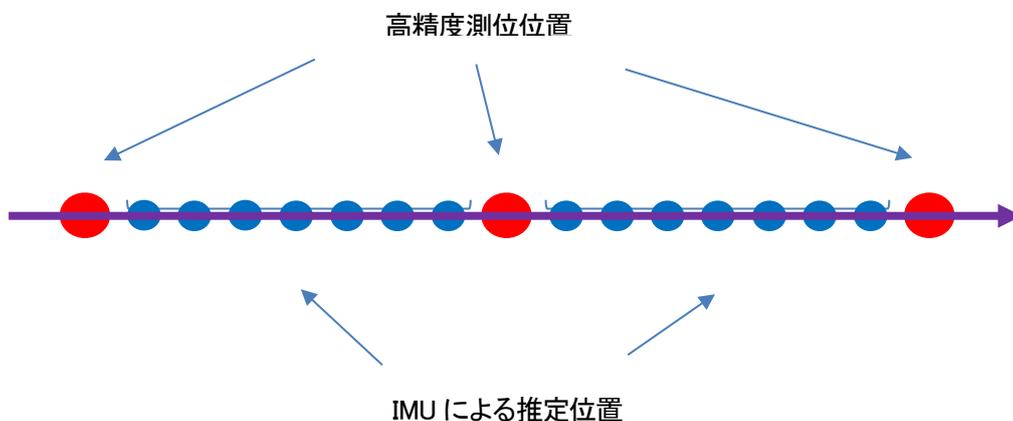
【3. 自律走行制御改善 (IMU)】

自己位置の測位には、準天頂衛星を用いた高精度測位端末 (AQLOC) を使用し、±10cm 程度の測位を行っているが、建物の陰、樹木の陰などでは、衛星捕捉数が減少すると、測位精度が低下する。

精度の低下した測位情報を使用すると、自己位置が、道路外になってしまうような現象が発生するため、このような場合に、自己位置を推定する必要がある。

IMU による自立航法

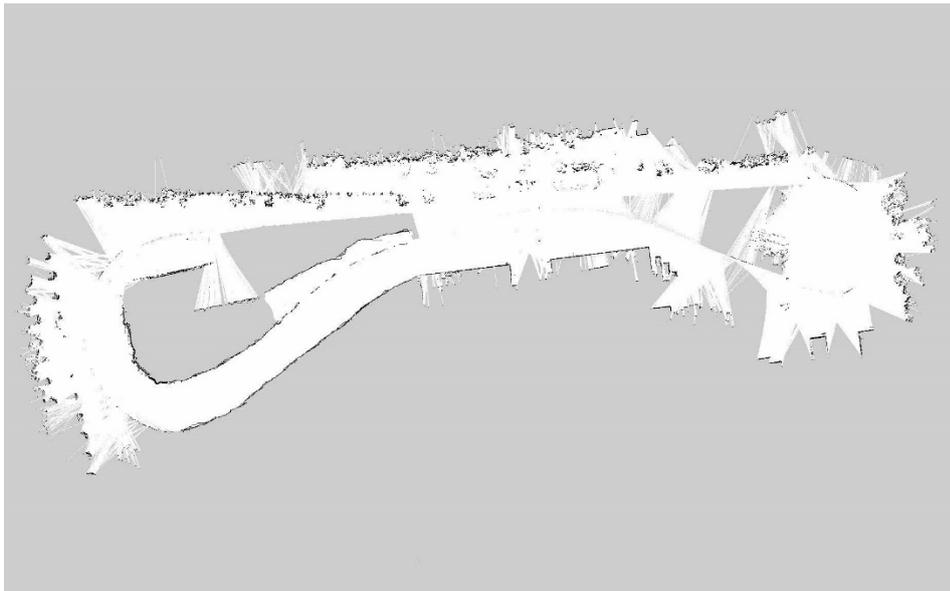
IMU (加速度センサ・角速度センサ・地磁気センサ) を使用して、車両の移動方向と移動量を計測することで、前回、高精度測位した位置からの移動量により、自己位置を推定する。



2D-SLAM(gmapping)による移動量検出

IMUによる移動量の推定の誤差が大きいため、長い時間、高精度測位が行えないと、自己位置が大きくずれてしまう。

この対策として、2D-SLAM を利用して、移動量を検出することで、高精度測位ができていない時間が長くなっても、自己位置を更新可能としている。

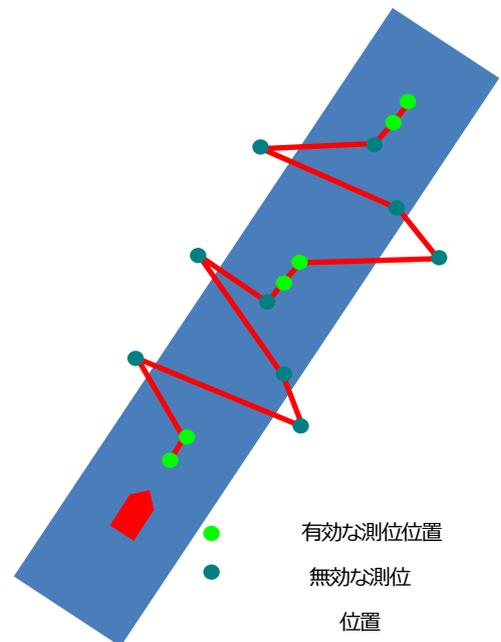


測位データのフィルタリング

測位端末からは、測位精度を示すステータスが出力されるため、高精度測位以外のステータスの測位データを使用しないようにした。しかし、高精度測位状態から、低精度に切り替わった瞬間には、ステータスが高精度測位のまま、測位データの精度が低下していることがあった。

このため、前回の測位位置から一定以上の距離が離れた測位データについては、廃棄するようにした。

これにより、道路から逸脱することなく、走行することが可能となった。



車両制御パラメータ、経路生成パラメータの最適化

車両制御、および、走行経路生成のためのパラメータを調整し、最適化を実施。

パラメータの一覧を以下に示す。

TebLocalPlanner	パラメータ名	値	
Trajectory	teb_autosize	TRUE	時間分解能を自動で変更するか
	dt_ref	0.2	経路の時間分解能の設定
	dt_hysteresis	0.01	時間分解能のサイズ変更のヒステリシス
	global_plan_overwrite_orientation	TRUE	ゴール方向の上書きを行うか
	max_global_plan_lookahead_dist	0	グローバル plan に沿った走行を行うか
	feasibility_check_no_poses	5	経路の実行可能性をどこまでチェックするか
Robot	max_vel_x	6.9	最高速
	max_vel_x_backwards	0.01	バックの速度
	max_vel_theta	0.6981	最高角速度
	acc_lim_x	7	最大加速度
	acc_lim_theta	0.16	最大角加速度
	min_turning_radius	4	最小回転半径
	footprint_model : type	polygon	車両の形
	footprint_model : vertices	[[-2.0, -0.75], [-2.0, 0.75], [1.5, 0.75], [1.5, -0.75]]	車両の形
GoalTrarance	xy_goal_tolerance	2	ゴールとのズレを何 m 許容するか
	yaw_goal_tolerance	0.8	ゴールとのズレを何 rad 許容するか
Obstacles	min_obstacle_dist	1.2	障害物からどれだけ離れるか
	include_costmap_obstacles	TRUE	障害物の読み込み
	costmap_obstacles_behind_robot_dist	0.1	後ろの何mまで障害物を考慮するか
	obstacle_poses_affected	5	障害物との距離を保つ為の点の数
	costmap_converter_plugin	""	障害物を点から別の形に変換
	costmap_converter_spin_thread	TRUE	別スレッドでコンバーターを動かすか
	costmap_converter_rate	15	コンバーターを処理レート
Optimization	no_inner_iterations	30	
	no_outer_iterations	9	
	optimization_activate	TRUE	最適化処理の実行
	optimization_verbose	FALSE	最適化処理のデバッグ文表示
	penalty_epsilon	0.01	車両制約のマージン
	weight_max_vel_x	50	最大速度の重み付け
	weight_max_vel_theta	30	最大角速度の重み付け

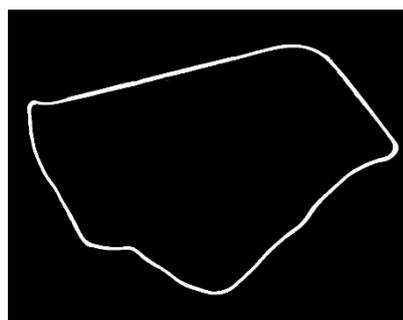
	weight_acc_lim_x	30	最大化速度の重み付け
	weight_acc_lim_theta	10	最大角加速度の重み付け
	weight_kinematics_nh	1000	非ホロノミックロボットを満たす重み付け
	weight_kinematics_forward_drive	1000	正面方向にのみ走行する重み付け
	weight_kinematics_turning_radius	50	最小回転半径の重み付け
	weight_optimaltime	500	経路生成の時間を縮小する重み付け
	weight_obstacle	50	障害物回避の重み付け
	weight_dynamic_obstacle	10	未使用
	selection_alternative_time_cost	TRUE	未使用

	パラメータ名	値	
move_base	planner_frequency	5	経路生成周期
	controller_frequency	5	制御周期
	max_planning_retries	-1	recovery 動作前の再経路探索試行回数
	conservation_reset_distance	3	リカバリー時に消去する障害物の距離
	recovery_behavior_enabled	FALSE	リカバリ動作を有効にするか
Global Planner	use_dijkstra	TRUE	ダイクストラ法で経路を生成するか
	use_grid_path	FALSE	グリッド境界をたどるパスを生成するか
	allow_unknown	TRUE	不明なコストを通れるようにするか
	default_tolerance	0.8	ゴール地点の許容値
	lethal_cost	253	障害物コスト
	neutral_cost	66	通常コスト
	publish_potential	TRUE	コストマップの公開
global_costmap	footprint	[[-2.0, -0.75], [-2.0, 0.75], [1.5, 0.75], [1.5, -0.75]]	車両の形
	width	800	マップの横幅
	height	800	マップの高さ
	resolution	0.5	マップの解像度
	global_frame	map	コストマップのグローバルフレーム
	robot_base_frame	base_link	ロボットのフレーム
	transform_tolerance	0.5	許容できるデータの遅延
	update_frequency	8	マップの更新周期
	publish_frequency	8	マップのパブリッシュ周期
	static_map	TRUE	静的なマップか
	rolling_window	TRUE	マップの中心座標をロボットと同じにするか

	always_send_full_costmap	FALSE	コストマップの変更部分のみを出力するか
	track_unknown_space	FALSE	不明な地点を不明で埋めるか空気で埋めるか
	plugins	staticLayer	使用する追加機能(コストマップの読み込みを使用)
local_costmap	footprint	[[-2.0, -0.75], [-2.0, 0.75], [1.5, 0.75], [1.5, -0.75]]	車両の形
	width	40	マップの横幅
	height	40	マップの高さ
	resolution	0.25	マップの解像度
	global_frame	odom	コストマップのグローバルフレーム
	robot_base_frame	base_link	ロボットのフレーム
	transform_tolerance	0.5	許容できるデータの遅延
	update_frequency	8	マップの更新周期
	publish_frequency	8	マップのパブリッシュ周期
	static_map	FALSE	静的なマップか
	rolling_window	TRUE	マップの中心座標をロボットと同じにするか
	always_send_full_costmap	FALSE	コストマップの変更部分のみを出力するか
	plugins	staticLayer	使用する追加機能(コストマップの読み込みを使用)

グローバルマップの活用

積雪環境では、道路の境界が不明確になり、障害物の回避動作などで、道路外を走行しようとすることがある。積雪の無い環境下の地図がある場合、その情報をグローバルマップに反映することで、道路外を走行しないようにすることが可能となる。



【4. 雪道での最終検証】

サブテーマ1で実施する自動運転の性能目標を検証した。

性能項目	目標値
連続走行距離	30km 以上
最高車速	時速 40km
路面	積雪 10cm の圧雪路
障害物	雪塊、吹き溜まりへの衝突回避率が 100% (ただし 1km あたり約 1 個の障害物を想定する)
天候	晴れまたは曇り

2020 年 2 月 19 日 芦別 北日本自動車大学校内テストコース

北日本自動車大学のオーバルコースと中央の多目的コースを使った 1 セット(約 2km)のコースを設定した。

路面: 除雪された圧雪路

天候: 晴れまたは曇り、時々降雪あり

障害物は、約 1 セット(約 2km)のコースに雪塊と人形の 2 つを設置



目標としていた 30Km の連続走行を達成。途中に設置した障害物は 100%回避。

但し、途中3回ほど、強い降雪の影響により、障害物を誤検知し停止したが、その後、走行を再開。



2020年2月20日 士別市 冬季閉鎖区間公道

設計目標は 40Km/h であったが、実験車両のタイヤは泥寧地用のため、40Km/h での実験走行は危険と判断。

30Km/h での自立走行を確認



2-1-2. 雪道対応セマンティック・セグメンテーション技術の開発

2-1-2-(1) 初年度の目標

雪道対応セマンティック・セグメンテーション技術の開発について 29 年度は、以下の目標に向けて活動を実施した。

・29 年度の目標

1. 識別クラス調査
2. フレームワークと実施環境の調査
3. データセット作成調査
4. データセット作成
5. 他の手法調査

初年度は、セマンティック・セグメンテーションを実現するために、識別クラスの調査を行い、識別クラスを決定する。また、ディープラーニングのフレームワークの調査と実施環境の構築を行い、その環境上でセマンティック・セグメンテーションを実現する手法の調査を行った。平行して、データセットの作成調査を行い、データセットの作成を行う。

2-1-2-(2) 具体的な内容

【1. 識別クラス調査】

初めに、セマンティック・セグメンテーションを実現するために識別クラスを決める必要がある。なぜなら識別クラスを決めなければセマンティック・セグメンテーションのためのデータセットを作成することができないからである。

まず、前提条件として以下を定義し、識別クラスを検討した。

- ・雪道を前提とし、市街地、実験場(苫東)に考慮したクラス分けとする。
- ・クラス数が少ないとニューラルネットワーク内部の構造が単純になってしまい性能を出せなくなる可能性があると考え、10 以上のクラス分けとする。
- ・Web で調査し、以下のサイトのように研究でやっているクラス分けを参考にする。

Class Definitions

Please click on the individual classes for details on their definitions.

Group	Classes
flat	road · sidewalk · parking⁺ · rail track⁺
human	person⁺ · rider⁺
vehicle	car⁺ · truck⁺ · bus⁺ · on rails⁺ · motorcycle⁺ · bicycle⁺ · caravan⁺⁺ · trailer⁺⁺
construction	building · wall · fence · guard rail⁺ · bridge⁺ · tunnel⁺
object	pole · pole group⁺ · traffic sign · traffic light
nature	vegetation · terrain
sky	sky
void	ground⁺ · dynamic⁺ · static⁺

<https://www.cityscapes-dataset.com/dataset-overview/#class-definitions>

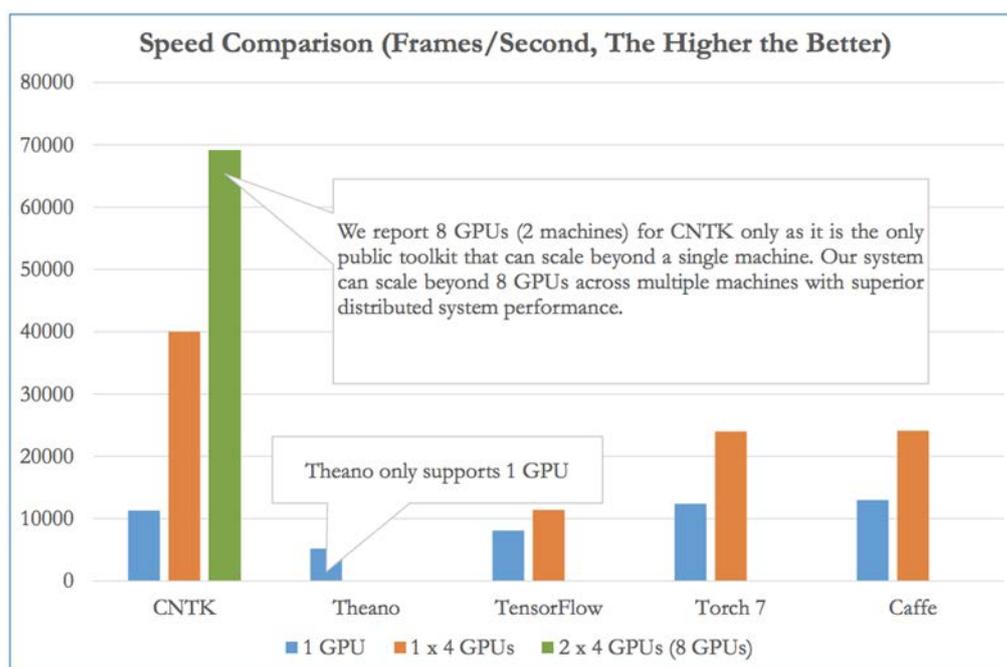
上記の前提条件と調査結果より以下 19 のクラス分けとすることで決定した。

1. Car (車)
2. Sky (空)
3. Roadway (車道)
4. Sidewalk (歩道)
5. Snow Mass (雪の塊)
6. Vegetation (樹木)
7. Person (人)
8. Animal (動物)
9. Building (建物)
10. Traffic Sign (標識)
11. Traffic light (信号)
12. Telegraph pole (電柱)
13. Truck (トラック)
14. Bus (バス)
15. Field (地面)
16. Snow blowing (雪の吹き溜まり)
17. Manhole (マンホール)
18. Motorbike (バイク)
19. Static Object (その他)

今後の課題として、セマンティック・セグメンテーションを実現するためにデータセットを作成する際に上記のクラス分けで足りないものがたくさん出てくる可能性と逆に出てこない可能性が考えられる。データセットを作成しながら、上記クラスからの変更を今後も検討していくことが必要である。

【2. フレームワークと実施環境の調査】

セマンティック・セグメンテーションを実現するためにディープラーニングのフレームワーク調査を行った。調査より、世の中に存在するフレームワークは常に更新されており、各バージョンでパフォーマンスが異なり、利用目的も違うことがわかった。



例えば CNTK は主に音声認識、Caffe は画像処理に使用されることが多い。

その中でも代表的なフレームワークは以下の調査結果より

- CNTK
- TensorFlow
- Caffe
- Keras

であることがわかった。

Aggregate popularity (30•contrib + 10•issues + 5•forks)•1e-3		
#1:	97.53	tensorflow/tensorflow
#2:	71.11	BVLC/caffe
#3:	43.70	fchollet/keras
#4:	32.07	Theano/Theano
#5:	31.96	dmlc/mxnet
#6:	19.51	deeplearning4j/deeplearning4j
#7:	15.63	Microsoft/CNTK
#8:	13.90	torch/torch7
#9:	9.03	pfnet/chainer
#10:	8.75	Lasagne/Lasagne
#11:	7.84	NVIDIA/DIGITS
#12:	7.83	mila-udem/blocks
#13:	5.95	karpathy/convnetjs
#14:	5.84	NervanaSystems/neon
#15:	4.91	tflearn/tflearn
#16:	3.28	amznlabs/amazon-dsstne
#17:	1.81	IDSIA/brainstorm
#18:	1.38	torchnet/torchnet

代表的なフレームワークである、CNTK、TensorFlow、Caffe、Keras を比較すると以下のようになる。

Software	Creator	License	Open Source	Written in	Interface
CNTK	Microsoft Research	MIT License	Yes	C++	Python (Keras), C++, Command line
TensorFlow	Google Brain team	Apache 2.0	Yes	C++, Python	Python (Keras), C/C++, Java, Go, R
Caffe	Berkeley Vision and Learning Center	BSD License	Yes	C++	Python, Matlab
Keras	François Chollet	MIT License	Yes	Python	Python, R

調査より Keras が TensorFlow、CNTK のインターフェースになることがわかった。

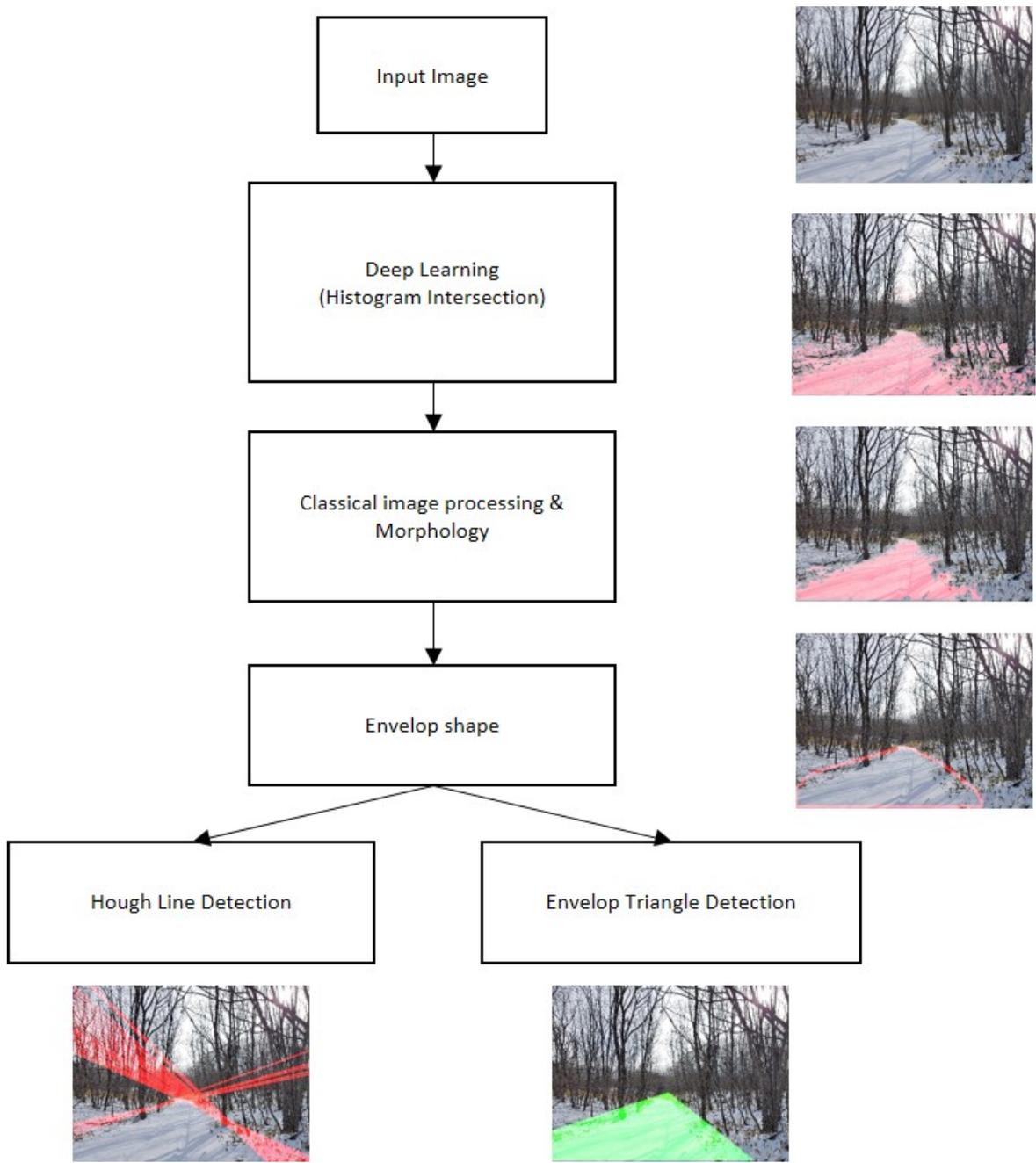
そのため、以下の組み合わせの何れかを使用することができる。

- Keras with backend TensorFlow
- Keras with backend CNTK
- Caffe

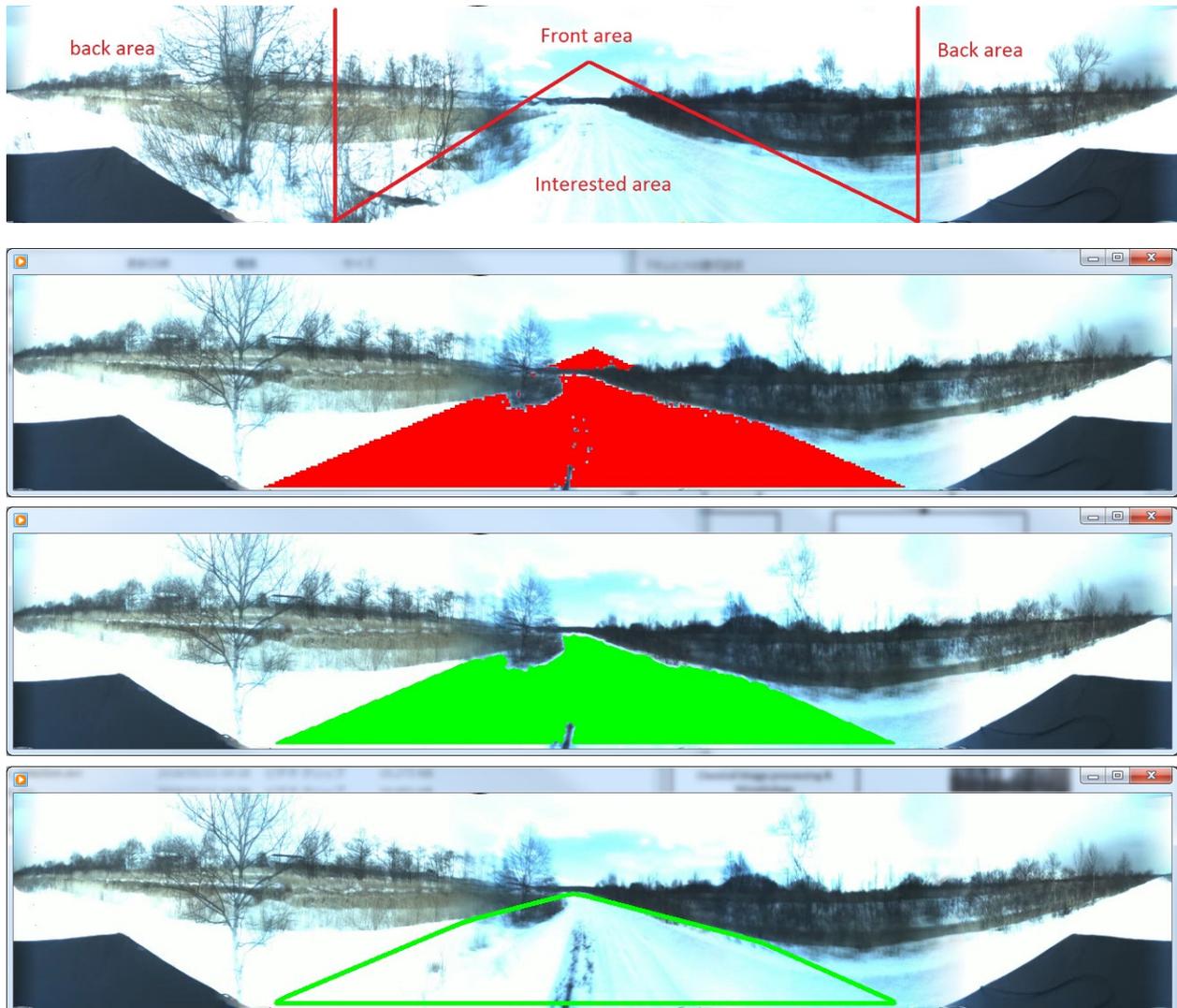
その中で本研究では、まず世の中の情報量が多い、Keras with backend TensorFlow をフレームワークとして選択することとした。

今後の課題として Keras with backend TensorFlow が本研究用とに本当にマッチしているかどうかを見極める必要がある。

以上の試みと平行して、クラスを極端に単純化し、走行可能領域を深層学習によって検出することを試みた。クラスとしては1) snow、2)bush という2つのクラスに絞り、各々の学習画像を25枚ずつ準備して学習を行った。処理の流れを以下の図に示す。



まず、深層学習によって雪である領域を走行可能領域の候補として検出する。その後、クラスタリングを行い、Hough 変換を適用することによりエンベロープの検出を行った。静止画に対して概ね良好な結果が得られたので、omni stereo camera によって得られた動画に対して本手法を適用した。その結果を以下に示す。なお認識を容易にするために以下のようなマスクを用いている。



これらの図は動画から切り出したものであるが、一番上の赤色の部分が深層学習によって認識された雪の部分、その下の緑色の部分がクラスタリングの結果、最後に示したものがエンベロープである。なお、この結果はマスク画像内の「interested area」の作り方に大きく依存している。従って、カーブや交差点、道路に起伏がある場合は環境に応じてこのエリアを動的に変化させる必要がある。さらに、実車を制御するためにはこの2次元的な情報から3次元的な情報を構築する必要があるため、omni stereo camera によって得られる奥行き情報と融合し、車輛の制御ストラテジーを構築することが今後の課題である。

【3. データセット作成調査】

データセット作成のための Annotation Tool の調査を行った。調査したところ世の中にフリーソフトとして使用できる Annotation Tool は以下の5つがあることがわかり、その5つについて調査を行った。

◆LabelMe (online / offline)

online: <https://github.com/CSAILVision/LabelMeAnnotationTool>

offline: <https://github.com/wkentaro/labelme>

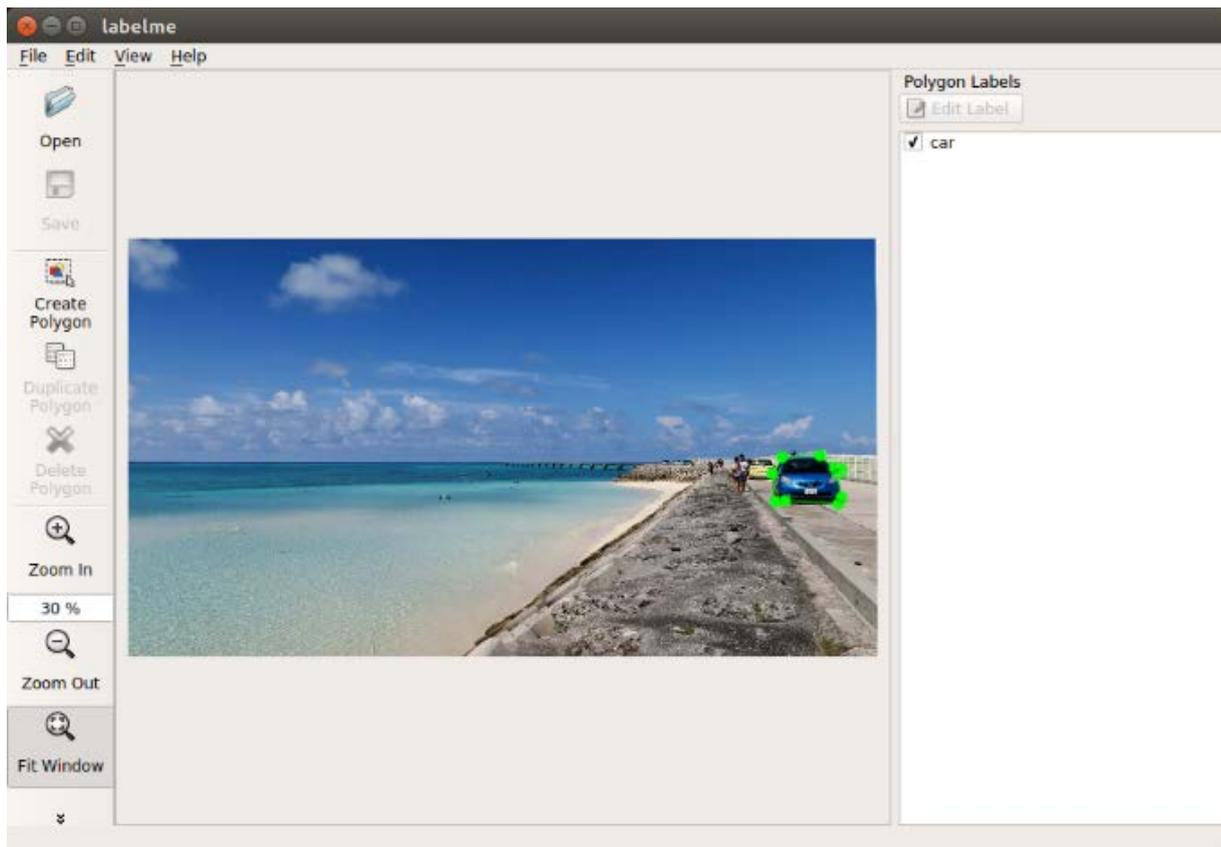
LabelMe online 版は以下の特徴がある。

- ・多機能(マスクツール、バウンディングボックス など)
- ・アカウントを作成し他のユーザーと共有できる。
- ・オンラインツールのため URL を知っているると画像にアクセスできてしまう。
そのため肖像権の問題が発生する。
- ・Github からソースコードを取得すればローカル環境を構築することができる。
- ・ライセンスは MIT ライセンス



LabelMe offline 版は以下の特徴がある。

- ・online 版とは違い公開されることはない。
- ・機能が少ない: ポリゴンのみ
- ・アノテーションファイルに重要なタグ(image size, occluded, id)がない。
- ・Linux 環境が必要
- ・ライセンスは GNU GPL v3



◆Microsoft VOTT

<https://github.com/Microsoft/VoTT>

Microsoft VOTT には以下の特徴がある。

- ・画像、動画にタグ付け可能。
- ・CNTK、Tensorflow(PascalVOC)または YOLO 形式にエクスポートすることが可能
- ・バウンディングボックス機能のみ
- ・オブジェクト検出には最適かもしれない。

◆LabelImg

<https://github.com/tzutalin/labelImg>

LabelImg には以下の特徴がある。

- ・バウンディングボックス機能のみ

◆Ratsnake

<http://is-innovation.eu/ratsnake/index.htm>

RatSnake には以下の特徴がある。

- ・機能は多くない。

- ・LabelMe annotation format をインポート/エクスポート可能
- ・タブを使って同時に多くの画像を扱うことが可能
- ・アノテーションファイルに重要なタグ (Occluded) がない
- ・バウンディングボックス機能がない

◆VGG Image Annotator (VIA)

<http://www.robots.ox.ac.uk/~vgg/software/via/>

VGG Image Annotator (VIA)には以下の特徴がある。

- ・軽量かつオフラインで作業可能
- ・アノテーションファイルにカスタマイズされたタグを追加することが可能
- ・異なる領域の形状をサポート: 長方形、円形、楕円形、多角形
- ・いくつかの重要な機能(名前の削除/リネーム, ポリゴンの削除)がない

結論:

機能として LabelMe online が本研究としてやりたいことができるツールであることがわかったため LabelMe online を使用したいが、オンラインツールのため URL を知っている画像にアクセスできてしまうため肖像権の問題が発生する。そこで LabelMe online のソースコードを GitHub からダウンロードをして、ローカル環境で使用できるように改造する方針とした。

今後の課題としては LabelMe online を使用してデータセットを作成する際に、効率よくデータセットが作成できるように効率の良いデータセットの作成方法の検討し、LabelMe online の改造を視野に入れて研究を進める必要がある。

【4. データセット作成】

3. データセット作成調査にて LabelMe online をローカル環境で使用する方針にしたため、LabelMe online をローカルで環境構築を調査し、環境構築を行った。環境構築方法を以下に示す。

◆事前準備

Label Me のソースコードを以下よりダウンロードする。

<https://github.com/CSAILVision/LabelMeAnnotationTool>

また、Ubuntu 16.04 をダウンロードし、PC にインストールする。

◆環境構築手順

Step 1:

Ubuntu ディストリビューションのアップデートとアップグレードを以下コマンドで実施

```
sudo apt-get update -y
sudo DEBIAN_FRONTEND=noninteractive apt-get -y -o Dpkg::Options::="--force-confdef" -o
Dpkg::Options::="--force-confold" upgrade
```

Step 2:

Label Me 実行に必要な“Apache”, “Perl”, “git”, をインストールし以下コマンドで実施

```
sudo apt-get install -y ¥
apache2 ¥
git ¥
libapache2-mod-perl2 ¥
libcgi-session-perl ¥
libapache2-mod-php ¥
make ¥
php
```

Step3:

以下のように Apache を設定する。

```
sudo a2enmod include
sudo a2enmod rewrite
sudo a2enmod cgi
```

Step4:

/etc/apache2/sites-available/000-default.conf を以下のように編集する。

```
<Directory "/var/www/html/LabelMeAnnotationTool">
    Options Indexes FollowSymLinks MultiViews Includes ExecCGI
    AddHandler cgi-script .cgi
    AllowOverride All
    Require all granted
    AddType text/html .shtml
    AddOutputFilter INCLUDES .shtml
    DirectoryIndex index.shtml
</Directory>
```

Step5:

GitHub から LabelMe をクローンし、LabelMe ファイルを Apache / web-server ディレクトリにコピーする。

```
git clone https://github.com/CSAILVision/LabelMeAnnotationTool.git
sudo cp -r ./LabelMeAnnotationTool/ /var/www/html/LabelMeAnnotationTool/
```

Step6:

Makefile を実行する。

```
cd /var/www/html/LabelMeAnnotationTool/  
sudo make
```

Step7:

LabelMe ファイルのアクセス許可を更新する

```
sudo chown -R www-data:www-data /var/www/html
```

Step8:

Apache を再起動する。

```
sudo service apache2 restart
```

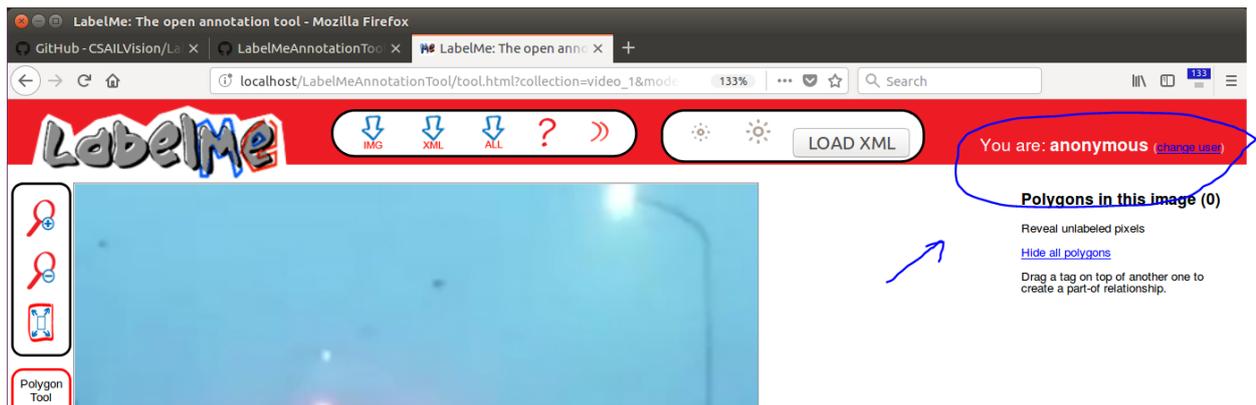
◆使い方

ローカル環境で LabelMe を実行するにはブラウザのアドレスバーに以下のように入力する。

<http://localhost/LabelMeAnnotationTool/tool.html>

◆注意点

編集は同じユーザー名でなければできない。例として X という名前のユーザーが注釈を作成した場合、別のユーザーが同じ画像を編集するためには右上のユーザー名も X でなければならない。



【5. 他の手法調査】

他の手法として SegNet を使用したセマンティック・セグメンテーションを実現するために、セマンティック・セグメンテーション実行環境のセットアップを以下の手順で実施した。

- ・開発者用ソフトウェア・スイート "Jetpack" のインストール
- ・機械学習用ライブラリ "Tensorflow" のインストール
- ・ニューラルネットワークライブラリ "Keras" のインストール
- ・SegNet 雪道画像のセマンティック・セグメンテーションの実行

◆開発者用ソフトウェア・スイート "Jetpack" のインストール

Jetpackとは、NVIDIA が提供するcuDNN,CUDA等の開発者用のツールやライブラリのソフトウェア・スイートのことである。

以下の手順でインストールを完了した。

Host-PC を用意

OS:Ubuntu14.04

Oracle VirtualBox5.2 を利用

Host-PC に Jetpack 3.1 ホスト用ツールをインストール(<https://developer.nvidia.com/embedded/jetpack>)

TX2 に Jetpack をインストール

VirtualBox 経由では USB 接続の認識に失敗するため、LAN ケーブルを接続し、ネットワーク経由でインストールを実施。

◆機械学習用ライブラリ "Tensorflow" のインストール

"Tensorflow" とは、Google が開発しオープンソースで公開している、機械学習に用いるためのソフトウェアライブラリである。

以下の手順でインストールを完了した。

参考: <https://syed-ahmed.gitbooks.io/nvidia-jetson-tx2-recipes/content/first-question.html>

Java, Python のインストール

Google が提供するビルドツール "Bazel" のインストール

apt-get ではインストールに失敗するため、上記参考サイトの手順に従ってソースを入手しコンパイルを実施。

メモリスワップの有効化

Tensorflow のビルドの実行時にメモリが不足するため、swapon コマンドを使用してメモリスワップを有効にする。

ビルド時に作成される中間ファイルによりストレージ容量が不足するため、SD カードをマウントし、スワップファイル(8G)は SD カード内に作成。

Tensorflow のビルド

◆ニューラルネットワークライブラリ "Keras" のインストール

"Keras" とは、Python で書かれたオープンソースニューラルネットワークライブラリである。TensorFlow, Theano 等の上部で動作し、TensorFlow のコアライブラリ開発からサポートされている。

pip コマンドによってインストールを実施し、問題なく完了した。

◆SegNet 雪道画像のセマンティック・セグメンテーションの実行

実施内容

- ・ SegNet 実行環境にて、学習用雪道イメージデータを学習する

【条件】

学習用雪道イメージデータ(360×480pix) : 74 件、Classes : 17

【学習処理】

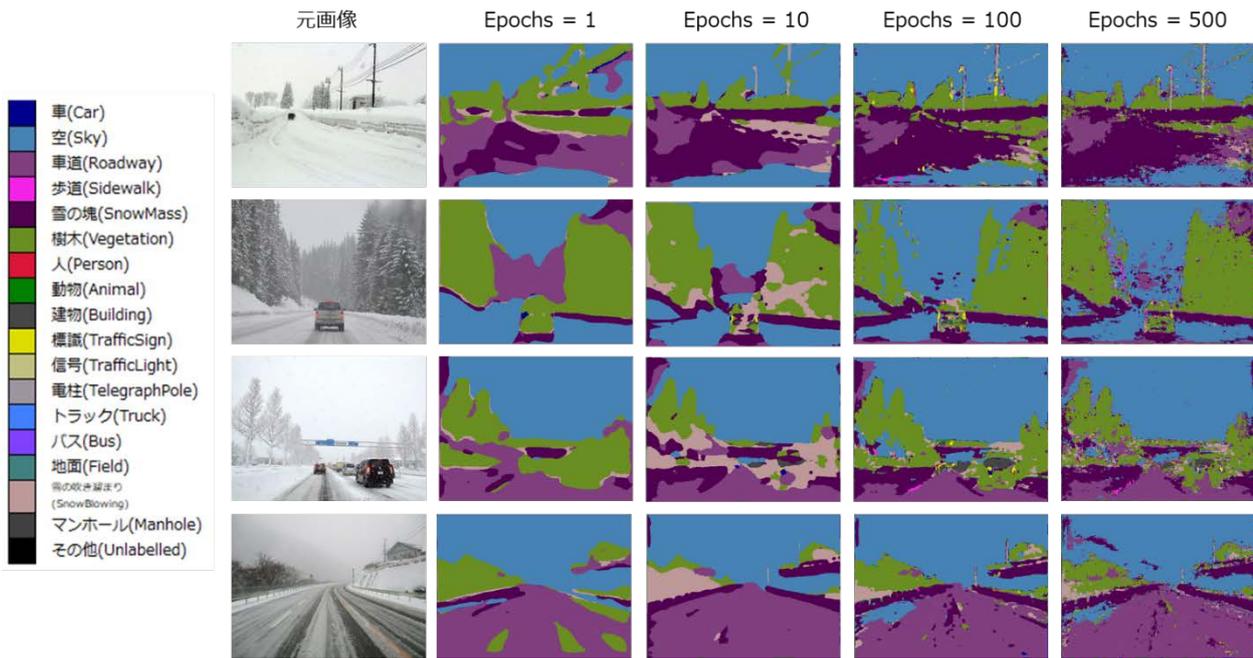
Epochs = 1/10/100/500 の 4 パターンで実施

- ・ 雪道画像 8 パターンを入力し予測結果を得る
- ・ 予測実行時間を計測する

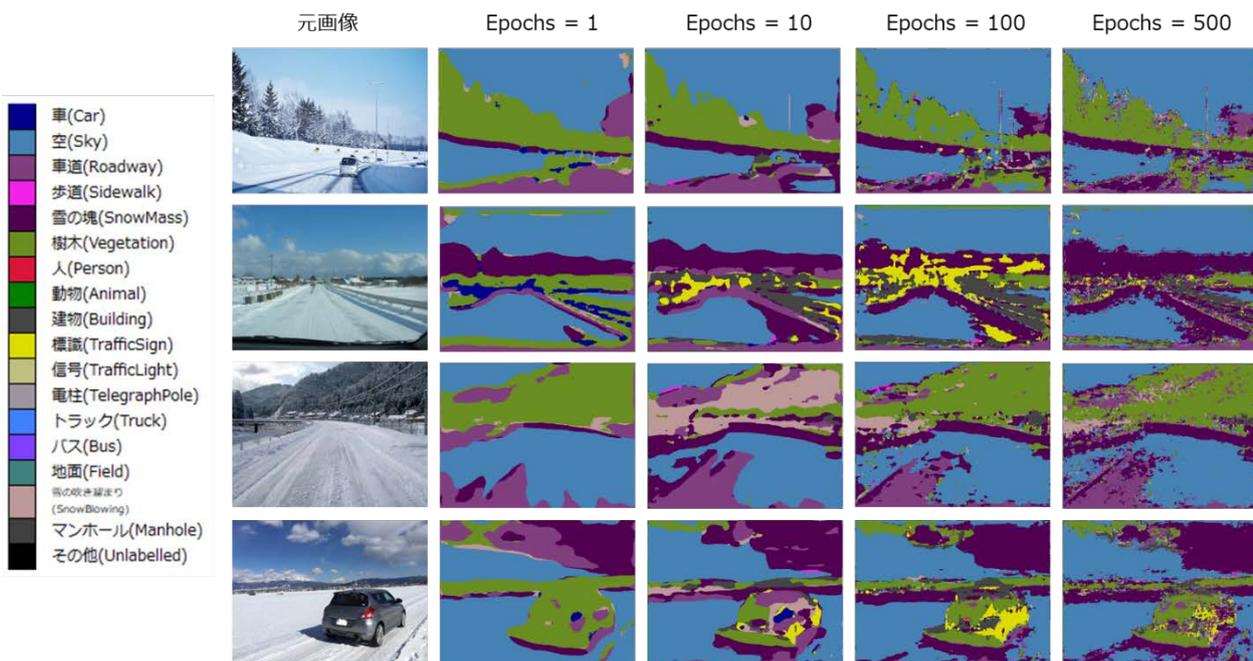


学習用雪道イメージデータの例

予測結果 1



予測結果 2



予測実行時間

	Epochs = 1	Epochs = 10	Epochs = 100	Epochs = 500
イメージデータ(360 × 480pix) : 74件	0.961 sec	1.041 sec	1.018 sec	0.908 sec

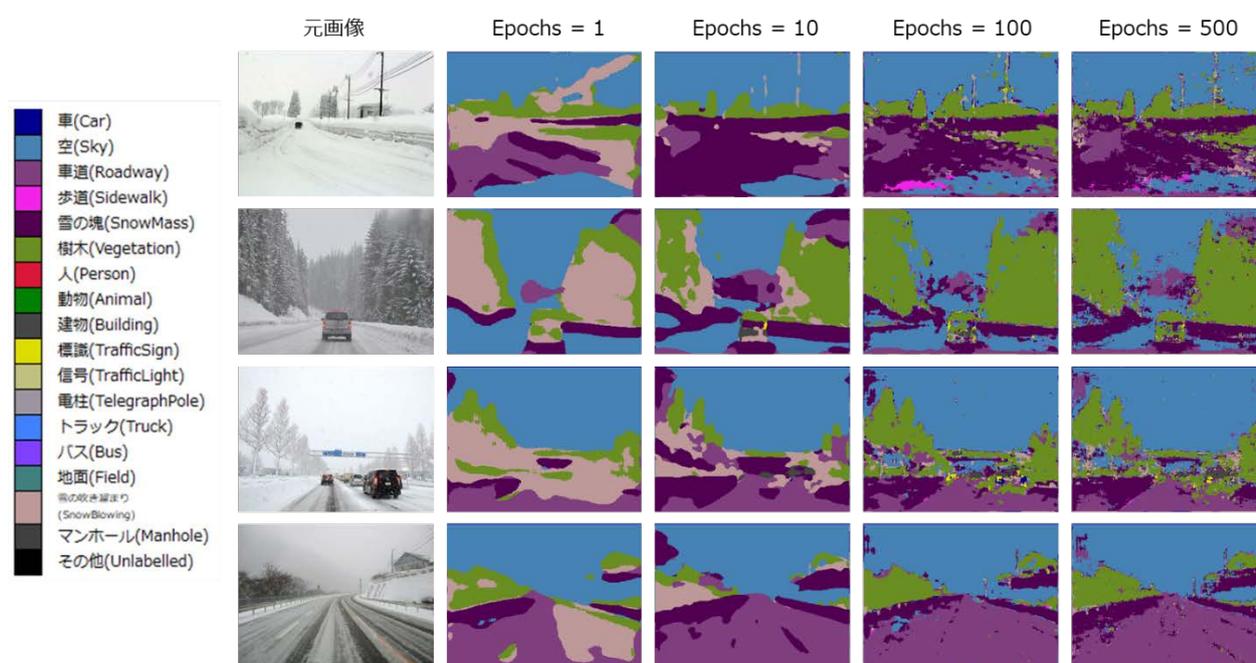
- ・イメージデータ(360 × 480pix)の予測実行時間は、0.908 ~ 1.041 sec
- ・予測実行時間と学習回数(Epochs)に関係は見られない

イメージデータサイズ 360 × 480pix → 240 × 360pix、学習用イメージデータ件数 74→37 件へ変更して実行した結果

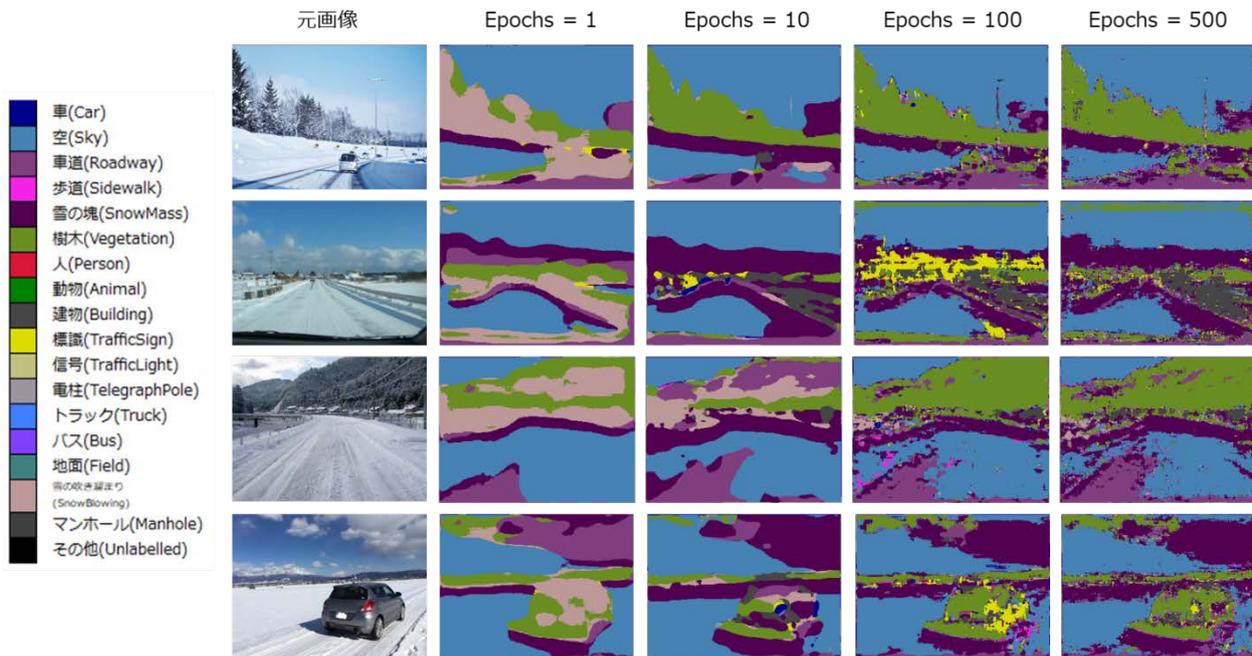
	Epochs = 1	Epochs = 10	Epochs = 100	Epochs = 500
①イメージデータ(240 × 360pix) : 74件	0.740 sec	0.778 sec	0.810 sec	0.848 sec
②イメージデータ(360 × 480pix) : 37件	0.945sec	1.067 sec	0.984 sec	0.967sec

- ・①イメージデータ(240 × 360pix)の予測実行時間は、0.740 ~ 0.848 sec
→ イメージデータ(360 × 480pix)と比べて、実行時間が小さくなった
- ・②イメージデータ(360 × 480pix)の予測実行時間は、0.945 ~ 1.067 sec
→ イメージデータ(360 × 480pix)と比べて、実行時間が小さくなることは確認できない

予測結果 1 (240 × 320)



予測結果 2 (240 × 320)



結果

- ・ 車道と空の予測結果に正しくない場合があった
 - 車道 → 空、または雪の塊と予測する場合があった
 - 空(にある雲)→雪の塊と予測する場合があった
- ・ 予測実行時間は 0.908 ~ 1.041 sec であった

今後の課題として予測結果で正しくない場面が見られるので学習するデータセットのパターンを増やして学習すること、予測実行時間を短くするための方法の検討が必要である。

2-1-2-(3) 2年度の目標

雪道対応セマンティック・セグメンテーション技術の開発について 30 年度は、以下の目標に向けて活動を実施。

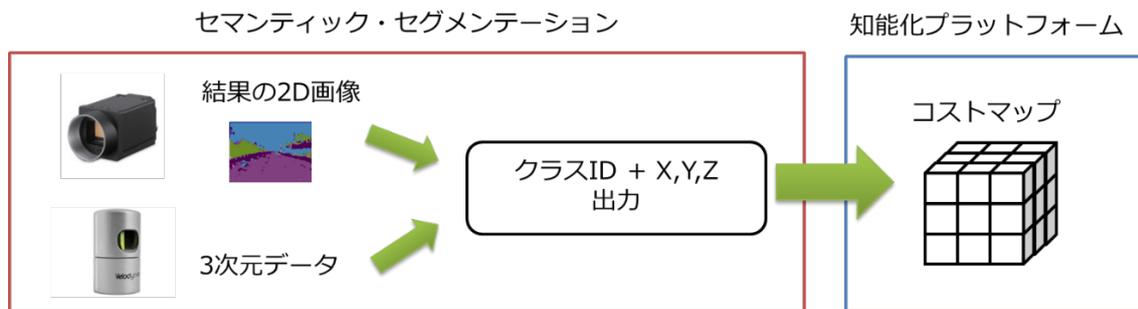
・30 年度の目標

1. ROS 上でのセマンティック・セグメンテーション実現と実装
2. 認識速度の向上検討
3. データセットの作り方再検討
4. データセット作成及び検証

2-1-2-(4) 具体的な内容

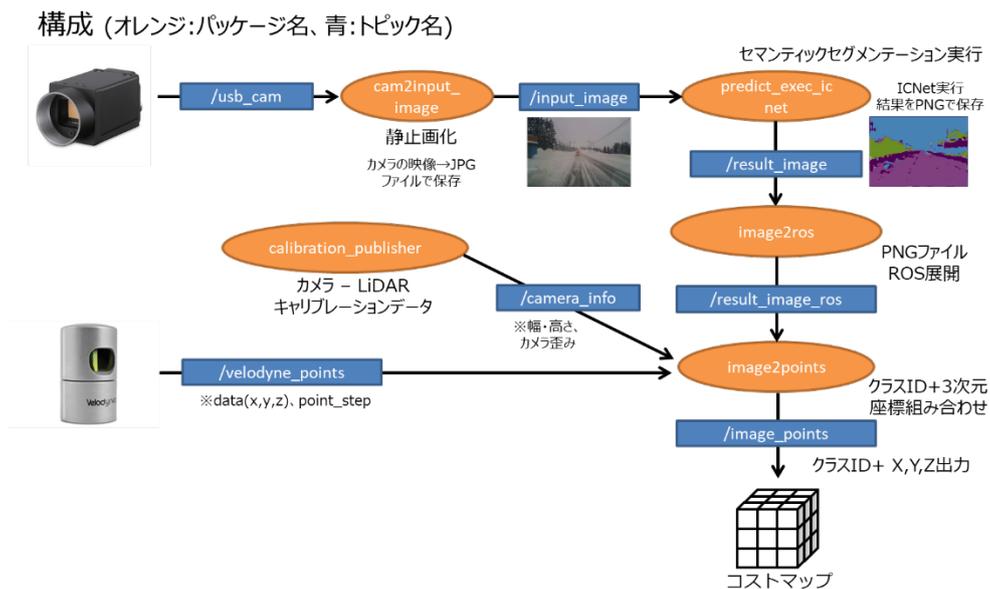
【5. ROS 上でのセマンティック・セグメンテーション実現と実装】

ROS 上でのセマンティック・セグメンテーション (ICNet) を実現した。結果の 2D 画像と 3 次元データを組み合わせて、各画素の識別されたクラス ID と 3 次元座標を出力し、コストマップを管理する智能化プラットフォームへ渡すところまで実装を行った。



ROS 上セマンティックセグメンテーションのイメージ

ROS セマンティック・セグメンテーション構成を以下に示す。

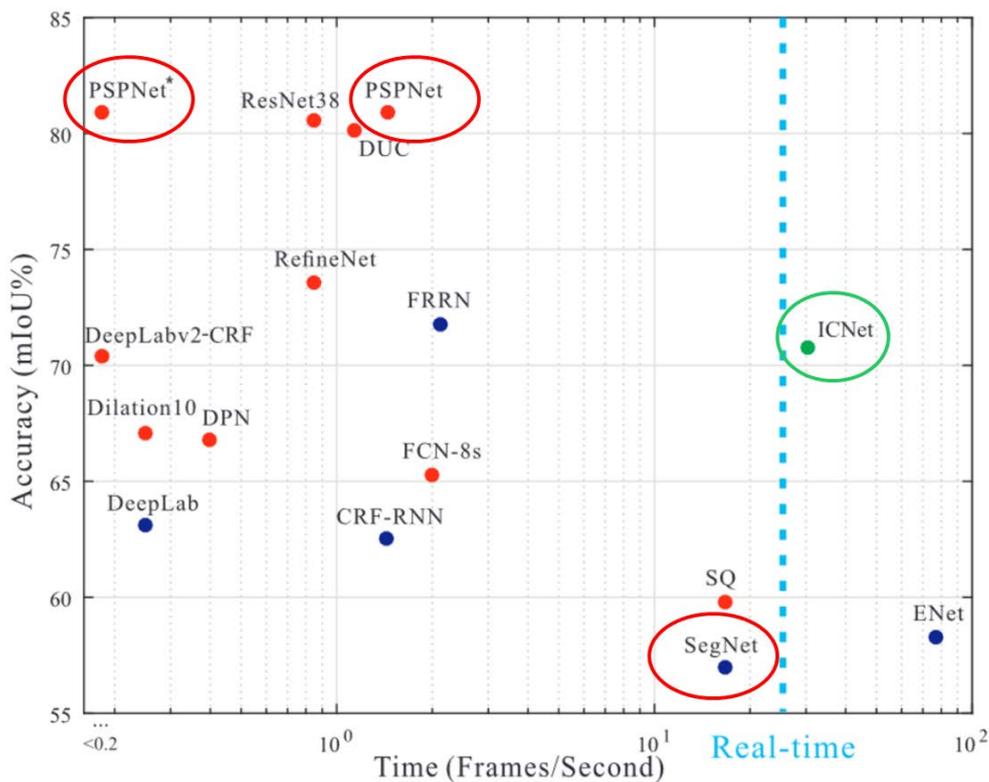


ROS 上でのセマンティック・セグメンテーションの構成

【6. 認識速度の向上検討】

昨年度の課題として「予測実行時間を短くするための方法の検討が必要」となっていた。そこで昨年調査した手法、FCN, SegNet, PSP-Net 以外の手法を Web サイトから論文の調査を実施した。論文の中に昨年見つけきれなかった ICNet という手法があることがわかり、ICNet について調査を行った。

ICNet for Real-Time Semantic Segmentation on High-Resolution Images (<https://arxiv.org/pdf/1704.08545.pdf>) によると ICNet は Accuracy が 70%以上の性能が出せ、リアルタイム性が担保できるという情報が書かれており、昨年、検証した PSP-Net よりは Accuracy は劣るが、リアルタイム性が確保できそうであることがわかったため、ICNet の調査と実装を行った。



また、Web サイトで ICNet を使い、1024x2048 のイメージをリアルタイムに変換している動画もアップされており、ICNet が本プロジェクトに適合しそうであることが推測できた。



<https://www.youtube.com/watch?v=qWI9idsCuLQ>

ICNet の調査を進めるとオープンソースで公開されており、またフレームワークとしては Tensorflow が使えることがわかったため、パラメータファイルを調整し実装を行った。

```
# If you want to apply to other datasets, change following four lines
DATA_DIR = '/PATH/TO/CITYSCAPES_DATASET'
DATA_LIST_PATH = './list/cityscapes_train_list.txt'
IGNORE_LABEL = 255 # The class number of background
INPUT_SIZE = '720, 720' # Input size for training

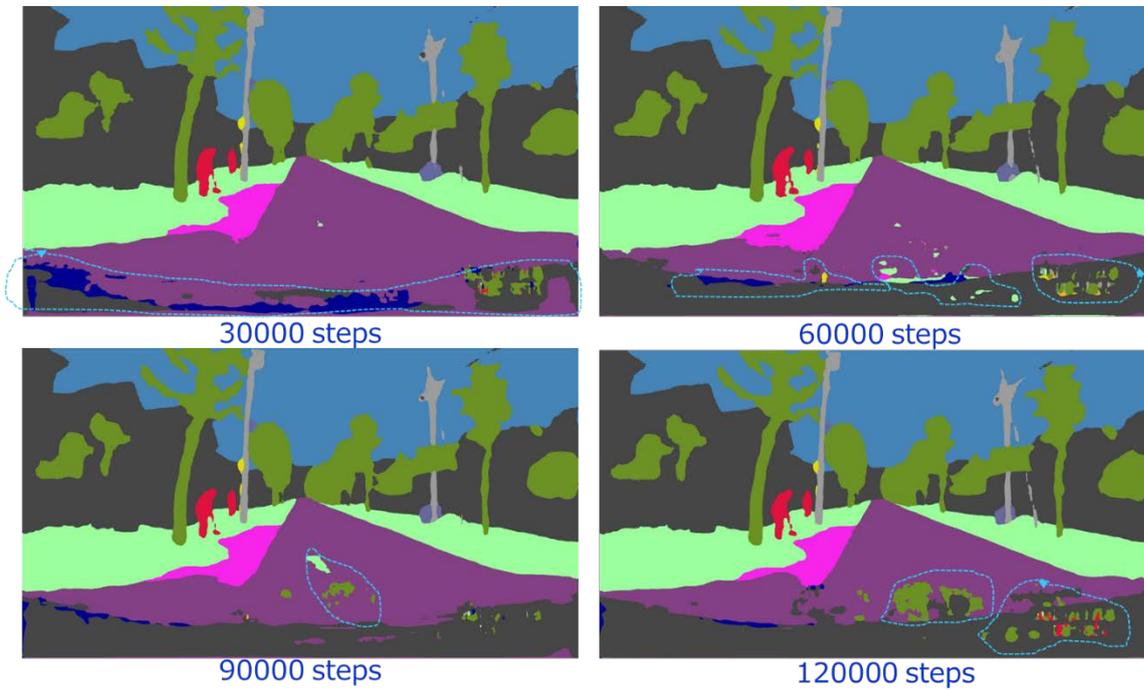
BATCH_SIZE = 16
LEARNING_RATE = 1e-3
MOMENTUM = 0.9
NUM_CLASSES = 19
NUM_STEPS = 60001
POWER = 0.9
RANDOM_SEED = 1234
WEIGHT_DECAY = 0.0001
PRETRAINED_MODEL = './model/icnet_cityscapes_trainval_90k_bnnomerge.npy'
SNAPSHOT_DIR = './snapshots/'
SAVE_NUM_IMAGES = 4
SAVE_PRED_EVERY = 50

# Loss Function = LAMBDA1 * sub4_loss + LAMBDA2 * sub24_loss + LAMBDA3 * sub124_loss
LAMBDA1 = 0.16
LAMBDA2 = 0.4
LAMBDA3 = 1.0
```

Parameters in train.py file

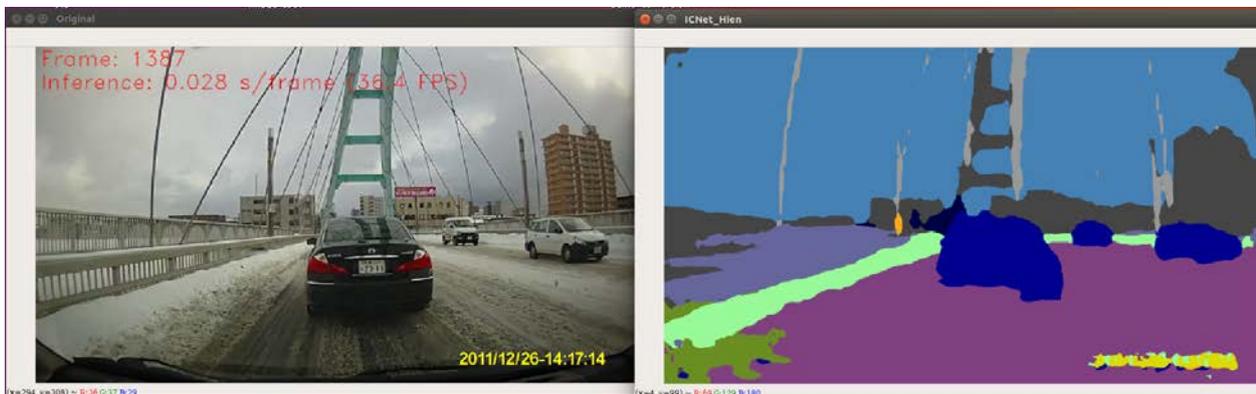
実装した ICNet を使用して変換についてのテストを行った。テスト結果を以下に示す。

・Step 数による変換の違い



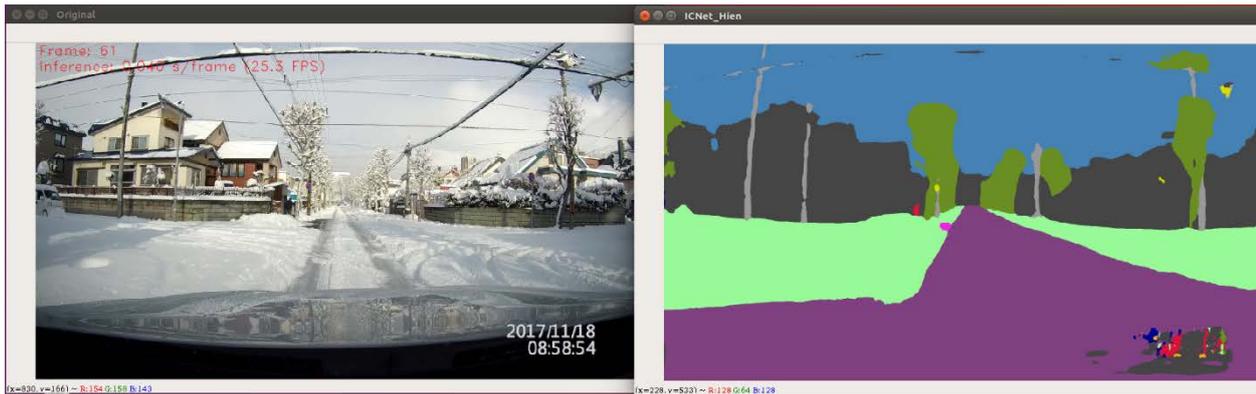
Step 数による変換の違いは大きく見られなかった。

・変換速度



For video: 1280x720

Semantic segmentation processing speed: 36.4 ~ 48.8 FPS



For video: 1920 x 1080

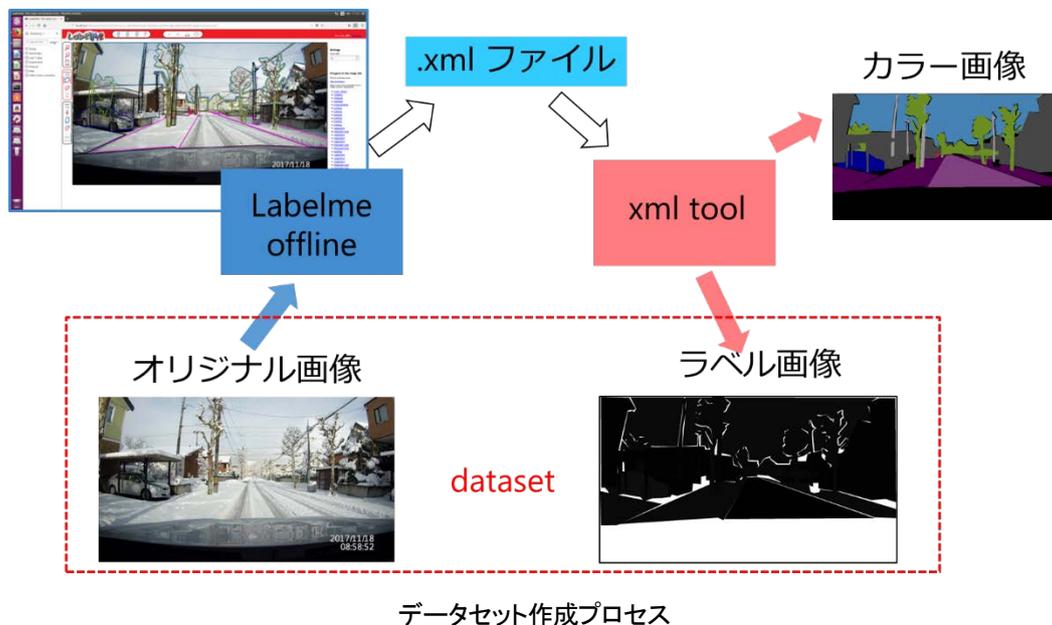
Semantic segmentation processing speed: 18.6 ~ 25.5 FPS

参考: 同環境で PSPNet の場合は 4.3 ~ 4.5 FPS

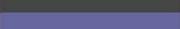
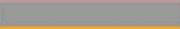
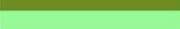
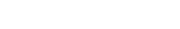
テスト結果より、FCN、SegNet、PSP-Net に比べて ICNet の処理速度は格段に速いため今後の研究は ICNet を使うことがよいということがわかった。

【7. データセットの作り方再検討】

昨年度の課題として「LabelMe online を使用してデータセットを作成する際に、効率よくデータセットが作成できるように効率の良いデータセットの作成方法の検討し、LabelMe online の改造を視野に入れて研究を進める必要がある」というものがあった。そこで、データセットのプロセス再検討とクラス定義、コンフィグ値の見直し、カラーID の見直しを行うこと、またデータセットをいかに効率よく増やすかの検討を行った。



データセットの作成プロセスを以下のように定義し、データセット作成者に迷いなくデータが作れるプロセスを定義した。また、コンフィグ値、各クラスの ID とカラーも再定義し、作成者毎にデータセットの作成が異なったり割り振る ID が異なることがないようにした。

ID	TrainID	Name	R	G	B	Colored	Gray scale for Train ID
1	0	roadway	128	64	128		
2	1	sidewalk	244	35	231		
3	2	building	69	69	69		
4	3	snow blowing	102	102	156		
5	4	snow mass	190	153	153		
6	5	telegraph pole	153	153	153		
7	6	traffic light	250	170	29		
8	7	traffic sign	219	219	0		
9	8	vegetation	106	142	35		
10	9	field	152	250	152		
11	10	sky	69	129	180		
12	11	person	219	19	60		
13	12	animal	255	0	0		
14	13	car	0	0	142		
15	14	truck	0	0	69		
16	15	bus	0	60	100		
17	16	manhole	0	79	100		
18	17	motorbike	0	0	230		
19	255	static object	56	36	36		
20	255	unlabeled	0	0	0		

クラス ID とカラー値

データセットを効率よく増やすためにm、データセット量産ツールを作成することを検討及び実装した。

・データセット量産ツール仕様

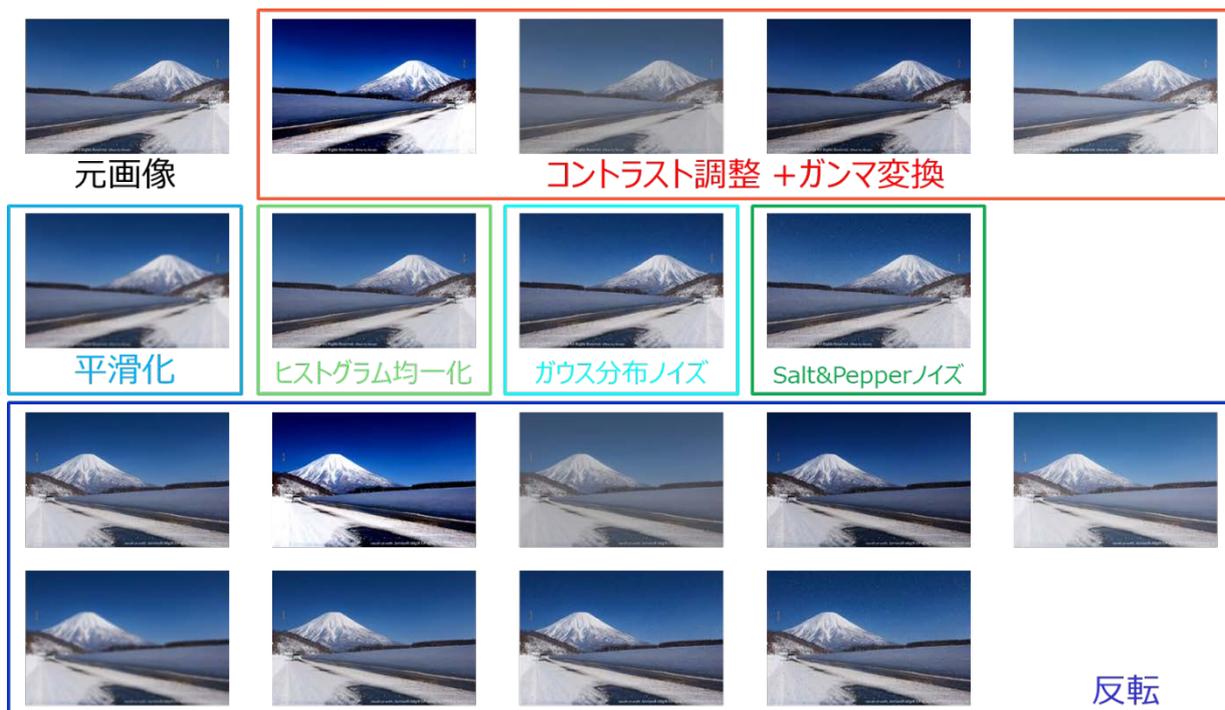
① 1枚の画像に対し、以下加工を施し18枚の画像を生成する

- ・コントラスト調整(強調、低減)
- ・ガンマ変換
- ・平滑化(ヘイカツカ)
- ・ヒストグラム均一化
- ・ガウス分布に基づくノイズ付加
- ・Salt&Pepper ノイズ付加
- ・反転

② ①で生成した画像をトリミングし、1枚につき4枚の画像を生成する

上記①及び②により1枚の画像から $18 \times 4 = 74$ 枚の画像が生成するツールを実装する。

環境として OpenCV 3.3.0、Python 3.5.2 を使用する。



データセット量産ツールの実行例

【8. データセット作成及び検証】

3. データセットの作り方再検討で作成したデータセットに対し検証を行った。

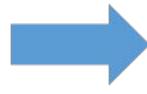
データセット量産ツールにかけたデータセットについて以下に示す。



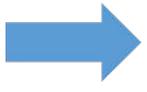
原画像



コントラスト1



コントラスト2



コントラスト3



コントラスト4



平滑化



ヒストグラム



ガウス分布



Salt&Pepper

データセット量産ツールを用いてデータセットを 18 倍に増やすことが可能であることがわかった。

次に mIoU を用いて変換精度についての調査を実施した。mIoU は、セマンティック・セグメンテーションアルゴリズムのパフォーマンスを評価するための標準指標であり、以下の式で求める。

$$IoU = \frac{Target \cap Prediction}{Target \cup Prediction}$$

今回の研究で用いている Tensorflow では、tf.contrib.metrics モジュールを使用して mIoU を計算が可能である。

`mIoU, update_op = tf.contrib.metrics.streaming_iou(pred, gt, num_classes=param['num_classes'])`

実際に1枚の画像を評価した結果が以下となる。

```

Restored model parameters from ./snapshots/Hien_ICNet_16_115_30K/model.ckpt-3000
0
evaluation: 100% | ██████████ | 1/1 [00:01<00:00, 1.34s/it]
IoU: [0.9857203 0. 0.9227694 0.9688166 0. 0.693464
0. 0. 0.8595467 0. 0.95039654 0.36743674
0. 0.7869062 0. 0. 0. 0.
0. ]
dpc22010@dpc22010:~/Desktop/ICNet-tensorflow_Hien$

```



Class_0	roadway		0.9857201
Class_1	sidewalk		0.0
Class_2	building		0.9227694
Class_3	snow blowing		0.968816
Class_4	snow mass		0.0
Class_5	telegraph pole		0.693464
Class_6	traffic light		0.0
Class_7	traffic sign		0.0
Class_8	vegetation		0.8595467
Class_9	field		0.0
Class_10	sky		0.95039654
Class_11	person		0.36743674
Class_12	animal		0
Class_13	car		0.7869062
Class_14	truck		0.0
Class_15	bus		0.0
Class_16	manhole		0.0
Class_17	motorbike		0.0

Percentage of accuracy

Roadway: 98.57%
Building: 92.28%
Snow blowing: 96.88%
Telegraph pole: 69.34%
Vegetation: 85.95%
Sky: 95.04%
Person: 36.74%
Car: 78.69%



これらの実装まで完了したため、来年度はセマンティックセグメンテーションの認識率について調査する準備ができた。

2-1-2-(5)3年度の目標

雪道対応セマンティック・セグメンテーション技術の開発について令和元年度は、以下の目標に向けて活動を実施。

・令和元年度の目標

1. サーモカメラによるセマンティックセグメンテーションの検討
2. サーモカメラによる轍検出と実装
3. サーモカメラによる轍検出検証（冬データ）

また、他センサーとの調停機能を円滑に行うためグリッドマップ方式の開発及び実装を実施した。

2-1-2-(6)具体的な内容

【9. サーモカメラによるセマンティックセグメンテーションの検討】

これまでの活動ではサーモカメラを雪上路面に使用したデータが少ないため、

- ・ 雪上環境におけるサーモカメラの特性理解
- ・ サーモカメラを用いた走行可能領域検出手法の検討

の2点を夏季期間に実施した。

先行研究として、近赤外線を用いた凍結および圧雪路面の評価を国立研究開発法人土木研究所 寒地土木研究所のチームが行っており、その結果によると圧雪の方が反射強度は高く（下グラフ右軸）

湿潤や凍結では1200～1400nmの光に対して反射強度をもっている。

今回の対象となる轍は圧雪であると考えられるため、圧雪とそれ以外の差異が反射強度の差が大きいレンジが良い観測波長と推定される。

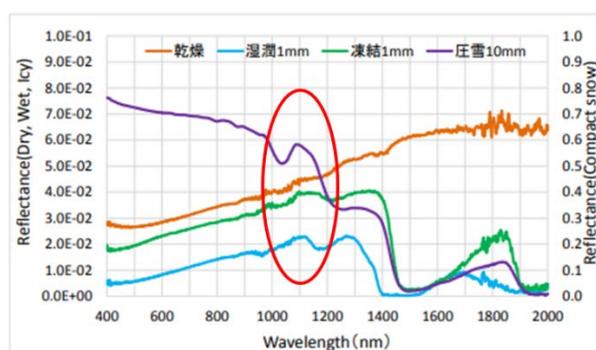
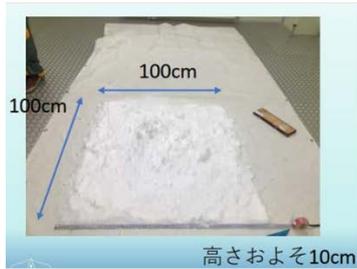


図3 各路面状態の計測結果比較（野外、65°）

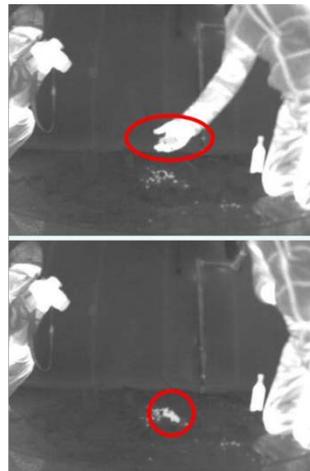
近赤外線を用いた凍結および圧雪路面の評価について
(国立研究開発法人土木研究所 寒地土木研究所 寒地交通チーム)

論文より各路面状態の計測結果

上記を踏まえて、2019年8月8日～9日にかけて北海道立総合研究機構工業試験場様の低温実験室を使用し、雪の代わりにかき氷を使用した実験を実施。サーモカメラの特性を検証した。



低温実験室におけるサーモカメラの特性検証の様子



低温実験室におけるサーモカメラの特性検証の結果画像

左図は体重 80 キロの人間が足で踏んだ後、数分後周囲と同じ色になり周囲と区別が出来なくなった。右上図は疑似雪を握りしめたところ。右下図のように放置したところ。20 分後には同様に周囲と区別が困難になった。上記から以下のことが判明した。

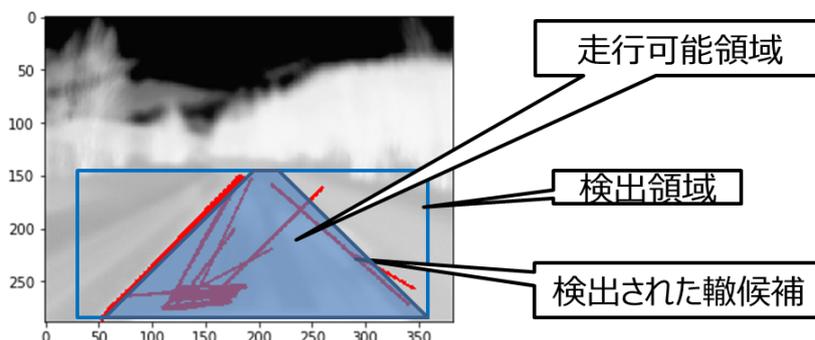
- 疑似雪（かき氷）と雪で大きく異なるのは結晶性。雪の方が結晶構造を持つため、空気を大きく含む可能性がある。
- 轍に見えていたのは圧縮されてできた水の可能性が高い。
- サーモカメラでは通常のカメラより情報量が少ないため、セマンティックセグメンテーションがうまく動かない可能性がある。
- 温度測定結果自体を活用するだけでセマンティックセグメンテーションを使用しなくても轍を元にした走行可能領域は検出可能である。

今回の検証および計算リソースなどを含めて総合的に考え、サーモカメラ画像をハフ変換（直線抽出）し路面走行可能領域の抽出を目指す方針に決定した。

【10. サーモカメラによる轍検出と実装】

サーモカメラの画像における轍検出では、OpenCV を活用した解析的な画像処理、ハフ(Hough)変換を用いた直線検出を

採用した。ハフ変換では白黒画像に2値化してパラメータに適合する直線を検出する処理である。
 今回の轍検出では下図のように検出した線と線の間を走行可能領域として設定することで、セマンティックセグメンテーションや Lidar の結果を補完することを目指した。



【11. サーマカメラによる轍検出検証（冬データ）】

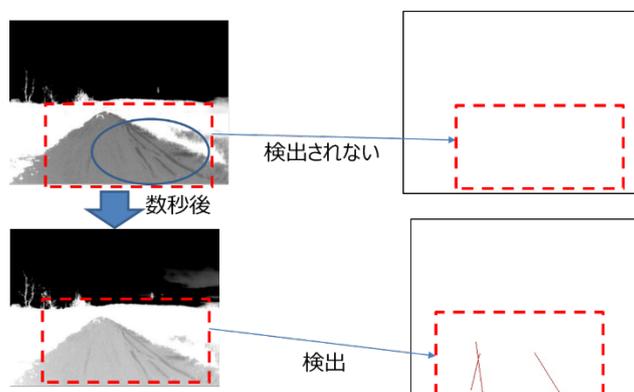
開発した轍検出モジュールを実証実験内で検証を行った。

実際に行った実験当日の天候は以下の通り

実験①: 2019年12月17日 芦別: 曇りのち雪。

実験②: 2020年1月22日 苫東: 晴れ

実験③: 2020年2月19日 芦別・士別: 晴れ



実験結果

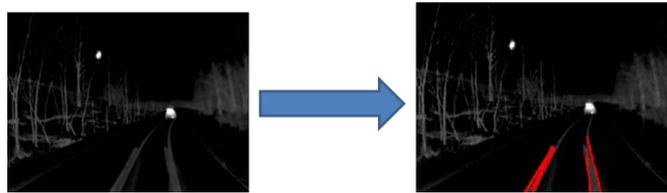
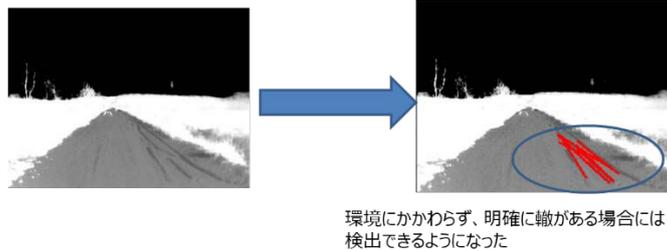
実験①の時点では、これまでのサーモカメラ画像を元に各種パラメータの調整を行っていた。しかし検証を進めていく中で、様々な条件下により轍検出を行うための2値化を決定するパラメータが環境により異なるため、様々な環境に合わせたユニバーサルなパラメータ決定では対応が難しくなってしまうことが分かった。例えば上記実験結果のように数秒の差異であっても轍検出結果が異なる結果となった。

そのため、今回のモジュールでは以下のようなロジックで動的にフィルターのパラメータ決定を行う動的フィルターの開発

を実施した。

- 検出対象領域の最大値、最小値を 0~255 にマッピングし置き換え
- 高温の領域は人や障害物である可能性が高いため、一定以上の温度として検出したものについては全て0として扱う。(サーモカメラの設定による)
- 平坦化処理を実施しノイズの影響を削減

上記処理の実行により、様々な状況下であっても轍検出が可能となった。



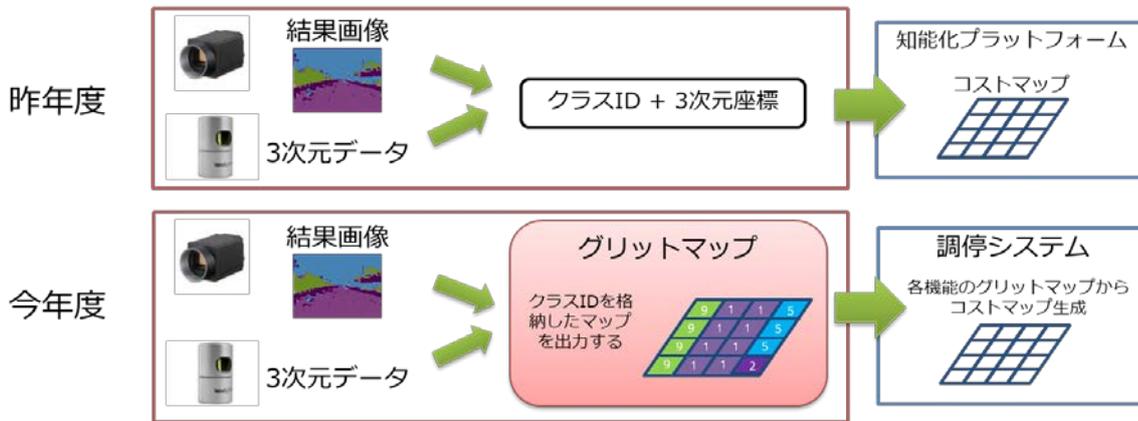
まとめると今回の検証では以下のことが言える。

- サーモカメラ自体、日差しや気温などの外乱要因による影響を非常に受けやすいことが分かった。
- そのため、状況により轍など対象物の状態が変わると特徴を捉えることが困難なケースが存在する。
- 今回様々な環境状態に適応できるよう、簡易的ではあるが動的フィルターロジックを開発・実装し、検出精度の改善を実現。(しかしながら、サーモカメラによる轍検出はあくまで補助である。元データで取得できない場合は機能しない。)
- サーモカメラを用いて、轍検出以外のオブジェクト検出の開発を行う場合、外乱要因による影響を抑制する機構を持たせる必要がある。→今回の場合、動的フィルター開発により実現。

【12. グリッドマップの作成】

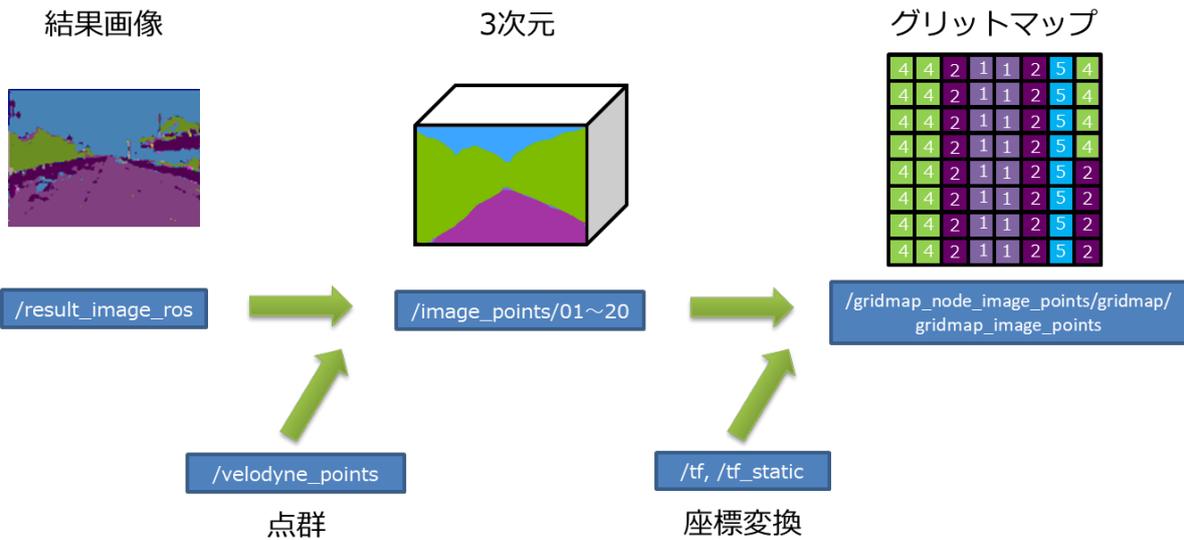
昨年度、ROS 上でのセマンティック・セグメンテーション(ICNet)を実現した。結果画像と3次元データを組み合わせて、各画素を識別したクラス ID と 3 次元座標を出力し、コストマップを管理する知能化プラットフォームへ渡すところまで実装した。

セマンティック・セグメンテーション

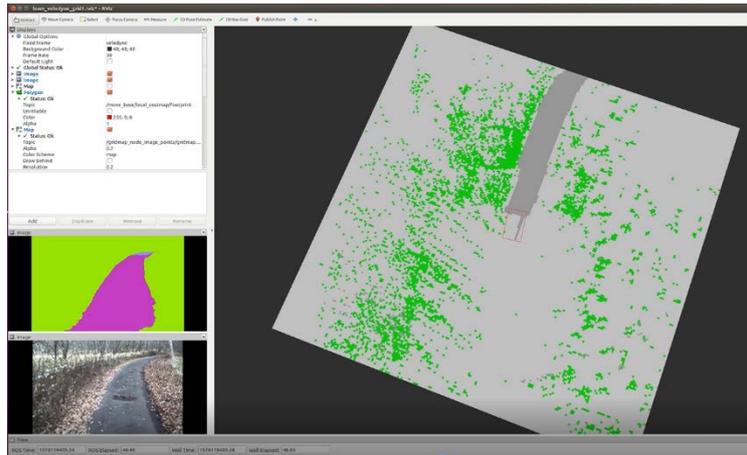


今年度は、「調停システム」の追加により、セマンティック・セグメンテーションの結果を「グリッドマップ」として出力する変更をおこなった。

グリッドマップについて説明する。グリッドマップは車両を中心とする160×160の格子状の地図である。セマンティック・セグメンテーションで分類したクラスID(1~20)を、Lidarで得られる点群とマッチングして3次元座標化し、TFを用いて地図に配置する事を行っている。



これにより、調停マップ側で特別な座標変換やキャリブレーションといった処理を行わずに車両上部からみた情報として処理が可能となる。調停側の詳細な処理については調停処理の説明の項で述べる。



実際の ROS 上でのグリッドマップキャプチャ

【13. グリッドマップの補間】

本来の Lidar 情報は点群であり球状にセンシングを行っている。そのため、セマンティックセグメンテーションの結果をそのままマッピングすることで、本来道として検出してはいけない場所を道として検出するケースが発生した。そこで、グリッドマップに配置されたセマンティックセグメンテーション結果の同一 ID で特定の範囲にあるものを繋ぎ、マップの粗の部分を補う処理を実装した。

処理対象とする ID は、苫東/芦別のコースをセマンティックセグメンテーションした結果、道の両脇に多く登場する以下とし、これらを繋いで道の両端を壁の様に、経路生成時のロジックに反映させるものとした。

対象とするクラス ID:

- 吹き溜まり (クラス ID = 4)
- 樹木 (クラス ID = 9)

処理の概要は以下のとおりである。

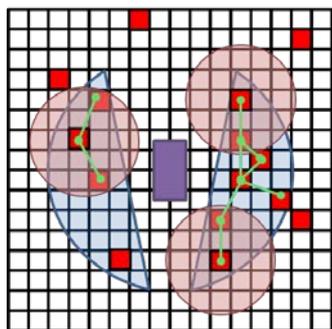
① 範囲の特定

マップ上の繋ぐ範囲を特定する。車両の右と左は繋がらない、waypoint の邪魔にならない範囲とする。



② 対象 ID の座標抽出と周辺検索

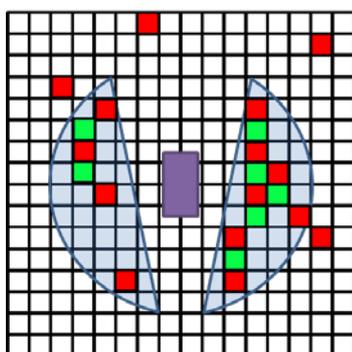
対象 ID の座標を抽出し、それぞれの座標の半径 r の中に同一 ID が存在するか検索する。



- : 車両
- : 対象IDのセル

ピンク：検索範囲(緑線は繋がる予定)

③ 範囲内で見つかった ID の座標を繋ぐ
範囲内で見つかった同一 ID 間のセルを埋めて繋ぐ。



- : 車両
- : 対象IDのセル
- : 対象ID間を繋いだセル

これにより、下図のように Lidar 情報が疎である場合でも正しい経路情報を決定し、調停処理側に適切なグリッドマップを提供することが可能となった。

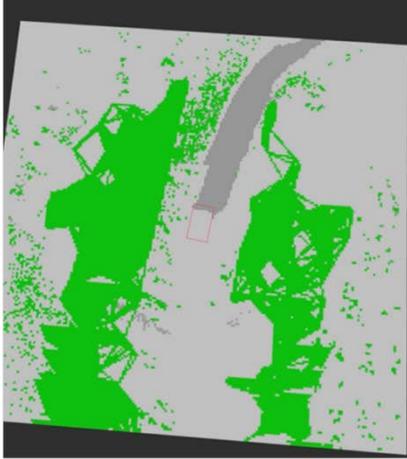
カメラ画像



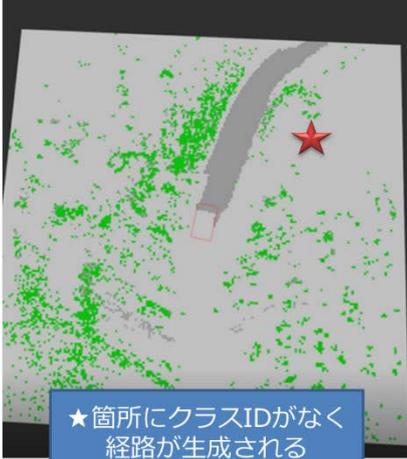
セマンティックセグメンテーション結果



補間あり



補間なし



★箇所クラスIDがなく経路が生成される

2-1-3. 雪道対応センサ・フュージョン技術の開発

2-1-3-(1)初年度の目標

・29年度の目標

1. 初年度用センサの選定
2. ROS 上でのデータ収集環境構築

2-1-3-(2)具体的な内容

【1. 初年度用センサの選定】

実証実験及び予備実験において、以下のセンサによってデータの収集を行った。

- 3次元 LiDAR Velodyne VLP-16
- 3次元 LiDAR Velodyne HDL-32e
- 3次元 LiDAR 北陽電機 YVT-35LX
- Depth カメラ Intel RealSense Euclid
- Depth カメラ Intel RealSense D415
- Depth カメラ Intel RealSense D435
- 2次元 LiDAR 北陽電機 UST-20LX
- 2次元 LiDAR 北陽電機 UTM-30LX
- 赤外線サーモグラフィ optris PI450
- 全方位ステレオ OccamVisionGroup OmniStereo
- 防水アクションカメラ SJCAM SJ7 STAR
- ビデオカメラ Sony HDR-CX680 TI
- 360度全方位カメラ FLIR 社 Ladybug5+
- GPS (iMAR ユニット等一式)
- Texas Instruments ミリ波レーダー、IWR1642

Snow-SLAMを実現するために、LiDAR-SLAMとVisual SLAMの基礎的な検証を実施することが初年度の目的であった。LiDAR-SLAMに適用するためのセンサとして、Velodyne社のVLP-16およびHDL-32e、さらに北陽電機のYVT-35LXについて検討を行った。北陽電機の製品はVelodyneと比較して距離レンジが短いこと、さらにはスキャン範囲が360度ではないこと、またVelodyneと併用すると何らかの干渉のような現象が見られることからVelodyneをメインとして利用することとした。VLP-16とHDL-32eの違いはレイヤーの数の違いであり、VLP-16は16層、HDL-32eは32層の情報量を持っている。この違いはSLAMを適用すると大きな違いとなって現れた。特に今年度実証実験を実施した苦東のように人工構造物(信号、ガードレール、建物、自動車、看板等)がほとんど無い環境では、LiDARによって得られる特徴量が非常に少ない。このような環境でLiDAR-SLAMを行うためにはセンサによって得られる情報量がSLAMの精度に大きく影響を及ぼすため、HDL-32eをメインのLiDARとして選択した。また、レンジの短い3次元距離センサとして3種類のIntel RealSenseによって予備実験を実施した。屋外で利用する場合にはD435が最も精度のよい測定を行えることが分かったが、レンジが短いこともあり、LiDARと組み合わせて利用することが必須である。

2次元 LiDARとして挙げた2つの北陽電機製の UST-20LX および UTM-30LX は測距原理としては3次元 LiDARと同じであるが、それぞれ反射波の反射強度、さらにはマルチエコーの情報を取得することが可能である。これらの特徴を利用することにより、降雪状況を監視するために利用できると考えられる。また、LiDAR-SLAMとして現在 Loam Velodyne を適用しているが、雑音の影響によってきれいなマップが得られておらず、この雑音処理を行うための重要な情報として反射波強度およびマルチエコーの情報をを用いることを現在検討中であり、翌年度の実証実験で導入すべく研究開発を継続する。

Visual-SLAM およびセマンティック・セグメンテーションに適用するための画像情報を取得するためのセンサとして、OccamVisionGroup 社 OmniStereo、SJCAM 社 SJ7 STAR、ビデオカメラ Sony 社 HDR-CX680TI、FLIR 社 Ladybug5+の検証を実施した。まず、Visual-SLAM に適用するための画像センサとして OmniStereo をメインに検証した。このカメラは、360度全方位の画像情報のみならず、ステレオカメラになっていることから深度情報も同時に獲得することができる。

Visual-SLAM を適用するためには十分なデータを得ることができたが、自動運転の分野で一般的に利用されているソフトウェア環境「Autoware」で利用可能な高解像度カメラを導入し、比較検討を実施することが今後の研究開発に必須だろうと考え、FLIR 社 Ladybug5+を導入した。セマンティック・セグメンテーション用の画像を取得するためのカメラとして、SJCAM 社 SJ7 STAR および Sony 社 HDR-CX680TI を導入している。一般公道を走行することができない Viking VI のみならず、公道を走行できる自動車に搭載した状態でデータを取得するためには、様々な自動車に容易に取り付けられるカメラが必須であり、その用途として SJCAM 社 SJ7 STAR を利用している。SJCAM 社 SJ7 STAR を搭載する場合はフロント部分に固定して撮影を実施しているが、搭載した状態で様々な角度の画像情報を撮るために Sony 社 HDR-CX680TI を利用している。

ミリ波センサ、サーモカメラは実証実験によってデータの取得を実施し、ミリ波センサについては相対速度が比較的精度よく得られている、サーモカメラについては走行路の轍に温度変化があり環境認識の補助として利用可能である、といった定性的な評価を得ることができた。翌年度、10月までに定性的な評価を実施し、Snow-SLAM に反映させるべく研究を進める。

GNSS(GPS)については、2018年4月よりみちびきとして知られている QZSS(Quasi-Zenith Satellite System)の本格的な運用が始まるとアナウンスされており、RTK-GPS 等を用いた位置精度の向上などには取り組まなかった。QZSS の開発を実施している三菱電機様に QZSS のデモを実施していただき、みちびきの本格運用が始まれば十分な精度が得られるのではないかと感触は得た。我々の目指すところはダイナミックマップ方式とは異なり、グローバルな位置精度が必須というわけではない。しかしながら、安価に自己位置精度を向上させることで Snow-SLAM の精度を向上させることが可能なことはいうまでもなく、三菱電機から最新の情報を収集しながら研究開発を進める必要がある。

【2. ROS 上でのデータ収集環境構築】

「【1. 初年度用センサの選定】」にて選定したセンサを使って、ROS 上にデータを収集できる環境を構築する。

選定したセンサは ROS に対応しているものであり、環境を構築するにあたりそれぞれ必要パッケージをインストールすることにより、使用可能となる。

使用可能とした後はセンサ毎にターミナルにコマンドを入力することにより計測可能。

以下、センサ毎のコマンドをまとめた表を示す。

センサ	コマンド
-----	------

2-1-3-(3)2年度の目標

雪道対応センサ・フュージョン技術の開発について30年度は、以下の目標に向けて活動を実施。

・30年度の目標

1. 知能化プラットフォーム育成(センサ・フュージョン)
2. 知能化プラットフォームとグローバル経路計画との連携

2-1-3-(4)具体的な内容

【1. 知能化プラットフォーム育成(センサ・フュージョン)】

車両制御プラットフォームの構築

gmapping と movebase により、車両の制御を実施。

位置測定は、GPS 端末(IMAR)を使用し、障害物検知は LIDAR(Velodyne:HDL-32E)を使用。

平地にて、指定した経路を走行し、障害物を回避するレベルまで完成したが、以下の問題が発覚した。

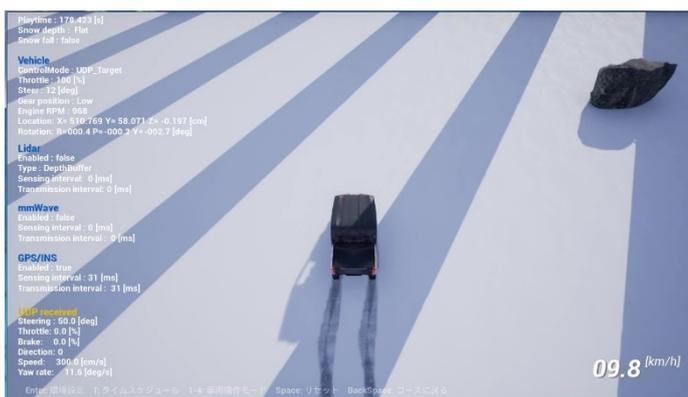
- ・蛇行走行
- ・障害物の誤検知
- ・カーブでの走行経路異常

①蛇行走行

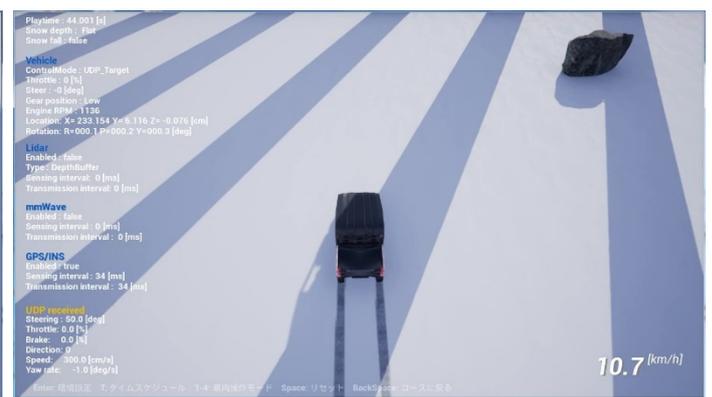
WaiPoint-WaiPoint 間を走行するときに、本来、直進するはずが、蛇行する現象が発生。

原因は、走行位置の計算にて、GPS 端末から取得した緯度・経度を UTM 座標に変換して使用すると、数値が大きいため、経路計算で誤差が発生するためであることが判明。

緯度・経度から UTM 座標に変換後、スタート地点からのオフセットに変換し、その値を使用して経路計算するように変更し、直進するように修正。



改善前

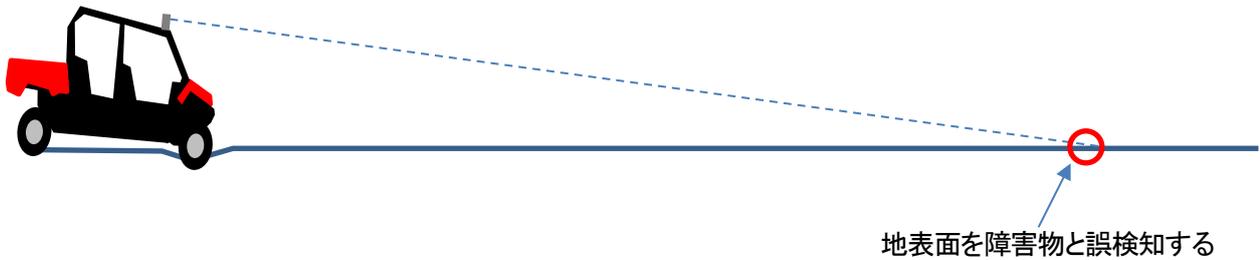


改善後

②障害物の誤検知

gmapping では、2D の障害物情報を使用することが前提となっているが、使用している LIDAR は 3D の情報を出力するため、3D-2D 変換を行ってから使用していた。

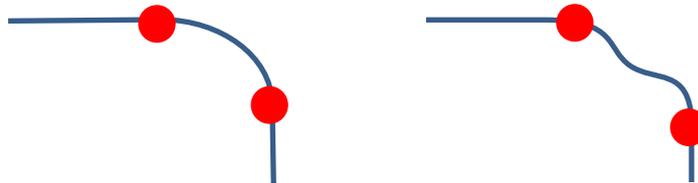
この 3D-2D 変換にて、路面の凹凸によって、車両の姿勢変化があった場合に、地表面を障害物として誤検知する現象が発生することが判明。



3D-LIDAR を使用しているにも関わらず、2D の障害物情報を扱う gmapping を使用するのでは、LIDAR の性能を活用できないことから、3D の情報を扱う Cartographer に変更することでこの問題を回避。

③カーブの走行経路異常

WaiPoint を2つ設定したカーブにおいて、理想的な経路は下図左側のようになるが、実際には、下図右側のような軌跡で走行する。(赤丸が WaiPoint)

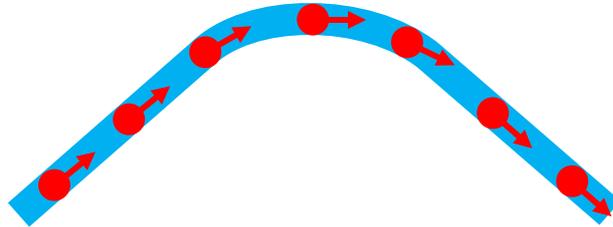


走行制御モデルとして車輪移動対向2輪型を使用していたが、使用している車両はステアリング型であるため、適切な制御、および、走行経路生成ができない状況となっていた。

このため、走行制御モデルをステアリング型に変更した。



走行制御モデルの変更に伴い、WaiPoint 通過時の車両の走行方向を指定可能となり、サブテーマ3(グローバル経路計画)から取得する WaiPoint 情報に通過時の走行方向を追加することで、カーブでの走行経路を理想的な走行ラインに設定することが可能となった。



準天頂衛星システムの利用

GPSによる測位位置を、地上に設置した固定局からの補正情報を使用してセンチメートル級の測位精度を実現するRTK方式を使用した測位端末を使用していたが、この方式では、固定局との無縁通信の圏内でのみ、高精度な測位が可能で、通信圏外では、通常のGPSの測位精度(数m)になってしまう。

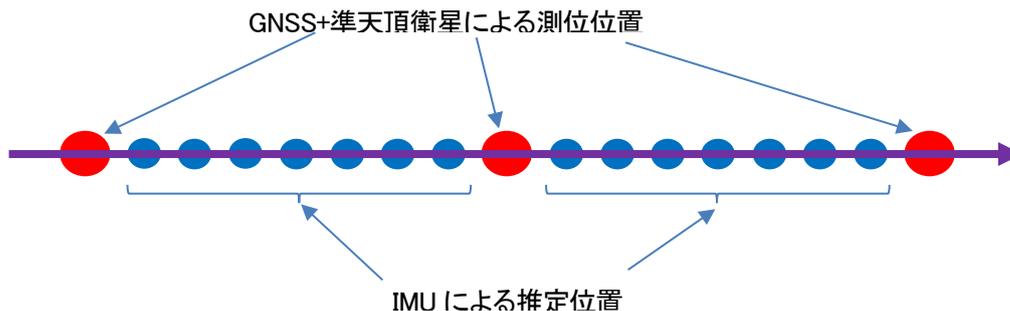
このため、広範囲を走行するシステムには不向きであることから、準天頂衛星(みちびき)からの補正情報を使用する測位端末(AQLOC)に変更した。

この変更により、GNSSによる測位が可能となり、従来のGPS衛星のみを使用した測位に比べて、衛星捕捉数が増加し、より高精度の測位が可能となった。

IMUによる自己位置推定

車両が移動する位置を、IMU(3軸加速度センサ+3軸角速度センサ)からの情報で推定する機能を実装した。

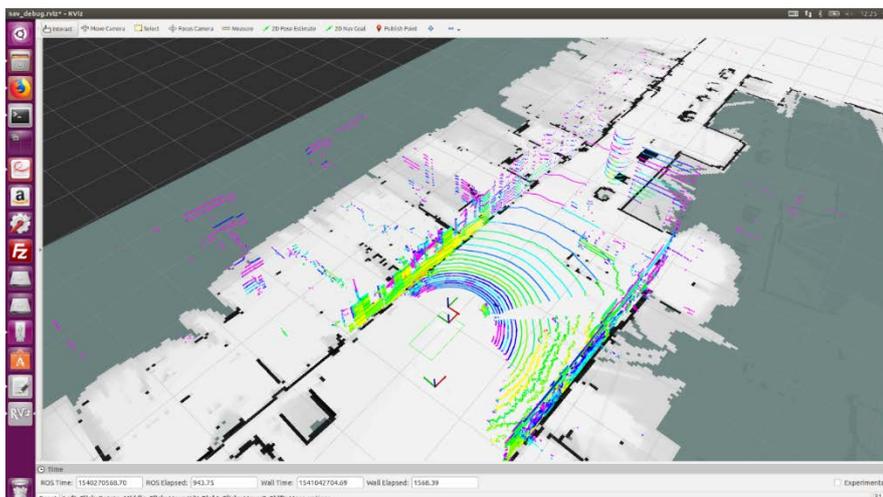
これにより、GNSS+準天頂衛星による測位データを取得する間隔(100mmsec)の間も、車両の位置を推定することが可能となり、走行安定性を向上することが可能となった。



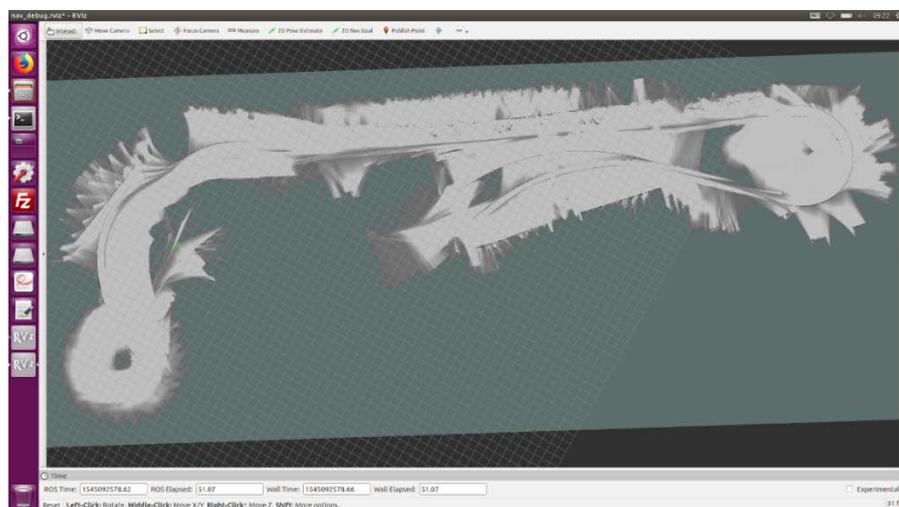
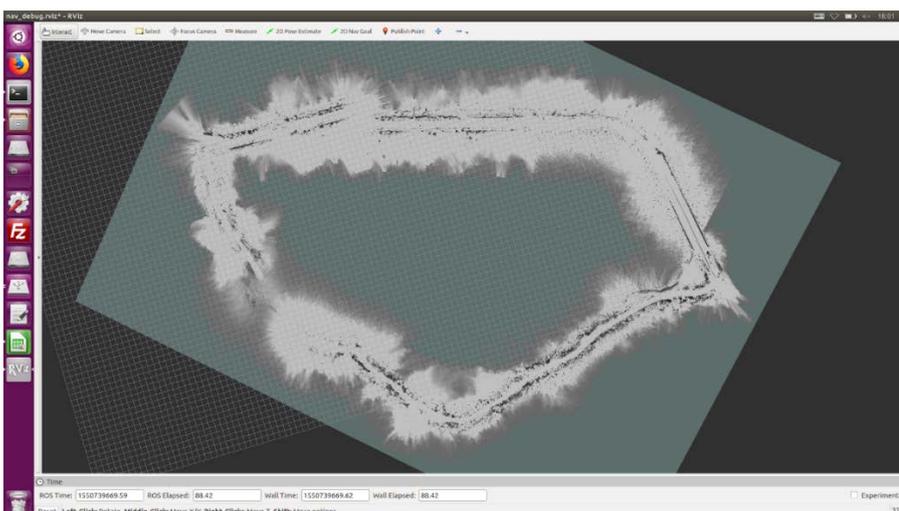
また、GNSS衛星捕捉数の低下、および、準天頂衛星を補足できない環境(建物の陰、樹木の下など)にて、GNSSの測位精度が低下した場合、測位位置が大きくずれることがあるが、このような場合にIMUによる推定位置を利用することで、誤差の大きな測位情報による経路の逸脱を防止することが可能となる。

LIDAR-SLAM

LIDAR-SLAMとして、Cartographerを実装し、自己位置推定、障害物回避を行うためのコストマップを作製。

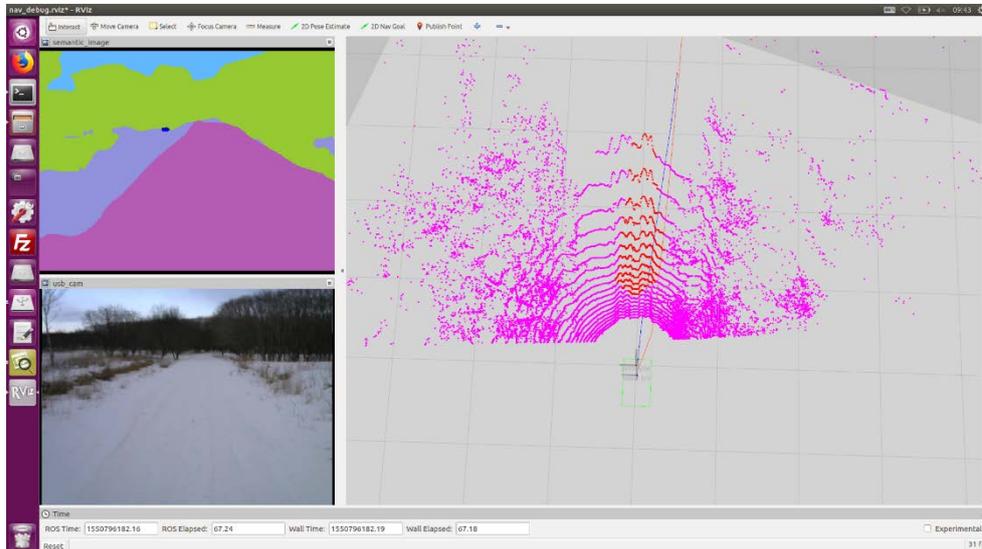


建物など、特徴点の多い場所では、地図生成が行えるが、今年度、走行実験を行った、苦東周回コース、および、北日本自動車大学校テストコースでは、特徴点が少なく、正常な地図を生成できなかった。



セマンティック・セグメンテーションによる走行可能箇所検出

セマンティック・セグメンテーションによりセグメンテーションを行った、対象物のうち、道路と認識した部分を、LIDAR より取得した点群情報と組み合わせることで、道路をコストマップに反映することで、道路以外を走行不可能な箇所として



SLAM とセマンティック・セグメンテーションとの連携

セマンティック・セグメンテーションから得られた結果を SLAM に渡す方法について調査、検討を行った。

【論文調査】

セマンティック・セグメンテーションから得られた結果を SLAM に渡す方法について、以下の論文を調査した。

論文 1. Season-Invariant Semantic Segmentation with A Deep Multimodal Network

『深層マルチモーダルネットワークを用いた季節不変セマンティック・セグメンテーション』

http://www.fsr.ethz.ch/papers/FSR_2017_paper_23.pdf

論文 2. VoxNet: A 3D Convolutional Neural Network for Real-Time Object Recognition

『VoxNet: リアルタイム物体認識のための 3 次元畳み込みニューラルネットワーク』

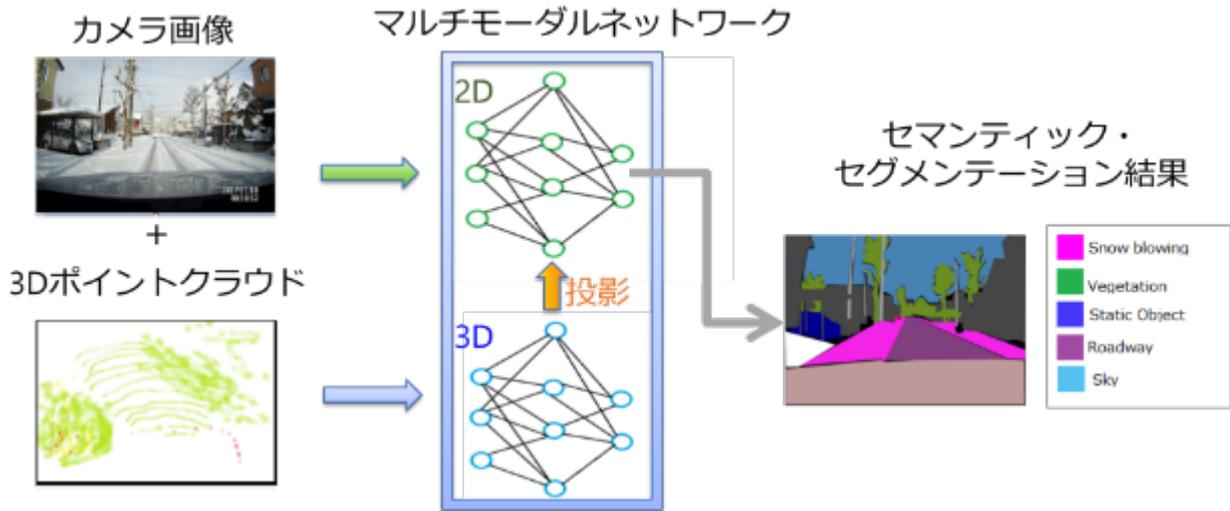
https://www.ri.cmu.edu/pub_files/2015/9/voxnet_maturana_scherer_iros15.pdf

論文 3. Joint Semantic Segmentation and 3D Reconstruction from Monocular Video

『セマンティック・セグメンテーションと単眼映像からの 3D 復元の結合』

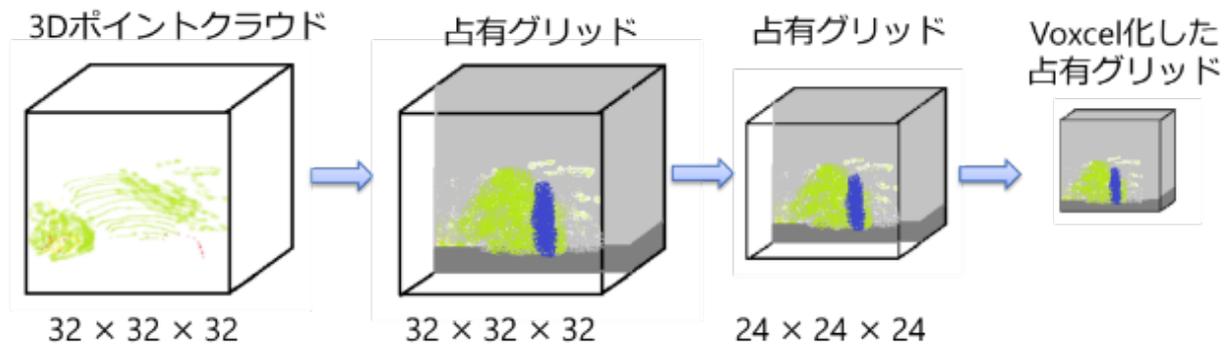
<https://www.cc.gatech.edu/~dellaert/ftp/Kundu14eccv.pdf>

論文 1 では、LiDAR の 3 次元特徴をカメラ画像に投影して、セマンティック・セグメンテーションを実施する。LiDAR データは、照明・天気および季節で生じる外観変化に影響を受けないことの記載があった。



LiDAR の 3 次元特徴をカメラ画像に投影[論文 1]

論文 2 では、3D ポイントクラウドデータを高速かつ正確に処理し、オブジェクト認識ができる VoxNet アーキテクチャの記載があった。



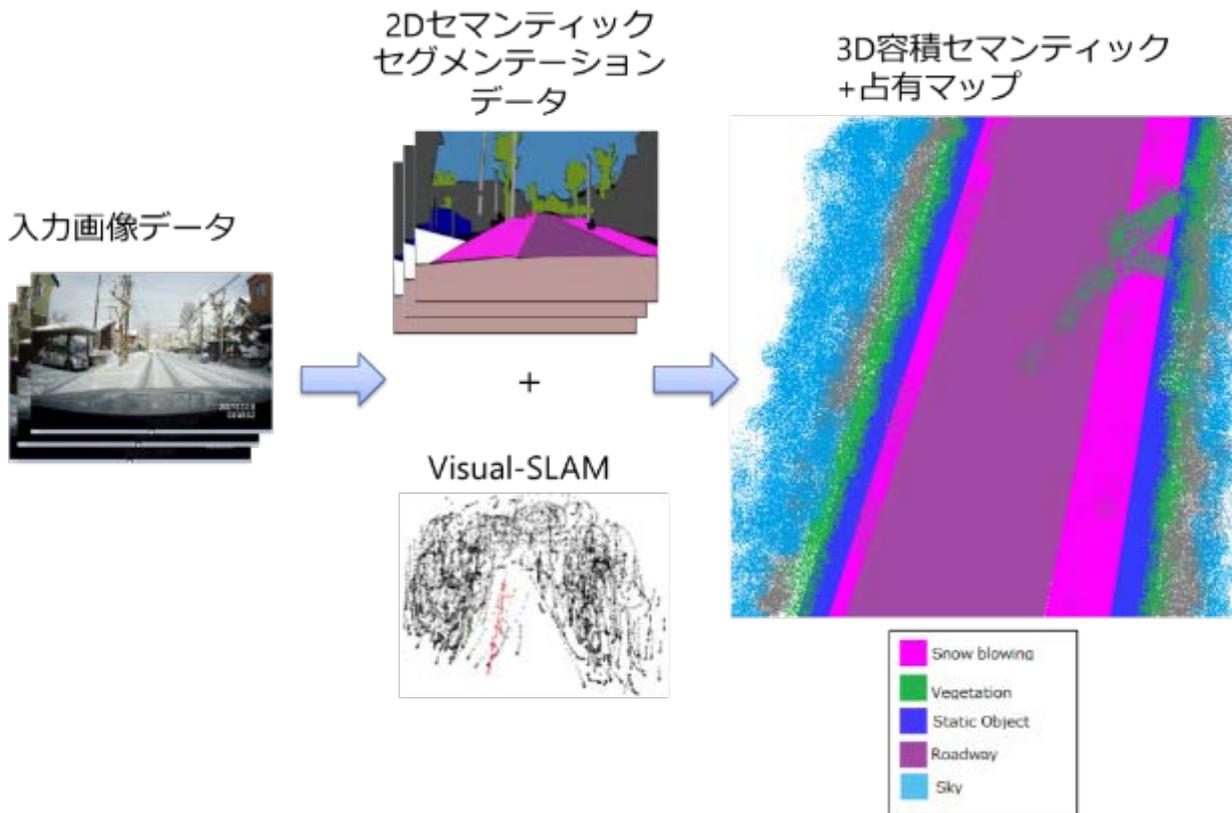
※VoxNet : 3D ポイントクラウドデータを占有グリッドでフィルターし、更に Voxel 化して、占有グリッドを分割することで高精度のオブジェクト認識を実現している。

VoxNet アーキテクチャ[論文 2]

論文 3 では、まず Visual-SLAM の 3D マップをカメラ画像に投影して、2D デプスマップを得る。次に、Visual-SLAM を 2D 画像データに逆写像する方法の記載があった。



3D マップのカメラ画像への投影[論文 3]

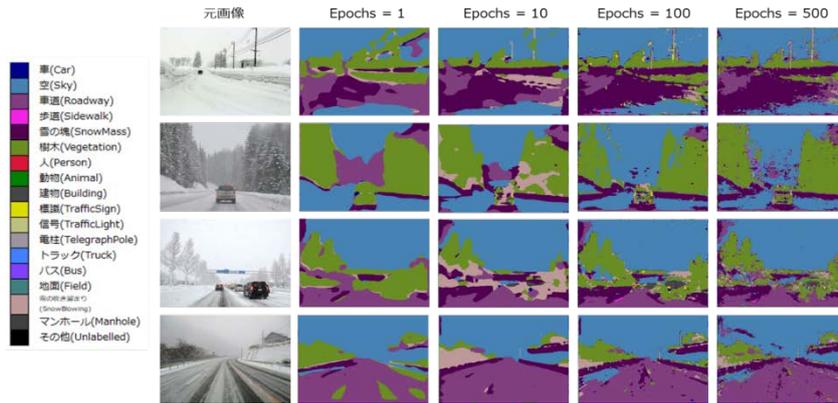


Visual-SLAM を 2D 画像データに逆写像[論文 3]

【開発方針】

セマンティック・セグメンテーション(以下、SS とする)の結果得られた以下の識別情報をもとに、SLAM に活用する。

- ・車、空、車道、人、といった 19 の分類されたクラス



H29 年度研究成果報告書から抜粋

SS は動画データから 2 次元の画像データを抽出し、解析を行う。

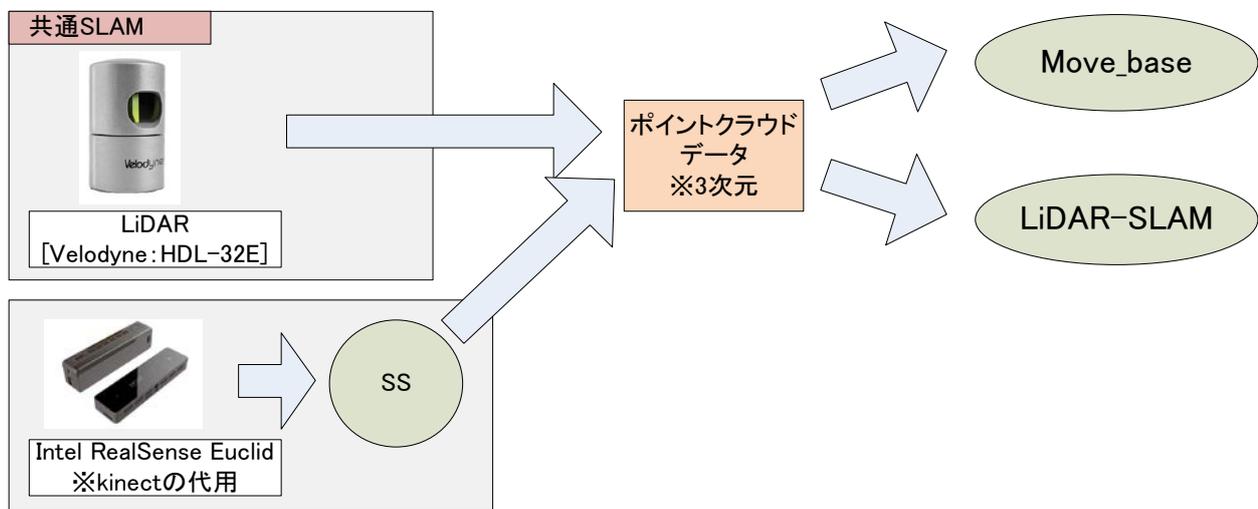
2 次元の座標情報とその分類されたクラス情報で管理される。

このため、この解析後のデータを知能化プラットフォーム側で、どのように扱うかを決定する必要がある。

案① 共通 SLAM

各センサ結果、および SS の結果得られた解析後データを

同一の 3 次元ポイントクラウド情報に集約する(揃える)



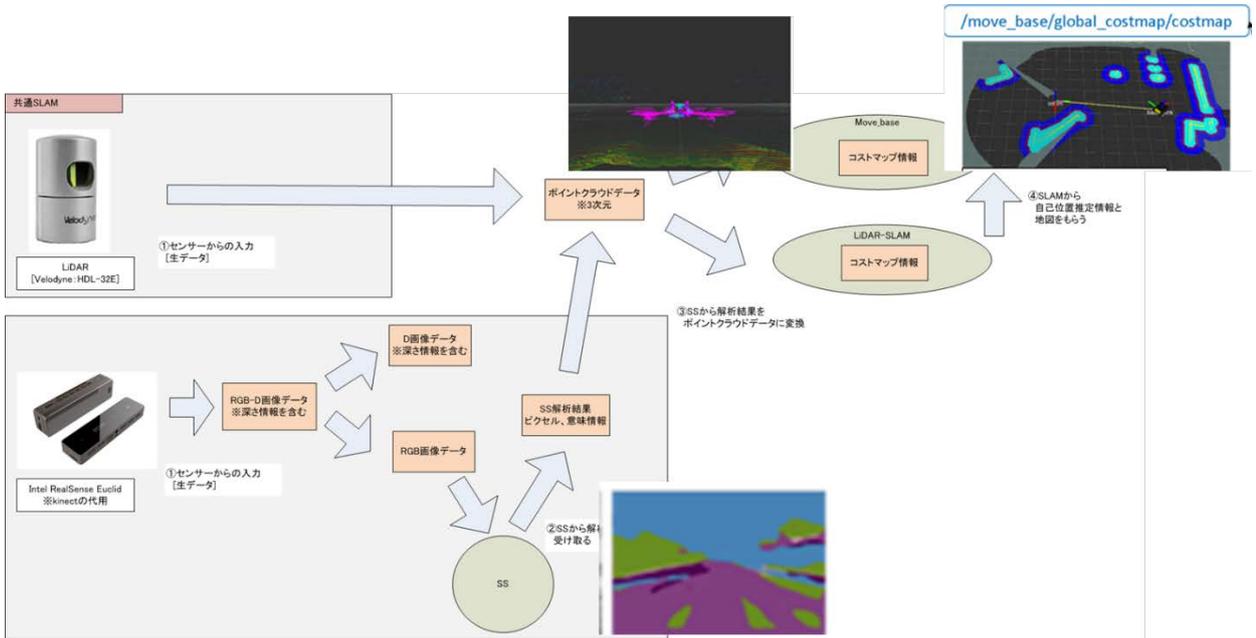
共通 SLAM(概要)

RGB-D 画像を取得可能なカメラである、RealSense を用いてデータ取りを行い、この RGB 画像データから SS 解析後、D 画像の奥行き情報を用いて 3 次元の情報に変換する。

- ① RealSense から RGB 画像データと D 画像データ(奥行き情報)を取得する。
- ② ①で取得した RGB 画像データから、SS を行う。
- ③ SS 結果に D 画像データの Point Cloud をマッピングする(※PCL を使用)。
⇒奥行き情報付きの SS 結果データが得られる

④ ③で得られた奥行情報付きの SS 結果データを、SLAM/move_base に入力する。

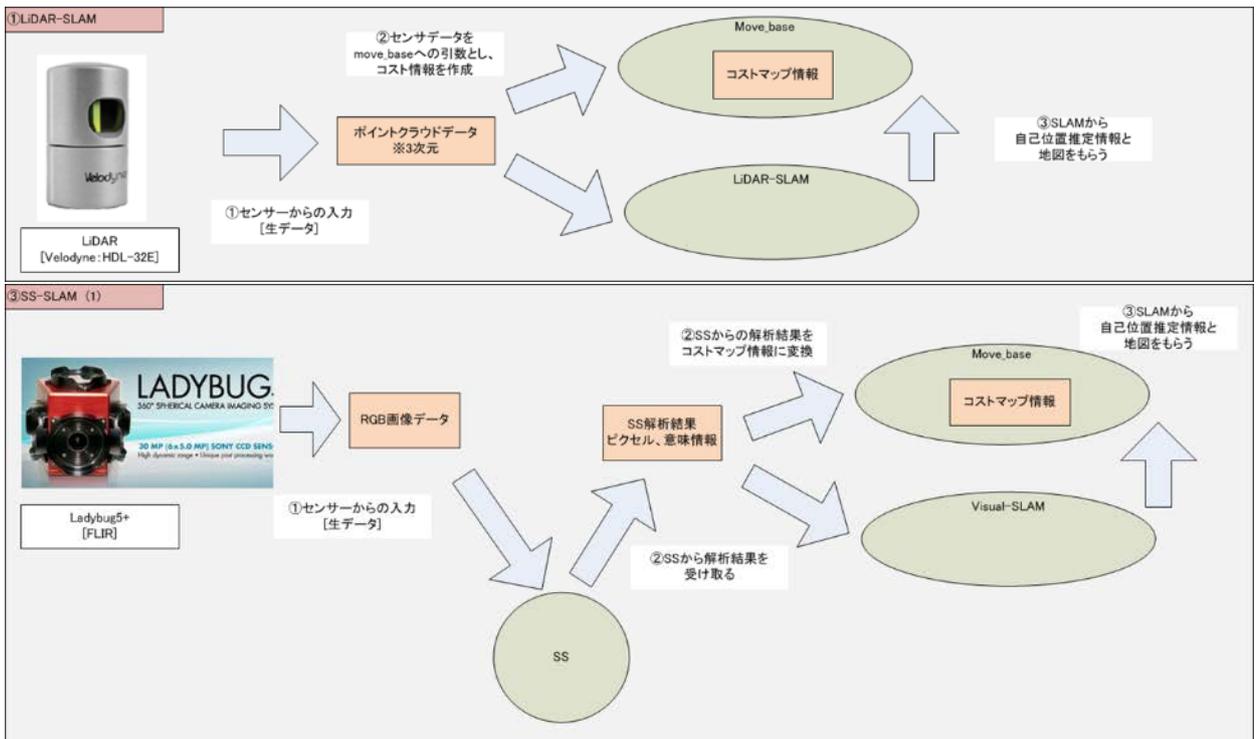
※Point Cloud Library: 2次元/3次元ポイントクラウド処理を行う、オープンソースフレームワーク



共通 SLAM(詳細)

案② 個別 SLAM

LIDAR から得られたセンシング結果を LIDAR-SLAM へ、
SS から得られたセンシング結果を Visual-SLAM へ渡す



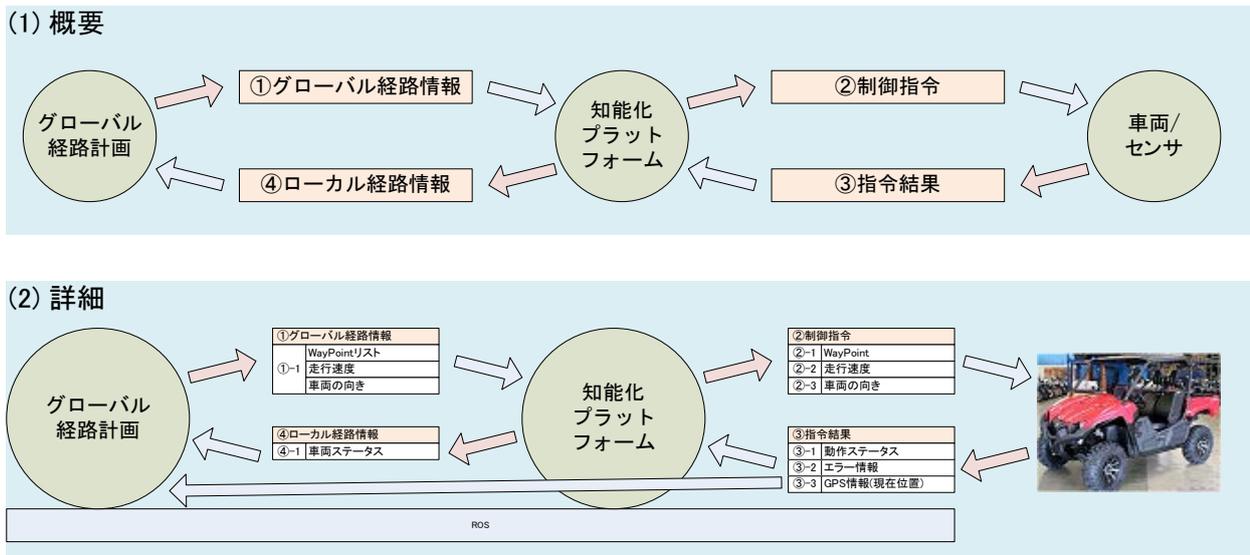
個別 SLAM

【2. 知能化プラットフォームとグローバル経路計画との連携】

車両が走行するためには、知能化プラットフォームとグローバル経路計画との間で WayPoint などのデータ授受を行う必要がある。このため通信インターフェース仕様を決定した。

知能化プラットフォームとグローバル経路計画との通信は rosbridge を介して行う。

知能化プラットフォームとグローバル経路計画、車両／センサとのインターフェースは以下となる。



知能化プラットフォームとグローバル経路計画、車両／センサとのインターフェース

グローバル経路計画から知能化プラットフォームへのインターフェースを以下に整理した。

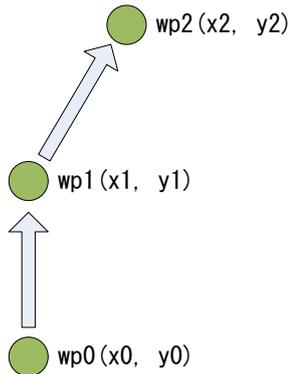
グローバル経路計画から知能化プラットフォームへ、グローバル経路情報(GRP_API_Waypoint)をスペース区切りの文字列データにて送信する。

グローバル経路計画→知能化プラットフォームのインターフェース

No.	データ					備考
	内容	説明	データ名称	データ型	取り得る範囲	
1	WayPoint	グローバルな緯度、経度情報	GRP_API_Waypoint	std_msgs/String 「経度(単位:度)、緯度(単位:度)、速度(単位:km/h)、角度(単位:度)」をスペース区切りの文字列で送信する。	経度 : -180.000000000 ~180.000000000 ※小数以下9桁まで有効 緯度 : -90.000000000 ~90.000000000 ※小数以下10桁まで有効	<ul style="list-style-type: none"> ・以下の場合には知能化プラットフォーム側でデータを破棄し、エラーとする。 ①範囲外の値が指定された場合 ②latitude または longitude の前回と今回の値の差が1度(距離として110km程度)より大きい場合
	走行速度	Waypoint-Waypoint 間の走行速度		※1	速度: 0~60 ※整数、基本的に固定	

					・範囲内の値が指定されており、実際の走行速度が 30km/h を超えた場合は、ワーニングのみとする。
車両の向き	WayPoint 通過時の車両の向き		角度 : -180.0~180.0 ※整数、北を 0 度とし、東方向を+、西方向を-とする。		範囲外の値が指定された場合、知能化プラットフォーム側でデータを破棄し、エラーとする。

※1 GRP_API_Waypoint のデータ構造を以下に記載する。上から順にWayPoint、速度、向きのデータがスペース区切りにて書き込まれる。



行番号	経度	緯度	速度	角度
1	141.735989667	42.6779203333	10	0
2	141.736014333	42.6779136667	10	10
3	141.736257167	42.6778523333	10	15
:	:	:	:	:

GRP_API_Waypoint のデータ構造

知能化プラットフォームからグローバル経路計画へのインターフェースを整理した。

車両ステータスは知能化プラットフォームからグローバル経路計画へ、文字列データにて送信する。

一方、GPS 情報は車両からグローバル経路計画へ直接送信する。

知能化プラットフォーム→グローバル経路計画のインターフェース

No.	データ					備考
	内容	説明	データ名称	データ型	取り得る範囲	
1	車両ステータス	エラーまたはワーニング発生内容の情報	GRP_API_Status	std_msgs/String	<ul style="list-style-type: none"> 正常 1 : 正常 ワーニング -1 : 実際の走行速度が 30km/h を超えた エラー -1001 : WayPoint の指定値異常 -1002 : WayPoint のインデックス範囲外 -1003 : 走行速度の指定値異常 -1004 : 車両の向きの指定値異常 -1005 : スタート時の GPS 情報異常 -1006 : 行き止まりに到達した システムエラー なし 	左記以外の値が指定された場合、エラーが発生したものとみなす。
2	スタート時の GPS 情報	GPS から得られるデータを直接送信	/gps/data	sensor_msgs/NavSatFix	—	—

2-1-3-(5)3年度の目標

雪道対応センサ・フュージョン技術の開発について令和元年度は、以下の目標に向けて活動を実施。

・令和元年度の目標

1. 調停システムの検討・設計
2. 調停システム実装
3. 調停システム評価と最適化

2-1-3-(6)具体的な内容

【1. 調停システムの検討・設計】

a). ID マップ検討

各センサからセンサ・フュージョンへ渡されるデータ形式を合わせるため ID マップの検討した。

①ID マップ内容

ラベル名				意味
header	seq			番号
	stamp	secs		時間
		nsecs		時間
	frame_id			map
info	map_load_time	secs		map 取得時間
		nsecs		map 取得時間
	resolution			分解能コストマップ的には 25cm
	width			map の横軸サイズ
	height			map の縦軸サイズ
	origin	position	x	map 中心 x 座標
			y	map 中心 y 座標
			z	map 中心 z 座標
		orientation	x:	map の x 軸クォータニオン
	y		map の y 軸クォータニオン	
z	map の z 軸クォータニオン			
w	map のクォータニオン			
data				ID

②ID

センサ	検出項目	ID
Lidar	不明	-1
	障害物有	0
	障害物無	100
カメラ (セマンティック・セグメンテーション)	不明／未取得	-1
	点群無し	0
	車道	1
	歩道	2
	建物	3
	吹雪	4
	雪塊	5
	電信柱	6
	信号機	7
	交通標識	8
	植生	9
	フィールド	10
	空	11
	人	12
	動物	13
	車	14
	トラック	15
	バス	16
	マンホール	17
バイク	18	
静的オブジェクト	19	
サーモ	不明／未取得	-1
	轍	21
	轍じゃない	22

d). ID マップの重ね合わせ検討

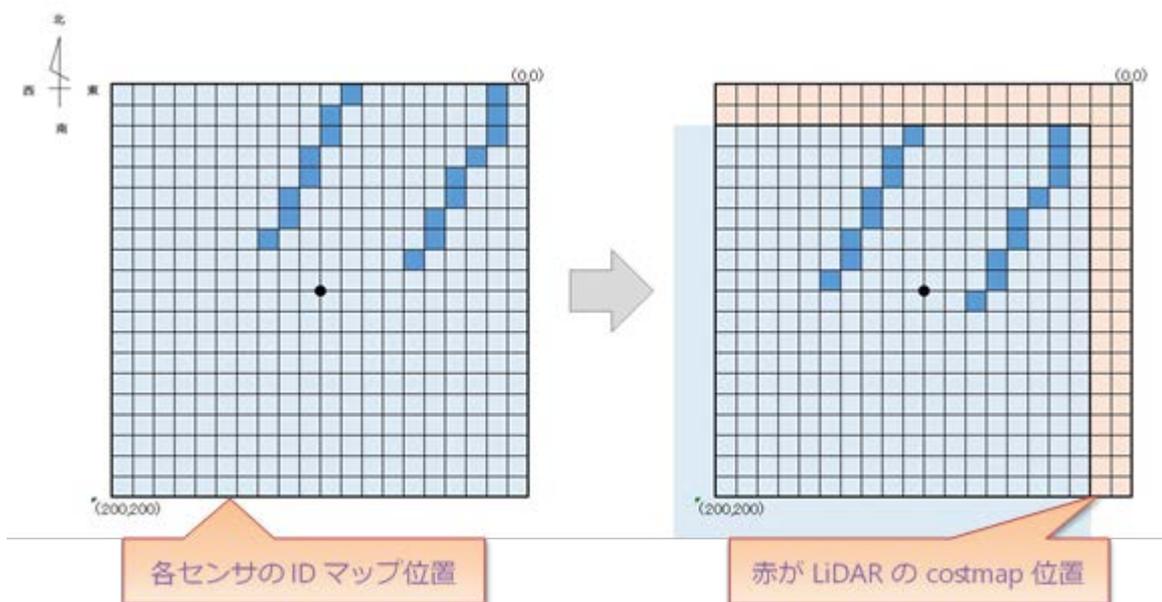
各センサからの ID マップを重ね合わせコストマップを生成する検討を行った

各センサからの情報は取得タイミングが同じでは無いため自己位置がずれた状態での重ね合わせとなってしまう。

そのため自己位置を合わせて重ね合わせる事で優良な情報を生成する事が出来るように検討した。

① 位置ずれ調整

各センサは Lidar の点群を利用し距離を計測しているが取得タイミングによりセンサ毎に若干のずれが生じてしまうため重ね合わせる際に自己位置からずれを合わせるように検討を行った。



c). 検出項目の重み付けによるコストマップ生成検討

調停機能として検出項目を絶対コスト、通常コストに分類し通常コストの場合コストを平均し障害物の確率を変化させコストマップを生成する検討を行った。

本検討でセンサ取得情報に誤認があった場合でも安全に走行する又は停止する事が可能となるように検討した。

① 絶対コスト

絶対に誤認してはいけないものは優先しコストを入れるように作成した。

検出項目：人、車、トラック、バス、バイク、動物、建物、電柱、信号機、交通標識、雪塊(コスト値:253)

② 通常コスト

絶対に誤認してはいけないものは優先しコストを入れるように作成した。

検出項目	コスト値
点群無し	0
車道	64
歩道	190
吹雪	100
植生	70
空	5
静的オブジェクト	0
マンホール	80
不明／未取得	※

※平均値数から削除

実証実験では問題が出る事が無いが走らせたいシチュエーションでも平均では出せない事がわかり
通常コストをシチュエーション別に生成するように再検討した。

また、詳細なコスト値を入力しても move_base 側では 3 段でしか認識していないためコストの値も 3 段の値
になるように再検討した。(検討d)

d). シチュエーション別コスト検討

各センサの検出項目合わせてシチュエーション別に分類しシチュエーションに 3 段階のコストを付けるコスト
マップを検討した。

本検討でより安全な走行及び処理を行う事が可能となるように検討した。

① シチュエーション別コスト

下記の検討結果にまとめた。

検出項目			コスト値
Lidar	カメラ	サーモ	
障害物有	未取得	未取得	×
障害物有	未取得	轍	×
障害物有	未取得	轍じゃない	×
障害物有	歩道	未取得	×
障害物有	歩道	轍	×
障害物有	歩道	轍じゃない	×
障害物有	点群無し	未取得	×
障害物有	点群無し	轍	×
障害物有	点群無し	轍じゃない	×
障害物有	静的オブジェクト	未取得	×
障害物有	静的オブジェクト	轍	×
障害物有	静的オブジェクト	轍じゃない	×

障害物有	吹溜り	未取得	×
障害物有	吹溜り	轍	×
障害物有	吹溜り	轍じゃない	×
障害物有	植生 vegetation	未取得	×
障害物有	植生 vegetation	轍	×
障害物有	植生 vegetation	轍じゃない	×
障害物有	車道	未取得	×
障害物有	車道	轍	×
障害物有	車道	轍じゃない	×
障害物有	建物、雪塊、電信柱、信号機、交通標識、人、動物、車、トラック、バス、バイク	未取得	×
障害物有	建物、雪塊、電信柱、信号機、交通標識、人、動物、車、トラック、バス、バイク	轍	×
障害物有	建物、雪塊、電信柱、信号機、交通標識、人、動物、車、トラック、バス、バイク	轍じゃない	×
障害物有	空	未取得	×
障害物有	空	轍	×
障害物有	空	轍じゃない	×
障害物有	マンホール	未取得	×
障害物有	マンホール	轍	×
障害物有	マンホール	轍じゃない	×
障害物有	フィールド	未取得	×
障害物有	フィールド	轍	×
障害物有	フィールド	轍じゃない	×
障害物無	未取得	未取得	○
障害物無	未取得	轍	○
障害物無	未取得	轍じゃない	△
障害物無	歩道	未取得	×
障害物無	歩道	轍	×
障害物無	歩道	轍じゃない	×
障害物無	点群無し	未取得	○
障害物無	点群無し	轍	○
障害物無	点群無し	轍じゃない	○
障害物無	静的オブジェクト	未取得	×
障害物無	静的オブジェクト	轍	×
障害物無	静的オブジェクト	轍じゃない	×
障害物無	吹溜り	未取得	△
障害物無	吹溜り	轍	○
障害物無	吹溜り	轍じゃない	△

障害物無	植生 vegetation	未取得	△
障害物無	植生 vegetation	轍	○
障害物無	植生 vegetation	轍じゃない	×
障害物無	車道	未取得	○
障害物無	車道	轍	○
障害物無	車道	轍じゃない	○
障害物無	建物、雪塊、電信柱、信号機、交通標識、人、動物、車、トラック、バス、バイク	未取得	×
障害物無	建物、雪塊、電信柱、信号機、交通標識、人、動物、車、トラック、バス、バイク	轍	×
障害物無	建物、雪塊、電信柱、信号機、交通標識、人、動物、車、トラック、バス、バイク	轍じゃない	×
障害物無	空	未取得	○
障害物無	空	轍	○
障害物無	空	轍じゃない	○
障害物無	マンホール	未取得	△
障害物無	マンホール	轍	△
障害物無	マンホール	轍じゃない	△
障害物無	フィールド	未取得	×
障害物無	フィールド	轍	×
障害物無	フィールド	轍じゃない	×
不明(検出範囲外)	未取得	未取得	△
不明(検出範囲外)	未取得	轍	○
不明(検出範囲外)	未取得	轍じゃない	×
不明(検出範囲外)	歩道	未取得	×
不明(検出範囲外)	歩道	轍	×
不明(検出範囲外)	歩道	轍じゃない	×
不明(検出範囲外)	点群無し	未取得	△
不明(検出範囲外)	点群無し	轍	△
不明(検出範囲外)	点群無し	轍じゃない	△
不明(検出範囲外)	静的オブジェクト	未取得	×
不明(検出範囲外)	静的オブジェクト	轍	×
不明(検出範囲外)	静的オブジェクト	轍じゃない	×
不明(検出範囲外)	吹溜り	未取得	△
不明(検出範囲外)	吹溜り	轍	○
不明(検出範囲外)	吹溜り	轍じゃない	△
不明(検出範囲外)	植生 vegetation	未取得	△
不明(検出範囲外)	植生 vegetation	轍	○
不明(検出範囲外)	植生 vegetation	轍じゃない	×

不明(検出範囲外)	車道	未取得	○
不明(検出範囲外)	車道	轍	○
不明(検出範囲外)	車道	轍じゃない	○
不明(検出範囲外)	建物、雪塊、電信柱、信号機、交通標識、人、動物、車、トラック、バス、バイク	未取得	×
不明(検出範囲外)	建物、雪塊、電信柱、信号機、交通標識、人、動物、車、トラック、バス、バイク	轍	×
不明(検出範囲外)	建物、雪塊、電信柱、信号機、交通標識、人、動物、車、トラック、バス、バイク	轍じゃない	×
不明(検出範囲外)	空	未取得	△
不明(検出範囲外)	空	轍	△
不明(検出範囲外)	空	轍じゃない	△
不明(検出範囲外)	マンホール	未取得	△
不明(検出範囲外)	マンホール	轍	△
不明(検出範囲外)	マンホール	轍じゃない	△
不明(検出範囲外)	フィールド	未取得	×
不明(検出範囲外)	フィールド	轍	×
不明(検出範囲外)	フィールド	轍じゃない	×

※○:コスト=0、△:コスト=50、×:コスト=100

e). 処理遅延対策検討

自タイム 100ms 内で取得出来たセンサ信号を使いコストマップを生成しセンサ故障の際等でセンサ信号が取得できなかった場合でも動作する方法を取っていた。

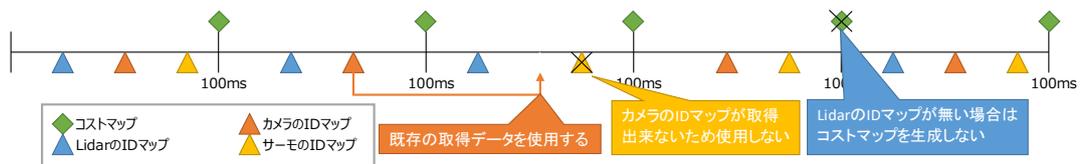
センサ側で遅延が多発する現象がありその際、突然走行ルートが無くなり急ブレーキが掛かる現象となったため何回かデータが来なくても既存のデータで処理するように検討を行った。

また、サーモの情報は道路で有る事の可能性を高めるための位置付けであるためセマンティックが取得出来ていない場合にサーモのデータを有効にすると走れるルートが無くなる事がわかったため、センサ毎に優先度をつける検討を行った。

センサ遅延は 300ms まで過去のデータを使用する様に修正。

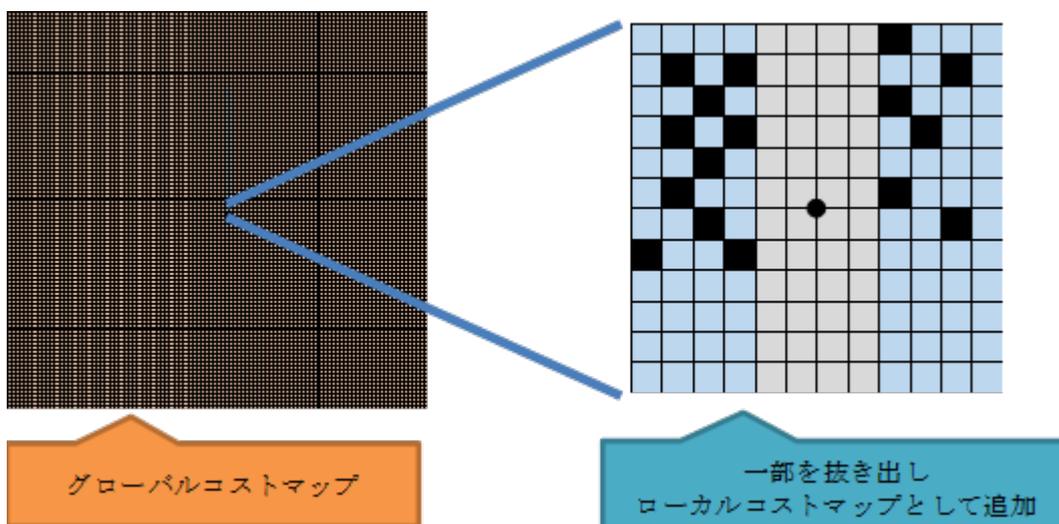
センサ優先度: Lidar > カメラ > サーモ

本検討により雪等でセンサが瞬時機能しなくても既存の情報で走行が可能となるように検討した。



f). グローバルコストマップのローカルコストマップ取り込み検討

センサでは判別出来ない道以外の場所、事前に走行実績のある場所等を走行する際に過去走行時のグローバルコストマップを使用する事により通れない場所をコストマップに追加出来る機能を追加検討した。



グローバルコストマップとローカルコストマップの生成タイミングが異なる、及び判別のサイズが異なるため合成する際にずれが生じてしまい車両ギリギリの幅を走行する場合には再度検討が必要となるが、夏の道路幅等のデータであれば有効であるため追加

【2. 調停システム実装】

- a). 自タイマ 100ms のタイミングで ID マップからコストマップを生成
100ms の割り込みタイミングを生成しメイン処理を行う。
- b). イベント割り込みで ID マップを取得
各センサが作成した ID マップをイベント割り込みで取得する。
- c). メイン処理
 - ① Lidar から取得した ID マップの自己位置とサイズを原点とし各センサのずれを補正し合成出来る ID マップを作成する。各センサが作成した ID マップをイベント割り込みで取得する。
 - ② ID マップをシチュエーションで合成しコストマップを作成する
- d). 合成処理
 - ① Lidar の情報が重要であるため Lidar 情報が取得出来ない場合は処理を行わない。
 - ② カメラの情報が無い場合はサーモの処理を行わない。
 - ③ グローバルコストマップはタイミングが異なるためデータが有れば合成する。

【3. 調停システム評価と最適化】

- a). 評価
 - ① 実証実験時に取得した ROSBAG データを使い検証
(ID マップ、ID マップ重ね合わせ、コストマップ生成、遅延対策、グローバルコストマップの取り込み)
 - ② シチュエーション別コスト
通常シチュエーションは実証実験で確認
あり得ないシチュエーションは疑似データを使用して確認
- b). 最適化
 - ① 処理が間に合わない場合の対策
ID マップの変更が有った部分のみの情報でも ID マップを作成出来るように作成しており負荷が掛かった際の対策として使用可能
 - ② 走行方向等の計算
走行方向等の情報は ID マップを作成する際に計算する事で負荷を分散

2-1-4. 実証実験

2-1-4-(1) 初年度の目標

・29年度の目標

1. 実証ルール作成と備品準備
2. 実証実験計画作成
3. 実証実験報告書作成

サブテーマ1-4では、実証実験を行うにあたって実証実験を安全に実施するためのルール策定と備品準備、実証実験の目的や実施内容等の実験計画の立案、そして、各回の実証実験について実施報告書を作成した。

初年度実施した実証実験は以下の通りである。

	日時	場所	実証実験の内容
第一回 実証実験	2017/12/26(火) 11:00~15:30	株式会社苫東特設実証コース	寒冷地域における凍結路面での各種センサーデータを取得した
	2017/1/23(火) 9:00~18:00	北海道大学J153室(保管庫)および周辺の駐車場	・準天頂衛星によるGPS機能の精度検証を実施した
	2017/1/24(水) 11:00~16:00	株式会社苫東特設実証コース	・雪道初期環境において、試作したSnow-SLAM用センサーフュージョンによるセンサーデータを取得した ・小型GPS機器による苫東特設コースの地図データ用作成用GPS情報を取得した
第二回 実証実験	2017/1/25(木) 9:00~11:00	三笠市西桂沢地区	・当プロジェクトで想定している環境モデルコース(除雪整備されている一般公道)にて、試作したSnow-SLAM用センサーフュージョンにより、検証用としてセンサーデータを取得した
	2018/2/26(月) 13:00~18:00	北海道大学J153室(保管庫)および周辺の駐車場	雪道自動運転プラットフォームの動作検証を実施した
第三回 実証実験	2018/2/27(火) 9:00~18:00	北海道大学創成科学研究棟東側	雪道自動運転プラットフォームでの自動運転機能を実証した
	2018/2/28(水) 11:00~16:00	株式会社苫東特設実証コース	・除雪雪道環境において、試作したSnow-SLAM用センサーフュージョンによるセンサーデータを取得した ・自動運転機能の動作を検証した

2-1-4-(2) 具体的な内容

【1. 実証ルール作成と備品準備】

実証実験を安全に行うための実証実験と備品準備を行った。具体的には以下作業を行った。

- ① リスクアセスメント(シミュレーション)
- ② リスクアセスメント(シミュレーション)結果を元にした備品の準備
- ③ リスクアセスメント(シミュレーション)結果を元にしたルールの策定

①に関して、打ち合わせを行い、実験の流れを1つ1つシミュレーションしながら、その時に起こり得るリスクおよび対策を洗い出した。

内容の一部を下記に示す。

タイミング	リスク	対策
実験前	実験コースが実験可能な状態でない	事前(前日)に実験コースの下見を実施し、実験可能か管理責任者が判断する
実験前	実験がスケジュール通り進まない	実験計画書を展開し、参加者全員に共有する
実験前	降雪量が多すぎて実験できない。	事前に天気が悪い場合は中止とする。また、必要に応じて、除雪を依頼する。
実験前	実験車両、実験備品、参加者の移動できない	事前に参加者、装備品を検討し、車両の手配を行う
キャリアカーの積み下ろし時	中型者の運転技術が未熟なため、操作ミスや事故を起こす。	事前に訓練する。安全確認を行う担当を必ず配置する。
キャリアカーの積み下ろし時	中型者の運転技術が未熟なため、操作ミスや事故を起こす。	キャリアカーへの積み下ろしは、専任者が実施する。
キャリアカーの積み下ろし時	積む時の脱輪やキャリアカーの枠の衝突。	安全確認を行う担当を配置する。バギーに乗るのは1人とする。また、ヘルメット・手袋を装着する。
キャリアカーの積み下ろし時	キャリアカー上での車両固定が甘い	固定具が必要 固定方法は、事前に車両運搬講習にてYAMAHA から講習を受ける
移動時	センサの故障	現地で取り付けるセンサと事前に取り付けるセンサを分ける。
移動時	センサの故障	車両の梱包を行う。
移動時	キャリアカーの到着が遅れる。	必要な場合はMLに連絡する。
実験時	データが取れていない場合に勝手な判断でバギーを動かすなどの操作を行い、事故を起こす。	データを確認する者と運転者を分ける。
実験時	実験時に車両が暴走して事故を起こす。	運転者のほか非常停止ボタンを押す担当を設ける。
実験時	安全ルールが守られない	実験開始時にブリーフィングを実施する
実験時	路面に障害物(ふきだまり、雪塊含む)があり、事故につながる	コースの安全確認を実施する
実験時	運転手の疲労が事項につながる	運転手、運転助手(4人)は交代できるようにする
実験時	寒さ対策が必要	防寒着、ストーブを準備する
実験時	バッテリーが無くなる	予備のバッテリーを準備する
実験時	部外者がエリアにする	一般道に通ずる部分にバリケードを

		準備する
実験時	実験車両のトラブルやコース上のトラブルが発生する	管理責任者と実験車両搭乗者がトランシーバを使って常に交信できる状態にする

②に関して、上記リスクアセスメント(シミュレーション)の結果を元にして備品の選定を行った。

物品	単価	数量	合計	備考
ヘルメット(M)	9650	0	0	http://amzn.asia/7M1d0P2 57-58cm
ヘルメット(L)	9599	0	0	http://amzn.asia/hiu41XS 59-60cm
防寒着(M)	7560	0	0	http://amzn.asia/5unkzwp 身長:160-170cm 胸囲:84-92cm ウエスト:72-80cm
防寒着(L)	7560	0	0	http://amzn.asia/gTFtvBZ 身長:170-180cm 胸囲:90-98cm ウエスト:78-88cm
防寒着(LL)	7560	0	0	http://amzn.asia/bi9NKlr 身長:175-185cm 胸囲:96-104cm ウエスト:86-96cm
防寒着(3L)	7084	0	0	http://amzn.asia/c0F1Ddz 身長:175-185cm 胸囲:102-110cm ウエスト:96-104cm
グローブ	1280	6	7680	http://store.workman.co.jp/item/item.html?i=2190
長靴(M)	5929	0	0	プロノスターマリンセーフティ(江丸先生の写真にあるもの) 25.0-25.5cm
長靴(L)	5929	0	0	プロノスターマリンセーフティ(江丸先生の写真にあるもの) 26.0-26.5cm
長靴(LL)	5929	0	0	プロノスターマリンセーフティ(江丸先生の写真にあるもの) 27.0-27.5cm
長靴(3L)	5929	0	0	プロノスターマリンセーフティ(江丸先生の写真にあるもの) 28.0-28.5cm
長靴(4L)	5929	0	0	プロノスターマリンセーフティ(江丸先生の写真にあるもの) 29.0-29.5cm
長靴(5L)	5929	0	0	プロノスターマリンセーフティ(江丸先生の写真にあるもの) 30.0-30.5cm
無線機	24800	3	74400	http://amzn.asia/a0yOisq Viking、先導車両、詰め所の想定

③に関して、上記項目での検討内容をもとに、実証実験ルールとして制定した。なお、実証実験ルールを記載したドキュメントは GroupSession 経由で展開済みである。

内容について以下に示す。

実験を行う上で、各役割を定義した。定義を以下に示す。

管理責任者 [↔]	実証実験の総責任者 [↔] 実証実験を実施する場合は、管理責任者1名以上の参加が必要 [↔]
操縦者(運転者) [↔]	実験車両を操縦(運転)する者、もしくは、助手席に同乗する者 [↔] 普通免許保持者 [↔]
車両運搬者 [↔]	運搬車両運転と実験車両積み下ろし操作の責任者兼実施者 [↔] 中型免許保持者 [↔]
安全管理者 [↔]	本実証試験実験ルールの運営を推進する者 [↔]
映像記録者 [↔]	実証実験の映像を記録する者 [↔]
実験参加者 [↔]	上記以外の実験参加者 [↔]

実験の際は、必ず実証実験の総責任者となる管理責任者を配置し、参加を必須とした。

また、操縦者(運転者)および、車両運搬者は事前教育を必須とした。内容は、車両講習会(運搬講習会)および、安全教育(Web コース)からなる。

また、運行前点検とチェックシートを作成した。内容を以下に示す。

Viking運行前点検シート

車両名				
点検者				
実施責任者				
試験目的				
試験場所				

点検項目	点検内容	点検日			
		/	/	/	/
原動機	エンジンオイル量				
	オイル注入口キャップ及びドレインボルトの緩み				
	オイルタンク類のバイピング状態				
	エンジン各部のオイルの漏れ				
	燃料タンク、配管部よりの漏れ有無				
動力伝達	ギヤケースオイル注入口キャップ及、ドレインボルトの緩み				
	上記ケースオイル漏れ				
足回り関係	アッパーアーム、ローアーム取付けボルトの緩み				
	Fr, Rr.サスペンション取付け部の緩み				
	ピボットシャフト部ボルトナットの緩み				
	ハブ締め付けボルトナット及びホイールナットの緩み				
	タイヤの亀裂の有無・摩耗量				
	タイヤのエア圧(F 1.45 / R1.55 [kgf/cm ²])				
制動装置	前後輪のブレーキ調整状態(含むパーキング)				
	ブレーキオイルの漏れ及びブレーキオイル量				
操舵系	ハンドル、コラム、ステアリング部の緩み				
	タイロッド及びナックル部の緩み				
点火器系	ヘッドライト、テールライト点灯チェック				
その他	油温、油圧取り出し部チェック				
	シートベルトの作動確認				
操作系	アクセルペダル操作支障ないこと				
	ブレーキペダル操作支障ないこと				
	ハンドル操作支障ないこと				
	シフトレバー操作支障ないこと				
	パーキングレバー操作支障ないこと				
	オート・マニュアル切り替えSW操作支障ないこと				
	非常停止SW操作支障ないこと				
制御装置	車室内制御機器等、着雪、着氷無きこと				

注1、上記各部点検時は、動作に支障をの恐れがある着雪、着氷、凍結無きことも確認すること
 注2、上記項目は、車両起動前に点検を行い、実施責任者の確認を得たのち、車両起動し、下記の各項目の確認を行うこと

車両起動後の確認(特に着雪、着氷、凍結の影響ないか確認すること)					
E/G始動後	アクセルペダル操作でE/G操作スムーズに行えること				
	シフトレバー操作で、シフト切り替えが行えること				
	非常停止SWによるE/G停止				
予備走行	適切な加減速行えること				
	ハンドル操作に支障がないこと				
	インジケータ、ブザー等、不具合無きこと				

6. → チェックシート

【実験計画概要】

実施日(期間)	
管理責任者	
操縦者(運転者)	
安全管理者	
車両運搬者	
映像記録者	
参加者	
目的	<input type="checkbox"/> データ収集 <input type="checkbox"/> 自立走行検証
実験車両	<input type="checkbox"/> Vking(運搬車含む) <input type="checkbox"/> ロー一般車両
場所	<input type="checkbox"/> 北大 <input type="checkbox"/> 苫東 <input type="checkbox"/> 公道

【チェックリスト】

事前準備

No	チェック項目	
①→	実験エリアの届け出は実施済みか？	<input type="checkbox"/>
②→	張り紙等による実験の周知徹底がされているか？	<input type="checkbox"/>
③→	単管バリケードがコースの公道側に設置されているか？(苫東のみ)。	<input type="checkbox"/>
④→	路面に実験の支障となる障害物、転落箇所はないか確認したか？	<input type="checkbox"/>
⑤→	直前に、実験エリアに人及び車両の進入はないか確認したか？	<input type="checkbox"/>
⑥→	管理責任者は、ブリーフィングを開催したか？ □□□□□□(確認事項:実験内容、役割分担、注意事項、チェックシート実施状況等)	<input type="checkbox"/>

運行前点検(Vikingのみ)

No	チェック項目	
①→	操縦者(運転者)は、必要装具(ヘルメット、ゴーグルまたはヘルメットのシールド、長袖長ズボン、手袋、足首が保護される靴)を装着しているか？(Vikingのみ)	<input type="checkbox"/>
②→	車両の運行前点検は実施済みか？(別紙□Vking 運行前点検シートを実施する)	<input type="checkbox"/>
③→	トランスミッターは正常に動作するか？	<input type="checkbox"/>

【2. 実証実験計画作成】

各実験の内容について、以下内容を含む計画としてまとめ、計画書を作成した。計画書は GroupSession 経由で展開済みである。

- 実験日時及び場所
- 参加者
- 物品リスト
- 実験日のスケジュール
- 実施場所の地図

以下に、計画書の内容の一部を抜粋して説明する。

スケジュールの一部を抜粋して示す。担当者、日時内容を記載して各実験日毎に記載した。

日時	担当	内容
8:30	車両運搬者/ 操縦者	ニッポンレンタカー北 21 条営業所に集合する。 カーキャリアおよびレンタカーのレンタル手続きを行い、北大保管庫に向かう。
9:00	全員	北大保管庫に集合する。集合後、カーキャリアおよびレンタカーに車両、荷物を積載する。 北大から苫東に向けて移動する。
9:00~11:00	全員	※移動中に幌が外れそうになった場合は輪厚 PA または三沢 PA に対応する。(場所は 2.4、2.5 参照) ※ガソリン・灯油の購入が必要であれば途中で購入。 苫東に集合する。
11:00~12:00	全員	ブリーフィングを行い、実験内容等を確認する。 その後、実験準備を行う。 昼食休憩
12:00~13:00	全員	※状況に応じて時間の変更や中止の場合あり。 ※昼食の購入が必要な者はこのタイミングで買い出しに行く。 ※一律の休憩が難しい場合は手の空いている操縦者又は車両運搬者(とその同伴者)が買い出しに向かう。 実験を行う。
13:00~15:30	全員	・ルート探索機能と連携した自動走行の確認 ・センサ類のデータ取得 撤収作業を行う。(車両・荷物の積載、整備等)
15:30~16:00	全員	ジェットヒーター返却作業も行う。 (業者への返送)
16:00~18:00	全員	苫東へ終了報告を行い、北大保管庫へ向かう。 北大保管庫に到着する。
18:00~	全員	バギーの格納および機材を収納する。 カーキャリアおよびレンタカーが空き次第、ニッポンレンタカー北 21 条営業所に向かい、返却手続きを行う。

以下に物品リストの例として、第三回実証実験における物品リストを示す。

各物品と準備する担当者、数量をまとめている。

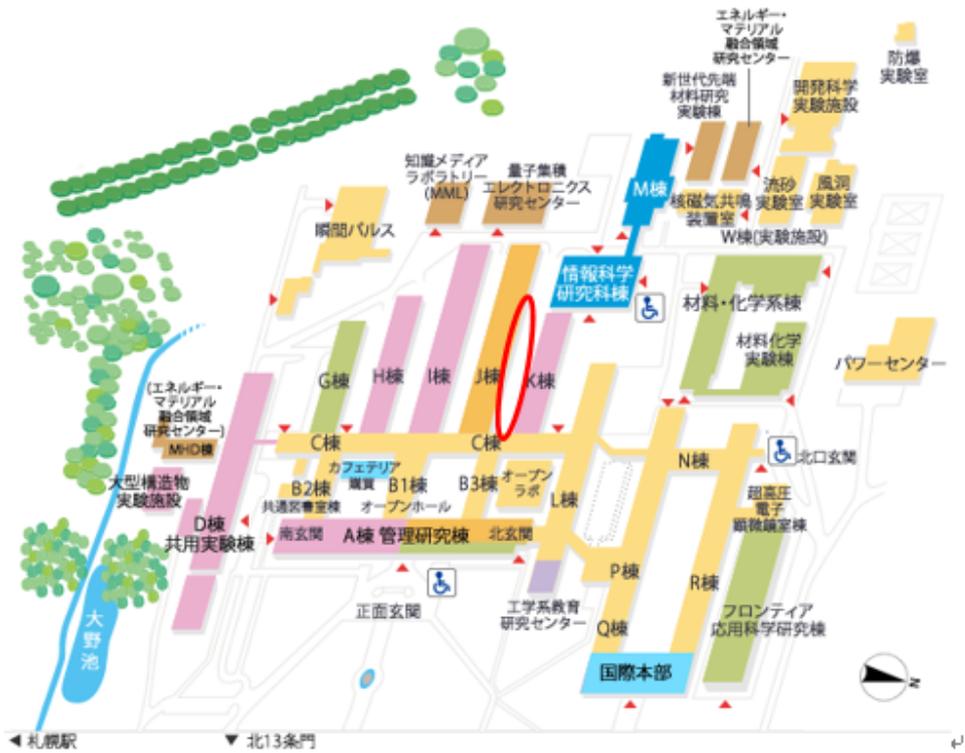
名称	数量	担当	備考
カーキャリア	1	車両運搬者	VikingVI の運搬に使用する。 人員・物資運搬用。
レンタカー	2	操縦者	・ハイエース(1 台) ・フィット(1 台)
測定装置一式	1	江丸	測定に必要な装置、装置台、計測 PC など。
12V バッテリ	1	江丸	測定装置駆動用。
100V インバータ	1	江丸	測定装置駆動用。
三脚	1	江丸	GPS の補正局用。
防寒装備	1	江丸	必要分のヘルメット、防寒着、グローブ、長靴 (苫東に保管中)
無線機	3	江丸	連絡用。
ビデオカメラ	1	江丸	映像記録者用。 MicroSD、バッテリー装着すること。
ほうき	1	江丸	雪払い用。
牽引ロープ	1	江丸	車両救出用
砂利	1	江丸	車両救出用(滑り止めに使用)
スコップ	1	江丸	車両救出用
ヒーター	2	江丸	暖房用。 インバータ駆動用。
ガソリン缶	1	江丸	※不足する場合は運転者が買出しに向かう。 ※基本的に苫東への移動途中の購入を想定し、 実験中に不足が発覚した場合は随時向かう。 ヒーター駆動用。
灯油缶	1	江丸	※不足する場合は運転者が買出しに向かう。 ※基本的に苫東への移動途中の購入を想定し、 実験中に不足が発覚した場合は随時向かう。 実証実験時の昼食。
昼食	1	各自	可能であれば、各々で用意(持ち込み)しても良い。 ※費用は自己負担(本研究事業の予算対象外)。

地図の例を以下に示す。実験場所に関して、住所および地図を記載した。また、苫東での実験時の際の緊急停車場所として、PA の情報も掲載した。

2.2 北海道大学工学部 J153 室(保管庫)⇩

住所⇩	〒060-8628 札幌市北区北 13 条西 8 丁目⇩
-----	------------------------------

⇩



【3. 実証実験報告書作成】

各実験について、以下内容を記載した報告書を作成した。計画書を作成した。報告書は GroupSession 経由で展開済みである。

- 実験日時及び場所
- 参加者
- 実施内容

報告書から一部内容を抜粋して以下に示す。実施内容に関しては、各日事を実施した内容を記載した。

1.4.3 → 1/25 実施内容

三笠市内での、一般車両に Lider、ステレオカメラ、ミリ波レーダ、赤外線カメラの各種センサを取り付け、雪道における各種センサのデータ取得を行った。

実験の様子について以下に示す。



2-1-4-(3)2年度の目標

・30年度の目標

1. 実証ルールの実施及び周知
2. 運行前点検とチェックシートの改定

サブテーマ1-4では、実証実験を行うにあたって実証実験を安全に実施するためのルール策定と実施、実証実験の目的や実施内容等の実験計画の立案、そして、各回の実証実験について実施報告書を作成した。

2年度実施した実証実験は以下の通りである。

	日時	場所	実証実験内容
第一回	2018/6/18(月) 14:00~18:00	北海道大学J153室(保管庫)及び 北海道大学創成科学研究棟東側	・車両整備。 ・みちびき端末の動作確認 ・IMARの測位状況確認 ・接続確認
	2018/6/19(火) 9:00~18:00	北海道大学J153室(保管庫)及び 北海道大学創成科学研究棟東側	・みちびき端末による走行確認
第二回	2018/7/9(月) 9:00~12:00	株式会社苫東内私有地(直線路)	・一般車両にて測位データ収集、Waypoint地点の座標記録
	2018/7/9(月) 13:00~18:00	北海道大学創成科学研究棟東側	・GPS,GNSSによる自律運転機能の検証
	2018/7/10(火) 9:00~18:00	株式会社苫東内私有地 (直線路及び旗竿地)	・GPS,GNSSによる自律運転機能の検証
第三回	2018/8/22(水) 13:00~17:00	北海道大学創成科学研究棟東側	・GPSによる自律運転機能の検証 □LIDARによる障害物回避機能の検証
	2018/8/23(木) 10:00~15:00	北海道大学創成科学研究棟東側	・GPSによる自律運転機能の検証 □LIDARによる障害物回避機能の検証
	2018/8/23(木) 9:00~18:00	北海道大学J153室(保管庫)	□サブテーマ1-サブテーマ3間I/F確認
第四回	2018/9/3(月) 11:00~17:00	株式会社苫東内私有地	・GPSによる自律運転機能の検証 ・LIDARによる障害物回避機能の検証 ・SLAMによる自立走行と障害物回避の検証
	2018/9/4(火) 10:00~17:00	株式会社苫東内私有地	・GPSによる自律運転機能の検証 ・LIDARによる障害物回避機能の検証 ・SLAMによる自立走行と障害物回避の検証
	2018/9/5(木) 10:00~18:00	株式会社苫東内私有地	・GPSによる自律運転機能の検証 ・LIDARによる障害物回避機能の検証 ・SLAMによる自立走行と障害物回避の検証
第五回	2018/9/26(水) 10:00~18:00	株式会社苫東内私有地(旗竿地)	・GPSによる自律運転機能の検証 (蛇行走行の原因調査&対策確認)
	2018/9/27(木) 9:00~18:00	株式会社苫東内私有地(旗竿地)	・GPSによる自律運転機能の検証 (蛇行走行の原因調査&対策確認)
	2018/9/23(金) 10:00~18:00	株式会社苫東内私有地(旗竿地)	・GPS+LIDARによる自立走行&障害物回避
第六回	2018/10/22(月) 9:30~16:00	北海道大学(構内道路)	・Cartographer (3D)によるLiDAR-SLAMデータ収集
	2018/10/23(木) 9:30~16:00	北海道大学創成科学研究棟東側	・Cartographer (3D)によるLiDAR-SLAMデータ収集
第七回	2018/11/20(火) 8:30~18:00	北日本自動車大学校 (車両テストコース)	・走行実験とみちびきの動作確認
	2018/11/21(木) 9:00~13:00	北日本自動車大学校 (車両テストコース)	・走行実験とみちびきの動作確認
	2018/11/22(木) 9:00~18:00	北海道大学J153室(保管庫)	・GPSによる自律運転機能の検証 ☑セマンティック・セグメンテーション結合検証 ・SLAMによる自立走行と障害物回避の検証
	2018/11/23(木) 10:00~14:00	北海道大学創成科学研究棟東側	・GPSによる自律運転機能の検証 ☑セマンティック・セグメンテーション結合検証 ・SLAMによる自立走行と障害物回避の検証

	日時	場所	実証実験内容
第八回	2018/12/17(火) 9:00~17:00	北海道大学J153室(保管庫)及び 北海道大学創成科学研究棟東側	・セマンティック・セグメンテーションの確認 ・シミュレータのためJ棟駐車場にてデータ取り
	2018/12/18(木) 14:00~15:00	北海道大学J153室(保管庫)	・セマンティック・セグメンテーションの確認 ・シミュレータのためJ棟駐車場にてデータ取り
	2018/12/19(木) 9:00~13:00	株式会社苫東内私有地 (周回路)	・周回路での自動運転 ・セマンティック・セグメンテーションのデータ確認
	2018/12/20(木) 9:00~13:00	株式会社苫東内私有地 (周回路)	・周回路での自動運転 ・セマンティック・セグメンテーションのデータ確認
第九回	2019/1/17(火) 9:00~17:00	北海道大学J153室(保管庫)及び 北海道大学創成科学研究棟東側	・セマンティック・セグメンテーションの動作検証 ・SLAM動作検証
	2019/1/18(木) 14:00~15:00	株式会社苫東内私有地 (周回路)	・周回路での自動運転
	2019/1/19(木) 9:00~13:00	北日本自動車大学校 (車両テストコース)	・車両テストコースでの自動運転
	2019/1/20(木) 9:00~13:00	北日本自動車大学校 (車両テストコース)	・車両テストコースでの自動運転
第十回	2019/2/12(火) 9:00~18:00	株式会社苫東内私有地 (周回路)	・SLAM動作検証
	2019/2/13(水) 9:00~18:00	株式会社苫東内私有地 (周回路)	・SLAM動作検証
第十一回	2019/2/19(火) 9:00~16:00	株式会社苫東内私有地 (周回路)	・SLAM動作検証
	2019/2/20(水) 9:00~16:00	株式会社苫東内私有地 (周回路)	・SLAM動作検証
	2019/2/21(木) 9:00~18:00	株式会社苫東内私有地 (周回路)	・SLAM動作検証
	2019/1/23(土) 9:00~18:00	士別市一般道	・一般道のセンサー情報確認

2-1-4-(4) 具体的な内容

【1. 実証ルールの実施及び周知】

初年度作成した実証ルールを基にルールの実施及び張り紙等周知を徹底し試験が安全に行えるように作業を行った。

また、実験車両の整備を含む運行前点検を徹底する事で安全に配慮した実践が行えるように作業を行った。

役割分担

事前に役割を分担する事でスムーズな試験実施が行えるように作業計画書内に記載し事前に展開及びグリーンフィングにて周知を行った。

管理責任者 [Ⓜ]	実証実験の総責任者 [Ⓜ] 実証実験を実施する場合は、管理責任者1名以上の参加が必要 [Ⓜ]
操縦者(運転者) [Ⓜ]	実験車両を操縦(運転)する者、もしくは、助手席に同乗する者 [Ⓜ] 普通免許保持者 [Ⓜ]
車両運搬者 [Ⓜ]	運搬車両運転と実験車両積み下ろし操作の責任者兼実施者 [Ⓜ] 中型免許保持者 [Ⓜ]
安全管理者 [Ⓜ]	本実証試験ルールの実証を推進する者 [Ⓜ]
映像記録者 [Ⓜ]	実証実験の映像を記録する者 [Ⓜ]
実験参加者 [Ⓜ]	上記以外の実験参加者 [Ⓜ]

【2. 運行前点検とチェックシートの改定】

新しい Viking 用の点検シートを頂いたため新しいものへ変更し、使用していないものの削除及び気が付いた点検内容の追加を行った。

内容を以下に示す。

Viking運行前点検シート

車両名				
点検者				
実施責任者				
試験目的				
試験場所				

点検項目	点検内容	点検日			
		/	/	/	/
原動機	エンジンオイル量				
	オイル注入口キャップ及びドレインボルトの緩み				
	オイルタンク類のバイピング状態				
	エンジン各部のオイルの漏れ				
	燃料タンク、配管部よりの漏れ有無 クーラントの漏れ及びクーラント量				
動力伝達	ギヤケースオイル注入口キャップ及、ドレインボルトの緩み				
	上記ケースオイル漏れ				
足回り関係	アッパーアーム、ロワーアーム取付けボルトの緩み				
	Fr,Rr.サスペンション取付け部の緩み				
	ピボットシャフト部ボルトナットの緩み				
	ハブ締め付けボルトナット及びホイールナットの緩み				
	タイヤの亀裂の有無・摩耗量				
	タイヤのエア圧(F 1.45 / R1.55 [kgf/cm ²])				
制動装置	前後輪のブレーキ調整状態(含むパーキング)				
	ブレーキオイルの漏れ及びブレーキオイル量				
操舵系	ハンドル、コラム、ステアリング部の緩み				
	タイロッド及びナックル部の緩み				
点火器系	ヘッドライト、テールライト点灯チェック				
その他	シートベルトの作動確認				
操作系	アクセルペダル操作支障ないこと				
	ブレーキペダル操作支障ないこと				
	ハンドル操作支障ないこと				
	シフトレバー操作支障ないこと				
	パーキングレバー操作支障ないこと				
	オート・マニュアル切り替えSW操作支障ないこと				
	非常停止SW操作支障ないこと				
制御装置	車室内制御機器等、着雪、着氷無いこと				

注1、上記各部点検時は、動作に支障をの恐れがある着雪、着氷、凍結無いことも確認すること
 注2、上記項目は、車両起動前に点検を行い、実施責任者の確認を得たのち、車両起動し、下記の各項目の確認を行うこと

車両起動後の確認(特に着雪、着氷、凍結の影響ないか確認すること)					
E/G始動後	アクセルペダル操作でE/G操作スムーズに行えること				
	シフトレバー操作で、シフト切り替えが行えること				
	非常停止SWIによるE/G停止				
予備走行	適切な加減速行えること				
	ハンドル操作に支障がないこと				
	インジケータ、ブザー等、不具合無いこと				

6.→ チェックシート

【実験計画概要】

実施日(期間)	
管理責任者	
操縦者(運転者)	
安全管理者	
車両運搬者	
映像記録者	
参加者	
目的	<input type="checkbox"/> データ収集 <input type="checkbox"/> 自立走行検証
実験車両	<input type="checkbox"/> Vking(運搬車含む) <input type="checkbox"/> 一般車両
場所	<input type="checkbox"/> 北大 <input type="checkbox"/> 苫東 <input type="checkbox"/> 公道

【チェックリスト】

事前準備

No	チェック項目	
①→	実験エリアの届け出は実施済みか？	<input type="checkbox"/>
②→	張り紙等による実験の周知徹底がされているか？	<input type="checkbox"/>
③→	単管バリケードがコースの公道側に設置されているか？(苫東のみ)。	<input type="checkbox"/>
④→	路面に実験の支障となる障害物、転落箇所はないか確認したか？	<input type="checkbox"/>
⑤→	直前に、実験エリアに人及び車両の進入はないか確認したか？	<input type="checkbox"/>
⑥→	管理責任者は、ブリーフィングを開催したか？ □□□□□□(確認事項:実験内容、役割分担、注意事項、チェックシート実施状況等)	<input type="checkbox"/>

運行前点検(Vikingのみ)

No	チェック項目	
①→	操縦者(運転者)は、必要装具(ヘルメット、ゴーグルまたはヘルメットのシールド、長袖長ズボン、手袋、足首が保護される靴)を装着しているか？(Vikingのみ)	<input type="checkbox"/>
②→	車両の運行前点検は実施済みか？(別紙Vking 運行前点検シートを実施する)	<input type="checkbox"/>
③→	トランスミッターは正常に動作するか？	<input type="checkbox"/>

2-1-4-(5)3年度の目標

・令和元年度の目標

1. 計画と報告書の作成
2. 運行前点検実施と安全管理の実施

サブテーマ1-4では、実証実験を行うにあたって実証実験を安全に実施するため今年度までに策定したルールの実施、実証実験の目的や実施内容等の実験計画の立案、そして、各回の実証実験について実施報告書を作成した。

3年度実施した実証実験は以下の通りである。

	日時	場所	実証実験内容
第一回	2019/06/19(水) 13:00~17:00	北海道大学・J棟駐車場、創成棟	IMU制御検証
	2019/06/20(木) 9:00~17:00	北海道大学・J棟駐車場、創成棟	IMU制御検証
第二回	2019/07/16(火) 13:00~17:00	株式会社苫東（周回路）	同一パラメータで冬と夏の違いを確認
	2019/07/17(水) 9:00~17:00	株式会社苫東（周回路）	夏データを考慮したパラメータ調整の確認 苫東周回路の夏時の地図作成
	2019/07/18(木) 9:00~17:00	株式会社苫東（周回路）	地図データを使用しての走行テスト
	2019/07/19(金) 9:00~16:00	株式会社苫東（周回路）	地図データを使用しての走行テスト
第三回	2019/08/19(月) 9:00~18:00	北海道大学・株式会社苫東	SLAM自己位置推定の打ち合わせ IMUによる自己位置推定の検証 SLAMデータ収集
	2019/08/20(火) 9:00~18:00	北海道大学・J棟駐車場、創成棟	コストマップの補正（コストを壁にする）の検証 カメラ-LiDARのキャリブレーション
	2019/08/21(水) 9:00~16:00	北海道大学・J棟駐車場、創成棟	IMUによる自己位置推定の検証 グローバル経路計画のパラメータ調整の検証
第四回	2019/09/09(月) 13:00~18:00	北海道大学・会議室、J棟駐車場	新しい調停、カメラ位置の変更、Lidar有効範囲変更を確認
	2019/06/20(木) 9:00~17:00	北海道大学・J棟駐車場、創成棟	新しい調停、カメラ位置の変更、Lidar有効範囲変更を確認
第五回	2019/09/17(火) 8:00~18:00	北日本自動車大学校 (車両テストコース)	GPSが取れる状態での高速走行試験
	2019/07/18(水) 9:00~18:00	北日本自動車大学校 (車両テストコース)	GPSが取れる状態での高速走行試験
	2019/07/19(木) 9:00~18:00	北日本自動車大学校 (車両テストコース)	GPSが取れる状態での高速走行試験
第六回	2019/10/23(水) 9:30~16:00	北海道大学・J棟	Lidarのフィルタ追加の実車確認 自己位置同定の実車確認 コストマップ生成の遅延修正実車確認
	2019/10/24(木) 9:00~16:00	北海道大学・創成棟	道以外のコストに線を引く実車確認 ローカル経路生成による蛇行の実車確認 サブテーマ③との連携で得たWaypointでの自動走行

	日時	場所	実証実験内容
第七回	2019/11/12(火) 8:30~18:00	株式会社苫東 (周回路)	サブテーマ③との連携 (地図と自己位置の補正) Lidarのフィルタ追加の実車確認 自己位置同定の実車確認 新しい調停の実車動作確認
	2019/11/13(水) 8:30~18:00	株式会社苫東 (周回路)	道以外のコストに線を引く実車確認 曲がり部分のWaypoint間隔 サーモの実車動作確認 新しい調停の認識確認
	2019/11/14(木) 8:30~18:00	株式会社苫東 (周回路)	waypointの通過認識確認 GPSの取得誤差確認
第八回	2019/12/16(月) 8:30~18:00	北日本自動車大学校 (車両テストコース)	waypointの確認 Lidarのフィルタ追加の実車確認 自己位置同定の実車確認 新しい調停の実車動作確認自動走行実験
	2019/12/17(火) 9:00~18:00	北日本自動車大学校 (車両テストコース)	全動作最終実車確認 Waypointの確認
	2019/07/18(水) 9:00~18:00	北日本自動車大学校 (車両テストコース)	サブテーマ③との連携 (地図と自己位置の補正) 全動作最終実車確認 セマンティックセグメンテーションの認識確認 障害物(車、人)・壁・道路 サーモの検出確認
	2019/07/19(木) 9:00~18:00	士別市一般道	全動作最終実車確認
第九回	2020/01/21(火) 8:30~18:00	株式会社苫東 (周回路)	サーモによる検出の検証 白トビ対策の確認 セマンティック (雪塊) 検証
	2020/01/22(水) 8:30~18:00	株式会社苫東 (周回路)	サブテーマ③との連携 周回コースを使ったシステム検証
	2020/01/23(木) 8:30~18:00	株式会社苫東 (周回路)	周回コースを使ったシステム検証
	2020/01/24(金) 8:30~18:00	北海道大学・J棟駐車場、創成棟	シミュレーション用のデータ取り
第十回	2020/02/17(月) 8:30~18:00	北日本自動車大学校 (車両テストコース)	全動作最終実車確認、グローバルコストマップの取り込み確認 Waypointの確認、コースの確認、サブテーマ③との連携
	2020/02/18(火) 8:30~18:00	北日本自動車大学校 (車両テストコース)	全動作最終実車確認 車両パラメータの調整確認、30km自動走行試験
	2020/02/19(水) 8:30~18:00	北日本自動車大学校 (車両テストコース)	全動作最終実車確認 30km自動走行試験
	2020/02/20(木) 8:30~18:00	士別市一般道	全動作最終実車確認 走行距離及び走行速度の確認

2-1-4-(6) 具体的な内容

【1. 実証ルールの実施及び周知】

2年度までに作成した実証ルールを基にルールの実施及び張り紙等周知を徹底し試験が安全に行えるように作業を行った。

また、実験車両の整備を含む運行前点検を徹底する事で安全に配慮した実践が行えるように作業を行った。

役割分担

事前に役割を分担する事でスムーズな試験実施が行えるように作業計画書内に記載し事前に展開及びグリーンフィングにて周知を行った。

管理責任者 [↔]	実証実験の総責任者 [↔] 実証実験を実施する場合は、管理責任者1名以上の参加が必要 [↔]
操縦者(運転者) [↔]	実験車両を操縦(運転)する者、もしくは、助手席に同乗する者 [↔] 普通免許保持者 [↔]
車両運搬者 [↔]	運搬車両運転と実験車両積み下ろし操作の責任者兼実施者 [↔] 中型免許保持者 [↔]
安全管理者 [↔]	本実証試験ルールの実験を推進する者 [↔]
映像記録者 [↔]	実証実験の映像を記録する者 [↔]
実験参加者 [↔]	上記以外の実験参加者 [↔]

【2. 運行前点検とチェックシートの改定】

新しいViking用の点検シートを頂いたため新しいものへ変更し、使用していないものの削除及び気が付いた点検内容の追加を行った。

内容を以下に示す。

Viking運行前点検シート

車両名				
点検者				
実施責任者				
試験目的				
試験場所				

点検項目	点検内容	点検日			
		/	/	/	/
原動機	エンジンオイル量				
	オイル注入口キャップ及びドレインボルトの緩み				
	オイルタンク類のバイピング状態				
	エンジン各部のオイルの漏れ				
	燃料タンク、配管部よりの漏れ有無 クーラントの漏れ及びクーラント量				
動力伝達	ギヤケースオイル注入口キャップ及、ドレインボルトの緩み				
	上記ケースオイル漏れ				
足回り関係	アッパーアーム、ロワーアーム取付けボルトの緩み				
	Fr,Rr.サスペンション取付け部の緩み				
	ピボットシャフト部ボルトナットの緩み				
	ハブ締め付けボルトナット及びホイールナットの緩み				
	タイヤの亀裂の有無・摩耗量				
	タイヤのエア圧(F 1.45 / R1.55 [kgf/cm ²])				
制動装置	前後輪のブレーキ調整状態(含むパーキング)				
	ブレーキオイルの漏れ及びブレーキオイル量				
操舵系	ハンドル、コラム、ステアリング部の緩み				
	タイロッド及びナックル部の緩み				
点火器系	ヘッドライト、テールライト点灯チェック				
その他	シートベルトの作動確認				
操作系	アクセルペダル操作支障ないこと				
	ブレーキペダル操作支障ないこと				
	ハンドル操作支障ないこと				
	シフトレバー操作支障ないこと				
	パーキングレバー操作支障ないこと				
	オート・マニュアル切り替えSW操作支障ないこと				
	非常停止SW操作支障ないこと				
制御装置	車室内制御機器等、着雪、着氷無いこと				

注1、上記各部点検時は、動作に支障をの恐れがある着雪、着氷、凍結無いことも確認すること
 注2、上記項目は、車両起動前に点検を行い、実施責任者の確認を得たのち、車両起動し、下記の各項目の確認を行うこと

車両起動後の確認(特に着雪、着氷、凍結の影響ないか確認すること)					
E/G始動後	アクセルペダル操作でE/G操作スムーズに行えること				
	シフトレバー操作で、シフト切り替えが行えること				
	非常停止SWによるE/G停止				
予備走行	適切な加減速行えること				
	ハンドル操作に支障がないこと				
	インジケータ、ブザー等、不具合無いこと				

6.→ チェックシート

【実験計画概要】

実施日(期間)	
管理責任者	
操縦者(運転者)	
安全管理者	
車両運搬者	
映像記録者	
参加者	
目的	<input type="checkbox"/> データ収集 <input type="checkbox"/> 自立走行検証
実験車両	<input type="checkbox"/> Vking(運搬車含む) <input type="checkbox"/> 一般車両
場所	<input type="checkbox"/> 北大 <input type="checkbox"/> 苫東 <input type="checkbox"/> 公道

【チェックリスト】

事前準備

No	チェック項目	
①→	実験エリアの届け出は実施済みか？	<input type="checkbox"/>
②→	張り紙等による実験の周知徹底がされているか？	<input type="checkbox"/>
③→	単管バリケードがコースの公道側に設置されているか？(苫東のみ)。	<input type="checkbox"/>
④→	路面に実験の支障となる障害物、転落箇所はないか確認したか？	<input type="checkbox"/>
⑤→	直前に、実験エリアに人及び車両の進入はないか確認したか？	<input type="checkbox"/>
⑥→	管理責任者は、ブリーフィングを開催したか？ ○○○○○○(確認事項:実験内容、役割分担、注意事項、チェックシート実施状況等)	<input type="checkbox"/>

運行前点検(Vikingのみ)

No	チェック項目	
①→	操縦者(運転者)は、必要装具(ヘルメット、ゴーグルまたはヘルメットのシールド、長袖長ズボン、手袋、足首が保護される靴)を装着しているか？(Vikingのみ)	<input type="checkbox"/>
②→	車両の運行前点検は実施済みか？(別紙Vking 運行前点検シートを実施する)	<input type="checkbox"/>
③→	トランシーバーは正常に動作するか？	<input type="checkbox"/>

2-1-5. 成果

2-1-5-1. 雪道対応 SLAM 技術開発

初年度成果:

初年度は、各種 SLAM 方式についての調査を行い、その結果下記 SLAM を選択した。さらに、収集した雪道データを取り込んで MAP の生成並びに自己位置の推定を実施した。

- ・ORB-SLAM (Visual SLAM として特徴量ベース)
- ・LiDAR-SLAM として現在最も一般的に用いられている Loam Velodyne

ただし、初年度の活動では意図した通りのコストマップは形成されておらず、2年度以降継続研究が必要であった。また、雪道対応自律化プラットフォームとして、Navigation Stack を中心とした、来年度以降の自律化プラットフォームベースとなるシステムを構築し、ベースとなる車両プラットフォームとの I/F 検証を実施した。

2年度成果:

2年度は、各種 LiDAR-SLAM の調査を行い、代表的な Gmapping, Hector-SLAM, Cartographer について性能評価を実施した。その結果、本プロジェクトでは Cartographer を採用し、自動運転機能の性能評価を進めた。カメラベースの Visual SLAM は、積雪環境では色特徴量が疎であるため、十分な特徴量を得ることができず、望ましい結果を得ることはできなかった。逆にサーモカメラは、轍の検出が可能であることが判明したため、知能化プラットフォームでの制御を行う際の経路指定に活用できると考えている。2年度の研究成果をベースに3年度の各技術の最適化、および、自動運転機能の性能評価を進める。

3年度成果:

3年度は、自律走行制御の改善を実施し、雪道対応セマンティック・セグメンテーション技術、雪道対応センサ・フュージョン技術とあわせて、自動運転の性能目標を達成することができた。また、センサ類のコストダウンに関してはし、セマンティック・セグメンテーションで使用しているカメラの検証を実施し、プロジェクトのスコープの範囲であれば、低解像度のカメラでもセマンティック・セグメンテーションによる走行可能領域の判別が可能であることを実証した。

2-1-5-2. 雪道対応セマンティック・セグメンテーション技術の開発

初年度成果:

雪道対応セマンティック・セグメンテーション技術の開発としての成果は以下の4つである。

- ・識別クラス分けとして19のクラス分けを決めた。
- ・ディープラーニングのフレームワークとして Keras with backend TensorFlow 使用することを決めた。
- ・データセットを作成するための Annotation Tool として LabelMe online をローカルで構築する環境を実現することができた。
- ・ディープラーニングのフレームワークとして Keras with backend TensorFlow 使用し、手法として SegNet を選択し、作成したデータセットを機械学習し、セマンティック・セグメンテーションの結果を得ることができる環境を実現することができた。

これら4つの成果を使用し、2年度は SALM との連携を検討し、結合することを進めることとした。

2年度成果

雪道対応セマンティック・セグメンテーション技術の開発としての成果は以下の4つである。

- ・ROS 上でのセマンティック・セグメンテーション実現
- ・セマンティック・セグメンテーションの認識速度改善

- 効率の良いデータセットの作成方法とデータセット量産方法の取得
- セマンティック・セグメンテーションの認識率の数値化

これら4つの成果を使用し、3年度はセマンティック・セグメンテーションの認識率を向上させることを進めることとした。

3年度成果:

3年度はサーモカメラによる轍検出を実現した。本機能により、雪道対応セマンティック・セグメンテーションで検出した道らしさを補完し、道らしさの精度向上を実現することを実証した。

2-1-5-3. 雪道対応センサ・フュージョン技術の開発

初年度成果:

以下のセンサを選択し、実証実験及び予備実験において収集を行い、各センサのデータを定性的に検証した。2年度以降はこれらのデータに対して定量的な評価を実施し、SLAM技術やセマンティック・セグメンテーション技術との融合も検討しながらセンサ・フュージョンの研究開発を進めることとした。

- 3次元 LiDAR Velodyne VLP-16
- 3次元 LiDAR Velodyne HDL-32e
- 3次元 LiDAR 北陽電機 YVT-35LX
- Depth カメラ Intel RealSense Euclid
- Depth カメラ Intel RealSense D415
- Depth カメラ Intel RealSense D435
- 2次元 LiDAR 北陽電機 UST-20LX
- 2次元 LiDAR 北陽電機 UTM-30LX
- 赤外線サーモグラフィ optris PI450
- 全方位ステレオ OccamVisionGroup OmniStereo
- 防水アクションカメラ SJCAM SJ7 STAR
- ビデオカメラ Sony HDR-CX680 TI
- 360度全方位カメラ FLIR 社 Ladybug5+
- GPS (iMAR ユニット等一式)
- Texas Instruments ミリ波レーダー、IWR1642

2年度成果:

2年度は、下記の組み合わせによる知能化プラットフォームの育成により自律走行を実現した。

- GPS 及び IMU による自律走行
- GPS + IMU 及び LidarSLAM(Cartographer) による自律走行
- GPS + IMU 及びセマンティックセグメンテーションによる自律走行
- GPS + IMU 及びサーモカメラによる自律走行

3年度これらの成果を評価し、システム全体でのセンサフュージョンにより最適な自律走行を実現することとした。

3年度成果:

3年度は、センサ・フュージョン技術として調停システムを開発し、各センサの情報を組み合わせて複合的に状況を判断することにより、自律走行を実現した。本成果により各センサの得意とする情報を有効に使用でき、複数のセン

サ及び高機能センサを搭載せずに自動走行が可能となることを実証した。

2-1-5-4. 実証実験

成果:

実証実験のリスクアセスメントから、運行前のチェックシート等を含んだ実験ルール作成、備品の調達を行い、実験スケジュールや備品リストを含む実験計画書を作成し、実装実験を実施した。実証実験後は、実施内容を含む実験報告書を作成した。上記の結果、事故を起こさずに予定通り、苫東における3回の実証実験、北海道大学北キャンパス創成棟東側における2回の実験等を実施した。

2年度以降は、自律走行の実証実験も多くなるため、初年度の実施内容を振り返った上での実験ルールの見直し検討、および各種実験の計画を行う。

成果:

運行前のチェックシート及び実験ルールに沿った確認や計画を実施した。

その結果、事故を起こす事無く予定通りの試験が実施出来た。

試験内容は天候や各プロジェクト状況により変更を行ったが試験計画書に基づいた試験を実施出来た。

3年度は、車両の劣化等も確認出来る点検項目を追加し、より安全な試験が行えるように実験ルールの見直し検討、および各種実験の計画を行う。

成果:

運行前のチェックシート及び実験ルールに沿った確認や計画を実施した。その結果、事故を起こす事無く予定通りの試験が実施出来た。3年度は、試験場所がテストコースや一般道等が加わりより一層の確認と注意を払い実施出来た。移動も増えたため実験車両以外にも運搬用の車両や試験関係者の移動用車両も注意を払い安全に実験実施し移動も含め怪我及び事故等も無く安全に実施出来た。

2-2. 【サブテーマ2】積雪環境仮想シミュレータ開発

・本サブテーマの目的

実車による評価は冬季間に限られてしまう。無積雪期間に開発を進めても実車で検証は冬季間に限られ効率的な開発を行えない。冬季積雪環境の仮想シミュレータを新たに開発し、「Snow-SLAM」方式による自動運転の性能検証を効率的に実施する。

2-2-1. 積雪環境仮想シミュレータの仕様策定

シミュレータを構築するために使用するツール、模擬対象となるセンサ、シミュレーションで走行するコースについて調査・検討を実施し、本プロジェクトで作成すべきシミュレータの環境、構成、機能を定義する。

その上で、シミュレータの作成を行い、掲げた性能目標を達成できていることの確認を実施する。

2-2-1-1. シミュレーション環境の調査

「Snow-SLAM」方式を実現するための各種機能は、自律化プラットフォーム層に実装される(図 1)。したがって、シミュレーション環境には自律化プラットフォーム層以外の車両プラットフォーム層の模擬と車両挙動、道路環境、各種環境センサの模擬が求められる。

開発するシミュレータの仮想化範囲を図 1 に示す。

冬季積雪環境を模擬するための必要となる機能および性能目標を基にシミュレーション環境の要求項目を決定。

要求項目を満たすことができるツールを選定するため、候補とするシミュレーション環境を調査し、使用するシミュレーション環境を選定する。

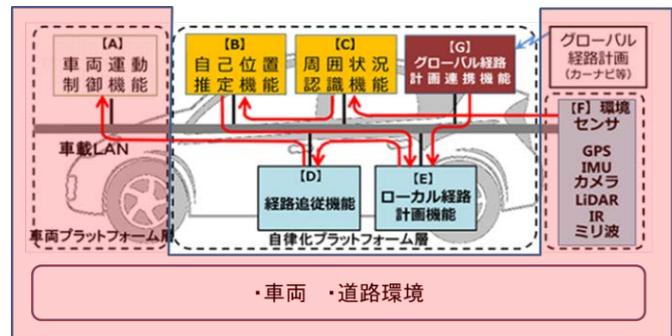


図 2-2-1-1. シミュレータの仮想化範囲

シミュレーション環境の開発ツールとしてUnreal Engineを採用することとした。

表1. の要件を基に調査した結果、Unreal EngineとUnityが候補となった。本プロジェクトの用途内であれば、2者に性能的な差はなかったが、下記2点を考慮してUnreal Engineを初年度の開発環境として選択した。

- ・Unreal Engineは、ADASの開発現場で採用実績がある
- ・Unityは商材として拡販する前の開発段階からツールの使用料が発生する

初年度は、Unreal Engineを主軸としてシミュレーション環境を開発するが、将来的にはUnityへの移植も考慮して開発を進めることとする。また、ROSと親和性が高いGazeboも継続の調査対象とする。

表 2-2-1-1. ベンチマーク結果

採用 ※1	性能・機能の評価結果 ※2	ツール名
○	○	Unreal Engine
△	○	Unity
×	×	Choreonoid

△	○	Gazebo
---	---	--------

※1

○：初年度の開発環境として選択するツール

△：初年度の開発環境としては選択しないが、今後の移植を視野に入れるツール

×：開発環境として選択しないツール

※2

2-2-1-2. シミュレータ構成の検討

2-2-1. シミュレーション環境の調査にて決定した使用ツールを用い、シミュレータと自律化プラットフォームの接続、入出力データを考慮し、シミュレーション環境のシステム構成の検討を行い、システム構成を決定する。

シミュレータのシステム構成は、UDP通信を使用する案を採用した。

採用環境は、自律化プラットフォームのロジック検証には使用可能であるが、I/F部分はUDPに置き換える必要がある。I/F部を実際と同じ仕様とする第二案はレベルアップ項目とする。

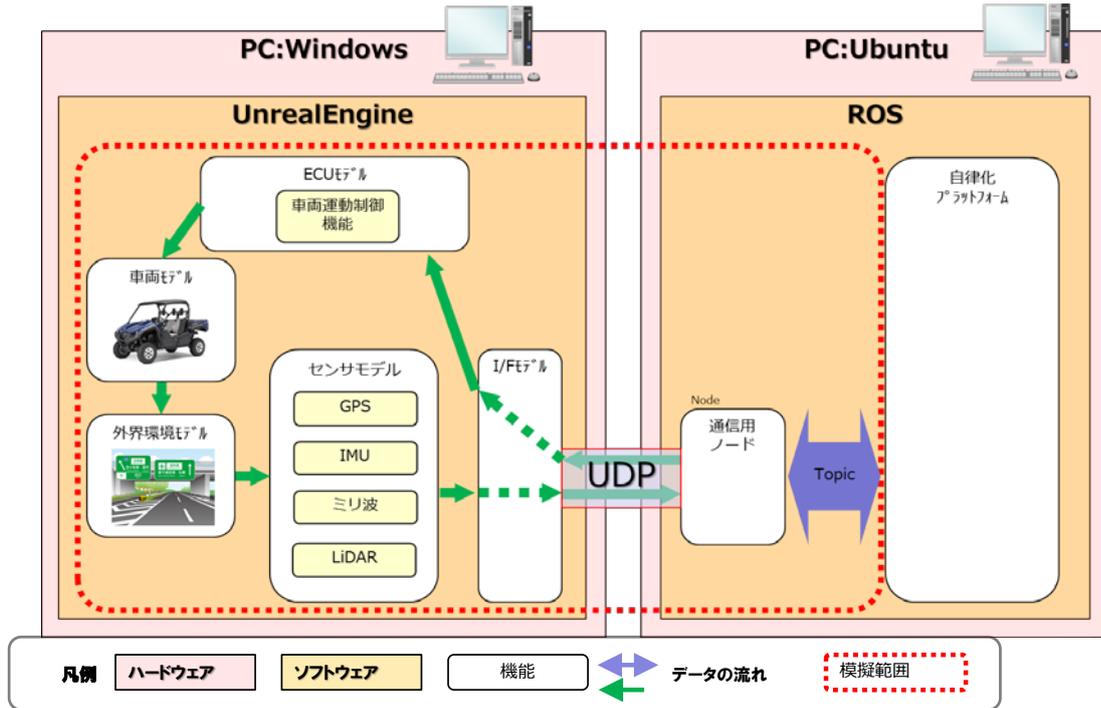


図 2-2-1-2. シミュレータ構成

表 2-2-1-2. 機能一覧

項目		説明
Unreal Engine	ECU モデル	車両の挙動を制御する「車両運動制御機能」を簡易的に模擬する
	車両モデル	車両 (Viking) の挙動を模擬するモデル
	外界環境モデル	道路、建物、樹木など外界環境を模擬するモデル
	センサモデル	外界環境から取り込んだ情報を各センサ仕様のデータに変換するモデル
	I/F モデル	Unreal Engine と自律化プラットフォームを接続するためのモデル。UDP 通信の送信・受信のデータ形式に変換して橋渡しを行う処理
ROS	ROS Bridge	自律化プラットフォームと Unreal Engine を接続する処理。ROS Topic で subscribe したメッセージを UDP で送信。

項目		説明
		また、UDP で受信した情報を ROS Topic で publish する。
	自律化プラットフォーム	検証対象となる制御アプリ。 雪道対応の SLAM 技術、セマンディックセグメンテーション技術、センサーフュージョン技術が該当する。

<特徴>

・模擬対象の機能はUnrealEngineに実装する

・センサ情報の受け渡し、及び制御信号はUDP通信に集約する

⇒ 実センサは、それぞれ異なったタイプのI/F(USB、CANなど)でデータを取り込んでいるが、個別に対応するためには個々の通信仕様に沿った処理を開発する必要がある。開発規模を低減するため、通信プロトコルをUDP 1つに限定する。

自律化プラットフォームとの通信仕様について下記表の内容とする。

表 2-2-1-2(1). シミュレータ→自律化プラットフォームへの通信仕

向き	内容	詳細	Topic 名
シミュレータ→ 自律化プラットフォーム	LiDAR センサ値	・計測タイミング ・照射方向 ・反射距離	<u>velodyne_node/</u> <u>velodyne_packets</u>
	GPS センサ値	・緯度 ・経度	<u>nmea_topic_driver/fix</u>
	IMU センサ値	・角速度(3 軸) ・角加速度(3 軸) ・車両の向き(3 軸)	-
	ミリ波レーダーセンサ値	・距離 ・水平角度 ・垂直角度 ・相対速度	<u>/mmwaveDataHdl/RScan</u>
	CAN (車両 PF の送信フレーム)	※次々ページで別途紹介	-

表 2-2-1-2(2). 自律化プラットフォーム→シミュレータへの通信仕様

向き	内容	詳細	Topic 名
自律化プラットフォーム	CAN	・目標車速	-

→シミュレータ	(自律化 PF の送信フレーム)	・目標舵角	
---------	------------------	-------	--

2-2-1-3. センサモデルの検討

図 1 の中に示される「環境センサ」で用いる各実センサについて調査を実施し、シミュレーション環境における模擬レベルおよび模擬方法について検討を実施する。

各センサモデルの模擬レベルの検討を実施し、初年度のシミュレータで実装する模擬レベルの選定結果を記す。模擬レベルはセンサ毎に複数案リストアップし、シミュレーションの開発目的である”自律化プラットフォームの検証”として、初年度は模擬レベル低を実装し、今後の他サブテーマチームからの要望を元に検討を実施し、対応を行うこととした。

1) LiDAR

LiDAR は Light Detection And Ranging(光による検知と測距)の略。

レーザで不可視スペクトルの光パルスを発射し、反射光の到達タイミングと反射強度を計測することで、対象の距離と性質を分析する。

模擬対象センサ: Velodyne VLP-16

表 2-2-1-3(1). LiDAR センサ仕様

項目	スペック
寸法	高さ 71.7mm × 直径 103.3mm
重量	約 830g
計測範囲・計測視野	水平 360° 全方位、垂直 30° (+15° ~ -15°)
計測距離	約 100m(1m~100m)
測距スピード	5~20Hz
測定ポイント数	約 300,000 ポイント/秒
測定精度	±3cm
動作電圧	9 - 32VDC

模擬レベル検討結果:

LiDAR の模擬レベル検討結果を表 3-2-1-3(2)に示す。

表 2-2-1-3(2). LiDAR 模擬レベル検討結果

レベル	模擬内容	選定結果
1	センサと対象物の距離から、センサが出力する計測結果を模擬	○
2	1に加え、対象物の素材による反射強度、拡散や他の光源による影響を模擬	-
3	2に加え、光のパルス発射、対象物による反射、受光素子が認識するタイミングや受光角度を模擬	-

模擬内容:

模擬対象の仕様である水平360度、垂直30度の範囲でオブジェクトを検出。
 オブジェクトまでの距離を算出し、点群データとして生成。
 検出した照射方向および、算出した距離値について出力。

2) ミリ波レーダー

レーダーは周囲に電波を放射し、その反射波の測定、分析を行い、対象物との距離、方向、大きさを測定する。
 ミリ波(millimeter wave)は、波長が1～10mmで、周波数が30～300GHzの電波で、天候に強く、長距離(100～200m)での検出が可能。

模擬対象センサ: Texas Instruments IWR1642

表 2-2-1-3(3). ミリ波レーダーセンサ

項目	スペック
測定範囲	±60°
測定距離	84m
測定ポイント数	-
距離分解能	37cm
角度分解能	約 15°
出力 I/F	-
データ	TBD

ミリ波レーダーの模擬レベル検討結果を表 3-2-1-3(4)に示す。

表 2-2-1-3(4). ミリ波レーダー模擬レベル検討結果

レベル	模擬内容	選定結果
1	センサと対象物の距離から、センサが出力する計測結果を模擬	○
2	1に加え、対象物の素材によるレーダーの吸収(即ち受信強度)を模擬	-
3	2に加え、ミリ波の発射、対象物による反射、受信素子が認識するタイミングや受信時のミリ波の強度を模擬	-

模擬内容:

模擬対象の仕様である水平±60度の範囲で、オブジェクトを検出。
 オブジェクトまでの距離、角度、相対速度を算出。
 検出した各オブジェクトの算出した値について出力。

3) GPS

GPS(Global Positioning System, Global Positioning Satellite)は衛星から発射した時刻信号の電波の到達時間などから、地球上の電波受信者の位置を 3 次元測位する

GPS 自体はアメリカ合衆国が開発したシステムを指し、一般には GNSS(Global Navigation Satellite System)と呼ぶが、本環境内では標準的な呼称となっている GPS を使用する。

模擬対象センサ:iMar iTraceRT-F200(GPS/IMU統合)

GPS の模擬レベル検討結果を表 3-2-1-3(5)に示す。

表 2-2-1-3(5). GPS 模擬レベル検討結果

レベル	模擬内容	選定結果
1	3D モデルの座標、速度等から情報を模擬	○
2	仮想の衛星を用意し、時刻信号の受信から模擬	-

模擬内容:

センサ地点の位置情報として、UTM 座標上の経度、緯度、高さ(標高)を算出。

算出した値について出力。

4) IMU

IMU(Inertial Measurement Unit)は慣性計測装置を指し、3 軸の角度(または角速度)と加速度を計測する。

模擬対象センサ:iMar iTraceRT-F200(GPS/IMU統合)

IMU の模擬レベル検討結果を表 3-2-1-3(6)に示す。

表 2-2-1-3(6) . IMU 模擬レベル検討結果

レベル	模擬内容	選定結果
1	センサ取り付け位置を考慮しない	○
2	1 に加え、車両の構造と取り付け位置を加味	-

模擬内容:

自車両中心点の角速度、角加速度、向きを算出。

算出した値について出力。

2-2-1-4. テストコースの検討

積雪路面状態をシミュレータ上に作成し、実装を行うセンサモデルの動作や車両モデルの走行挙動について、模擬動作の検証を実施するためのテストコースの仕様を検討する

テスト環境に作成する走行コースは実車での試験走行に用いる北海道苫小牧東部地区を再現する周回コースとする
作成する走行コースを下記図 5の苫小牧東部地区の地図上の白線として示す。



図 2-2-1-4. テストコース

2-2-2-1. シミュレータ環境の構築

前述 2-2. で調査、検討を実施した結果から定義したシミュレーション環境の実装を行う。

構築したシミュレータ環境を図 6 に示す。

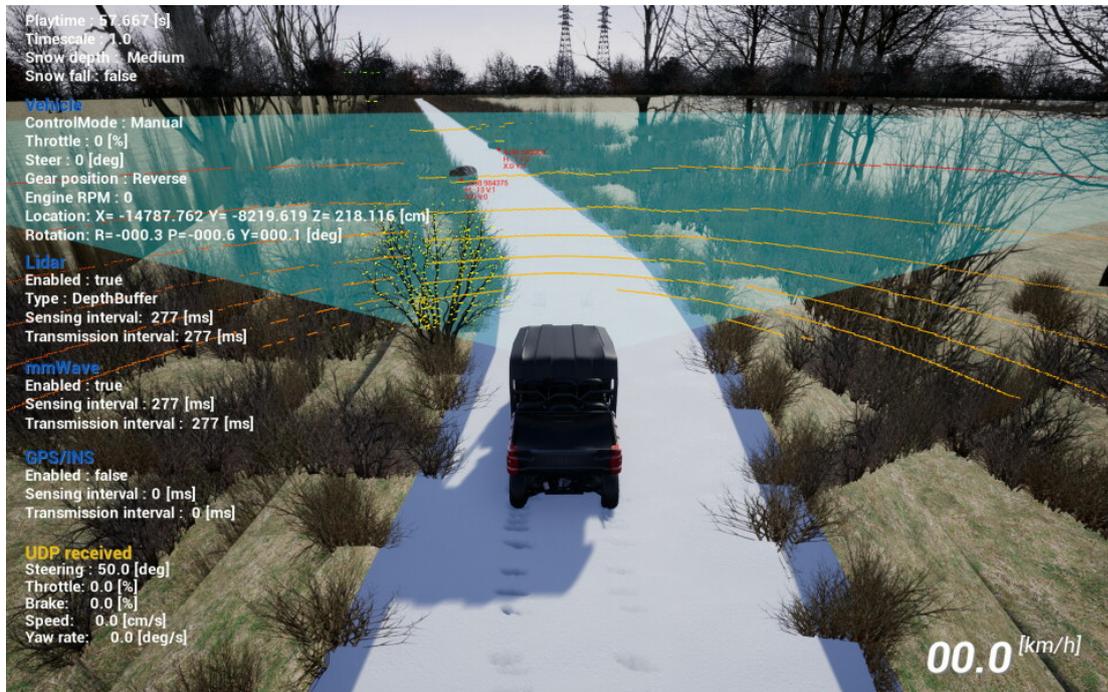


図 2-2-2-1. 構築したシミュレータ環境

2-2-2-2. テストコースの実装

2-2-1-4. で検討を行った結果を元に積雪環境(テストコース)の実装を行う。

作成したシミュレータ環境内のテストコースを下記に示す。



図 2-2-2-2. テストコース

2-2-2-3. センサモデルの実装

2-2-1-3. で検討した結果を元にセンサモデルの実装を行う。

シミュレータ側の実装は「ViViD」にて行うため、ROS 側のドライバ部分に該当する処理の実装が中心となる。
実装したセンサモデルについて下記に示す。

1) LiDAR

UE4 を用いた環境で LiDAR の点群を生成する方法として、LineTrace を用いる方法、深度バッファを用いる方法の2つが考えられる。

それぞれ有利不利があるため、両手法について実装し、設定により切り替え動作検証できるようにする。

2つの手法の差を表 2-2-2-3(1)に示す

表 2-2-2-3(1). 2つの手法の差

	LineTrace	深度バッファ
角度分解能	制限無し	最大 0.00061
距離分解能	1mm 以下 ほぼ誤差なし	200m で数 mm 遠距離になるほど精度が落ちる
処理速度	遅い	速い

同条件で比較した場合の処理時間は以下の表 2-2-2-3(2)のようになる。

表 2-2-2-3(2). フレームあたり処理時間

	標準処理時間(msec)	適用後処理時間(msec)	処理時間(msec)
LineTrace	8.3	42	33.7
深度バッファ	8.3	10.5	2.2

a) LineTrace

UE4 の BP 関数 LineTraceByChannel を使用し、1 点ごとにライントレースを行う。

角度分解能、距離の制限はない。

距離はレンダリング時の遠平面が制限となるが、制限なしとみなせる。

LineTraceByChannel は、所定のラインに沿ってコリジョン トレースを実行し、トレースがヒットした (トレース チャンネルに反応した) 最初のオブジェクトを返す。

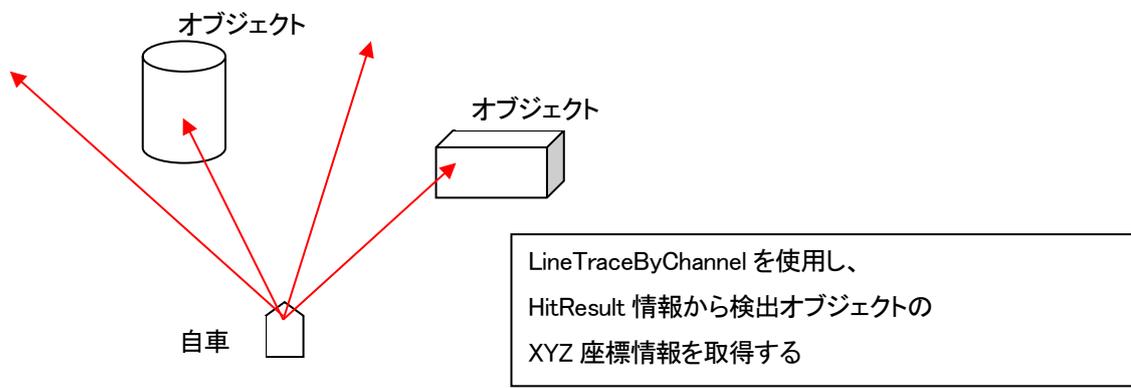


図 2-2-2-3(3). LineTrace 処理のイメージ

b) 深度バッファ

UE4 のレンダリングパイプラインで計算される深度バッファを利用し、距離から座標を求める。水平、垂直の分解能はカメラの画角とレンダリング解像度に依存する。

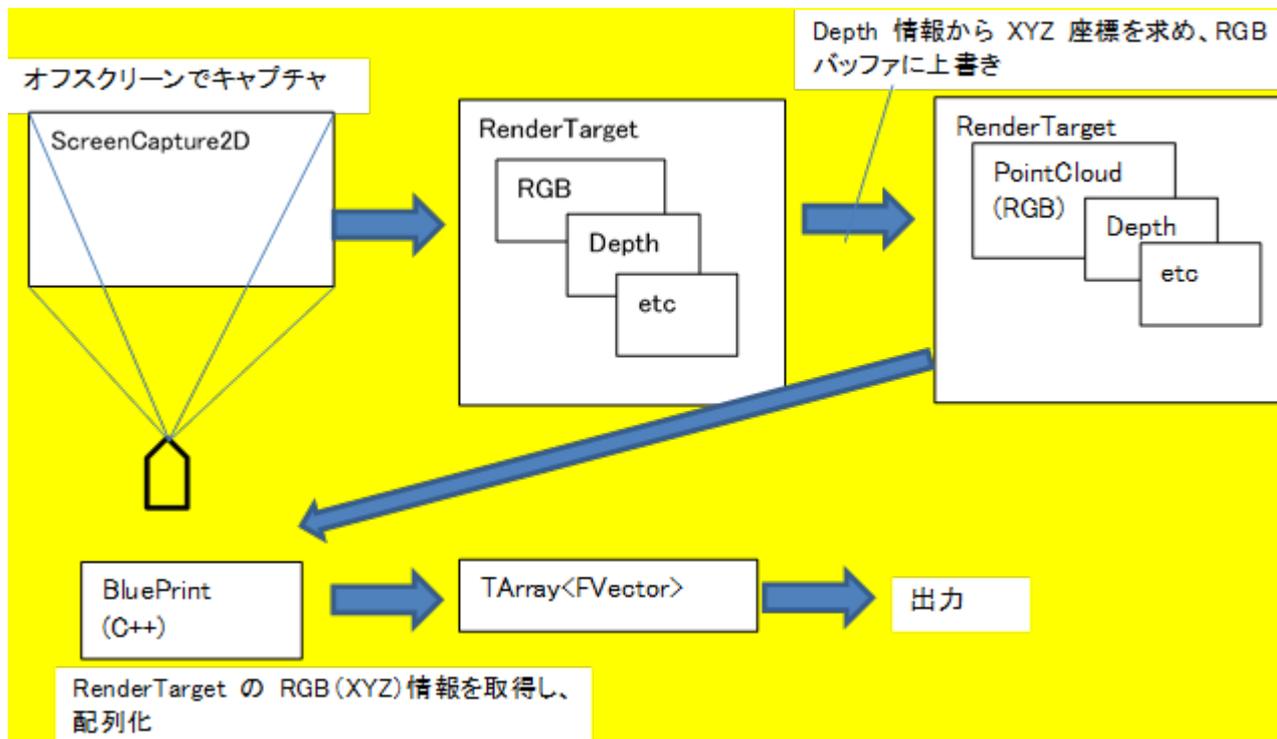


図 2-2-2-3(4). 深度バッファ利用の処理の流れ

2) ミリ波レーダー

センサモデルではレーダー範囲を模した形状のオブジェクトと、検出対象オブジェクトとの交差判定により検出を行う。センサ範囲のオブジェクトとの交差を判定し、交差したオブジェクトを構成する頂点の中でセンサに最も近い点を検出点とする。

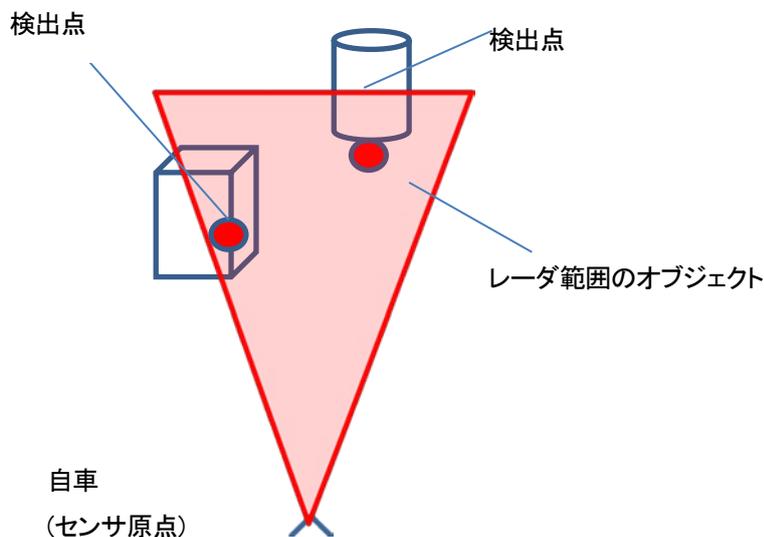


図 2-2-2-3(5). センサ範囲と交差

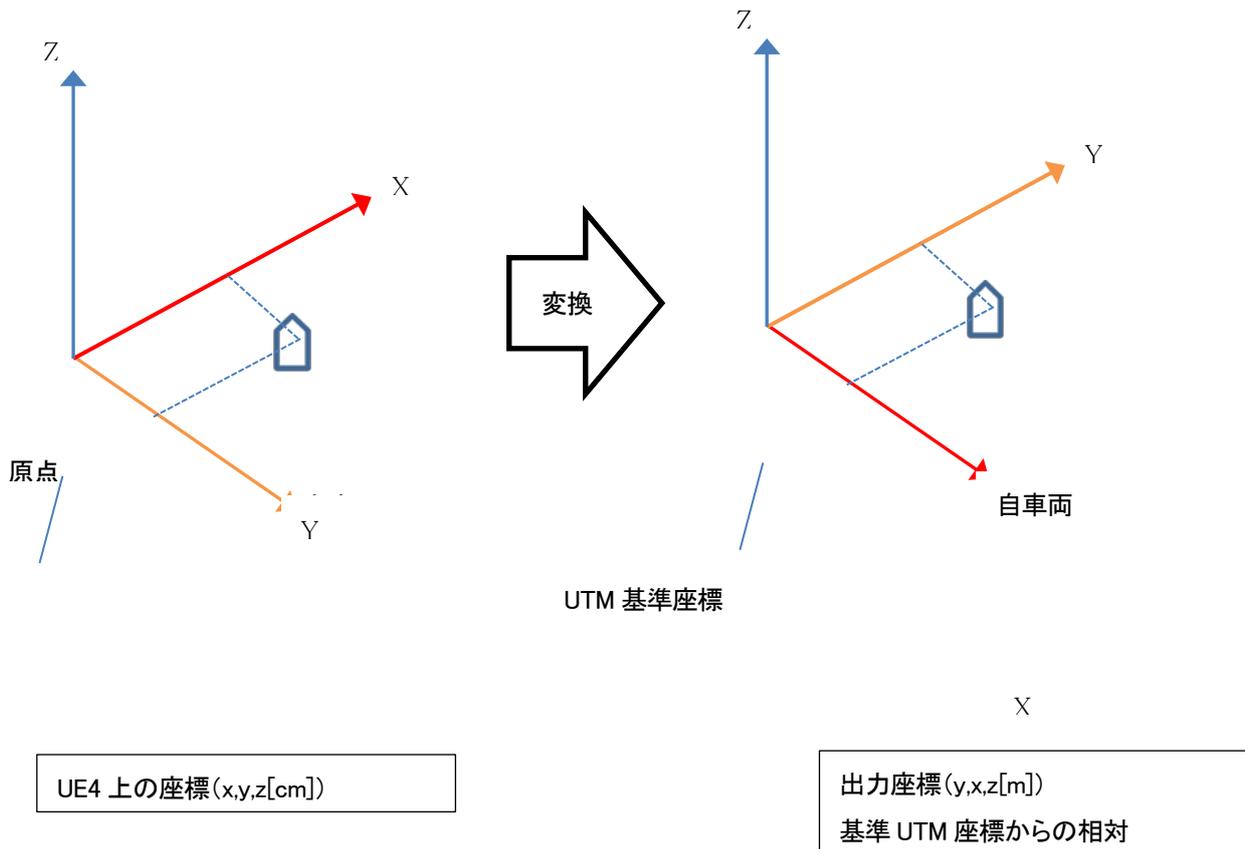
3) GPS

シミュレータ内は左手座標系で構成されているため、出力は右手座標系に変換を行う。

距離単位についても、シミュレータ内の cm から m 単位に変換し、出力を行う。

UTM 座標として出力するため、ユーザ設定により基準となる UTM 座標を決める。

UE4 上の座標(x,y,z(cm))を座標系、距離単位を変換し、基準 UTM 座標から相対的な UTM 座標として算出し、出力する。



4) IMU

シミュレータの車両モデルから速度、角速度情報を取得し、出力情報を算出する。

回転方向について図 12 に示す。

シミュレータ内は左手座標系で構成されているため、各出力情報は右手座標系に変換を行う。

距離単位についてもシミュレータ内の cm から m 単位に変換し、出力する。

座標系および距離単位については図 11 参照。

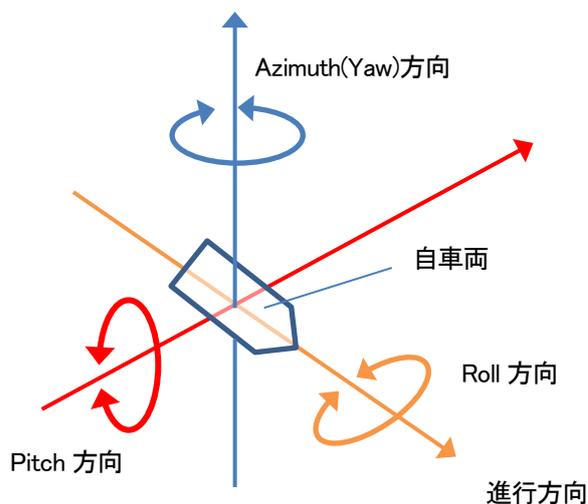


図 2-2-2-3(7). 車両の回転方向

表 2-2-2-3(8). 各センサ値の算出方法

センサ値名	単位	算出方法
角速度(Roll 方向)	deg/s	各方向: (現在の角度-前回の角度)/(現在の時間-前回の時間) ※1
角速度(Pitch 方向)		
角速度(Azimuth 方向)		
角加速度(Roll 方向)	deg/s ²	各方向: (現在の角度-前回の角度)/(現在の時間-前回の時間) ² ※1 ※2
角加速度(Pitch 方向)		
角加速度(Azimuth 方向)		
車両向き(Roll 方向)	deg	各方向: 現在の角度 / 360 の余り(剰余)
車両向き(Pitch 方向)		
車両向き(Azimuth 方向)		
速度ベクトル (東方向、X 軸向き)	m/s	速度値./100 [cm/s から m/s へ変換]
速度ベクトル (北方向、Y 軸向き)		
速度ベクトル (垂直方向、Z 軸向き)		

※1 整数型で出力するため、算出値を 100 倍した後、整数部を出力する。

※2 重力影響あり、重力影響なしの値については同じ値を使用する。

2-2-3. 積雪環境仮想シミュレータの動作検証

2-2-3-1. 動作検証項目の作成

開発するシミュレータについて、動作検証項目を検討した。

初年度は試作であるため、基本的な部分のみの確認とする。

●通信テスト

【ROS⇒UE4】

- ・目標速度、目標舵角が正しくシミュレータに送信されることを確認する。

【UE4⇒ROS】

○CAN通信データ(各車両制御ECUが出力するデータ)

- ・各出力値がシミュレータ上の値と一致することを確認する。

○LiDAR

- ・物体を認識している。
- ・各出力値がシミュレータ上の値と一致することを確認する。

○ミリ波レーダー

- ・物体を認識している。
- ・各出力値がシミュレータ上の値と一致することを確認する。

○GPS

- ・物体を認識している。
- ・各出力値がシミュレータ上の値と一致することを確認する。

○IMU

- ・物体を認識している。
- ・各出力値がシミュレータ上の値と一致することを確認する。

●動作テスト

- ・車両挙動が指令値を受けた速度、旋回速度になっていることを確認する。

2-2-3-2. 動作検証の実施

積雪環境仮想シミュレータの動作検証を実施した。自動運転機能はサブテーマ1で並行して開発中であるため、別途動作検証用のアプリを作成し動作検証を進めることとした。自動運転を行う実機では、自動運転機能が CAN 通信を行うノードを用いて、「目標速度」、「目標ヨーレート」を車両制御 ECU に送信することで自動運転を実現している。対してシミュレーション環境では CAN 通信を行うノードを、シミュレータ用の通信ノードに置換する。実機と CAN 通信を行う代わりに、シミュレータと UDP 通信を行い、仮想車両を操作する。下記 GUI(図 3-2-2-3(9))はその通信ノードに向けて、トピックを送信するアプリである。

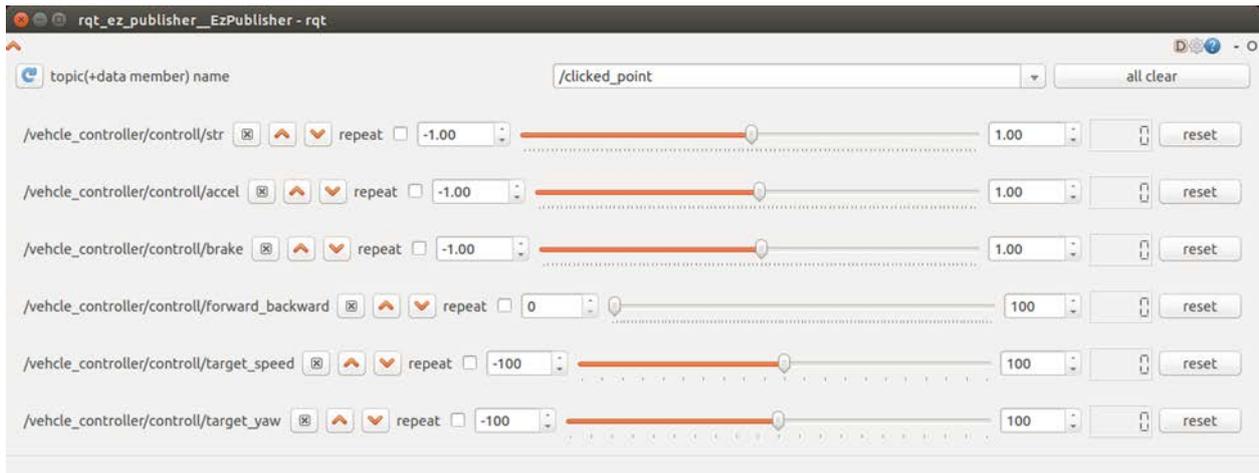


図 2-2-2-3(9). 積雪環境仮想シミュレータ動作検証用 GUI

2-2-4. センサモデルの追加開発

2年度は、積雪環境仮想シミュレータに対して、下記3つのセンサモデルの追加を実施する。

- ・ LiDAR (HDL-32e)
- ・ 全方位マルチカメラ (LadyBug5+)
- ・ 赤外線サーモグラフィ (PI450)

2-2-4-1. LiDAR Velodyne HDL-32e

1) 模擬対象のセンサ仕様

本シミュレータに対して、Velodyne社のLiDAR「HDL-32e」を模擬対象のセンサとして追加する。模擬対象のセンサ仕様を下記に記す。

表 2-2-4-1(1) センサ仕様(Velodyne HDL-32e)

仕様	説明
測定範囲・測定視野	水平 360° 全方位 / 垂直 41.3° (+10.67° ~ -30.67°)
測定距離	約 100m (1m~100m)
測定スピード	5~20Hz
測定ポイント数	約 700,000 ポイント/秒
測定精度	±2cm (1σ@25m)
重量	約 1.2kg
寸法	高さ 144.2mm × 直径 85.3mm
耐衝撃	500m/秒 ² 振幅、11 ミリ秒間
耐振動	5 Hz ~ 2000Hz、3 Grms
耐環境	IP67 対応
環境温度	動作時 : -10°C ~ 50°C、保管時 : -30°C ~ 90°C
出力インターフェース	イーサネット 100Mbps (IP アドレス変更可)
データ	UDP パケット(距離、回転角度データ)

2) センサモデルの実装検討

2017年度に実装した、Velodyne社のLiDAR「VLP-16」と同様の手法で実装するため省略する。詳細は、「H29年度研究成果報告書.docx」を参照とする。

3) センサモデルの実装結果

センサモデルの実装結果について記載する。以下に、シミュレーション中の画面と、設定画面及び、設定可能項目を記す。

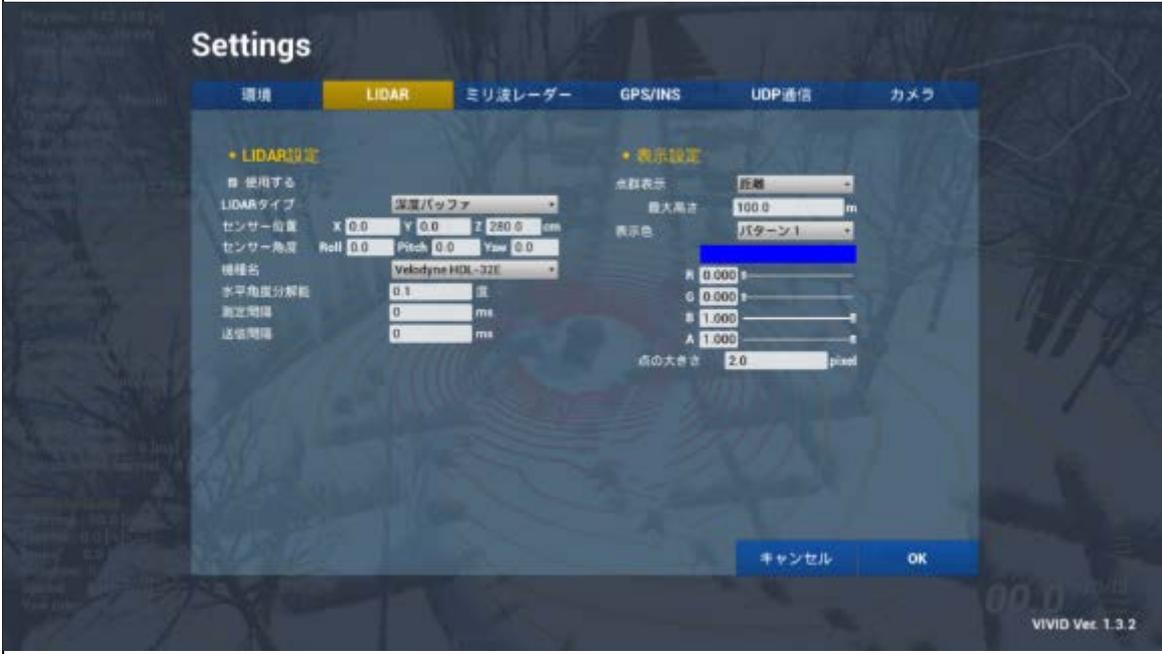
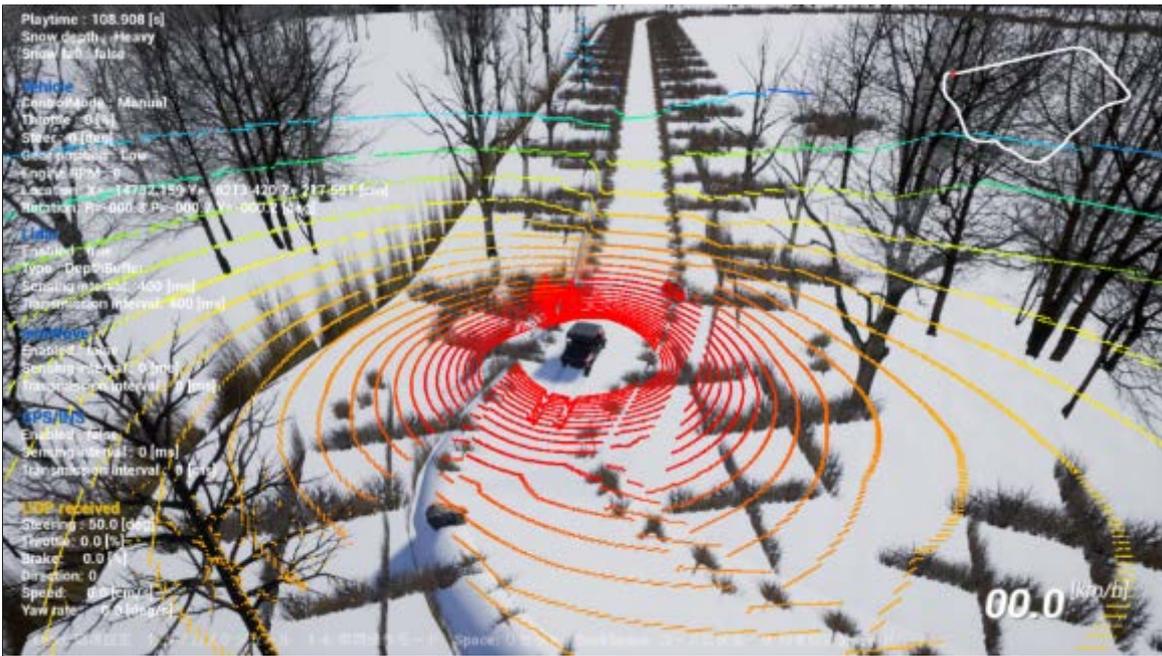


図 2-2-4-1 LiDAR Velodyne HDL-32e の実装結果

表 2-2-4-1(2) LiDAR Velodyne HDL-32e の設定値

分類1	分類2	名称	設定値	デフォルト値	備考
LIDAR	LIDAR	有効/無効	有効		
			無効	○	
		LIDAR タイプ	深度バッファ	○	
			ライトレース		
		センサ位置	(x, y, z) -500.0~500.0cm	(0, 0, 250)	車両中心からの 相対位置
		センサ角度	(roll, pitch, yaw) -180.0~180.0	(0, 0, 0)	基準は車両進 行方向
		機種名	VLP-16 HDL-32E	VLP-16	
		水平角度分解能(水平)	0.1~360.0°	0.1°	VLP-16 は 0.1 ~0.4° を設定 できる
		測定間隔	ms 指定	0	
		送信間隔	ms 指定	0	
	表示設定	点群表示	なし		
			高さ	マルチカメラ	
			距離		
		表示色	パターン1	○	
			パターン2		
			単色		
(r, g, b, a)	(0, 0, 1, 1)				
点の大きさ	0.0~10.0 pixel	2.0 pixel			

2-2-4-2. 全方位マルチカメラ Ladybug

1) 模擬対象のセンサ仕様

本シミュレータに対して、FLIR 社の全方位マルチカメラ「LadyBug5+」を模擬対象のセンサとして追加する。模擬対象のセンサ仕様を下記に記す。

表 2-2-4-3(1) センサ仕様(LadyBug5+)

仕様	説明
フレームレート	JPEG 圧縮時 : 30FPS、非圧縮時 : 14.5FPS
メガピクセル	30 MP (5 MP x 6 センサ)
彩度	カラー
センサ	Sony IMX264 CMOS, 2/3", 3.45 μm
シャッター	グローバルシャッター

重量	3.0kg
寸法	高さ 160mm × 直径 197mm
耐環境	IP65
環境温度	動作時：-20℃～50℃、保管時：-30℃～60℃
出力インターフェース	USB 3.1 Gen 1

2) センサモデルの実装検討

全方位マルチカメラ Ladybug には、画像取得、全方位画像制作、パノラマ画像制作、カメラ設定を管理するための豊富な機能を備えたソフトウェア開発キット(SDK)が付属している。

本シミュレータでは、LadyBug5+で撮影した画像を全方位画像カラー画像として出力することを模擬対象として、センサモデルの検討を実施する。

a) SceneCaptureCube

SceneCaptureCube で6面のキャプチャを行い、TextureRenderTargetCube を経由してマテリアルを生成する。生成したマテリアルをウィジェットに表示し、画面出力する。

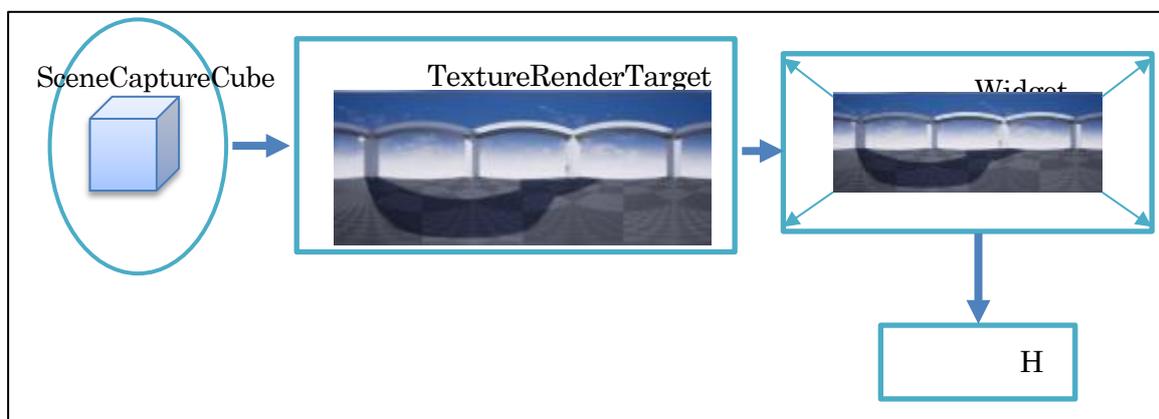


図 2-2-4-2(1) SceneCaptureCube

3) センサモデルの実装結果

センサモデルの実装結果について記載する。以下に、シミュレーション中の画面と、設定画面及び、設定可能項目を記す。



図 2-2-4-2(2) 全方位マルチカメラ Ladybug の実装結果-1



図 2-2-4-2(3) 全方位マルチカメラ Ladybug の実装結果-2

表 2-2-4-2(2)全方位マルチカメラ Ladybug の設定値

分類1	分類2	名称	設定値	デフォルト値	備考
カメラ	全方位カメラ	機種名	Ladybug5+	Ladybug5+	
		アスペクト比	固定	○	
			画面に合わせる		
カメラ位置	(x, y, z) -500.0~500.0cm	(0, 0, 300)	車両中心からの 相対位置		

2-2-4-3. 赤外線サーモグラフィ optris PI450

1) 模擬対象のセンサ仕様

2年度は、積雪環境シミュレータに対して、Optris 社の赤外線サーモグラフィ「PI450」を模擬対象のセンサとして追加する。対象のセンサ仕様を下記に記す。

表 2-2-4-3(1) センサ仕様(赤外線サーモグラフィ optris PI450)

仕様	説明
フレームレート	80Hz
センサ	非冷却 FPA マイクロボロメーター
測定波長域	7.5 ~ 13 μm
測定温度レンジ	-20 ~ 100°C、0 ~ 250°C、150 ~ 900°C
感度 (最大 NETD)	0.04K

測定精度	±2°Cまたは±2%
レンズ	O13 : FOV 13° × 10° 、 O29 : FOV 29° × 22° 、 O53 : FOV 53° × 40° 、 O80 : FOV 80° × 56°
重量	320g
寸法	46 × 56 × 90mm
耐環境	IP67 対応
環境温度	動作時 : 0°C ~ 50°C、保管時 : -40°C ~ 70°C
耐衝撃・振動	25G、IEC68-2-29/2G、IEC 68-2-6
出力インターフェース	USB 2.0

2) センサモデルの実装検討

赤外線サーモグラフィは、センサに入射する赤外線エネルギーを測定し、見かけの温度に変換して、温度分布を画像表示する装置である。センサに入射する赤外線エネルギーは、物体自身が放射するエネルギー以外にも、反射エネルギーや透過エネルギーが含まれる。

本シミュレータは、「物体の放射エネルギー」と「反射エネルギー」を模擬対象として、センサモデルの検討を実施する。

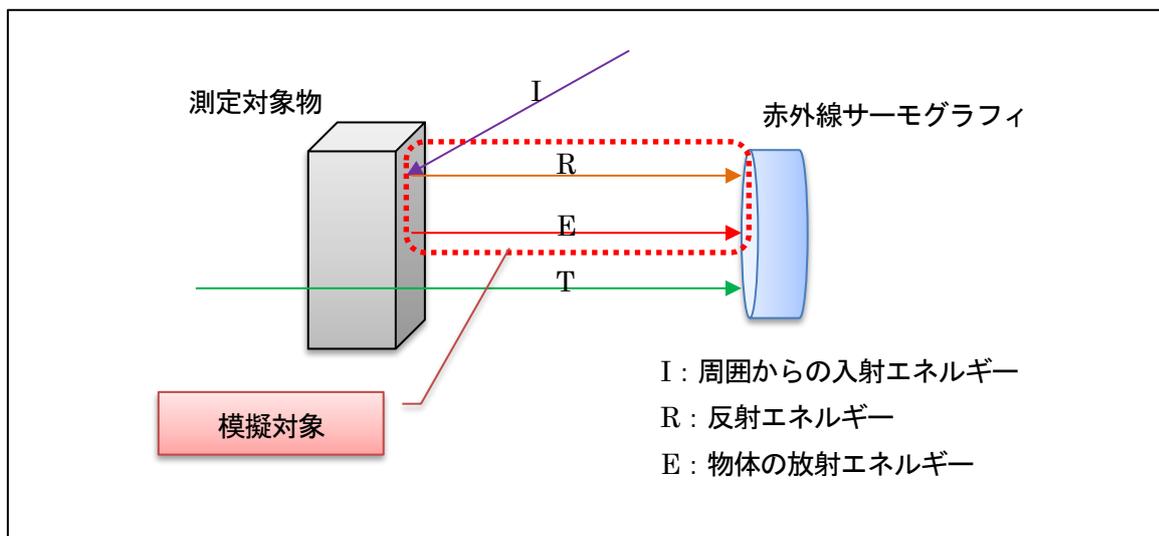


図 2-2-4-3(1) 赤外線サーモグラフィ

a) マテリアルによる温度情報の付加

マテリアルは最低温度を黒、最高温度を白にしたグレースケールとし、カラー変換はポストプロセス処理により行う。

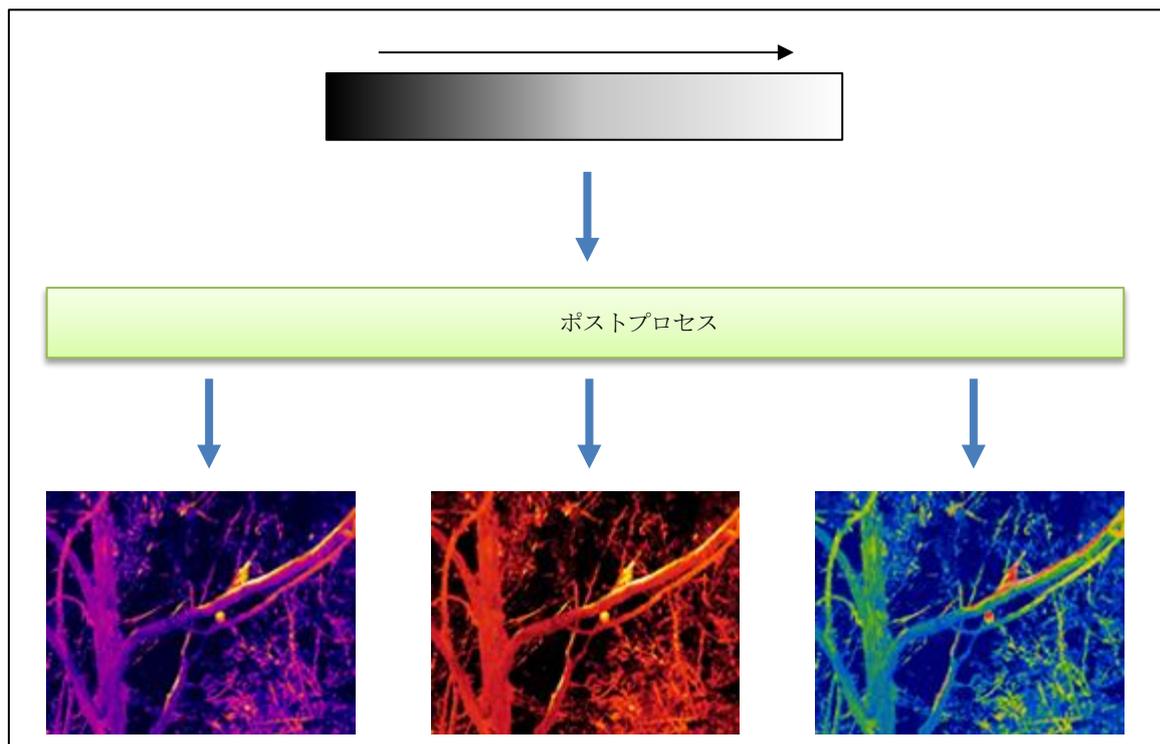


図 2-2-4-3(2) マテリアルによる温度情報の付加

b)温度範囲

マテリアルの黒～白までの範囲が表現可能な温度範囲となる。

必要な範囲は状況により異なるため、各マテリアルを調整する必要がある。

例えば、20°～80°の範囲を表現したい場合は、その範囲を内包した黒を0°、白を100°としたマテリアルを用意する必要がある。

また、閾値により一定値以上（または以下）を切り捨て（切り上げ）することも可能とする。

c)温度変化

温度変化前後のマテリアル（テクスチャ）を用意し、時間で変化させる。

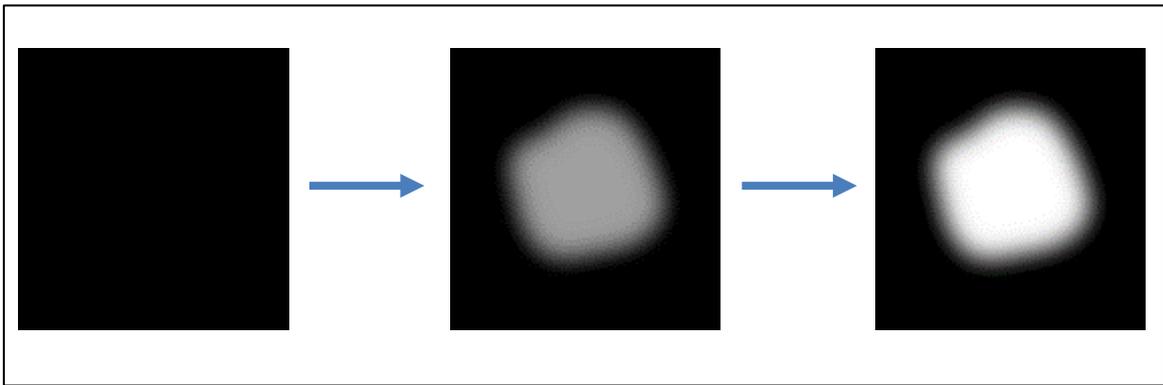


図 2-2-4-3(3) 温度変化

d)可視光画像との切り替え

可視光画像との切り替えを可能とするため、各オブジェクトに可視光用のマテリアルと赤外線サーモグラフィ用のマテリアルを用意する。

表示モード切り替え機能を追加し、表示モードによりマテリアルを切り替える。

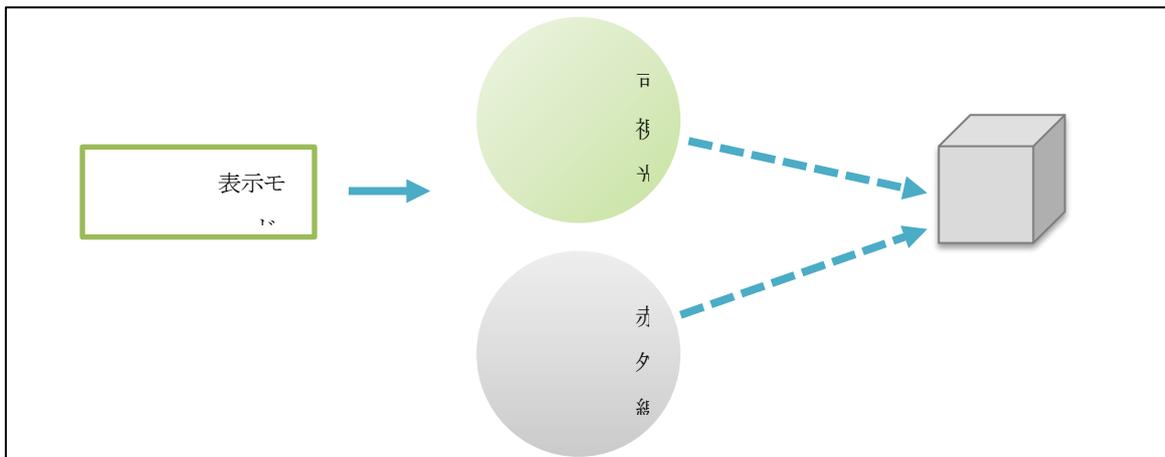


図 2-2-4-3(4) 可視光画像との切り替え

e)赤外線の反射

赤外線の反射を再現するため、オブジェクトの鏡面成分を利用し、温度を合成する。

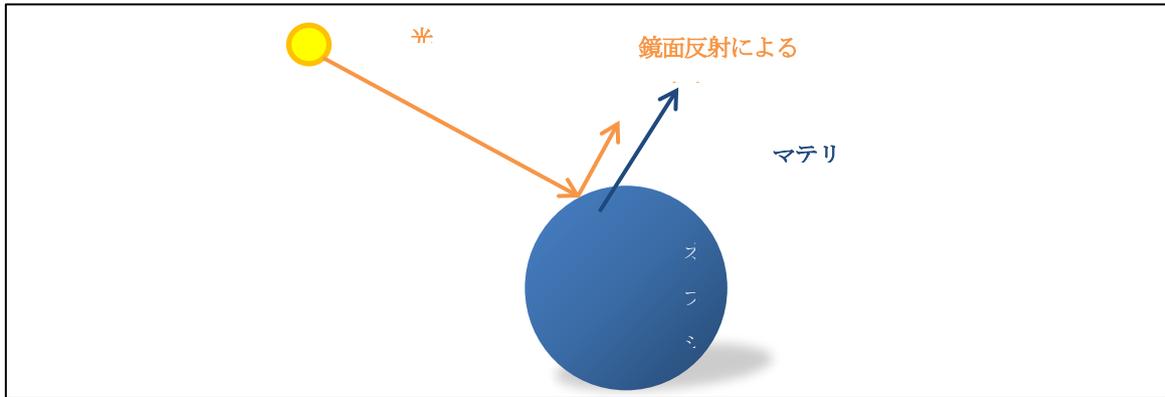


図 2-2-4-3(5) 赤外線の反射

f)マテリアルの温度

本シミュレータでは、物体からの放射エネルギーによる温度を、下記の通りオブジェクトの材質ごとに設定する。ただし設定する温度は、外気温 0°Cで数時間日光に当たった場合の表面温度とする。

表 2-2-4-3(2) オブジェクトの表面温度

材質	表面温度(摂氏)	備考
雪	0	-
砂利	15	-
土	15	-
アスファルト	24	-
草	7	-
樹木	0	-
石	15	-
ガードレール	30	-
鉄塔(鉄)	30	-
空(大気)	-56.5	高度 11 km(36,090 ft)から 20km (65,620 ft)まで、 一定の気温 (-56.5°C)
雲(高度 5000m)	-32.5	1000m 毎に 6.5 度低下

3) センサモデルの実装結果

センサモデルの実装結果について記載する。以下に、シミュレーション中の画面と、設定画面及び、設定可能項目を記す。



図 2-2-4-3(6) 赤外線サーモグラフィ optris PI450 の実装結果

表 2-2-4-2(3) 全方位マルチカメラ Ladybug の設定値

分類1	分類2	名称	設定値	デフォルト値	備考
カメラ	サーモグラフィー	機種名	Optris PI450	Optris PI450	
		レンズ	O13		
			O29		
			O53		
			O80	○	
		カラー		○	
					
					
					
					
		カメラ位置	(x, y, z) -500.0~500.0cm	(90, 0, 180)	車両中心からの 相対位置
カメラ角度	(roll, pitch, yaw) -180.0~180.0	(0, 0, 0)	基準は車両進 行方向		

2-2-5. シミュレーションの改良

2-2-5-1. 知能化 PF との結合

知能化 PF の机上での動作検証を実現するため、シミュレーションと知能化 PF の結合を検討する。

<基本構成の検討>

知能化PF側の実装を極力変更せず、実車とシミュレーションの両方でシームレスな動作検証を可能とするため、ROS側にシミュレーション専用のノード(以降、シミュレーションノードとする)を用意し、実車とシミュレーションの差分を吸収する構成とする。

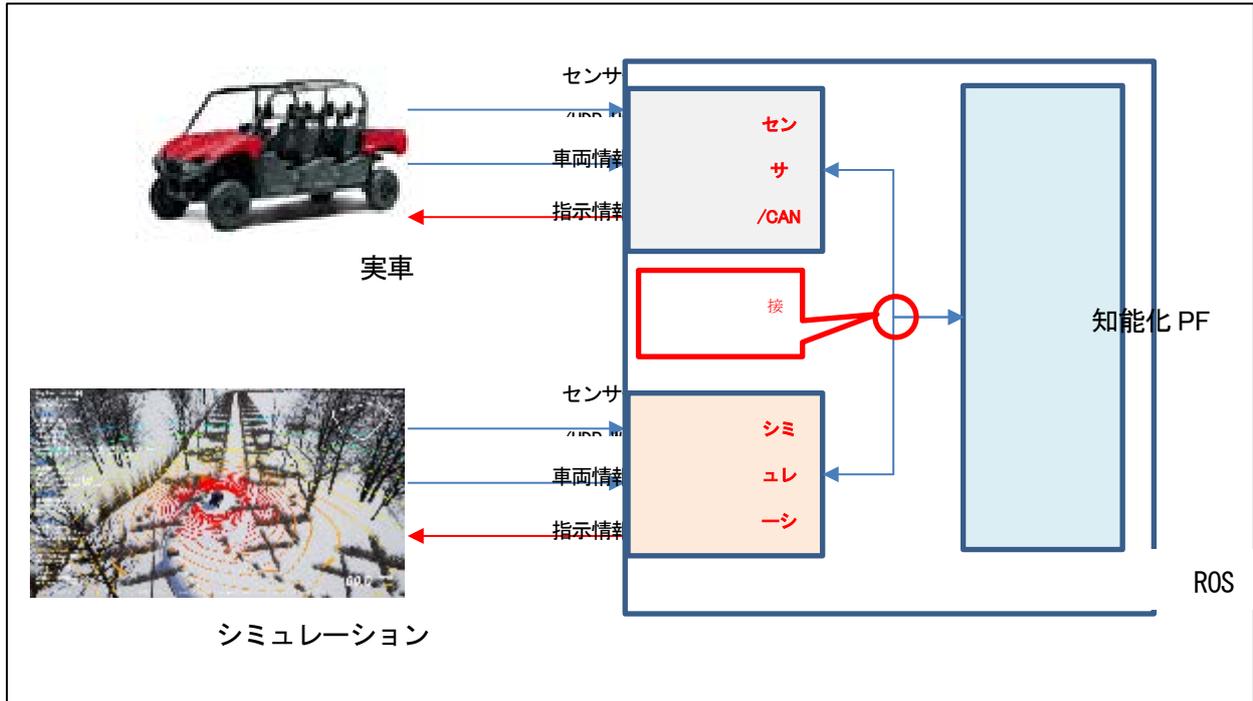


図 2-2-5-1(1) シミュレーションの基本構成

<I/Fの選定>

シミュレーションと知能化PFの間で必要となるI/Fを明確にする。

基本的に、シミュレーションからはセンサ値と車両情報を出力し、知能化PFからは車両への指示を出力する。

表 2-2-5-1(1) シミュレーションと知能化PFの間で必要となるI/F

シミュレーション			知能化PF									
概要	詳細	単位		概要	topic名	メンバ				単位		
UDP : 車両情報 1	車両向き(Roll方向)	deg	→	odometry (IMAR)	imar_ odom	pose	pose	orientation	x			
	車両向き(Pitch方向)	deg							y			
	車両向き(Azimuth方向)	deg							z			
									w			
	車両位置 経度	180/2^39 deg	→					position	x	deg		
	車両位置 緯度	180/2^39 deg	→						y	deg		
	車両位置 高さ(標高)	m	→						z	m		
車速情報 (車両前方方向)	m/s	→			twist	twist	linear	x	m/s			
固定値: "world"			→			header	frame_id					
	ROS内の情報: rospy.Time.now()		→					stamp				
UDP : 車両情報 1	角加速度(Roll方向) 重力影響なし	deg/s ²	→	IMU	imu/data	linear_accel	x			m/s ²		
	角加速度(Pitch方向) 重力影響なし	deg/s ²	→					eratio	n	y		m/s ²
	角加速度(Azimuth方向) 重力影響なし	deg/s ²	→								z	m/s ²
	角速度(Roll方向)	deg/s	→					angular_ velocity	x	rad/sec		
	角速度(Pitch方向)	deg/s	→						y	rad/sec		
	角速度(Azimuth方向)	deg/s	→						z	rad/sec		
	固定値: "base_link"					→			header	frame_id		
ROS内の情報: rospy.Time.now()			→			stamp						
UDP : 車両情報 1	車両位置 経度	180/2^39 deg	→	GPS	gps/data	latitude						
	車両位置 緯度	180/2^39 deg	→			longitude						
固定値: "base_link"			→			header	frame_id					
	ROS内の情報: rospy.Time.now()		→					stamp				
UDP: LIDAR			→	LIDAR	Velodyne _packets							
UDP :	車速情報	m/s	→	車両情報	Viking	speed				m/s		

車両情報 1	(車両前方方向)				_state				
	角速度(Azimuth方向)	deg/s	→			yaw_rate			deg/s
USB : カメラ			→	サーモ カメラ	thermal_ image				
UDP: 車両制御 (目標値)	目標車速	cm/s	←	指示	cmd_vel	linear	x		m/s
	目標コーレート	0.1deg/s	←			angular	z		rad/s

<シミュレータノードの設計>

以下の機能を有するシミュレータノードを設計する。

■シミュレーションノードの機能

- 機能1. シミュレーションから送信される UDP のパケットを受信
- 機能2. web カメラ形式のシミュレーション画面を受信
- 機能3. 機能1、2のシミュレーション情報を、実際のセンサと同じ topic データに変換
- 機能4. 機能3の topic を Publish
- 機能5. 知能化 PF の指示情報(topic)を Subscribe
- 機能6. 機能5の topic を UDP のパケットに変換し送信

また、上記機能の実現にあたり2つの案を検討した。

結果、案①が案②に対して実装に要する工数が少なく、性能差がないため案①を採用した。

案	内容	補足
案①	<ul style="list-style-type: none"> ・Linux の socket インターフェースを用いて UDP のパケット送受信を実現する。 ・パケット、topic 間の変換は新規実装する。 	
案②	<ul style="list-style-type: none"> ・既存のパッケージ「rosbridge_suite」を用いて UDP のパケット送受信と、パケット、topic 間の変換を実現する。 	<p>rosbridge_suite は、ROS の topic がウェブソケットを介して通信する機能を提供するサーバである。</p> <p>ウェブソケットを介して topic をやり取りする場合、ROS 外の通信対象(今回はシミュレータ)は JSON インターフェースのデータを送受信する必要がある。</p> <p>シミュレータは JSON に対応していないため、送受信ロジックに改修が必要となる。</p> <p>◆参考： http://wiki.ros.org/rosbridge_suite</p>

<シミュレータノードの実装>

設計に従って、シミュレーションノードを実装した。実装後の rqt_graph は以下の通り。

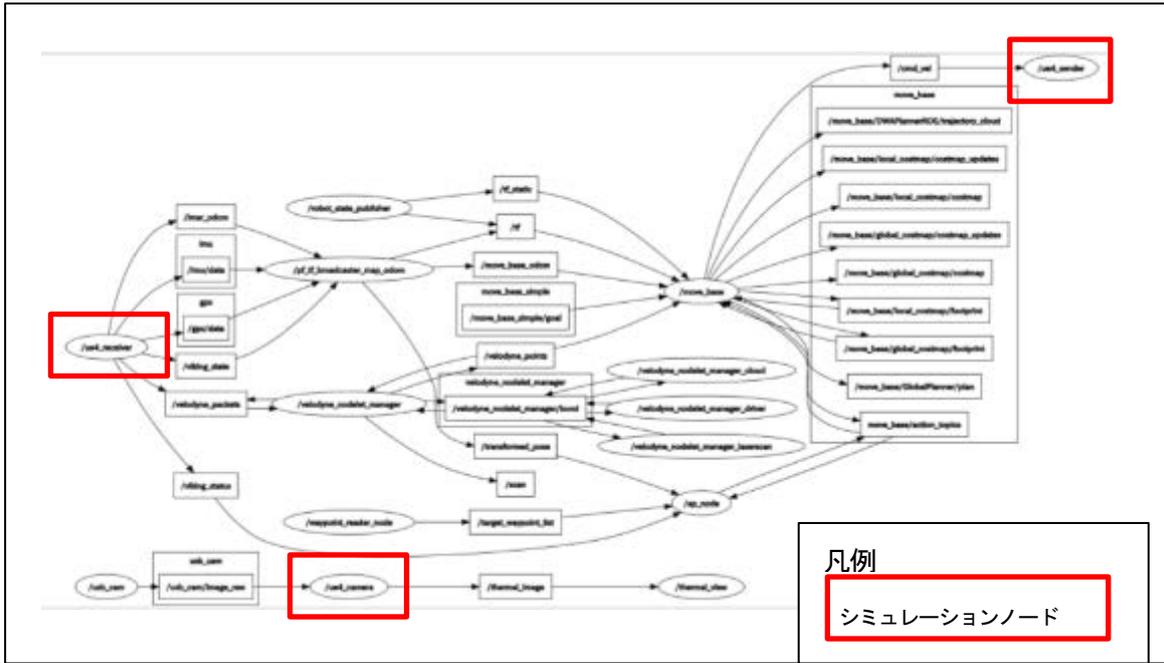


図 2-2-5-1(2) 実装後の rqt_graph

2-2-5-2. 仮想環境改良

2年度は、積雪環境仮想シミュレータに対して、下記3つの改良を検討する。

- ・ 障害物の設置機能
- ・ 道路脇への雪壁追加機能
- ・ 芦別コースの追加

1) 障害物の設置機能

a)機能検討

コース上の任意の場所に、障害物を設置する機能の追加を検討する。

設置する障害物は段ボール箱をイメージした直方体とし、ユーザの簡単な操作で複数配置可能な機能を検討する。

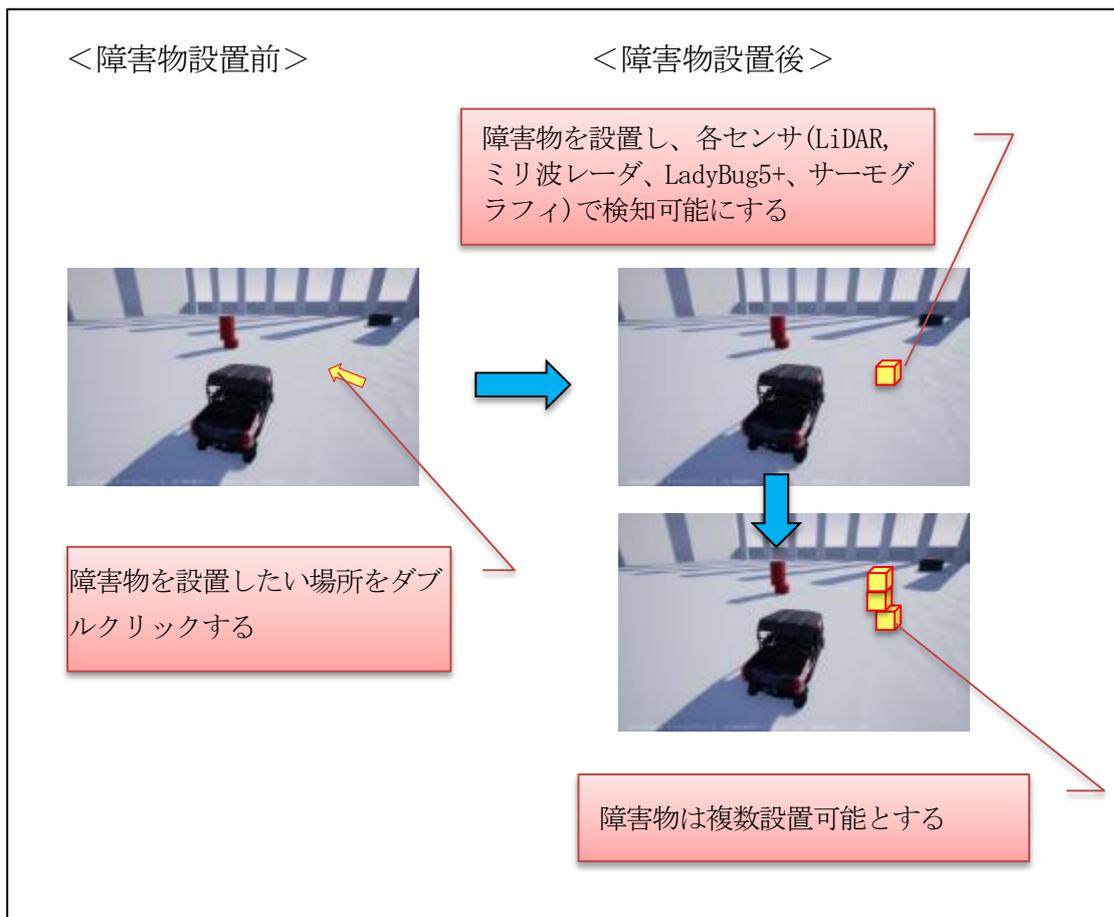


図 2-2-5-2(1) 障害物の設置機能

b)実装結果

障害物の設置機能の実装結果について記載する。以下に、シミュレーション中の画面を記す。

- ・ LiDAR Velodyne HDL-32e

障害物の設置機能によって障害物を設置し、LiDAR Velodyne HDL-32e で検知することができた。

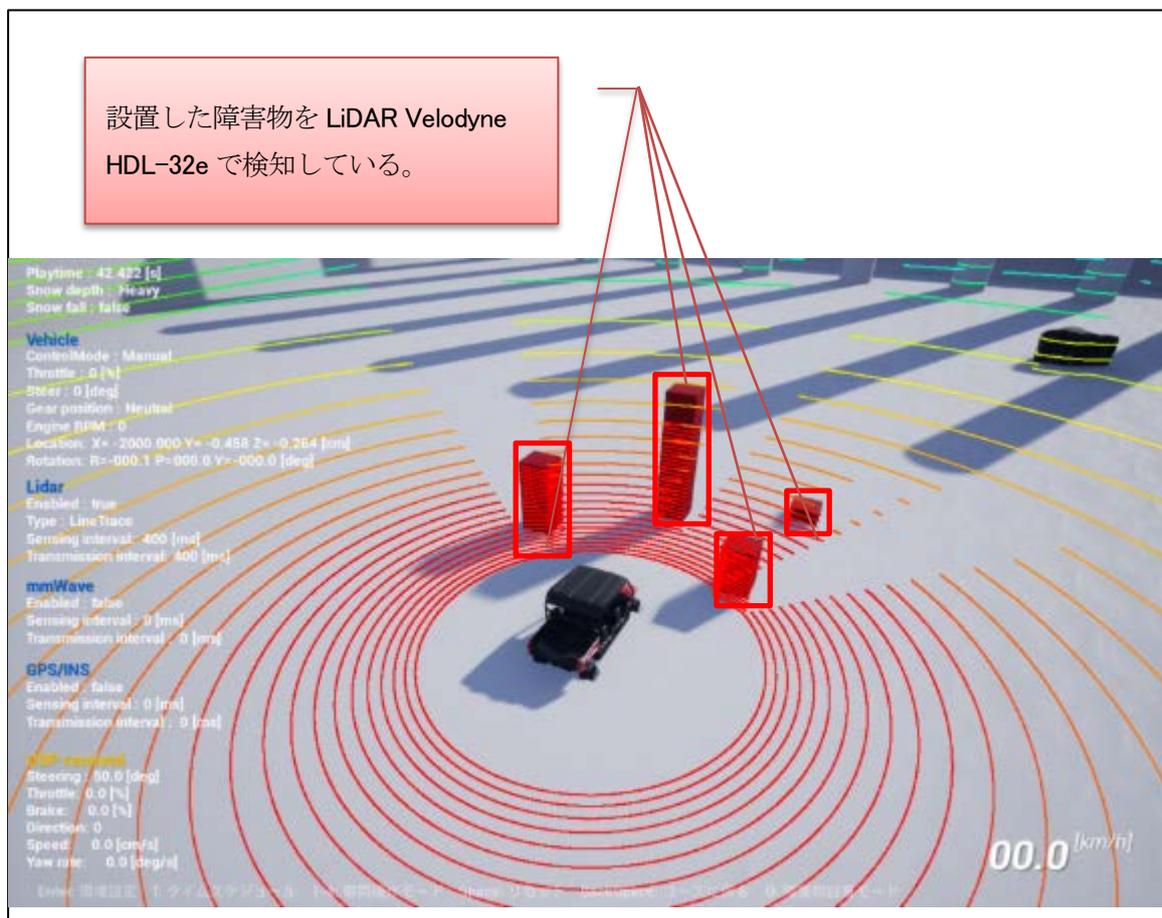


図 2-2-5-2(2) 障害物の検知結果(LiDAR Velodyne HDL-32e)

・ミリ波レーダ

障害物の設置機能によって障害物を設置し、ミリ波レーダで検知することができた。

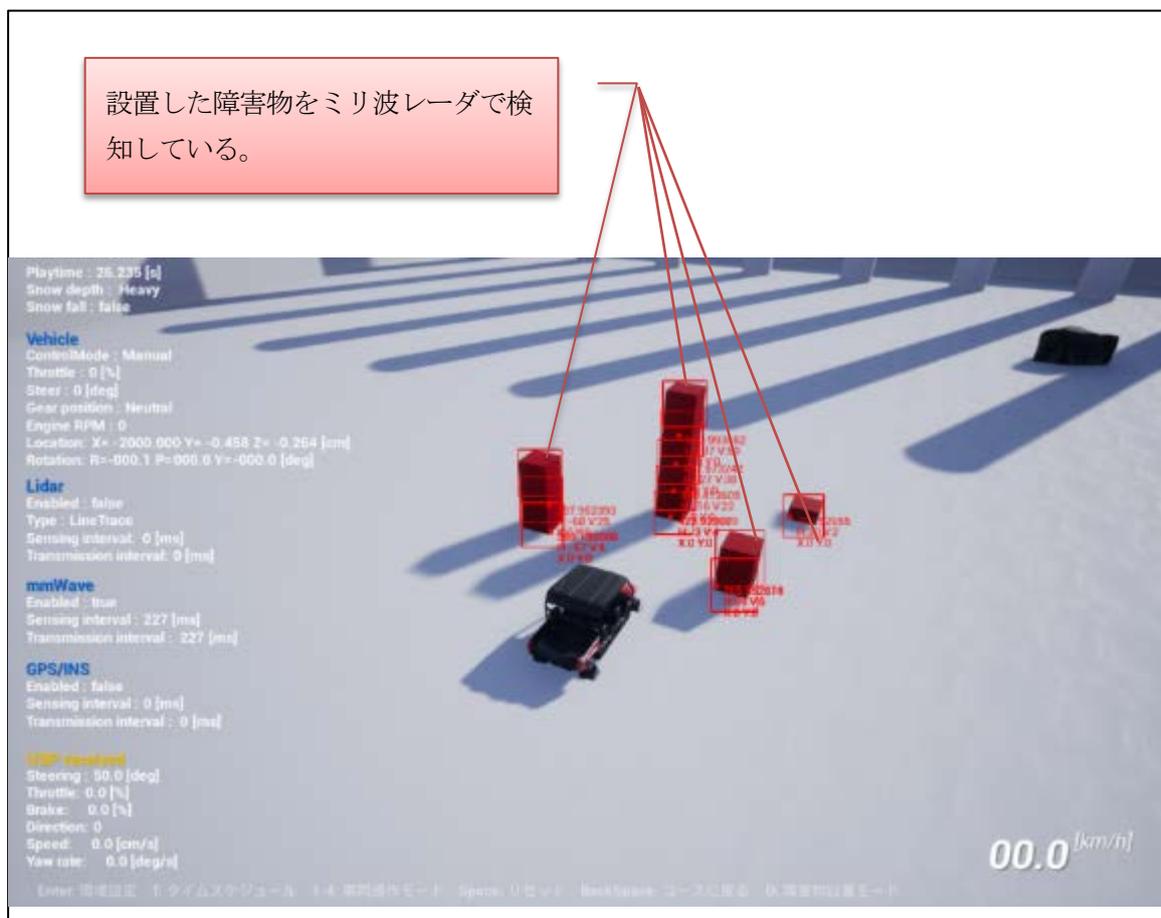


図 2-2-5-2(3) 障害物の検知結果(ミリ波レーダ)

・全方位マルチカメラ Ladybug

障害物の設置機能によって障害物を設置し、全方位マルチカメラ Ladybug で検知することができた。



図 2-2-5-2(4) 障害物の検知結果(全方位マルチカメラ Ladybug)

・ 赤外線サーモグラフィ optris PI450

障害物の設置機能によって障害物を設置し、赤外線サーモグラフィ optris PI450 で検知することができた。

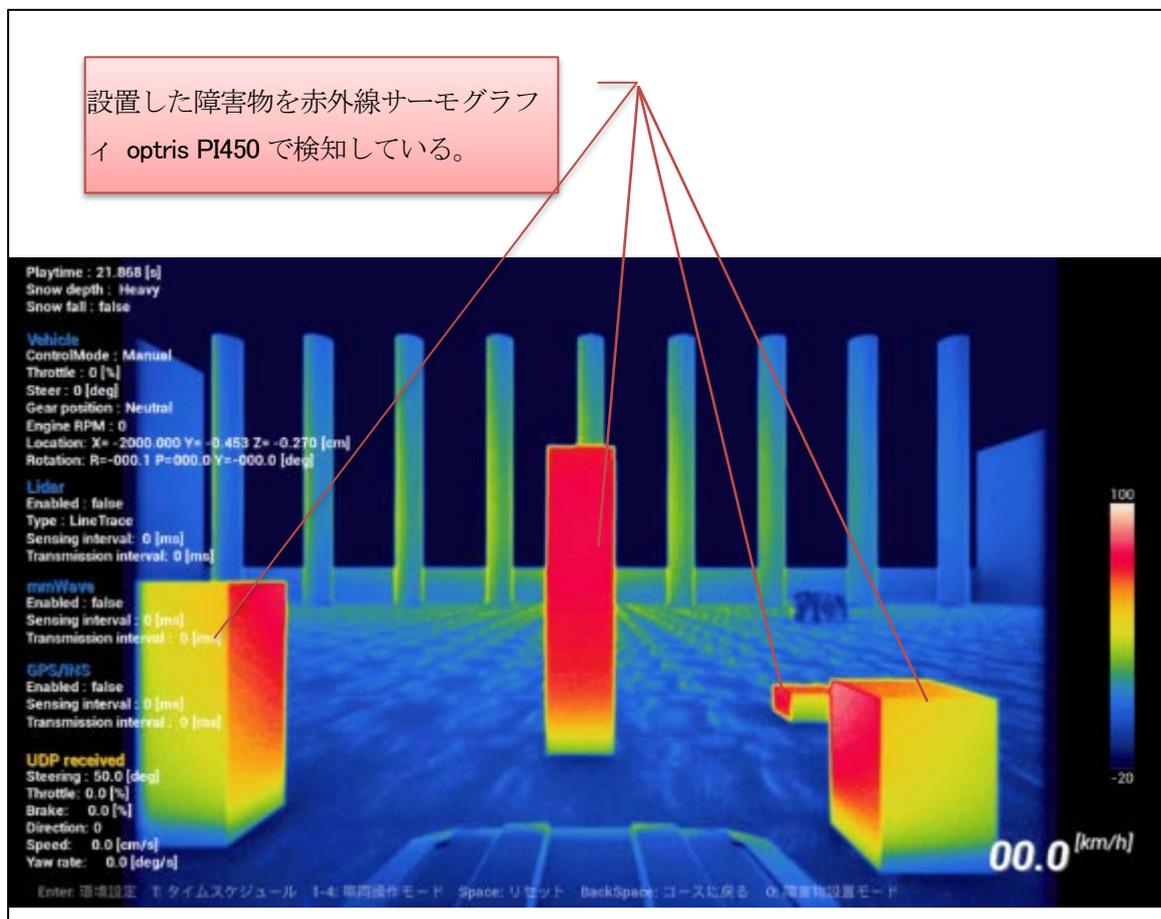


図 2-2-5-2(5) 障害物の検知結果(赤外線サーモグラフィ optris PI450)

2) 道路脇への雪壁追加機能

現実のコースでは除雪作業によって、道路脇に雪壁がある状態での自動運転ケースが想定される。道路脇に一様な雪壁がある状態は、特徴点が少なく、雪道対応 SLAM や雪道対応セマンティック・セグメンテーションを正確に処理できないことが想定される。

以上の理由により、積雪環境仮想シミュレータに対して、「道路脇への雪壁追加機能」の追加を検討する。ただし2年度は検討のみ実施し、実装は3年度に実施とする。

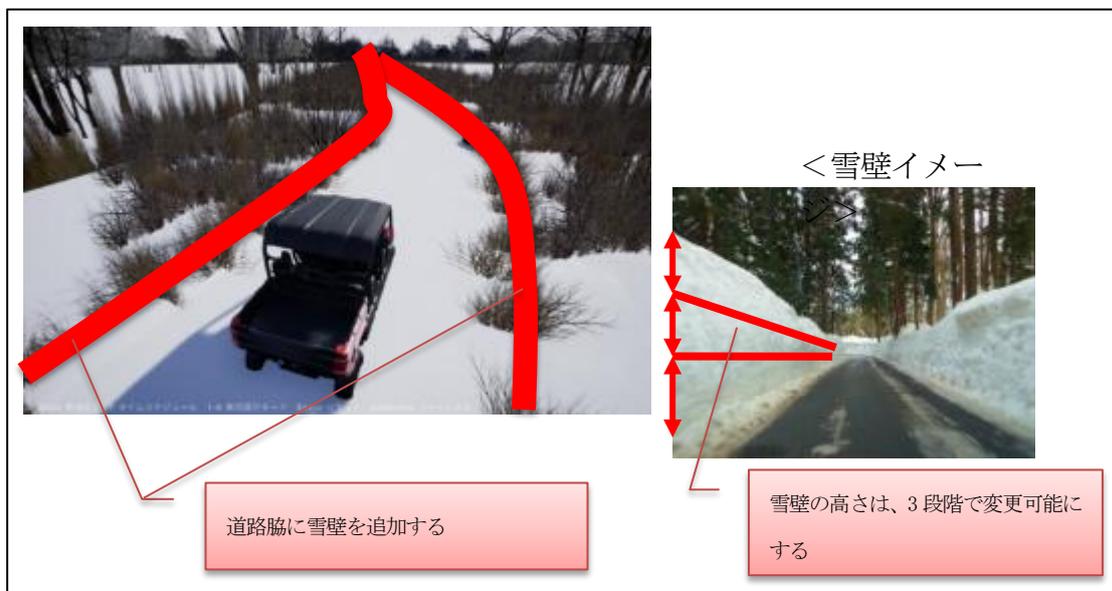


図 2-2-5-2(2) 道路脇への雪壁追加機能

表 2-2-5-2 雪壁追加機能 設定内容

設定	内容
対応コース	(株)苦東・周回路
	(株)苦東・ストレート
	芦別 ※「3) 芦別コース追加」で検討する新コース
雪壁の有無	無し
	あり
雪壁の高さ	30cm, 60cm, 120cm

3) 芦別コースの追加

北日本自動車大学校内テストコース(芦別コース)の追加要望を受けて、コース追加を検討する。
ただし2年度は検討のみ実施し、実装は3年度に実施とする。



図 2-2-5-2(3) 北日本自動車大学校(<http://www.a-tec.jp/>)

検討結果を下記にまとめる。

なお、コースに設置する設置物は、全て下記のセンサで検出できるようにする。

- ・ LiDAR(Velodyne VLP-16/HDL-32)
- ・ ミリ波レーダ(Texas Instruments IWR1642)
- ・ 赤外線サーモグラフィ(Optris I450)
- ・ 全方位マルチカメラ(Ladybug5+)

表 2-2-5-2(2) 芦別コース 検討内容

設定	内容
対応コース	周回オーバルコース(全長 900m)
	※バンクにも対応する
	多目的コース(140m×20m)
設置物	①コース脇の木々
	②道路標識
	③建物
	④電柱
	⑤ガードレール
	⑥道路の分岐

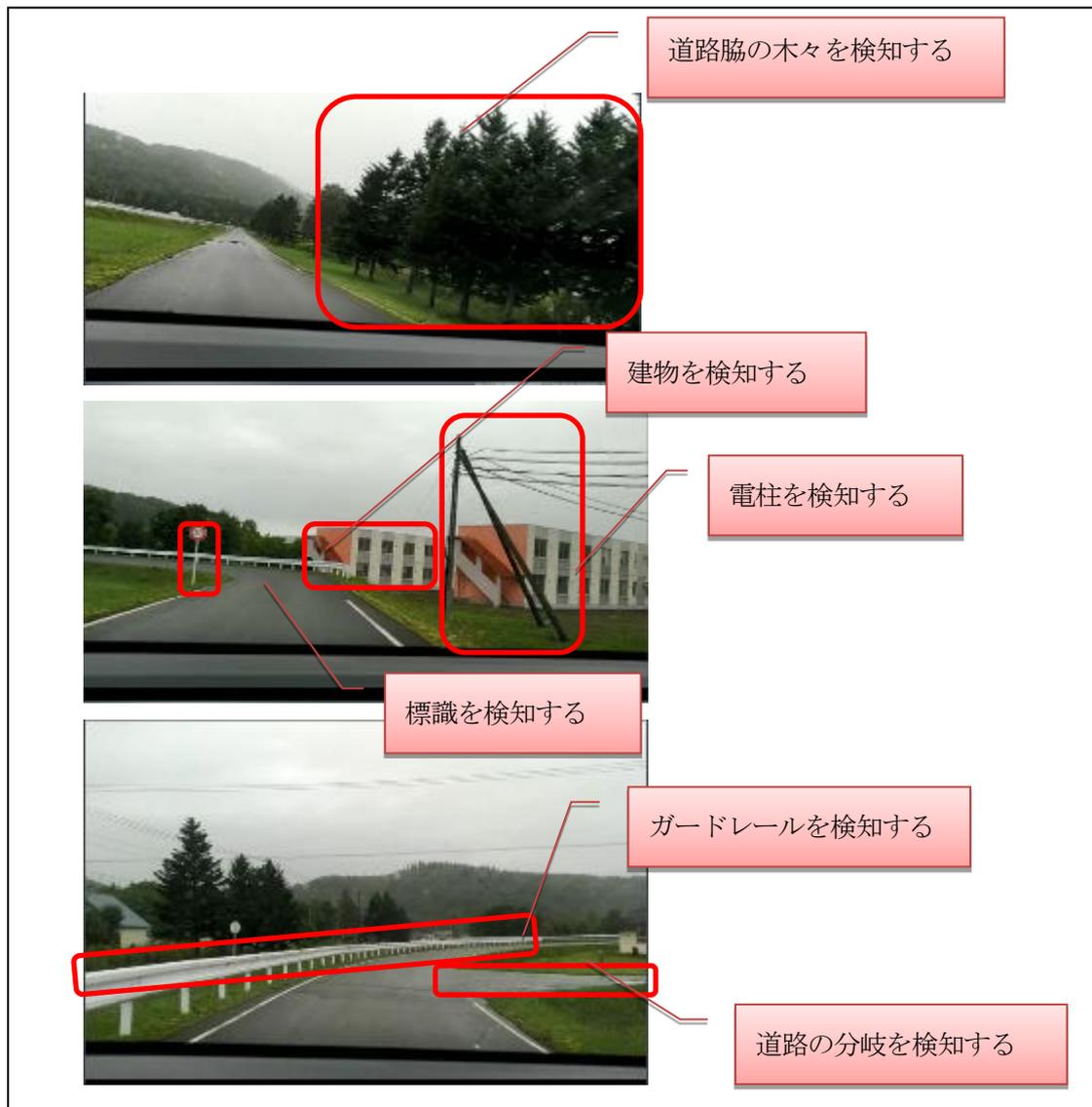


図 2-2-5-2(4) 設置物のイメージ

2-2-5-3. 積雪環境シミュレータに求める性能まとめ

制御検証にシミュレータを活用するために必要と考えるシミュレータの機能の性能を検討する。

自動運転車に用いる制御技術と車両に取り付けるセンサに応じて、シミュレータに求められる機能及び性能が変化する。例えば、グローバル経路計画の技術を用いて自動運転させる場合は、自車両の地点情報を正確に知る必要がある。自車両が地点情報を知るため、GPSを使用するのが一般的である。制御ソフトは、GPSから得られた位置情報から自車両の現在位置を推定して次の目的地まで導く制御を実施する。本研究で使用する制御ごとに求める性能をまとめる。

表 2-2-5-3(1) シミュレーション環境に求めるセンサ性能一覧

要素技術	センサ	機能 / 性能
グローバル経路計画技術	GPS	センサ地点の位置情報として、UTM 座標上の経度、緯度、高さ(標高)を取得できること
雪道対応 SLAM 技術	GPS	センサ地点の位置情報として、UTM 座標上

要素技術	センサ	機能 / 性能
		の経度、緯度、高さ(標高)を取得できること
	IMU	車両挙動情報として、自車両中心点の角速度、角加速度、向きを取得できること
	LiDAR	水平360度、垂直30度の範囲でオブジェクトを検出し、オブジェクトまでの距離を点群データとして取得できること
	ミリ波レーダ	水平±60度の範囲で、オブジェクトを検出し、オブジェクトまでの距離、角度、相対速度を取得できること
	全方位ステレオカメラ	車両中心からの全方位画像をカラー画像として出力できること
	赤外線サーモカメラ	道路、オブジェクトについて、設定した温度情報に基づき、温度分布を画像表示できること
雪道対応セマンティック・セグメンテーション技術	カメラ	車両中心からの前方をカラー画像として出力できること。 前方の範囲については対象カメラの画角情報に応じて設定できること

表 2-2-5-3(2) シミュレーション環境に求める環境性能一覧

項目	要求機能	機能 / 性能
天気	快晴	-
	曇天	雲の量(40%、60%、80%、100%)
	降雨	雨量(少量、中量、多量)
		雨滴サイズ(大、中、小、ランダム)
	霧	霧の濃さ(薄い、濃い)
	雷	雷の量(少ない、多い)
時間	時間に応じた環境変化	時間指定(24時間を1時間刻み)
路面	路面状態	土、アスファルト、コンクリート
	路面摩擦係数	上記路面状態毎に路面摩擦係数を設定できること

表 2-2-5-3(3) シミュレーション環境の車両モデルに求める環境性能一覧

項目	要求機能	機能 / 性能
車両	重量	任意の重量設定
	サイズ	車体規格
	エンジン	小排気量、中排気量、大排気量
	ブレーキ	0~100%
	駆動形式	2 輪駆動、4 輪駆動
	タイヤ	スリップ、スタック

2-2-5-3—(1)シミュレーション性能

積雪環境の仮想シミュレータに求めるセンサ性能を検討する。

1) LiDAR

LiDAR の機能・特性からシミュレータに必要な機能を検討する。

LiDAR はレーザで不可視スペクトルの光パルスを発射し、反射光の到達タイミングと反射強度を計測することで、対象の距離と性質を分析する。

LiDAR の特性上、積雪に関しては雪を透過せず、積雪後の表面状態との距離情報が取得されることになる。

そのため、積雪後に車両が走行したことで積雪面にできる轍の有無も判別することが可能である。

実際には積雪面の細かい凸凹や雪の表面色によるレーザの乱反射によりセンサの検出誤差や、取得する反射強度の強弱に影響を与えられられる。

ただし、現時点では反射強度の用途は予定していない。また、高精度にシミュレーションさせるためには、非常に複雑な演算が必要となる。模擬機能の実装難易度や、シミュレータの実用的な実行速度、開発にかかる費用を考慮すると現時点での実装は難しいと判断する。

降雪時は LiDAR でちらつく雪を検出しまうことにより、ノイズのような点群が検出される。

そのため、降雪量によっては LiDAR で取得したデータが使えなくなる可能性がある。

シミュレータで降雪状態を再現することで降雪時のロジック検証に用いることが可能である。

表 2-2-5-3(4) 積雪環境における LiDAR の模擬に求める性能

項目	要求性能	実装状況
積雪	積もった雪を透過することなく LiDAR で検出すること	○
轍	轍の凸凹形状を LiDAR で検出すること	○
降雪	降ってくる雪を LiDAR で検出すること	×

2) ミリ波レーダ

積雪/降雪状態におけるミリ波レーダの機能・特性からシミュレータに必要な性能を検討する。

ミリ波レーダは、周囲に電波を放射し、反射波の測定、分析を行うことで対象物との距離、方向、大きさを測定する。

ミリ波(millimeter wave)は、波長が1～10mm、周波数が30～300GHzの電波で、天候に強く、長距離(100～200m)での検出が可能な特性を持っている。

雪に対して透過性があり、雪下のオブジェクトの検出が可能と考えられる。ただし、積雪量および雪の密度によってはミリ波の反射信号が減衰する。新雪状態のときは、3cm程度の積雪まで雪下のオブジェクトを検知することが可能だが、それ以上の積雪の場合は減衰率が大きく、検出が困難になるという研究結果(※1)がある。

(※1)富士通テン技報 Vol.34 No.1

ミリ波レーダによるリブ式白線検知

https://www.denso-ten.com/jp/gihou/jp_pdf/62/62-2.pdf

シミュレータとしては、積雪量、積雪密度の調整と、積雪状況に応じた反射信号の減衰を考慮した、検出信号をシミュレーションする必要がある。

表 2-2-5-3(5) 積雪環境におけるミリ波レーダの模擬に求める性能

項目	要求性能	実装状況
積雪	積もった雪の下に存在する物体をミリ波レーダで検出すること	×
積雪密度	密度に応じてミリ波レーダの透過量が増減すること	×
降雪	降ってくる雪はミリ波レーダで検出しないこと	○

3) 赤外線サーモグラフィ

積雪/降雪状態における赤外線サーモグラフィの機能・特性からシミュレータに必要な性能を検討する。

赤外線サーモグラフィは、センサ範囲内の対象物からの赤外線放射を画像化して温度に換算する。

積雪状態下においては、積雪量および積雪下の対象物の放射エネルギー量により、検出状態が変わることが考えられる。検証実験の結果から、赤外線サーモグラフィを使用することで道路面の轍の形状取得ができることがわかっている。轍有無の判断へのセンサ利用を想定して、シミュレータでも轍が判断できる状態の再現が必要と考える。

表 2-2-5-3(6) 積雪環境における赤外線サーモグラフィの模擬に求める性能

項目	要求性能	実装状況
積雪	積もった雪の下に存在する物体を赤外線サーモグラフィで検出すること	×
轍	轍の凸凹形状に応じて赤外線サーモグラフィの検出温度が増減すること	○

4) 前方カメラ

積雪/降雪状態における前方カメラの機能・特性からシミュレータに必要な性能を検討する。

セマンティック・セグメンテーション技術を用いることで、カメラ画像から走行可能な領域判別や障害物の位置を判別することができる。セマンティック・セグメンテーション技術への活用を考慮して、シミュレータが模擬するカメラ画像のリアルな 3DCG の描写が求められる。また、降雪、積雪状態をカメラ画像として取得する機能が必要と考える。

表 2-2-5-3(7) 積雪環境における前方カメラの模擬に求める性能

項目	要求性能	実装状況
積雪	積雪状態の画像を前方カメラで取得できること	○
降雪	降雪状態の画像を前方カメラで取得できること	○
轍	轍の凸凹形状の画像を前方カメラで取得できること	○

2-2-5-3-(2)シミュレーション環境

センサ以外に必要なとする環境性能を検討する。

表 2-2-5-3(8)シミュレーション環境に求める性能

項目	要求性能	実装状況
天気	地面に積もった雪の量を設定できること	○
	積雪状態の画像を取得できること	○
降雪	地面に積もった雪の量を設定できること	○

	降雪状態の画像を取得できること	○
--	-----------------	---

表 2-2-5-3(9) 積雪環境における外界モデルに求める性能

項目	要求性能	実装状況
雪の吹き溜まり	シミュレーション環境に雪の吹き溜まりを配置できること	×
雪の塊	シミュレーション環境に雪の塊を配置できること	×
雪壁	シミュレーション環境に雪壁を配置できること	×

2-2-5-3—(3)テストシナリオ

シミュレーションに実装する性能を動的に変更しながら実行するための性能について記載する。

シミュレーション環境の検討結果をもとにテストシナリオの設定値について検討を実施した内容を下記にまとめる。
2年度は検討であるため、設定項目の精査実施までとする。

1) 基本設定

テストシナリオの基本となる設定を表 2-2-5-3(10)に示す。

表 2-2-5-3(10) 基本設定の一覧

設定項目	設定値	備考
シミュレーション時間倍率	時間倍率(1~100)	シミュレーションの実行速度を任意に設定する
コース設定	コース A, B, C	用意した複数のコースから任意のコースを選択する
積雪深さ	0cm/20cm/30cm/50cm	-

2) 天候設定

天候に関する設定を表 2-2-5-3(11)に示す。

表 2-2-5-3(11) 天候設定の一覧

設定項目	設定値	備考
天気	晴れ、曇り	シミュレーション中に天候を任意に設定する
時間帯	朝、昼、夕方、	設定する時間帯に応じて太陽の位置が変化する

3) 車両設定

車両に関する設定を表 2-2-5-3(12)に示す。

表 2-2-5-3(12) 車両制御設定の一覧

設定項目	設定値	備考
知能化 PF ソフトウェアバージョン	バージョン値	-
制御ソフトウェア定数値		制御ソフトウェアの各定数データ

4) 障害物設定

障害物設定に関する設定を表 2-2-5-3(13)に示す。

表 2-2-5-3(13) 障害物設定の一覧

設定項目	設定値	備考
静止物体	雪の塊、樹木、建物、標識、信号、電柱、雪の吹き溜まり、マンホール	コース上にユーザが任意の場所に設定できるものとする。
動作物体	人、動物、トラック、バス、バイク	コース上にユーザが任意の場所に設定できるものとする。

5) 記録データ

走行データとして保存する記録項目を表 2-2-5-3(14)に示す。

表 2-2-5-3(14) 記録データ一覧

記録項目	記録形式	備考
シミュレーション動画	一般的な動画形式	-
シミュレータ内データ	Rosbag	-

2-2-6. 芦別コースの追加

3年度は、積雪環境仮想シミュレータに芦別コースを追加する。

1) 芦別コースの仕様検討

北日本自動車大学校内テストコース(以下、芦別コース)をシミュレータへ実装するにあたり、模擬するコースおよび設置物の仕様検討を実施した。

模擬対象となる実際の芦別コースの写真を図 3-2-1(1)に示す。



図 3-2-1(1) 北日本自動車大学校(<http://www.a-tec.jp/>)

模擬対象の設置物は、「Snow-SLAM」方式において自動運転の性能検証で必要となる各センサで検出できるものとする。模擬レベルについても性能検証で必要となる模擬内容を基に、シミュレーションの処理速度等を考慮した上で決定した。

コースに設置する設置物は、下記のセンサで検出する。

- ・ LiDAR(Velodyne VLP-16/HDL-32)
- ・ ミリ波レーダ(Texas Instruments IWR1642)
- ・ 赤外線サーモグラフィ(Optris I450)
- ・ 全方位マルチカメラ(Ladybug5+)

性能検証で必要となる各センサで検知すべき設置物を図 3-2-1(2)に示す。

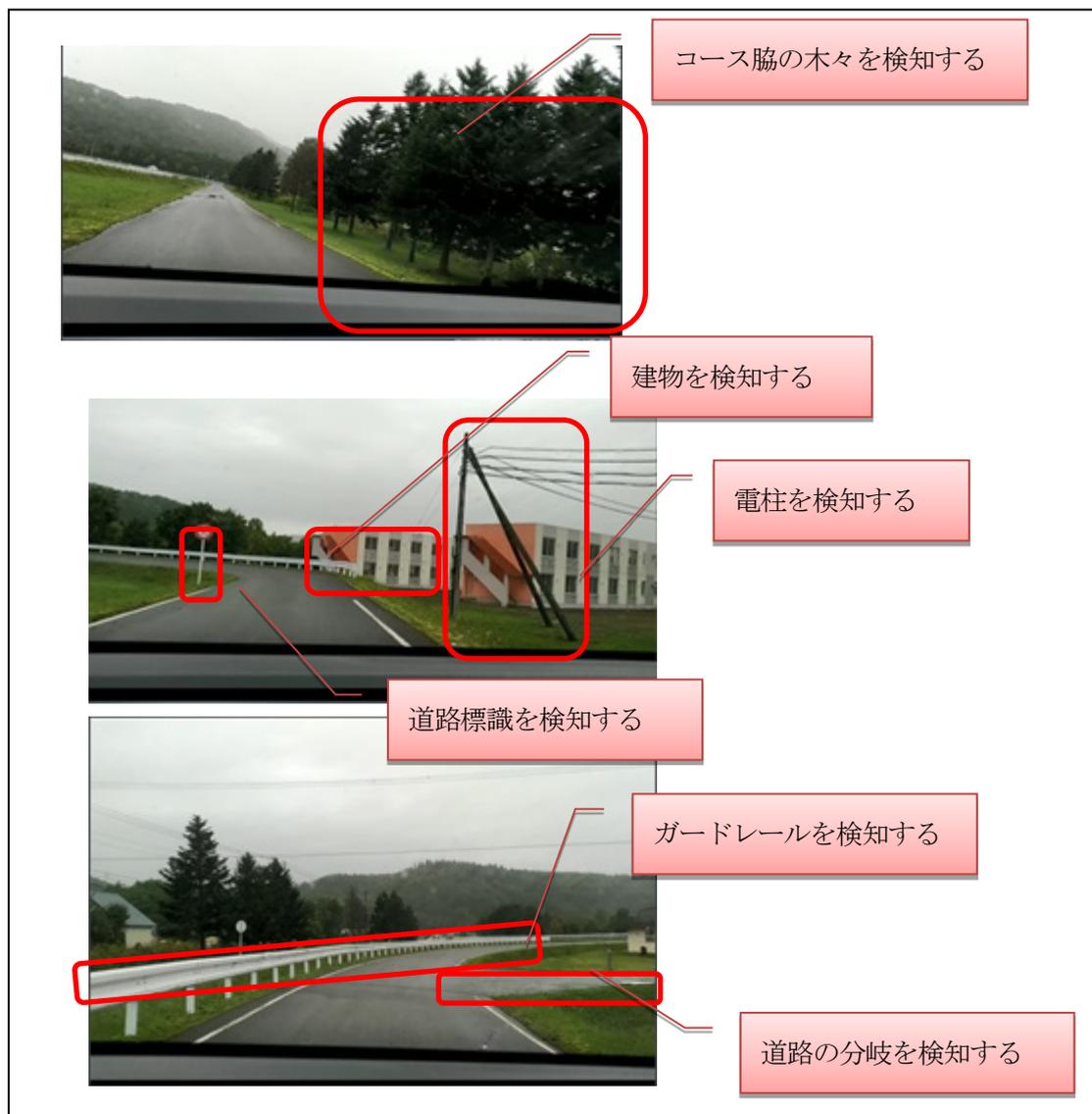


図 3-2-1(2) センサで検知する設置物

図 3-2-1(1)で示す芦別コースのうち、模擬対象とする対応コースおよび設置物を表 3-2-1(1)に示す。表 3-2-1(1)に示した模擬対象の仕様検討を実施した。

表 3-2-1(1) 模擬対象とする対応コースおよび設置物

設定	模擬対象
対応コース	周回オーバルコース(全長 900m) ※バンクも模擬対象に含む
	多目的コース(140m × 20m)
設置物	①コース脇の木々
	②道路標識
	③建物
	④電柱
	⑤ガードレール

対応コースについて実装内容を検討した結果、下記に示す内容とした。

対応コースについてのレイアウトイメージを図 3-2-1(3)に示す。

- ・ 周回オーバルコース(全長 900m)と多目的コース(140m*20m)の 2 コースをまとめて、
 芦別コースとして模擬する
- ・ コース内のバンク(12%バンク、10%バンク)を模擬する

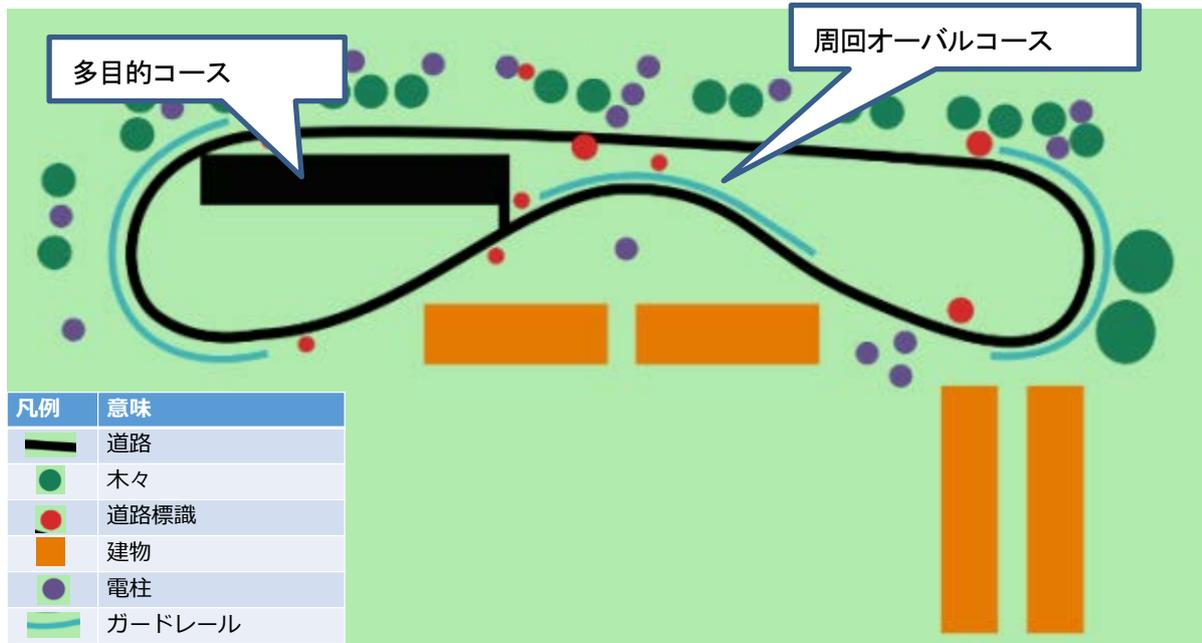


図 3-2-1(3) 芦別コースレイアウトイメージ

次に、芦別コース上に設置物として配置する木々、道路標識、建物、電柱、ガードレールの模擬内容について検討を実施した。

コース脇の木々の模擬内容の検討結果について、表 3-2-1(2)に示す。

また、模擬イメージを図 3-2-1(4)に示す。

表 3-2-1(2) コース脇の木々の模擬内容検討結果

模擬範囲	詳細	実装判断
モデリング範囲		
簡易モデリング	既存の 3D モデルを使用する	○
詳細モデリング	写真,動画を元にモデリングする	×
種類		

	木の種類	針葉樹 1 種類のみ配置する	○
		広葉樹と針葉樹の 2 種類を配置する	×
		写真,動画を元に実際の種類を配置する	×
	木の状態	葉っぱあり固定する	○
		葉っぱなし固定する	×
		葉っぱありなしを選択可能にする	×
周辺環境からの影響			
	雨	影響を受けない	○
		葉っぱを濡らす	×
	雪	影響を受けない	○
		経過時間に合わせて木に雪が積もる	×
	風	影響を受けない	○
		葉っぱや枝が揺れる	×

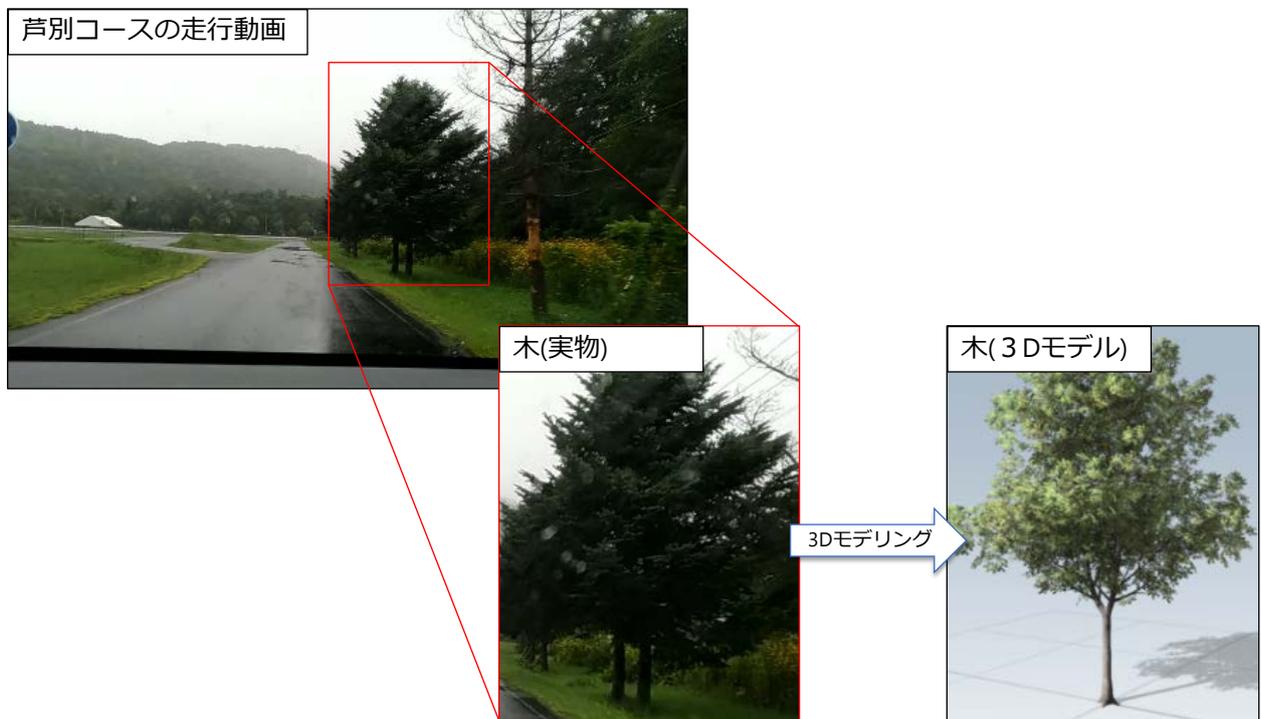


図 3-2-1(4) コース脇の木々の模擬イメージ図

道路標識の模擬内容の検討結果について、表 3-2-1(3)に示す。
 また、模擬イメージを図 3-2-1(5)に示す。

表 3-2-1(3) 道路標識の模擬内容検討結果

模擬範囲	詳細	実装判断
モデリング範囲		
簡易モデリング	既存の 3D モデルを使用する	○
詳細モデリング	写真,動画を元にモデリングする	×
種類		
道路標識の種類	1 種類のみにする(最高速度)	○
	実際に配置している種類を網羅する	×
周辺環境からの影響		
雨	影響を受けない	○
	道路標識を濡らす	×
雪	影響を受けない	○
	経過時間に合わせて道路標識に雪が積もる	×

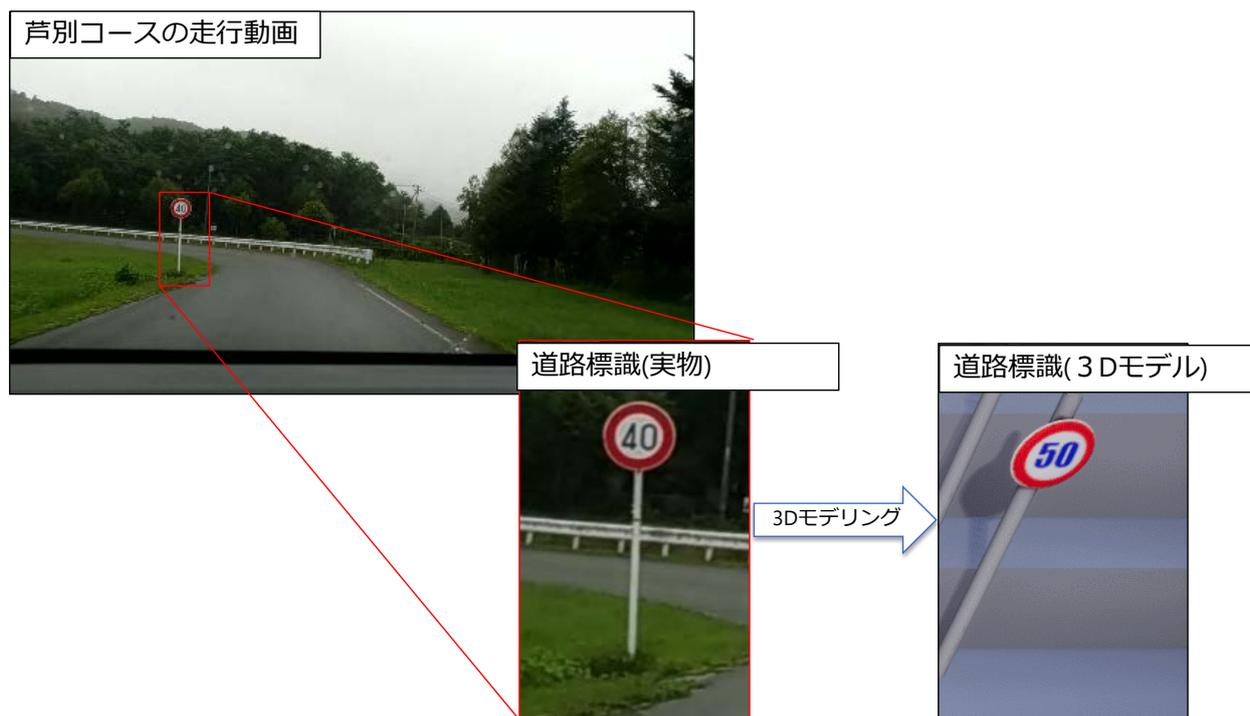


図 3-2-1(5) 道路標識の模擬イメージ図

建物の模擬内容の検討結果について、表 3-2-1(4)に示す。

また、模擬イメージを図 3-2-1(6)に示す。

表 3-2-1(4) 建物の模擬内容検討結果

模擬範囲	詳細	実装判断
モデリング範囲		
簡易モデリング	既存の 3D モデルを使用する	○
詳細モデリング	写真, 動画を元にモデリングする	×
種類		
建物の形状	直方体の 3D モデルにコンクリートのテクスチャを貼る	○
	単色でシルエットが実際の形状になっている	×
	実際の建物を再現する	×
周辺環境からの影響		
雨	影響を受けない	○
	建物を濡らす	×
雪	影響を受けない	○
	経過時間に合わせて建物に雪が積もる	×

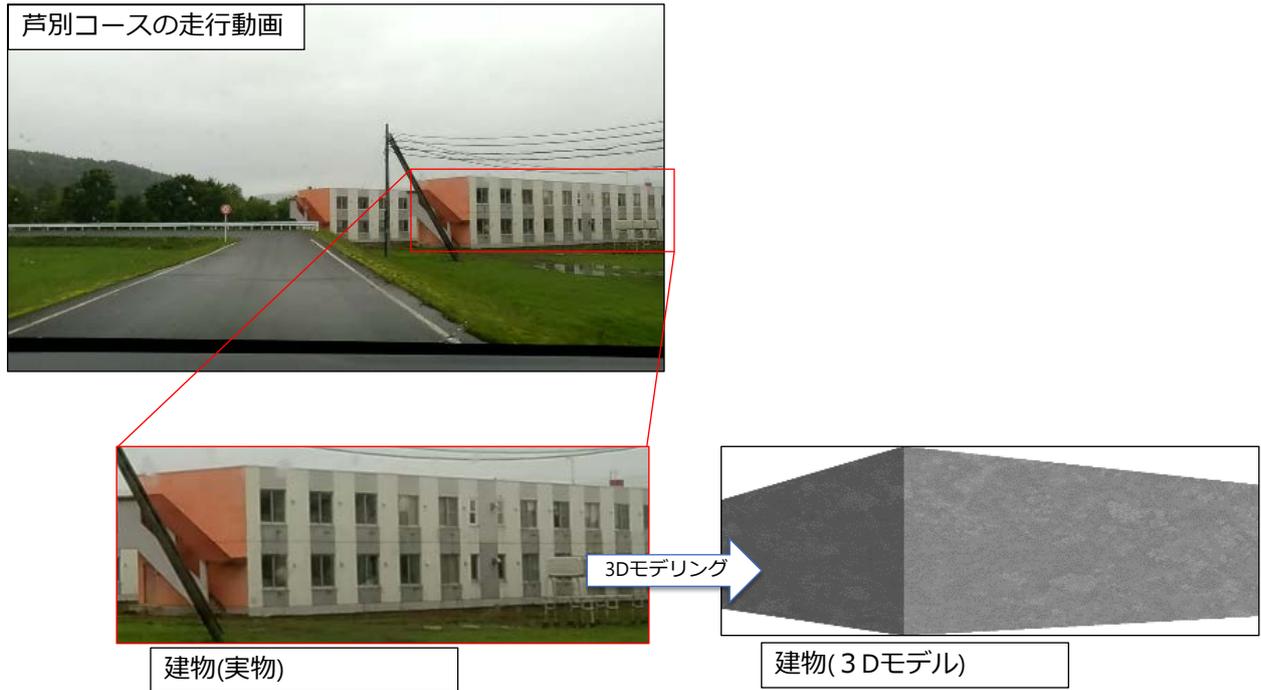


図 3-2-1(6) 建物の模擬イメージ図

電柱の模擬内容の検討結果について、表 3-2-1(5)に示す。

また、模擬イメージを図 3-2-1(7)に示す。

表 3-2-1(5) 電柱の模擬内容検討結果

模擬範囲	詳細	実装判断
モデリング範囲		
簡易モデリング	既存の 3D モデルを使用する	○
詳細モデリング	写真, 動画を元にモデリングする	×
種類		
電柱の種類	1 種類のみにする	○
	3 つ足の電柱にする	×
	実際の電柱を再現する	×
周辺環境からの影響		
雨	影響を受けない	○
	電柱, 電線を濡らす	×
雪	影響を受けない	○

		経過時間に合わせて電柱, 電線に雪が積もる	×
	風	影響を受けない	○
		電線が揺れる	×

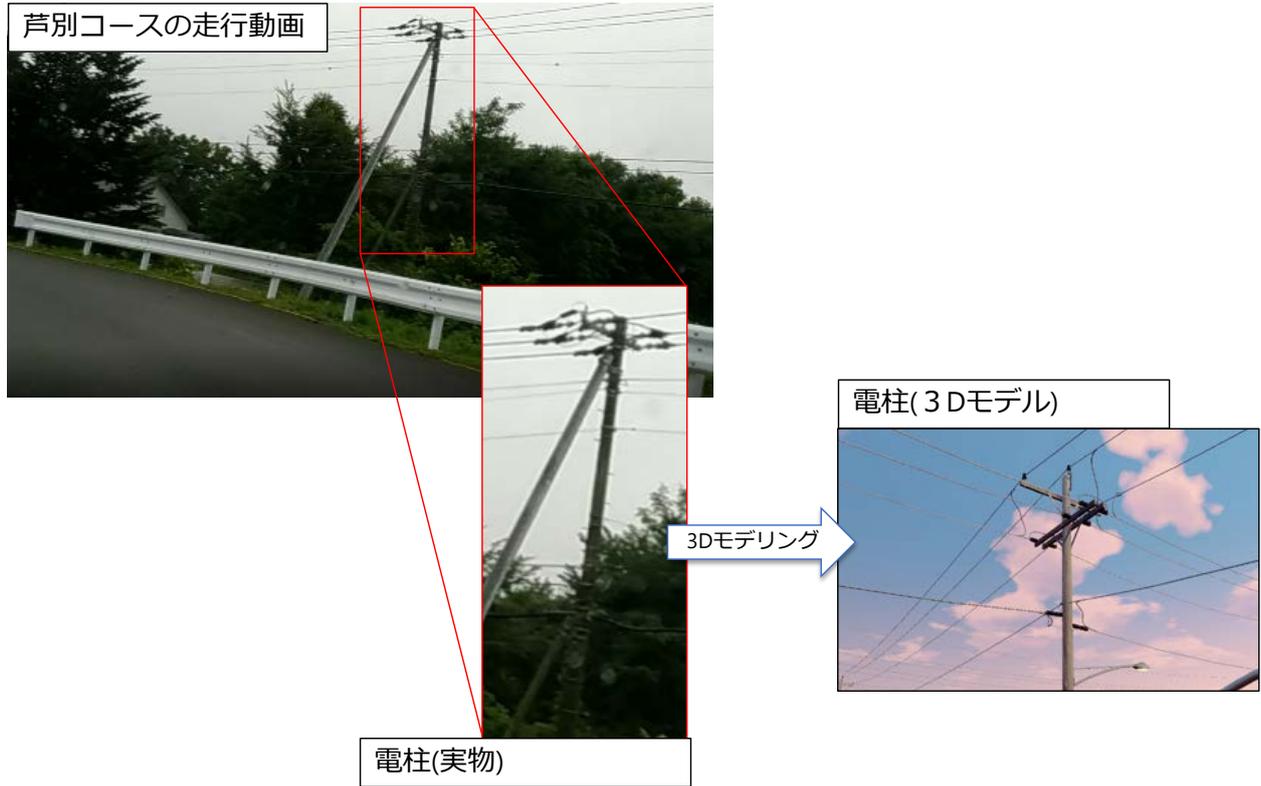


図 3-2-1(7) 電柱の模擬イメージ図

ガードレールの模擬内容の検討結果について、表 3-2-1(6)に示す。

また、模擬イメージを図 3-2-1(8)に示す。

表 3-2-1(6) ガードレールの模擬内容検討結果

模擬範囲	詳細	実装判断
モデリング範囲		
簡易モデリング	既存の 3D モデルを使用する	○
詳細モデリング	写真, 動画を元にモデリングする	×
種類		
ガードレールの種類	1 種類のみにし、留め具部分の凹凸を再現しない	○

		実際のガードレールを再現する	×
周辺環境からの影響			
	雨	影響を受けない	○
		ガードレールを濡らす	×
	雪	影響を受けない	○
		経過時間に合わせてガードレールに雪が積もる	×

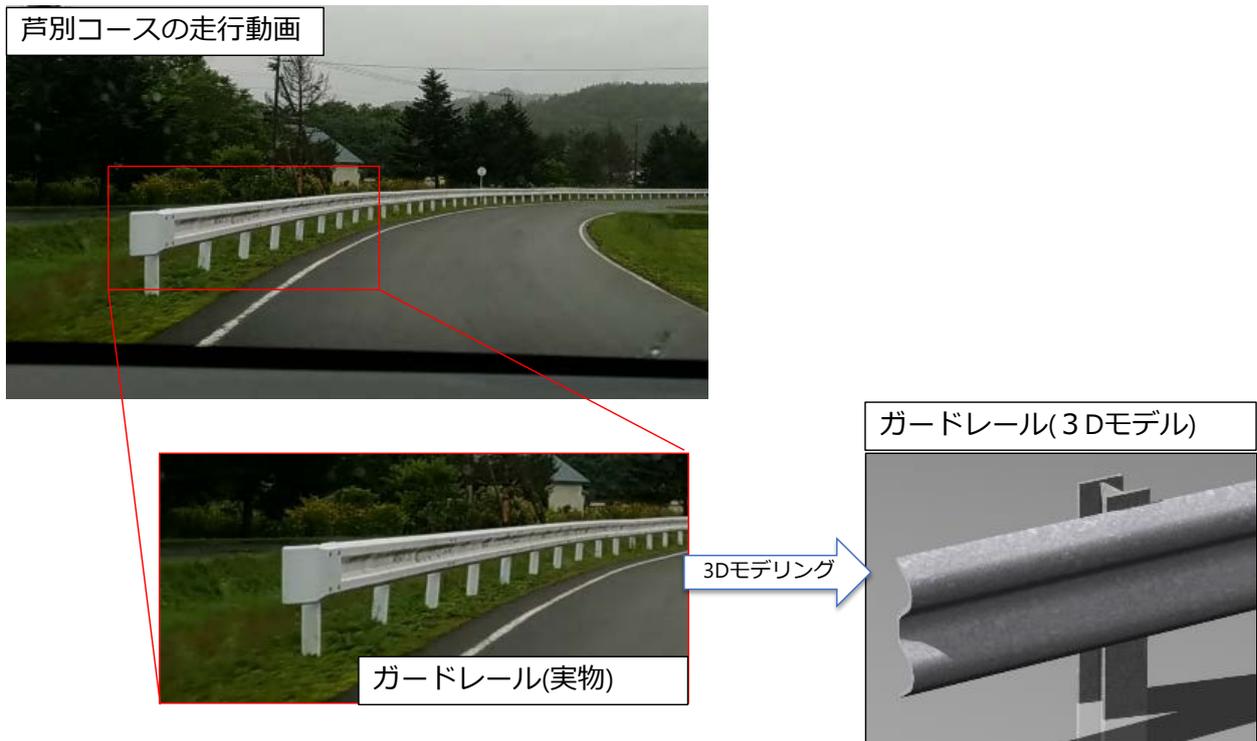


図 3-2-1(8) ガードレールの模擬イメージ図

2)実装 (芦別コースモデル)

検討した仕様を元にシミュレータに芦別コースの実装を行った。

シミュレータに実装を行った芦別コースおよびコース上の設置物の実装結果を以降に記載する。

また、設置物については、実コースをどのようにシミュレータで実装したかが分かるように、実装の基となった実コースの画像も併せて記載する。

実装を行った芦別コースの全体図として、シミュレータ上で俯瞰した絵を図 3-2-1(9)に示す。

コース上に、周回オーバルコースおよび多目的コースを模擬している。

周回オーバルコース

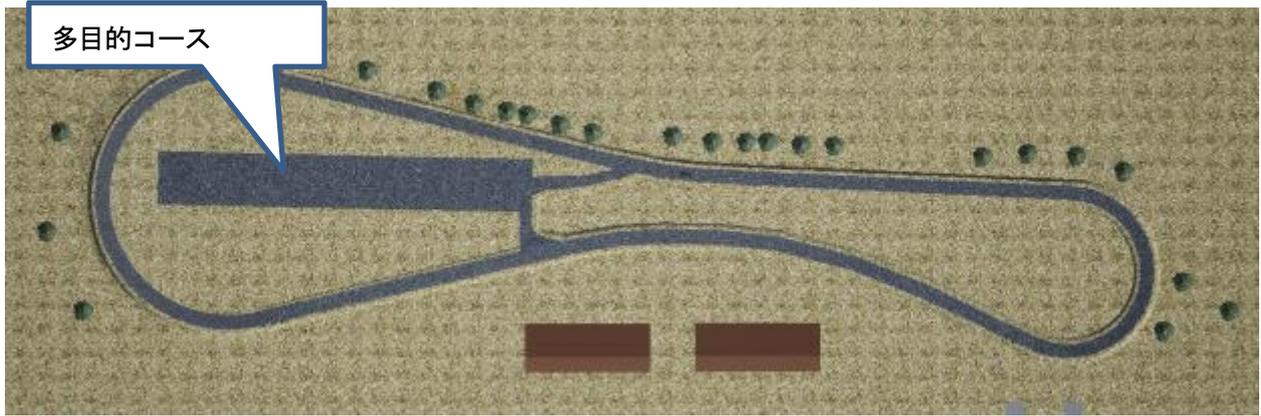


図 3-2-1(9) 芦別コースの全体図

芦別コースの周回オーバルコースの実装結果の一部を図 3-2-1(10)に示す。



図 3-2-1(10) 周回オーバルコース実装結果(左：実コース, 右：シミュレータ)

芦別コースの多目的コースの実装結果の一部を図 3-2-1(11)に示す。



図 3-2-1(11) 多目的コース実装結果(左：実コース, 右：シミュレータ)

コース脇の木々の実装結果の一部を図 3-2-1(12)に示す。



図 3-2-1(12) コース脇の木々実装結果(左：実コース, 右：シミュレータ)

道路標識の実装結果の一部を図 3-2-1(13)に示す。



図 3-2-1(13) 道路標識の実装結果(左：実コース, 右：シミュレータ)

建物の実装結果の一部を図 3-2-1(14)に示す。



図 3-2-1(14) 建物の実装結果(左：実コース, 右：シミュレータ)

電柱の実装結果の一部を図 3-2-1(15)に示す。



図 3-2-1(15) 電柱の実装結果(左：実コース, 右：シミュレータ)

ガードレールの実装結果の一部を図 3-2-1(16)に示す。



図 3-2-1(16) ガードレールの実装結果(左：実コース, 右：シミュレータ)

3)動作検証

動作検証として、2)実装（芦別コースモデル）で実装した各設置物に対して、LiDAR・ミリ波レーダ・赤外線サーモグラフィの3種類のセンサで、正しくセンシングできるかどうかを検証した。

センシングの判定基準は、実装済みの苫東コースを元に、それぞれ以下の観点で検証した。

表 3-2-1(7)センサ毎のセンシング判定基準

センサ	判定基準
LiDAR	検出対象に対し、LiDARが検出したことを示す点群が表示されていること
赤外線 サーモグラフィ	検出対象が、赤外線サーモグラフィ画像において、地面と比べて極端に異なる温度を出していないこと
ミリ波レーダ	検出対象に対し、ミリ波レーダで検出したことを示す赤枠が表示されていること

以下に、苫東コースでの設置物のセンシングの例を示す。



図 3-2-1(17) 苫東コースでのセンシング例①(左：LiDAR 右：サーモグラフィ)

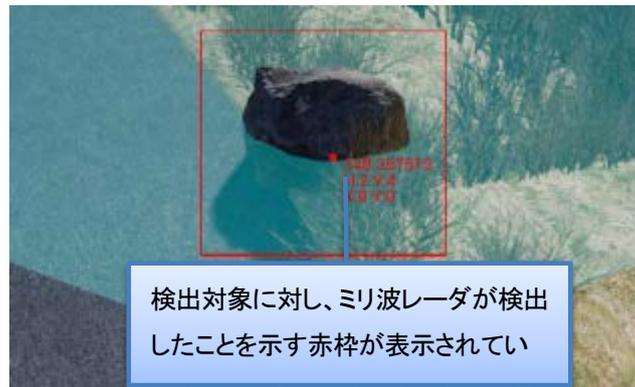


図 3-2-1(18) 苫東コースでのセンシング例②(ミリ波レーダ)

表 3-2-1(7)の判定基準で、センシングが正しくできたか検証した結果を以下に示す。

表 3-2-1(8)各設置物のセンシング検証結果

設置物	センシング		
	LiDAR	赤外線サーモグラフィ	ミリ波レーダ
木々	○	○	×
道路標識	○	○	○
建物	○	○	×
電柱	○	○	×
ガードレール	○	○	×

以降、各設置物の LiDAR・サーモグラフィ・ミリ波レーダでのセンシング結果を図で示す。

- ・木々

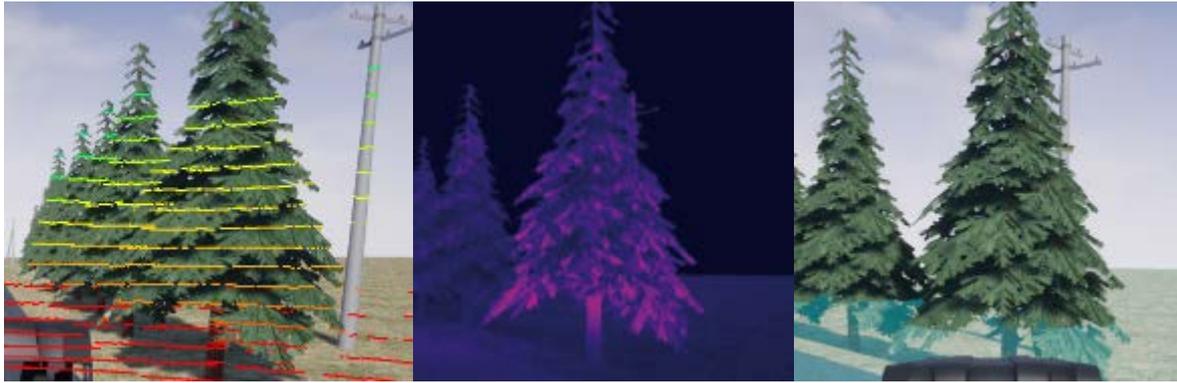


図 3-2-1(19) 木々のセンシング結果(左：LiDAR 中：サーモグラフィ 右：ミリ波レーダ)

・道路標識

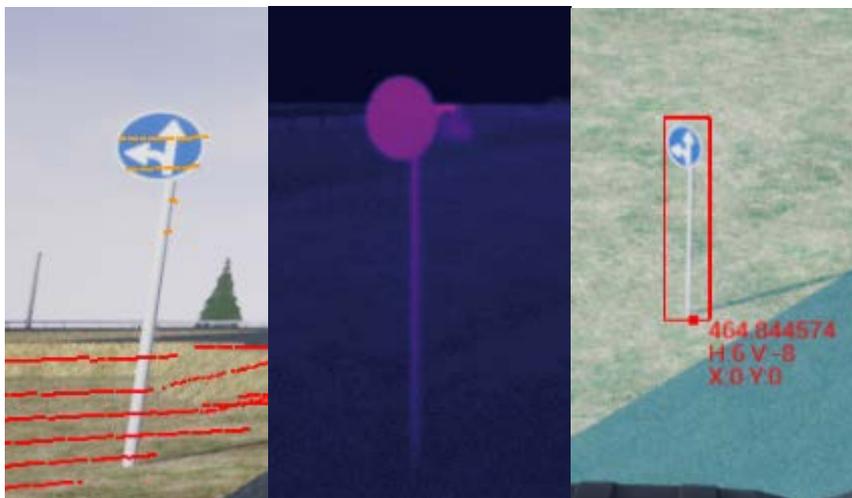


図 3-2-1(20) 道路標識のセンシング結果(左：LiDAR 中：サーモグラフィ 右：ミリ波レーダ)

・建物

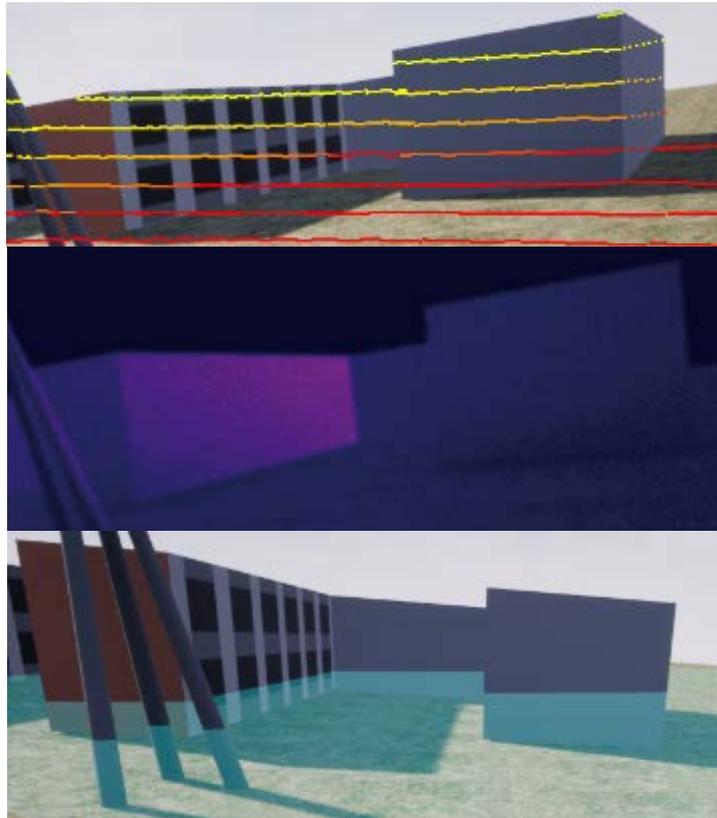


図 3-2-1(21) 建物のセンシング結果(左 : LiDAR 中 : サーモグラフィ 右 : ミリ波レーダ)

・電柱

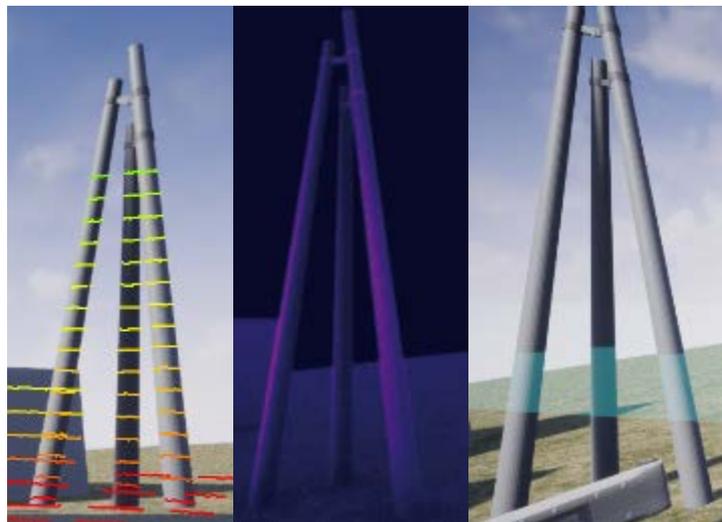


図 3-2-1(22) 電柱のセンシング結果(左 : LiDAR 中 : サーモグラフィ 右 : ミリ波レーダ)

・ ガードレール

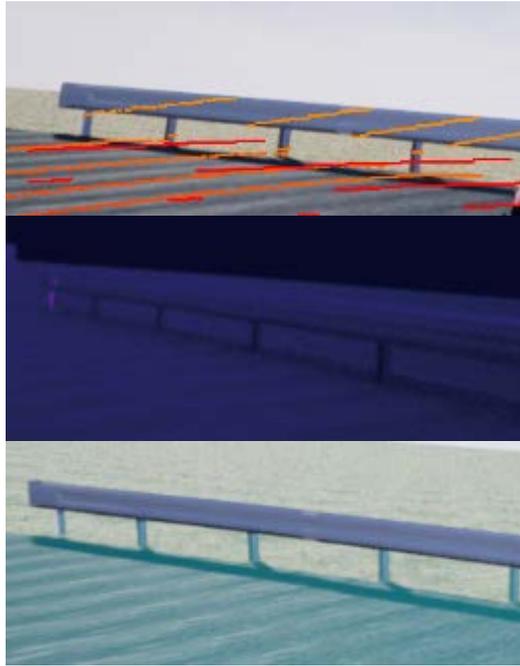


図 3-2-1(23) 電柱のセンシング結果(左 : LiDAR 中 : サーモグラフィ 右 : ミリ波レーダ)

2-2-7. シナリオ実行機能の追加

3年度は、積雪環境仮想シミュレータにシナリオ実行機能を追加する。

1)シナリオ実行の仕様検討

シミュレータを用いた制御ロジックの検証や検証時のデータ計測作業を効率的に行うことを目的に、シナリオ実行機能を追加する。シナリオ実行機能は、事前に設定したシナリオに応じて車両の走行状況（降雪状況や障害物の場所など）を変化させる機能を想定する。シミュレータに追加するシナリオ実行機能の仕様を検討する。シナリオ実行機能の検討結果を表 3-2-2(1)に示す。

表 2-2-7(1) 積雪環境におけるシナリオ実行機能に求める性能

項目		要求性能	実装判断	判断理由
対象コース		苦東コースと芦別コースに追加すること	○	-
変更条件		シナリオは対象コースを1周走行するごとに変更できること	○	-
設定	コース	ラウンドコース、苦東コース、ストレートコース、テストコースを変更できること	×	シナリオ実行時のコースは、選択している状態からのシミュレーションを想定する。よって本研究では対応しない。
	車両操作モード	手動入力、自動走行、UDP 入力、UDP 入力（目標値）	×	シナリオ実行時の走行モードは、ROSからの指示（UDP 入力）に応じて走行する使い方を想定する。よって本研究では対応しない。
	積雪量	シナリオ機能から変更できること	○	-
	降雪表現	シナリオ機能から変更できること	×	降雪条件の切り換えは、将来的には必要と考えるが優先度は低いと判断する。よって本研究では対応しない。
	雪壁	シナリオ機能から変更できること 設定は高さも変更できること	○	-
	障害物	シナリオ機能から変更できること コース上に複数個配置できること	○	-

2)シナリオ内容の検討

積雪環境仮想シミュレータに実装するシナリオの設定内容を検討する。

シナリオの内容は、変化させる条件が増えると膨大なシナリオ数になっていく。例えば、雪壁の高さを変化させる条件を考える場合、高さを5段階で変更可能かつ左右異なる条件を網羅しようとするると右壁：5段階×左壁：

5段階で計25パターンのシナリオが必要となる。さらに雪壁以外の条件を追加するとパターンは膨大な数に膨らみ、すべてをシミュレータに実装するのは現実的ではなくなる。

利便性を考慮して、シミュレータに実装するシナリオは必要最低限として、かつ利用者が任意に変更することが可能な仕様で実装することとする。

上記を踏まえて、シミュレータに実装するシナリオの条件には、積雪条件（積雪量、雪壁の状態）と制御検証への利用実績がある障害物の配置を含める。また、各条件は、最大、最小、中間となる値を選択する。実装するシナリオ内容を表 3-2-2(2)に示す。

表 2-2-7(2) 積雪環境におけるシナリオ内容

パターン	積雪量	雪壁	障害物
1	50cm	200cm / 200cm	6個
2	0cm	0cm / 0cm	4個
3	20cm	70cm / 70cm	0個

3)実装（シミュレータ側）

シミュレータ側のシナリオ実行機能は、コース上に当たり判定のない壁を追加し、車両がその壁を通過することでシナリオが更新される実装とした。通過時に発生するシナリオ更新イベント”LapEvent”の実装を図 3-2-2(1)に示す。

図中赤枠で囲った Sequence ブロックは、LapEvent で実施する処理を処理順にまとめたものである。Sequence ブロックには Then0 から Then4 の処理があり、Then 0 から順番に処理が実施されることを表現している。Then 0～4 では、下記の処理を実施している。

Then 0 : 前回のシナリオ更新イベントで生成された障害物を削除

Then 1 : 今回のシナリオ用の障害物を生成

Then 2 : 積雪量と雪壁の設定を更新

Then 3 : シナリオ周回数をインクリメント

Then 4 : ROS トピック保存フラグ更新(ROS 通信用の処理)

障害物の数と位置、積雪量、雪壁の高さについては、あらかじめ設定用のファイルを作成しておき、それをインポートすることで設定する。

車両がコースを1週して壁を通過するたび、上記のシナリオ更新イベントを呼び出すことで、シナリオ実行機能を実装している。

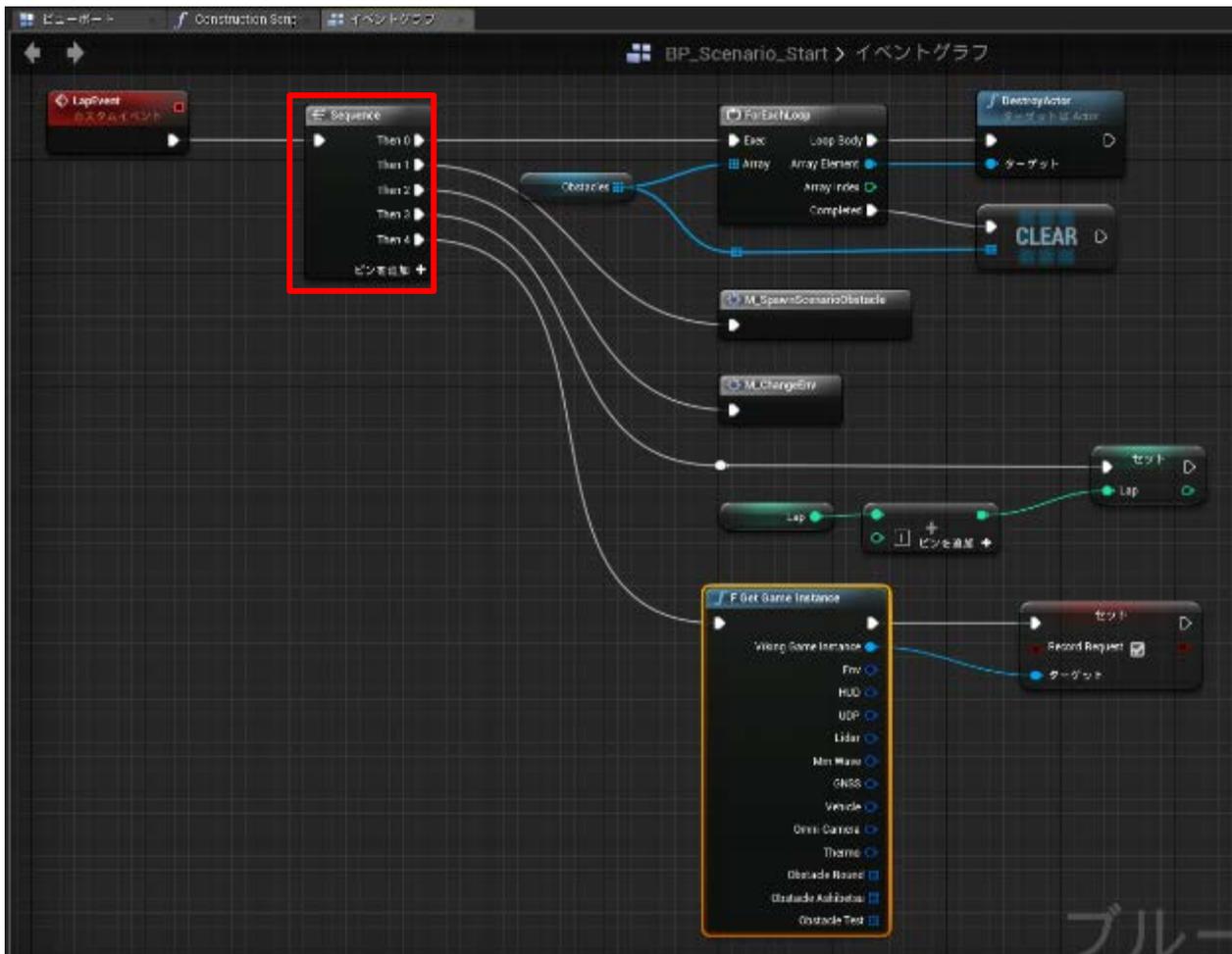


図 2-2-7(1) シナリオ更新イベント

4)実装 (ROS 側)

3)実装 (シミュレータ側) で実装された、ROS トピック保存フラグの送信に応じて、ROS トピック保存機能 rosbag を用いて、シナリオ実行中の ROS トピックを保存する処理を実装した。

5)動作検証

1)シナリオ実行の仕様検討で定義した仕様を満たしているか、シナリオ更新時の挙動を確認した。

表 3-2-2(1)で定義された仕様は以下の5項目になる。

- ・ 芦別コースと苫東コースに機能を追加すること
- ・ シナリオは対象コースを1周走行するごとに変更できること
- ・ 積雪量をシナリオ機能から変更できること
- ・ 雪壁の表示・非表示、高さをシナリオ機能から変更できること
- ・ 障害物の配置をシナリオ機能から変更できること

これらを確認するため、シナリオ更新時、表 3-2-2(2)のシナリオ内容通りに、環境が変化するかどうかを確認した。起動時(任意の環境)からパターン3までの遷移を以下に示す。

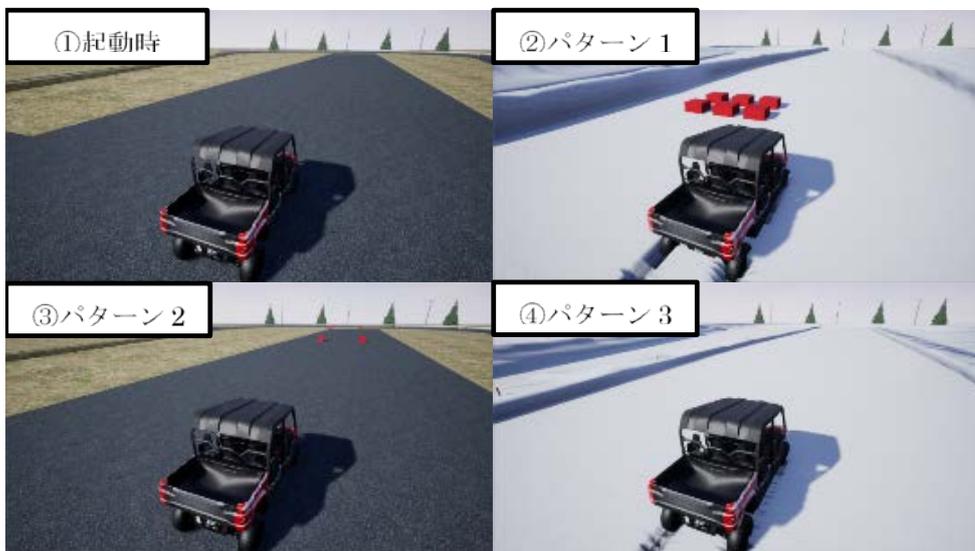


図 2-2-7(2) シナリオ周回時の環境の変化(芦別コース)

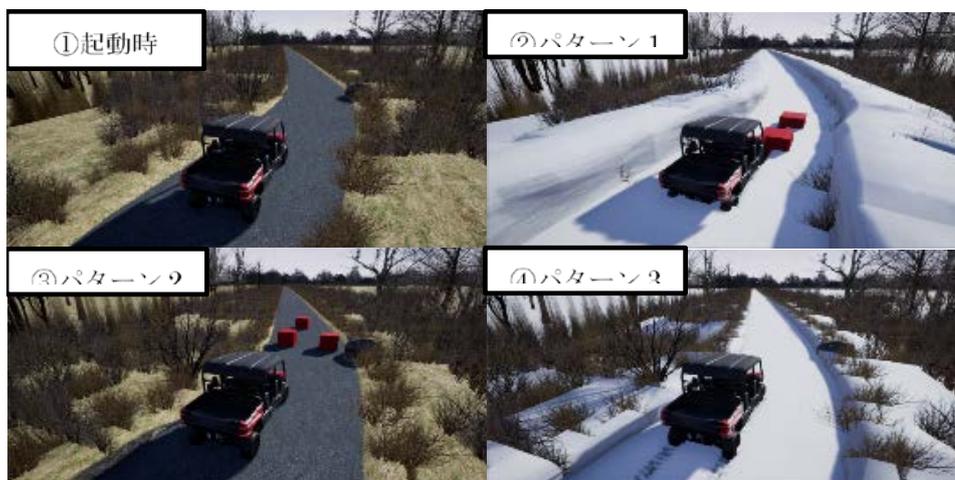


図 2-2-7(3) シナリオ周回時の環境の変化(苫東コース)

2-2-8. シミュレータ改善

3年度は、積雪環境仮想シミュレータに下記改善を実施する。改善項目を表 3-2-3(1)に示す。

本項では、「カメラ」は「車載カメラ」を示し、「全方位マルチカメラ」は「全方位マルチカメラ(Ladybug5+)」を示すものとする。

表 2-2-8(1) シミュレータ改善項目

項目	説明
雪壁	除雪した際に道路両端にできる雪壁を再現する。仕様を検討して実装する。
カメラセンサ改善	赤外線サーモグラフィとカメラを同時に出力する方法を検討する。

2-2-8-1. 雪壁

1)雪壁仕様検討

積雪環境の除雪時に道路両端にできる雪壁を模擬する機能を検討する。

積雪状況や除雪状況に応じて雪壁の形状が変わることを想定して検討した結果を表 3-2-3-1(1)に示す。

表 2-2-8-1(1) 積雪環境における雪壁の模擬に求める性能

項目	要求性能	実装判断	判断理由
表示	雪壁の表示、非表示を切り替えられること	○	-
高さ	変更	○	-
	自由度	○	-
よごれ	土などによる雪壁のよごれを表現できること	×	雪壁のよごれ表現は、将来的には必要と考えるが優先度は低いと考える。よって本研究では実装しない。
崩れ	融雪等による雪壁の崩れた状態を表現できること	×	雪壁が崩れた状態の表現は、将来的には必要と考えるが優先度は低いと考える。よって本研究では実装しない。
道路境界表示	雪壁表示時はスノーポールを表現できること 実装検討対象は芦別コースとする	○	-
検出	雪壁と車両が接触する場合、透過せず	○	-

項目		要求性能	実装判断	判断理由
		に衝突すること		
セン シ ン グ	LiDAR	雪壁がLiDARで検出されること	○	-
	ミリ波レーダ	雪壁がミリ波レーダにて検出されること	×	ミリ波レーダも同様に本研究での利用優先度が低いと判断して実装しない
	全方位マルチカメラ	雪壁が全方位マルチカメラに映ること	○	-
	赤外線サーモグラフィ	雪壁が赤外線サーモグラフィに映ること	○	-
対象コース		苫東コースと芦別コースに追加すること	○	-

3)雪壁動作検証

1) 雪壁仕様検討で定義した仕様を満たしているか、設定を変更し、シミュレータの動作を確認した。

以下に、定義した仕様毎に、動作確認結果を示す。各仕様の動作確認結果は、コースによる差は発生しないため、芦別コースの結果のみを示す。ただし、「雪壁は苫東コースと芦別コースに追加すること」の動作確認については、各コースに雪壁が追加されたことを確認するため、苫東コース・芦別コースの両方のコースでの動作結果を示す。

- ・雪壁は苫東コースと芦別コースに追加すること

苫東コース・芦別コースでの動作結果を図 3-2-3-1(3)に示す。



図 2-2-8-1(3) 左：苫東コース 右：芦別コース

- ・雪壁は左右の壁が同じ高さではなく、それぞれを任意の高さに設定可能であること

左右に異なる高さ(左側 70cm・右側 200cm)を設定した場合の動作結果を図 3-2-3-1(4)に示す。



図 2-2-8-1(4) 左側 70cm・右側 200cm

- ・雪壁の高さを変更できること。ただし、本研究では 20cm、40cm、70cm、150cm、200cm の 5 段階とする
- ・雪壁の表示、非表示を切り替えられること

各段階の高さ(非表示・20cm・40cm・70cm・150cm・200cm)を設定した場合の動作結果を図 3-2-3-1(5)に示す。



図 2-2-8-1(5) 左上から、非表示・20cm・40cm・70cm・150cm・200cm

- ・雪壁表示時はスノーポールを表現できること。実装検討対象は芦別コースとする
- スノーポールが表示されている場合(雪壁 40cm)の動作結果を図 3-2-3-1(6)に示す。

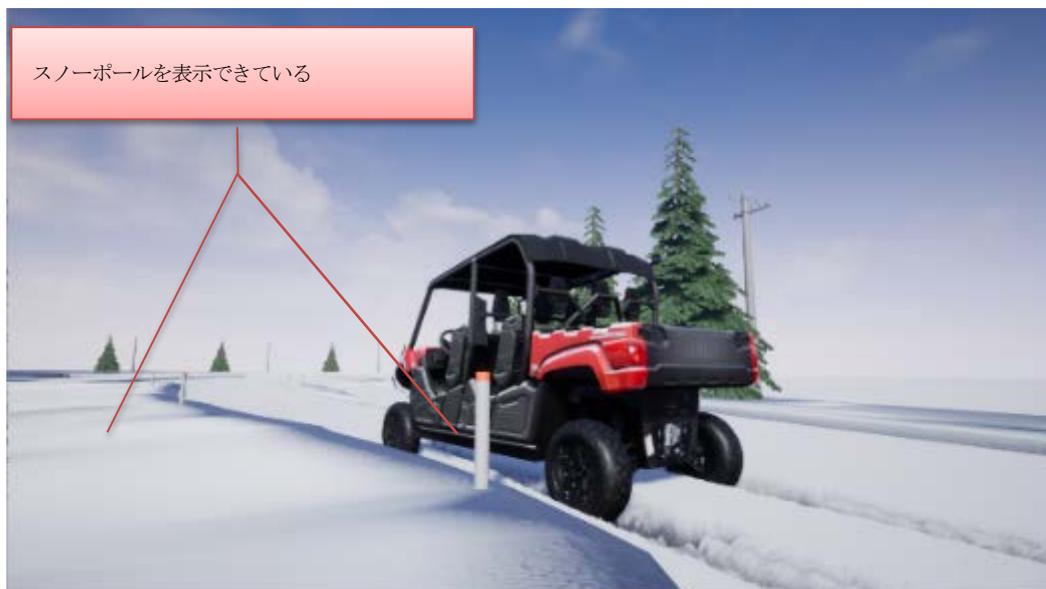


図 2-2-8-1(6) 雪壁 40cm、スノーポール表示

- ・雪壁と車両が接触する場合、透過せずに衝突すること
雪壁と衝突した場合(雪壁 40cm)の動作結果を図 3-2-3-1(7)に示す。



図 2-2-8-1(7) 雪壁 40cm、雪壁衝突時

- ・雪壁が LiDAR で検出されること
 - ・雪壁が全方位マルチカメラに映ること
 - ・雪壁が赤外線サーモグラフィに映ること
- LiDAR での動作結果を図 3-2-3-1(8)、
全方位マルチカメラ、赤外線サーモグラフィでの動作結果を 3-2-3-1(9)に示す。



図 2-2-8-1(8) LiDAR 動作結果



図 2-2-8-1(9) 左：全方位マルチカメラ 右：赤外線サーモグラフィ

2-2-8-2. カメラセンサ改善策検討

1) カメラセンサ改善の仕様

積雪環境仮想シミュレータには、カメラ（図 3-2-3-2(1)）と赤外線サーモグラフィ（図 3-2-3-2(2)）の画像を出力する機能を昨年度までに開発済みである。カメラセンサは選択式としており、どちらか一方を選択して使用する。

しかし近年、複数カメラセンサから得られたデータを活用したセンサ・フュージョンの検証に対してシミュレータを活用するニーズが出てきた。そのため、カメラセンサ改善策として、複数カメラセンサを同時にシミュレートすることを検討した。センサ・フュージョンの検証にシミュレータを活用できるようにするため、種類が異なる2つ以上のカメラを同時に出力する機能及びシステム構成を検討する。



図 2-2-8-2(1) カメラのイメージ



図 2-2-8-2(2) 赤外線サーモグラフィのイメージ

① カメラ画像の同時生成

昨年度の積雪環境仮想シミュレータのカメラセンサの画像は、選択中のカメラごとにワールド設定（例：重力、光源）の設定値を変更し、取得されたカメラ画像に各カメラセンサ用の変換処理を施すことで生成している。

カメラセンサの切り替え時に設定するワールド設定には UE4 内部の光源となる太陽光の調整が含まれている。太陽光は、カメラ選択時は“明るく”、赤外線サーモグラフィ選択時は“暗く”する設定をしている。赤外線サーモグラフィ選択時の太陽光設定を変更したイメージを図 3-2-3-2(3)に示す。カメラセンサの種類ごとに相反するワールド設定が必要なため、1つのシミュレータで2つのカメラセンサ画像を同時に生成することができない。

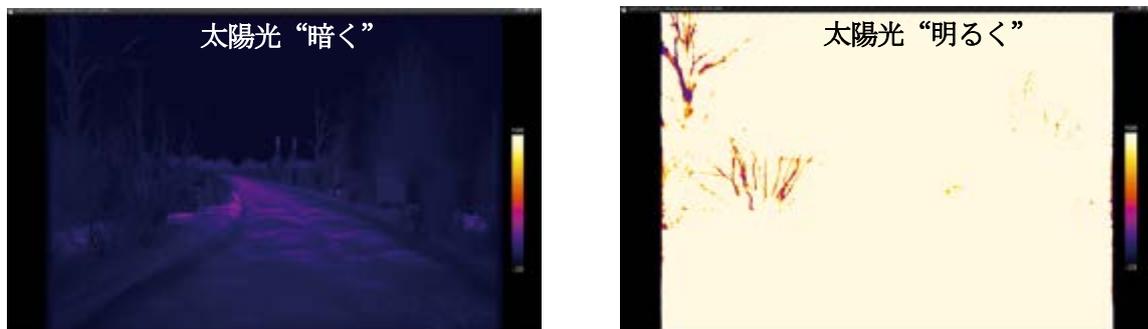


図 2-2-8-2(3) 赤外線サーモグラフィの光源変更イメージ

種類が異なるカメラ画像を1つのシミュレータで模擬できないため、共存できないカメラの種類ごとにシミュレータを起動する方法を考えた。この方法により、複数カメラセンサ画像の同時生成が可能となる。本研究では、シミュレータ同士を同期させることで同一のシチュエーション時の画像を生成する方式を採用する。

② 動作環境

「カメラ画像の同時生成」で記載した方式を実現するためのシミュレータ動作環境を検討する。検討結果を表 3-2-3-2(1)に記載する。

ROS と接続する場合は、LAN 経由で Topic 形式のデータを送信する構成 3 が最も使い勝手が良い構成である。ただし、Topic 以外のデータ形式で受け渡しする場合は、構成 2 の方が汎用的に利用可能である。

表 2-2-8-2(1) 動作環境検討結果

◎：優位性あり ○：問題なし △：劣る ×：問題あり

	構成1 (PC×1台、LAN出力)	構成2 (PC×2台、HDMI出力)	構成3 (PC×2台、LAN出力)
導入費用	◎	△	○
シミュレータの 安定性	×	○	○
セットアップの 容易性	◎	△	○
総合評価	×	△	○

① 構成1 (PC×1台、LAN出力)

シミュレータをPC1台で動作させる構成である。システムの構成案を図 3-2-3-2(4)に記載する。

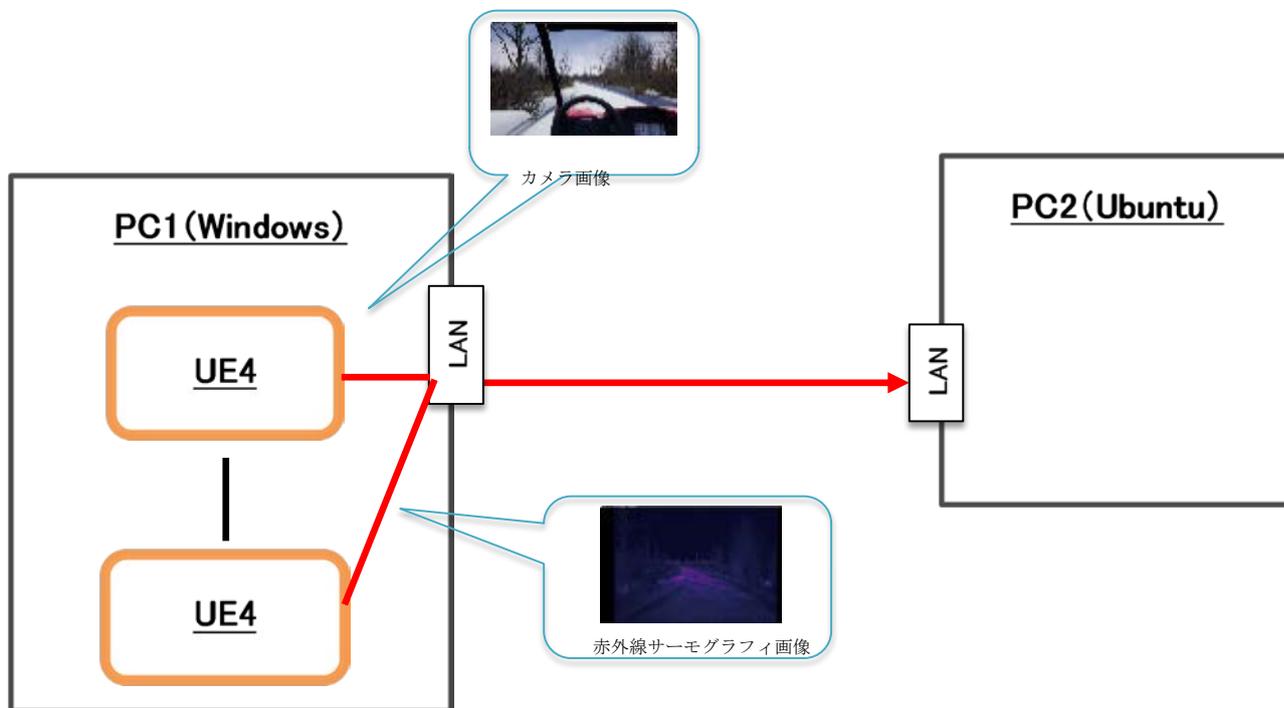


図 2-2-8-2(4) 構成1のシミュレータ構成案のイメージ

表 2-2-8-2(2) 構成1の接続機器一覧

項目	仕様	備考
PC1	Windows OS	シミュレータ実行用のPC。グラフィックボード
PC2	Ubuntu OS	ROS 動作用のPC。シミュレータとの接続にLAN、USBポートが必要。

・メリット

PC同士をLANで接続するのみのため、セットアップにかかる時間が最も少ない。

・デメリット

1台のPC上で複数のシミュレータを動作させるため、PCにかかる負荷が大きくなる。その結果、シミュレーション中のFPSが低下してコマ送りのような動作になる懸念がある。そのため、Ubuntu側の制御に必要なFPSを達成できずに制御検討に使用できないことが考えられる。また、PCのスペックによってはシミュレータ自体を動作させられないことも考えられる。

構成2 (PC×2台、HDMI 出力)

シミュレータを2台のPCで動作させる環境構成である。PCのI/FとしてHDMIを利用するシステム構成案を図3-2-3-2(5)に記載する。

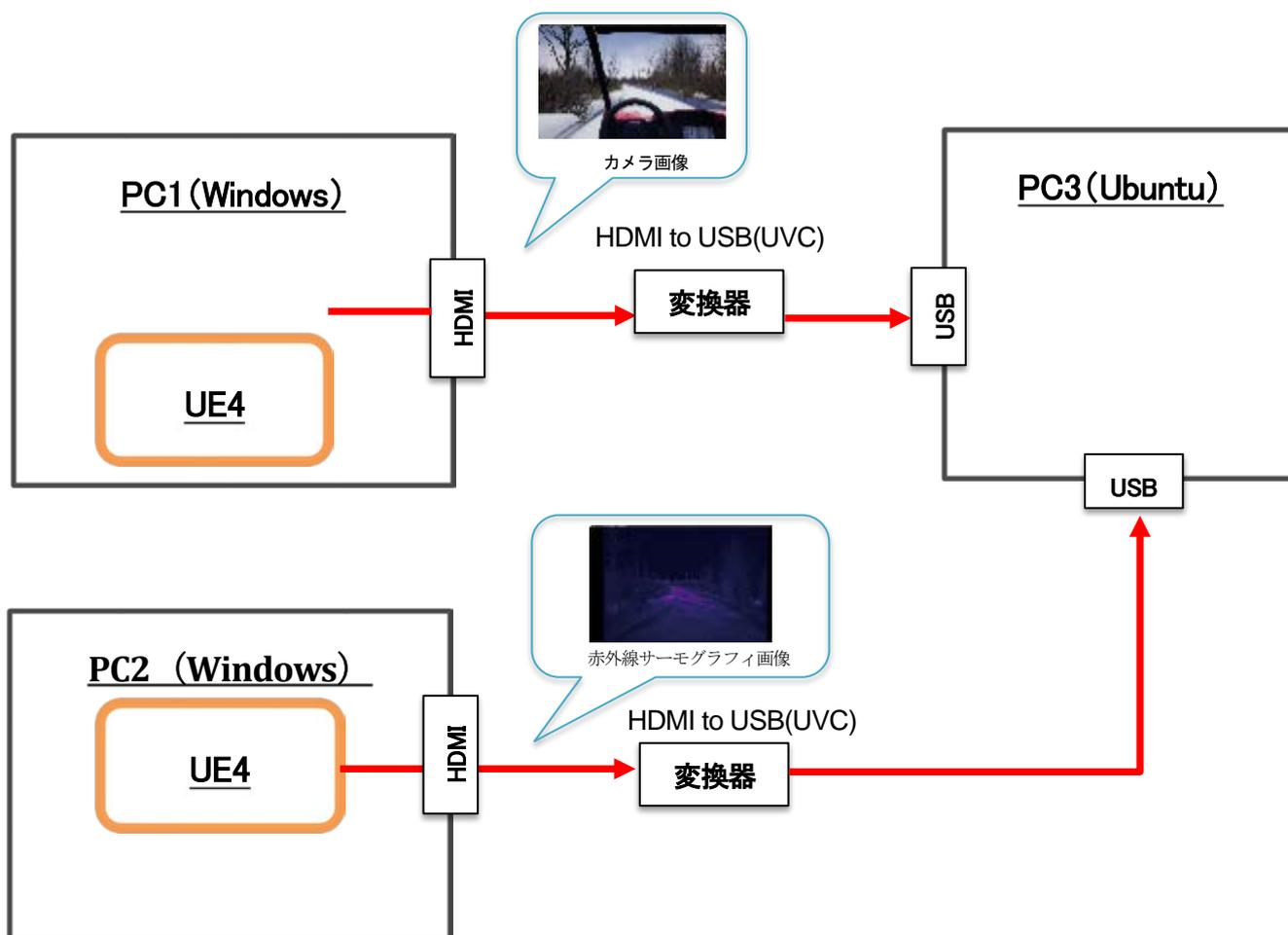


図 2-2-8-2(5) 構成2のシミュレータ構成案のイメージ

表 2-2-8-2(3) 構成2の接続機器一覧

項目	仕様	備考
PC1 / PC2	Windows OS	シミュレータ実行用のPC。グラフィックボード
PC3	Ubuntu OS	ROS 動作用のPC。シミュレータとの接続にLAN、USBポートが必要。
変換器	入力:HDMI 映像出力:USB UVC規格。USB 3.0対応(※1)	HDMIの出力画像をUSB Web CameraとしてPCに取り込む

※1 USB2.0でも対応可能だがキャプチャー画面の画質、フレームレートが低下する

- ・メリット
シミュレータの実行環境を分散させるため、構成1よりも安定した動作ができる。
- ・デメリット

USBカメラ規格のUVC（USB Video Class）形式への変換が必要である。HDMIからUSBに変換するハードウェアを用いることで対応が可能。1万円/台程度の比較的安価に購入することが可能であるが費用が発生する。また、PC3のUSBポートが不足する場合は別途USBハブの購入が必要となる。環境セットアップ時に変換機やハブの接続が必要になるため、他の構成案と比べて利便性が劣る。

② 構成3（PC×2台、LAN出力）

シミュレータを2台のPCで動作させる環境構成である。PCのI/FとしてLANを利用するシステム構成案を図3-2-3-1(6)に記載する。

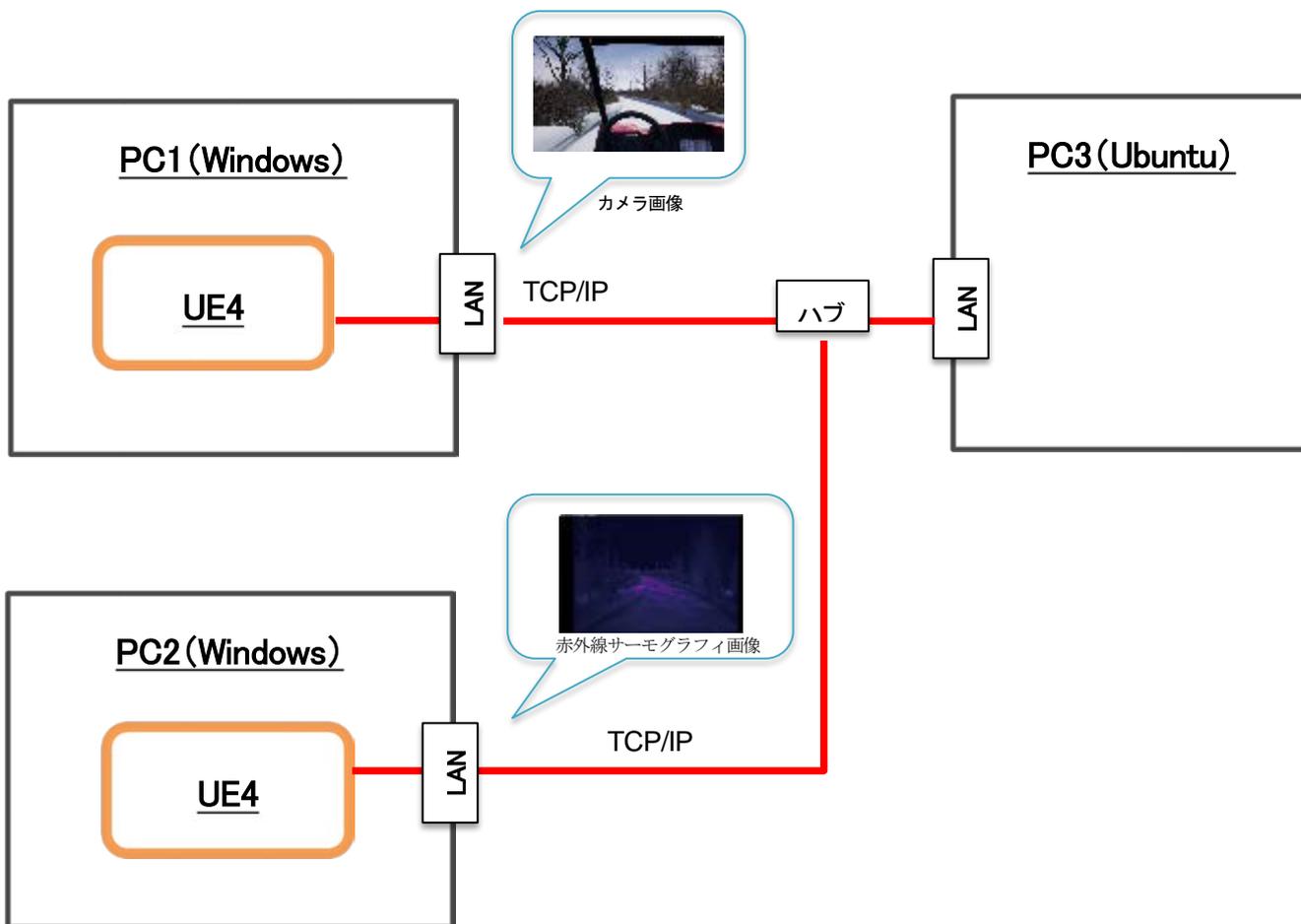


図 2-2-8-2(6) 構成3のシミュレータ構成案のイメージ

表 2-2-8-2(4) 構成3の接続機器一覧

項目	仕様	備考
PC1 / PC2	Windows OS	シミュレータ実行用のPC。グラフィックボード
PC3	Ubuntu OS	ROS 動作用のPC。シミュレータとの接続にLAN、USBポートが必要。
ハブ	-	イーサネットを複数接続するためのハブ。

・メリット

シミュレータの実行環境を分散させるため、構成1よりも安定した動作ができる。
接続するケーブルがLANのみとなり、シミュレータの準備工数が構成2よりも少なくなすむ。

- ・デメリット
特になし。

2-2-9. 成果

初年度成果:

サブテーマ2の初年度の成果として、積雪環境仮想シミュレータを構築することができた。



図 2-2-9. 積雪環境仮想シミュレータ

積雪環境仮想シミュレータは、Windows OS 上で動作するシミュレータ本体部分(図 2-2-9)と、ROS(Linux OS 上)で動作する通信部分とで構成する。OS 間は UDP で通信を行う為、2つのPCをクロス LAN ケーブルで繋ぐ方法や、同一PCの仮想マシン上でそれぞれを動作、及び通信させる等の使い方が可能である。

尚、作成したテストコースは下記の3種類であり、自由に選択が可能。

- ①実証実験コース(株 苫東、周回路)
- ②実証実験コース(株 苫東、直線路)
- ③仮想コース(広め、自由に走行可能、複数の障害物あり)

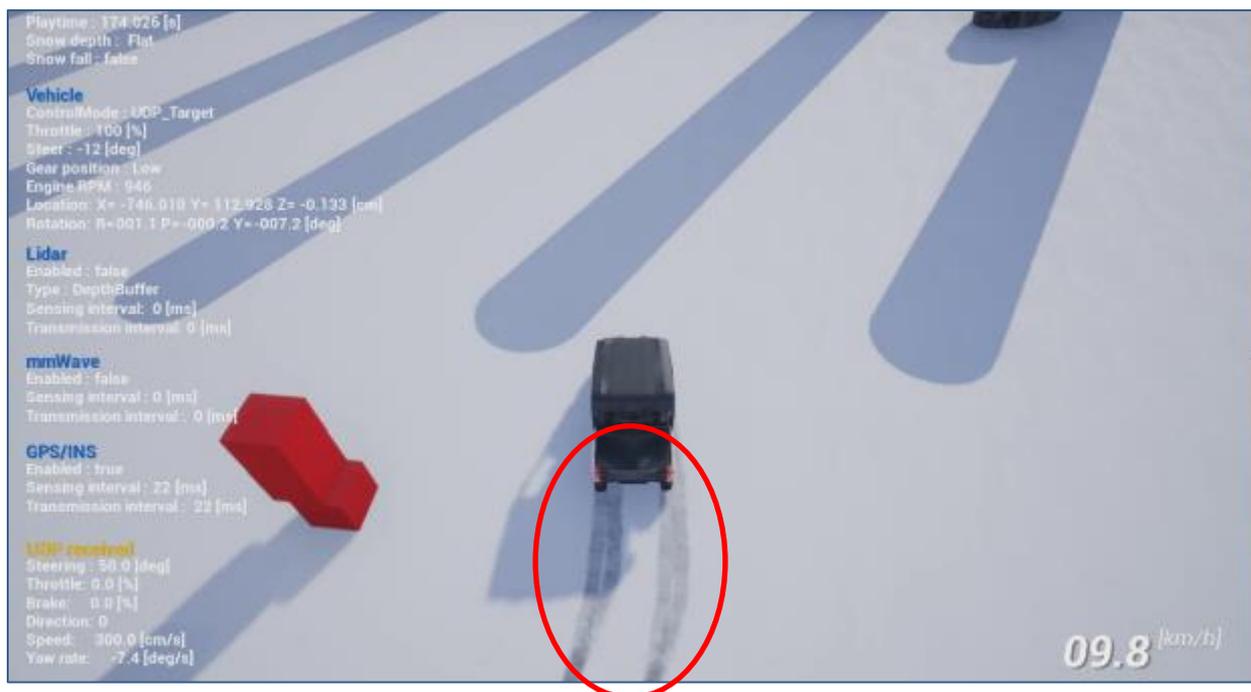
また、路面状態は雪、砂利、土、アスファルト、草から選択可能であり、積雪量は 0cm、20cm、30cm、50cm の中から選択可能とした。他にも、各センサの設定も一定の範囲内で調整が可能である。以下に、各設定画面と設定値を示す。

今回試作した積雪環境仮想シミュレータを用いることにより、サブテーマ1、サブテーマ3で開発したアプリの動作検証をシミュレータ上で行うことが可能になった。評価可能な範囲は、現状では実車評価を行う前のロジックの確からしさ等、定性的な部分となる。また、SLAM やセマンティック・セグメンテーションの目的である、積雪環境で機能する自己位置推定を実現するために、様々なセンサでデータ収集を行い、性能目標を達成するため検討が進められている。初年度は追加されたセンサのモデルを作成し、シミュレーション環境に対して組み込む作業を実施した。その結果、実証実験を行わずとも、実車に近い状態でのシミュレーションを安全に行うことを可能となった。

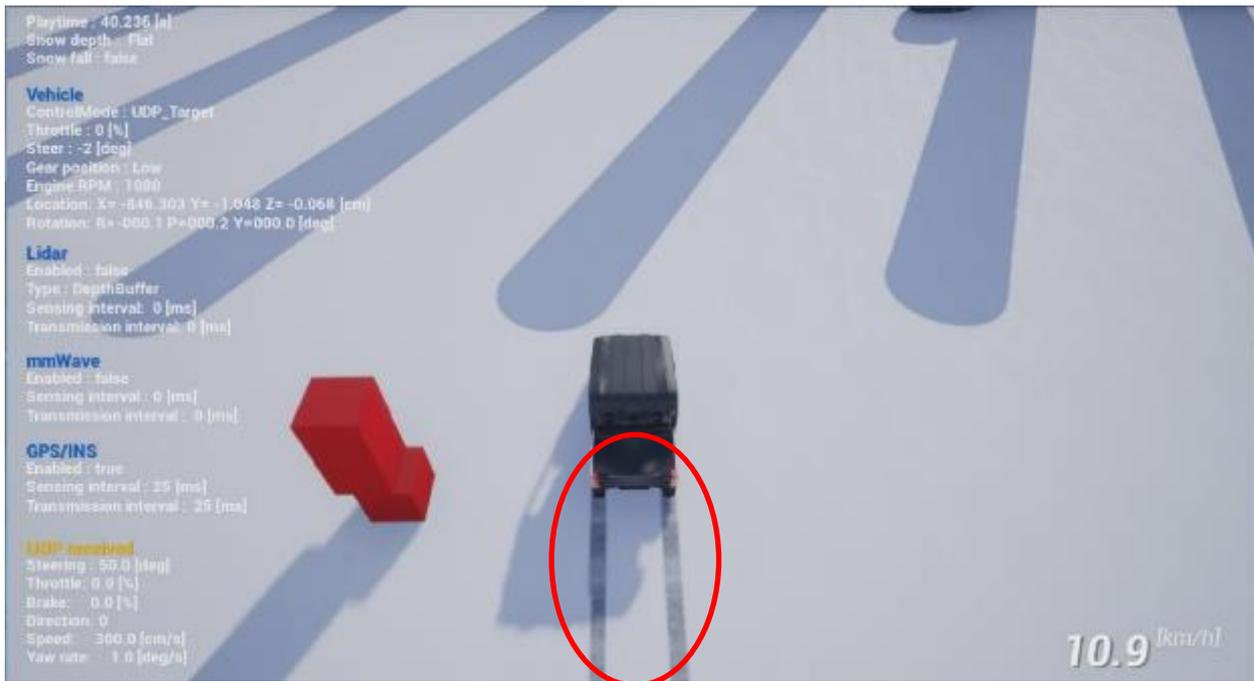
2年度成果

サブテーマ2にて開発した積雪環境仮想シミュレータを、サブテーマ1にて実際に使用した。主な用途は実車で発生した異常動作の問題解決であり、実証実験時に起きた不可解な動作をシミュレータ上で再現させ、原因の特定、問題の解決、修正されたことの確認までをシミュレータ上にて行った。これにより、限られた実証実験の時間を有効に活用する事に貢献できた。

以下に、実際にシミュレータを用いて問題解決を行った一例を示す。実車にて、雪道を自動走行させた際に意図しない蛇行が発生する現象が発生した。シミュレータにて同様に自動走行させたところ、実車同様に蛇行する現象を確認できた。原因を調査したところ、パラメータの設定値に問題がある事が分り、それを修正することにより蛇行しなくなる事までを、シミュレータ上で確認することができた。



パラメータ修正前の蛇行運転結果



パラメータ修正後、蛇行しなくなった結果

3年度成果

3年度は以下の研究を実施した。

- 芦別コースの追加
- シナリオ実行機能の追加
- シミュレータ改善

3年度は実証実験に近い環境をシミュレーションで実施するための機能改良を実施した。1つめは、実証実験のコースとなる芦別コースの追加である。2つめは、積雪環境の再現範囲を拡張するため、雪壁を再現する機能を追加した。また、シミュレータの利便性をあげるためシナリオ実行機能を追加する作業を実施した。その結果、実車を用いた実証実験により近い状態でのシミュレーションを行うことが可能となった。

2-3. 【サブテーマ3】グローバル経路計画との連携技術開発

本取組では、グローバル経路計画連との連携する標準的なインターフェースを搭載したシステムを開発する。また、Snow-SLAM方式の自動運転制御システムとの協調動作を行い、移動支援アプリを用いた積雪環境での交通支援サービスを実現することを目標とした。

2-3-1. 初年度の取り組み

2-3-1-1. 初年度の目標

グローバル経路計画と連携するための標準的なインターフェース機能を搭載したシステムの試作開発を行う。

(1) 課題

ナビゲーションとの連携は既存のシステムやアプリをそのまま使えることが可能なシステムが必要。

本研究事業で使用する自動運転制御システムはナビゲーションシステムと連携する機能に対応していない。

(2) 課題への対策

サブテーマ3では、従来のカーナビや地図サービス等（グローバル経路計画機能）が自動運転制御システム（ローカル経路計画機能）と連携することを可能にするためのシステム（グローバル経路計画連携機能）を開発する。多様なシステムやアプリからの利用を想定して、ナビゲーションシステムに必要となる機能を標準インターフェースとして提供する。

また、ローカル経路計画機能へ指定する走行ルートに対して、waypoint と呼ばれる途中通過点を外部から設定するためのオープンなAPIを設計して公開する。waypointの指定方法は、地図サービス上から設定可能なインターフェースを開発する。

2-3-1-2. 具体的な内容

2年度の活動概要は以下の通り

(1) グローバル経路計画連携機能のインターフェース仕様の策定

グローバル経路計画連携機能に必要な経路計画のシステム、ローカル経路計画機能との連携方法、waypointの設定方法の調査を行い、標準的なインターフェース仕様の検討と策定を行う。

(2) グローバル経路計画連携機能の開発

(1) インターフェース仕様の策定により、利用するグローバル経路計画機能のシステム構成、ローカル計画機能との連携実現のためのシステムを設計し、グローバル経路計画連携機能の試作を開発・実装する。

(3) 実証実験

私道である実証実験場はグローバル経路計画機能を動作させるためには、実験に利用可能な地図情報が必要となる。そのため、実証実験に使用する道路の地点情報を、GPS機器を用いて収集し、収集したデータから地図データを作成する。

2-3-1-2-2(1)グローバル経路計画連携機能の仕様策定

waypoint と呼ばれる途中通過点を設定可能とするためのシステム調査、及びサブテーマ1の主機能であるROS

との連携方法についての調査を行った。

① グローバル経路計画システムの調査

waypoint を生成するために、オープンソースのシステムや、スマートフォンで動作するナビゲーションアプリ等を調査した。調査対象のシステム候補として、ナビゲーションアプリ、web アプリケーション、Linux 上で動作するシステムの3つに分けることができる。以下に各特長を記載する。

1. ナビゲーションアプリ

例として代表的なものに MapFan、NAVITIME 等がある。

MapFan は Autoware と呼ばれる自動運転システムの waypoint 生成方法の一つで利用された事例があるため、MapFan を導入できるか調査した。

MapFan を利用する場合、既存のナビゲーションアプリから waypoint の生成が可能と考えられたが、Lidar 等のセンサ情報から独自のベクターマップを生成する必要があり、また、MapFan の仕様変更により、Autoware での実用事例とは異なる手法が必要であり、本研究には適用できないと判断した。

NAVITIME やその他のナビゲーションアプリは、waypoint 生成に利用可能な地点情報がアプリケーションから取得できない制約があり、自動運転用の走行ルートそのものを作成することが困難になるため、グローバル経路計画連携機能として利用できないと判断した。

2. Web アプリ

例として、GoogleMap、Yahoo!Map 等がある。

GoogleMap は GoogleCar を用いた地図の作成や、Google 社の独自技術による自動運転の研究などでも利用されていた事例がある。

GoogleMap を利用する場合、Google Maps Directions API を使用することで、経路の検索、waypoint の自動生成に利用可能な経度・緯度の地点情報群が取得できることが分かった。しかし、GoogleMap の API の無償利用は回数制限があり、オンラインでの利用を前提としているため利用状況に制限がかかってしまうことが分かった。また、商用製品として利用する場合は、有償対応が必要となり、試作開発での利用には向いていないと判断した。

3. Linux 上で動作するシステム

Linux 上で動作するシステムとして、地図情報を OSM(OpenStreetMap)、経路検索 API 部分を OSRM、graphhopper とするシステムが考えられる。OSM を利用することで、オフラインでの地図閲覧と、オフラインの地図情報を元とした経路検索(OSRM、graphhopper)が可能である。また、OSRM、graphhopper で得られる経路情報のデータから waypoint の自動生成に利用可能な経度緯度の地点情報群の取得も可能である。OSM はオープンソースとして提供されているため、API の利用回数制限などの制約もなく、地図タイルサーバーを Linux 上に構築することで、地図データのカスタマイズが可能となる。

OSM の経路探索の API は複数用意されており、OSRM と graphhopper ではほぼ同じ機能を有している。しかし、同じ道路に対して OSM の地図情報から取得する地点情報群の取得数に違いがあることが判明した。OSRM は、一部省略された地点情報群となっており、graphhopper は、OSRM より多くの地点情報群が取得できる。そのため、グローバル経路計画に適しているのは地図に対して詳細な waypoint の設定が可能な graphhopper となる。

上記の調査結果から、ナビゲーションアプリは waypoint 生成に利用できるデータの取得ができず、web アプリは利用制限があるため、waypoint を自動生成するために利用するシステムは、利用制限がなく、waypoint 生成に利用可能な地点情報群が取得できる OSM と graphhopper を利用することとした。

システム調査の詳細は『2-2-2-1_グローバル経路計画システム調査 1.pptx』『2-2-2-1_グローバル経路計画システム調査 2.pptx』を参照。

② グローバル経路計画連携方法の検討

グローバル経路計画機能で設定した waypoint をローカル経路計画機能に渡す連携機能が必要となる。サブテーマ 1 の自動運転システムには ROS が利用されており、ローカル経路計画とグローバル経路計画を連携させるためには、ROS と連携させる必要がある。ROS との連携は 4 つの方法があり、以下に各特長を記載する。

1. ROS 同士での連携

グローバル経路計画側に ROS 環境を構築することで、ローカル経路計画側の ROS と、ROS 同士で連携させることを想定した。ROS 同士の連携は、連携する端末が同じネットワーク上にあり、ROS の環境変数「ROS_MASTER_URI」に IP アドレスと port 番号を設定することで別端末同士の通信は容易に行うことができる。ただし、グローバル経路計画機能側に ROS 環境の構築が必要となり、非 ROS 環境による連携はできない。これは、従来のナビゲーションシステムに対して手を加える必要があり、連携機能としての要求を満たさないため、採用しないこととした。

2. rosjava と Android NDK による連携

rosjava は、Java によって ROS 上のシステムとの通信を実装することができ、Android のアプリケーションでも利用することができる。ただし、コンパイラの構築が複雑であり、ROS の最新バージョン (kinetic) には対応しておらず、アクティブなライブラリではないため、本事業では利用できない。

3. ファイルの書き出しによる連携

単純にグローバル経路計画側で waypoint 群を書き出したファイルを、何らかの方法でローカル計画側に渡し、読み込ませる方法がある。ローカル経路計画とグローバル経路計画の連携リアルタイム性が必要な処理、例えばローカル経路計画側の GNSS 情報と車両ステータス情報を一定周期でグローバル経路計画側に送ることを想定するとファイルによる連携は不向きになる。

4. rosbridge と roslibjs による連携

rosbridge は、JSON メッセージで WebSoket を通して ROS との通信が可能になり、roslibjs で JSON インターフェースを JavaScript で利用できる。ローカル経路計画側の ROS で rosbridge を起動し、同じネットワーク上にいることで、ROS 環境と非 ROS 環境の通信が容易に行うことができ、JavaScript で実装できるため web ブラウザ上、Android アプリ上など様々なプラットフォームで動作させることができる。

上記の調査から、ROS 同士の連携は、複数端末の連携システムが ROS のシステムとして用意されており、連携が容易ではあるが、ROS 環境同士という制約がある。rosjava と Android NDK は、実装環境を整えることが難しい。ファイル書き出しによる連携は、リアルタイム性がなく機能制限を受けることになる。そのため、ROS 環境と非 ROS

環境の通信が容易に行えて複数プラットフォームに対応できる rosbridge と roslibjs による連携方法を利用することとした。

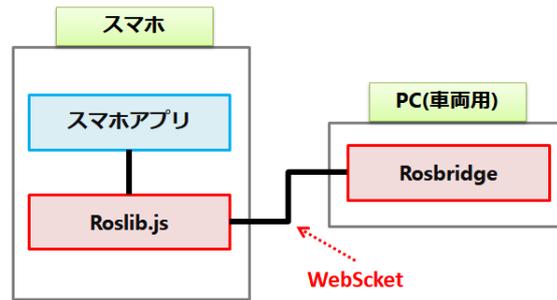


図 2-3-1_1 通信連携方法

ROS 連携方法の調査の詳細は『2-2-2-2_グローバル経路計画 ROS 連携方法調査.pptx』を参照。

③ waypoint 設定、API・I/F 仕様策定

グローバル経路計画連携機能で提供する waypoint の設定方法、及び、利用者が操作するアプリケーションに提供するインターフェースを検討した。

1. waypoint の生成に利用する地点情報群の設定

waypoint の取得・設定は、OSM の経路探索 API を利用することで実現可能である。

経路探索 API の graphhopper に対して、経度・緯度を座標とした 2 点以上の地点情報を設定することで、2 点間の経路を検索することができる。これは OSM の道路ネットワークを解析して 2 点間のルートを導き出すルーティングエンジンである。

graphhopper から得られる結果は、2 点間のルートを示すための経度・緯度の地点情報群となる。

グローバル経路計画連携機能で設定する waypoint は、この地点情報群を waypoint としてナビゲーションシステム、ローカル経路計画機能に提供することで連携機能を実現する。

2. ナビゲーションシステムの検討

グローバル経路計画で生成する waypoint 情報は、元となる地点情報を設定するためのユーザーインターフェースが必要である。グローバル経路計画連携機能の開発のための、ユーザーインターフェースを含んだサンプルアプリを作成する必要がある。

サンプルのアプリケーションは、グローバル経路計画連携機能を利用・検証することが目的であるため、比較的容易に作成できる Android アプリで作成することとした。

Android のアプリケーションには、地図アプリや、ナビゲーションアプリ等も提供されており、利用者が比較的容易に導入・操作できると思われる。

サンプルアプリの開発は、ローカル経路計画機能との連携に利用する rosbridge と roslibjs の ROS 通信ライブラリをそのまま利用するために、JavaScript での開発を行う。

JavaScript での Android アプリが作成可能な Cordova を開発環境として、Cordova のフレームワークは Ionic を利用することとした。Cordova は、HTML5、CSS、JavaScript などの web クライアントサイドの技術が用いられており、モバイルアプリが作成可能なフレームワークとなっている。

Ionic は、モバイルアプリによく使われるコンポーネントを再利用、改造できる形で提供するフレームワークである。

2-3-1-1-(2) グローバル経路計画連携機能の開発

インターフェース仕様の策定で調査した内容からグローバル経路計画連携機能の設計と実装を行った。

① グローバル経路計画連携機能のユースケース

利用するグローバル経路計画のシステム、ローカル経路計画機能との連携機能、利用者操作によるナビゲーションシステムを考慮した、グローバル経路計画連携機能のユースケースを以下に記載する。

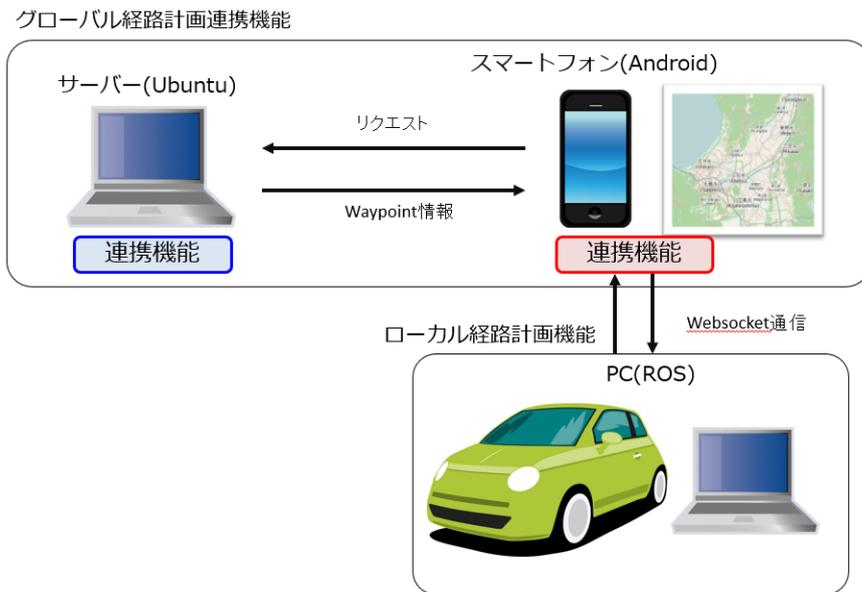


図 2-3-1_2 グローバル経路計画連携機能のユースケース図

【ユースケースの構想】

- 利用者が Android(スマートフォン、タブレット)機器に表示されている地図上に 2 点以上の地点情報(経度緯度)を設定して Ubuntu にリクエストする。
- Ubuntu 側で受け取った 2 点の地点情報から、走行経路を検索し、経路情報の waypoint を生成して Android 機器に waypoint 情報として返す。
- Android 機器の地図上に waypoint 情報を表示し、確認後にローカル経路機能を持つ ROS が搭載してある車両 PC に WebSocket 通信で送信する。
- ローカル経路計画機能は受信した waypoint に従い自動走行をする。

② グローバル経路計画連携機能のソフトウェア構成

上記のユースケースを元にしたグローバル経路計画連携機能のシステム構成を以下に示す。

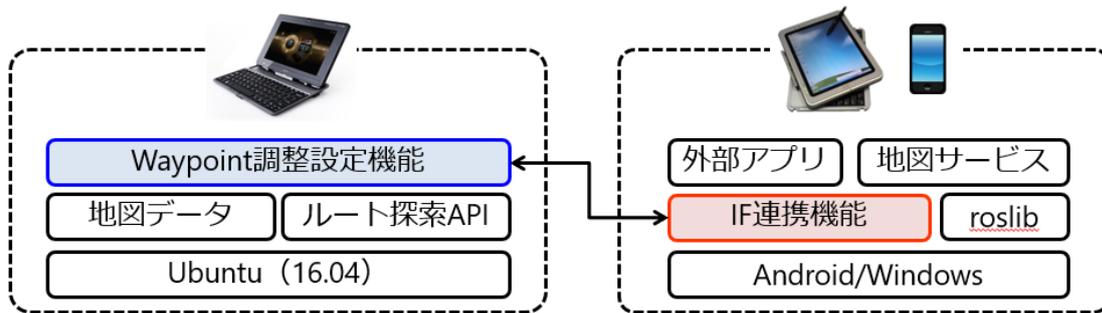


図 2-3-1_3 グローバル経路計画連携機能のソフトウェア構成図

グローバル経路計画連携機能は、2つのシステム上で動作するミドルウェアとして実装する。

主機能として Ubuntu16.04 上に搭載される「Waypoint 調整設定機能」と、Android 機器（ナビゲーションシステム）に搭載される「IF 連携機能」の2機能での構成とした。

「Waypoint 調整設定機能」は Ubuntu 上で動作するミドルウェアである。これは、OSM や graphhopper が Linux 上で動作するシステムであり、また、地図データを Linux 上に構築することで、カスタマイズした地図データを元にした経路探索も想定したためである。

「IF 連携機能」はローカル経路計画機能との連携機能、利用者の地図操作（走行計画）を実現するための標準的なインターフェースを提供するものである。この2つの機能が連動することにより、グローバル経路計画連携機能を実現する。上記の検討内容から、グローバル経路計画連携機能システム仕様書を作成した。

③ ナビゲーションシステムへ提供する API

グローバル経路計画連携機能の内、「IF 連携機能ミドルウェア」は、ナビゲーションシステム（外部アプリケーション）が利用するための API を提供する。

提供する API は、Ubuntu 側のシステムや、ローカル経路計画側のシステム（ROS）を考慮しなくても、ナビゲーションシステムに連携可能となる機能が提供できるよう設計した。

以下に、IF 連携機能が提供する機能一覧を記載する。

機能名	機能概要
ルート設定機能	始点、終点、中継点を元にルート探索のリクエストを行う
waypoint の手動設定機能	始点、終点、中継地点を任意に設定する
waypoint 間隔の補正機能	waypoint 挿入時の地点間の間隔を指定する
waypoint の再設定機能	設定された地点情報をクリアして、新しい地点情報を設定する
標高情報、走行不可のルート設定の切替機能	標高情報の取得、走行不可ルートについてルート探索時に有効・無効を指定する
ROS へのルート情報の送信	ルート探索および挿入した waypoint を含めた地点情報群をローカル経路計画（ROS）に対して送信する
地点情報取得機能	外部アプリケーションに対して地点情報を提供する

表 2-3-1_4 グローバル経路計画連携機能の API 一覧

ROS へのルート情報の送信は、rosbridge、roslibjs を利用することで、ROS との通信機能を提供する。この機能を利用することで、本機能を利用するナビゲーションシステム側は ROS との通信方式を考慮しなくても連携が可能となる。

地点情報の取得は、ナビゲーションシステムが OSM 以外の地図サービスを使用する場合を想定して、IF 連携機能が取得・保持している waypoint の地点情報群をアプリケーション側に渡す機能である。地点情報群は経度・緯度で構成されているため、他の地図サービスなどに対しても利用が可能であり、提供した地点情報を使うことで、他の地図にも waypoint を表示することができる。

④ ルート探索、waypoint 生成を行う機能

グローバル経路計画連携機能の内、「Waypoint 調整設定ミドルウェア」は、経路探索や waypoint の生成機能など、地図データの操作や、経路探索 API の操作機能を提供する。

Waypoint 調整設定ミドルウェアを搭載する Ubuntu 上に OSM の地図データを構築することで、オフラインでの利用や、カスタマイズした地図データを利用することが可能になる。

また、OSM でのルート探索を行うための経路探索 API として graphhopper を構築する。graphhopper を利用することで、ルート探索及び、各地点間の waypoint の生成が可能となる。

これらのシステムは IF 連携機能からのリクエストを受けて処理・操作する。

以下に、Waypoint 調整設定機能が提供する機能一覧を記載する。

機能名	機能概要
ルート探索機能	始点、終点、中継点を元に graphhopper を利用したルート探索処理を行う
waypoint 補正機能	ルート探索によって取得された地点情報群の各地点間に対して waypoint を挿入する
複数ルート探索機能	複数ルートの検索処理を行う（有効/無効の切替可能）
標高情報の取得機能	地点情報の取得時に、その地点の標高情報を合わせて取得する（有効/無効の切替可能）
走行不可ルート設定機能	走行不可の地点を入力することで、その地点を回避したルート探索処理を行う

表 2-3-1_5 グローバル経路計画連携機能の内部機能一覧

標高情報の取得は、走行する道路の勾配を事前に取得することにより、自動運転における加速・減速を判断する情報として、ローカル経路計画機能に提供することを想定した。

走行不可ルートの設定は、将来的なオンライン情報の活用を想定し、走行不可となるルートをあらかじめ設定することにより、走行可能な道路から走行ルートを決めることが実現できるよう考慮した。

⑤ グローバル経路計画連携機能のシステム連携図

ユースケース、ソフトウェア構成を元に詳細なシステム連携図を以下に記載する。

システム連携図は、グローバル経路計画機能（Ubuntu）側に OSM と graphhopper、ローカル経路計画機能に

rosbridge、ナビゲーションシステム (Android) 側に roslibjs を配置した、3つのコンポーネント間の連携を以下に示す。

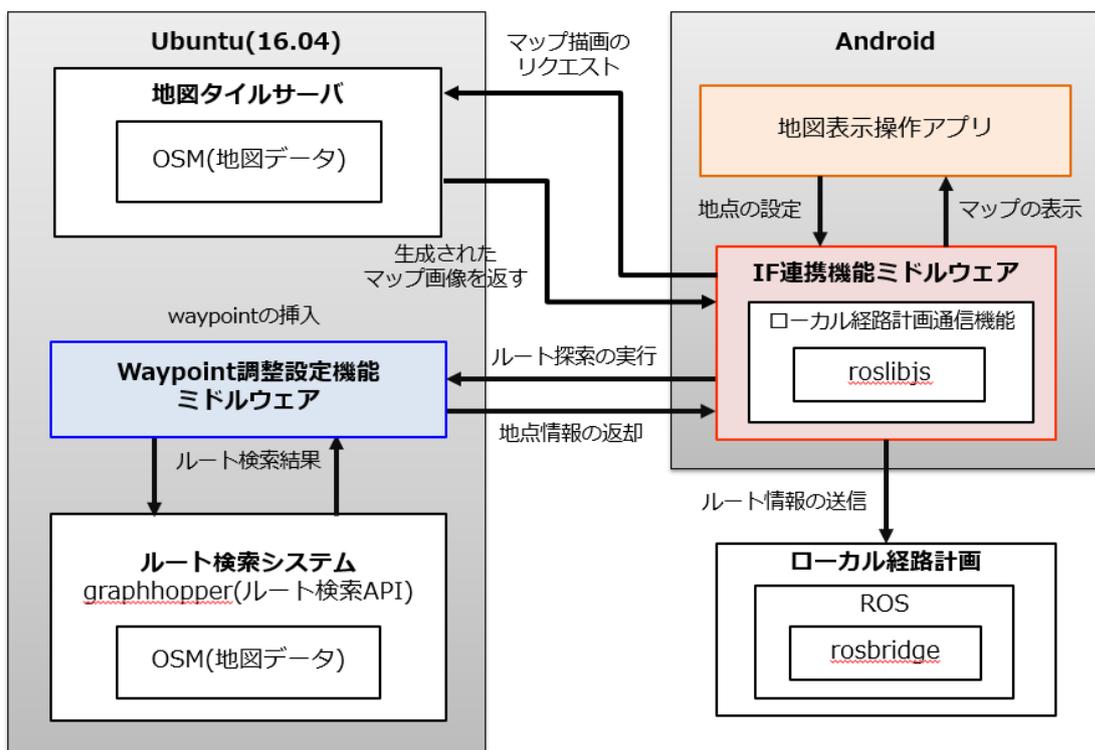


図 2-3-1_6 グローバル経路計画連携機能のシステム連携図

1. Waypoint 調整設定機能ミドルウェアの連携説明

Waypoint 調整設定機能は、Android 機器の IF 連携機能から地点情報を受信することで処理を行う。

ルート検索 API の graphhopper を利用して、OSM (OpenStreetMap) の地図データに対して指定された地点間の走行ルートを探査し、その経路上の地点情報から、waypoint 情報を生成する。この時、IF 連携機能で waypoint の距離間隔の補正等を必要とするリクエストがあった場合に補正に対応した処理を行う。

2. IF 連携機能ミドルウェアの連携説明

IF 連携機能は、OSM の地図をユーザーインターフェースとして表示する。利用者は地図を操作して地点を設定し、Waypoint 調整設定機能へのリクエストを行う。

Waypoint 調整設定機能にて生成された waypoint を取得すると、地図上に waypoint を表示して、ユーザーに走行ルートを示す。

また、IF 連携機能で保持する waypoint 情報をローカル経路計画機能の ROS に対して roslibjs と rosbridge を利用して送信を行う。

⑥ グローバル経路計画連携機能の実装

上述のユースケース、ソフトウェア構成、システム連携を元に、Waypoint 調整設定機能、IF 連携機能を開発した。

開発言語は JavaScript を用いて開発した。これは、OSM の地図データが JavaScript などの Web ベースでの利用を前提しており、経路探索 API の graphhopper についても同様に JavaScript での利用が可能のためである。

また、ナビゲーションシステムに提供する API として、外部アプリケーションが他の地図サービスを利用することを想定しており、JavaScript 言語の API を提供した方が、OSM 以外の地図サービスからの利用も容易となるためである。

グローバル経路計画連携機能として開発したシステムを用いて、地図データに対して waypoint が正常に設定・取得できるかを検証した。

以下に検証に使用した図を示す。



図 2-3-1_7 通常経路と waypoint 挿入比較図

通常、OSM と graphhopper を使ったルート検索は図 2-2-2_6 の左絵に示す通り、黄色の線のみでの描画となる。

自動運転でローカル経路計画機能に必要な地点情報は別途、graphhopper から取得する必要があるが、グローバル経路計画連携機能では、その地点情報を取得する機能を開発した。取得した地点情報群を表示した図が図 2-2-2_6 の右絵になる。

OSM の地点情報は graphhopper の標準仕様で提供される直線以外の道路（曲がり角や角度のある場所）に対して取得できるが、それだけでは自動運転に必要な waypoint とはならない。そのため、取得した地点情報の各地点間に対して、個別に地点情報を挿入する機能を開発した。

これにより、自動運転で必要な地点情報群を生成・取得して、ローカル経路計画機能に渡すことができるようになった。

ローカル経路計画機能の連携部分に関しては、roslibjs と rosbridge を使った通信を用いて、通信の疎通を確認した。

実際に開発したシステムを用いて、サブテーマ 1 の ROS を搭載した PC との Websocket による無線通信を行った。サブテーマ 3 のグローバル経路連携機能から送信したデータが、サブテーマ 1 側の PC で受信できることを確認した。

これにより、グローバル経路計画連携機能が、ローカル経路計画機能との連携が実現できていることを確認できた。

これらの機能をナビゲーションシステムに対する標準的なインターフェースとして提供することで、従来のカーナビや地図サービスなどでも利用することが可能である。

⑦ 単体試験評価

グローバル経路計画連携機能として開発したシステムが、正常に動作することを確認する必要がある。

また、本研究事業の中でシステムの改良を重ねていくことも想定され、改良したシステムによって、既に開発した機能が正常に動作し続けることも確認する必要がある。

このため、グローバル経路計画連携機能に単体評価を行うテストコードを実装することで、検証しながら開発を進めていく方針とした。

単体評価には、Jasmin を使用する。Jasmin は JavaScript で利用できるテストフレームワークであり、開発したコードを利用して、テストコードを作成することで、開発対象のコードの単体試験評価が実行できる。

Jasmin と合わせて Karma を使用する。Karma はブラウザ上で単体評価を実行するテストランナーであり、特定環境に依存せずに使用することができ、自動テスト及びテスト結果の出力が可能である。

2-3-1-1-(3)実証実験

本研究事業で車両の走行テストに利用する実証実験場は、苫東株式会社が保有する私有地（私道）であるため、waypoint 設定で利用する OSM、graphhopper のような公道を主とする経路検索は対象外となる。

サブテーマで利用する OSM、graphhopper による経路検索が可能な状態にする必要があるため、実証実験場の地図データを独自に作成して、OSM に反映することで経路探索を実現する。

そのため、GPS 機器を利用して実証実験場の地点情報の収集を行った。

①実証実験の目的

実証実験場の地点情報を、GPS 機器を用いて収集する。

収集した GPS データを地図サービスに表示して、地点情報が正常に取得できているかを確認する。

検証した結果、収集した道路に対して正常に取得できたデータを元に、地図データを作成する。

②GPS 機器の機能評価

実証実験場の GPS データを取得するため、スマホアプリの GPS ロガーアプリ、GPS 機器「HOLUX GPS ロガー M-241」の GPS 取得精度の事前検証を行った。

スマホアプリは GPSLogger (Android) を利用して、札幌市中央区を対象にした GPS データの収集を行った。

GPSLogger は 1 秒周期での GPS の収集が可能であり、取得したデータを GMX/KML ファイルとして出力することも可能である。出力したファイルを地図サービスに適用すれば、取得した地点情報を容易に確認することができる。

GPSLogger による検証では、人が歩道を歩いたときの GPS データを収集したが、収集したデータが正常に取得できなかった。これは、そもそものスマートフォンの GPS 機器自体の性能が低いため、取得できるデータもずれてしまう。そのため、スマホアプリによる GPS データの収集は本研究事業では使用しない方針とした。

GPS 機器「HOLUX GPS ロガー M-241」は、時間周期、距離周期での計測が可能であり、移動地点と時間から推定

移動速度の出力が可能な機器である。

精度としては、スマホアプリよりは良いが、登山用に使用されるポータブルタイプのため、数メートルから十数メートルとの誤差は発生してしまう。

本機器を用いて、札幌市中央区を対象にした GPS データの収集を行った結果、精度としては想定通り数メートルの誤差が発生する場合があるものの、比較的歩行した地点に即した地点情報が取得できていることが分かった。

この事前検証から、HOLUX GPS ロガー M-241 を利用して、実証実験場での GPS データの収集を行うこととした。

③実証実験場の GPS データの取得

実証実験場の地図を作成するための GPS データの収集を実証実験第 2 回、第 3 回に行った。データ収集方法は、小型の GPS 機器「HOLUX GPS ロガー M-241」を利用して、実験車両の Viking に乗せて収集した。

実証実験場は主に 2 ルートあり、直線路を 4 往復、周回路を 7 周分の GPS データの収集を行った。

HOLUX GPS ロガー M-241 の性能上、道路の周りが比較的開けたような場所（GPS データの収集を妨害する要因がない場所）であっても、GPS を配信する衛星の位置などによっては、地点情報がずれてしまうことが懸念されるため、同じコースを複数回収集することで、誤差を減少させることを考慮した。

④GPS データの検証

収集した GPS データを国土地理院の地図データや電子国土基本図に反映して正常に取得できているかを検証した。

国土地理院は国土交通省の管轄組織であり、国土地理院は日本国内における測量の基礎となる測量を提供する機関である。また、日本の基本図である地形図を発行しており、GPS データの検証・評価用に使用できるものである。

1. 実証実験第 2 回

取得した GPS データを国土地理院の地図に反映して検証した結果、直線路は道路に即した形で正常に地点情報が取得できていることがわかった。

周回路については道路からはみ出している部分が複数箇所あり、取得した GPS データが道路からずれて取得されていた。このため、実証実験第 3 回で再度、周回路に対して GPS データの収集を行うこととした。

2. 実証実験第 3 回

周回路に対して、再度取得した GPS データを検証したところ、第 2 回で道路からはみ出している部分に対して、道路に即した形で正常に地点情報が取得できていた。

⑤独自の地図データの作成

独自の地図データの作成は、取得した GPS を元に JOSM と呼ばれるツールで作成することができる。

以下に収集した GPS データと、JOSM で作成した地図データを重ねた図を示す。

1. 直線路の独自地図データ

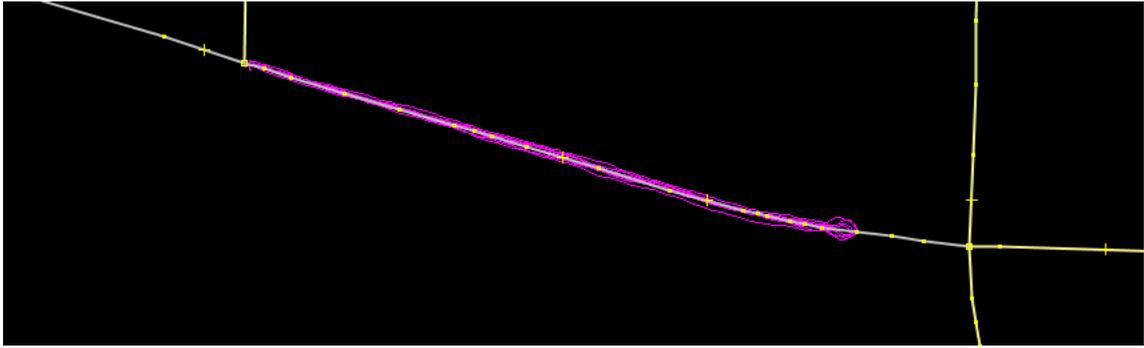


図 2-3-1_8 直線路の地図データと GPS データ

2. 周回路の独自地図データ

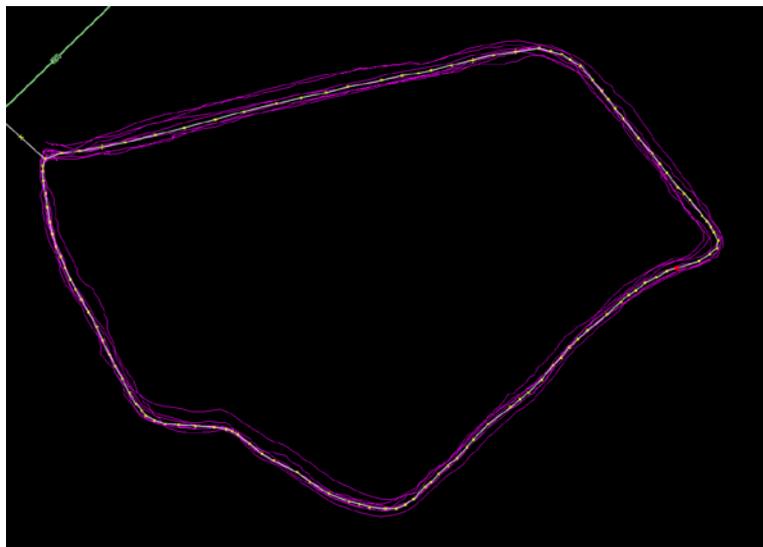


図 2-3-1_9 周回路の地図データと GPS データ

黄色の丸は作成した地図データであり、紫色の線は取得した GPS データである。
 紫色の線の内側に対して、地図データを整形することで、道路に即した地図データとなる。

2-3-2. 2年度の取り組み

2-3-2-1. 2年度の目標

・ Snow-SLAM との協調動作の最適化開発

グローバル経路計画連携機能が生成する Waypoint 情報を、Snow-SLAM の走行に適した地点に調整（補正）する機能を開発する。

(1) 課題

- ① ROS による走行制御は、地点情報以外に各地点に対して速度・進入角度を設定する必要がある。最終地点も同様に角度を設定する必要がある。
- ② ROS の走行制御は、地点の曲がる角度が大きい場合、ステアリングの切り返しが発生する。
- ③ 雪道走行では、夏道の道路に比べて、減速に要する距離が長くなるため、カーブ進入地点より手前で減速する必要がある。

(2) 課題への対策

- ① 各地点の進入角度を事前に設定し、Waypoint に付与する。
- ② 各地点間を通る滑らかな曲線を計算し、その曲線上を通る Waypoint を追加することで角度のある地点におけるステアリングの切り返しを抑制する。
- ③ カーブ前後の Waypoint に対して速度調整を行い、減速・加速する地点を設定する。

2-3-2-2. 具体的な内容

2年度は以下の開発を行う。

(1) Waypoint の速度・進入角度の設定機能の開発

グローバル経路計画連携機能から生成する各 Waypoint に速度・進入角度を設定する機能を開発する。

(2) Waypoint の経路補正機能の開発

グローバル経路計画連携機能から生成する各 Waypoint に付与する進入角度に対して、カーブでの Waypoint をなめらかな曲線となるよう補正する機能を開発する。

(3) 雪道走行での速度調整機能の検討

グローバル経路計画連携機能から生成する各 Waypoint に付与する速度に対して、カーブ前後の Waypoint を通過する際の速度の調整を行う機能を検討する。

(4) 実証実験

実験車両に搭載する知能化プラットフォームとグローバル経路計画連携を通信連携して Waypoint 情報に従って走行する実験を行い、ナビゲーション連携の実現に対応する。

2-3-2-2-(1)Waypoint の速度・進入角度の設定機能の開発

グローバル経路計画連携機能が地図データから取得する Waypoint に対して、速度・進入角度を設定する機能を追加する。

ア) 速度

速度はユーザが任意に設定できるように、アプリケーション画面に設定項目を設ける。設定された値を Waypoint 生成後に、各 Waypoint に対して付与する。

イ) 進入角度

地図データから取得する Waypoint は連続した地点情報（緯度・経度）である。3 点間の地点から各点における角度を計算する。開始地点は角度を 0 とし、最終地点は 1 つ前の地点と同じ角度を設定する。

速度、進入角度を付与した Waypoint 情報は「経度・緯度・速度・角度」の順で構成する。この構成については 知能化プラットフォームとインタフェースを整合して決定した。

141.735989667 42.6779203333 10 0
141.736014333 42.6779136667 10 10
141.736257167 42.6778523333 10 15

図 2-3-2-2-1 Waypoint 構成例

2-3-2-2-(2)Waypoint を補正する技術の開発

① Waypoint を補正する技術の調査

グローバル経路計画連携機能が地図データから取得する Waypoint を補正するための技術に関する調査を行った。

ア) クロソイド曲線

クロソイド曲線とは、曲率を一定割合で変化させていった場合に描かれる軌跡のことを言う。一定速度で走行中に、ハンドルを一定の速度で回したときに車が描く軌跡である。路線の直線部分と円弧曲線部分をつなぐ緩和曲線として、クロソイドの一部が利用されている。

直進と円弧だけで構成されたカーブを運転する場合、円弧の始まりと終わりでハンドルを一気に操作する必要がある。

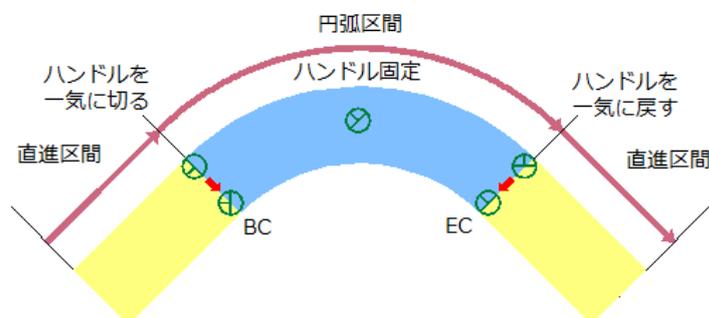


図 2-3-2-2-1 クロソイド区間なしのカーブの走行イメージ

上記の場合、急激なハンドル操作による横Gでの搭乗者へのダメージや横転の可能性がある。これに対して、直進と円弧の間にクロソイド区間を設けたカーブを運転する場合、ハンドル操作に余裕ができる。

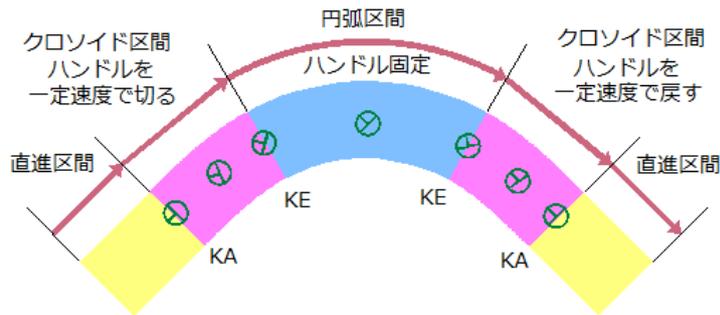


図 2-3-2-2-2 クロソイド区間ありのカーブの走行イメージ

イ) Spline補間曲線

スプライン補間とは複数の点が与えられた時に、点の間を3次の多項式で近似し、滑らかに補間する手法である。それによって描かれる曲線を *Spline*補間曲線 (*Spline*曲線) という。

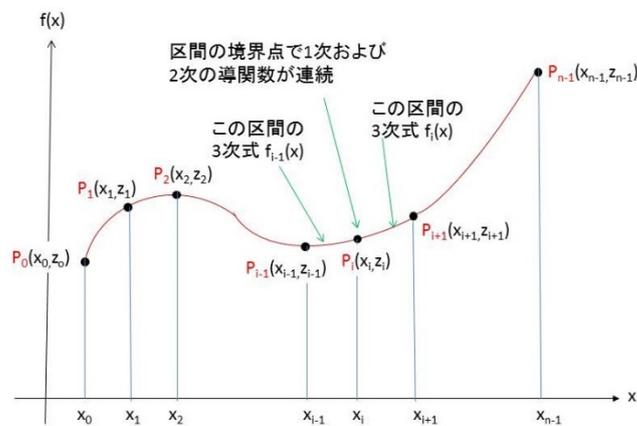


図 2-3-2-2-3 3次 *Spline*曲線

クロソイド曲線は現実の道路設計に用いられている技術であり、カーブでのなめらかな走行を実現するためには適した技術である。しかし、クロソイド曲線に求める場合、以下の情報を考慮する必要がある。

- 1) 車体
 - 速度
 - 車体サイズ
 - 内/外輪差
- 2) 道路
 - waypoint 間の角度
 - 障害物
 - 前方車両

つまり、自動走行中にリアルタイムでの計算処理が必要であり、車両側からの情報と地図データの情報から計算する必要がある。計算処理に遅れが発生した場合、正常な Waypoint を導き出すことができず、自動走行に障害をきたす恐れがある。

上記に対して、*Spline*曲線は地図データにおける緯度・経度情報から算出することが可能であり、車両側からの情報がなくても、グローバル経路連携機能単独での実現が可能である。

*Spline*曲線を用いたロボットなどの移動制御を行う技術は一般的に使用されており、グローバル経路連携機能でも本技術を適用した。(参考文献: 小森谷 清・谷江 和雄 (1990) 「スプライン曲線による車輪型移動ロボットの軌

道制御」.)

また、本研究事業で開発しているグローバル経路連携機能は、自動走行する前（停止状態時）に Waypoint を渡して、車両の自動走行開始で走行位置を判断するため使用することを目的としており、リアルタイムでの動的な走行制御には対応しない。実際の走行時における走行地点を決定するのは知能化プラットフォーム側であり、各種センサ情報から障害物や実際に走行できる位置を特定して走行する必要があるためである。

グローバル経路連携機能が地図データから取得する Waypoint に対して、*Spline*曲線を用いてカーブでの走行位置を事前に補正するよう対応する方針とした。

② Waypoint を補正する技術の開発

ア) 経路補正機能の方針

- 地図データから取得した Waypoint 群から、一定角度以上の点群を抽出し、なめらかな軌跡を描く曲線（スプライン補間曲線）を求める。
- その曲線上を通る点を取得して、Waypoint に挿入する。
- 地図データから取得した Waypoint は必ず通るようにする（逸脱してしまうと道路からはみ出してしまう恐れがあるため）
- スプライン補間による補正地点は、一定角度以上2点間の間に挿入する。

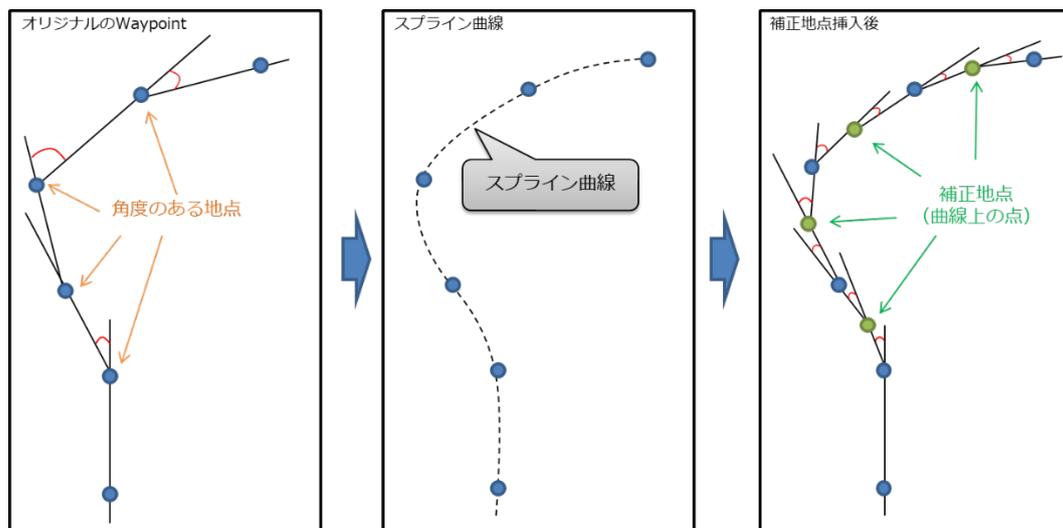


図 2-3-2-2-4 *Spline*曲線による Waypoint の補間図

イ) 経路補正機能導入による課題と解決

Waypoint に補正地点を追加して、各 Waypoint の角度を再計算した結果、以下の課題があった。

1. 角度の大きい地点でも補正角度が小さくならない場合がある。

次の地点までの距離が短い場合、直角になってしまう。また、道路の形状として、どうしても角度が大きくなる地点は存在する。

2. 角度の小さい地点でも補正してしまう。
3. 補正する最小角度を設定する必要がある。

上記、課題に対して、以下の対策を行った。

1. 補正地点の挿入数を変更する。
2. 補正地点は各点の角度差から制限する。
3. 補正する地点間の距離から制限する。

1. 補正地点の挿入数を変更する。

角度の大きい地点に対して、その前後に挿入する数を増やす対応を行ったが、地点が多くなり過ぎた場合、各地点間の距離が近くなってしまう。

知能化プラットフォームで制御される走行経路は、ROS が走行位置を判断している。ROS の走行では、次の Waypoint に対して半径 5m 以内に侵入した時点で、その地点に到達したとみなして、その次の地点へ移動する。そのため、Waypoint 数を増やしても、その地点が無視されてしまうため補正の効果が期待できない。

上記理由により ROS は Waypoint の直上を厳密に通るわけではなく、また、車両側でのセンサによる走行位置決定の障害にならないようにするため、本対応は不要とした。

2. 補正地点は各点の角度差から制限する。

各地点の角度差が大きい地点（20 度以上）は補正の対象とした。この場合、20 度以上の前後の地点を含めてスプライン関数を求める必要があるため、前後の角度差が 20 度以下（10 度以下の場合）でも対象とする。逆に、角度差の小さい地点（10 度以下）は補正対象外とした。

角度の小さい地点を補正の対象としなければ、連続した点群に対する曲線とならないため、10 度以外の地点が曲線からはみ出してしまい、なめらかな曲線ではなくなった。そのため、本対応は不要とした。

3. 補正する地点間の距離から制限する。

補正する地点間の距離を算出し、その距離が近い場合は補正した地点を挿入しないよう対応した。補正する地点は取得する Waypoint によって異なるため、適した距離を算出するために、実験での検証を想定した。そのため、本機能をユーザが設定する画面に入力項目を設けることによって対応した。

この対応により、直角路に近い区間では、各地点の角度が補正前より大きくなることは避けることができるようになった。

上記対応により、Waypoint を補正することが可能となり、カーブでのなめらかな走行を実現するための Waypoint を生成することができるようになった。

2-3-2-2-(3) 雪道走行での速度調整機能の検討

雪道走行では、カーブへ進入する手前で速度を落としても、路面との摩擦係数が小さいため、夏道と比較して速度が落ちにくい。そのため、カーブより手前の地点から段階的に速度を落とすことが必要であり、その地点を Waypoint に含める対応を行う。

2 年度では、その地点を求めるための調査・機能の検討を行った。

① 雪道における車両の速度調査

雪道では夏道と比較して路面での摩擦係数が以下のように小さくなる。

- 夏道：0.7～0.9

- 冬道：0.2～0.4

また、時速40キロで「圧雪路」と「アイスバーン」を走行し、急ブレーキを踏んだ時の制動距離はタイヤの種類によって以下のような差がある。

【圧雪路の場合（単位はメートル）】

- ノーマルタイヤ：29.9
- スタッドレスタイヤ：17.3
- ノーマルタイヤ+チェーン：28.4

【アイスバーン】

- ノーマルタイヤ：105.4
- スタッドレスタイヤ：78.5
- ノーマルタイヤ+チェーン：59.0

圧雪路では、スタッドレスタイヤが最も短い距離で停止するが、アイスバーンでは、チェーン装着が最も短い距離で停止することが実験から判明している（参考文献：JAF ユーザーテスト『http://www.jaf.or.jp/eco-safety/safety/usertest/tire/detail2_1.htm』）

上記のデータから、冬道では夏道に比べて車両の速度が落ちにくい状態になるのは明白である。

② 雪道走行での速度調整機能の検討

ROS の走行制御における速度調整はカーブでのステアリングによって速度が減速することはあるが、ステアリングの制御角が小さい場合は一定速度（カーブ進入時の速度）のまま走行する。

また、Waypoint に付与した速度に対して ROS の速度の加減速は、その Waypoint を通過したタイミングで行われる。例えば、ある地点（A 地点）で速度が 20km/h、その次の地点（B 地点）で速度が 10km/h と指定した場合、ROS は A 地点から B 地点に対して徐々に速度を落とすのではなく、A から B までは 20km/h のまま走行し、B 地点に到達したタイミングで 20km/h から 10km/h まで減速する。

このため、カーブの進入地点の速度を小さくしても、速度を落とし始めるときにはカーブに侵入してしまう。また、上記の調査結果からも冬道での速度を落とすためには、長い距離を要する。

したがって、カーブよりも手前の直線路の地点かつ、十分に速度を落とすことのできる距離から速度の調整を行う必要がある。

ア) カーブ地点前後の速度調整の方針

- カーブ進入手前の Waypoint に対して速度の減速を指定する。
- カーブ通過後の Waypoint に対して速度の加速（元に戻す）を指定する。
- 地図データから取得したカーブ進入手前の地点では、十分な減速を行える距離が不明のため、カーブに進入する手前に地点を追加して、その地点に対して減速を指定する。

速度を調整するための Waypoint を挿入する地点はカーブより手前の直線路に対してであり、カーブの進入地点から一定距離以上の地点を導出する。直線路は連続した Waypoint のため、距離が分かれば、地点情報を得るのは容易である。

イ) カーブ地点前後の速度調整機能の検討

- カーブ前後の Waypoint の速度を調整する。
- カーブ地点での速度を 10km/h 以下と仮定して、30～50 メートルの手前の地点に対して、速度を減速する指定を行う。その地点に Waypoint がない場合、Waypoint を追加する。
- カーブ地点付近は一定の速度 (10km/h) を保つように指定する。
- カーブ通過後は元の速度を指定する。

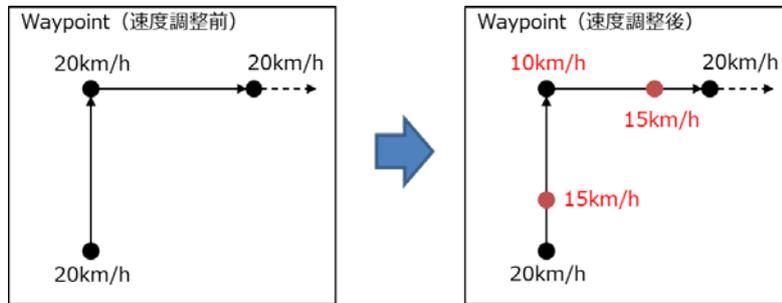


図 2-3-2-2-1 速度調整用の Waypoint の挿入イメージ

ウ) 実装方針

- カーブ地点 (角度のある地点) は 5km/h で走行するよう速度を指定する。
- カーブ進入手前から段階的に速度を減速し、カーブ通過後の直線路で元の速度に戻す (10m 間隔で 5km/h ずつ減速または加速するものとする)

	-40m	-30m	-20m	-10m	0	+10m	+20m	+30m	+40m
10km/h	10	10	10	10	5	10	10	10	10
20km/h	20	20	15	10	5	10	15	20	20
30km/h	30	25	15	10	5	10	15	20	30
40km/h	35	20	15	10	5	10	15	20	30
50km/h	40	30	20	10	5	10	15	20	30

↑カーブ開始地点 ↑カーブ終了地点

図 2-3-2-2-2 カーブ地点前後の Waypoint で指定する速度

図 3-2-2-3-2 の速度はまだ検討段階であり、本機能の対応は「積雪環境の仮想シミュレータ」を用いて評価を行い、精度の向上を行うことを想定している。

2-3-2-2-(4)実証実験

初年度に開発したグローバル経路計画連携機能に、2年度対応した Waypoint 補正機能を含めた状態で、知能化プラットフォームと通信連携を行い、グローバル経路計画連携機能が生成した Waypoint による自動運転走行の実験を行った。

(1) 実証実験

【観点】

直線、カーブ地点における Waypoint と GPS による走行確認を行った。

1. 曲がる際の速度の加減速（ROS 制御による速度はどうなっているか）
2. 車両の走行位置（目視による確認+道路のどの辺りを走行しているか）
3. カーブ走行の安定性（搭乗者の乗り心地感）

【検証結果】

① 北大の創成棟付近（2018/12/17）

ア) 知能化プラットフォームとの連携確認

知能化プラットフォームからグローバル経路連携機能への車両の自己位置の送信は GPS 機器が動作できなかったため、実施できなかった。

グローバル経路連携機能から知能化プラットフォームへの Waypoint（速度、角度含む）の送信は、動作が確認でき、担当者間で Waypoint の値が問題ないことを確認した。

イ) 車両走行確認

車両側の調整不良により走行確認は実施できなかった。

本実験により、グローバル経路計画連携機能が生成する Waypoint の値が正常に送受信できることは確認できたが、車両の走行確認は実施できなかったため、次の苫東での実証実験に持ち越した。

② 苫東の周回路（1 回目：2018/12/20）

ア) 車両走行確認

車両側の調整不良により走行確認は数回のみ実施した。

グローバル経路計画連携機能から送信した Waypoint を、知能化プラットフォームが受信して自動運転走行を開始すると、指定した Waypoint とは反対の方向へ走行する結果となった。グローバル経路計画連携機能が送信した Waypoint の地点情報や角度には問題がなかったが、知能化プラットフォーム側の角度計算処理で東西の計算に誤りがあり、正常に走行することができなかった。

知能化プラットフォーム側の角度計算を修正する時間が必要だったため、車両の走行確認は中止、次の苫東での実証実験に持ち越した。

③ 苫東の周回路（2 回目：2019/2/21）

ア) 知能化プラットフォームとの連携確認

知能化プラットフォームからグローバル経路連携機能への車両の自己位置の送信を確認し、グローバル経路計画連携機能の地図上に自己位置が表示されることを確認した。

イ) 車両の走行確認

知能化プラットフォームの角度計算が修正され、Waypoint に従って走行することを確認した。

本実験では、グローバル経路計画連携機能は速度調整機能には対応しておらず、実験時の速度は 10km/h 前後での走行とした。また、走行ルートは周回路を 1 周して確認したが、GPS データの不具合などにより走行を一時停止して、区間ごとでの走行確認を行った。

以下、実証実験での検証観点について確認した結果を記載する。

1. 曲がる際の速度の加減速

ROSによるカーブ地点での速度の加減速について、一定以上の角度（ハンドル制御が大きい場合）がないと減速せずに、そのままの速度で走行することを確認した。

2. 車両の走行位置

道路の真ん中付近を走行することを確認した。

ただし、場所によってはGPS機器が受信する地点がずれてしまう場合があり、道路脇に寄って走行してしまう場面も見受けられた。これは苫東の周回路脇に木が生い茂っているため、木々が多い場所ではGPS機器の受信に影響を与えてしまうためである。影響が少ない場所ではGPS機器が正常に地点を取得でき、走行位置は道路の真ん中付近を安定して走行していることを確認した。

3. カーブ走行の安定性

道路の真ん中付近を走行することを確認した。

カーブの開始から終わりまで滑らかに走行することを確認した。また、進入角度による切り返しが抑制できていることも確認した。

一部のカーブ地点では、多少の横滑りは発生していたが、これは雪や轍による影響である。雪は20～30cm程度の積雪であり、気温によって少し溶けた状態となっていた。また、実証実験の繰り返しで轍ができてしまい、そこでタイヤが滑ってしまうことはあった。

搭乗者の意見としては、手動で運転した場合でも同様の横滑りは発生しており、道路状況が悪かったことが要因であると考えられる。

道路状況を踏まえても、実証実験でよく搭乗している者によれば安定して走行していたとの意見を得ることができた。

実証実験の結果から、Waypointの補正機能によって、各地点の進入角度や走行位置に必要以上にずれるなどの影響がなく、かつ、カーブからの逸脱や、曲がり切れなかったなどの問題も発生しないことを実証した。

2-3-3. 3年度の取り組み

2-3-3-1. 3年度の目標

- ・移動支援アプリを用いた実証実験と性能改善
 - ・実際のユーザが車両の走行経路を指定可能とするアプリケーションを開発する。
 - ・移動支援アプリと車両との連携を行う機能を拡張する。

(1) 課題

- ④ 実際のユーザが操作することを想定したユーザインターフェースを用意する必要がある。
- ⑤ 自律走行中は、ユーザ観点として車両が正常な状態で走行しているかを示す必要がある。
- ⑥ 雪道走行では、夏道の道路に比べて、減速に要する距離が長くなるため、カーブ進入地点より手前で減速する必要がある。

(2) 課題への対策

- ④ ユーザが操作するカーナビゲーションと類似するアプリケーションを開発し、ユーザ操作によって走行ルートを決定させる。
- ⑤ アプリケーションに車両の状態を表示し、異常などの状態を検知してユーザに知らせる機能を開発する。
- ⑥ カーブ前後のWaypointに対して速度調整を行い、減速・加速する地点を設定する。また、各Waypointの地点に対して速度を設定する機能を設ける。

2-3-3-2. 具体的な内容

(1) 移動支援アプリの開発

Android 端末でグローバル経路計画連携機能を用いた走行ルートの決定を行うためのユーザーインターフェースアプリケーションを開発する。

グローバル経路計画連携機能はAndroid アプリから利用可能なライブラリとしてJava 言語のインタフェースを用意する。

- 移動支援アプリは地図サーバと通信して地図情報を画面に表示する。
- ユーザは地図上の任意の地点を選択して走行経路を検索・決定する。
- アプリが車両と通信して走行情報 (Waypoint) を送信し、車両が自律走行を行う。

(2) 車両連携機能の拡張

車両の状態を取得するインターフェースを用意し、車両から各種情報を受信して、ユーザに通知する機能を開発する。

- (ア) アプリの画面上に、車両の走行状態、異常状態 (GPS 取得失敗) を表示する。
- (イ) アプリの画面上に、自律走行に使用している各センサモジュール (LiDAR/GPS/SS/サーモ) の状態を表示する。

(3) 雪道走行での速度調整機能の開発

昨年度、検討した内容を基に、グローバル経路計画連携機能が生成する各Waypoint に付与する走行速度を、カーブ前後の地点に対しては速度の調整を行う機能を開発する。

(4) 実証実験

移動支援アプリの操作によって、実験車両に搭載する知能化プラットフォームとグローバル経路計画連携を通信連携し、移動支援アプリから送信する Waypoint による自律走行、および知能化プラットフォームから受信する車両情報を移動支援アプリに表示する機能の実現に対応する。

2-3-3-2-1 移動支援アプリの開発

昨年度まで開発していたシステムは、OSM の地図データ、ルート検索 API の GraphHopper、サンプルアプリ、グローバル経路計画連携機能がすべて Ubuntu PC 上に搭載されたものであった。

ユーザによる操作可能なナビゲーションシステムは、ユーザが実際に操作するための端末が必要であり、本研究では Android 端末によるアプリケーションの開発を行うこととした。これは、カーナビゲーションシステムとして、現在広く利用されているものは車両メーカーや車載メーカーが開発する車両と一体化したナビゲーションシステムだけでなく、スマートフォンで動作するナビアプリケーション等も利用されるようになってきており、ユーザが操作に慣れた端末上で、自律走行に必要な操作をさせることを考慮したものである。

以下に、システム構成図を示す。

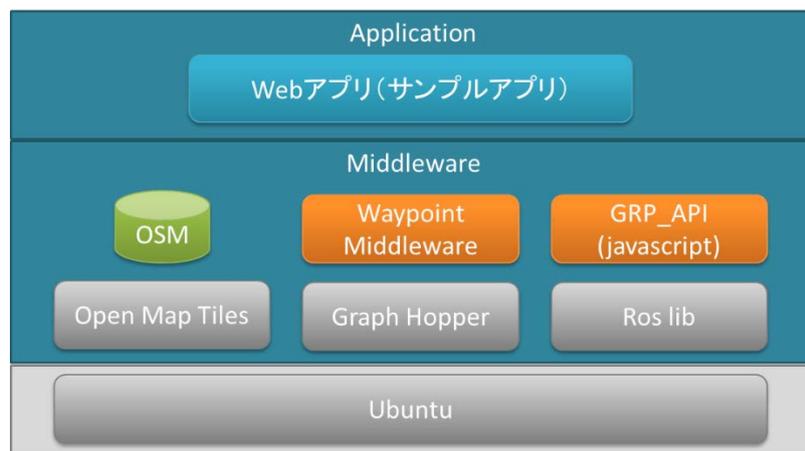


図 2-3-2-2-1 昨年度まで開発したシステム構成

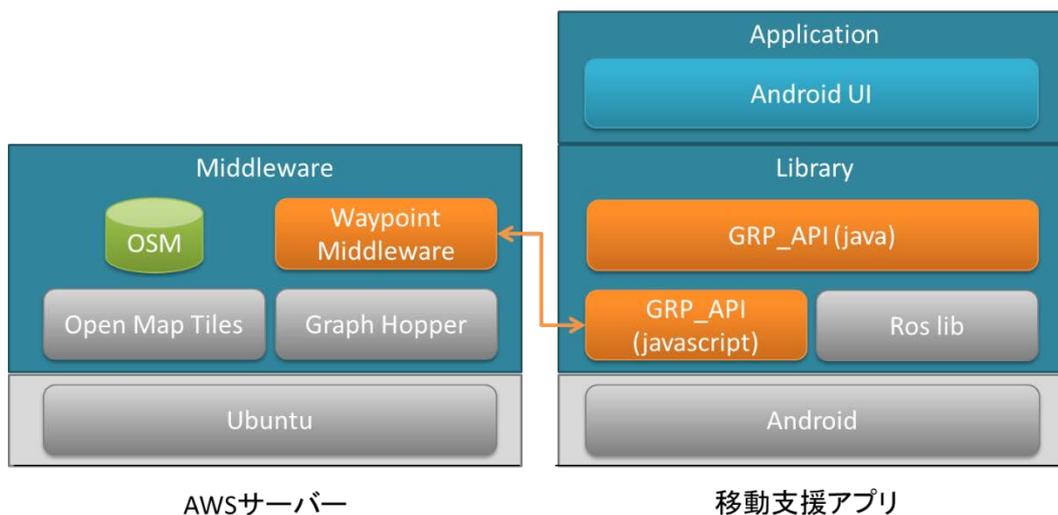


図 2-3-2-2-2 3 年度開発したシステム構成

ア) AWS サーバ

- WaypointMiddleware : OSM から地図画像の取得、地点指定による Waypoint の取得を行う。
- OSM : 地図データ。
- GraphHopper : オープンソースの API であり、ルート探索(1つ/複数の切替が可能)を行う。

イ) 移動支援アプリ

- GRP_API (javascript) : WaypointMiddleware と通信連携して、地図データの取得、Waypoint の取得を行う。また、Roslib を介して車両と通信連携を行う。
- GRP_API (java) : 「GRP_API (javascript) 」を AndroidJava インターフェースで拡張したライブラリ。
- Android UI : 地図表示アプリケーション。
- Roslib : 車両(知能化プラットフォーム)との通信インターフェース。

移動支援アプリは、本研究における試作的な開発物であり、グローバル経路計画連携機能は、図 3-2-2-1-2 に示す WaypointMiddleware、GRP_API (javascript/java) が対象である。

3 年度は、グローバル経路計画連携機能を Android アプリケーションからでも利用可能とするための AndroidJava インターフェースを拡張した。これにより、従来の Android アプリケーション開発者は、自律走行車両と連携するアプリケーションを開発する際に、地図サーバへの通信アクセスや Waypoint の取得、ルート検索などの機能が GRP_API のインターフェースを利用することで容易に実現可能となった。同様に車両の通信連携や、車両から取得する情報なども GRP_API のインターフェースを利用することで容易に実現可能となった。

2-3-3-2-(2) 車両連携機能の拡張

自律走行する車両の搭乗者は、従来自分で制御していた運転操作等を車両システムに任せることになる。その場合、搭乗者は車両が正常に自律走行できているか不安を抱くと思われる。そのため、車両が正常に走行可能な状態であり、安心して搭乗できることを搭乗者に示す必要がある。

サブテーマ③が開発する移動支援アプリは搭乗者が操作するナビゲーションシステムであり、搭乗者にとって必要な情報である車両の状態を通知する機能を追加することとした。

知能化プラットフォームから取得可能な情報を精査し、以下に示す情報を知能化プラットフォームから通知してもらい、アプリケーション上に表示する機能を開発した。

ア) 車両ステータス

- 車両の走行状態：走行中/停車中の状態を通知する。
- ハンドルの舵角：走行中の舵角を通知する。
- 車両本体の故障検知：車両本体の故障を検知した場合に通知する。

イ) センサモジュールステータス

- センサの名称、センサの稼働状態を通知する。

ウ) コストマップステータス

- 知能化プラットフォームの障害物の検知状態を通知する。

通知する車両情報について、選定した理由を以下に示す。

ア) 車両ステータス

自律走行中は知能化プラットフォームが車両の走行/停止やハンドル操作を制御しており、ROS を含めた知能化プラットフォームがどのように走行や停止を判断しているかは搭乗者にはわからない。

例えば、初年度の実証実験中に発生した車両の動作として、目的地に到着した後に、ROS によってハンドル操作や車両が前後へ移動するなどの動作が発生したときがあった。これは、目的地の地点に対して、ROS が車両の向きを合わせようとした場合や、距離が離れているために地点近くに移動しようと判断して発生するものであった。

こういった事象を踏まえた上で、車両が停止したのは、走行中の一時停止なのか、完全に停車したのか等の情報を搭乗者に通知することによって、自律走行への安心性を高めることができると考える。同様に、車両本体で故障が発生した場合、搭乗者に通知することで、手動で走行の停止を促すことも必要であると考え、これらの通知機能が必要であると判断した。

イ) センサモジュールステータス

搭乗者は、各センサがどのような制御に使用されているかは詳しく把握していない。しかし、自律走行に必要なセンサが正常に稼働しているかを通知することで、搭乗者は安全に走行できていると安心することができる。また、センサが異常状態の場合には、自律走行が正常に走行できない可能性があり、搭乗者に通知して手動で走行の停止を促すことは重要であると考え、センサの稼働状態を通知する機能が必要であると判断した。

ウ) コストマップステータス

コストマップとは ROS がセンサから障害物を検知して、ダイナミックマップ化したものであり、コストマップ上にある障害物の地点は走行不可と判断するために使用される。

各センサは正しく稼働している状態にあっても、障害物を検知できない状況が発生すると、コストマップとして正しく生成できない場合がある。

例えば、実証実験中の雪道走行において、積雪や降雪によっては障害物を正しく検出できないときがあった。他の状況として、走行する経路に遮蔽物がない場合も障害物の検知率が低くなってしまい、コストマップ上ではほとんどの場所が走行可能と判断されてしまう。実際に、人の目には障害物がある（走行できない）と判断できる場合でも、ROS では走行可能と判断されてしまった場合、その地点へ移動する可能性がある。

そのため、障害物の検知率が低い場合やコストマップが正常に生成できていない場合は搭乗者にそれを通知し、停車するなどの措置を取れるようにする必要があると考え、この機能を追加した。

上記の各ステータスは、知能化プラットフォームと通信連携して実現するため、新たなインターフェースを追加して開発した。

検討内容の詳細、および、インターフェースの詳細は『3-2-2-2-1_グローバル連携機能と車両との連携 IF 整合資料.xlsx』として取りまとめた。

アプリケーション上に表示する各ステータスのユーザインターフェースの詳細は『3-2-2-2-2_移動支援アプリのユーザインターフェース資料.pptx』として取りまとめた。

2-3-3-2-(3) 雪道走行での速度調整機能の開発

雪道走行では、カーブへ進入する手前で速度を落としても、路面との摩擦係数が小さいため、夏道と比較して速度が落ちにくい。そのため、カーブより手前の地点から段階的に速度を落とすことが必要である。3年度は、昨年度検討した速度調整機能の内容を元に、Waypoint に付与する走行速度を調整する機能を開発した。走行速度の調整機能を開発するにあたり、Waypoint の地点補正についても考慮した上で対応を行う必要があった。Waypoint は

GraphHopper を使用して地図データからルート情報を取得したものである。GraphHopper にはルート上の地点を追加補完する機能があり、昨年度までに開発したシステムにおいては指定した距離で等間隔に補正する機能を設けていた。しかし、実証実験において、直線路を走行する際に、必要以上に Waypoint が存在する場合、その Waypoint へ向かってしまうため、走行がふらつくといった報告があり、Waypoint の補正機能を見直す必要があった。

③ Waypoint の補正機能の改善

Waypoint の補正機能に対して、以下の改善を行った。

- 直線はルート検索した結果の地点のみを設定する。
- カーブは設定間隔で補正地点を追加する。

カーブの特定は、連続した Waypoint の 3 点間から角度を算出し、指定角度以上の地点間を対象として、GraphHopper によるルート検索を再度実行し、その際に指定された補正間隔で地点を追加することとした。

④ Waypoint の速度調整機能の開発

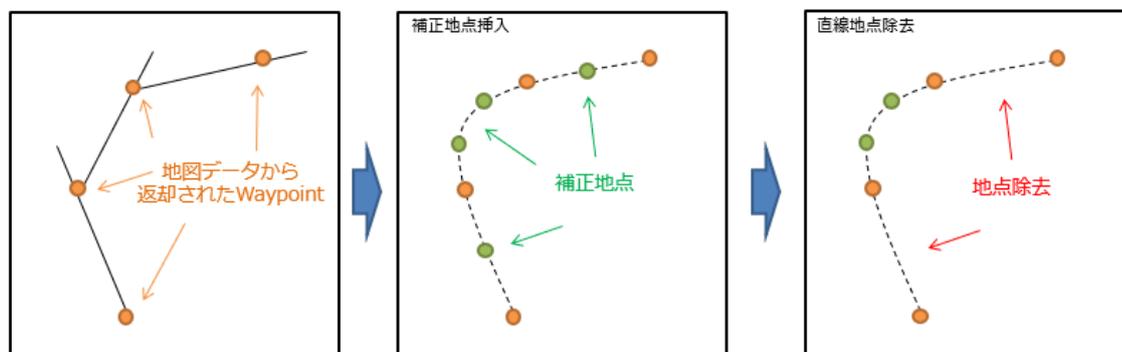
Waypoint に対して自動で走行速度を調整する機能を追加した。

初期速度はアプリケーションに設定項目を設けて、任意の整数値 (km/h) で入力可能とした。

速度調整機能は以下の方針で開発した。

- 直線は初期速度を設定する。
- カーブの進入手前の直線路からカーブ地点までの間を最低 5km/h まで段階的に減速させる。
- カーブの通過後から直線路までの間を段階的に初期速度まで加速させる。

カーブの特定は Waypoint 補正機能と同様の処理で算出する。



上述の機能を追加した移動支援アプリの Waypoint を使用して、実証実験にて自律走行を検証した。

Waypoint の補正によって設定された地点や走行速度は適切であり、車両が直線路やカーブ地点を問題なく走行することを実証実験にて検証できた。

2-3-3-2-(4)実証実験

3 年度開発した移動支援アプリを用いて自律走行の実証実験を行った。

① 実証実験の概要

以下の機能について検証を行った。

- 移動支援アプリはAWS サーバと通信して地図データを取得してアプリケーション上に表示し、ユーザ操作によってルート検索を行う。ルート検索は移動支援アプリからAWS サーバへ通信し、ルート検索結果をアプリケーション上に表示する。
- 移動支援アプリは知能化プラットフォームと通信連携を行い、取得したWaypointを渡して自律走行を開始する。
- 車両の状態や各センサの稼働状態を知能化プラットフォームから受信し、アプリケーション上に表示する。

また、Waypointの補正などによって自律走行の影響についても検証した。

② 実証実験の内容

④ 苫東 (2019/11)

ア) 知能化プラットフォームとの連携確認

移動支援アプリと知能化プラットフォームの通信連携を行い、Waypointの送信および、車両の自己位置を受信して地図上に表示できることを確認した。

イ) 車両走行確認

移動支援アプリの送信したWaypointで自律走行を行った際に、実際の道路からずれた場所にWaypointが設定されていることがわかった。

苫東の地図データは初年度にサブテーマ③担当者が登山用のGPS機器を用いて地点情報を取得し、それを基点に地図データを作成したものであったため、地図データの精度に問題があることが判明した。

地図データについては、サブテーマ①のGPS機器でみちびきから取得したGPS情報を元に地図データを修正して、次回の実証実験で効果を検証することとした。

このため、本実験においては安全性を考慮して、移動支援アプリのWaypointによる自律走行は断念した。

⑤ 苫東 (2019/12)

ア) 知能化プラットフォームとの連携確認

移動支援アプリと知能化プラットフォームの通信連携ができなかった。前回確認した移動支援アプリから追加対応したのは、Waypointの補正機能の改善処理であり、知能化プラットフォームとの通信処理は修正していなかった。

前回使用した端末とは別の端末を使ったことによる影響の可能性もあり、環境面も含めた調査を行うこととした。

イ) 車両走行確認

車両の走行に関して、Waypointは移動支援アプリから通信によって直接送信することができなかったため、移動支援アプリで取得したWaypointのデータを手動(USB)でサブテーマ①に渡して、サブテーマ①がデータ取り込んで自動走行の検証を行った。

苫東の地図データを修正したことにより、Waypointの地点情報のずれは解消されていることを確認した。また、Waypointの補正機能を改善したことにより、直線路、カーブ地点での走行は安定した走行が可能となった。特に直線路でのふらつきなどは発生しておらず、走行への影響が改善していることがわかった。

⑥ 芦別・士別 (2020/2)

ウ) 知能化プラットフォームとの連携確認

前回と同じく、通信連携ができなかった。

移動支援アプリと過去の知能化プラットフォームでは通信連携できていることは確認できた。知能化プラットフォーム側は通信処理の修正対応はしていないため、知能化プラットフォームが動作する PC の環境面での問題の可能性があり、サブテーマ①の担当者に調査を依頼した。

エ) 車両の走行確認

前回と同じく、通信連携できなかったため、Waypoint のデータを手動で渡して、自律走行を検証した。

芦別においては、周回路の走行のみ検証を実施し、問題なく走行できていることを確認した。

士別においては、カーブと直線が連続した一般道路 (冬季閉鎖区間) での走行を検証し、問題なく走行できていることを確認した。

本実験においては速度の自動調整機能に対応した Waypoint を使用しており、カーブ地点の進入手前から段階的に速度を落として、カーブ通過後に段階的に速度を加速している。

ただし、Waypoint に設定する速度はあくまでも走行における指針となるものであり、実際の走行速度は知能化プラットフォームが自己位置や周囲の障害物などを元に制御するため、設定速度通りに走行する訳ではないが、移動支援アプリが設定した速度によって急な加減速や走行が不安定になるなどの影響はないことが確認できた。

③ 実証実験の結果

実証実験の結果から、Waypoint に対する地点補正や走行速度の調整機能によって、各地点における走行位置がずれるなどの影響はなく、カーブから逸脱した走行や、曲がり切れなかったなどの問題も発生しなかった。また、走行中にふらつくなどの不安定な走行になるなどの問題が発生しないことを検証することができた。

2-3-4. 成果

初年度成果:

初年度は以下の研究実施を行った。

- グローバル経路計画連携機能の仕様策定
- グローバル経路計画連携機能の開発
- 実証実験

グローバル経路計画連携機能の目的である waypoint と呼ばれる途中通過点を外部から設定するための API を作成するために、利用可能なオープンソースの調査を行い、waypoint 設定に利用するオープンソースシステムを決定した。また、waypoint 情報をグローバル経路計画側からローカル経路計画側に送信する連携方法を決定した。

決定した利用オープンソースシステム、連携方法を元に機能試作として「グローバル経路計画とローカル経路計画の連携を実現するための連携機能の正常系開発」および「waypoint 設定 API 公開 I/F 仕様の策定と実装を行った。OSM(OpenStreetMap)上で目的地、ルートに対して waypoint を設定する際に、実証実験で利用可能な MAP データが必要とされたため、GPS データと国土地理院の地図より実証実験場の MAP データを作成し、自動運転走行の実証実験の際に利用可能な基本データを作成した。

2年度成果:

グローバル経路計画 (OpenStreetMap) から取得した Waypoint (地点情報) に速度・角度を設定する機能を追加対応したことで、知能化プラットフォームで動作する ROS が自動走行に必要なパラメータを送信して自動運転走行の連携が可能となった。Waypoint に設定する速度はユーザが任意に設定できるようアプリケーション画面に追加した。これにより、本研究での実証実験時にソースコードの修正等を行わなくても、実験時に速度を変えて検証することができるようになった。Waypoint に設定する進入角度を調整するため、*Spline*補間曲線から滑らかな曲線を導出し、その曲線上を通る Waypoint を新たに追加することで、各地点における進入角度を小さくすることができるようになった。これにより、カーブ地点での余計なステアリングの切り返しを抑制し、タイヤのグリップ力 (横滑りへの抵抗力) が増すことで横滑りやオーバーステアの発生を抑制できるようになった。2年度に対応した Waypoint の補正機能は、グローバル経路計画連携機能のミドルウェアとして開発したものであり、初年度開発したアプリケーションに提供するインターフェースは変更していないため、初年度のインターフェース仕様のままで使用することが可能である。

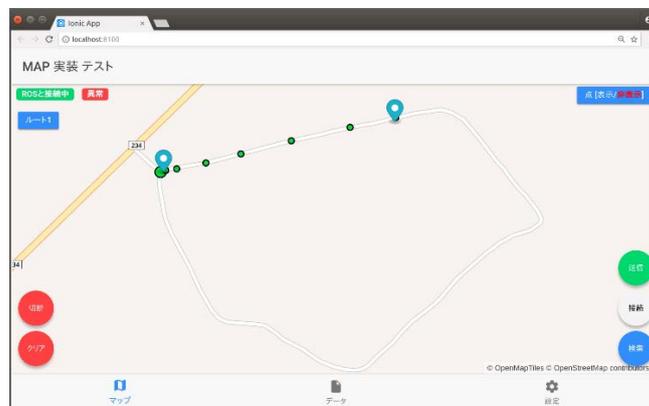


図 2-3-4 (1) グローバル経路計画連携機能のマップ画面

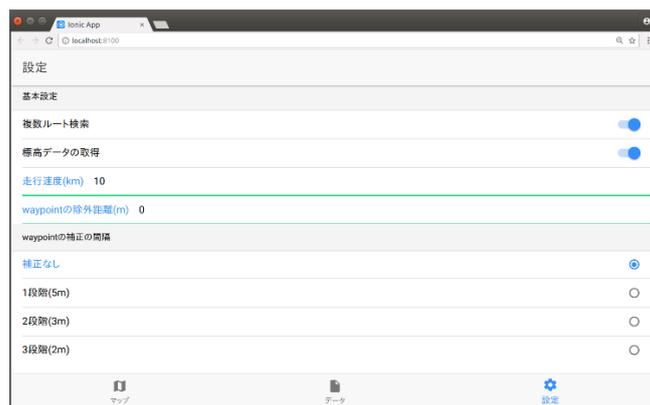


図 2-3-4(2) グローバル経路計画連携機能の設定画面

3年度成果：

エンドユーザを想定して開発した移動支援アプリから走行ルートを決して自律走行を行わせる連携システムを開発した。移動支援アプリは昨年度まで開発した機能と、3年度対応した機能が含まれており、各機能はライブラリとして利用している。そのため、他のアプリケーションやカーナビゲーションシステムにおいても、開発したライブラリを使用することで同様の機能を実現することは可能である。特に、地図サーバとの通信連携や、知能化プラットフォームとの通信連携においては、対抗先である各システムのインターフェースを意識しなくても、ライブラリが提供するインターフェースを利用することで容易に連携することが可能である。

開発したライブラリは地図サーバから地図データを取得するだけでなく、地図画像を取得してアプリケーション上に表示することが可能である。ユーザが地図上をタップして地点を選択するインターフェースも設けており、地図画像の取得や描画処理などもライブラリが提供する。また、地図のルートを設定するのはスマートフォンの操作に慣れたユーザは容易であると思われるが、操作に慣れていないユーザにとっては少し難しいかもしれない。そのため、一度設定したルート情報をスマートフォンに保存・読み込む機能を追加した。これにより、一度保存した走行ルートであれば、再度ルートの設定や検索などを行わなくても、スマートフォンに保存されたルート情報を読み込ませることで同じ走行ルートをすぐに設定することが可能であり、スマートフォンでの簡易な操作を実現した。

3章 全体総括

3-1. 複数年の研究開発成果

本研究は、自動運転に欠かせない周囲状況・自己位置把握技術を、積雪路面にも対応させるものであり、自動運転の高性能化、高機能化に資するものである。北海道を含む寒冷地域の特徴である冬季の積雪環境では、除排雪による道路幅員の大幅な変化などのため、高精度地図データによる自動運転方式では対応が難しい環境である。そこで本研究は、現在の自動運転技術を積雪環境に適用した場合に生じる課題の一つを解決するための新技術確立し、積雪寒冷地での自動運転車を実現に役立てることを目的とし、さらに将来的な自動運転車関連事業の実現を目的に研究開発を実施した。その主機能である「積雪環境で機能する自己位置推定技術開発」では、LiDAR-SALM 技術を基礎として、積雪によって刻々と変化する道路環境に対して、複数のセンサーと AI(雪道対応セマンティック・セグメンテーション技術)を組み合わせた積雪環境向けセンサーフュージョン技術を開発し、積雪環境で機能する自己位置推定技術を開発した。当初計画では対応可能な天候を「晴れまたは曇り」としていたため、冬季の実証実験で降雪となり頻繁に中断してしまったことから、降雪環境にも対応可能となる改善を施した。これにより、厳冬期の乾雪(こな雪)のような比較的小さな雪であれば、一定程度の降雪にも対応が可能となった。

積雪環境仮想シミュレータ開発では、実車による屋外での自動運転試験走行はリスクが高いことに加え、評価期間が冬季に限られてしまうことや、試験環境パターンが制限されるなどで、実車走行試験に膨大な時間と費用を要する問題を解決するツールとして本研究で開発したものである。開発した積雪環境仮想シミュレータは、本研究で実証実験の走行コース2コース(苫東コース、芦別コース)と汎用的な1コースに対応しており、それぞれのコースで道路摩擦係数、障害物有無、積雪状態、雪壁有無等を任意に設定することが可能である。仮想シミュレータで、車両に搭載されている各種センサーの出力値を疑似的に出力されることができ、ネットワークを介して自動運転制御ソフトウェアと接続することで、積雪環境下での実証実験ができない無積雪期であっても、積雪環境仮想シミュレータを利用して自己位置推定技術の動作テスト・検証することを実現した。その結果、実証実験の前に不具合を発見することができ開発の効率化を実現した。また、実証実験の場合に発生する実験場所の手配、実車の運搬などの対応工数や人件費を含むコスト削減にも効果があった。以上のことより、シミュレータの開発目的である「研究開発における生産性向上」に貢献した。

移動支援アプリ(グローバル経路計画連携機能)では、智能化プラットフォームと通信連携を行うことで、搭乗者が走行ルートを決定(ナビ機能)し、指定した区間での自動運転走行をテストコースや一般道で実証した。グローバル経路計画連携機能が提供するインターフェースを使用することで種々のアプリケーション、サービスが智能化プラットフォームと連携して、自動運転に必要な走行ルートの設定を行うことが可能となった。

これらの成果を統合し、実験車両に搭載して、自動運転によりテストコースにて 30km 以上の連続走行、障害物(雪塊)回避、一般道での自立走行を実証し、当初設定した性能目標は概ね達成した。

3-2. 研究開発後の課題と事業化について

自動運転車の販売や関連サービスの事業化実現には、以下のように、技術的課題以外の社会・制度上の課題も多く存在する：

- 自動運転自体の安全性を示し社会に受け入れられること、
- 道路交通法等の関連法規の整備が整い受け入れられること、
- 万が一の事故発生時の責任所在の考え方が社会に受け入れられること、
- 保険等の補償の制度が整うこと、など。

こうした課題が解決してこそ、真の意味で自動運転に関する事業が成功したといえるだろう。そのためには、上記の課題解決をただ待っているだけでなく、自動運転車の利便性や安全性を広く紹介し、社会が受け入れる環境が整うように働きかけていくことが重要である。本研究では、これを事業化戦略の中核となる考え方とし、以下の事業計画を進めていく：

【1. 自動運転車の社会実験支援】

- ① 社会実験用積雪路面自動運転車両の販売（リース）

【2. 自動運転の技術力向上支援】

- ② 積雪環境の自動運転仮想シミュレータ
- ③ 積雪路自動運転制御ソフトウェア

本研究は他に例をみない、雪道での自動運転技術の高度化を目指すものであるため、特に積雪寒冷地域において、こうした活動を展開していく計画である。

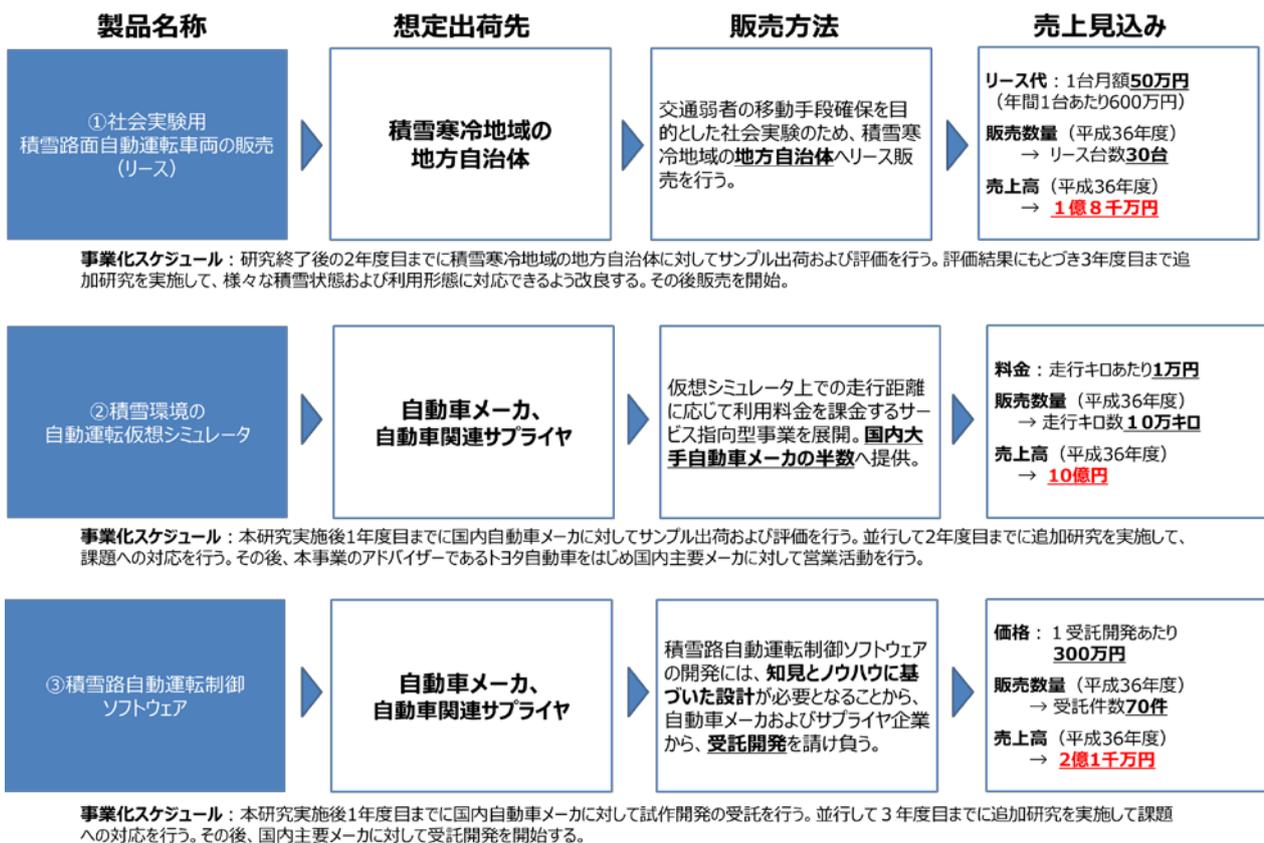


図 3-3. 事業化製品と事業化計画