

学位論文 博士(工学)

車両の自動運転に向けた走行レーン地図と
車載カメラ画像の照合による自己位置推定に
関する研究

2018年2月

慶應義塾大学大学院理工学研究科

原 孝 介

概要

自動車の自動運転や高度安全システムの実現のため、自車が走行するレーンを高い精度で認識する必要がある。しかしながら複雑な交差点や合流分岐を含む市街地や都市高速では認識の難しいシーンがある。このため、レーン形状が細かく記載された走行レーン地図を準備し、画像と地図を照合することで地図上での自己位置を推定してレーン形状を参照する方法が有力である。

本研究では低コストで高精度なレーン認識の実現を目的として、自己位置推定の方法と地図の自動生成法を提案する。

まず自己位置推定の方法として、車載された複数のカメラ画像から検出した線分特徴と地図を照合することで、自己位置を高精度に推定する方法を提案する。自己位置推定の手がかりとなる路面上の白線、黄線、道路標示、縁石など異なる特徴を、2次元路面上の線分特徴として統一的に表現する。さらに線分特徴は車両座標系に鳥瞰変換することで、カメラの構成に依存せず自己位置の推定に用いることができる。

次にレーザ計測された三次元点群から勾配画像を利用して地図を自動生成する方法を提案する。自動運転で使われる走行レーン地図の作成プロセスはモバイルマッピングシステムによるレーザ計測と、人手による図化の二つのステップから構成される。本研究ではコストの大部分を占める人手による図化を自動化するため、レーザ計測された三次元点群を勾配画像に変換し、線分検出器を適用することで高精度な地図を生成する。

本手法の評価として自動抽出された地図と人手による地図を比較したところ、91.9%の精度で83.9%の地図を復元できた。これは人手のコストの約70%を削減したことに相当する。さらに自己位置推定システムは非線形二次計画問題として設計し、リアルタイムに推定が可能である。複雑な都市高速や市街地における走行評価によって自己位置の精度を測定したところ、自動運転の要求よりも高精度な自己位置が得られた。

Abstract

Highly accurate lane recognition is required for realizing autonomous vehicle driving and advanced safety systems. However, there are scenes that are difficult to recognize in urban areas. Therefore, the conventional methods for autonomous driving in cities is to keep the vehicle on a lane by preparing a lane-level map with detailed information of lanes and signs and using accurately estimated vehicle position. The purpose of this research is to improve lane recognition accuracy and cost of map reconstruction.

First, we propose a method of accurately estimating vehicle position by matching the map and line segment features detected from multi-camera images. Features such as white road lines, yellow road lines, road signs and curb stones, which could be used as clues for vehicle localization, were expressed as line segment features on a two-dimensional road plane in unified manner. The line segment features are transformed into the vehicle coordinate system using bird's-eye view transformation and can be used for vehicle localization regardless of the camera configuration.

Further, we propose an automatic method for map synthesis from laser point clouds that are scanned by a Mobile Mapping System (MMS). In this method, we apply a gradient image processing to detect map line segments.

Results of the comparison of auto-extracted map and handmade map for the evaluation achieved over 83.9% extraction rate with an accuracy of 91.9%. It is equivalent to reducing 70% of the cost when human made the map. The proposed localization system is designed as a nonlinear quadratic programming problem and is estimated in real time. Vehicle localization was tested under for city driving conditions, and the vehicle position was identified more accurately than the requirement for autonomous driving.

目次

概要	3
ABSTRACT	4
1. はじめに	9
1.1. 自動運転の実用化に向けた取り組み	9
1.2. 代表的な自動運転の実験	10
1.3. 自動運転のための環境認識	12
1.4. 自動運転のためのレーン認識	13
1.4.1. レーン認識に求められる精度	13
1.4.2. 都市高速の難しいシーン	14
1.4.3. 市街地の難しいシーン	15
1.5. 走行レーン地図と自己位置推定によるレーン認識	17
1.6. 走行レーン地図の整備	19
1.6.1. 自己位置推定のための地図	19
1.6.2. 地図作成のプロセス	19
1.6.3. 地図整備のコストの課題	20
1.6.4. 自己位置推定のための特徴の不足の課題	23
1.7. 自己位置の推定	24
1.7.1. 自己位置推定の方法	24
1.7.2. 自己位置推定に適した車載センサ	24
1.7.3. 自己位置推定に利用できるカメラ	25
1.7.4. マルチカメラの利用の課題	26
1.7.5. 道路画像からの特徴抽出の課題	26
1.8. 地図整備と自己位置推定の課題と分析	27
1.9. 本研究の提案	28
1.9.1. 地図の曖昧さのモデル化	29
1.9.2. マルチカメラ構成と線分特徴による自己位置推定	29
1.9.3. 地図の自動図化	29
1.10. 本論文の構成	30
2. 関連研究	31
2.1. GNSS による自己位置推定	31
2.2. レーザスキャナによる自己位置推定	32
2.3. カメラ画像による自己位置推定	33
2.3.1. 衛星画像/空撮画像との照合	33

2.3.2.	過去に撮影された画像特徴との照合	34
2.3.3.	ベクトル地図との照合	35
2.4.	走行レーン地図の生成	36
2.4.1.	一般車両のセンサ情報を利用する方法	36
2.4.2.	MMSによって収集されたカメラ画像やレーザ点群を利用する方法... ..	37
2.5.	関連研究のまとめと本研究の位置付け	37
3.	車両の運動モデルと自己位置推定	39
3.1.	路面地図上での自己位置の表現	39
3.1.1.	地図の道路勾配を使った車両姿勢の分解	39
3.1.2.	自車近傍の路面勾配の計算	40
3.2.	車両の運動モデル	42
3.2.1.	共分散行列の設計	43
3.2.2.	ヤコビ行列の計算	43
3.3.	拡張カルマンフィルタによる自己位置の推定	46
3.4.	非線形最小二乗問題による自己軌跡の推定	46
4.	線分特徴による観測モデル	48
4.1.	車載カメラ画像におけるコーナー点とエッジ線	48
4.2.	マルチカメラによる線分の観測と自己位置推定	48
4.3.	線分地図	49
4.4.	運動モデルによる自己位置の予測	50
4.5.	画像中の線分の鳥瞰変換	51
4.5.1.	鳥瞰変換の誤差モデル	52
4.5.2.	鳥瞰変換の結果	53
4.6.	地図線分と画像線分の対応付け	55
4.7.	自己位置の誤差モデル	56
4.8.	自己位置に対する観測誤差と線形化	57
4.9.	自己位置推定での利用	60
5.	曖昧さを含む地図の観測モデル	61
5.1.	白線種別と白線形状の関係の調査	61
5.2.	白線断面の観測方法	62
5.3.	白線の観測による誤差モデル	64
5.4.	自己位置推定での利用	64
6.	走行レーン地図の自動生成	65
6.1.	地図の自動生成の課題	65

6.2.	勾配画像への変換と線分抽出による地図生成.....	66
6.3.	MMSによる路面計測の詳細.....	68
6.4.	路面点群の抽出.....	70
6.5.	反射強度の勾配画像の生成.....	73
6.6.	地図線分の検出と高精度化.....	75
6.7.	縁石検出.....	77
6.8.	処理領域の分割と結合.....	80
6.8.1.	処理領域の分割.....	80
6.8.2.	重複領域の線分の処理.....	80
6.9.	縁石外領域の白線除去.....	83
7.	自動生成した走行レーン地図の評価.....	85
7.1.	人手による地図との比較.....	85
7.1.1.	評価経路.....	85
7.1.2.	評価基準.....	87
7.1.3.	評価結果.....	87
7.1.4.	地図生成の具体例.....	88
7.2.	考察.....	88
8.	自己位置推定の評価.....	91
8.1.	市街地での自己位置推定の評価.....	91
8.1.1.	評価車両.....	91
8.1.2.	評価経路.....	92
8.1.3.	走行レーン地図の準備.....	93
8.1.4.	比較手法.....	94
8.1.5.	評価基準.....	95
8.1.6.	自己位置の評価結果.....	96
8.1.7.	各経路の詳細.....	97
8.1.8.	計算時間の評価.....	100
8.1.9.	考察.....	100
8.2.	自動生成地図による自己位置の評価.....	101
8.2.1.	人手地図と自動生成地図の自己位置の評価.....	101
8.2.2.	考察.....	101
8.3.	都市高速での自己位置の評価.....	103
8.3.1.	評価車両.....	103
8.3.2.	評価経路.....	103
8.3.3.	評価真値の作成方法.....	103
8.3.4.	自己位置の評価結果.....	105

8.3.5. 計算時間の評価.....	106
8.3.6. 考察.....	106
9. 結論.....	107
9.1. 総括.....	107
9.2. 今後の展望.....	108
謝辞.....	110
参考文献.....	111

1. はじめに

1.1. 自動運転の実用化に向けた取り組み

自動車の運転は、道路や他車両、歩行者、自転車などを知覚し、交通法規に従って自車の経路を決め、ステアリングやアクセル、ブレーキを操作するといった、人にとっても教習が必要な難しい行動である。このため必ずしも正しい運転行動ができるとは限らず、交通事故により世界で約 124 万人が死亡し 2,000~5,000 万人が負傷している[11][24]。特に高齢者は自動車・歩行者間の交通事故を起こしやすい傾向があり、高齢化社会の進行に伴ってその防止が求められている[57]。このための技術として人の運転操作を補助するための先進運転支援システム(ADAS: Advanced Driver Assistant System)や、その集大成である自動運転が注目されている。

特に近年になって自動運転を実用化するための研究が急激に進められている。この背景には技術的、および法的な側面が考えられる。まず技術的には人の運転行動が機械によって代替可能になってきたことが挙げられる。運転行動を認知・判断・操作に分類するなら、認知はカメラやミリ波レーダ、レーザスキャナ、IMU(Inertial Measurement Unit)、GNSS(Global Navigation Satellite System)といったセンサ[69]で代替され、判断は高性能な計算機を使った信号処理で可能になりつつあり、運転操作は ECU(Electronic Control Unit)による電子制御[48]で実現される。次に法的な側面については、米国ネバダ州で自動運転車の公道走行実験の許可する法律ができたのを皮切りに、現在では危険時にドライバが操作介入を行い、安全の維持とその責任を負う自動運転であれば現行法で実現できるとされている。特に国内では公道での自動運転の実験に特別な認可や車両ナンバーの取得も必要ないとされており[64]、実際に通常の手続きでナンバープレートを取得した自動運転車両が公道での実証実験やデモ走行を行っている[55] [64]。このためメーカ各社にとって自動運転は現実的な製品として捉えられ、急激に研究開発が進められている。これらの研究の多くは自動化の段階の定義(表 1.1)[62]におけるレベル 3~4 を目標としている。この結果、先駆的な自動化レベル 2 相当の製品[10]や、国内においても 2020 年の東京オリンピック・パラリンピックに向けて自動化レベル 3 の実現を目標設定が発表されている[62]。しかしながら即座に人の認知・判断に匹敵するセンサ・信号処理システムの開発は困難であり、現実的な解を模索している段階である。

表 1.1 自動化レベル及びそれを実現する自動走行・運転支援システムの定義[62]

自動運転レベル	概要	左記を実現するシステム	
レベル1	加速・操舵・制動のいずれかをシステムが行う状態	安全運転支援システム	
レベル2	加速・操舵・制動のうち複数の操作をシステムが行う状態	準自動走行システム	自動走行システム
レベル3	加速・操舵・制動の全てをシステムが行い、システムが要請したときはドライバが対応する状態		
レベル4	加速・操舵・制動を全てドライバ以外が行い、ドライバが全く関与しない状態	完全自動走行システム	

1.2. 代表的な自動運転の実験

このような自動運転研究の火付け役となったのは DARPA が主催した DARPA Urban Challenge(2007)[28][63]である。この実験では市街地を想定した 96[km]の道路を 6 チームが完走した。この研究を引き継ぎ、2010 年から Google は自動運転車の市街地の公道での試験を行っており、総走行距離が 190 万[km]を超えている[70]。この実験は非常に大きな実績を示す一方で、従来の自動車のセンサシステムと大きくかけ離れた点がある。その最も特徴的なのが自動運転車の屋根に取り付けられたレーザスキャナ Velodyne 社 HDL64e[36]である(図 1.1)。これは DARPA Urban Challenge において完走した 6 チーム中 5 チームで装備されており[28]、その技術を発展させたものと考えられる。その理由として、市街地においては歩行者と車両が混在し、またその走行可能領域を表すレーンも複雑であるため、高精度かつロバストな歩行者や障害物、レーンの認識が要求されることが挙げられる。これに対応するには従来から車載されているカメラや低密度なレーザスキャナを使った認識では不足であったものと推察される。しかしながら従来の車両開発の視点からはいくつかの課題がある。まず HDL64e は 1 台あたりの価格が 7 万ドルとされ、一般的な車両の価格を大きく越える。次に 64 個のレーザ距離計を搭載した大型のセンサヘッドを 10[Hz]で回転させるため、一般に 10 年が耐用年数とされる自動車においては故障発生が課題となる。さらに天井に大型のセンサを搭載する意匠には強い抵抗がある上、車高が高くなるため高架下など天井が低いところに進入できなくなる。さら雨滴等の汚れに対するメンテナンス性にも課題がある。一般的に PCS(Pre-Crash Safety system)[65]や LKA(Lane Keeping Assist system)[61]向けのカメラはフロントガラス裏のルームミラーの付近に取り付けられるが、これはフロン

トガラスに付着した汚れをワイパーで拭き取ることができるためである。天井につけたセンサではこのようなメンテナンスはできない。

一方で従来の自動車の開発を発展させた自動運転の代表的な研究に、Daimler の Bertha drive[42]が挙げられる。Bertha drive では既成部品に近いスペックのセンサのみで、市街地 103[km]の自動運転が行われた。具体的なセンサの構成として、ステレオを含むカメラと、ロングレンジ、およびショートレンジのミリ波レーダを複数台搭載している。現在の製品の自然な延長であるため、市街地での実績のある構成では最も量産に近い設計と考えられる。

本研究も同様に量産に近い安価な構成で、複雑な市街地や都市高速での自動運転に役立つ技術の開発を狙いとし、特に環境認識技術に焦点をあてるものである。

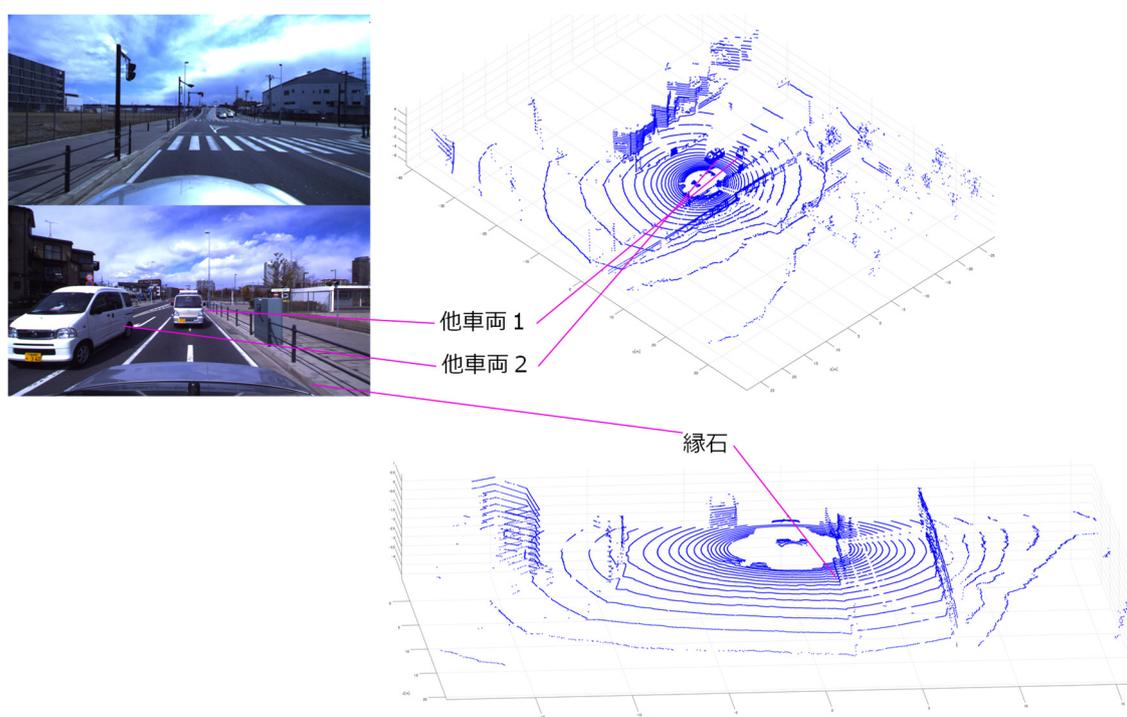


図 1.1 Velodyne 社 HDL32e によって得られるレーザ点群 (前方・後方カメラ画像(左)と、HDL32e のセンシング結果(右)を示す。他車両や縁石などが明瞭に識別できる。なお HDL32e は HDL64e の半分の解像度である。)

1.3. 自動運転のための環境認識

自動運転の研究における環境認識には、一般にカメラやレーザスキャナ、ミリ波レーダが用いられてきた。これらが検出する対象はダイナミックマップ[49]の分類を参考にすると、1秒から1時間という短い時間で変化する動的情報と、1ヶ月といった長い時間で変化する静的情報に分けて考えることができる。

動的情報には周辺車両や歩行者、落下物などが挙げられる。これらの認識には、古くから障害物が衝突を回避するための自動ブレーキ PCS を目的として、多くの研究[14][25][29]がある。その多くは撮影したカメラ画像やレーザスキャナといったセンサ信号を、事前に学習した識別器にあてはめて、障害物の有無を判定する。しかしながら、市街地において自動運転に耐えうる精度とロバスト性を保証するのは難しい問題である。Bertha Drive においては実績があるものの、十分に準備されたデモ用の経路の走行であり、自由な経路での自動運転は実現されていない。カメラ画像から歩行者を検出し、衝突が予測される際には車両を緊急停止させる装置が市販されているが、確実に動作するものではない。近年、ブレークスルーのあった Deep Learning を用いた歩行者認識の研究[5]を見ても、性能向上は見られるものの自動運転に求められる「絶対ぶつからない」精度を保証しうるものではない。現状において唯一実績があるのは、Google の自動運転が用いている精度、解像度ともに優れた高性能レーザスキャナを搭載する方法である。しかしながら前述のような価格や故障、意匠、メンテナンス性といった課題がある。高解像度なレーザスキャナを可動部なしに小型・安価に実現する研究[22]もあるものの、即座に量産しうるものではないと考えられる。これらの課題は市街地において自動運転レベル 3 に耐えうるものではなく、各社が市街地を避けて自動車専用道路における自動運転を目標としている要因のひとつと考えられる。

一方で静的情報には白線や路肩縁、またそれらを統合した走行可能領域や車線中心、道路標示や交通標識などが挙げられ、自車が走るべきレーンや交通規則を認識する。古くから両側の白線に対する逸脱防止を目的とした LKA や、ドライバへの注意喚起を目的として研究されてきた。しかしながら、動的情報の認識と同様に、センサ情報からの認識には限界があり、「絶対間違えない」精度の実現は不可能である。

本研究では複雑な市街地や都市高速における走行レーンの認識を扱う。次にその要求や達成の難しいシーンについて詳細を説明する。

1.4. 自動運転のためのレーン認識

1.4.1. レーン認識に求められる精度

レーン認識では、車両を中心とした周囲のレーン形状を、カメラやレーザスキャナなどのセンサ情報から認識する。この認識を誤ると、レーンからはみ出すような走行となり事故につながる。このため自動運転にとって非常に重要であり、その要求精度は高い水準が求められる。ここでは要求される認識精度や認識距離、フレームレートを検討する。

まず認識精度について、市街地での実績にある Bertha drive の論文[43]では、経路の約 6%の横方向のクリアランスが 0.35[m]以下であったことから、制御器とレーン認識を結合した誤差を 0.2[m]以下に抑える必要があったと報告している。また自動駐車論文[68]においても、目標を 0.2[m]として設計されている。良好な路面を想定して制御器の誤差が小さいと仮定すると、自車周辺のレーン認識における要求精度は 0.2[m]と考えることができる。

次に認識距離については、車両が走行する予定の経路を、どの距離まで計画するかにより依存する。もし遠方までのレーン形状が不明であるなら、レーン形状が変化した際に余裕がなくなり急旋回、急減速が発生し、危険や快適性の問題がでてしまう。具体的な認識距離の参考として、自動運転のための経路計画に使われるモデル予測制御を参考にする[40][44]。例えば[44]では経路を計画する未来の時間幅を(予測ホライズン)を 10[s]としている。仮に車両の速度を 10[m/s](≈ 40 [km/h])とすると 100[m]に相当する。つまり自動運転のためには 100[m]先までの道路を認識しなければならない。

自動運転に使うレーン認識は、リアルタイムに動作する必要がある。一般に LKA のレーン認識や、プリクラッシュセーフティに使われるカメラ画像は 10[fps]で処理される。このため 10[fps]=100[ms]で処理できることが目標となる。

まとめると 100[m]先までの道路形状を、自車周辺では 0.2[m]の精度で、100[ms]の処理時間以内に検出する必要がある(表 1.2)。単純な道路であれば既存の画像処理を駆使することで可能となることもあるが、本研究が対象とする市街地や都市高速では難しいシーンが多く存在する。これらについて次に紹介する。

表 1.2 本研究が扱うレーン認識への要求

内容	要求
道路形状の認識距離	100[m]
自車周辺の横方向の認識精度	0.2[m]
処理時間	100[ms]

1.4.2.都市高速の難しいシーン

前述のように自動車専用道における自動運転は、実現性の高い製品として期待されている。一方で、自動車専用道路であってもレーンの認識の難しいシーンがあり、特に都市高速では顕著である。国内において特に難しいとされているのが首都高速である。図 1.2 は首都高速都心環状線(C1)で撮影された画像である。まず合流・分岐では複数のレーンを認識しなくてはならず、推定変数が増えることに加えて、自車線と離れているため難しいことがある。白線付近にある減速標示などの道路標示を誤って認識するがあり、また補修痕は白線を強調するフィルタで一緒に強調され検出されることが多い。白線はかすれて不明瞭なことがある。縦断勾配や急カーブでは、遠方までの認識が難しくなる。トンネル出入口では、カメラのゲインや絞りの調整が間に合わず、黒飛び白飛びがある。このように画像処理による認識が難しいシーンは多くあり、自動運転の要求に応えられるとは限らない。

合流・分岐



複雑な道路標示



かすれた白線, 補修痕



縦断勾配



急カーブ



トンネル出入口



図 1.2 都市高速におけるレーン認識が難しいシーン

1.4.3.市街地の難しいシーン

市街地にはさらに難しいシーンが存在する(図 1.3). まずレーン構成や道路標示が複雑であることが多い. このような状況で, 仮にレーンごとに道路を分解できたとしても, 停止線や白点線, 止まれ, 横断歩道など, 交通法規に照らし意味を解釈し, 制御に反映する必要があり非常に難しい課題となる. また, 広い交差点では遠方におけるセンサの解像度が足りないことがある. 市街地においては交差点などでレーン形状が複雑になることが多い. 例えば白線認識の研究[21]では白線をクロソイド曲線で近似した 3 次多項式で表現しているが, 100[m]の範囲を単一の 3 次多項式に当てはめるのは難しく, より変数の多いモデルを使わざるを得ない. この結果として精度やロバスト性を落ちる課題がある. 道路端が路駐車両等による遮蔽によって画像に写っていないことがある. 人は路駐車両がある先がどのようになっているか, 既存の道路の知識に照らして推測することができるが, まだ機械にはできない. また, 白線検出において頻出する問題として, かすれ, 急カーブ, 急勾配, 補修痕が上げられる. かすれは路面のテクスチャに対して S/N が悪くなるため検出が難しくなる. 急カーブ・急勾配は白線に仮定したモデルが合わなくなることが多い. 補修痕は白線を強調するために用いられるフィルタで誤って強調されることが多い. 影も同様にエッジ抽出で検出されやすい. また電信柱, 電力柱, 街路灯などの細い路側物は, レーザやステレオで距離を測定しようとしたとき, 解像度が問題になり検出が難しい. さらに歩道にはバリアフリーの観点からスロープがある. スロープはレーザを使った縁石の高さによる道路境界の判定が難しくなる. これらのレーンを構成する各要素を認識できたとしても, それを交通法規や道路標示に照らし目的地までの経路と統合して, 自車の制御に反映する必要があり. これらの不確実なセンサ情報を統合し, 意思決定を行う仕組みは存在しない.

交差点：複雑なレーン形状・道路標示，交通法規の解釈，遠方解像度の不足



オクルージョン



かすれ



補修痕

狭い車線

張り出したポール



急カーブ



勾配後の交差点 スロープ



影



図 1.3 市街地におけるレーン認識が難しいシーン

1.5. 走行レーン地図と自己位置推定によるレーン認識

前節でカメラやレーザスキャナといった車載センサによる環境認識にとって、自動運転に必要なレーン認識が難しいことを説明した。このような白線や路肩縁などの個々のレーン構成要素を認識する課題を避けるため、自動運転に利用されてきたのは自己位置推定と地図による方法である。具体的にはカーナビに使われるより遥かに情報の多い地図を事前に準備し、推定した自己位置を使って地図を参照することで認識の代用とするものである(図 1.4)。つまり白線や縁石などのレーン形状や走行可能領域、交通標識など、工事が行われないうちに変化しない静的情報は全て地図に持っておき、高精度に推定した自己位置を使って、自車周辺の情報を得る。自己位置の推定には例えばカメラから抽出した白線や横断歩道などの手がかりを地図と照合し、一致する自己位置を探索する。地図上に走行可能な領域や、道路標示、交通法規の解釈の結果は、経路計画や制御にとって扱い易い形で事前にデータベース化しておく。自己位置推定はカメラやレーザスキャナなどのセンサ情報と地図を照合し、もっとも良く一致したところを正しい自己位置とする方法である。照合に使う特徴を増やすことで、市街地においてもロバストに動作することが多い。Bertha drive に加え、具体的な内容は公開されていないものの Google の自動運転車両も同様の方式とされており、ほぼ全ての市街地の自動運転の実験はこの方式に基づいている。

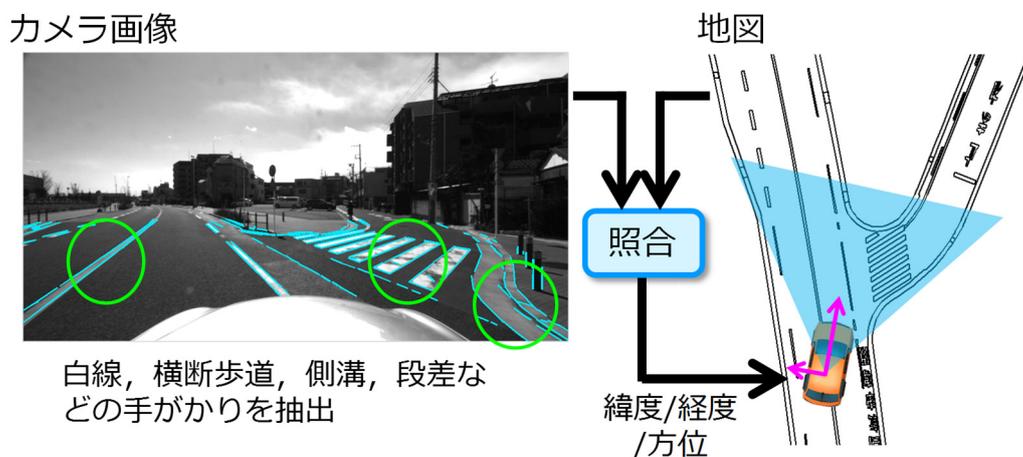


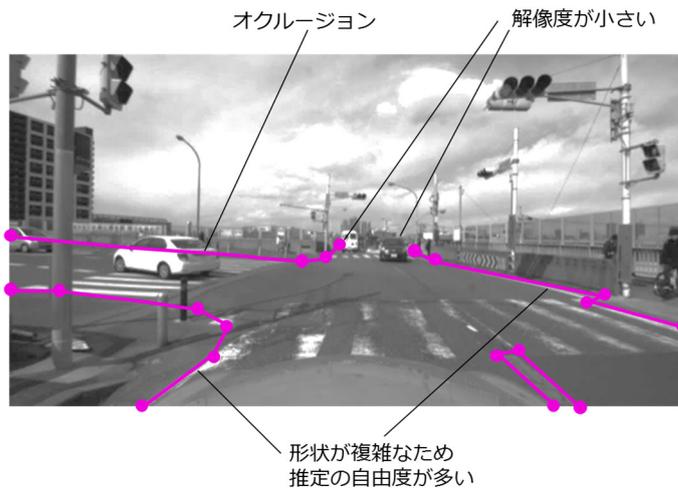
図 1.4 地図と自己位置推定によるレーン認識

自己位置推定によるレーン認識は、レーン形状の直接認識と比較すると、高精度・ロバスト性の観点から有利な点がある(図 1.5)。まずレーン形状の直接認識では遠方の白線や複雑な道路形状を直接認識しなくてはならないのに対し、自己位置推定では近縁の白線や、縁石、建造物、ポールなど大型の物体を手がかりにできる。次に直接認

識では全ての道路構成要素を個々に認識する必要があるのに対し、自己位置推定では複数の構成要素を統合することで自己位置のみを推定すればよく、ロバスト化・高精度化を実現しやすい。これらのことからレーンの直接認識に比べると、複雑な環境である市街地においてもロバスト化・高精度化が実現しやすいと考えられる。

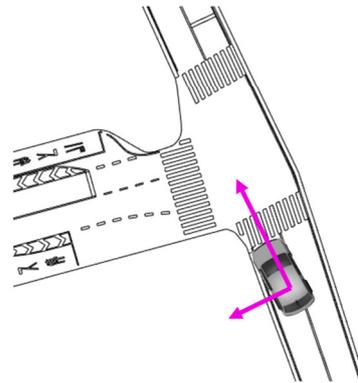
本研究では地図と自己位置推定による方法を、自動運転における有力なレーン認識方法と捉え、次節より地図と自己位置推定の詳細を分析する。

レーン形状の直接認識



自己位置推定

あらゆる特徴を抽出し利用



車両の位置・方位のわずか3自由度を推定
レーン形状は推定位置を使って地図を参照

図 1.5 レーンの直接認識と自己位置推定の違い

1.6. 走行レーン地図の整備

1.6.1. 自己位置推定のための地図

レーン認識に必要な走行レーン地図は、主に測量メーカーによって開発が進められている。国内においても、自動運转向けの地図の開発が進められている[59]。これらは従来からあるカーナビに使われる地図の延長として取り組まれている。しかしながらカーナビが道路を単純な線として表現するのみであったのに対し、走行レーン地図では白線や縁石などの詳細を記載する。このため記載情報が増えることで、コストが大きく上がっている。さらに自動運転を目的とした地図は、正確に記載されていなければ事故につながる危険がある。このため白線の塗り直しなどの工事や地震の度に頻繁に更新する必要がある、継続的にメンテナンスコストが発生する。このため地図の作成におけるコストの低減は重要と考えられる。

なお地図に記載する情報は、自己位置の推定に使う情報と、経路計画などにレーンに記載されている情報が、必ずしも一致している必要はない。しかしながら記載情報が増えることはコストがかかるため、同一の情報で自己位置推定も経路計画もできることが望ましいと考えられ、本研究も同様に位置付けられる。

1.6.2. 地図作成のプロセス

地図を作成するプロセスは、実際の現場をセンシングし形状や色の情報を集める測量ステップと、集めた情報から線分やスプラインによって表現される図を起こす図化ステップの二つに分けることができる(図 1.6)。

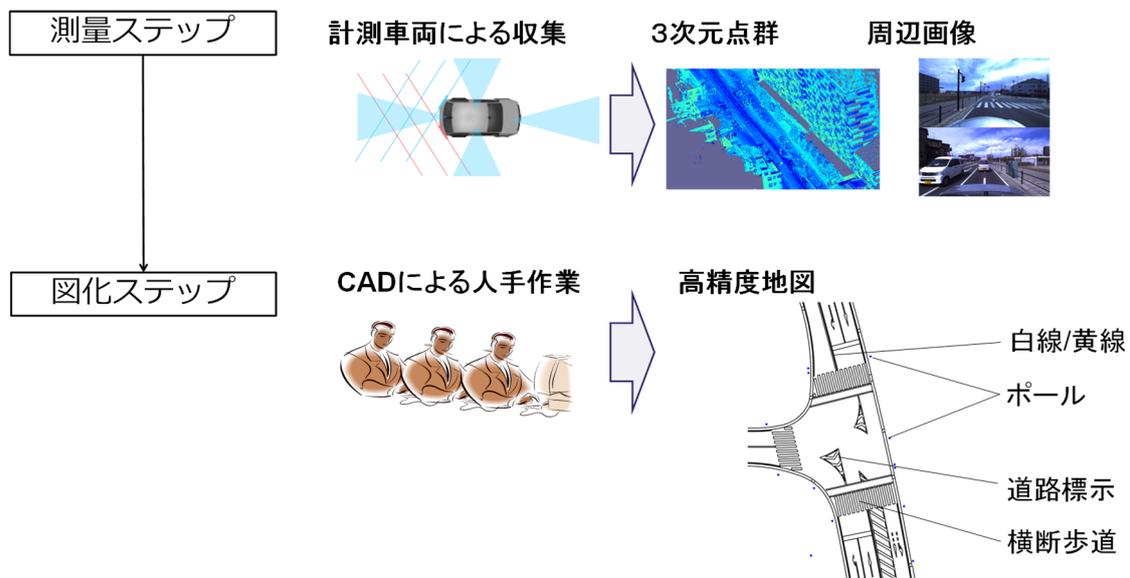


図 1.6 地図作成のプロセス

測量ステップにおいては、近年になって Mobile Mapping System(MMS)[8][46]による道路計測が主流になってきている。MMS は高精度な測位装置とレーザスキャナ、カメラが積まれた車両であり、一日に数十 km もの道路周辺の高密度な 3 次元点群を収集することができる(図 1.7)。測位装置には、既知点からの補正観測情報を利用した RTK-GPS(Real Time Kinematic GPS)[54]と、高精度な IMU(Inertial Measurement Unit)[60]を用いることで、リアルタイムに数[cm]程度の誤差での精密な測位が行われる。またレーザスキャナ[73]には速い周期で高密度に点群を収集できる機種が用いられ、自車線周辺では数[mm]の解像度を持っている。一般に車両の後方に鉛直方向に向けてレーザスキャナを搭載しており、走行することで道路を輪切りにするように 3 次元点群を取得できる。レーザスキャナから得られる 3 次元点群には、各点の 3 次元位置に加えて、レーザスキャナから取得される反射強度の情報が付加されており、白線などの道路ペイントを識別することができる。

一方で図化のステップでは、人が CAD ツール上で点群の反射強度や形状を見ながら書き入れるのが主流である。従来のカーナビ向けの地図であれば、書き入れる線分の数が少ないため全国を整備することができた。しかしながら自動運転やそれに必要な自己位置推定に用いられる地図は必要な情報が多い。特に自己位置推定においては、手がかりとなる情報が多いほど精度やロバスト性を向上させられる。このため MMS の計測コストに比べると、図化作業のコストがより大きな課題であった。

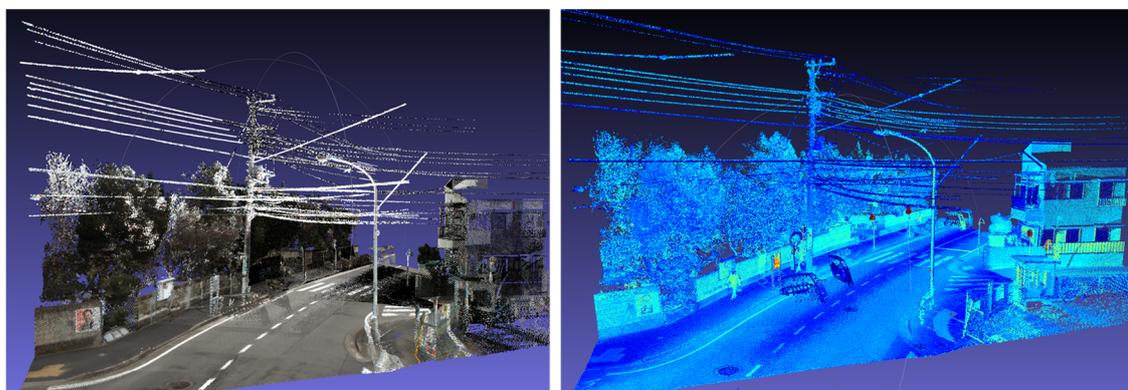


図 1.7 MMS によって収集された高密度 3 次元点群

(右図は点群にカラーカメラによる着色をした。左図はレーザスキャナで取得された反射強度により着色した)

1.6.3.地図整備のコストの課題

前節で地図作成のうち図化作業のコストが大きな課題であることを述べた。これをそのまま日本国内の地図整備に適用することを考えてみる。まず日本国内の道路は全

体で 127 万 km あり，自動運転で初めに対象となる高速自動車国道は 9 千 km，一般国道は 7 万 km である[50]．127 万 km もの詳細なレーン形状を人海戦術で整備するには多大なコストを要する．現状の方法のまま国内全ての道路の走行レーン地図を整備するのは非現実的であり破綻する．特に白線，道路標示，構造物は補修や改修のための道路工事によって更新される．例えば本研究で用いた新川崎駅周辺の道路では，約 3 年間のうちに距離にして約 27%が変化していた(図 1.8)．多くは道路標示や白線の塗り直しであるが，一部は新しい建物がたつなど道路構造も変わっている．このような道路の更新があった場合，自動運転のためには地図を更新する必要があり，メンテナンスコストの発生が課題となる．

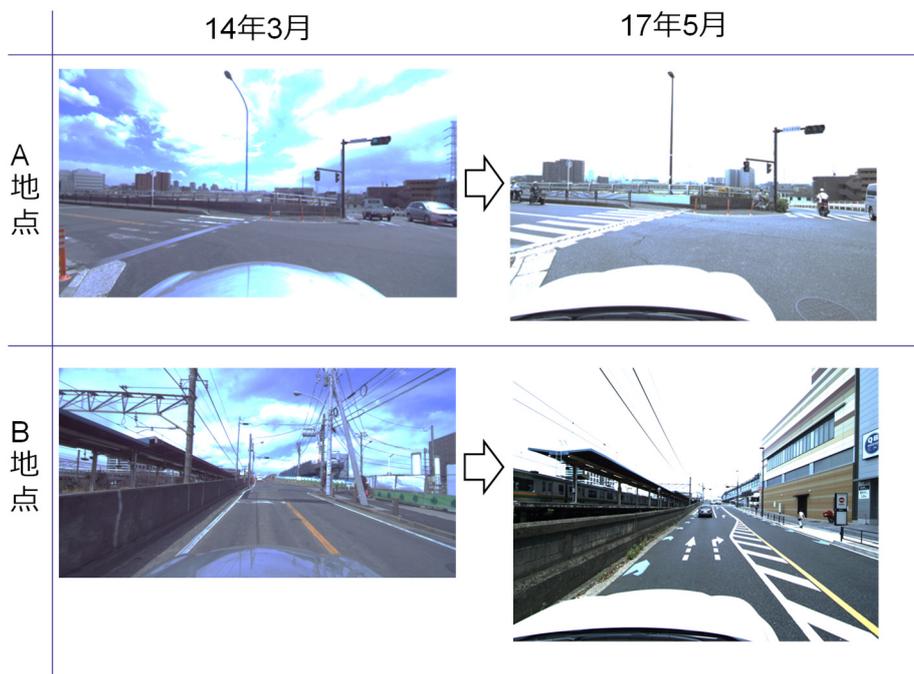
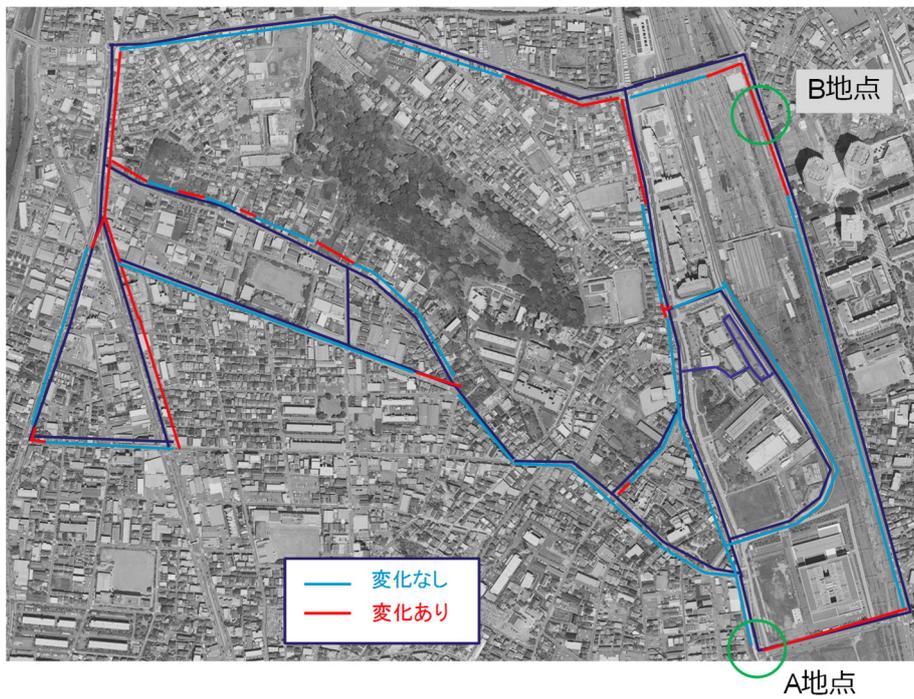
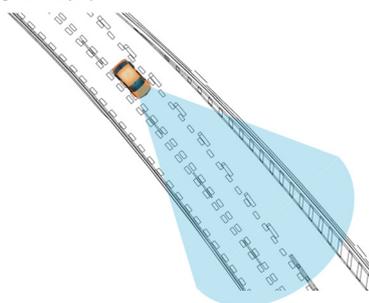


図 1.8 道路工事による地図の経年変化

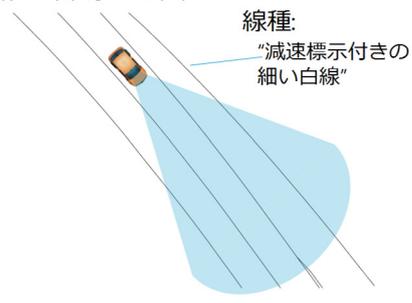
1.6.4.自己位置推定のための特徴の不足の課題

自己位置推定のために地図とセンサ情報を対応付けるには、センサ上での見えと地図上での見えが一致していると、照合が容易になる。具体的にカメラ画像を扱う場合で考えると、画像上での見えと一致するような仕様で、地図の白線が記述されていることが好ましい。しかしながらコストやデータ容量の制約から、形状を簡略化した白線種別が割り当てられていることがある。具体的な理想的な地図と、現実的な整備の仕様のもとでの地図を示す(図 1.9)。理想的な地図では白線と路面の境界を表すエッジが含まれており、自己位置を決める手がかりとしやすい。一方で現実的な地図は形状が白線種別に簡略化されている。白線種別を表すラベルには「実線」「点線」「細い」「太い」「白色」「黄色」, 「減速標示」とその位置, 「2重線」「3重線」といった表現が用いられる。この白線種別の設計が白線形状を正確に決める仕様になっていることが望ましいが、往々にして曖昧さを含む。このため白線種別から想定した見えと、実際の画像上での見えの違いが自己位置推定の誤差の原因となることがある。

白線形状を詳細に記載した理想的な地図



線種別のみを記載した整備の容易な地図



同位置の参考画像



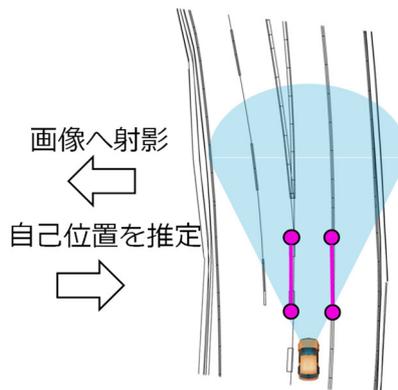
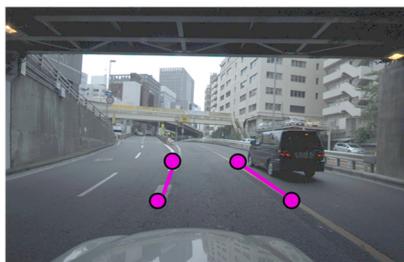
図 1.9 理想的な地図と整備の容易な地図

1.7.自己位置の推定

1.7.1.自己位置推定の方法

自己位置推定では，車載されたセンサの情報と地図を照合し，一致するように自己位置を補正する．本研究で扱うカメラ画像を例に説明する(図 1.10)．正しい自己位置を使って，地図を画像に射影すれば，画像上と地図は重畳される．しかしながら，誤った自己位置では，地図と画像は重畳されない．これを誤差として自己位置を推定する．

正しい自己位置



誤って推定された自己位置

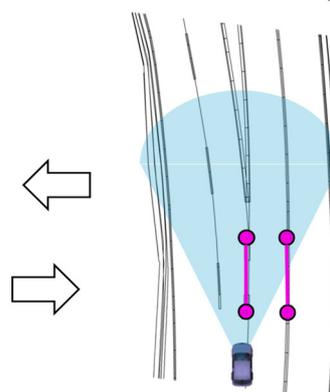
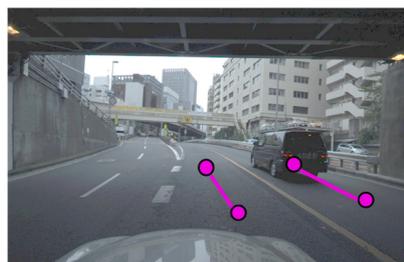


図 1.10 自己位置推定の方法

1.7.2.自己位置推定に適した車載センサ

現在車載されている，または車載可能なセンサをまとめ(表 1.3) ，自己位置推定に適したセンサを検討する．まず環境を観測できるセンサには，カメラとレーザスキャナ，ミリ波レーダがある．まずミリ波レーダは走行レーン地図と対応が悪く，照合には使えない．このためレーザスキャナとカメラが自己位置推定に使われる．カメラは白線や道路標示，横断歩道のようなテクスチャに基づく対象の検出が得意である一方で，直接に距離の取得ができないため縁石等の検出が難しい．一方でレーザスキャナは距離を直接取得できるため縁石の検出が得意であるものの，白線の検出は反射強度の S/N が悪く遠方までの検出が難しい．またカメラに比べると解像度が悪い．さらに車載に

においては一般にレーザの走査にポリゴンミラーを使うため可動部が必要となり、価格や故障率の課題がある。これらを踏まえると、縁石に比べると白線は情報が多く、また搭載性も良いことから、カメラが適していると考えられる。

他のセンサとして高精度 GNSS は要求を満たすものの、高価であることに加えて都市部での安定性に欠ける。一般の GNSS にカーナビのマップマッチングを組み合わせても数 m 程度の誤差が出るため要求を満たす精度は出ない。

カメラが環境条件によって常に利用できるとは限らないのに対し、IMU や車輪速などは環境条件への依存が少なく安定した出力が得られる。このようなデバイスは定常出力デバイスと呼ばれている。これらのセンサを利用することで、よりロバストな推定が可能になる。

これより本研究ではカメラを中心に検討し、補助的なセンサとして車輪速を用いる。

表 1.3 自己位置推定に関する車載センサ

	定常出力	絶対位置	相対精度	環境対応	その他
レーザ スキャナ	×	－	○	霧, 雪, 前景物 (車両, 歩行者など) で精度が下がる	細街路で精度が良い
カメラ	×	－	○	夜, 逆光霧, 雪, 前景物で精度が下がる	細街路で精度が良い
ミリ波レーダ	×	－	△	反射物が無いことがある	地図が無い
高精度GNSS	×	○	－	市街地, 地下, トンネルで受信できない	高価である
ナビ+GNSS	×	△	－	市街地, 地下, トンネルで受信できない	
IMU, 車輪速, ステア角	○	－	△	車輪速はタイヤ径の変化や路面の影響を受ける	積分誤差がたまる

1.7.3. 自己位置推定に利用できるカメラ

最近の市販されている車両の多くには、カメラが搭載されている。LKA や PCS を目的としてルームミラー付近に高解像度のカメラを取り付ける事が多い。さらに駐車時の支援を目的としたアラウンドビューモニタ[58]やバックガイドモニタのため、サイドミラーや車両後方にカメラが取り付けられる。これらのカメラは既存製品の延長として扱いやすい利点がある。このように車両が進行できる方向である前後のカメラは歩行者検出等の用途に用いられるため、高解像度のカメラが取り付けられることが多く、画像処理に適している。

マルチカメラの利用について Pless らの研究[26]によると、一般化エピポーラ幾何[7]によって表現されるエッセンシャル行列の分析から、複数のカメラを用いる場合、対

向した位置に取り付けると精度が良いことが報告されている。SLAM の研究において視野の広いカメラを利用することはその姿勢推定に有利であり、多くの研究で推奨されている[15]。道路環境においても広い交差点を曲がるシーンでは前方に手がかりが少ないことがあり、視野を広げることは高精度化、ロバスト性の確保に必要である。

視野を広げるためにはマルチカメラのかわりに全方位カメラを利用することも考えられる。関連研究として、全方位カメラによる SLAM も提案されている[30]。全方位画像はカメラ中心が一つであるためモデル化が容易になるメリットがある。Google などは全方位のレーザを使っている。しかしながら全方位で撮影するためにはセンサを車の天井に貼り付ける必要があり、前述の意匠やメンテナンス性といった課題がある。

また歩行者検出の精度を上げるため、前方カメラとしてステレオカメラを取り付けることがある。しかしながらステレオカメラは単眼に比べて高価であるばかりでなく、経年変化によってキャリブレーションが変化するなど、単眼に比べて扱いにくい課題がある。

これらの理由から、車両の前後に単眼カメラをつけた構成が、車両メーカーにとって扱いやすく、かつ画像処理に適していると考えられる。

1.7.4.マルチカメラの利用の課題

前項で述べたように、車載されている複数のカメラを利用することで自己位置推定の精度が上がると考えられるが、そのための方法は明らかではない。特に車種によって搭載されるカメラ構成は異なるため、カメラが増減してもアルゴリズムを大きく変えない工夫があると良いと考えられる。

1.7.5.道路画像からの特徴抽出の課題

レーン認識で述べたように道路は様々な要素で構成される。具体的には白線、黄線、道路標示、縁石、雨水桝などである。道路画像を処理することで自己位置を補正する手がかりを抽出するとき、これらを個別に抽出する特徴抽出器を構成することもできるが、計算コストやアルゴリズムの複雑さが課題となる。そこで地図と相性が良く、単一の方法で特徴を抽出できる方法が必要と考えられる。

1.8.地図整備と自己位置推定の課題と分析

ここまでに複雑な都市高速や市街地における自動運転に向けた取り組みを説明してきた。自動運転のためのレーン認識には、画像処理のみでは直接の認識が難しいシーンがあった。これに対し、予め準備した走行レーン地図を使って自己位置を推定し、地図の参照によってレーン認識の代替とする方法が有望と考えられる。しかしながら地図は作成にもメンテナンスにも大きなコストがかかる。従来の地図作成プロセスで走行レーン地図を日本全国道路全て(127万 km)を集めるのは非現実的である。

そこで道路種別より、道路環境の複雑さと、総延長について考えてみる。前述の通り高速自動車国道は9千 kmであり、道路全体の127万 kmに比べると0.7%にすぎない。MMSが1日に数十 kmを計測できることから測量は可能であり、図化も地図に記載する特徴を減らせば、人海戦術で地図が作成できると予想できる。しかしながら高速自動車国道のうち、本研究が着目する都市高速は非常に複雑であり、記載の少ない地図でロバスト・高精度に自己位置を推定する方法が課題となる。

次に残りの99.3%を占める市街地を考えてみる。まず都市高速に比べて市街地には交差点や、複雑な道路標示がある。このため記載の少ない地図を使うのは良い方法といえず、記載の多い地図を扱う方法を考える。地図作成のうちMMSによる測量は、Googleストリートビューが実現されていることを考えれば、コストがかかるものの現実には不可能ではない。一方で図化を人海戦術で行うのは現実的ではなく、低コスト化の方法が必要となる。また自己位置推定では、地図に記載された特徴と、カメラ画像を処理して抽出される特徴の照合により、自己位置の補正量を決めていた(図 1.11)。これを改良するのであれば、地図の記載を増やして特徴を増やすことに加え、画像処理を工夫することで得られる特徴を増やすことが考えられる。そこで道路上の多くの特徴を抽出する画像処理が課題と考えられる。

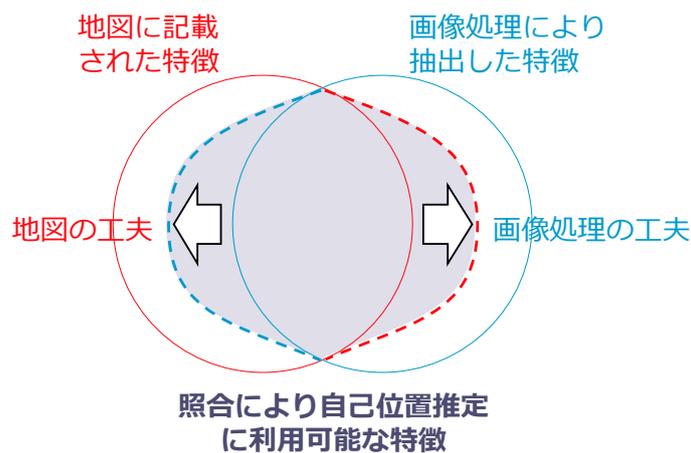


図 1.11 自己位置推定における改良の考え方

1.9.本研究の提案

本研究では複雑な都市高速や市街地において自動運転に必要なレーン認識を、低コストで高精度に実現することを目的とする。このために環境の複雑さと道路の総延長に対して、自己位置を補正する画像処理アルゴリズムと、事前に準備する地図の両側面から、改良方法を提案する(図 1.12).

まず総延長が短く、人手の作図が可能な都市高速では、現実的なコストで作られた記載された特徴が少ない地図に対応する方法を提案する。具体的には、地図の曖昧さをモデル化し、確率分布として扱うことでロバストに地図と画像を対応付ける方法を提案する。

次に総延長が長く、人手の作図が不可能な市街地では、地図を自動で図化することで、人手による作図コストを大きく削減する方法を提案する。自動化により低コストで得られる記載の多い地図は、人手で作図した地図の詳細化であるため、自然な移行が可能である。さらに車載された複数のカメラを利用し、道路の白線や黄線、縁石、道路標示といった多くの特徴を線分として抽出することで、自己位置推定に扱える特徴を増やす方法を提案する。

これらの提案により、都市高速から市街地までの複雑な道路でのレーン認識を、低コストで高精度に実現することを目標とする。以降に提案の詳細を述べる。

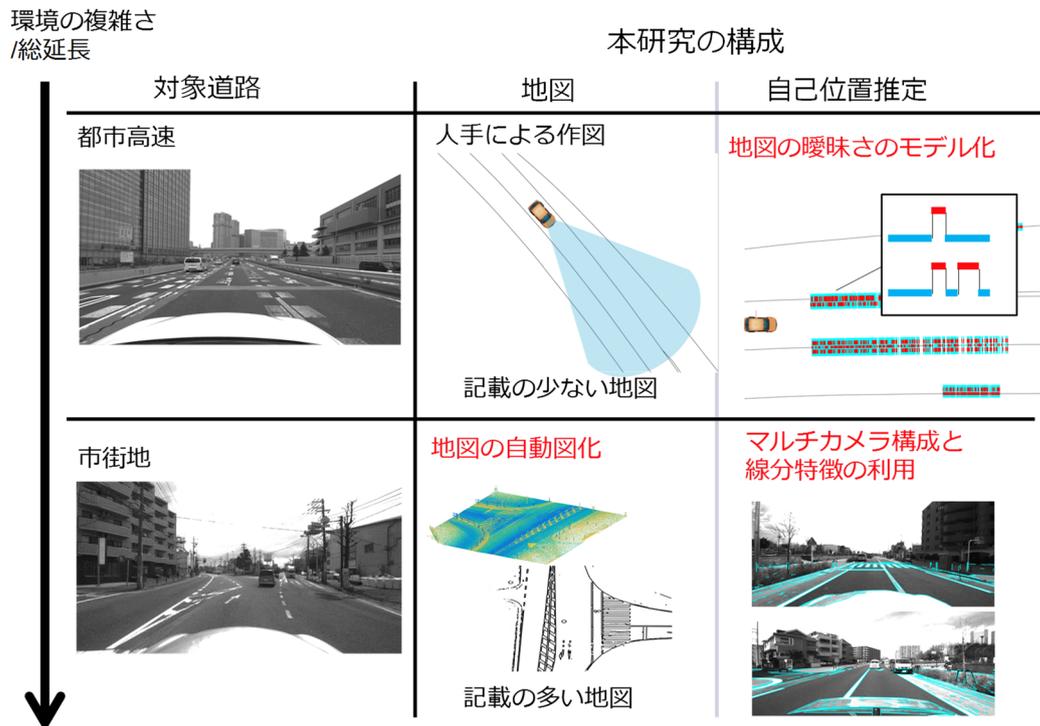


図 1.12 本研究の提案の構成

1.9.1.地図の曖昧さのモデル化

低コストで作られた記載の少ない地図に対応するため、地図上の記載が、画像上の見えに対して曖昧さが残る課題に対応する方法を提案する。このために、地図の曖昧さを確率分布として扱う方法を提案する。事前に白線形状の曖昧さを確率的にモデル化し、白線形状を確率的に発生させることで、ロバストに地図と画像を対応付けられることを示す。自己位置推定は非線形二次計画問題として定式化し、リアルタイム推定を実現する。複雑な市街地における走行評価によって自己位置の精度を測定し、自動運転の要求よりも高精度な自己位置が得られることを示す。

1.9.2.マルチカメラ構成と線分特徴による自己位置推定

自己位置を推定する画像処理アルゴリズムとして、多くの道路の特徴を捉える線分特徴と、それを使ったマルチカメラ構成への対応方法を提案する。まず、道路上にある白線、黄線、道路標示、縁石など異なる特徴を、線分検出を用いることで一元的に抽出する。これによりシンプルな構成で、多くの特徴を捉えることができる。さらにマルチカメラとして既存製品に適した構成である車両の前後に搭載された単眼カメラに対応する。検出した線分は車両座標系に鳥瞰変換して用いることで、カメラの構成に依存しない処理を実現する。自己位置推定には非線形二次計画問題の一部である拡張カルマンフィルタとして定式化し、リアルタイム推定を実現する。複雑な市街地における走行評価によって自己位置の精度を測定し、自動運転の要求よりも高精度な自己位置が得られることを示す。

1.9.3.地図の自動図化

地図整備の工夫として、地図の作図工程に踏み込み、コストのかかる人手作業を自動化することで、地図自体を自動で生成する方法を提案する。具体的にはMMSによって収集された3次元点群から、線分で表現される地図を自動的に抽出する。白線や黄線、道路標示、縁石といった多くの特徴について、エッジ部分が3次元線分で表現された地図が抽出できる。このために勾配画像処理を利用した効率的かつロバストな線分地図の抽出方法を提案する。抽出したい対象は高さ方向に重複が無い場合、二次元の画像に情報を落とすことで演算量を減らすことができる。このとき勾配画像を用いることで、複数回の計測で変化する反射強度に対してロバストに処理できる。さらに点群の反射強度から得られる勾配画像と、点群の高さから得られる勾配画像をそれぞれ生成することで、ほぼ同様の処理で白線などの道路ペイントと縁石の両方を扱うことができる。生成した地図を自己位置推定に適用して評価し、人手で作図した地図と同等の高精度な自己位置推定ができることを示す。

1.10. 本論文の構成

本章では自動運転の現状を概説し、自動運転に必要なレーン認識では地図と自己位置推定による方法が有力であることを説明した。市街地や都市高速では複雑な道路環境がある一方で、道路の総延長は膨大なため地図を準備するには多くのコストがかかる。この課題に対し地図の準備と自己位置推定の両側面から本研究のアプローチを提案した。次章においては関連研究を自己位置推定に用いるセンサと地図の違いによって整理し、それらに対する本研究の位置づけを示す。3章において自己位置推定の基礎となる車両の運動モデルと基本的な定式化を行う。地図上での自己位置の表現を定義し、拡張カルマンフィルタ、非線形最小二乗問題による自己位置の補正方法を述べる。4-5章では本研究が提案する自己位置推定方法を説明する。4章ではマルチカメラ画像から抽出した線分を使って、自己位置を補正する方法を提案する。5章では人手で作られた記載の少ない地図に対応するため、地図の曖昧さを確率でモデル化し、モデルを使った自己位置推定法を説明する。6章では4-5章で使った走行レーン地図をMMSで計測した3次元点群から、勾配画像処理を利用して自動生成する方法を述べる。7章では自動生成した地図の評価を示し、人手に比べて低コストで地図を作れることを示す。8章では自己位置推定の評価を示し、自動運転の目標とした横方向の誤差が0.2[m]以下であること、およびリアルタイム性を達成できることを示す。さらに自動で生成した地図でも人手と同等の自己位置の推定精度が出せることを確認し、本提案が有効であることを示す。9章で本論文をまとめ、今後の展望について述べる。

2. 関連研究

自動車における自己位置の推定は，センサデバイスと照合に用いる緯度経度と紐付いたデータの種類の2つの観点から分類される．まず初めにセンサによる分類を考え，GNSSによって直接に緯度・経度を取得する方法について説明し，次にレーザスキャナと地図を照合する方法，本研究が対象とするカメラを使った自己位置推定について述べる．カメラを使った自己位置推定は，照合に用いるデータの種類によって三通りに分けて述べる．空撮画像(もしくは衛星画像)を用いる方法，過去に撮影された車載カメラ画像を用いる方法，本研究が扱う線分やスプライン曲線による地図を用いる方法である．また，地図を生成する方法として，MMSによって収集されたレーザ点群を利用する方法，一般車両に搭載されたセンサを利用する方法について述べる．最後に本研究の位置付けを整理してまとめる．

2.1. GNSS による自己位置推定

近年，RTK-GPS や準天頂衛星の補正情報を利用する高精度な GNSS が製品化されている．観測できる衛星数や付近の遮蔽物などの条件を整えることで数[cm]の推定精度を実現できることから，自動運転の要求を満たし，実験にも用いられている[38][56]．しかしながら一般に数百万円と高額であることや大型の円盤上のアンテナの意匠の問題があり，市販自動車への搭載は難しい．また GNSS は絶対位置を計測するため，大陸プレートの移動や地震による変動を受けると，地図を作成した時点の位置と，最新の位置が異なる問題がある．また最低でも4基以上の衛星の電波を受信する必要があり，高い建物の多い都市部では安定しない．特に道路においては，2階建て等の低い建物であっても，建物と道路の距離が小さいと十分に衛星を遮る要因となりうる．このため細街路においても安定しないことが多い．

例として本研究で評価に利用した新川崎駅周辺にて，GNSS の精度をあらわす Quality indicator(NMDA GPGGA メッセージ)が RTK-Fixed であった区間を示す(図 2.1)．Quality indicator はメーカー固有の GNSS のソフトウェアによって内部で判定されているものであるが，RTK-Fixed であれば，マルチパスによる誤認識を除けば多くの場合で数[cm]の誤差に収まっていると考えてよい．新川崎駅周辺は，一般的な住宅街でありごく一部に高いマンションがあるものの多くは2～4階程度の建物である．またさらに実験を行うにあたって，受信状況の良い経路を慎重に選択している．RTK-GPS 装置には Novatel OEM615 を用い，米国の衛星群である GPS に加えて，ロシアの GLONASS も受信した．このように GNSS にとって有利な環境を選択してさえ頻繁に Fixed を外れてしまいうことが読み取れる．このため RTK-GPS をそのまま LKA や自動運転に用いることはできない．特に扱いの難しい問題として，RTK-GPS はその誤差を自己申告するが，誤差が数[cm]と申告しているにもかかわらず，1[m]以上ずれることがある．これはマル

チパスの影響であり，その識別が難しいためと推察される．このため申告された誤差が小さい時だけ使うといった利用ですら難しい．これらの理由から GNSS は主に自己位置推定の初期位置の設定や，レーザ等の別の自己位置推定とのフュージョンで用いられる[20].

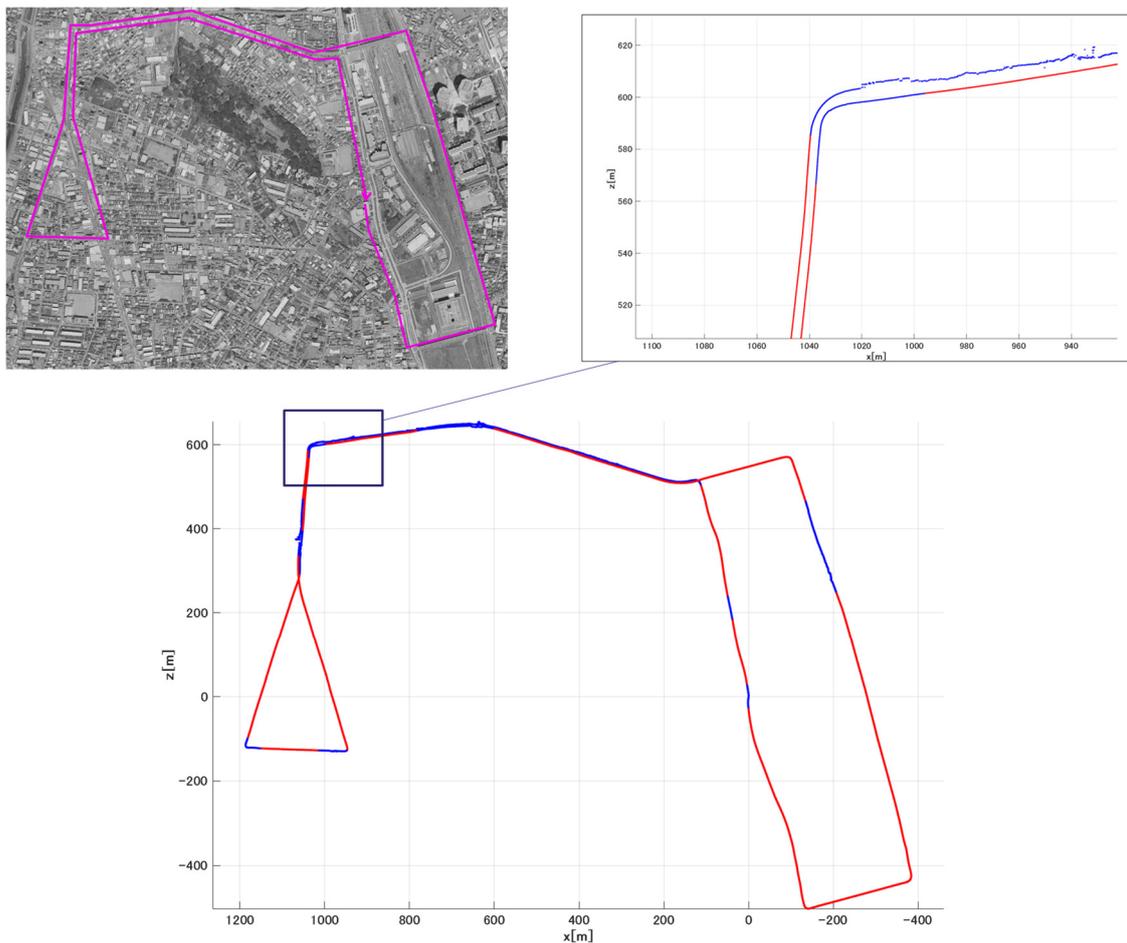


図 2.1 RTK-GPS の Quality indicator が Fixed であった区間

(左上図は新川崎駅周辺における走行経路を表している．下図の赤線は RTK-GPS が fixed であった区間，青線は float であった区間を表している．右上図のように float であった区間は計測位置に大きな誤差が発生する)

2.2. レーザスキャナによる自己位置推定

レーザスキャナの反射強度を地図の照合に用いた研究[20]がある．具体的にはあらかじめ走行した道路について，レーザスキャナの反射強度の値による画像を生成し地図とする．再び走行した際には同様に反射強度を照合に用いる．反射強度とオドメトリ，

GNSS を含むコスト関数について、非線形二次計画問題を解くことで自己位置を推定した。またレーザスキャナにより検出したポール位置を使って照合する研究がある[3]。電力柱、電信柱や、街路灯などは工事によって位置が変わることが少ないメリットがある。事前に取得した道路周辺のポール位置を地図とし、走行時には同様にレーザスキャナによって検出したポール位置と照合することで自己位置を推定した。本研究で扱うカメラと比較すると、レーザスキャナは一般にカメラに比べて高価である一方で、距離を直接検出できる。このため自車周辺を鳥瞰表現した反射強度のマップを扱うこと[20]ポールを扱うこと[3]に利点がある。

2.3.カメラ画像による自己位置推定

前節にて述べたようにカメラは直接に距離の取得ができないため縁石等の検出が難しいものの、白線や道路標示、横断歩道のようにテクスチャに基づく対象の検出が得意である。搭載性という意味でも、最近の市販車両の多くには安全用途のカメラが複数搭載されている。特にマルチカメラ構成ができれば視野が広がることから、自己位置推定の精度向上に有利であると考えられる。

2.3.1.衛星画像/空撮画像との照合

空撮画像や衛星画像は、近年 10~25[cm/pixel]の解像度のものを比較的安価に手に入れることができる(図 2.2)。空撮画像による自己位置推定[23]は、車載カメラ画像と空撮画像に写った路面ペイントを使って画像同士の位置合わせすることで自己位置を求める。画像同士の位置合わせは、文献[23]で使われたように SURF[1]などの局所特徴を用いるなど、多くの研究資産を活かすことができるメリットがある。しかしながら空撮画像を用いるのは製品化に際して品質とデータ容量の課題がある。市販の空撮画像は複数の画像を接続して生成されており、その位置決め誤差の影響により絶対位置がずれていることがある。またさらに空撮画像には駐車/走行中の車両による遮蔽や建物の影が含まれていることから、安定した品質を保つことが難しく、また品質基準の定義そのものが課題と考えられる。例えば図 2.2 の横断歩道に着目すると、白線間の黒い領域が、白い領域に比べて狭く見える。本来の横断歩道の白黒領域が等間隔であることを踏まえると、白領域が膨張していると考えられる。これは撮影条件や画像処理による輝度値の補正によって膨張したためと考えられる。さらに白線の幅は 15~20[cm]であり[51]、上記解像度では要求精度を出すのが難しい課題がある。



図 2.2 空撮画像/衛星画像の例
(25[cm/pixel]の空撮画像の例である．右図は左図を拡大した画像である)

2.3.2.過去に撮影された画像特徴との照合

一般に画像を使った SLAM の研究ではコーナー検出[32][37]とそのデスクリプタによる照合[1]を使う事が多い．これを応用し，過去に撮影された車載カメラ画像を用いる方法[16][37]は，路面だけでなく，画像に写った建物などの道路周辺の立体物を手がかりにできるメリットがある．画像間の特徴点の対応付けは一般に頑健であるメリットがあるものの，空撮画像と同様にデータ容量が問題になることや，デスクリプタの内容は目視による確認が困難であり品質を保証するのは困難と考えられる．また天候や太陽の位置によって，光源が変化するためロバスト性を上げる必要があること[17]や，草木などが季節で変化する課題がある．

具体的に車載カメラ画像に Harris[9]を適用し，特徴点を抽出した結果を示す(図 2.3)．緑の点が検出した特徴点である．まず目立って多くのコーナーが抽出されたのは街路樹であるが，このような特徴点は付近に似たテクスチャがあることが多く安定しない．また空やアンテナ，建物などに多くの特徴が見られるが，本研究で扱う地図と対応させるのは難しい．また相対的に建物が暗くなっており特徴点の数が減っているが，屋外では建物の影になっている場所と，そうでない場所で明暗が激しく，特徴点の選択に悪影響を及ぼすためである．横断歩道や白線のコーナーから特徴点が抽出されているものの，線としての構造が失われていることから，必ずしも有効な特徴とはいえない．このようにコーナー検出に基づく手法は，必ずしも車載カメラ画像に適しているとはいえないことがある．



図 2.3 道路におけるコーナー検出の例

2.3.3.ベクトル地図との照合

線分やスプライン曲線によって表現される路面の地図を用いる方法では白線、黄線、道路標示、側溝、縁石などの位置が高精度に記載された地図を用いる。このような長距離の複雑な地図を作れるようになった背景にはMMS(Mobile Mapping System)による測量技術[27]の発展が挙げられる。このような地図は従来のカーナビで使われる地図の自然な延長で考えられるため、データフォーマットや品質基準についての取り扱いが容易である。また自己位置推定のアプリケーションとして、自動運転やレーンキープアシストを想定した場合、白線や縁石といったレーンの境界や、道路標示といった情報は、走行軌跡を決める上で必要になる[44]。具体的には経路計画を行う際に、レーン左右の走行可能領域を利用し、それは線分やスプライン曲線などのベクトル地図を利用して高速に計算する。このため走行レーン地図の作成は避けることが難しいと考えられる。このようにアプリケーションにとって需要の多いデータであることは、地図の整備が十分になされることが期待できる。

このような路面の地図を用いた自己位置推定として、例えば Laneloc[31]は、ステレオカメラを用いて白線や縁石を個別に検出し、地図とマッチングすることにより自己位置推定を行った。Nedevschi ら[19]も同様に、ステレオカメラによって白線の境界や、縁石、停止線などを個別に検出している。これらの課題は白線や縁石など検出対象ごとに異なる特徴検出器を準備しなければならず構成が複雑になることであった。またステレオカメラを用いると縁石や路面が検出できるメリットがあるものの、コストやマルチカメラ化で不利になる。単眼で単一の特徴検出器を用いる研究[4]もあるが、白線検出技術に基づいたものであるため、複雑な交差点を曲がるようなケースには対応

できないものと推察される。一方で特徴検出器を用いず直接照合する方法として、レーザスキャナの反射強度を照合に用いた研究[20]、単眼カメラから得られる Structure tensor を基準として照合する研究[18]がある。どちらもパーティクルフィルタによって自己位置を最適化している。種別ごとに検出器を並べるのに比べ構成が簡略になるメリットがあるが、パーティクルフィルタは一般的に演算量で不利になる事が多く、車載を考えた際に課題になることがあった。市街地の自動運転で実績のある方法[45]は、車両から撮影した画像を用いる方法[16]と、線分による地図を用いる方法[31]のハイブリッドである。前方/後方のマルチカメラの構成をとっているものの、二つの方法は別のカメラを利用して別々に自己位置を算出し、最後に別の車両モデルでフュージョンする複雑な構成をとっている。このためマルチカメラ構成に適したアルゴリズムであるとは言えない。本研究に最も近い研究として、Yu らの方法[41]が挙げられる。本研究と同様に線分特徴を利用しているものの、車両の位置姿勢を 6 自由度で表現しているため、点特徴や建物のエッジの縦線の併用を必要とする。本来車両は路面勾配があるとはいえ 2 次元の路面上を走るものであるため 3 自由度しか持っておらず、より単純な構成が望まれる。また線と点を組み合わせた複雑な特徴で RANSAC による最適化を行っているため、フレームレートが遅い課題がある。

2.4. 走行レーン地図の生成

走行レーン地図は、自己位置推定のみならず経路計画にとっても必要であり、自動運転にとって必須である。走行レーン地図を作成するための元となる計測情報で分類し、MMS による計測に基づく方法と、一般車両による計測に基づく方法の二つを紹介する。

2.4.1. 一般車両のセンサ情報を利用する方法

一般車両の撮影した画像を、ネットワークを経由してセンターで収集することで、地図作成に必要な情報を収集することが提案されている[33]。これは一般の車両に搭載された GPS を利用し、特にその GPS のドップラーを利用することで高精度に車両軌跡を推定[71]し、高精度に推定された位置の付属した路面の鳥瞰をセンターで収集することで地図としている。地図作成のための MMS 測量が不要になるアイデアと言える。しかしながらカメラや GPS は路面勾配や、縁石といった高さに関する情報を集めにくいことや、長いトンネルや都市部など、GPS が長期にわたって受信できない場所で課題がある。また、この研究によって得られるのは、空撮画像を道路部分だけ高解像度にしたような画像であり、走行レーン地図にするには図化のステップが必要と考えられ、システムとしては不十分である。一般車両の撮影した画像から、線を使った Visual SLAM によって地図をつくる研究[13][35]がある。これは走行レーン地図に相当する地図を生成できるため、国内全て道路を網羅するような地図を生成するためには有望で

ある。しかしながら文献に記載された精度より、まだ実用的ではないと考えられる。

別の方法として、2.3.2 項で述べたカメラ画像と過去に撮影された画像の照合によって自己位置推定をする方法[16][37]では、過去に自動抽出した特徴量が位置に紐付いたデータベース、すなわち地図として扱われる。しかしながら、自動運転に必要な経路計画等に使えるような地図ではないことや、品質保証が難しい課題がある。

2.4.2.MMS によって収集されたカメラ画像やレーザ点群を利用する方法

現在主流となっている地図作成の方法は、MMS による測量と人手による図化の 2 ステップである。MMS によって収集されたレーザ点群から、地図とするのは人が CAD ツール上で書き入れており、この作業は図化と呼ばれている。MMS は一日に数十 km もの道路周辺の高密度な 3 次元点群を収集できることから、人手による図化のコストが支配的である。そこで人手作業の支援を目的とした実用的な半自動化の研究がありツール化されている[12][72]。またカメラ画像による白線検出の延長として、3次元点群と対応付けて地図とするシステムが実用化されている[46]。しかしながらカメラ画像を用いるのはカメラパラメータや取り付け位置のキャリブレーションが精度に影響を及ぼす課題がある。このため人手による図化ではカメラ画像は参考に止め、主には反射強度を用いられている。3次元点群の持つ反射強度から、道路標示や白線の領域を求める研究[8]があるが、線分や曲線のようなベクトル地図の生成は対象としていない。また特に自己位置推定などの実用に耐えうるには、白線のみならず各種の道路標示、横断歩道、縁石といった多くの特徴が網羅されている必要がある。これら全てを自動的に抽出できるアルゴリズムは提案されていない。

2.5.関連研究のまとめと本研究の位置付け

自己位置推定と地図生成の観点から関連研究を紹介してきた。これをまとめて本研究の位置付けを述べる。まず市街地や都市高速における自己位置推定のためのセンサと地図を俯瞰する。まず GNSS による推定は都市部で安定しないことからそのまま利用できない。レーザスキャナによる推定は、自車周辺の鳥瞰変換や、ポールの検出に利点があるものの、カメラに比べて高価である。本研究で扱うカメラは白線や道路標示、横断歩道のようにテクスチャに基づく対象の検出が得意であり、搭載性の面でも有望である。このため本研究ではカメラを扱うものとした。

次に本研究が扱うカメラとの照合に用いる地図には、衛星/空撮画像、画像特徴のデータベース、ベクトル地図の 3 種がある。しかしながら衛星/空撮画像、画像特徴のデータベースは解像度や光源の変化、移動体による遮蔽、品質の扱いなどの問題があるため、カーナビの自然な延長で考えられるベクトル地図が有力であり、本研究でも線分で表現されたベクトル地図を扱う。

カメラ画像とベクトル地図を照合する自己位置推定には多くの研究がある。しかし

ながら道路上には白線や縁石など異なる多くの特徴があるが、これらを単一の方法で有効活用する方法は提案されていない。またマルチカメラが扱えれば視野が広がり、自己位置の精度に大きく貢献すると考えられるが、車種によってカメラ構成が異なる課題があり、カメラ構成が変わっても処理が変わらない方法は提案されていない。本研究では、線分特徴を用いることで、路面上の白線、黄線、道路標示、縁石など異なる特徴を一元的に抽出して扱える。また、検出した線分を車両座標系に統一して扱うことで、カメラの構成に依存しない処理を実現する。

次に地図生成においては、作成元の計測情報として、一般車両と MMS が挙げられた。しかしながら一般車両からの計測では、精度や情報不足の観点からまだ難しく実用的ではない。MMS による計測では、白線検出の延長として画像から計算する研究はあるものの、膨大なレーザ点群から直接に線分地図を生成する方法は提案されていない。本研究は膨大なレーザ点群を勾配画像に変換することで、線分で描かれた地図を直接生成する方法を提案する。

さらに本研究は、自動生成した地図でも自己位置推定ができることを評価しており、線分で描かれた地図の生成と自己位置推定のセットで評価した唯一の研究である。

3. 車両の運動モデルと自己位置推定

カメラを使った自己位置推定では地図上での車両の位置姿勢を求める。このためにまず地図と車両とカメラを扱う座標系を定義し、それに基づいて車両が従う運動モデルを設計する。カメラと地図の照合によって得られる観測は正規分布の誤差をもつと仮定し、非線形二次計画問題として自己位置の補正方法を説明する。なお具体的な観測モデルは次章以降で説明し、ここではより一般的な表現で述べる。

3.1. 路面地図上での自己位置の表現

路面地図上での車両の運動を表現するため、それぞれの座標系を定義する。まず車両運動を記述するのに便利な後輪車輪軸上の中心に、図 3.1 で示す車両座標系を設定する。車載カメラは車両に取り付けられており、剛体として扱われる。 n 番目のカメラ座標系における 3 次元点 \mathbf{p}_{c_n} と、車両座標系における点 \mathbf{p}_v の関係は、車両に対するカメラの取り付け位置を表す回転・並進行列 \mathbf{R}_{vc_n} 、 \mathbf{t}_{vc_n} によって定義され、次の式で対応付けられる。これらは定数として扱い、事前のキャリブレーションによって求めておく。

$$\mathbf{p}_v = \mathbf{R}_{c_nv} \mathbf{p}_{c_n} + \mathbf{t}_{c_nv} \quad (3.1)$$

地図座標系の点 \mathbf{p}_m と車両座標系の点は離散化された時刻 k における 6 自由度の回転・並進行列 \mathbf{R}_{vw}^k 、 \mathbf{t}_{vw}^k より次の式で対応付けられる。

$$\mathbf{p}_m = \mathbf{R}_{vw}^k \mathbf{p}_v + \mathbf{t}_{vw}^k \quad (3.2)$$

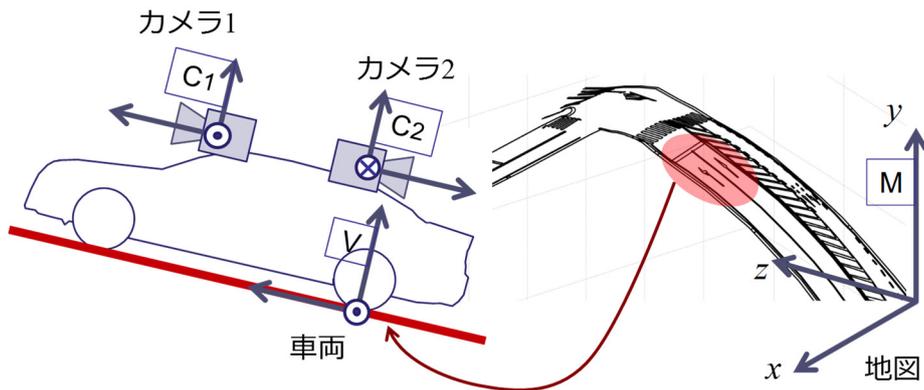


図 3.1 座標系の定義と地図の平面近似

3.1.1. 地図の道路勾配を使った車両姿勢の分解

市街地や都市高速道路では勾配の強い道路があるため、道路全体を平面と仮定した

平面上位置+Yaw 角回転の 3 自由度の車両姿勢の推定では対応できない. しかしながら回転・並進の 6 自由度を全て推定する場合, 推定変数が増えることから精度の低下が予想される. そこで本研究では地図から情報を得ることで 3 自由度に制限する(図 3.1). 具体的には車両は地図を局所的に近似した平面上を走行しているものと仮定することで路面の勾配を得る. 路面の勾配は, 車両の Roll/Pitch 軸の回転 \mathbf{R}_{slope}^k に対応する. これにより車両の姿勢 \mathbf{R}_{vw}^k は, 地図から得られる \mathbf{R}_{slope}^k と, 推定が必要な \mathbf{R}_{yaw}^k の二つに分解できる.

$$\begin{aligned}\mathbf{R}_{vw}^k &= \mathbf{R}_{slope}^k \mathbf{R}_{yaw}^k \\ \mathbf{R}_{yaw}^k &= \text{Yaw}(\theta_{vw}^k)\end{aligned}\quad (3.3)$$

また車両の位置 $\mathbf{t}_{vw}^k = [x_{vw}^k, y_{vw}^k, z_{vw}^k]^T$ についても同様に 3 自由度があるが, 車両が平面上を走行していると仮定すると, 地図から高さ y_{vw}^k を得ることができる. これらをまとめると, 本研究が時刻 k において推定する変数は, 車両の Yaw 回転を表すと θ_{vw}^k と, 平面上での並進を表す x_{vw}^k, z_{vw}^k の 3 自由度となる. 以降, 表記を容易にするため添字を省略し, $\mathbf{x}^k = [x^k, z^k, \theta^k]^T$ と書くものとする.

3.1.2. 自車近傍の路面勾配の計算

前項で用いた自車近傍の路面勾配を表す回転行列 \mathbf{R}_{slope}^k を求める方法について述べる. 自車近傍の地図の平面近似には, 地図に含まれる線分の端点の 3 次元座標を利用する. 直線道路などは線分の端点が疎であることがあり, そのまま利用すると近傍の点の数が極端に減るため, 線分を 1.0[m] の等間隔に分割することで点の密度を揃える. また平面近似する近傍の範囲として, 道路の幅員の規定[51]を参考にすると, 自車両を中心とした 3.5[m] の範囲が適切であると考えられる. しかしながら広い交差点などでは, 3.5[m] の範囲に平面近似に十分な点が無いことがあり, ランク落ちや精度低下を招くことがある. このためその 3 倍に相当する 10.5[m] の範囲の点を平面近似に用い, 推定したい位置からの距離に応じて, 標準偏差 3.5[m] の正規分布によって重みをつけた(図 3.2). 具体的には求めたい位置 $[x, z]^T$ から高さ y を回帰する関数 $y = a_c + a_x x + a_z z$ を地図の 3 次元点 $[m_x^i, m_y^i, m_z^i]^T$ から学習する. 具体的には次のコスト E を最小とする平面 $\mathbf{a} = [a_c, a_x, a_z]^T$ を求めた.

$$\begin{aligned}E &= \sum_{i \in \{i | d^i < 3\sigma\}} w^i (m_y^i - a_c - a_x m_x^i - a_z m_z^i)^2 \\ w^i &= N(d^i | \mu, \sigma^2) \\ d^i &= \sqrt{(m_x^i - x)^2 + (m_z^i - z)^2} \\ \mu &= 0 \\ \sigma &= 3.5\end{aligned}\quad (3.4)$$

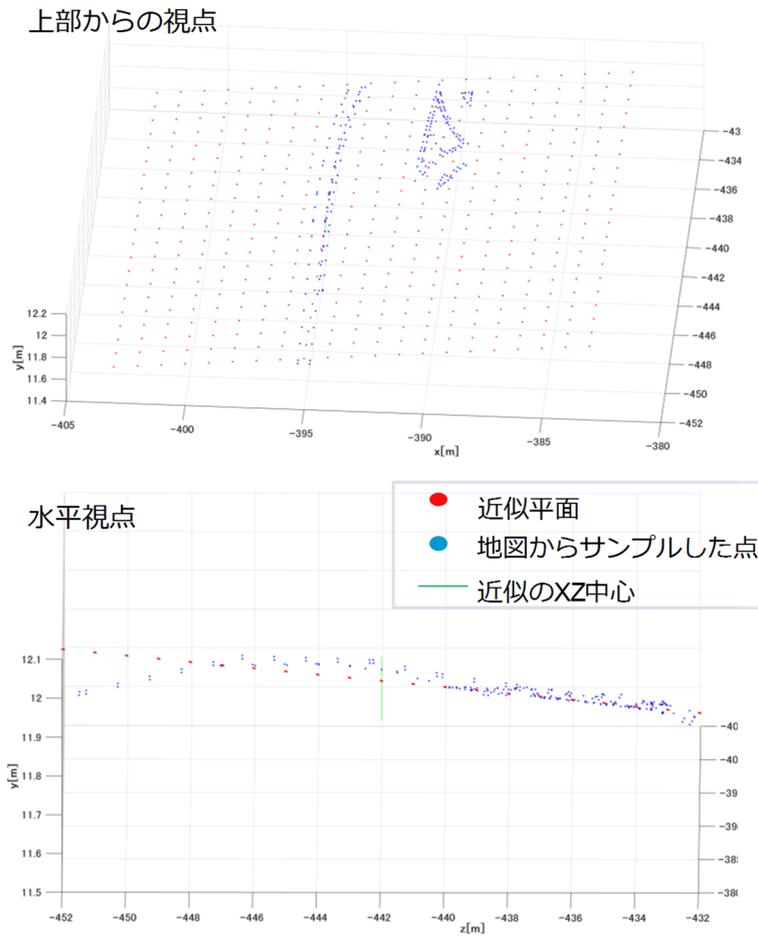


図 3.2 地図の平面近似

近似した平面に対応した回転 \mathbf{R}_{slope}^k を計算する．バイアス成分を除いて，平面 $y = a_x x + a_z z$ を考え，平面上の単位ベクトル $\boldsymbol{\gamma}$ を $x=0, z=1$ とする．

$$\boldsymbol{\gamma} = \frac{1}{\sqrt{1 + a_z^2}} \begin{bmatrix} 0 \\ a_z \\ 1 \end{bmatrix} \quad (3.5)$$

平面の法線ベクトルは次になる．

$$\boldsymbol{\beta} = \frac{1}{\sqrt{1 + a_x^2 + a_z^2}} \begin{bmatrix} a_x \\ -1 \\ a_z \end{bmatrix} \quad (3.6)$$

これより回転行列は次で構成できる．

$$\mathbf{R}_{slope}^k = [\boldsymbol{\beta} \times \boldsymbol{\gamma}, \boldsymbol{\beta}, \boldsymbol{\gamma}] \quad (3.7)$$

また前項の y_{vw}^k も $y_{vw}^k = a_c + a_x x_{vw}^k + a_z z_{vw}^k$ として求めた.

なおこの処理は地図が与えられれば事前計算が可能である. このため実験に用いる XZ 平面を 1.0×1.0 [m] のグリッドに区切って事前計算を行い, 自己位置に応じて参照するものとした.

3.2. 車両の運動モデル

自動車は車輪によって運動が拘束される. 本研究ではロボットの分野における 2 輪モデル[34]を参考に, 後輪 2 輪に着目した運動モデルを利用する. また最近の多くの自動車には, 走行を安定させることを目的として, 自車の相対運動を捉えるセンサが搭載されておりアプリケーションは CAN(Control Area Network)を介して利用することができる. そのひとつに左右後輪の車輪速 v_R^k, v_L^k があり, 次の式により車両速度 v^k と, 角速度 ω^k が計算できる. d_r は後輪車輪間の距離を表す. これをまとめて制御量 $\mathbf{u}^k = [v^k, \omega^k]^T$ とおく. このような定常センサのデータはオドメトリと呼ばれる.

$$\begin{aligned} v^k &= (v_R^k + v_L^k) / 2 \\ \omega^k &= (v_R^k - v_L^k) / 2d_r \end{aligned} \quad (3.8)$$

オドメトリを使って, 車両運動を正規分布でモデル化する. まず時刻 $k-1$ における自己位置 \mathbf{x}^{k-1} が, 平均 $\boldsymbol{\mu}^{k-1}$, 共分散行列 $\boldsymbol{\Sigma}^{k-1}$ の正規分布に従うとする. 同様に時刻 k における予測位置 $\bar{\mathbf{x}}^k$ も平均 $\bar{\boldsymbol{\mu}}^k$, 共分散行列 $\bar{\boldsymbol{\Sigma}}^k$ の正規分布に従うとする. 自己位置 \mathbf{x}^{k-1} から, 予測位置 $\bar{\mathbf{x}}^k$ を次の式で求める. Δt は時刻の差分である. なお予測誤差は平均 $\mathbf{0}$, 共分散行列 \mathbf{Q}^k の正規分布に従うものとし, その設計は次項で説明する. 共分散行列はヤコビ行列 $\mathbf{F}^k = \partial \mathbf{f} / \partial \mathbf{x}^{k-1}$ によって線形化して近似した.

$$\begin{aligned} \bar{\boldsymbol{\mu}}^k &= \mathbf{f}(\boldsymbol{\mu}^{k-1}, \mathbf{u}^k) \\ &= \boldsymbol{\mu}^{k-1} + \begin{bmatrix} v^k \Delta t \operatorname{sinc}(\omega^k \Delta t / 2) \sin(\theta^{k-1} + \omega^k \Delta t / 2) \\ v^k \Delta t \operatorname{sinc}(\omega^k \Delta t / 2) \cos(\theta^{k-1} + \omega^k \Delta t / 2) \\ \omega^k \Delta t \end{bmatrix} \\ &= \boldsymbol{\mu}^{k-1} + \begin{bmatrix} \frac{v^k}{\omega^k} \cos(\theta^{k-1}) - \frac{v^k}{\omega^k} \cos(\theta^{k-1} + \omega^k \Delta t) \\ -\frac{v^k}{\omega^k} \sin(\theta^{k-1}) + \frac{v^k}{\omega^k} \sin(\theta^{k-1} + \omega^k \Delta t) \\ \omega^k \Delta t \end{bmatrix} \\ \bar{\boldsymbol{\Sigma}}^k &= \mathbf{F}^k \boldsymbol{\Sigma}^{k-1} \mathbf{F}^{kT} + \mathbf{Q}^k \end{aligned} \quad (3.9)$$

3.2.1. 共分散行列の設計

一般に設計のしやすさから予測誤差を表す \mathbf{Q}^k はパラメータとして対角行列で与えることがある。しかしながら車両の場合 \mathbf{Q}^k を対角行列に設計すると、角度誤差が無い時の誤差楕円は、道路平面上を正円状に広がることになる(図 3.3a)。実際の車両は左右に並進移動することは無いため正しい誤差モデルとはいえない。そこで主な誤差は速度、角速度を含む制御量 \mathbf{u} にあるとし、走行距離に応じて広がるものとした(図 3.3b)。具体的にはヤコビ行列 $\partial \mathbf{f} / \partial \mathbf{u}^t$ による一次の近似を用いて次式で表した。

$$\mathbf{Q}^k = \left(\frac{\partial \mathbf{f}}{\partial \mathbf{u}^k} \mathbf{Q}_u \frac{\partial \mathbf{f}^T}{\partial \mathbf{u}^k} \right) \Delta d + \mathbf{Q}_c \Delta t \quad (3.10)$$

Δd は一時刻前との走行距離の差分である。 \mathbf{Q}_u は制御量に対応した誤差を表現する対角行列であり、 \mathbf{Q}_c はモデルに従わない小さな誤差を表現する対角行列である。

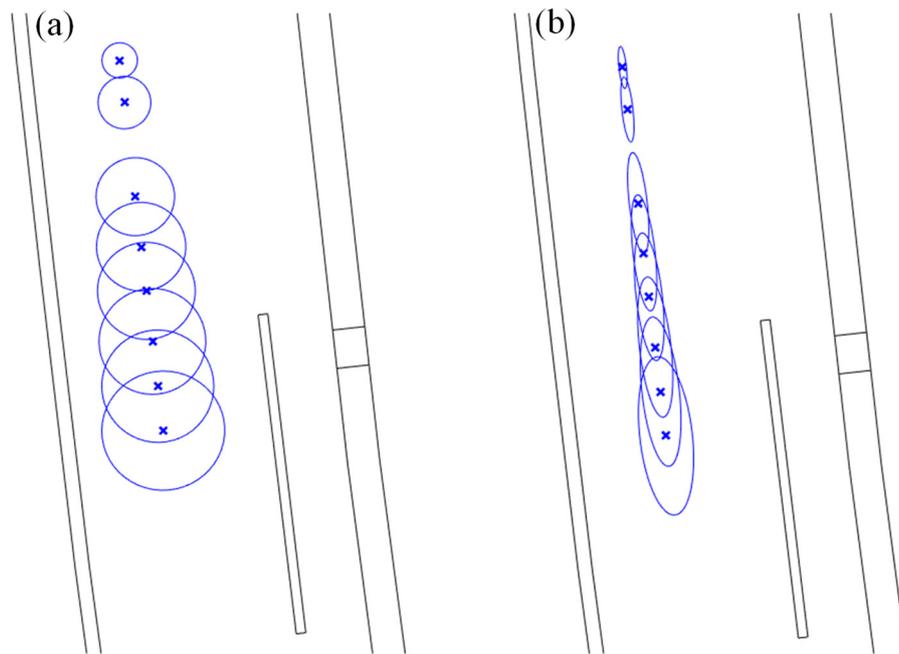


図 3.3 車両運動の予測誤差

3.2.2. ヤコビ行列の計算

運動に関するヤコビ行列の具体的な計算について記載する。

まず $\partial \mathbf{f} / \partial \mathbf{x}^{k-1}$ は次のように計算できる

$$\mathbf{F}^k = \frac{\partial \mathbf{f}}{\partial \mathbf{x}^{k-1}} = \begin{bmatrix} \frac{\partial x^k}{\partial x^{k-1}} & \frac{\partial x^k}{\partial z^{k-1}} & \frac{\partial x^k}{\partial \theta^{k-1}} \\ \frac{\partial z^k}{\partial x^{k-1}} & \frac{\partial z^k}{\partial z^{k-1}} & \frac{\partial z^k}{\partial \theta^{k-1}} \\ \frac{\partial \theta^k}{\partial x^{k-1}} & \frac{\partial \theta^k}{\partial z^{k-1}} & \frac{\partial \theta^k}{\partial \theta^{k-1}} \end{bmatrix} = \begin{bmatrix} 1 & 0 & \frac{\partial x^k}{\partial \theta^{k-1}} \\ 0 & 1 & \frac{\partial z^k}{\partial \theta^{k-1}} \\ 0 & 0 & 1 \end{bmatrix} \quad (3.11)$$

このとき、 $\omega^k \neq 0$ においては次の式を用いる

$$\begin{aligned} \frac{\partial x^k}{\partial \theta^{k-1}} &= \frac{v^k}{\omega^k} \left(-\sin(\theta^{k-1}) + \sin(\theta^{k-1} + \omega^k \Delta t) \right) \\ \frac{\partial z^k}{\partial \theta^{k-1}} &= \frac{v^k}{\omega^k} \left(-\cos(\theta^{k-1}) + \cos(\theta^{k-1} + \omega^k \Delta t) \right) \end{aligned} \quad (3.12)$$

$\omega^k = 0$ においては、次の式を用いる。

$$\begin{aligned} \frac{\partial x^k}{\partial \theta^{k-1}} &= +v^k \Delta t \cos(\theta^{k-1}) \\ \frac{\partial z^k}{\partial \theta^{k-1}} &= -v^k \Delta t \sin(\theta^{k-1}) \end{aligned} \quad (3.13)$$

次に $\partial \mathbf{f} / \partial \mathbf{u}^k$ は次の式で計算される。

$$\frac{\partial \mathbf{f}}{\partial \mathbf{u}^k} = \begin{bmatrix} \frac{\partial x^k}{\partial v^k} & \frac{\partial x^k}{\partial \omega^k} \\ \frac{\partial z^k}{\partial v^k} & \frac{\partial z^k}{\partial \omega^k} \\ \frac{\partial \theta^k}{\partial v^k} & \frac{\partial \theta^k}{\partial \omega^k} \end{bmatrix} = \begin{bmatrix} \frac{\partial x^k}{\partial v^k} & \frac{\partial x^k}{\partial \omega^k} \\ \frac{\partial z^k}{\partial v^k} & \frac{\partial z^k}{\partial \omega^k} \\ 0 & \Delta t \end{bmatrix} \quad (3.14)$$

$\omega^k \neq 0$ においては次の式を用いる

$$\begin{aligned} \frac{\partial x^k}{\partial v^k} &= +\frac{1}{\omega^k} \cos(\theta^{k-1}) - \frac{1}{\omega^k} \cos(\theta^{k-1} + \omega^k \Delta t) \\ \frac{\partial x^k}{\partial \omega^k} &= -\frac{v^k}{\omega^{k2}} \cos(\theta^{k-1}) + \frac{v^k}{\omega^{k2}} \cos(\theta^{k-1} + \omega^k \Delta t) \\ &\quad + \frac{v^k}{\omega^k} \Delta t \sin(\theta^{k-1} + \omega^k \Delta t) \\ \frac{\partial z^k}{\partial v^k} &= -\frac{1}{\omega^k} \sin(\theta^{k-1}) + \frac{1}{\omega^k} \sin(\theta^{k-1} + \omega^k \Delta t) \\ \frac{\partial z^k}{\partial \omega^k} &= +\frac{v^k}{\omega^{k2}} \sin(\theta^{k-1}) - \frac{v^k}{\omega^{k2}} \sin(\theta^{k-1} + \omega^k \Delta t) \\ &\quad + \frac{v^k}{\omega^k} \Delta t \cos(\theta^{k-1} + \omega^k \Delta t) \end{aligned} \quad (3.15)$$

$\omega^k = 0$ においては次の式を用いる.

$$\begin{aligned}\frac{\partial x^k}{\partial v^k} &= \Delta t \sin(\theta^{k-1}) \\ \frac{\partial x^k}{\partial \omega^k} &= \frac{+v^k \Delta t^2 \cos(\theta^{k-1})}{2} \\ \frac{\partial z^k}{\partial v^k} &= \Delta t \cos(\theta^{k-1}) \\ \frac{\partial z^k}{\partial \omega^k} &= \frac{-v^k \Delta t^2 \sin(\theta^{k-1})}{2}\end{aligned}\tag{3.16}$$

3.3. 拡張カルマンフィルタによる自己位置の推定

拡張カルマンフィルタ(EKF) [34]による, 自己位置推定方法を述べる. 観測の方法については次章以降で詳しく述べるものとし, ここでは観測の誤差が局所線形化により正規分布で表現できる場合の, 一般的な方法として説明する.

時刻 k における観測の誤差を $\boldsymbol{\delta}^k$ とし, その共分散行列を \mathbf{R}^k とする. また誤差を局所線形化するため, ヤコビ行列を $\mathbf{H}^k = \partial \boldsymbol{\delta}^k / \partial \mathbf{x}^k$ とする. 時刻ごとに予測ステップと観測ステップ計算することで自己位置 \mathbf{x}^k を補正できる.

① 予測ステップ

$$\begin{aligned}\bar{\boldsymbol{\mu}}^k &= \mathbf{f}(\boldsymbol{\mu}^{k-1}, \mathbf{u}^k) \\ \bar{\boldsymbol{\Sigma}}^k &= \mathbf{F}^k \boldsymbol{\Sigma}^{k-1} \mathbf{F}^{kT} + \mathbf{Q}^k\end{aligned}\tag{3.17}$$

② 観測ステップ

$$\begin{aligned}\mathbf{K}^k &= \bar{\boldsymbol{\Sigma}}^k \mathbf{H}^{kT} (\mathbf{H}^k \bar{\boldsymbol{\Sigma}}^k \mathbf{H}^{kT} + \mathbf{R}^k)^{-1} \\ \boldsymbol{\mu}^k &= \bar{\boldsymbol{\mu}}^k - \mathbf{K}^k \boldsymbol{\delta} \\ \boldsymbol{\Sigma}^k &= (\mathbf{I} - \mathbf{K}^k \mathbf{H}^k) \bar{\boldsymbol{\Sigma}}^k\end{aligned}\tag{3.18}$$

3.4. 非線形最小二乗問題による自己軌跡の推定

過去の推定位置が誤っているとき, 最新時刻の観測を使って, 過去の位置を再解釈すると, より高精度に推定できる. このため自己の軌跡として最新時刻 k から $k-\tau$ 時刻まで遡った自己軌跡 $\mathbf{x}^{k-\tau}$, $\mathbf{x}^{k-\tau+1}$, ..., \mathbf{x}^k を推定する. τ を 1 とすることでカルマンフィルタと等価になり, カルマンフィルタを包含している. 具体的な定式化には文献[34]に記載されている Graph SLAM を参考にした.

まずオドメトリによる予測誤差を次の式で表現する. これは前に説明した共分散行列 \mathbf{Q}^k の正規分布に従う誤差を持つ.

$$\boldsymbol{\varepsilon}^k = \mathbf{x}^k - \mathbf{f}(\mathbf{x}^{k-1}, \mathbf{u}^k)\tag{3.19}$$

次に観測の誤差 $\boldsymbol{\delta}^k$ とその共分散行列 \mathbf{R}^k とあわせてコスト関数 E を定義し, E を最小とする軌跡 $\mathbf{x}^{k-\tau}$, $\mathbf{x}^{k-\tau+1}$, ..., \mathbf{x}^k を求める.

$$\begin{aligned}
& E(\mathbf{x}^{K-\tau}, \mathbf{x}^{K-\tau+1}, \dots, \mathbf{x}^K) \\
&= \sum_{j=k-\tau+1}^k \boldsymbol{\varepsilon}^j (\mathbf{Q}^j)^{-1} \boldsymbol{\varepsilon}^{jT} + \sum_{j=k-\tau+1}^k \boldsymbol{\delta}^j (\mathbf{R}^j)^{-1} \boldsymbol{\delta}^{jT} \\
&+ (\mathbf{x}^{K-\tau} - \boldsymbol{\mu}^{k-\tau}) \boldsymbol{\Sigma}^{k-\tau} (\mathbf{x}^{K-\tau} - \boldsymbol{\mu}^{k-\tau})^T
\end{aligned} \tag{3.20}$$

このコスト関数は軌跡の誤差モデルは非線形最小二乗問題であるので、レーベンバーグマーカート法などで解くことができる。特に文献[34]では Graph SLAM として効率的に解く方法を紹介されており、本研究もそれに倣って実装を行った。最適化には $3(\tau+1) \times 3(\tau+1)$ サイズの連立方程式の解法を必要とするが、行列は幅 6 のバンド行列となっており τ に対する線形オーダーの計算で解くことができる。

また、全ての時刻の姿勢を計算するのはやや冗長といえるため、各時刻をフレームと呼び、キーフレームと従属フレームに分離し、キーフレームのみの姿勢を推定した(図 3.4)。従属フレームはオドメトリによる予測位置を計算し、従属フレームで観測された白線も誤差モデルに用いた。これにより推定変数を大きく削減できる。後述の評価では $\tau+1=50$ フレームの最適化を行うが、2 フレームに 1 フレームのキーフレームとしたため、推定変数は本来 150 個であるところを 75 個に削減した。50 フレームは、高速道路で 20[m/s]で走行、10[fps]で撮影したと仮定すると、100[m]の距離に相当する。つまり本手法では 100[m]分の白線とオドメトリを利用して自己位置を推定する。このため急カーブの多い都市高速道路では、車両の前後方向の補正が働く効果も見られた。

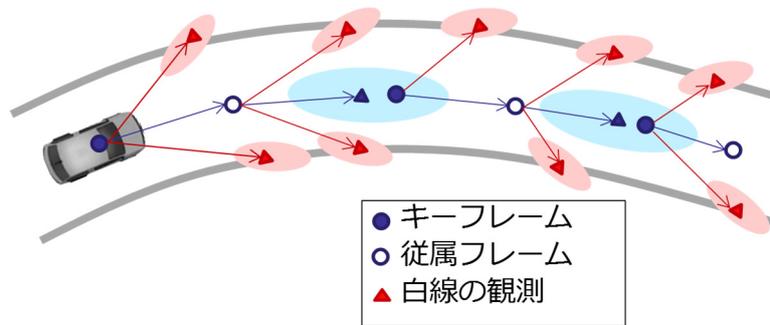


図 3.4 キーフレームと従属フレーム

4. 線分特徴による観測モデル

前章で紹介した自己位置推定のうち，本章では具体的な観測モデルを説明する．特に本研究の提案の一つである，線分特徴を使った自己位置推定について述べる．自動車に多くあるマルチカメラの構成を想定し，それぞれのカメラから道路に多く有る線分を捉えることで，効率の良い特徴抽出を実現する．また，抽出した線分を使って自己位置を補正するための誤差を定義する．

4.1. 車載カメラ画像におけるコーナー点とエッジ線

文献[16]のようにコーナー検出による点をデータベース化することで，自己位置を推定する方法がある．一方で，車載カメラ画像にコーナー検出をかけると，空との境界や，街路樹など，安定しない場所からも多数の検出がでてしまう課題があった．そこで道路の特徴に着目すると，基本的に車両は直進することから，道路の白線や縁石には直線が多く含まれることがわかる．その比較としてコーナー検出と線分検出の例を示す(図 4.1)．コーナー点では，白線や縁石の端点しか用いることができず，長いエッジ部分が捨てられている．一方で，線分検出では，エッジに加えて，コーナー部分も線分端点として捉えることができている．この点に着目し，本研究では線分検出を使った自己位置推定を提案する．

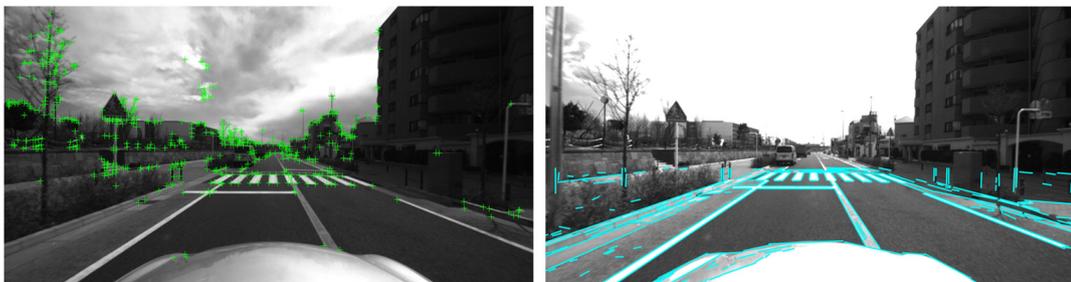


図 4.1 車載カメラ画像からのコーナー点とエッジ線の抽出

4.2. マルチカメラによる線分の観測と自己位置推定

提案法は線分特徴の抽出とその鳥瞰変換，地図との照合，自己位置の予測と補正のステップに分けられる(図 4.2)．入力は車両に搭載された複数のカメラによる画像系列と後輪車輪速である．ここではカメラは車両の前後に取り付けられているものとして説明するが，カメラ台数や位置の変更は容易な方法として設計されている．出力は地図上での自己位置であり，自車両の位置・姿勢の3自由度の状態を推定する．

地図と照合するための手がかりとして、画像から抽出される線分特徴を用いる。このためにまず各カメラ画像について線分検出を行う。それらは車両座標系へ鳥瞰変換され、カメラ座標系に非依存の特徴として扱われる。このため、カメラ台数やカメラの取付け位置が変更されても容易に対応できる。次に鳥瞰変換された線分と地図の線分を対応付け誤差と共分散行列を計算し、自己位置推定の観測モデルとして扱う。

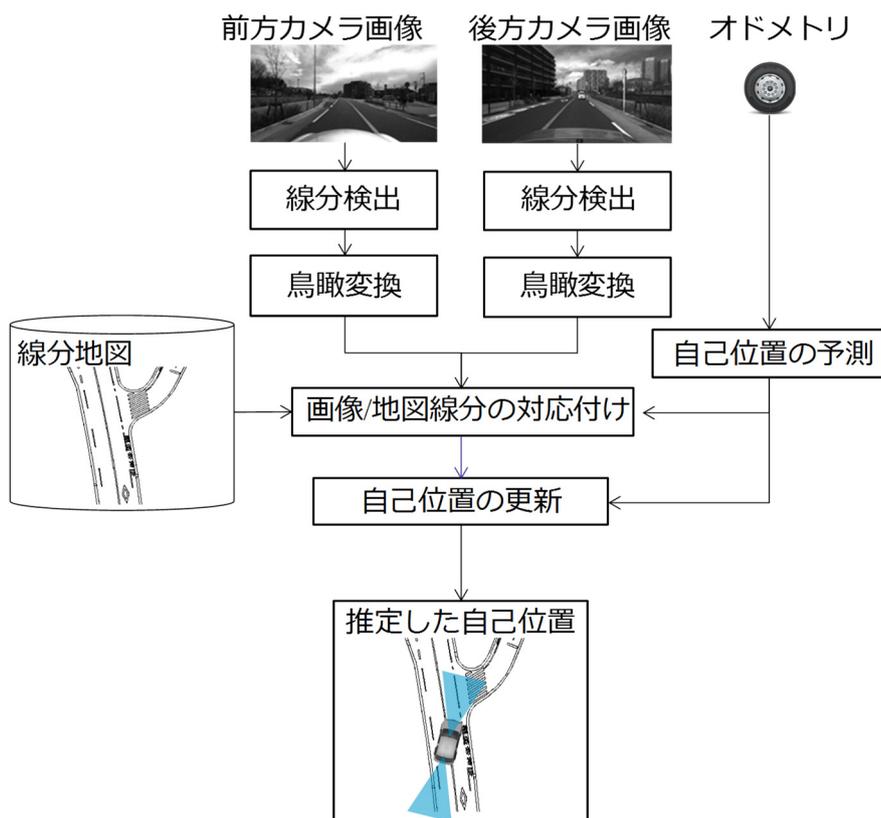


図 4.2 自己位置推定のシステム構成

4.3.線分地図

本研究では線分で表現された地図を用いる(図 4.3)。地図には白線、黄線、道路標示、縁石、側溝、雨水柵が記載されており、各線分の端点の3次元座標が与えられている。ここで、白線や黄線、道路標示はその白線領域の内側が明るく、外側のアスファルトは暗く撮影される。一方で縁石や側溝、雨水柵は、内側、外側の区別が無く、明暗の方向が決まらない。そこでこれらの違いを地図にラベルとして記載した。

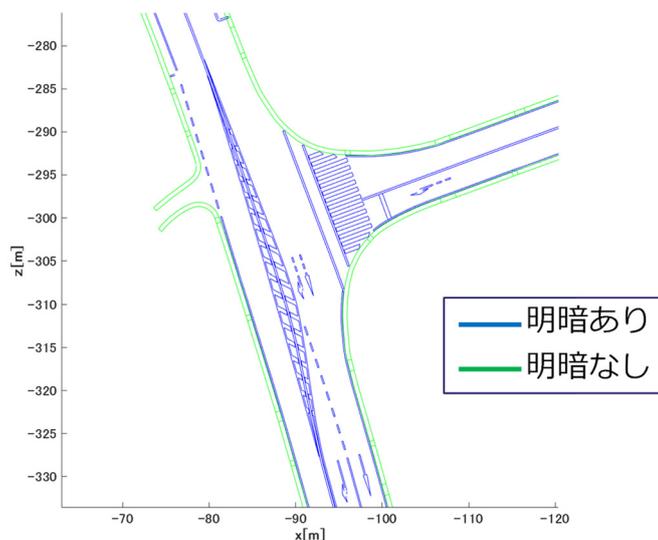
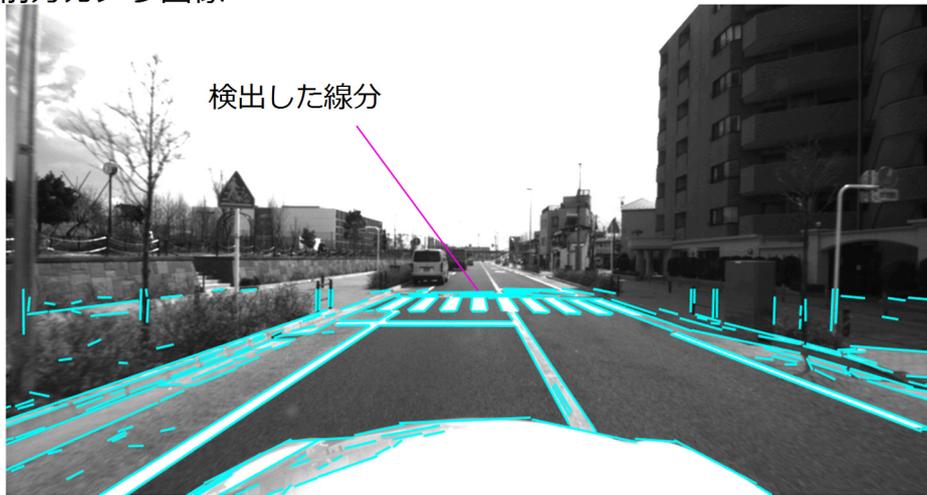


図 4.3 明暗の有無をラベル付けされた線分地図

4.4. 運動モデルによる自己位置の予測

SLAM や自己位置推定の研究においては、画像中の特徴的な領域を抽出することで演算量を削減する。本研究ではこのために線分抽出を用いる。具体的には複数の車載カメラから得られるマルチカメラ画像に対して、それぞれ線分検出を行う。線分の抽出には LSD(Line Segment Detector)[6]を用いた。LSD は画像中の画素の勾配方向が揃っている箇所をクラスタリングすることで、線分の候補を抽出し、さらに各候補について勾配方向の揃い方が統計的に有意なものを抽出して精度の高い線分検出を行う。実時間 SLAM[10]の研究にも用いられており、高速かつ誤検出の少ない方法である。これを車載カメラ画像に適用した結果を示す(図 4.4)。青い線は検出した線分を表し、白線や横断歩道に加え、側溝や縁石からも線分が抽出できていることがわかる。この結果から多くの路面上の手がかりは、線分検出によって抽出可能であると考えられ、自己位置推定に有望な特徴といえる。またそれぞれ線としての構造が保存されており、地図との対応付けが良い。なおカメラは車両に水平についているものとし、明らかに路面ではない一定以上の高さの検出処理は行わなかった。また自車両の映り込みの線分は画像中の位置で判定して除くものとした。

前方カメラ画像



後方カメラ画像



図 4.4 マルチカメラ画像における線分検出.

4.5.画像中の線分の鳥瞰変換

車両設計においては、カメラの構成が車種によって大きく異なることが多い。このためカメラ構成が変更されても後段の処理に影響が出ないことが望ましい。そこで本研究では、複数のカメラから検出された線分を車両座標系に変換して扱う。このために画像中の線分を鳥瞰変換によって車両座標系へ射影し、その誤差分布を検討する。 n 番目のカメラの取り付け位置の回転 \mathbf{R}_{vc_n} 、並進 \mathbf{t}_{vc_n} およびカメラの内部パラメータは事前に与えられているものとする。また路面は車両近傍において平面であることを仮定する。これにより線分検出によって得られる線分の開始点 \mathbf{n}_s 、終了点 \mathbf{n}_e は、鳥瞰変

換 \mathbf{b} によって車両座標系における $\mathbf{q}_s, \mathbf{q}_e$ に変換される． $\mathbf{n}_s, \mathbf{n}_e$ は正規化画像上での位置である． 開始点と終了点の変換は同じ扱いであるので， $\mathbf{q} = \mathbf{b}(\mathbf{n})$ と添字を省略すると，変換式は次式で与えられる．

$$\mathbf{q} = \mathbf{b}(\mathbf{n}) = \begin{bmatrix} \frac{-n_v t_z (n_u r_{33} - r_{13}) + t_y (n_u r_{33} - r_{13}) + n_u t_z (n_v r_{33} - r_{23}) - t_x (n_v r_{33} - r_{23})}{(n_v r_{31} - r_{21})(n_u r_{33} - r_{13}) - (n_u r_{31} - r_{11})(n_v r_{33} - r_{23})} \\ \frac{-n_v t_z (n_u r_{31} - r_{11}) + t_y (n_u r_{31} - r_{11}) + n_u t_z (n_v r_{31} - r_{21}) - t_x (n_v r_{31} - r_{21})}{(n_v r_{33} - r_{23})(n_u r_{31} - r_{11}) - (n_u r_{33} - r_{13})(n_v r_{31} - r_{21})} \end{bmatrix} \quad (4.1)$$

但し，次に設定される．

$$\begin{aligned} \mathbf{q} &= [q_x, q_z]^T \\ \mathbf{n} &= [n_u, n_v]^T \\ \mathbf{R}_{vc_n} &= \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \\ \mathbf{t}_{vc_n} &= [t_x, t_y, t_z]^T \end{aligned} \quad (4.2)$$

4.5.1. 鳥瞰変換の誤差モデル

画像中から線分を検出について，車両周辺では高解像度に撮影されるため多くの画素が使われるのに対し，遠方では少数の画素しか使われない．このため車両周辺に比べて遠方は，観測誤差が大きいことが直感的に理解できる．そこで線分の鳥瞰変換後の誤差モデルを，線分の検出位置や長さに対応づけて検討した．まず画像上における検出位置の誤差の大きな要因にはレンズがあり，一般に中心から離れるに従い誤差が大きくなると考えられる．あらかじめ歪み補正[2]を行っていたとしても，誤差が残ることがある．そこで画像上での検出誤差を画像中心からの位置の2乗によって表現する．また線分を抽出する際には，複数の画素が使われており，長い線分ほど正確であると考えられる．このため正規分布におけるサンプル数と誤差の関係を参考に，線分の画像上での長さ l の平方根に標準偏差が反比例するとした．これらをまとめ画像上での端点の誤差の共分散行列 \mathbf{R}_l を次の対角行列として表現した． c_1, c_2 は定数である．

$$\mathbf{R}_l = \text{diag}[(c_1 n_u^2 + c_2)^2, (c_1 n_v^2 + c_2)^2] / l \quad (4.3)$$

さらに鳥瞰変換後の車両座標系における共分散行列 \mathbf{R}_q を，ヤコビ行列 $\partial \mathbf{b} / \partial \mathbf{n}$ による1次の近似を行い次の式で示す．

$$\mathbf{R}_q = \frac{\partial \mathbf{b}}{\partial \mathbf{n}} \mathbf{R}_l \frac{\partial \mathbf{b}^T}{\partial \mathbf{n}} \quad (4.4)$$

ここで鳥瞰変換の式をそのままヤコビ行列を計算するのは，式が複雑であるため演算

量が多い。また車両にカメラはほぼ水平に取り付けられていることを利用し、次のより簡単な式で近似する。

$$\mathbf{R}_q \approx \frac{\partial \mathbf{b}'}{\partial \mathbf{n}} \mathbf{R}_l \frac{\partial \mathbf{b}'^T}{\partial \mathbf{n}}$$
$$\mathbf{b}'(\mathbf{n}) = \begin{bmatrix} t_y n_u / n_v \\ t_y / n_v \end{bmatrix} \quad (4.5)$$

この場合のヤコビ行列は次で計算される。

$$\frac{\partial \mathbf{b}'}{\partial \mathbf{n}} = \begin{bmatrix} t_y / n_v & -t_y n_u / n_v^2 \\ 0 & -t_y / n_v^2 \end{bmatrix} \quad (4.6)$$

4.5.2.鳥瞰変換の結果

画像線分の鳥瞰変換の例を示す(図 4.5)。黒線は地図を表し、青線は鳥瞰変換した画像線分、赤楕円は線分端点の誤差楕円を表している。画像中で手前側かつ中心付近に検出した線分は誤差が小さく、遠方の線分は誤差が大きくなっているのが見て取れる。なお共分散が大きい線分は、自己位置の補正で利用しないように閾値処理した。これにより誤差の大きい画像周辺の線分や、短い線分が優先的に取り除かれる。

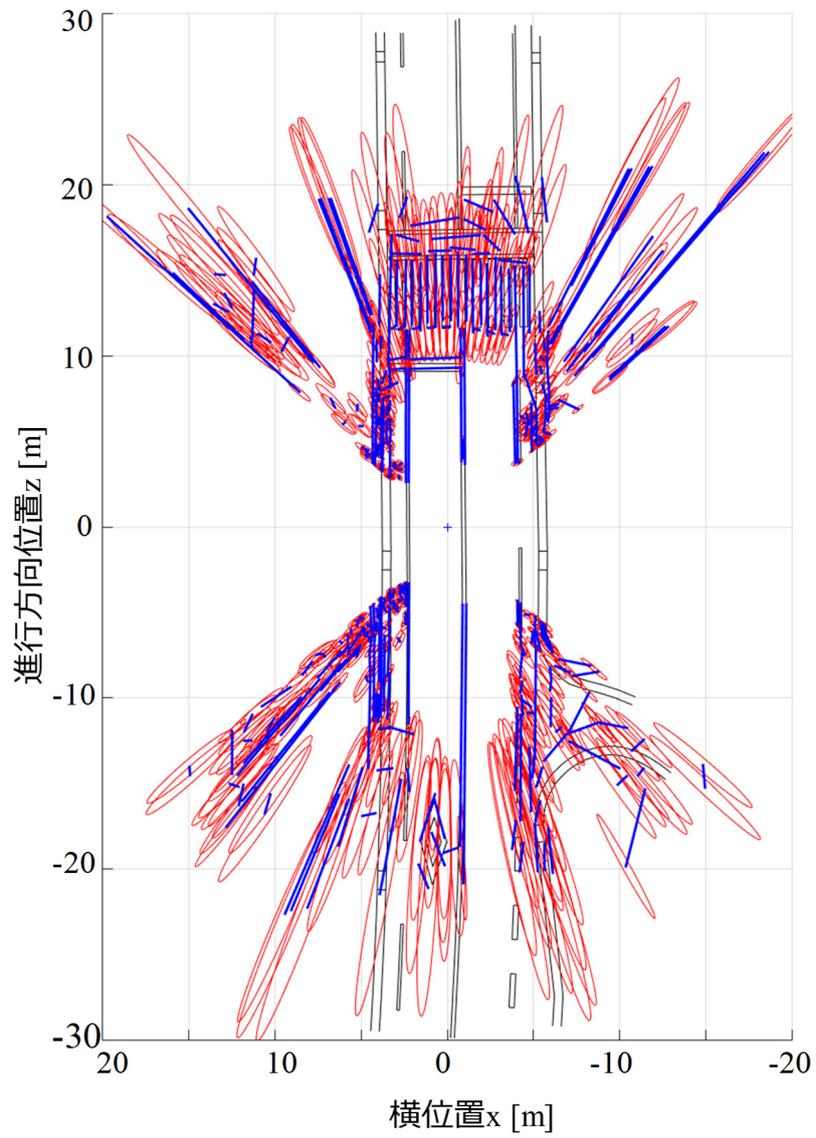


図 4.5 線分の鳥瞰変換と観測誤差.

4.6.地図線分と画像線分の対応付け

自己位置の補正のため画像線分 $\mathbf{q}_s, \mathbf{q}_e$ と、地図線分 $\mathbf{m}_s, \mathbf{m}_e$ を対応付ける処理を説明する．車両座標系における線分端点は，前章で説明した車両の運動モデルによる予測状態 $\bar{\mathbf{x}}^k$ によって世界座標系へ変換され，近傍にある地図線分と対応付けることができる．しかしながら，白線などは両側のエッジに線分が検出されるため，単純に近傍と対応付けると，誤ったエッジに対応付けられる可能性がある．そこで線分の両側の濃淡から勾配の向きを計算し，地図線分と濃淡の方向を一致させる(図 4.6)．これにより誤対応を大きく減らすことができる．まとめると①画像線分から地図線分への射影距離が閾値以下であること，②角度差が閾値以下であること，③勾配方向が同じ方向であることを条件とした．なお線分は途切れて検出されることがあるため，一つの画像線分に対し，条件を満たす複数の地図線分が対応付けされる．さらに誤った対応付けを除くため，各線分の対応付けについて射影距離と角度の両方でロバスト推定の一種である Bisquare の M 推定[47]を行った．具体的な対応付けの例を示す(図 4.7)．赤線は地図線分，青線は画像線分であり，緑線はその対応付けを表している．水色の円は次節で説明する観測誤差 R_p を表している(本来は 1 次元であるが，表記のため，正円で表現している)．単純に射影距離が小さいだけでは誤対応するようなケースでも，正しく対応付けができていることが分かる．

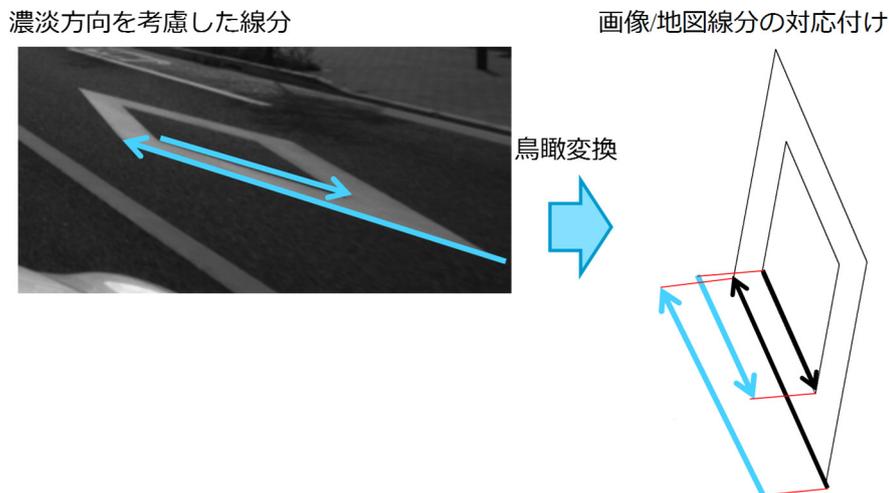


図 4.6 勾配方向を考慮した対応付け

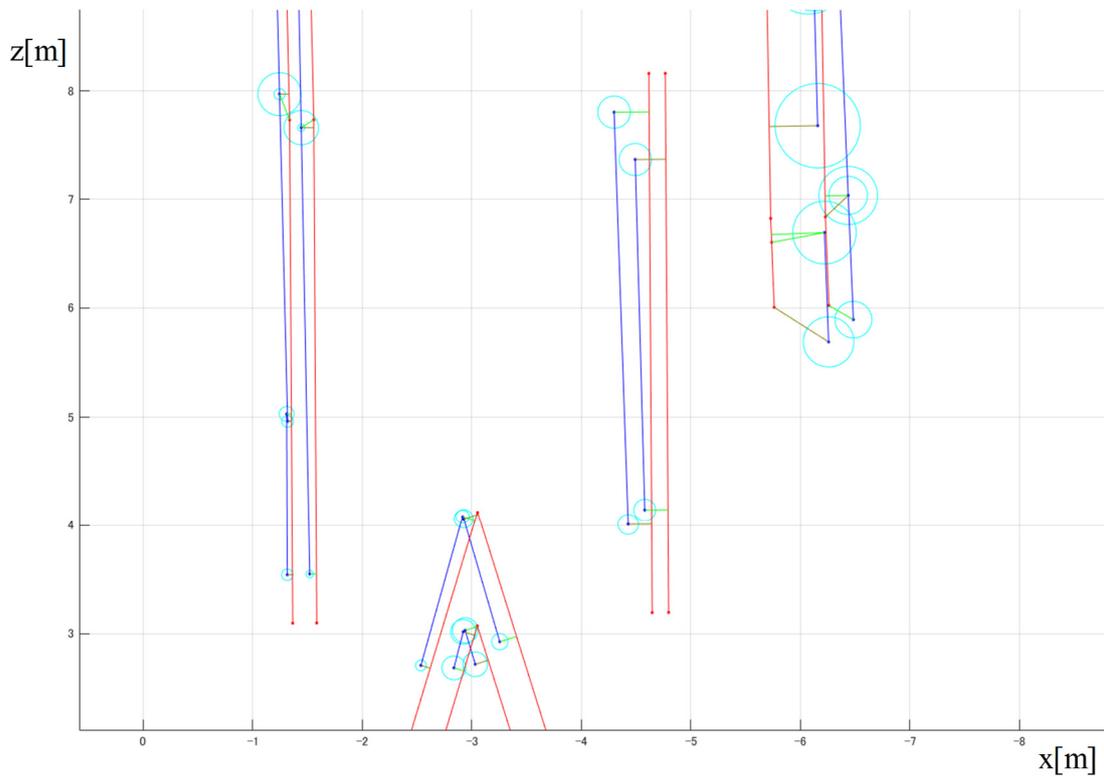


図 4.7 対応付けの例

4.7. 自己位置の誤差モデル

対応付けられた画像線分 $\mathbf{q}_s, \mathbf{q}_e$ と、地図線分 $\mathbf{m}_s, \mathbf{m}_e$ の間の誤差を定義し、自己位置の補正を行う。線分検出の特徴として、その方向が多くの画素から計算されるため正確であるのに対し、端点は途切れやすく正確ではない課題がある。特にセンターラインなどの長い線が画像中で部分的に観測された場合、それがセンターラインのどの部分を示しているか不明である。そこで図 4.8 で示す二つの基準で誤差を定義した。

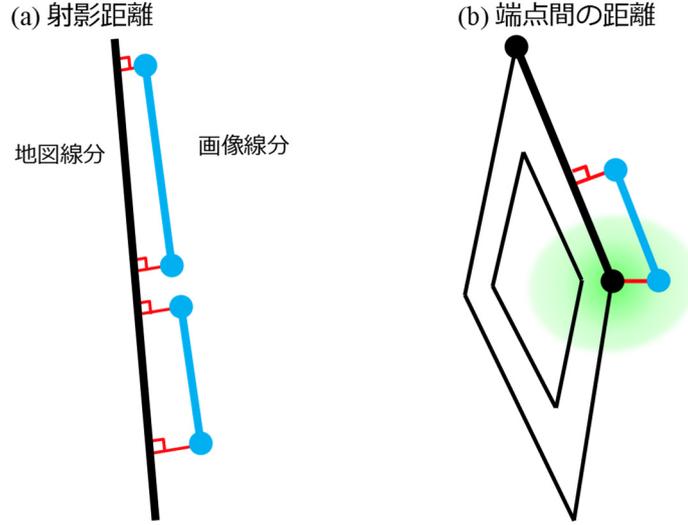


図 4.8 地図線分と画像線分の誤差

まず基本的な誤差として、画像線分の端点を地図線分に射影した距離を用いる(図 4.8a)．主には車両左右の白線や縁石、車両と直交する停止線などの手がかりに対応する．地図線分 $\mathbf{m}_s, \mathbf{m}_e$ から計算される直線の方程式を $a_x x_w + a_z z_w + a_c = 0$, $\sqrt{a_x^2 + a_z^2} = 1$ としたとき、画像線分の端点 \mathbf{q} の射影誤差 δ_p^k は次の式で表される．

$$\delta_p^k = [a_x \quad 0 \quad a_z \quad a_c] \begin{bmatrix} \mathbf{R}_{slope}^k \mathbf{R}_{vw}^k \mathbf{q} + \mathbf{t}_{vw}^t \\ 1 \end{bmatrix} \quad (4.7)$$

この観測誤差は前述の共分散行列 \mathbf{R}_q に基づき、地図線分の垂線に射影した R_p として計算できる．

$$R_p = [a_x \quad a_z] \mathbf{R}_q [a_x \quad a_z]^T \quad (4.8)$$

次に自己位置の予測誤差が小さい場合、地図線分の端点と画像線分の端点は近い位置にくることが期待される．そこで端点間の距離が閾値以下にある場合には、端点間の距離も誤差に含めた(図 4.8b)．これにより破線の白線や横断歩道の端点などから、車両の進行方向の位置を補正する手がかりが得られる．観測誤差には \mathbf{R}_q を使い、 $\mathbf{m}_s, \mathbf{m}_e$ のどちらかの端点を \mathbf{b} と置き、次の式で誤差を定義する．

$$\delta_q^k = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} (\mathbf{q} - \mathbf{R}_{vw}^k \mathbf{R}_{slope}^k \mathbf{b} - \mathbf{t}_{vw}^k) \quad (4.9)$$

4.8.自己位置に対する観測誤差と線形化

これらの誤差モデルに基づき、複数の線分の対応付けから得られる誤差を並べた誤

差 $\boldsymbol{\delta}^k = [\delta_{p,1}^k, \delta_{p,2}^k, \dots, \boldsymbol{\delta}_{q,1}^{k,T}, \boldsymbol{\delta}_{q,2}^{k,T}, \dots]^T$ と、共分散行列 $\mathbf{R}^k = \text{diag}([R_{p,1}, R_{p,2}, \dots, \mathbf{R}_{q,1}, \mathbf{R}_{q,2}, \dots])$ を定義し、誤差が 0 になるように、自己位置を推定する。フレームレートが十分に早ければ、自己位置は予測位置から大きくはずれない。このために誤差関数を予測位置周辺で線形化する拡張カルマンフィルタによって補正できる。

射影誤差 δ_p はヤコビ行列 \mathbf{H}_p^k を使って次のように線形化される。

$$\begin{aligned} \delta_p^k(\boldsymbol{\mu}^k, \mathbf{q}, \mathbf{a}) &= \mathbf{a}^T \cdot [(\mathbf{R}_{slope}^k \mathbf{R}_{vw}^k \mathbf{q} + \mathbf{t}_{vw}^k)^T, 1]^T \\ &\approx \delta_p^k(\bar{\boldsymbol{\mu}}^k, \mathbf{q}, \mathbf{a}) + \mathbf{H}_p^k(\boldsymbol{\mu}^k - \bar{\boldsymbol{\mu}}^k) \end{aligned} \quad (4.10)$$

ヤコビ行列 \mathbf{H}_p^k は次のように計算される。

$$\mathbf{H}_p^k = \begin{bmatrix} \frac{\partial \delta_p^k}{\partial x^k} & \frac{\partial \delta_p^k}{\partial z^k} & \frac{\partial \delta_p^k}{\partial \theta^k} \end{bmatrix} \quad (4.11)$$

但し、次のように計算できる。

$$\begin{aligned} \frac{\partial \delta_p^k}{\partial x^k} &= a_x \\ \frac{\partial \delta_p^k}{\partial z^k} &= a_z \\ \frac{\partial \delta_p^k}{\partial \theta^k} &= a_x l_x + a_z l_z \end{aligned} \quad (4.12)$$

$$\mathbf{l} = \begin{bmatrix} l_x \\ l_y \\ l_z \end{bmatrix} = \mathbf{R}_{slope}^k \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ -1 & 0 & 0 \end{bmatrix} \mathbf{R}_{vw}^k \mathbf{q}$$

同様に端点の誤差 δ_q^k もヤコビ行列 \mathbf{H}_q^k を使って次のように線形化される。

$$\begin{aligned}\delta_q^k(\boldsymbol{\mu}^k, \mathbf{q}, \mathbf{b}) &= \mathbf{I}_{23} \{ \mathbf{q} - \mathbf{R}_{vw}^k \mathbf{R}_{slope}^{kT} (\mathbf{b} - \mathbf{t}_{vw}^k) \} \\ &\approx \delta_q^k(\bar{\boldsymbol{\mu}}^k, \mathbf{q}, \mathbf{b}) + \mathbf{H}_q^k (\boldsymbol{\mu}^k - \bar{\boldsymbol{\mu}}^k)\end{aligned}\quad (4.13)$$

具体的には次のように展開できる。

$$\begin{aligned}\delta_q^k(\boldsymbol{\mu}^k, \mathbf{q}, \mathbf{b}) &= [\delta_{qx}^k, \delta_{qz}^k]^T \\ \delta_{qx}^k &= q_x - \cos \theta^k (r_{11}(b_x - x^k) + r_{31}(b_z - z^k)) \\ &\quad + \sin \theta^k (r_{13}(b_x - x^k) + r_{33}(b_z - z^k)) \\ \delta_{qz}^k &= q_z - \sin \theta^k (r_{11}(b_x - x^k) + r_{31}(b_z - z^k)) \\ &\quad - \cos \theta^k (r_{13}(b_x - x^k) + r_{33}(b_z - z^k))\end{aligned}\quad (4.14)$$

但し次のように定義した。

$$\begin{aligned}\mathbf{R}_{slope}^k &= \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \\ \mathbf{b} &= \begin{bmatrix} b_x \\ b_y \\ b_z \end{bmatrix}\end{aligned}\quad (4.15)$$

ヤコビ行列 \mathbf{H}_q は次のように計算できる.

$$\mathbf{H}_q^k = \begin{bmatrix} \frac{\partial \delta_{qx}^k}{\partial x^k} & \frac{\partial \delta_{qx}^k}{\partial z^k} & \frac{\partial \delta_{qx}^k}{\partial \theta^k} \\ \frac{\partial \delta_{qz}^k}{\partial x^k} & \frac{\partial \delta_{qz}^k}{\partial z^k} & \frac{\partial \delta_{qz}^k}{\partial \theta^k} \end{bmatrix} \quad (4.16)$$

但し, 次のように計算できる.

$$\begin{aligned} \frac{\partial \delta_{qx}^k}{\partial x^k} &= +r_{11} \cos \theta^k - r_{13} \sin \theta^k \\ \frac{\partial \delta_{qz}^k}{\partial x^k} &= +r_{11} \sin \theta^k + r_{13} \cos \theta^k \\ \frac{\partial \delta_{qx}^k}{\partial z^k} &= +r_{31} \cos \theta^k - r_{33} \sin \theta^k \\ \frac{\partial \delta_{qz}^k}{\partial z^k} &= +r_{31} \sin \theta^k + r_{33} \cos \theta^k \\ \frac{\partial \delta_{qx}^k}{\partial \theta^k} &= +\sin \theta^k (r_{11}(b_x - x^k) + r_{31}(b_z - z^k)) \\ &\quad + \cos \theta^k (r_{13}(b_x - x^k) + r_{33}(b_z - z^k)) \\ \frac{\partial \delta_{qz}^k}{\partial \theta^k} &= -\cos \theta^k (r_{11}(b_x - x^k) + r_{31}(b_z - z^k)) \\ &\quad + \sin \theta^k (r_{13}(b_x - x^k) + r_{33}(b_z - z^k)) \end{aligned} \quad (4.17)$$

4.9. 自己位置推定での利用

前節で説明した観測誤差を, 3.3 節にて説明した拡張カルマンフィルタによる自己位置の推定の問題に当てはめ, 自己位置を推定した. その結果は 8.1 節, 8.2 節にて説明する.

5. 曖昧さを含む地図の観測モデル

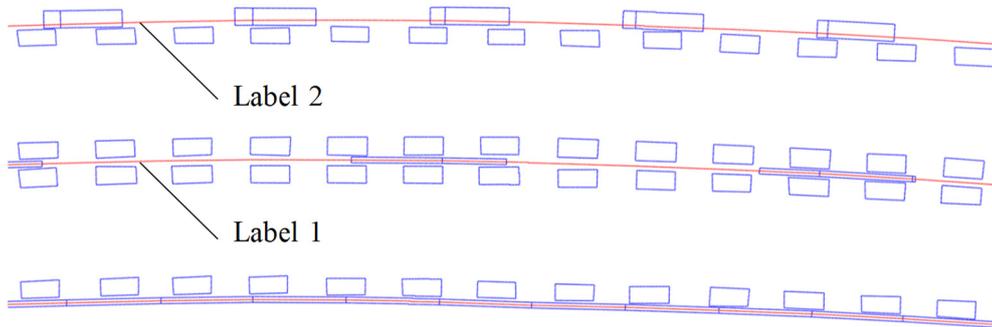
本章では人手による記載の少ない地図を使って自己位置を推定するための、観測モデルを提案する。具体的には走行レーン地図から得られる曖昧な白線種別と、画像上での見えを確率的に対応付けることで精度を上げる方法を提案する。このためにまず白線種別と白線形状の関係を調査し、次に調査結果を使った白線形状の確率分布を定義する。白線形状をパーティクルとしてサンプルすることで画像中の白線を検出し、自己位置を補正する方法を説明する。

5.1. 白線種別と白線形状の関係の調査

地図と画像を対応付けるには、画像上での見えと一致するような仕様で、地図の白線が記述されていることが好ましい。特にコントラストの高い白線と路面の境界を表す白線形状が一意に決まれば、自己位置を決める手がかりになる。しかしながらコストやデータ容量の制約から、形状を簡略化した白線種別が割り当てられていることがある。白線種別を表すラベルには「実線」「点線」「細い」「太い」「白色」「黄色」、「減速標示」とその位置、「2重線」「3重線」といった表現が用いられる。この白線種別の設計が白線形状を正確に決める仕様になっていることが望ましいが、往々にして曖昧さを含む。このため白線種別から想定した見えと、実際の画像上での見えの違いが自己位置推定の誤差の原因となることがある。

そこで白線種別と白線形状の関係をモデル化する。まず首都高速 C1 環状線内回りの約 14[km]の道路について、白線形状の詳細な調査を行い、形状が簡略化された走行レーン地図と比較した。調査では白線の形状について白い領域を囲むように、四角形で記述した地図を作成した。走行レーン地図と白線の形状の実際の違いを示す(図 5.1a)。図の例では上記の白線種別を表すラベルより、Label 1 は「両側に減速標示付きの細い白線」、Label 2 は「右側に減速標示付きの太い白線」が指定されている。矩形の線は白線の形状を表し、中心の線は走行レーン地図の線を表している。ここで白線種別ごとの白線断面に着目する。白線の断面をスライスし、白領域である確率を計算した(図 5.1b)。具体的には白線種別ごとにランダムに 10,000 箇所 of 白線断面を集め、k-means クラスタリングによって分割し、白線種別 l に対する断面パターン s の生起確率 $p(s|l)$ を求めた。Label 1 では断面によって 3 通りに分類されることがわかる。つまり白線種別を元に画像上でパターンマッチングを行う際には、この分類に対応する必要がある。また Label 2 では約 0.15[m]も中心位置がずれていることがわかる。これは 0.20[m]の目標精度に対して無視できない量といえる。

(a) 走行レーン地図と詳細な白線形状



(b) 白線の断面パタンの生起確率

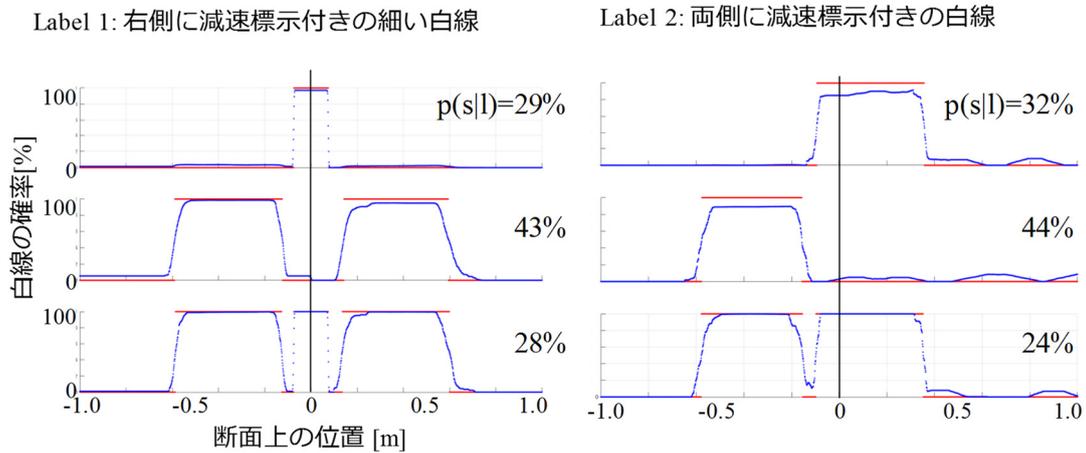


図 5.1 走行レーン地図における断面パタンの確率

5.2. 白線断面の観測方法

推定した自己位置 \mathbf{x}^k が正しく、かつ地図の白線種別 l から得られる候補より正しい断面パターン s が選択されていれば、走行レーン地図を画像に射影した白線と、実際に撮影された白線は重畳され、正規化相互相関などの指標が高く計算される。しかしながら、自己位置は誤差をもち、また正しい断面も不明である。そこでパーティクルフィルタに倣い、確率表現された白線断面、および自己位置をサンプリングし評価した。具体的には次の4つの手順で計算する(図 5.2)。

① 自車前方の地図の白線上の1点を画像上で一様になるようにサンプルし、その点の白線種別 l より $p(s|l)$ に従って断面パターンをサンプリングする。具体的には $i \in \{1, \dots, I\}$

番目のサンプルとして自車前方 30[m]の範囲の地図の白線から 1 点を選ぶ. このとき地図上で一様に選ぶと画像に射影したときに遠方に集中してしまい, 自車近傍の高解像度の領域にサンプルが少なくなる. このため画像上で一様になるようにサンプルする. 断面パタンのサンプルには前節で求めた分布を用いた.

② 自己位置をサンプルし, 断面パターンを地図座標系から車両座標系を通じて画像へ射影する. 具体的には次節で説明するオドメトリに基づいて自己位置の予測分布を求め, 分布からサンプルした自己位置を用いる. 自己位置が決まると白線断面を車両座標系, および画像上に射影できる. 特に車両座標系における白線断面の中心点を $\mathbf{q}_i^k = [q_{x,i}^k, q_{y,i}^k, q_{z,i}^k]^T$ と置き, 後で用いる.

③ 射影された白線断面と, 撮影画像を参照して正規化相互相関 κ_i^k を計算する.

④ 閾値処理により相関の高いパーティクルを抽出する. 具体的には正規化相互相関が 0.5 以上としたときの閾値処理を行った.

この方法を評価するため, 図 5.1 の位置で(a)全ての白線を最も頻出する「細い白線」と設定した場合と, (b)本研究が提案する断面のモデルを利用した場合を比較した. それぞれ誤った白線の中心を捕らえた個数を人手でカウントしたところ, 誤り率は(a)が 50.3%, (b)が 16.1%であり, 大きく効果があることがわかった.

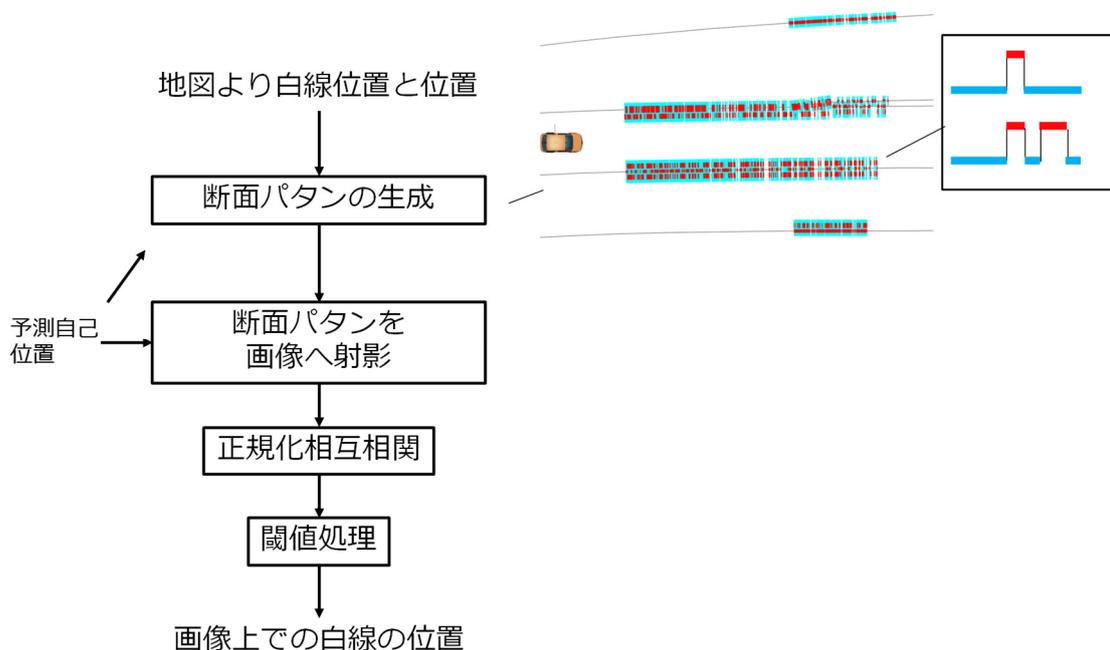


図 5.2 断面パタンのパーティクルによる白線検出

5.3. 白線の観測による誤差モデル

車両座標系における白線位置の観測 \mathbf{q}_i^k を，地図座標系において対応する白線に射影した距離を誤差として補正する．観測 \mathbf{q}_i^k はカメラから見て路面の観測であるため，前章で説明した鳥瞰変換の誤差モデル \mathbf{R}_q^k をもつとした．地図座標系の $j \in \{1, \dots, J\}$ 番目の白線を表す直線の方程式を $n_x^j x + n_y^j y + d^j = 0$ ， $\mathbf{n}^j = [n_x^j, n_y^j, 0]^T$ ， $\|\mathbf{n}^j\|_2 = 1$ とする． i 番目の観測点 \mathbf{q}_i^k に対応した地図の白線のインデックスを $r_i^k \in \{1, \dots, J\}$ とし，その直線パラメータを $\mathbf{n}^{r_i^k}, d^{r_i^k}$ とする． i 番目の観測に対応した射影誤差 δ_i^k は次の式で表現される． $w(\kappa_i^k)$ は正規化相互相関 κ_i^k が 0.8 を超えたら 1，それ以外は 0 を出力する．つまり正規化相互相関が低いサンプルは無視されるとした．

$$\delta_i^k = w(\kappa_i^k) \{ \mathbf{n}^{r_i^k T} (\mathbf{R}_{slope}^k \mathbf{R}_{vm}^k \mathbf{q}_i^k + \mathbf{t}_{vm}^k) + d^{r_i^k} \} \quad (5.1)$$

この誤差の共分散行列は \mathbf{q}_i^k と $\mathbf{R}_{q,i}^k$ より，一次の近似により次で表現できる．

$$R_i^k = \frac{\partial \delta}{\partial \mathbf{q}} \mathbf{R}_i^k \frac{\partial \delta^T}{\partial \mathbf{q}} \quad (5.2)$$

これをまとめて観測誤差として扱った．

$$\begin{aligned} \boldsymbol{\delta}^k &= [\delta_1^k, \dots, \delta_I^k]^T \\ \mathbf{R}^k &= \text{diag}([R_1^k, \dots, R_I^k]) \end{aligned} \quad (5.3)$$

5.4. 自己位置推定での利用

前節で説明した観測誤差を，3.4 節にて説明した非線形最小二乗問題による自己軌跡の推定の問題に当てはめた．その結果は 8.3 節にて説明する．

6. 走行レーン地図の自動生成

前章までに走行レーン地図を使った自己位置推定の方法を述べた。しかしながら、走行レーン地図にはコストがかかる課題がある。そこで本章においては、地図作成コストの中で支配的である人手による図化の作業を自動化する方法を提案する。具体的には MMS によって収集された 3 次元点群から、勾配画像処理を利用して自動的に線分地図を生成する方法を説明する。

6.1. 地図の自動生成の課題

MMS のレーザスキャナから得られる 3 次元点群から、地図を生成する上での課題を説明する。MMS による計測は、他車両などによる遮蔽を避けるため往路・復路などレーンを変えながら複数回計測し、それらの点群を重ねて用いるのが一般的である。このため 3 次元点群は数が膨大であり、直接にエッジ部分を求めるのが困難であることが挙げられる。例えば図 6.1 は 30x30[m] の区間を MMS による 3 回の走行と 2 機のレーザスキャナによって収集された点群である。この点群の総数は 732 万点である。これに直接に 3 次元的なエッジ抽出方法を当てはめて有効なエッジを探すような方法では演算時間が膨大になる。一様にサンプルすることで点群を減らすこともできるが、エッジ等の重要部分において形状が鈍ってしまう。

次に点群には他車両や歩行者、道路周辺構造物などの路面以外の物体が含まれていることが上げられる(図 6.1)。これをそのまま用いると誤った地図が生成されるため路面のみの点群の抽出が必要である。

また白線を検出する際に複数回の計測を重ねるには反射強度が異なることや、点群の疎密があるなど、均質なセンシングでない課題がある(図 6.2)。上記複数回の計測を重ねた時、計測ごとにレーザスキャナから対象物までの距離や向きが異なることから、反射強度が違って検出されることがあり、反射強度を直接用いる抽出方法では対応が難しい課題がある。

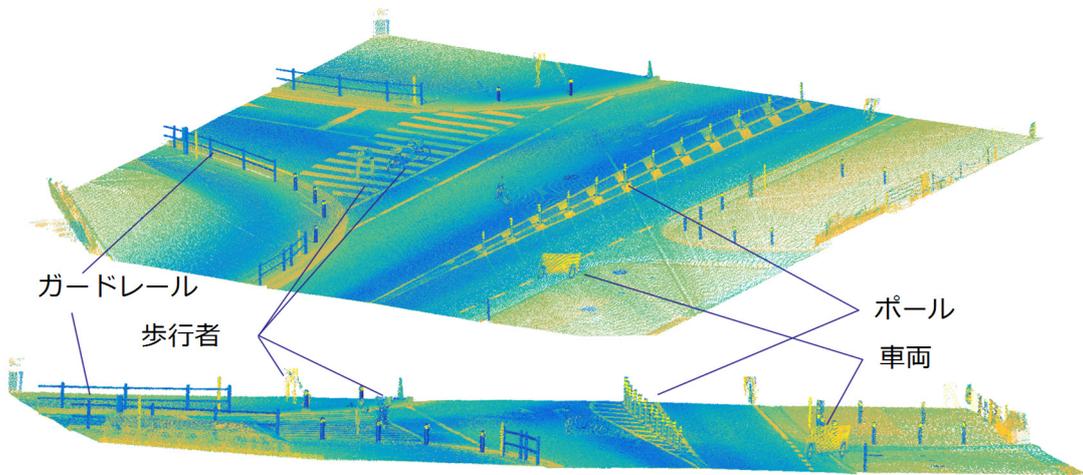


図 6.1 路面と路面以外の点群

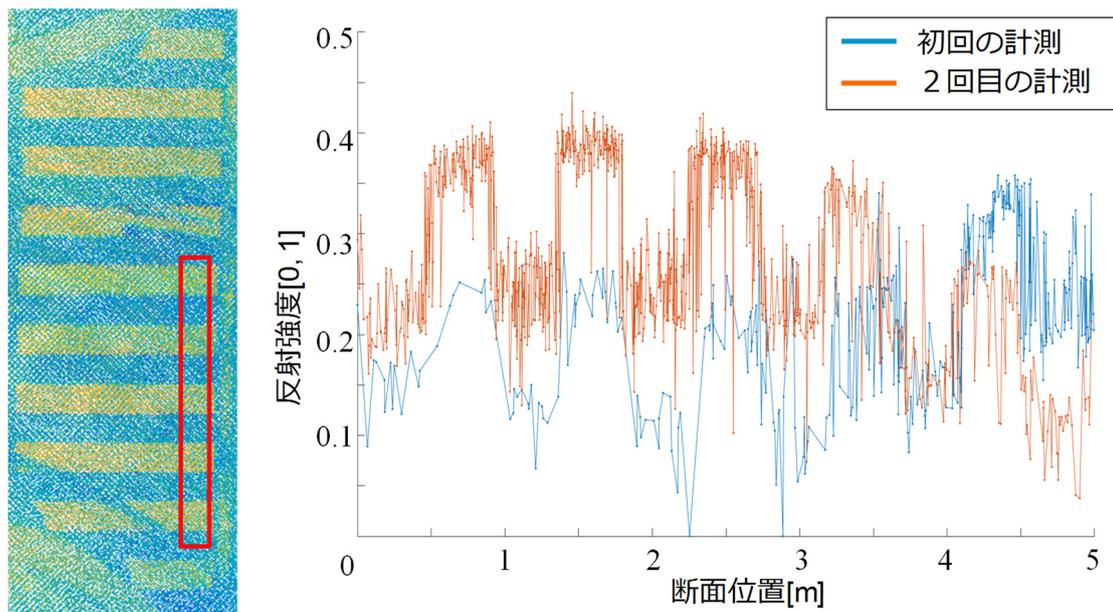


図 6.2 白線の複数回計測

6.2. 勾配画像への変換と線分抽出による地図生成

前節で説明したように地図の自動図化には、点群が多く演算時間が膨大になること、

計測ごとに反射強度が異なることや疎密があること、路面以外の対象物が含まれることの課題があった。そこで3次元点群から直接にエッジを求めるのではなく、勾配画像を生成して画像処理によって線分を検出する方法を提案する。また路面の点群の抽出には、複数回の計測結果を利用する方法を提案する。白線のような反射強度から得られる地図と、縁石のような高さから得られる地図の生成は、ほぼ同様の処理で解決される。具体的な自動生成の処理の手順を示す(図 6.3)。まず3次元点群から、路面のみの点群を抽出する。次に反射強度もしくは高さから勾配画像を生成し、線分検出を行う。この線分を閾値処理し、分割処理した領域を結合し、さらに地図全体を見渡した縁石外領域の除去を行って、最終的な地図線分とした。次節より各ブロックの詳細を説明する。

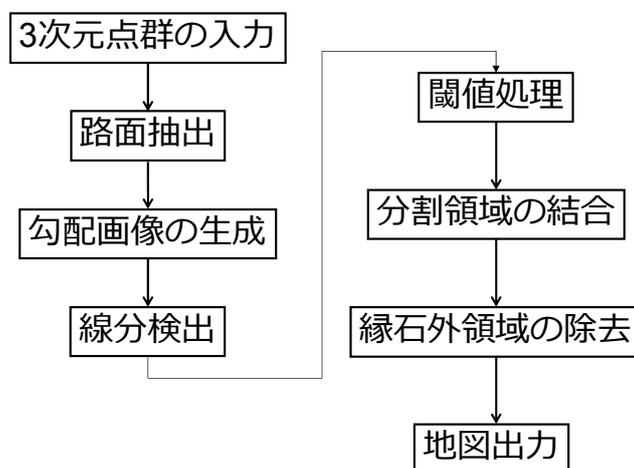


図 6.3 地図自動生成の流れ

6.3.MMS による路面計測の詳細

MMS による路面計測では，車線を変えながら複数回，また複数のレーザスキャナによって計測する．これにより他車両などによる遮蔽を避けられることや，また点群の密度を上げることができる．本研究で用いた MMS 計測では，車両後部に 2 機のレーザスキャナが取り付けられている(図 6.4)．これにより車両が走行すると道路を輪切りにするようにメッシュ状の 3 次元点群が得られる．さらにレーンを変えながら複数回の計測を行う．地図を作る際に測量車両が走行した軌跡の例を示す(図 6.5)．複数の軌跡を重ねると，その車両位置決め誤差によって点群が十分に重ならないことが考えられるが，測量車両には RTK-GPS や高精度な IMU によって位置決めがなされており，無視して良い水準であった．図 6.6 に複数の軌跡，および複数のレーザスキャナで収集した 3 次元点群を示す．点群の着色は反射強度を表している．軌跡とレーザスキャナによって同一の道路でも異なる点群が得られている．

複数回の計測の性質をまとめると次の 4 点になる．

- ① 計測タイミング，および計測方向が異なるため，他車両などの移動体の遮蔽の有無が異なる．
- ② MMS から計測対象までの距離が異なるため，点群の反射強度が異なる．
- ③ MMS から計測対象までの距離が異なるため，点群の疎密が異なる．
- ④ 点群は 30x30[m]の空間で数百万点と数が膨大である．

次節以降で説明する路面点群の抽出では①の性質を利用し，勾配画像を用いた地図生成は②～④の課題に対応するものである．

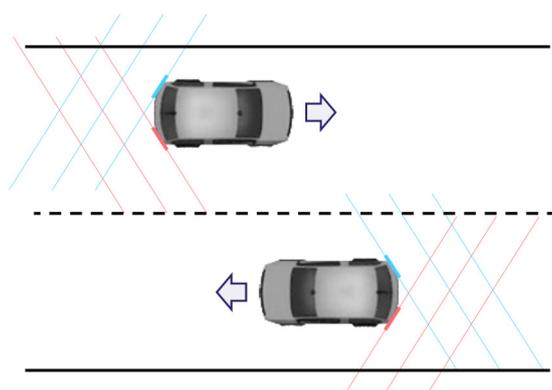


図 6.4 複数のレーザスキャナによる計測

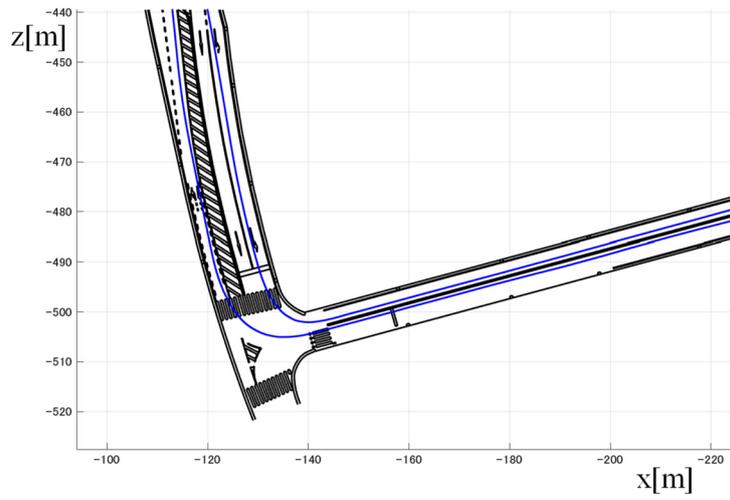


図 6.5 車線を変えた複数回の道路計測
(青点が車両の走行軌跡を表す)

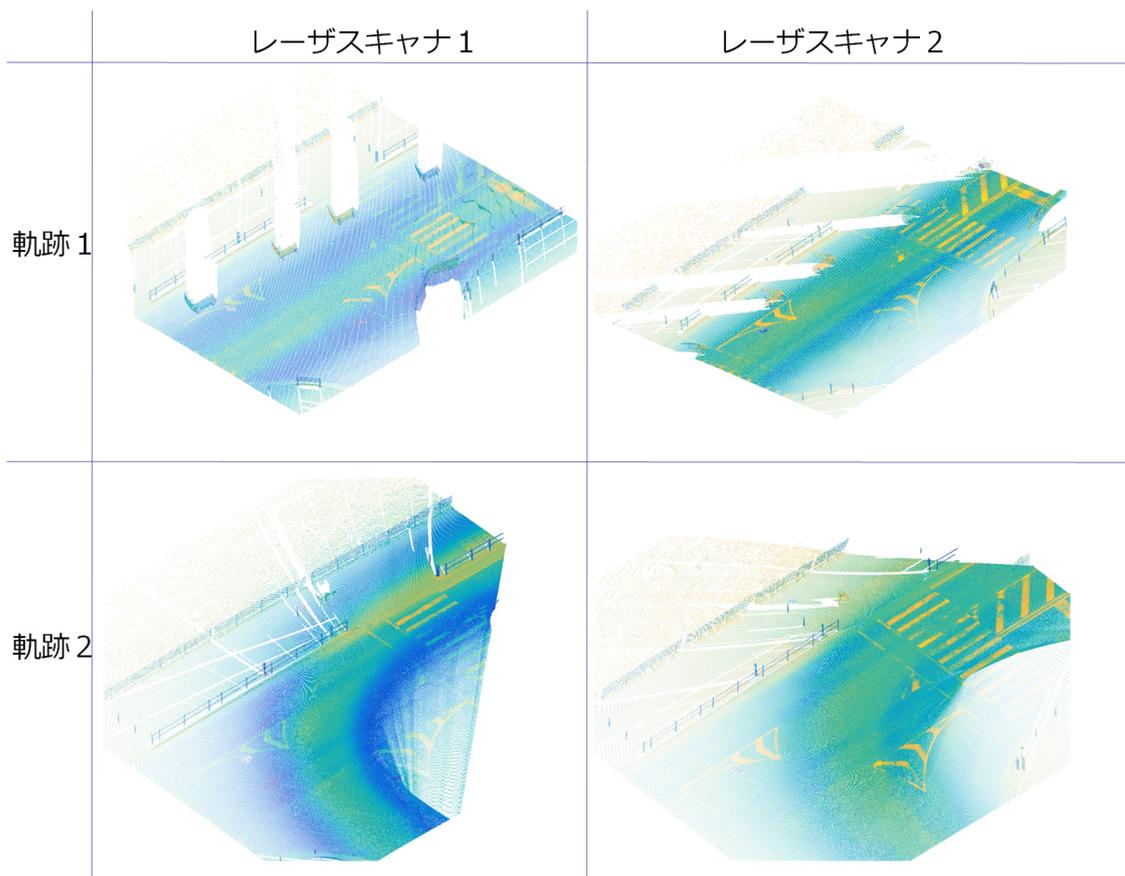


図 6.6 複数の軌跡とレーザスキャナごと計測結果

6.4.路面点群の抽出

MMS から得られる 3次元点群には、他車両や歩行者、道路周辺構造物などの路面以外の対象物が含まれており、そのまま利用すると誤った地図が生成される。このため路面の点群のみを抽出する処理が必要となる。特に他車両は直方体状であるため、長い線として抽出されやすく、その除去は重要である。そこで路面が局所的に平面である性質を利用し、路面を表す点群のみを分離する処理を行う。なお記載した数値は主には国土交通省による道路技術基準の体系に基づく寸法[51]を参考に、レーザスキャナの性質、演算量とあわせて決めている。

点群の各点には 3次元の位置と反射強度が含まれている。位置は測量で良く用いられる平面直角座標系[53]で表現される。緯度経度を用いると高緯度地域と低緯度地域の補正が問題になるが、平面直角座標系ではエリアごとに平面に近似して処理できるため、通常単位がメートルで表現された直交座標系で扱うことができる。反射強度はレーザスキャナのメーカーによって決められる数字であり 0~65,535 の値をとる。ここでは 0.0~1.0 に規格化して用いる。ここで計測領域は 25[mm]間隔のグリッドを区切って処理するものとした。一度に 30 x 30[m]の領域を処理するのであれば、1,200 x 1,200[個]のセルに分割される。

上述の 25x25[mm]に区切られた各セルについて、周囲が平面であることを仮定して路面を分離する。まず中心からグリッドの解像度より広い 500[mm]の矩形範囲の点群を集め、各点の高さ Y と、 XZ 平面での座標(x, z)の関係を、平面で近似する(図 6.7)。平面の推定にはロバスト推定の一環である最小メジアン法を用いる。次に平面近似の結果を使って中心セルに所属する点群が地図生成に利用できる路面や縁石・壁面を判定する。この判定には次の五つの基準を適用する。

- ① 平面から 50mm 以上離れた点群を路面外とした(図 6.8a)。
- ② ①で路面外とされた点群の割合が 10%を超える場合、全ての点群を路面外とした。
- ③ 平面の傾きが 20 度を超える場合、全ての点群を路面外とした。
- ④ 平面が 30m x 30m の領域の各平面の高さのメジアン値から大きく異なる場合、全ての点群を路面外とした。
- ⑤ XZ 平面上で孤立した平面領域を画像処理のラベリング処理で求め路面外とした。

なお往路・復路など複数回の計測について、それぞれ遮蔽の原因となる他車両が路面上にいる場合といない場合があるため、路面点群の抽出処理は計測ごとに処理することに注意する。

次に縁石や壁面の点群を残すため、抽出した平面領域を 50mm 膨張させた領域の点群を再判定する。このために複数回の計測で個別に抽出した平面を利用する。図 6.8b のケース 1 では、他の計測回で計測した平面の領域上に点群が載っている。このような場合は縁石とはみなさず棄却する。ケース 2 は縁石でよく見られ、二つの平面に挟まれている点群を残す。ケース 3 では平面に含まれない領域の点について、250mm ま

での高さの点群を壁面として残す. 図 6.9 にこれらの処理によって路面抽出した結果を示す. 図中では処理例を示すため A-D 地点を設定した. A,B は複雑な道路ペイントや他車両の映り込みがある計測を選択した. C,D は交差点で曲がった計測を選択した.

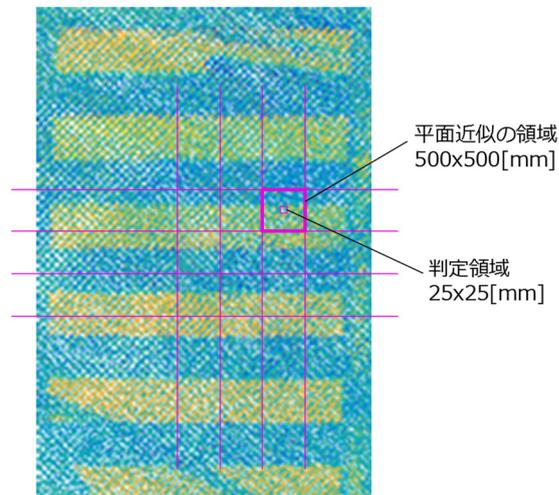


図 6.7 グリッドの縮尺イメージ

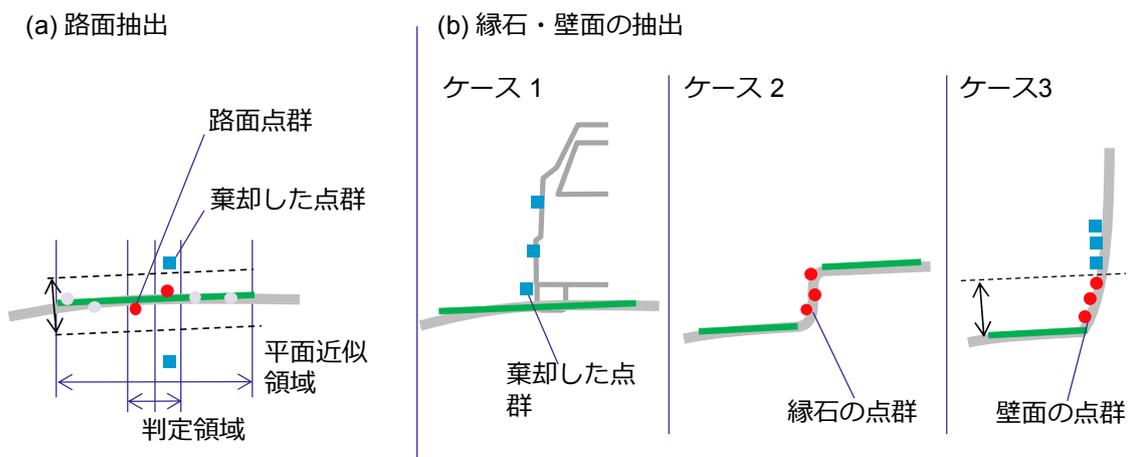
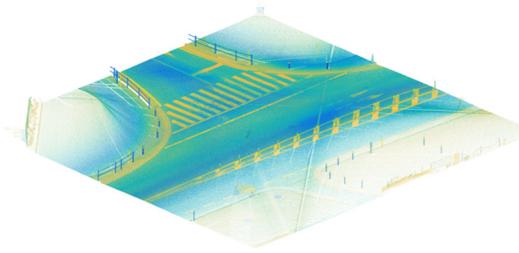
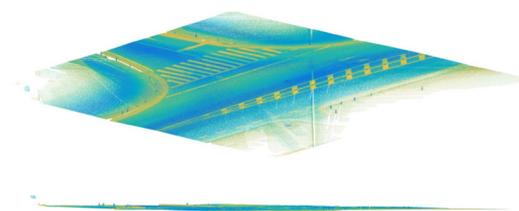


図 6.8 各ケースにおける路面点群抽出

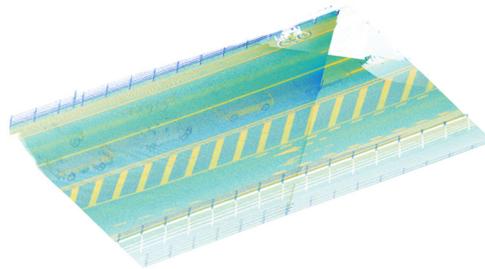
処理前-A



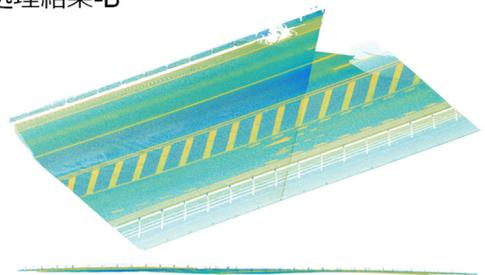
処理結果-A



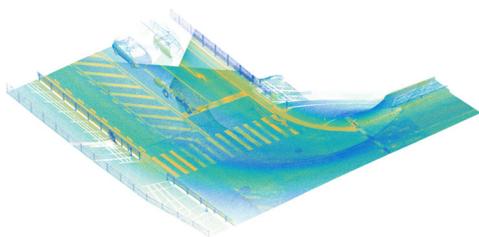
処理前-B



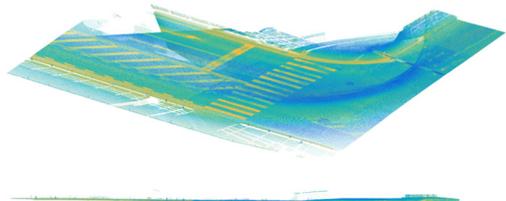
処理結果-B



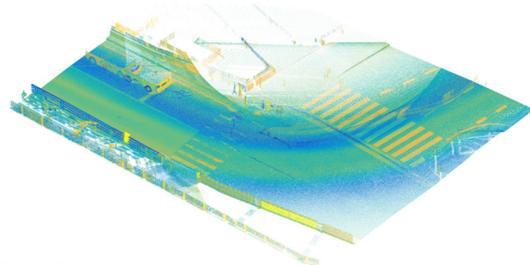
処理前-C



処理結果-C



処理前-D



処理結果-D

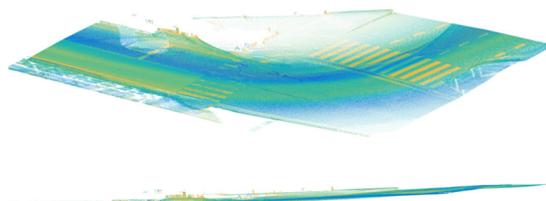


図 6.9 レーザ点群から路面表面の抽出

6.5. 反射強度の勾配画像の生成

前節の処理により，地図生成に必要な路面の点群のみを抽出することができた．次にこの路面の点群から，地図を構成する要素である白線や道路標示，横断歩道などをあらかず道路ペイントを検出する．このような構成要素は点群の3次元位置だけでは抽出できない．しかしながら路面の点群は3次元座標に加え，反射強度が付加されている．図 6.1 における明るい色は反射強度が高い領域を表している．反射強度はセンサ固有の処理によって 0 から 1 の値をとる．そこで，道路面内の反射強度の二次元分布から地図生成に必要な要素を検出することを考える．しかし MMS による複数回の計測では，反射強度や点群の密度が異なることがある．先に説明した図 6.2 は 2 回の計測について，横断歩道の点群の反射強度を抜き出してプロットしたものであるが，その値や密度が異なっていることが読み取れる．このため単純な反射強度に対する閾値処理のような方法では白線領域を抽出することは難しい．しかしながらどちらの計測回でも，白線領域とその他で反射強度に差があることから，反射強度の勾配情報は共通に利用できると思われる．そこで反射強度を画像の輝度値に見立てて，画像処理で用いられる勾配強度画像，勾配方向画像を生成する．路面抽出と同様に緯度経度に相当する XZ 平面でグリッドを区切り，各グリッドにおける点群の反射強度の勾配を計算する．ここでグリッドの解像度は地図の位置精度に影響するため高いほうが望ましい．しかしながら点群には疎密があるため，各グリッドに直接射影して画像を作るような方法では，点群の少ない空間が隙間になってしまい扱いが難しくなる．このため点群に対して局所的な多項式近似を行うことで勾配画像を生成する．路面抽出と同様に，25mm 間隔の各セルについて中心から 500mm の矩形範囲の点群を集める．集めた各点 i の反射強度 I_i と，XZ 平面での座標 $[x_i, z_i]^T$ の関係を，次の 3 次多項式で近似した．

$$\begin{aligned} I_i = & a_1 + a_2 x_i + a_3 z_i + a_4 x_i^2 + a_5 x_i z_i + a_6 z_i^2 \\ & + a_7 x_i^3 + a_8 x_i^2 z_i + a_9 x_i z_i^2 + a_{10} z_i^3 \end{aligned} \quad (6.1)$$

なお近似にはセル中心がより正確になるように，グリッド中心から点までの距離に対して標準偏差 50mm のガウス関数で重みをつけた．多項式の次数を 3 次にしたのは勾配を表現するため奇関数である必要があり，また 1 次では白線の角など，2 つの線分が接続される箇所では近似精度が落ちるためである．この関数を微分することで，グリッド中心の勾配強度 $\text{sqrt}(a_2^2 + a_3^2)$ と勾配方向 $\text{atan2}(a_2, a_3)$ が得られる．全てのセルについて勾配強度と勾配方向を計算した結果を示す(図 6.10)．白線や道路標示のエッジで勾配強度と勾配方向が推定されていることがわかる．

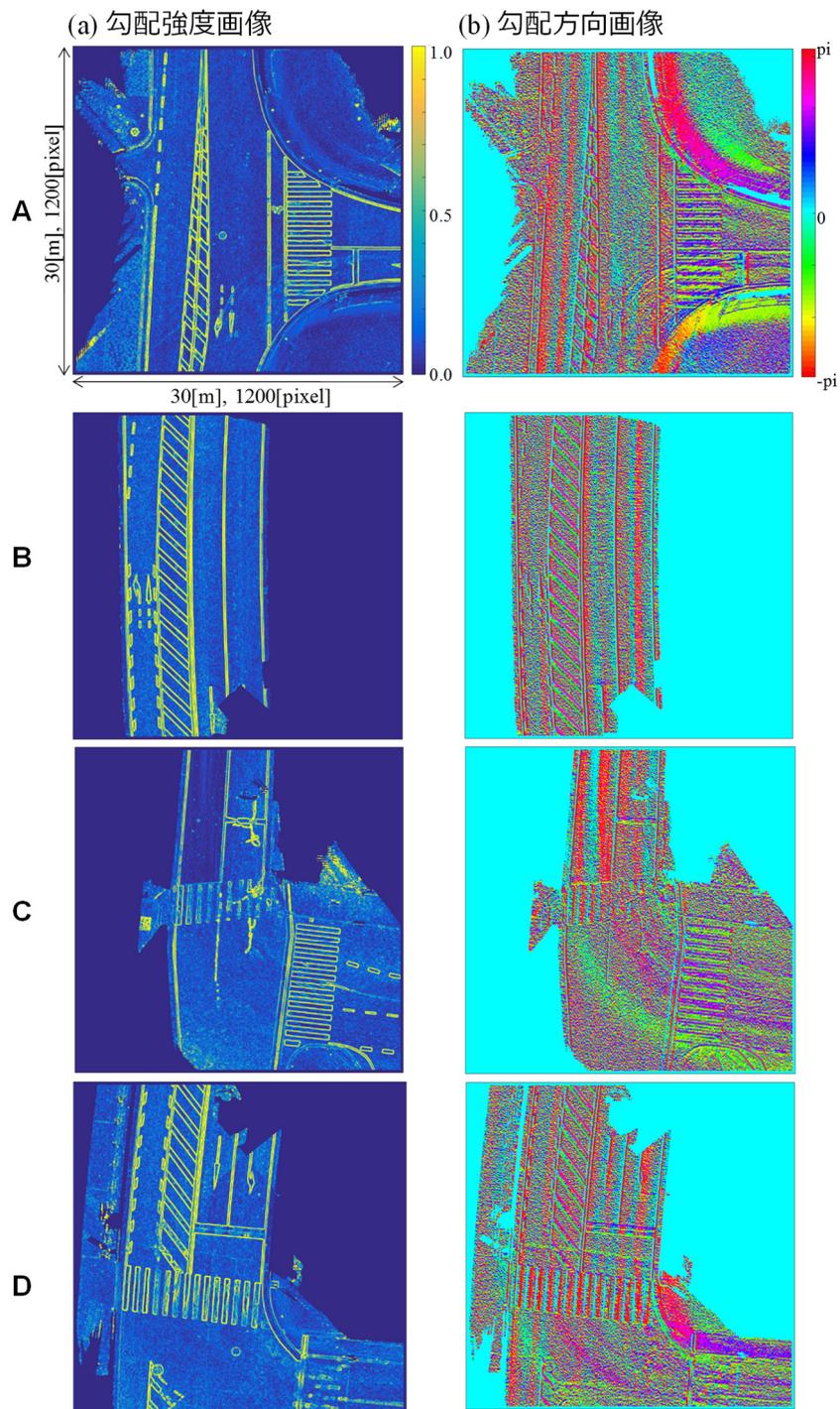


図 6.10 反射強度の勾配画像

6.6.地図線分の検出と高精度化

勾配強度と勾配方向の画像に対し，線分検出をかけることで白線，黄線，道路標示を検出する．線分検出には LSD[6]を用いた．LSD はグレースケール画像から勾配強度，勾配方向を計算し，勾配方向が揃っている領域を線分として抜き出す方法である．ここでは上記の点群から生成した勾配強度，勾配方向画像を直接用いて線分を検出した(図 6.11)．さらに得られた線分に対し，線分周辺 150mm の範囲の点群の高さを最小メジアン法による平面近似をかけることで線分の高さを抽出した．

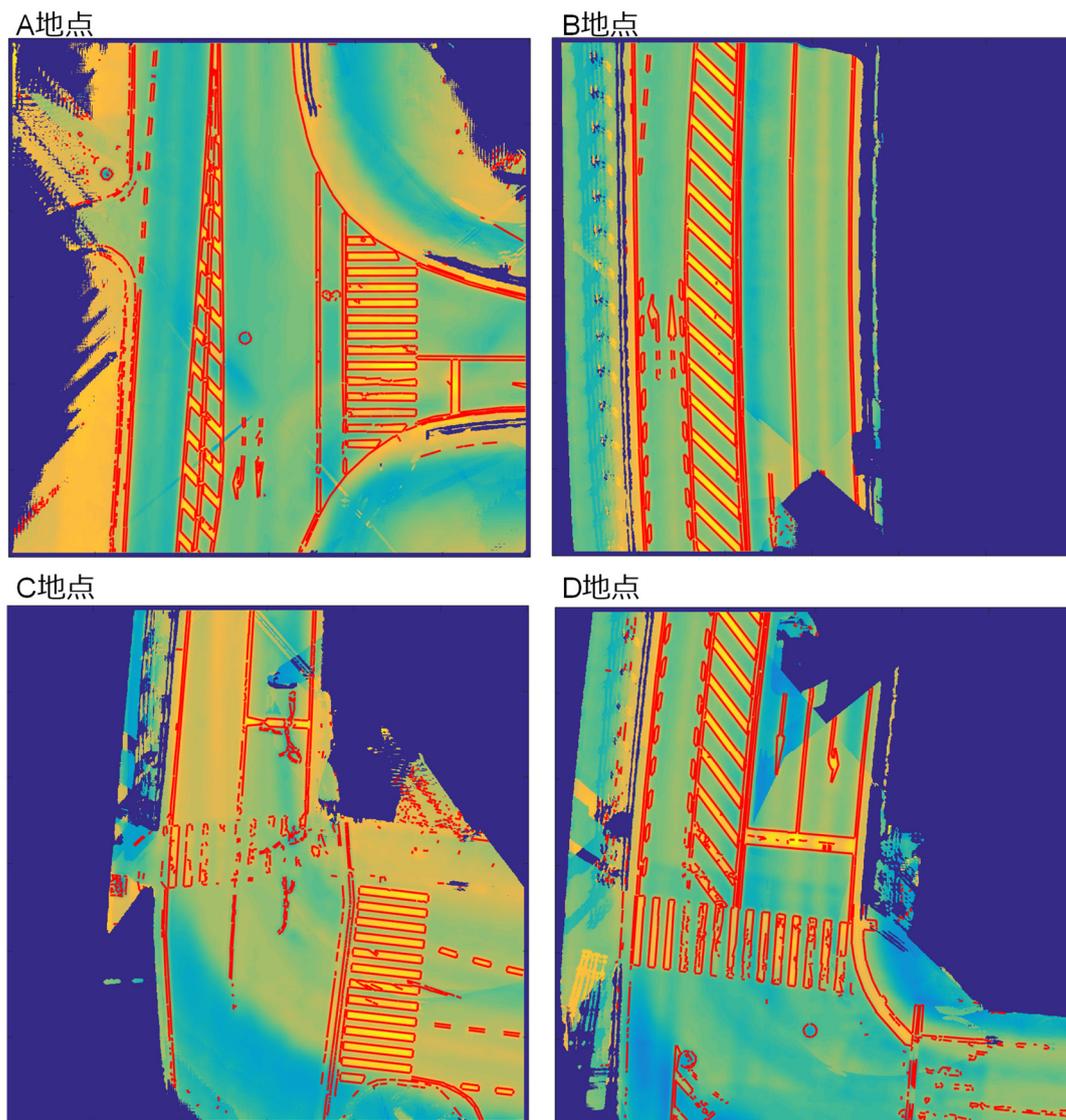


図 6.11 線分検出結果

LSD によって検出された線分には、多くの誤検出が含まれ、また勾配画像の解像度に依存した量子化誤差が発生する。また LSD の原理上、エッジ付近に同じ方向のグラデーションがあると位置がずれる事がある。このため再び点群と対応付けて位置の補正を行う。具体的には線分の周辺 100mm の範囲の点群に、シグモイド関数の近似を行った(図 6.12)。次の式より誤差 E が最小になるように反射強度のバイアス I_h 、反射強度の差 A 、回転の補正 θ 、法線方向の並進補正 x_h を計算した。定数 $k=200$ はシグモイド関数の傾きを表している。この結果より線分の位置の修正を行った。なお最適化を安定させるため、初期値は線分の中心と法線に従ってあらかじめ回転・並進しておくものとする。

$$E(I_h, A, \theta, x_h) = \sum_i (I_i - \bar{I}_i)^2 \quad (6.2)$$

$$\bar{I}_i = I_h + \frac{A}{1 + \exp\{-k((x_i - x_h) \cos \theta + y_i \sin \theta)\}} \quad (6.3)$$

また誤検出された線分を取り除くため、シグモイド関数の境界で点群の反射強度に差があることを利用する。これには平均値の差の t 検定を利用した。つまりシグモイド関数の中心を挟んで、平均値に差が無いと判定された線分は棄却される。これによりデータ点数が少ない箇所では反射強度の差が大きい線分が採用され、データ点数の多い箇所では反射強度の差が小さくとも採用される。図 6.13 に判定結果の例を示す。誤った線が取り除かれていることがわかる。

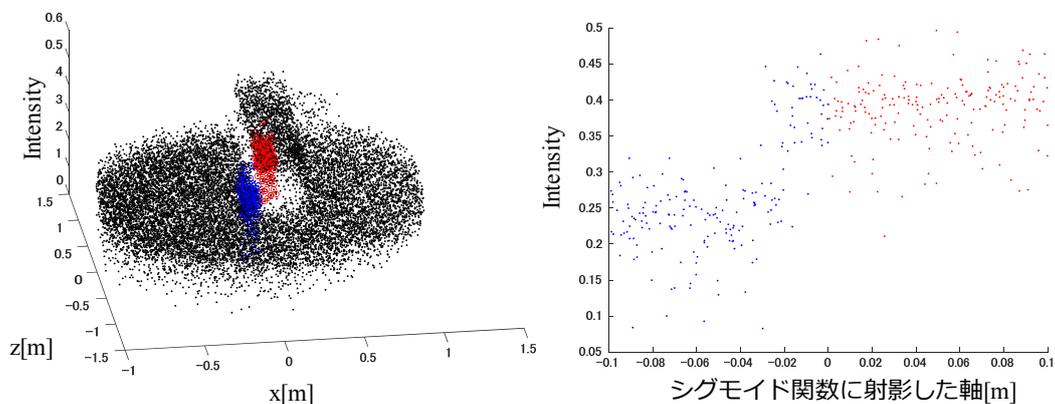


図 6.12 シグモイド関数による近似

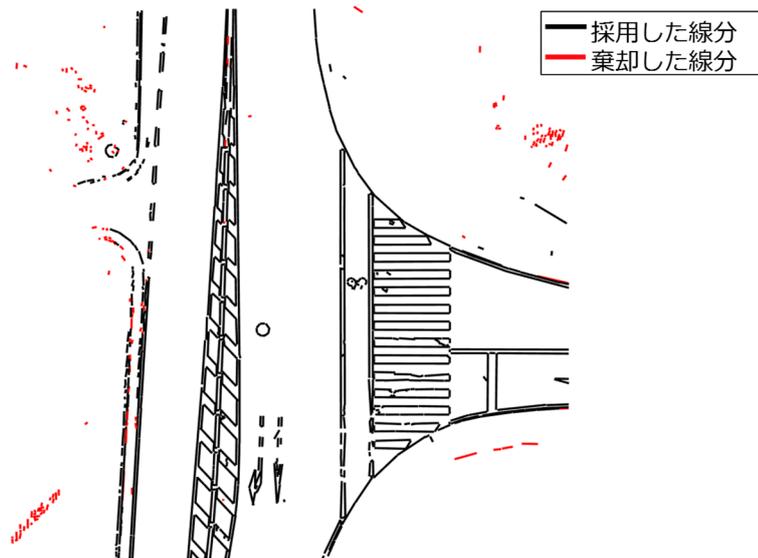


図 6.13 t 検定による閾値処理

6.7.縁石検出

縁石検出も上記道路ペイントと同様に，勾配画像の生成と線分検出によって地図を生成する．しかし，縁石は反射強度の勾配では検出が難しいので，点群の3次元位置を利用した高さの違いを利用するために，反射強度ではなく距離の勾配画像を生成する．縁石は道路ペイントに比べて単純な図形であるので近似多項式の次元は1次とした．図 6.14 に勾配画像の例を示す．勾配強度画像の中心にクロスしたような線が見えるが，これは他車両による遮蔽がある計測では遮られて見えなかった部分が，別の計測では見えることがありその境界で発生したものである．ここから誤検出される線分は，計測ごとの線分境界の点の密度の差を測ることで容易に取り除くことができる．また，縁石の高さは道路ペイントに合わせて下部の路面高さとし，シグモイド関数の近似によって求めた．また縁石の内側の線分を除くため，線分の両端点を計測経路への射影し，他の縁石を表す線分と交差する場合には除く処理を行った．図 6.15 に縁石処理の結果を示す．道路外に誤検出が多いが，これは後述の処理で緩和される．

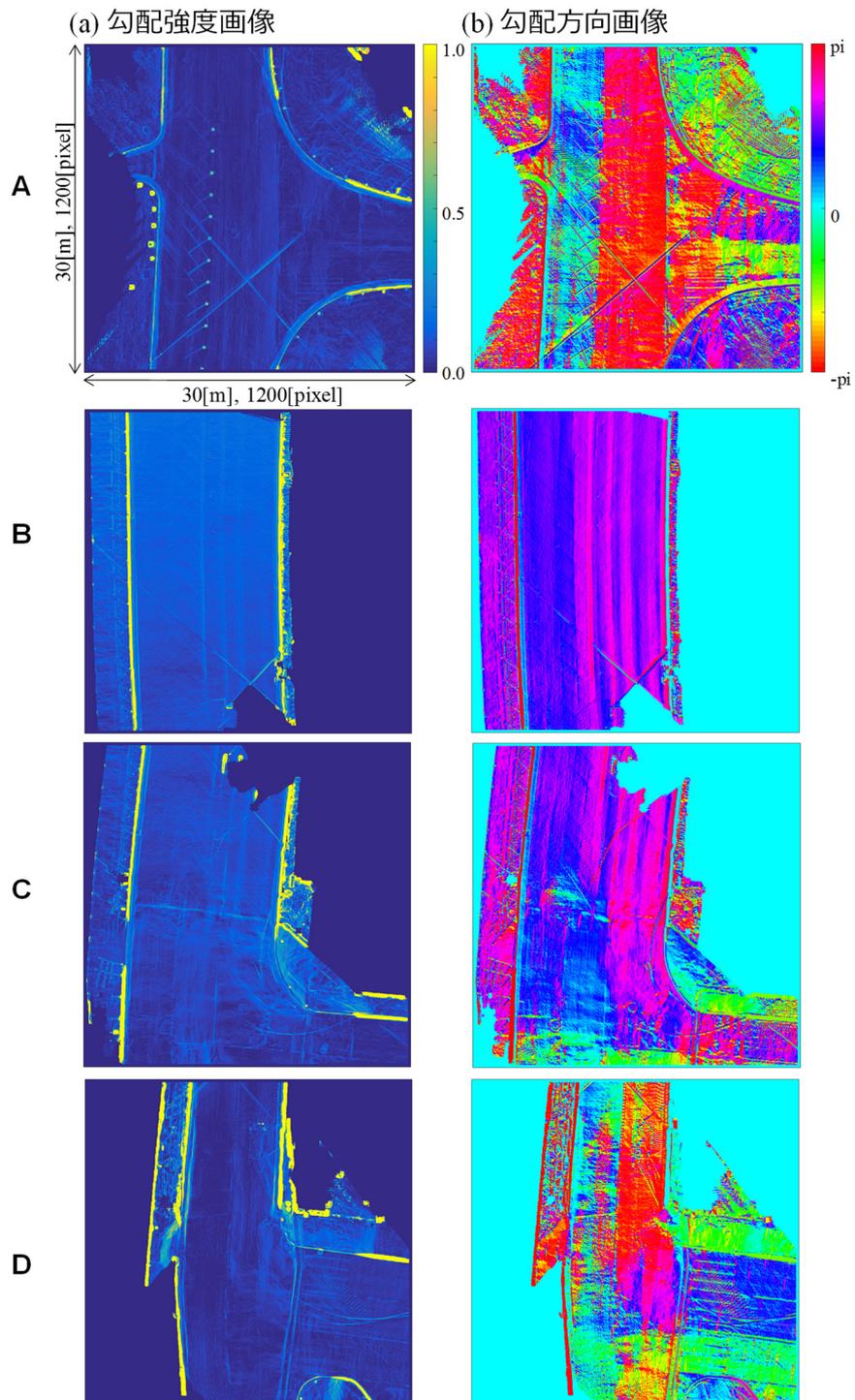
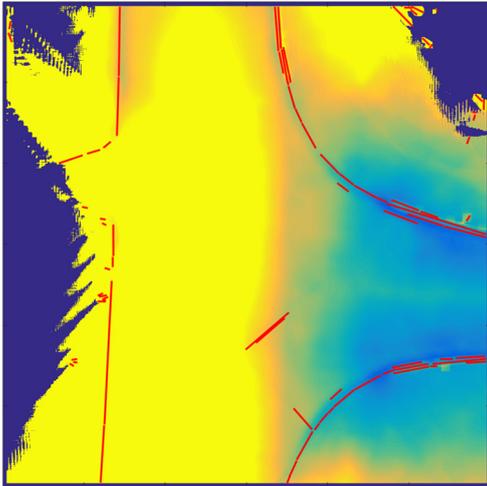
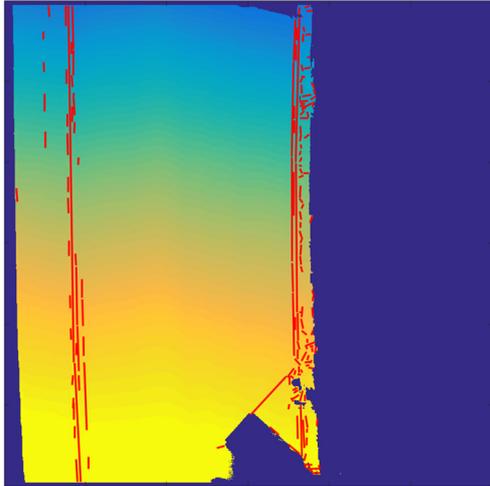


図 6.14 高さの勾配画像

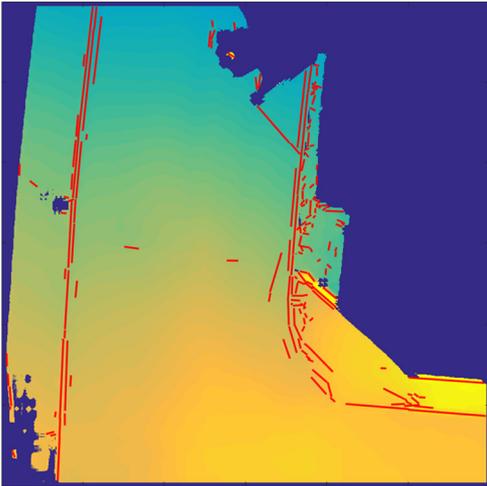
A地点



B地点



C地点



D地点

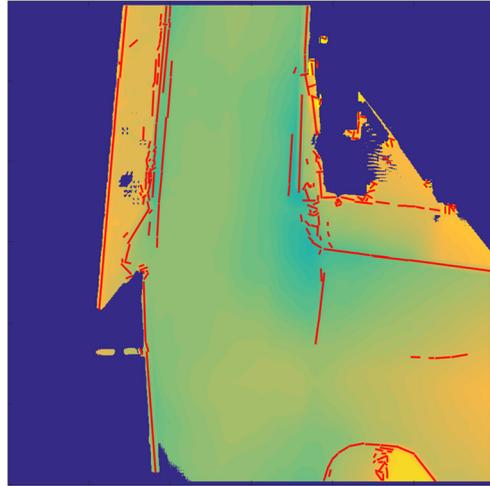


图 6.15 縁石検出

6.8. 処理領域の分割と結合

6.8.1. 処理領域の分割

MMS によって収集された点群は、前述の通り 30×30 [m]の区間で 732 万点と膨大であり、道路全体を一度に計算することはできない。このため処理領域を重複させながら分割して処理する。具体的には計測車両の走行軌跡を元に累積走行距離を計算し、10m 間隔で分割した。1 回の処理領域は 30m であるので大きく重複して処理される。図 6.16 に処理領域の位置の例を示す。青線は計測車両の走行軌跡であり、赤丸が処理領域の中心を表している。赤丸の位置ごとに 30m の処理領域を設定した。

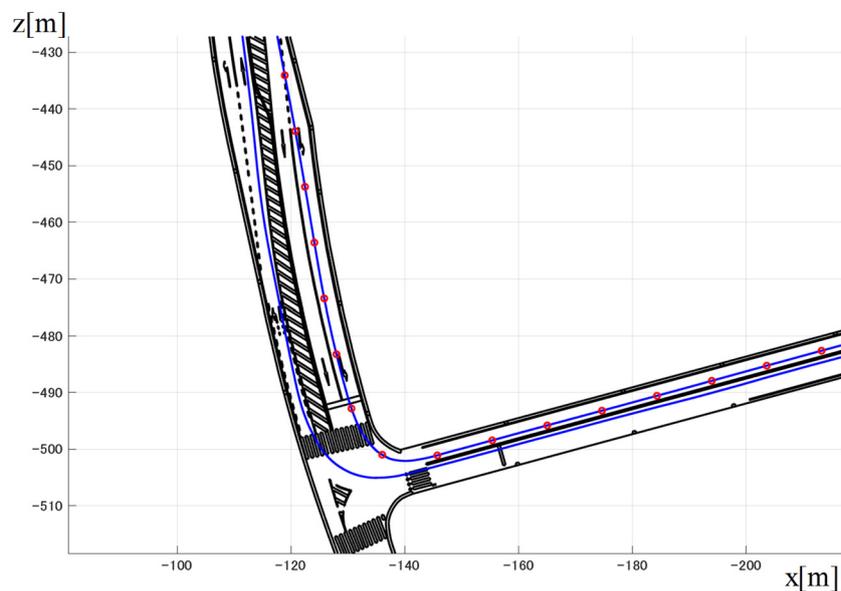


図 6.16 処理領域の分割

6.8.2. 重複領域の線分の処理

処理領域が重複する領域では、それぞれの領域から抽出された線分が得られる。これらの線分は同じ白線や縁石から得られたものであるため、同じ位置に検出されることが望ましい。しかしながら、処理領域が異なると LSD による画素のクラスタリング結果が異なるため、必ずしも同じ線が得られないことがある。また、境界付近では途切れるため、これを結合する必要がある。このため次の 2 つの処理で重複・分割された線を結合・削除し、一つの線に統合した(図 6.19)。

① 片方に含まれる線分の削除

ある線分の両端点から別の線分への射影距離 d_1 , d_2 が共に閾値(0.3m)より小さく、かつ、端点を射影した位置が線分内に含まれているときは削除する(図 6.17).



図 6.17 片方の線に含まれる線の削除

② 二つの線分の角度差が小さい場合の結合

二つの線分の角度差が小さく、かつ端点から射影した位置のうち、片方が含まれているときは、二つの端点の midpoint、もしくは線分が交差している場合は交点に移動し結合する(図 6.18)

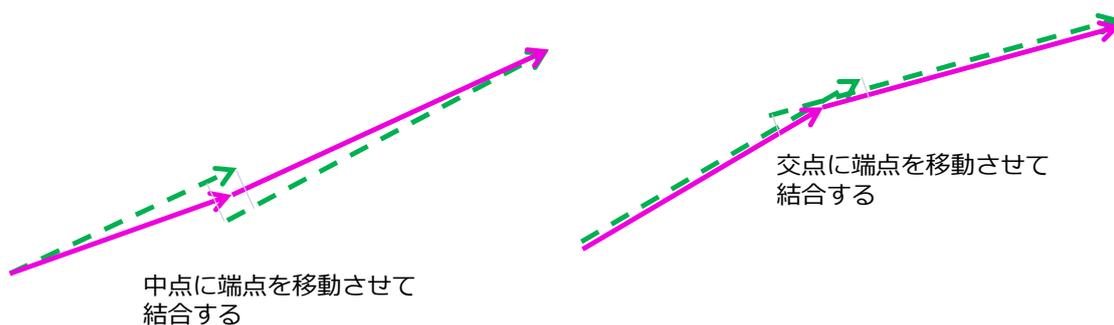


図 6.18 二つの線分の角度差が小さい場合の結合

処理前



処理後

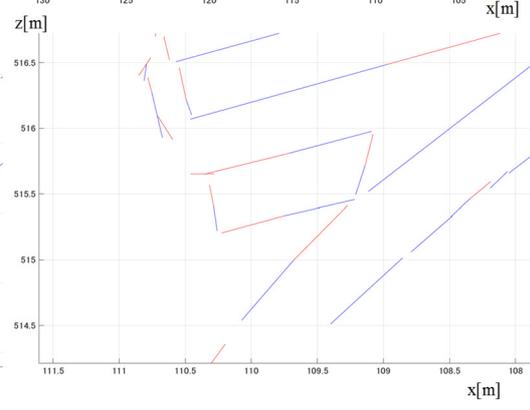
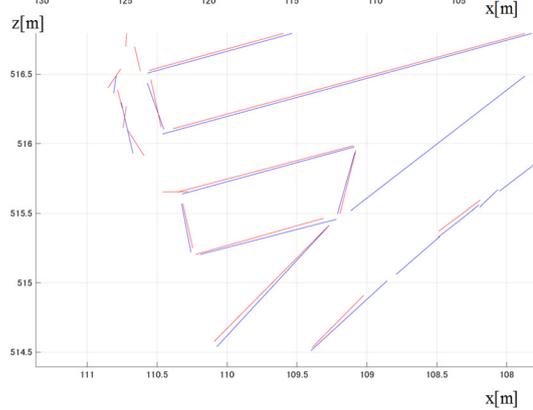


図 6.19 結合処理の結果
(下図は丸の部分の拡大図である)

6.9.縁石外領域の白線除去

白線，および縁石の検出において，レーザスキャナは道路領域外も計測しているため，道路領域外に不要な白線，縁石が検出されることがある．ここで白線のある道路領域は，縁石の内側と考えられ，また縁石が二重にあることはない．そこで図 6.20 で示すような次の二つの基準で縁石の外側の線分を取り除いた．

① 外側の縁石線分の除去

計測車両が走行した軌跡に対して，縁石線分の両端点を射影する線を求め，両端点とも，別の縁石線分と交差する線分を取り除いた．

② 縁石より外側の白線線分の除去

計測車両が走行した軌跡に対して，白線線分の両端点を射影する線を求め，両端点とも，縁石線分と交差する線分を取り除いた．

図 6.21 に処理結果の例を示す．青線は白線線分，赤線は縁石線分，緑線は取り除かれた白線線分，シアン線は取り除かれた縁石線分を表している．縁石外の多くの誤った線が取り除かれていることがわかる．

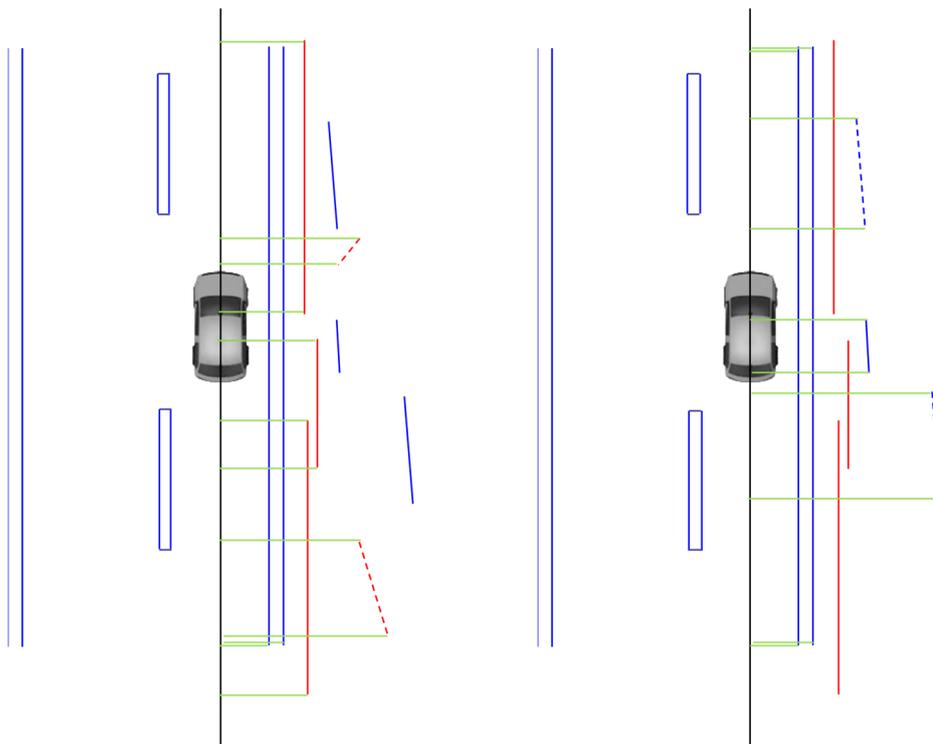


図 6.20 縁石外領域の白線の除去

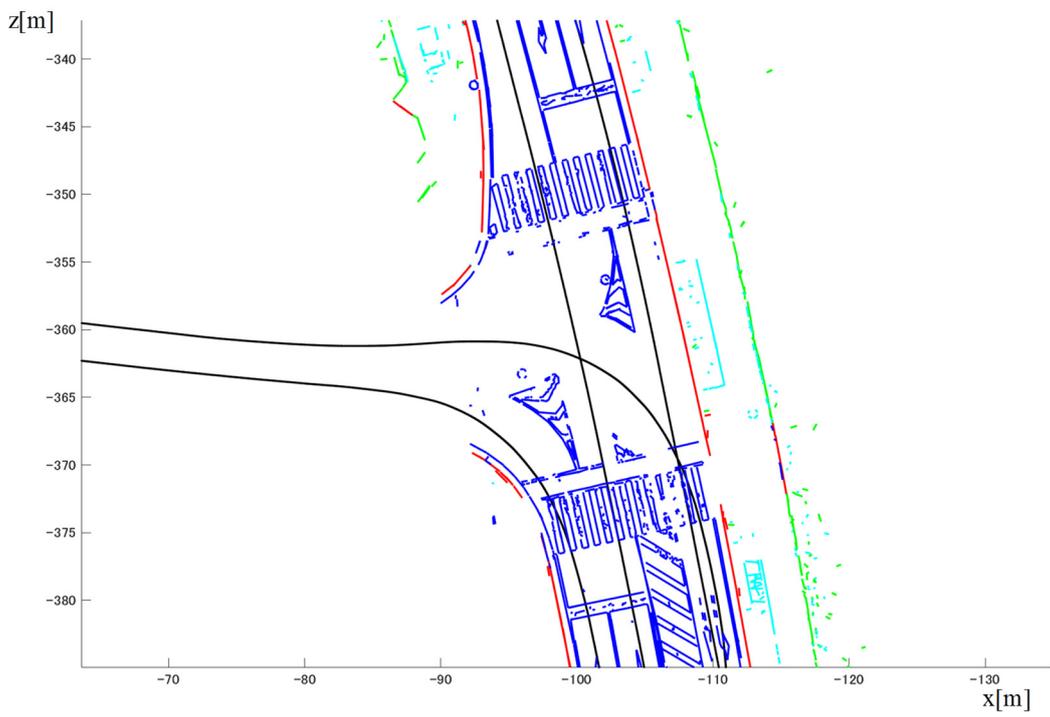
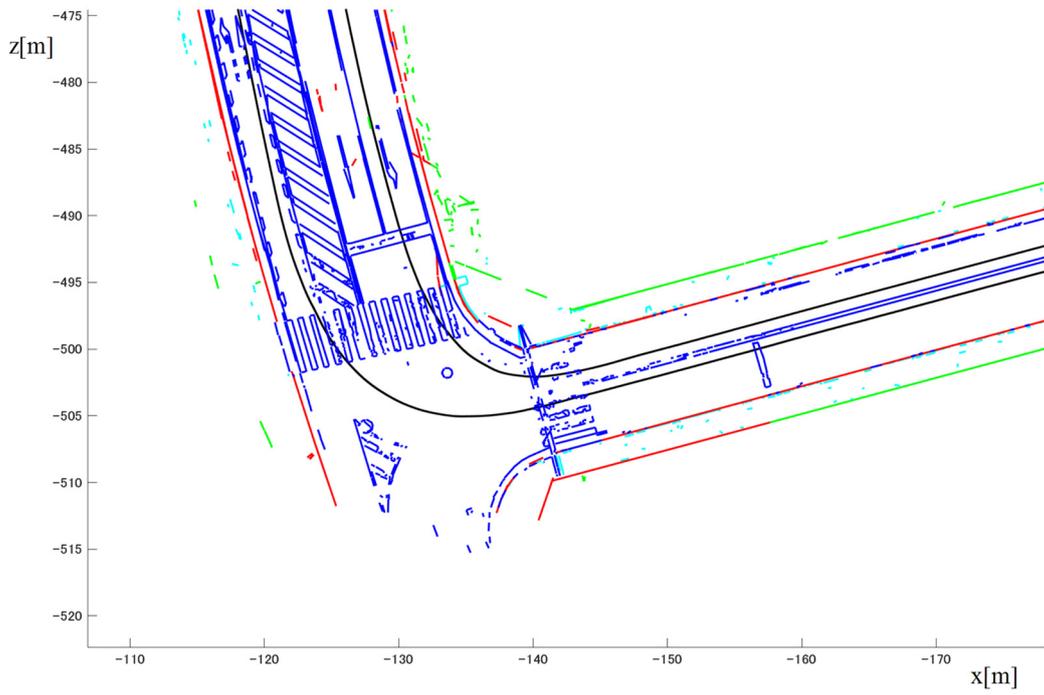


図 6.21 縁石外の線の除去の処理結果

7. 自動生成した走行レーン地図の評価

提案法によって自動生成した地図と，人手によって作られた地図を比較評価し，高い精度で一致することを検証する．これにより人手に比べて低コストに地図を作れることを示す．

7.1. 人手による地図との比較

自動生成した地図(生成地図) について，人手によって作られた地図(人手地図) と比較し評価を行った．

7.1.1. 評価経路

地図の評価経路(図 7.1)は 2.5km の主要道路であり，交差点や陸橋による高低差と約 3 度の勾配のある箇所を含んでいる．3次元点群の収集に用いた MMS には位置計測として Applanix POS/LV520 が，レーザスキャナとして RIEGL VQ250 が 2 機搭載されている．自動地図と人手地図の作成には，同じ計測による 3次元点群を用いた．人手地図には専門業者によって高精度に作図されたものを用いた．誤差が 100mm 以内であることが保証されており，白線，黄線，道路標示，縁石が 3次元線分として表現されている．

また，生成地図と人手地図の具体的な生成例を示すため，図 7.2 に示す複雑な道路ペイントがあった領域を抜き出し，以降に地図の生成事例として示す．A は比較的明瞭な交差点であり，B は陸橋の勾配がかかった直進路，C,D は勾配後の交差点の例である．

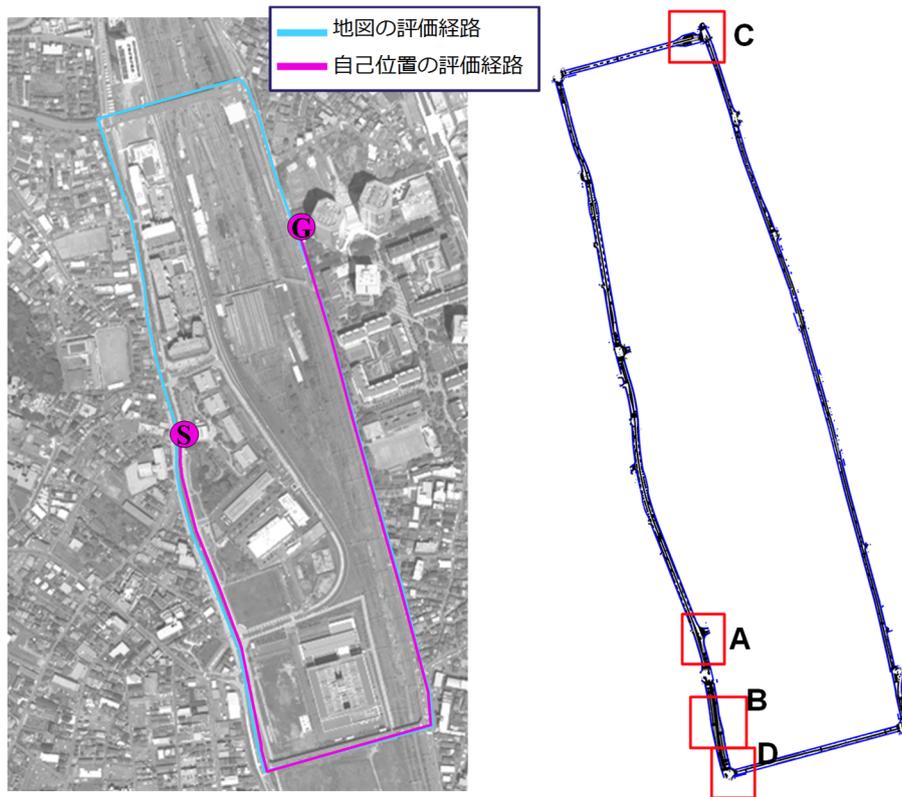


図 7.1 評価経路

A地点



B地点



C地点



D地点



図 7.2 複雑な道路ペイントがある領域の例の位置におけるカメラ画像(参考)

7.1.2. 評価基準

定量的な評価のため生成地図と人手地図が一致していることを示す基準を、生成した線分の両端点から評価地図への射影距離が 100mm 以内であるものとする。この基準を用いて線分が一致、もしくは不一致であった長さを True Positive Rate(TPR), および Precision(PRS)によって評価した。TPR は人手地図の線分が、生成地図と一致した長さの割合である。つまり TPR が高ければ人手地図の多くの線分を自動で抽出できたといえる。PRS は生成地図の線分が、評価地図と一致した割合を示している。PRS が高ければ、生成地図に誤検出した無駄な線分が少ないといえる。

7.1.3. 評価結果

地図の線種別ごと(図 7.3)の特性を調べるため、それぞれ分離して集計した。種別ごとおよび全体の評価結果を示す(表 7.1)。PRS が 91.9%と十分に高いとき 83.9%と高い網羅性で地図を自動で生成することができた。横断歩道がやや低めに出ているのは、短い線分の割合が大きいことや、かすれた箇所を人が補った影響と考えられる。また計測車両から離れた箇所について、点群の密度が低いことから自動生成地図の結果が悪くなることがあった。

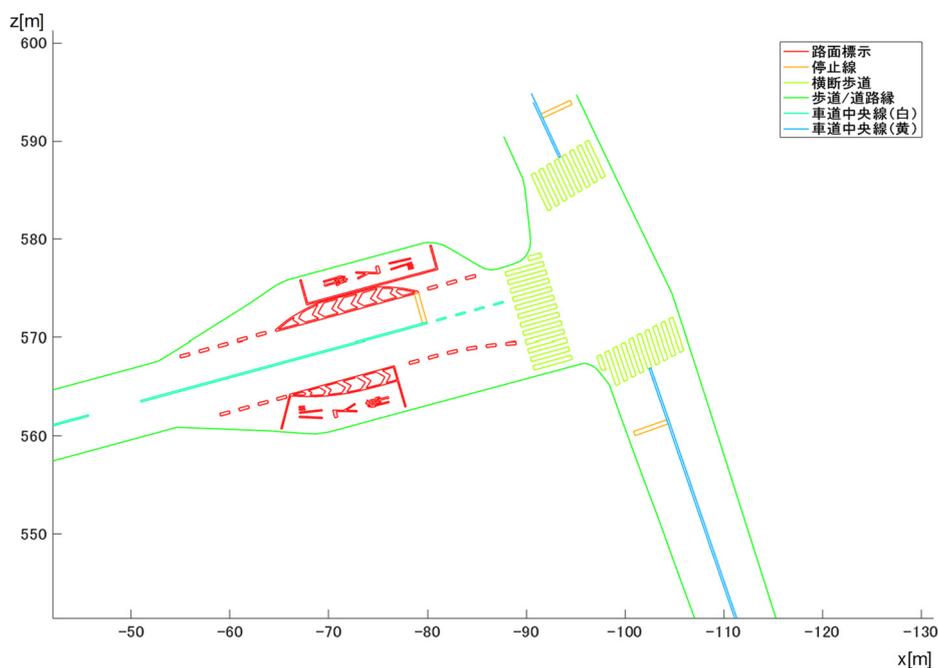


図 7.3 評価用の線種別

表 7.1 線種別ごとの地図生成の精度
(Precision = 91.9[%])

線種別	True positive rate[%]
白線	81.9
黄線	85.2
道路標示	86.0
停止線	82.7
横断歩道	72.2
縁石	90.2
全体	83.9

7.1.4.地図生成の具体例

次に生成地図と人手地図の具体的な生成例を示すため、で複雑な道路ペイントがあった領域(図 7.3)を抜き出して地図の生成結果を示す(図 7.4-7.5). (a)-(d)の結果より、道路の白線の道路標示の様子が高精度に生成されていることが確認できる. (e)-(h)の例はやや難しい箇所を表しており、まとめると次の課題があることがわかった.

- ① 横断歩道や白線などのかすれ
- ② 縁石がスロープで途切れる
- ③ 混雑した交差点などでは常に車が止まっているため路面が見えない
- ④ 縁石の外側まで復元してしまう

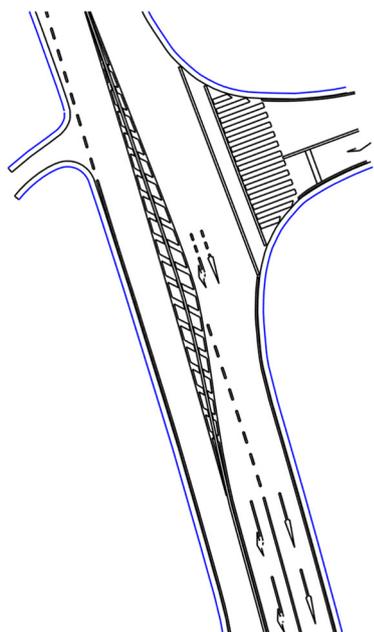
また陸橋における道路の継ぎ目やマンホールなど、人手地図を作る際に仕様に入れなかった箇所を検出することがあった. これは評価では誤検出として扱われているが、後述する自己位置推定では有効に働く結果が見られた.

7.2.考察

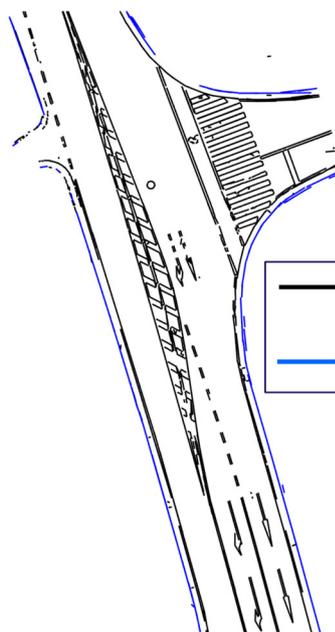
評価結果では人手による地図と比べ 91.9%の精度で人手地図の 83.9%が自動生成されている. 誤った線の修正 8.1[%] (= 100 - 91.9), および不足線の書き入れ 16.1 [%](= 100 - 83.9)を考慮すると、人手に要していた時間の 75.8%(= 100 - 8.1 - 16.1)が削減できると期待でき、十分な低コスト化の手助けになることが言える.

また自動生成では橋の道路の継ぎ目やマンホールなど、人手で地図を作る際に仕様に入れなかった箇所を検出することがある. これは自然な道路上の自然なエッジを抽出しており、自己位置推定に対しては良好に働く結果が見られた.

(a) 人手地図 - A

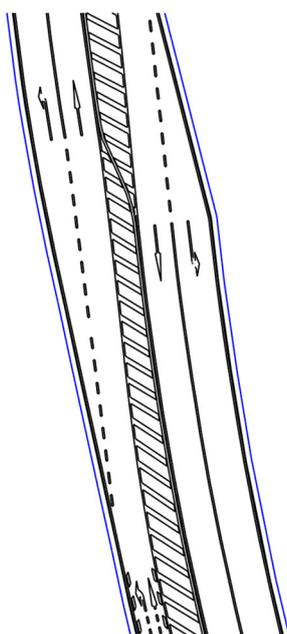


(b) 生成地図 - A



— 路面ペイント
（白線・黄線など）
— 縁石

(c) 人手地図 - B



(d) 生成地図 - B

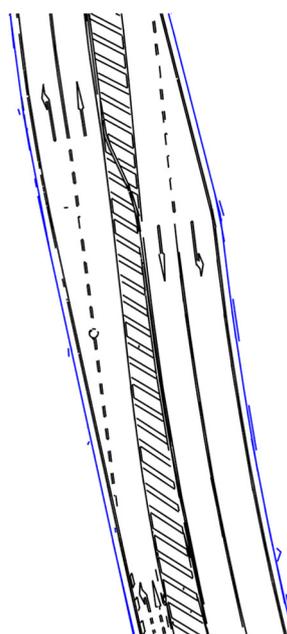
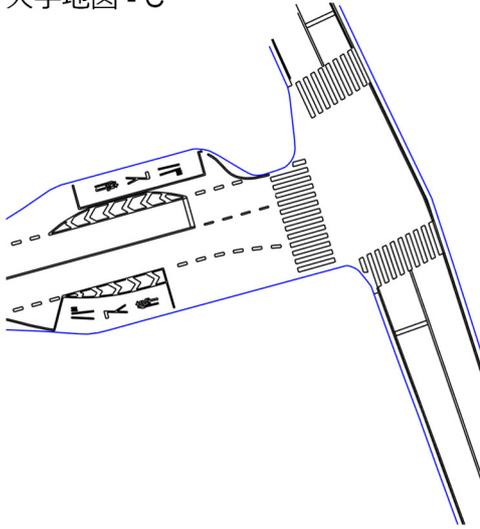
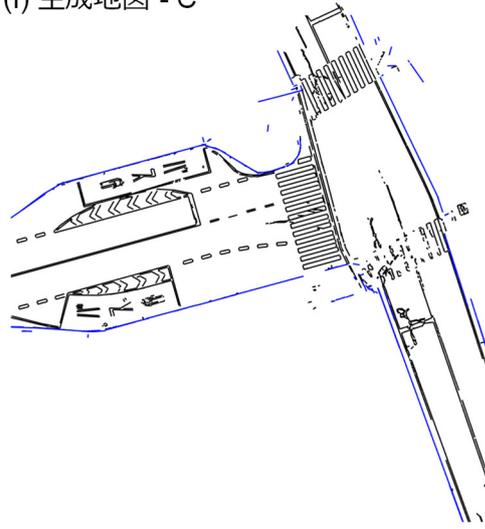


図 7.4 人手地図と生成地図の比較

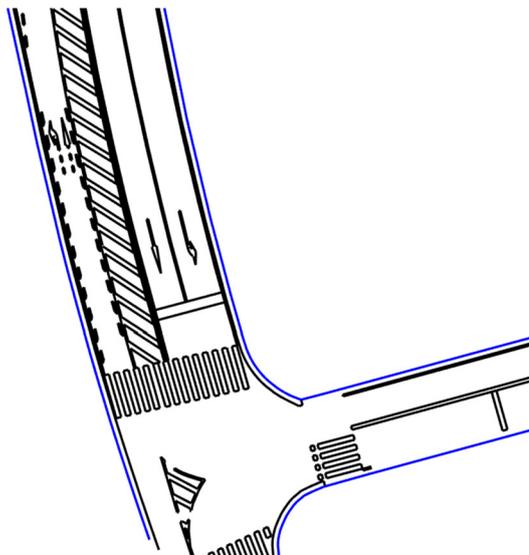
(e) 人手地図 - C



(f) 生成地図 - C



(g) 人手地図 - D



(h) 生成地図 - D

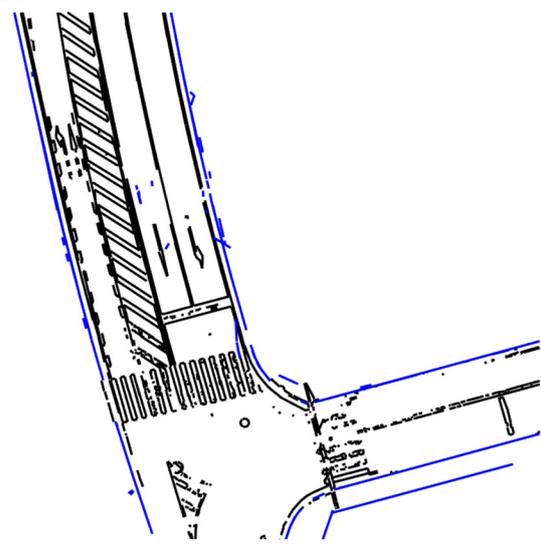


図 7.5 人手地図と生成地図の比較(続き)

8. 自己位置推定の評価

市街地と都市高速における自己位置推定の評価について述べる。市街地では道路から線分を検出すること、マルチカメラを利用することで高精度な自己位置が推定できることを示す。さらに走行レーン地図を自動生成した地図に置き換えても自己位置の精度が落ちないことを示す。都市高速の評価では現実的な設定として、曖昧な白線形状しか持たない地図を用いても、高精度な自己位置推定ができることを示す。

8.1. 市街地での自己位置推定の評価

市街地を複数のカメラを搭載した評価車両で走行し、RTK-GPSによる真値と比較することで評価を行った。推定した自己位置の精度、および演算速度について報告する。なお本節では地図には人手で図化した地図を用い、自動図化による自己位置の評価は次節にて述べる。

8.1.1. 評価車両

評価車両(図 8.1)として前方, 後方に向けて産業用 USB3.0 カメラ Lumenera Lm225 を搭載し, 走行データの収集を行った。画像は 100[ms]に一度の周期で取得しており, 2,048x1,088[pixel]の解像度で撮影された画像を, 縮小処理によって 1,024x544[pixel]に圧縮して用いた。実際の処理領域は路面が写っている領域に限定できるため, 前方カメラの処理領域は 1,024x229[pixel], 後方カメラの処理領域は 1,024x247[pixel]である。レンズには広角で低歪みの KOWA LM5JC10M($f=5[\text{mm}]$)を用いた。後輪車輪速[m/s]は車両に標準搭載されているセンサの情報を CAN 経由で取得して用いた。車輪速の周期は約 33[ms]であり, 画像取得時刻まで運動モデルによる予測を続ける。自己位置の真値として RTK-GPS(Novatel OEM615)を用いて緯度経度を取得し, 地図座標系である平面直角座標系に変換して利用した。方位については位置の時間微分から算出ができるものの真値として扱うには精度が悪く評価対象から外した。なお本研究は自己位置の修正に焦点を当てており, 初期位置には真値を用いた。また夜間や悪天候についての対応は本研究では扱わないものとし, 2014 年 3 月の晴天日において AM12 時前後の時間帯で走行データを収集した。



図 8.1 評価車両

8.1.2. 評価経路

評価経路には新川崎駅周辺において走行し、RTK-Fixed であった三通りのケースを抽出した(図 8.2) . いずれのケースも交差点を曲がるシーンを含むように選択した. ケース 1 は 1.4km の長距離である. 難しい箇所として陸橋による高低差と約 3 度の勾配のある箇所や, 片側に白線が無い箇所, 広い交差点を含んでいる. ケース 2 は交差点を左折した後に, ゆるいカーブが続く道である. ケース 3 では右折後に細街路に進入して走行した.

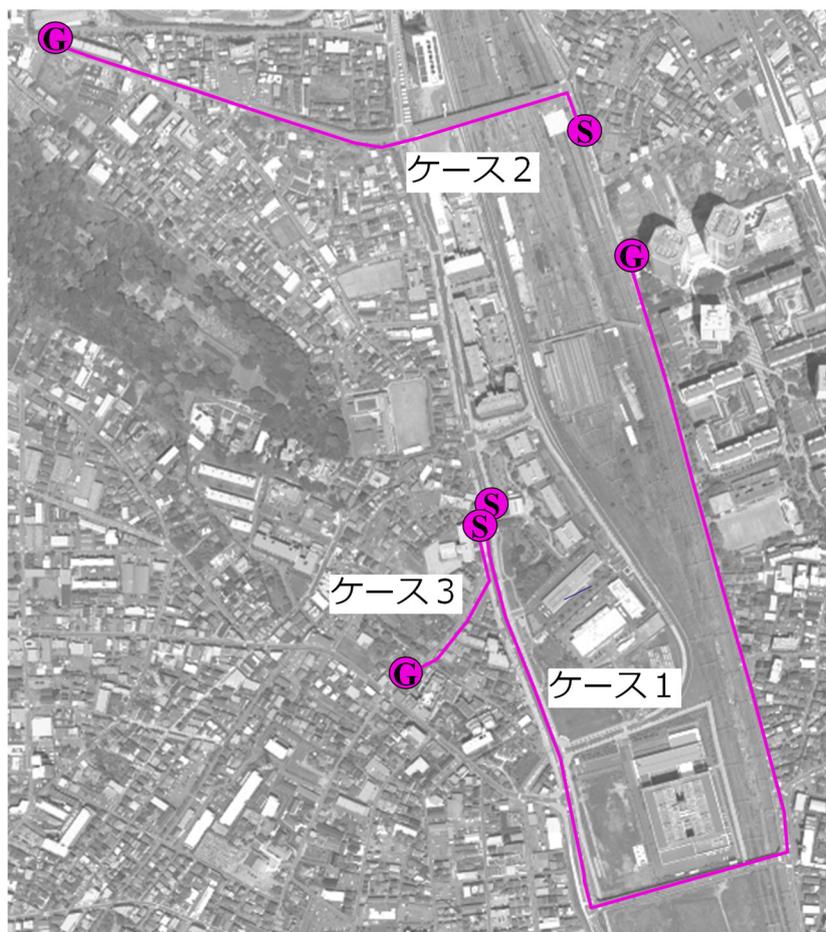


図 8.2 評価経路

8.1.3. 走行レーン地図の準備

これらの道路環境について、線分地図を準備した(図 8.3)。地図は専門業者の手によって高精度に測量されたものであり、MMS による測量によって地図を作成した後、検証用基準点を別途測量し比較することで誤差が 0.1[m]以内であることが保証されている。地図には、路面にペイントされた白線、黄線、道路標示、さらに縁石、側溝、雨水桝といった道路構造を線分として記載している。ここでは白線、縁石といったラベルは用いず前述した対応付けのための明暗の有無のみをラベルとして用いる。具体的には、白線や黄線、道路標示は明暗を持つとし、他の縁石等は明暗を持たないとした。

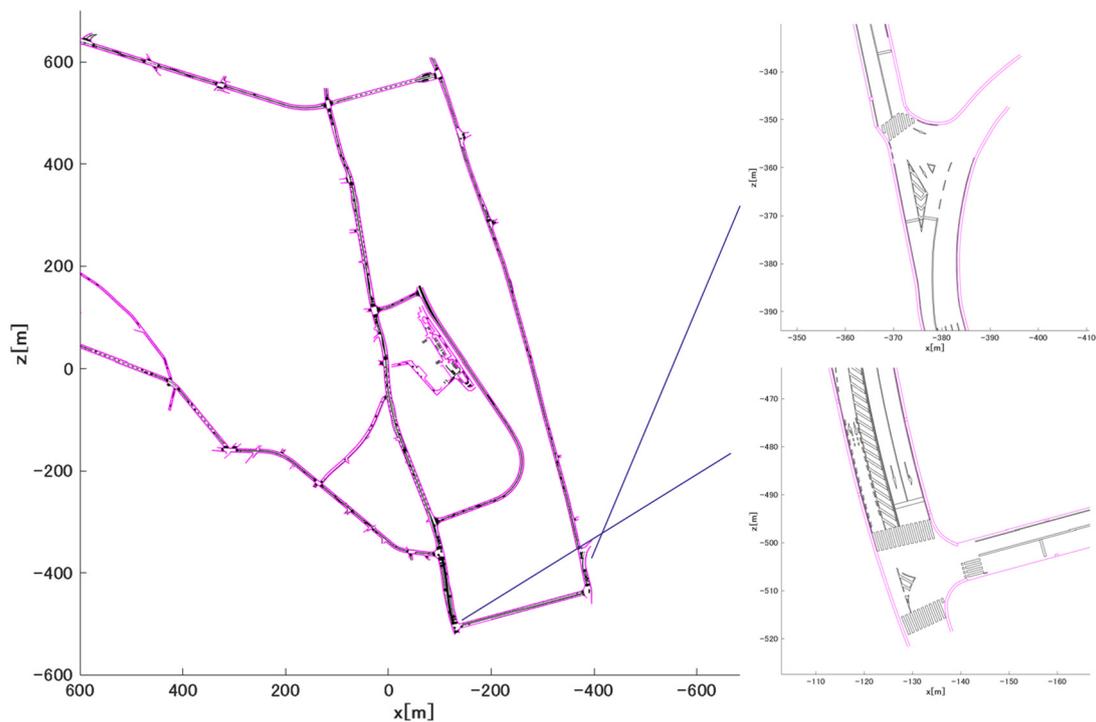


図 8.3 照合に用いる線分地図と拡大サンプル
(黒線は明暗を持つ線であり，赤線は明暗を持たない)

8.1.4.比較手法

本研究では評価のため車載画像データと走行レーン地図，および真値となる位置情報を必要とする．このため他手法との比較が難しい課題となる．特に走行レーン地図は測量が必要であるため公開されているデータが見つからない．**Open street map** を用いた研究[31]があるものの，位置決めに使えるほどの白線等の形状まで細かく記されたエリアは少なくとも新川崎駅周辺では見つからない(おそらく国内でも見つからない)．

そこで文献[20]に記載されているレーザスキャナの反射強度を用いた自己位置推定を参考に実装した方法と比較する．文献はレーザスキャナから得られる点群の反射強度と，地図を正規化相関によって比較し，パーティクルフィルタによって自己位置を推定する．ここで文献ではレーザの反射強度のラスタ地図を用いているのに対し，本研究は線分地図を扱うため，そのままでは適用できない．そこで白線や黄線など明らかに路面より反射が強い領域を 1，路面を 0 として正規化相関を計算した．レーザスキャナとして評価車両のルーフに搭載された **Velodyne HDL32e** を用いた．反射強度の S/N を考え車両周辺の 15[m] の範囲を自己位置推定に用いた．この範囲で路面に当たるのは水平方向に全方位，垂直方向に約 17 レイヤであり，十分に高い解像度を持っている．この手法を使った位置決めの例を示す(図 8.4)．点群のうち青は反射強度の低い点，赤

は反射強度の高い点を表している．つまり反射強度の高い点が白線の内側に入っていれば，精度良く自己位置が推定できている．補正前とくらべて補正後は，レーザの反射強度と地図の一致が良いことがわかる．

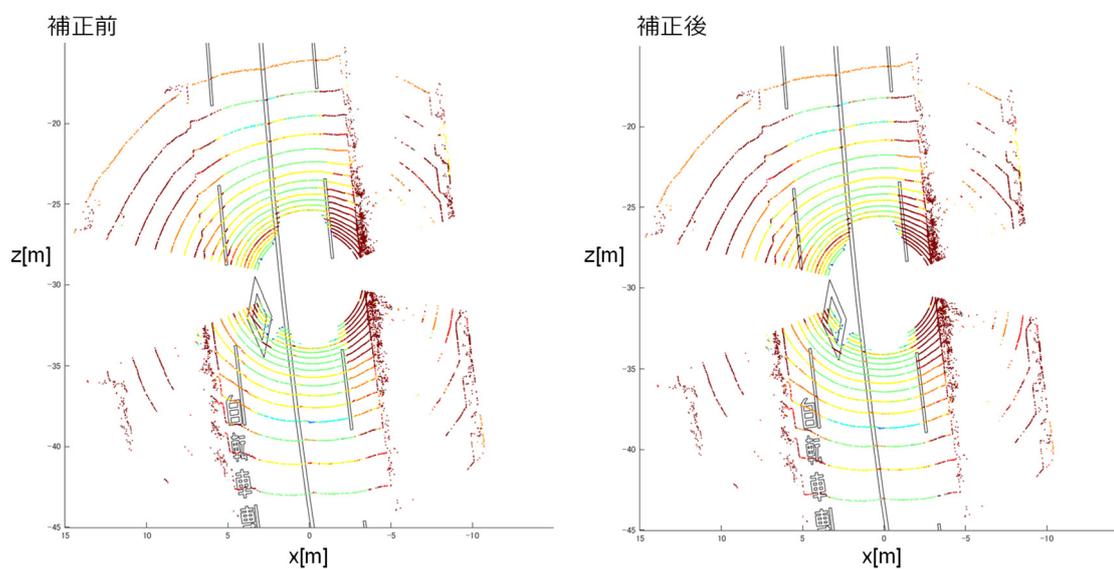


図 8.4 レーザスキャナによる自己位置推定

8.1.5. 評価基準

自己位置の推定結果と真値の精度評価について，誤差を車両の進行方向と横方向に分解して考える(図 8.5)．自動運転やレーンキープといったアプリケーションを想定すると，レーン内の位置を決めるための横方向の誤差に精度が求められる，本研究も着目するものとする．

また，自動運転やレーンキープでは演算のリアルタイム性が求められる．このため計算時間も測定した．

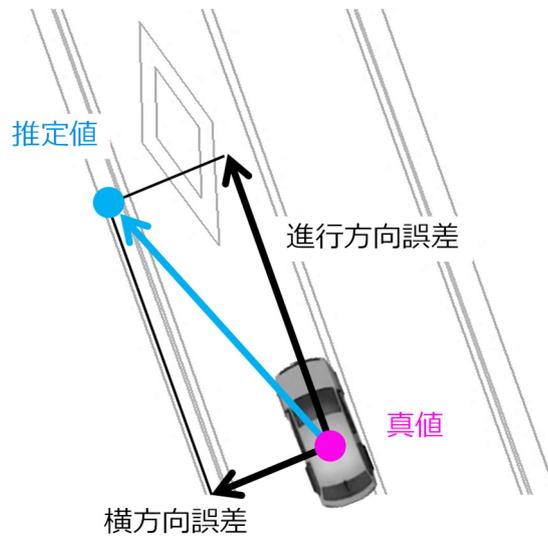


図 8.5 評価のための誤差の分解

8.1.6.自己位置の評価結果

各評価経路について前方+後方のマルチカメラと前方単眼の条件における平均誤差を示す(表 8.1). マルチカメラ条件における横方向の平均誤差はいずれも 0.1[m]を下回った. このため 1 章にて目標とした 0.2[m]を下回っており, 目標を達成している. また, 縦方向を含めても平均誤差は 1[m]以内に収まっている. 評価環境やセンサ, 地図といった条件が異なるため比較は困難であるものの, 例えば文献[19]にまとめられた複数の論文の推定精度と比較しても匹敵, ないしは高い精度を実現しているといえる.

表 8.1 推定した自己位置の平均誤差

評価経路	走行距離 [m]	比較手法	横方向誤差 [m]	進行方向誤差 [m]
ケース 1	1381	提案法(前方+後方)	0.049	0.21
		提案法(前方のみ)	0.087	0.19
		レーザに基づく比較手法	N/A	N/A
ケース 2	534	提案法(前方+後方)	0.058	0.63
		提案法(前方のみ)	0.083	0.69
		レーザに基づく比較手法	0.059	1.10
ケース 3	148	提案法(前方+後方)	0.096	0.23
		提案法(前方のみ)	0.13	0.17
		レーザに基づく比較手法	0.053	0.16

8.1.7.各経路の詳細

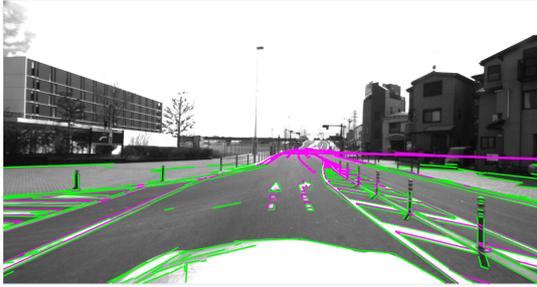
自己位置の推定について、いくつかの事例を示す(図 8.6). グリーンの線は画像から検出した線分であり、マゼンダは推定した自己位置を使って地図を画像に投影した結果である。キャリブレーションの問題はあるものの、推定した自己位置が正しければ、画像と地図は正確に重なる。(a)は複雑な白線や道路標示があるシーンである。(b)は陸橋の3度の勾配を登っているシーンである。(c)は路駐車両を避けるシーンであり、(d)は細街路を走行しているシーンである。(e)の手がかりの少ないシーンについては後述するが、いずれのシーンでも正確に自己位置が推定できていることが示されている。

またケース1における推定誤差の分布を示す(図 8.7). これは画像取得のタイミングごとの誤差をプロットしたものである。誤差はほぼ0近辺にある。進行方向に誤差が増えている箇所は、長い直線を走行した時に手がかりがなかったためである。ただ、このような地図上での手がかりが無い、つまり付近の地形に変化が無いということは、誤差が大きくなったとしてもアプリケーション上問題になりにくい場所である。一方で単眼のケースでは横方向にも誤差が増えている箇所がある。これは図 8.6e のケースであり、陸橋を登ったところで、手がかりの少ない交差点を曲がったときに発生したものである。交差点を曲がる際にはオドメトリによる誤差が大きくなりやすい。このとき、横断歩道が大きく掠れているなどの理由で、手がかりが少ないと、誤差が大きくなることがある。しかしマルチカメラ条件では、後方カメラが横断歩道を捉えることで、誤差が大きくなるのを防ぐことができ、有効性を確認した。

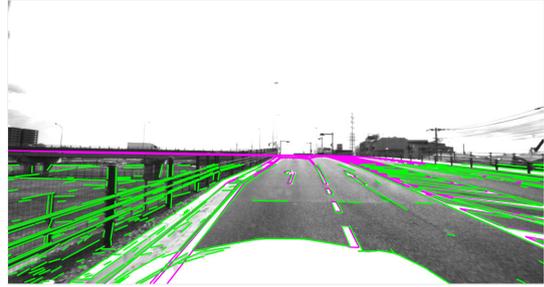
また、レーザによる手法との比較(表 8.1, 図 8.7)を考察する。ケース3では本研究とほぼ同等の精度である一方で、ケース1では、陸橋付近の直線道路で進行方向の誤差が大きくなった結果、以降の推定が難しい状態に陥った。またケース2でも同様に、ゆるいカーブを走行するシーンにおいて、進行方向の誤差が大きくなった。本研究は雨水桷や側溝などレーザの反射強度と無関係な手がかりや、点線状の白線の端点を利用することで、進行方向の誤差の補正している。一方で文献[20]と線分地図の組み合わせでは進行方向の手がかりに乏しく、誤差が大きくなったと考えられる。

さらに線分検出によって明瞭な白線以外の手がかりを検出している効果を確認するため、地図から縁石、側溝、雨水桷といった道路構造に基づく線分を削除し、白線や黄線、道路標示といった路面のペイントのみの地図に対して、本手法を適用する実験を追加で行った。実験条件は前方+後方カメラとした。その結果、ケース2では横方向の平均誤差が0.065[m]、進行方向の平均誤差が0.45[m]と、ほぼ同等の精度であったのに対し、ケース1では交差点を曲がるシーン、ケース3では細街路において自己位置が大きく外れ、以降の推定が難しい状態になった。この結果より縁石などの手がかりは、主にロバスト性の向上に寄与していることがわかった。

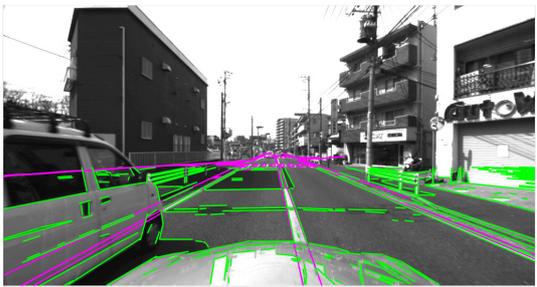
(a) 複雑な白線や道路標示



(b) 陸橋の3度の勾配



(c) 路駐車両の回避



(d) 白線が左のみの細街路



(e) 手がかりの少ない交差点

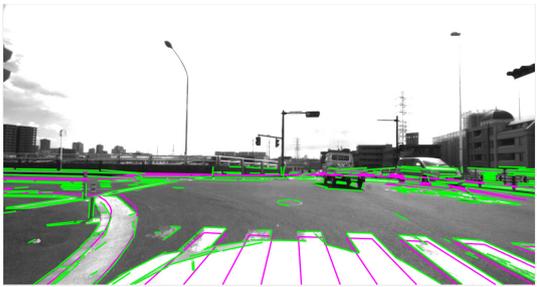


図 8.6 地図を画像上に投影した結果

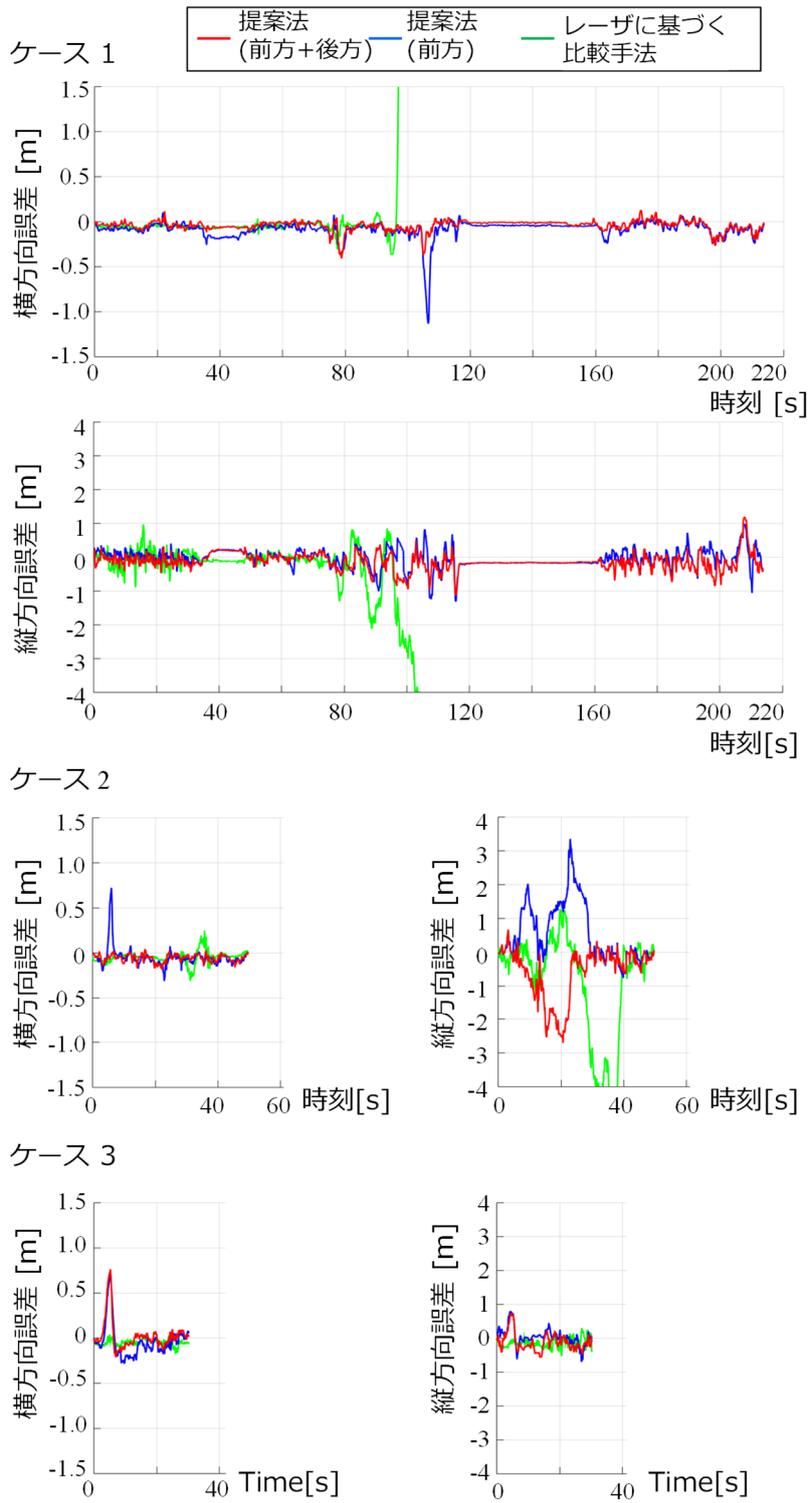


図 8.7 推定誤差の分布

8.1.8. 計算時間の評価

本研究の計算時間について、ケース 1 における平均計算時間の内訳を示す(表 8.2). 演算には Intel core i7 3.4GHz の PC を利用した. シミュレーションは MATLAB と C で実装されている. 画像の取得周期が 100[ms]であるので、十分にリアルタイム演算できると言える.

表 8.2 計算時間

	計算時間[ms]
線分抽出(前方)	13
線分抽出(後方)	15
自己位置推定	12
合計	41

8.1.9. 考察

評価結果より本研究は実用化に必要とされている横方向の誤差 0.2[m]以下を達成し、高い精度で自己位置を推定できることを示した. 全方位 LIDAR による方法と同等、またはそれ以上の精度をカメラで実現できた. またリアルタイムに計算できることから、実用的と言える.

一方で、本研究を利用する際の注意もある. 自己位置の推定が難しい環境として、まず白線や縁石などの手がかりが存在しない道路では、本質的に本研究による推定は難しい. 同様に密集した路駐車両など、手がかりが完全に隠された状況もまた、本研究で扱うことができない. そのような状況においては地図と画像線分の対応付けが得られずオドメトリによる運動モデルに従うことになる. この状態が長く続くことで自己位置の誤差が大きく蓄積すると、画像と地図の対応付け範囲を超えてしまい、GPS などによる初期化が必要となる. 高精度 GPS と同様に本研究も全ての環境で利用できるとは限らず、自動運転などの安全に関わるアプリに用いる際には、多重系を組んだ上で弱点を補い合うような運用がなされるものと想定している. またカメラにとって夜間や逆光、雨天などは難しい問題であり今後の課題と考えている. また他車両などの移動体や路駐車両から得られる地図と対応しない線分を、閾値処理とロバスト推定の枠組みで除いたが、直接的に検出すればより動作が安定する可能性があり改良の余地がある. また個別の事例を検証し誤差のピークを抑える工夫も必要になると考えられる.

8.2.自動生成地図による自己位置の評価

8.2.1.人手地図と自動生成地図の自己位置の評価

前節の市街地での自己位置推定の評価に，自動生成した地図を適用することで，自動生成した地図の精度を検証する．評価区間は地図を生成した区間(図 7.1)のうち RTK-GPS の受信状態が良好であった 1.4[km]の区間(図 8.2 ケース 1) を利用した．自己位置の推定結果と真値の比較評価について，推定誤差は車両の進行方向と横方向に分解して考える．生成地図と人手地図の条件，前方後方のマルチカメラと前方単眼の条件について，1 フレームごとの誤差の分布(図 8.8)，および平均誤差(表 8.3)を示す．自動運転やレーンキープといったアプリケーションを想定すると，レーン内の位置を決めるための横方向の誤差に精度が求められる．

8.2.2.考察

生成地図，人手地図とも横方向の平均誤差は 0.1[m]を下回っており十分に高精度であることを確認した．特に精度を保つためには地図に多くの線分が含まれていることが重要であり，生成地図が十分多くの線分を持っていることを表している．評価環境やセンサ，地図といった条件が異なるため比較は困難であるものの，例えば文献[19]にまとめられた複数の論文の推定精度と比較しても高精度な結果と言える．一方で進行方向の誤差については人手地図に比べ誤差が増えている．これは人手地図に比べて，生成地図の端点が不正確であることや，途切れることがあったためと考えられる．

表 8.3 生成地図と人手地図の自己位置の推定誤差

地図種別	カメラ設定	横方向誤差[m]	進行方向誤差[m]
生成地図	前方+後方	0.050	0.66
人手地図	前方+後方	0.048	0.26
生成地図	前方のみ	0.088	0.40
人手地図	前方のみ	0.086	0.28

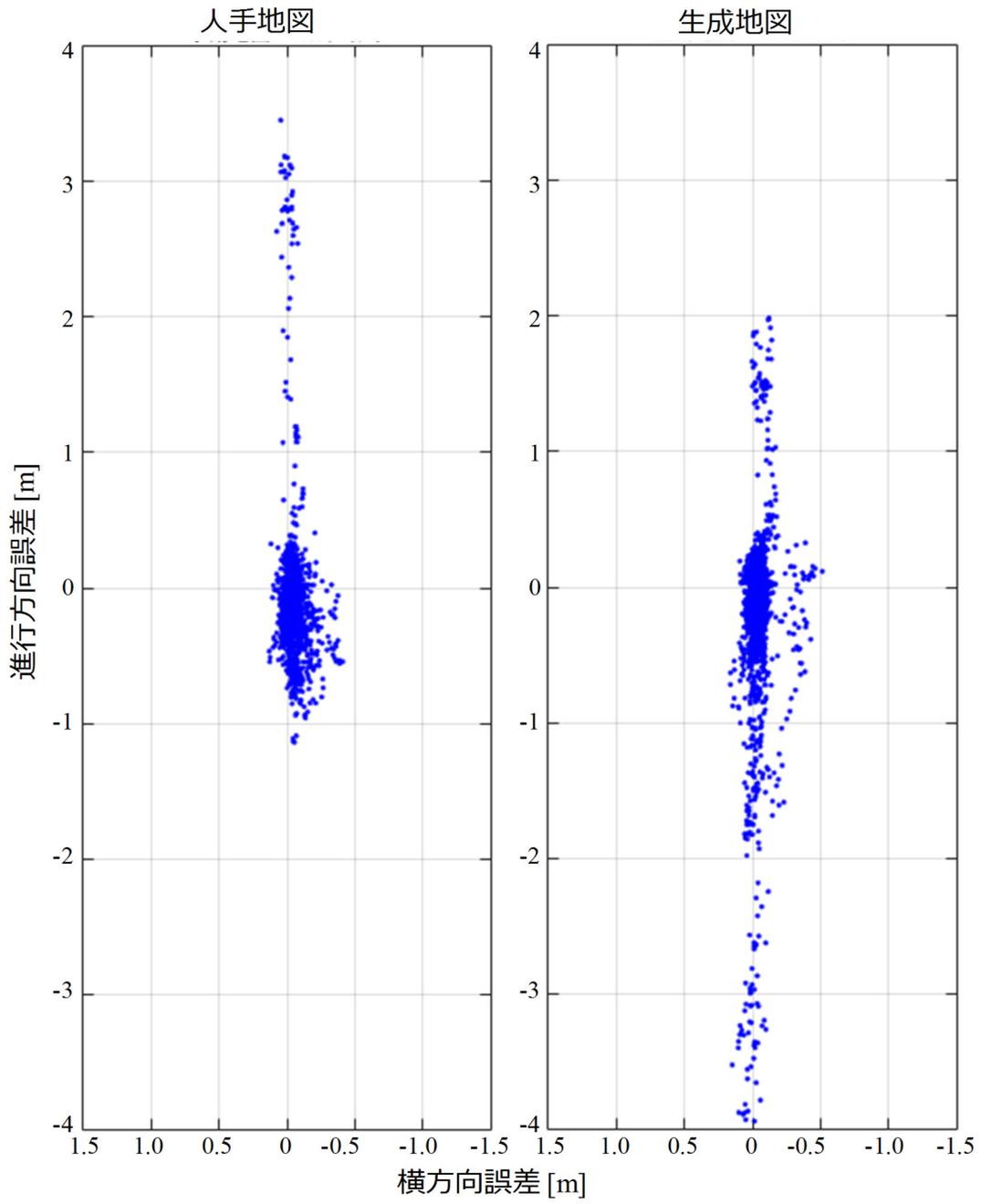


図 8.8 生成地図と人手地図の誤差の分布

8.3.都市高速での自己位置の評価

都市高速では曖昧な白線形状しか持たない地図を用いても、高精度な自己位置推定ができることを示す。

8.3.1.評価車両

評価車両として前方に向けて産業用カメラ(PointGray Flea3)を搭載し、10[fps]、2,080x1,552[pixel]で撮影された画像を、縮小処理によって1/2に圧縮して用いた。また、車両に標準搭載されている車輪速の情報を取得しオドメトリとして用いた。

8.3.2.評価経路

評価区間は首都高速 C1 環状線の内回りを一周する約 14[km]とした(図 8.9) (地図出展[52])。撮影した画像は 11,000 フレームであった。経路上には急勾配や急カーブが頻繁に含まれており非常に複雑である。なお夜間や悪天候については本研究では扱わないものとし、晴天日の昼に走行データを収集した。本研究では自己位置の逐次更新に焦点を当てており、初期位置には真値を用いた。

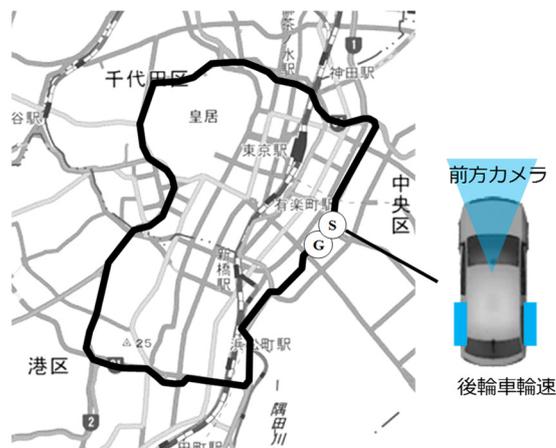


図 8.9 都市高速での自己位置の評価経路

8.3.3.評価真値の作成方法

自己位置の評価には高精度な GNSS によって取得された絶対位置(緯度経度)を真値として比較することが多い[31]。しかしながら首都高速では道路付近に高い建物が多く、ほとんどの位置において RTK による補正が有効にならない。このため例えば目標精度を 1 章で紹介した 0.2[m]としたとき、十分な計測精度を得ることができない。これは走行レーン地図の作成に使われた測量用車両も同様であり、地図の絶対位置は目標精

度に対する十分な精度を持たない。このため地図に対する相対位置の真値を作成することで評価を行う。具体的には次の3ステップで真値を作成した。

- ① 評価車両に搭載されたレーザスキャナによる点群と、走行レーン地図の作成に使われた MMS による 3 次元点群を自動で位置合わせする
- ② 地図を①による推定位置で射影した画像を参考に、人手によって位置を補正する
- ③ 運動モデルに基づいて位置を補間する。

MMS による点群は走行レーン地図の作成に使われたものであるため、地図と点群は正確に一致する。①では GPS から得られる初期位置を元に、Velodyne HDL32e から得られたレーザ点群を、MMS 計測で得られたレーザ点群に Go-ICP[39]によって位置合わせすることで、車両の位置・方位を求めた。Go-ICP は分枝限定法による最適化によって大域最適解が保証される。しかしながら他車両による遮蔽や、移動物、凹凸の不足、演算量の不足などの問題から、必ずしも所望の真値が得られるとは限らない。

このため②の人手による補正を必要とした。人手による補正では、白線の詳細形状を持った地図を画像に射影し、一致するように車両の位置・方位を GUI 上(図 8.10)を使い手動で調整した。しかしながら人手によっても白線などの手がかりが少ない場所では正確に入力できないことがあった。特に点線状の白線が無い長い道路では車の進行方向の手がかりに乏しいことがあった。

そこで人が入力した位置・方位を固定した上で、③の運動モデルを使った最適化問題を解くことで補間を行った。コスト関数には文献[44]を参考に、オドメトリの誤差、加速度/角加速度の二乗和、人手で設定した基準点からの誤差、レーン端への射影距離と近傍の基準点の射影距離の差を誤差とした。

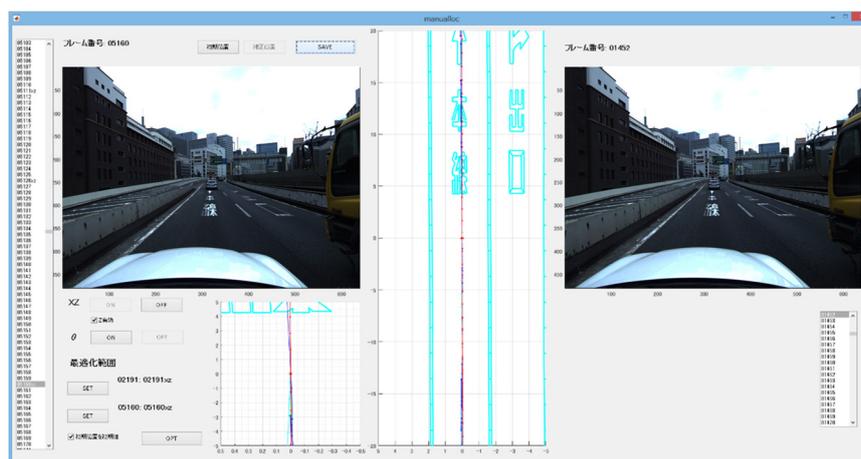


図 8.10 自己位置の真値の入力画面

8.3.4.自己位置の評価結果

自己位置の推定結果と前項で求めた真値を比較し評価した。市街地の実験と同様に誤差を車両の進行方向と横方向に分解して考える。自動運転やレーンキープでは、横方向に精度が求められる。進行方向については走行レーン地図に手がかりとなる特徴が乏しく、画像との直接の対応付けからはほとんど補正されない。一般には走行レーン地図に白点線の端点や標識等を記載し手がかりとすることで補正することも考えられるが、そのような手がかりは必ずしもあるとは限らず、ここでは厳しい条件を想定して見送った。しかしながら、本研究では車両軌跡を推定するため、首都高速のように急カーブの多い経路では、道路形状に特徴があるため、進行方向も補正される。このため進行方向の誤差が抑えられ、大きく外れることなく一周することが可能であった。

比較対象には自己位置推定[31]で良く用いられる拡張カルマンフィルタを設定した。真値と推定した自己位置の平均誤差を示す(表 8.4)。EKF に比べると、大きく誤差が下がっていることが分かる。なお EKF は急カーブで大きく推定が外れて戻れなくなってしまったため、その時点まで(1~1,200 フレーム)の評価とした。

提案法の横方向の推定誤差についてヒストグラムを示す(図 8.11)。多くのフレームで目標である 0.2[m]を下回っている。0.2[m]を超えている箇所では、急カーブの際に進行方向の誤差の影響もあって追従しきれず誤差が大きくなるケースが見られた。

表 8.4 自己位置推定の平均誤差

比較手法	横方向誤差 [m]	進行方向誤差 [m]
提案法	0.09	3.02
EKF	0.30	4.78

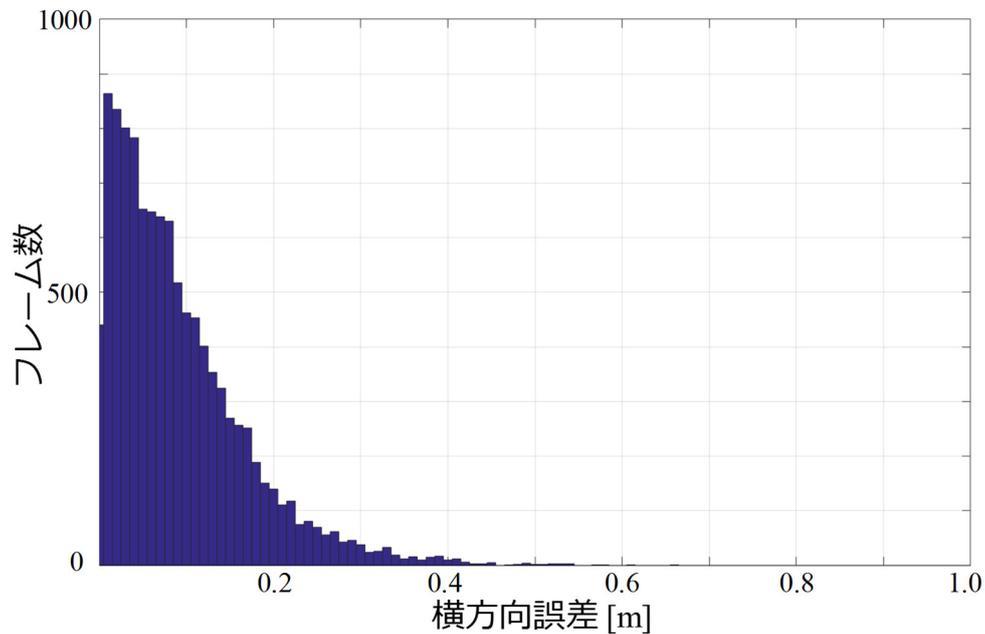


図 8.11 横方向の誤差のヒストグラム

8.3.5. 計算時間の評価

提案法の計算時間について評価を行った。計算環境には Intel core i7 3.4GHz の PC を用いた。シミュレーションソフトウェアは MATLAB で実装されており、MATLAB Coder によって C 言語に変換して測定した。その結果、1 フレームあたりの演算時間は、提案法は 39[ms]、EKF は 26[ms]であった。軌跡推定の内訳は白線検出のマッチング処理が 24[ms]、位置の最適化が 15[ms]であった。EKF よりは遅いものの 1 フレームは 100[ms] であるので、十分にリアルタイム計算が可能である。

8.3.6. 考察

複雑な首都高速 C1 環状線で評価を行い、目標に対し十分に高精度な自己位置を得た。走行距離も 14[km]と長く、安定して結果が得られることがわかる。これにより、走行レーン地図に画像と十分に対応する情報が記載されて無くとも、事前に画像の見えと対応付ける確率分布を作っておけば、対応できることを示した。本手法においては進行方向の補正は加えなかったが、一般には標識や白線の端点などの位置を元に補正する。本手法でもコスト関数を修正することで容易に拡張でき、精度向上が期待できる。

9. 結論

9.1. 総括

本研究は複雑な市街地や都市高速において自動運転に必要なレーン認識を、安価で高精度に実現することを目的として、自己位置を補正する画像処理アルゴリズムと、事前に準備する地図の両側面から改良方法を提案した。

まず都市高速では記載特徴の少ない地図に画像の見えに対する曖昧さが残る課題に対し、事前に白線形状の曖昧さを確率的にモデル化する方法を提案した。評価によりリアルタイム性と、自動運転の要求より高精度な自己位置が推定できることを示した。

次に市街地では、線分表現による地図と、マルチカメラ画像から抽出した線分特徴を照合することで、高精度な自己位置を推定する方法を提案した。線分検出を用いることで路面上の白線、黄線、道路標示、縁石など異なる特徴を一元的に抽出する。抽出した線分は車両座標系に鳥瞰変換して用いることで、カメラの構成に依存しない処理を実現した。自己位置推定には拡張カルマンフィルタを適用し、リアルタイム推定を実現した。評価により自動運転の要求より高精度な自己位置が推定できることを示した。

さらに自動生成した地図を人手によって作られた地図と比較し、91.9%の精度で人手地図の83.9%が網羅できたことを確認した。これは人手によって地図を作った際の7割のコストを削減したことに相当する。さらに自動生成した地図を、自己位置推定の問題に適用したところ、人手によって作られた地図を使った場合と同等の自動運転の要求より高精度な自己位置が得られた。これらの結果より本研究は実用的な水準に達していると考えられる。

9.2. 今後の展望

現在の自動運転技術においては地図の整備が不可欠である。しかしながら、自動運転の対応道路が自動車専用道路から市街地へと広がるにつれ、地図を手作りする従来法はコストの面で破綻する。これに対応するため本研究でも検討した段階的な地図の自動生成が進むと予想される。自動運転が道路別に自動車専用道路、幹線道路、生活道路と段階的に広がると考え、今後の展望を検討する。

まず総延長が短く複雑さも限定的な自動車専用道路では、人手による作図が中心となる。このうち都市高速のような複雑な環境においては、本研究が提案する地図の曖昧さを確率的に扱う方法が高精度・ロバストな自己位置推定に役立つと考えられる。

次に国道や県道といった幹線道路では、交差点や複雑な道路標示への対応を迫られる。このため地図に記載する特徴を増やす必要があり、一方で総延長が伸びることから人手による作図は限界を迎え自動化が進むと考えられる。本研究が提案するように、MMS による測量結果から、自動で地図を生成し、人手で修正・保証して地図を配信するのが一つの姿と考えられる(図 9.1)。また自動車専用道路と異なり、白線が掠れて残っていないことも多く、本研究が提案する画像から縁石も含めて線分特徴として抽出する方法が役立つ可能性がある。

最後に最も総延長の長い生活道路を考える。Google ストリートビューが実現されていることを考えると、MMS 計測は不可能ではない。しかしながら大きなコストがかかることから、限定的になると考えられる。MMS による計測の代替として、一般車両のカメラやレーザスキャナのデータをデータセンターで収集することで、地図を構築する方法が考えられる。著者らは一般車両のカメラ画像から地図を作る方法の研究 [13][35]も進めているが精度の問題から実現は難しい。そこで段階的に自動化を進めるのであれば、まずはメンテナンスやチェンジディテクションを一般車両のカメラ画像から行う研究が考えられる。1章にて道路工事による変化を紹介したが、縁石やポールなどの道路の3次元構造と、白線などの道路ペイントでは、道路ペイントの工事が圧倒的に多い。このため MMS で一度だけ3次元構造を計測し、道路ペイントの修正は一般車両のカメラ画像から行うようにすれば、現実的な精度が得られる可能性がある。

今後は本研究をより実用化に近づけるとともに、地図に係る自動運転の問題を解決したいと考えている。

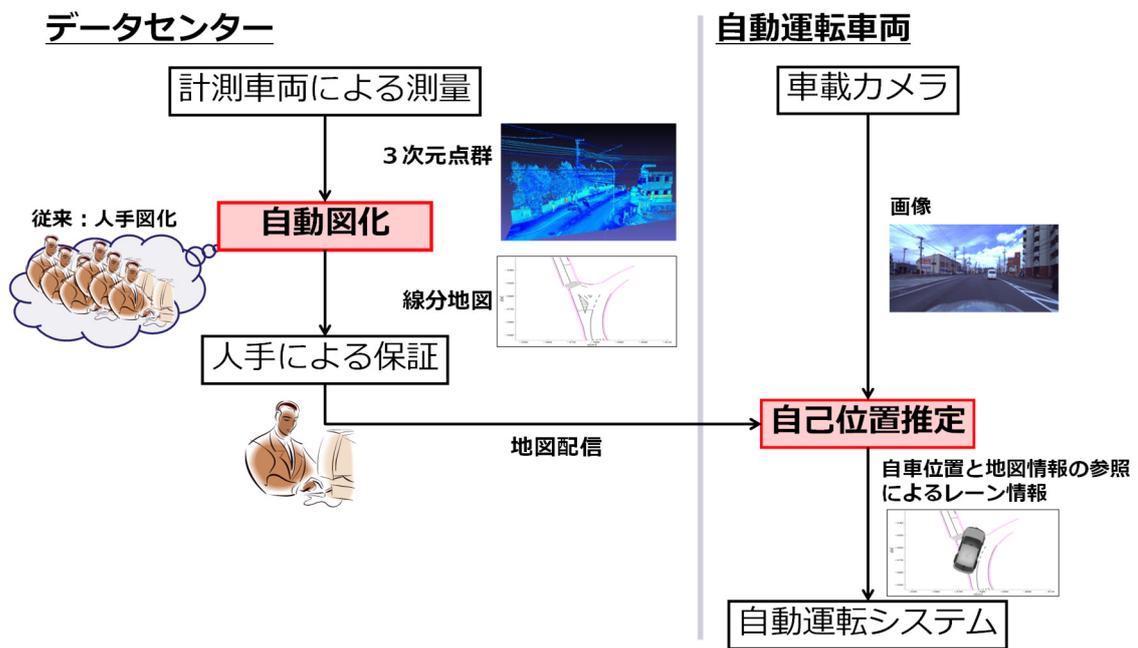


図 9.1 市街地の自動運転の構成

謝辞

本研究は、著者が慶應義塾大学大学院理工学研究科在学中に、同大学理工学部 齋藤英雄教授のもとで行ったものである。研究を遂行するにあたりご指導を頂き、貴重な助言を賜りました齋藤英雄教授に深く感謝いたします。研究を始めた頃には、研究を構成する方法が全くわかっていない不勉強な状態でした。先生のご指導のおかげで道筋を見つけ研究を進め、本論文をまとめることができました。心から感謝いたします。

そして本論文の副査をお引き受けいただいた、同大学理工学部 藤代一成教授、青木義満教授、杉本麻樹准教授に深く感謝いたします。年末年始のご多忙の中、公聴会や博士論文の内容でご対応を頂き、多くの助言を賜りました。また青木義満教授には共同研究でもお世話になり、別の視点から技術的な視野を広げることができました。ここに深く感謝いたします。

また、同大学大学院政策・メディア研究科 大前学教授には、共同研究を通じて自動運転に関する様々な知識や、実際の自動運転車両を通じた自動運転のシステム全体を勉強させていただき、自動運転に対する考え方を学ぶことができました。深く感謝いたします。

著者が勤務する株式会社デンソーアイティラボラトリーの平林裕司氏、山内康孝氏、岩崎弘利氏には、博士課程に通うことを快諾いただきご支援をいただきましたこと、深く感謝いたします。特に山内康孝氏には博士課程に通うことを相談させていただき、社内で様々なご支援、励ましをいただきました。深く感謝いたします。さらに同社の山野千晴氏、高野岳氏、鈴木幸一郎氏、安達仁吾氏、関川雄介氏には、多くの技術的な議論をさせていただきました。特に高野岳氏には研究の初期において、自動運転に関する様々な検討や実験を共にさせていただきました。本論文のテーマを見つけ、研究することができたのは高野岳氏のおかげです。心から感謝いたします。

共同研究を通じて関連技術の勉強を共にさせていただいた齋藤英雄研究室の川崎敦史氏、上原啓氏に深く感謝いたします。様々な実験結果を拝見することで、車載カメラ画像処理に関する勉強ができました。深く感謝いたします。

最後に、温かい目で見守ってくれた両親や友人諸氏に心から感謝いたします。

2018年1月10日

原 孝介

参考文献

- [1] H. Bay, T. Tuytelaars, L. Van Gool, "Surf: Speeded up robust features", European Conference on Computer Vision, pp.404-417, 2006.
- [2] J. Bouguet, "Camera Calibration Toolbox for Matlab", http://www.vision.caltech.edu/bouguetj/calib_doc/, accessed 2013-06-16.
- [3] C. Brenner, "Global localization of vehicles using local pole patterns", Annual Symposium of the German Association for Pattern Recognition, pp.61-70, 2009.
- [4] F. Chausse, J. Laneurit, R. Chapuis, "Vehicle localization on a digital map using particles filtering", IEEE Intelligent Vehicles Symposium, pp.243-248, 2005.
- [5] H. Fukui, T. Yamashita, Y. Yamauchi, H. Fujiyoshi, H. Murase, "Pedestrian detection based on deep convolutional neural network with ensemble inference network", IEEE Intelligent Vehicles Symposium, pp.223-228, 2015.
- [6] R. Gioi, J. Jakubowicz, M. Morel, G. Randall, "LSD: A fast line segment detector with a false detection control", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol.32, no.4, pp.722-732, 2008.
- [7] M. D. Grossberg, S. K. Nayar, "A general imaging model and a method for finding its parameters", International Conference on Computer Vision, vol.2, pp.108-115, 2001.
- [8] H. Guan, J. Li, Y. Yu, M. Chapman, C. Wang, "Automated Road Information Extraction From Mobile Laser Scanning Data", IEEE Transactions on Intelligent Transportation Systems, vol. 16, no. 1, pp. 194-205, 2014.
- [9] C. Harris, M. Stephens, "A combined corner and edge detector", Alvey Vision Conference, vol. 15, no. 50, pp. 10-5244, 1988.
- [10] K. Hirose, H. Saito, "Real-time Line-based SLAM for AR", International Workshop on Benchmark Test Schemes for AR/MR Geometric Registration and Tracking Method, 2012.
- [11] ITF, "Road safety annual report", OECD Publishing, 2016.
- [12] K. Ishikawa, Y. Amano, T. Hashizume, J. Takiguchi, "A Study of Precise Road Feature Localization using Mobile Mapping System", IEEE/ASME International Conference on Advanced Intelligent Mechatronics, pp.1-6, 2007.
- [13] A. Kawasaki, H. Saito, K. Hara, "Motion estimation for non-overlapping cameras by improvement of feature points matching based on urban 3D structure", International Conference on Image Processing, pp.1230-1234, 2015.
- [14] K. Kidono, T. Miyasaka, A. Watanabe, T. Naito, J. Miura, "Pedestrian Recognition Using High-definition LIDAR", IEEE Intelligent Vehicles Symposium, pp.405-410, 2011.
- [15] G. Klein, D. Murray, "Parallel tracking and mapping for small AR workspaces", ACM International Symposium on Mixed and Augmented Reality, pp.225-234, 2007.
- [16] H. Latégahn, M. Schreiber, J. Ziegler, C. Stiller, "Urban localization with camera and inertial measurement unit", IEEE Intelligent Vehicles Symposium, pp.719-724, 2013.
- [17] H. Latégahn, J. Beck, C. Stiller, "DIRD is an illumination robust descriptor", IEEE Intelligent Vehicles Symposium, pp. 756-761, 2014.

-
- [18] N. Mattern, R. Schubert, G. Wanielik, "High-accurate vehicle localization using digital maps and coherency images", IEEE Intelligent Vehicles Symposium, pp.462-469, 2010.
- [19] S.Nedeveschi, V. Popescu, R. Danescu, T. Marita, F. Oniga, "Accurate ego-vehicle global localization at intersections through alignment of visual data with digital map", IEEE Transactions on Intelligent Transportation Systems, vol. 14, no. 2, pp. 673-687, 2013.
- [20] J. Levinson , M. Montemerlo, S. Thrun., "Map-Based Precision Vehicle Localization in Urban Environments", Robotics: Science and Systems, 2007.
- [21] J. McCall , M. Trivedi, "Video-based lane estimation and tracking for driver assistance: Survey, System, and Evaluation", IEEE Transactions on Intelligent Transportation Systems, vol.7, no.1, pp. 20-37, 2006.
- [22] C. Niclass , M. Soga, H. Matsubara, M. Ogawa, M. Kagami, "A 0.18 μ m CMOS SoC for a 100m-Range 10fps 200 \times 96-Pixel Time-of-Flight Depth Sensor", ISSCC Dig. Tech Papers, pp.488-489, 2013.
- [23] M. Noda, "Vehicle ego-localization by matching in-vehicle camera images to an aerial image", Asian Conference on Computer Vision 2010 Workshops - Computer Vision, pp.163-173, 2011.
- [24] W.H.Organization, "Global status report on road safety", Geneva: WHO Press, 2015.
- [25] D. Pfeiffer, U. Franke , "Towards a Global Optimal Multi-Layer Stixel Representation of Dense 3D Data", British Machine Vision Conference, vol. 11, pp.51.1-51.12, 2011.
- [26] R. Pless, "Using many cameras as one", IEEE Conference on Computer Vision and Pattern, vol. 2, pp.587-593, 2003.
- [27] I. Puente, H. Gonzalez-Jorge, J. Martinez-Sanchez , P. Arias , "Review of mobile mapping and surveying technologies", Measurement, vol.46, no.7, pp.2127-2145, 2013.
- [28] C. Rouff , M. Hinchey, "Experience from the DARPA Urban Challenge", 2011.
- [29] I. Sato , C. Yamano, H. Yanagawa., "Crossing obstacle detection with a vehicle-mounted camera", IEEE Intelligent Vehicles Symposium, pp. 60-65, 2011.
- [30] D. Scaramuzza , R. Siegwart, "Appearance-guided monocular omnidirectional visual odometry for outdoor ground vehicles", IEEE Transactions on Robotics, , vol.24, no.5, pp.1015-1026, 2008.
- [31] M. Schreiber, C. Knoppel, U. Franke, "Laneloc: Lane marking based localization using highly accurate maps", IEEE Intelligent Vehicles Symposium, pp.449-454, 2013.
- [32] J. Shi , "Good features to track", IEEE Conference on Computer Vision and Pattern, pp. 593-600, 1994.
- [33] E. Teramoto, Y. Kojima, J. Meguro, N. Suzuki, "Active Safety Development of the PRECISE Automotive Integrated Positioning System and High-accuracy Digital Map Generation", R&D Review of Toyota CRDL, vol.43, no. 1, pp.13-23, 2012.
- [34] S. Thrun, W. Burgard , D. Fox, "Probabilistic robotics", MIT press, 2005.
- [35] K. Uehara, H. Saito, K. Hara, "Line-based SLAM Considering Directional Distribution of Line Features in an Urban Environment", International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications, pp. 255-264, 2017.
-

-
- [36] Velodyne, "HDL-64E", <http://velodynelidar.com/hdl-64e.html>, accessed 2017-11-13.
- [37] D. Wong, D. Deguchi, I. Ide, H. Murase, "Single camera vehicle localization using SURF scale and dynamic time warping", IEEE Intelligent Vehicles Symposium, pp.681-686, 2014.
- [38] B. F. Wu, T. T. Lee, H. H. Chang, J. J. Jiang, C. N. Lien, T. Y. Liao, J. W. Perng, "GPS navigation based autonomous driving system design for intelligent vehicles", IEEE International Conference on Systems, Man and Cybernetics, pp.3294-3299, 2007.
- [39] J. Yang, H. Li, D. Campbell, Y. Jia, "Go-ICP: a globally optimal solution to 3D ICP point-set registration", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol.38, No.11, pp.2241-2254, 2016.
- [40] B. Yi, S. Gottschling, J. Ferdinand, N. Simm, F. Bonarens, C. Stiller, "Real time integrated vehicle dynamics control and trajectory planning with MPC for critical maneuvers", IEEE Intelligent Vehicles Symposium Proceedings, pp.584-589, 2016.
- [41] Y. Yu, H. Zhao, F. Davoine, J. Cui, H. Zha, "Monocular visual localization using road structural features", IEEE Intelligent Vehicles Symposium Proceedings, pp.693-699, 2014.
- [42] J. Ziegler, T. Dang, U. Franke, H. Lategahn, P. Bender, M. Schreiber, T. Strauss, N. Appenrodt, C. Keller, E. Kaus, C. Stiller, R. Herrtwich, "Making Bertha Drive — An Autonomous Journey on a Historic Route", IEEE Intelligent Transportation Systems Magazine, vol. 6, no. 2, pp. 8-20, 2014.
- [43] J. Ziegler, H. Lategahn, M. Schreiber, C. G. Keller, C. Knoppel, J. Hipp, C. Stiller, "Ziegler et al. - 2014 - Video Based Localization for BERTHA", IEEE Intelligent Vehicles Symposium, pp. 1231-1238, 2014.
- [44] J. Ziegler, P. Bender, T. Dang, C. Stiller, "Trajectory planning for Bertha - A local, continuous method", IEEE Intelligent Vehicles Symposium, pp.450-457, 2014.
- [45] J. Ziegler, H. Lategahn, M. Schreiber, C. G. Keller, C. Knoppel, J. Hipp, M. Haueis, C. Stiller, "Video based localization for Bertha", IEEE Intelligent Vehicles Symposium, pp.1231-1238, 2014.
- [46] アイサンテクノロジー, "モービルマッピングシステム MMS-Tracer", <http://www.whatmms.com/>, accessed 2015-06-04.
- [47] 蓑谷千鳳彦, "頑健回帰推定", 朝倉書店, 2016.
- [48] 伊東維年, "カーエレクトロニクス化の進展とその課題", 産業経営研究, vol. 29, pp.65-88, 2010.
- [49] 葛巻清吾, "SIP「自動走行システム」", SIP シンポジウム 2016, 2016.
- [50] 国土交通省, "道に関する各種データ集", http://www.mlit.go.jp/road/soudan/soudan_10b_01.html, accessed 2017-11-13.
- [51] 国土交通省, "道路規準の体系", <http://www.mlit.go.jp/road/sign/kijyun/taikei01.html>, accessed 2015-6-16.
- [52] 国土地理院, "地理院地図", <http://maps.gsi.go.jp/>, accessed 2017-03-24.
- [53] 国土地理院, "日本の測地系", <http://www.gsi.go.jp/sokuchikijun/datum-main.html#p7>, accessed 2017-11-14.
- [54] 佐田達典, "GPS 測量技術", オーム社, 2003.
- [55] 自動車技術会春季大会フォーラムテキスト, "近未来の ITS 技術—ITS 世界会議東
-

-
- 京でのショーケースデモより", 2014.
- [56] 菅沼直樹, 魚住剛弘, "GNSS/INS と白線検出の融合による自動運転自動車の自己位置推定", 自動車技術会論文集, vol. 42, no. 5, pp.1151-1156, 2011.
- [57] 菅沼直樹, 林悠太郎, 永田大記, 高橋謙太, "高齢過疎地域における自動運転自動車の市街地公道実証実験概要", 自動車技術会 春季学術講演会 講演予稿集, pp.390-394, 2015.
- [58] 鈴木政康, "アラウンドビューモニタの開発", 日本機械学会誌, vol.111, no. 1073, p.52, 2008.
- [59] ダイナミックマップ構築検討コンソーシアム, "自動走行システムの実現に向けた諸課題とその解決の方向性に関する調査・検討におけるダイナミックマップ構築に向けた試作・評価に係る調査検討", SIP(戦略的イノベーション創造プログラム)・自動走行システム, 2016.
- [60] 多摩川精機株式会社, "ジャイロセンサ技術", 東京電機大学出版局, 2011.
- [61] トヨタ自動車, "レーンキーピングアシスト", http://www.toyota.co.jp/jpn/tech/safety/technology/technology_file/active/lka.html, accessed 2017-11-13.
- [62] 内閣府政策統括官(科学技術・イノベーション担当), "SIP(戦略的イノベーション創造プログラム)自動走行システム研究開発計画", 2015.
- [63] 永井正夫, ポンサトーン・ラクシンチャラーンサク, "カー・ロボティクス", ZMP パブリッシング, 2010.
- [64] 中山幸二, "車載センシング技術の開発と ADAS, 自動運転システムへの応用(12章)", 技術情報協会, 2017.
- [65] 西鍛冶聡, 福島正信, 山本康典, "マツダ・プリクラッシュ・セーフティ・システムの開発", マツダ技報, vol. 24, pp.150-153.
- [66] 日経スマートコンソーシアム記事, "夢の『オートパイロットシステム』(自動車の自動運転)実現への道のり", 2014.
- [67] 日本経済新聞, "テスラのオートパイロットは「自動運転」にあらず", 2016.
- [68] 増田広司, 水谷麻美, 木村正博, "ITS 応用 自動運転の第一歩", Toshiba review, vol. 55, no. 11, pp.23-26, 2000.
- [69] 松ヶ谷和沖, "自動運転を支えるセンシング技術", DENSO TECHNICAL REVIEW, pp.13-21, 2016.
- [70] 三菱総合研究所, "自律型モビリティに係る最近の動向", 情報通信審議会 情報通信技術分科会 技術戦略委員会 先端技術 WG, 2016.
- [71] 目黒淳一, 小島祥子, 鈴木徳祥, 寺本英二, "gps ドップラーと慣性センサの統合による車両軌跡推定手法の提案", 情報処理学会論文誌, vol.53, no.1, pp.212-222, 2012.
- [72] 米陀佳祐, 楊辰兮, 三田誠一, "公道走行環境の認識に向けた高精度地図データベースの検討", 計測自動制御システム・情報部門学術講演会, pp.325-326, 2014.
- [73] リーグルジャパン, "モバイルレーザーキャナー", <http://www.riegl-japan.co.jp/product/mobile/>, accessed 2017-11-13.
-