

情報工学コース卒業研究報告

小型無人移動体を用いた
3次元地図の自動生成に関する研究

尾崎 宏樹

名古屋大学 工学部 電気電子・情報工学科

2012年2月

概要

近年、計算機の処理能力向上とコンピュータグラフィックス技術の進歩に伴い、携帯端末などの小型計算機でも3次元モデルの描画が可能になったことで、実世界の3次元モデルを利用したサービスが登場し、3次元地図についての研究・開発が活発に行われている。しかし、3次元地図の作成にかかるコストは非常に高く、3次元地図を利用したシステムを運用することは困難である。3次元地図を広く普及させるために、3次元地図生成コストの低減は重要な課題である。そこで、本研究では、RGB-Dカメラを搭載した小型無人移動体を利用して、屋内における3次元地図を自動的に生成する仕組みを実現した。

本研究では、データの収集、センサーデータの重ね合わせ、点群データのポリゴン化の大きく3つの手順から3次元地図の生成を行う。それぞれの手順について詳しく説明する。

データの収集 我々の研究室では、小型無人動移体SUV (Small Unmanned Vehicle) と呼ばれる、自律走行が可能な小型ロボットの研究・開発を行っている。SUVは、環境内を走行することで2次元の環境地図を生成する機能を持ち、この環境地図を利用することで位置推定や自律走行が可能となる。環境地図を基にデータ収集経路を生成し、経路に沿ってSUVを走行させることで、3次元地図の生成に必要なデータの収集を行った。収集したデータは、SUVに搭載されたRGB-Dカメラによるセンサーデータとデータ記録時のSUVの位置情報である。

センサーデータの重ね合わせ RGB-Dカメラから得られたセンサーデータの重ね合わせを行う方法として、フレーム間の3次元の点群と画像特徴点の重ね合わせを行うRGBD-ICP (Iterative Closest Point) という手法が提案されている。本研究では、RGBD-ICPアルゴリズムを拡張し、フレーム間の3次元の点群と画像特徴点だけでなく、センサーデータと環境地図の対応関係も利用して、重ね合わせを行う手法を実現した。環境地図と対応付けられた3次元地図を生成することで、既存の位置推定の仕組みをそのまま利用して、3次元地図をナビゲーションシステムなどに応用することができる。

点群データのポリゴン化 センサーデータの重ね合わせによって生成した3次元地図は、膨大な数の点の集合であり、データ量が非常に大きく扱いづらい。そこで建物の内装を構成する主要な要素である壁や床といった平面に着目し、点群データから平面を抽出しポリゴン化することで、3次元地図のデータ量を大幅に削減する仕組みを実現した。

目次

第1章	はじめに	1
第2章	小型無人移動体 (SUV)	5
2.1	SUV のコンセプト	5
2.2	システム構成	6
2.2.1	二輪走行機構と本体内蔵のセンサー	7
2.2.2	レーザーレンジセンサー	8
2.2.3	RGB-D カメラ	8
2.3	環境地図生成	10
2.4	自己位置推定	11
2.5	SUV の自律走行	14
2.6	移動障害物の検出	16
第3章	3次元地図の自動生成	19
3.1	3次元地図生成	19
3.1.1	床の検出と位置推定	20
3.1.2	センサーデータの重ね合わせ	23
3.1.3	環境地図との対応付け	27
3.2	3次元地図のポリゴン化	29
3.2.1	面の検出とテクスチャの作成	29
3.2.2	面の重ね合わせ	31
3.3	経路生成	33
3.3.1	ノードの生成	34
3.3.2	ノード間の接続	37
3.3.3	経路生成のアルゴリズム	38
3.3.4	経路生成範囲の指定	40
3.3.5	生成した経路の評価	41
第4章	展望	45
4.1	様々な情報の提示	46
4.2	ナビゲーションへの利用	47
4.3	AR への応用	49

4.4	複数 SUV の連携	50
第 5 章	関連研究	53
5.1	RGB-D Mapping: Using Depth Cameras for Dense 3D Modeling of Indoor Environments	53
5.2	レーザレンジファインダ搭載移動ロボットによる動的環境の 3 次元地図生成	54
5.3	施設におけるバリアフリー情報の 3 次元地図としての視覚化	54
5.4	格子点配置を用いた自律移動ロボットによる環境掃引経路計画	55
第 6 章	まとめと今後の課題	57
6.1	まとめ	57
6.2	今後の課題	58
6.2.1	地図精度の向上	58
6.2.2	3 次元オブジェクトの配置	58
6.2.3	データ収集方法の改善	58
6.2.4	3 次元地図を利用した応用システムの開発	59
	謝辞	60
	参考文献	63

第1章 はじめに

近年、計算機の処理能力向上とコンピュータグラフィックス技術の進歩に伴い、携帯端末などの小型計算機でも3次元モデルの描画が可能になったことで、カーナビや景観シミュレーション、Google Earth¹など、実世界の3次元モデルを利用したサービスが登場し、3次元地図についての研究・開発が活発に行われている[23, 26]。

人は、普段から3次元空間の中で生活しているため、従来の2次元地図を用いて周囲の環境を認識する場合、普段慣れていない2次元の地図上の位置と3次元の実環境を対応付けて判断しなければならない。しかし、一度に広い範囲を俯瞰することができるため、2次元地図は地図全体を見渡すのに適している。一方、3次元地図は、実環境と同様に3次元空間であり、自由に視点が移動することができるため、実世界での視点を再現することもでき、より直感的に周囲の環境を認識できる。このことから、3次元地図は周囲の環境を認識することに適していると言える。そのため、大きな駅や、ショッピングモールなどの複雑で大きな建物では、2次元地図よりも3次元地図による情報提示が適していると思われる。

しかしながら、3次元地図の作成にかかるコストは非常に高く、例えば屋内施設の場合、内装を変更すると3次元地図も作り直さなければならないため、3次元地図を利用したシステムの運用は困難である。そこで、本研究では、屋内における3次元地図を自動的に生成することを目指す。

3次元地図の生成には、ロボットの移動とレーザーレンジセンサーのチルトによる3次元計測を繰り返し、3次元のスキャンマッチングによる重ね合わせによって実現するもの[20]、単眼カメラの画像から特徴点の3次元情報を取得するもの[13]、ステレオカメラを用いるもの[1]、複数のレーザーレンジセンサーとカメラを合わせて利用する手法[10]などが提案されている。

しかし、レーザーレンジセンサーを用いた手法では、平面をセンシングするため重ね合わせが難しく、Simultaneous Localization and Mapping (SLAM) におけるループの検出が難しいという問題点があり、カメラを用いた手法では、多くの計算量が必要になるほか、暗い場所や画像特徴点の少ない環境では3次元地図の生成が難しいという問題点がある。

近年では、RGB-Dカメラと呼ばれるRGB画像と深度 (Depth) 画像を同時に取

¹<http://earth.google.com/>

得可能なセンサーである Kinect²が Microsoft から発売され、物体の3次元形状を復元できるセンサーが安価に入手できるようになり、ジェスチャ[5]やモーショントラッキング[17]、3D スキャン[4]といった研究が活発に行われている。それらの研究に3次元地図の生成も含まれ、3次元の点群と画像特徴点を利用して精密な3次元地図を生成する手法である RGBD-ICP アルゴリズムが提案されている[12]。この手法では、3次元の点群と画像特徴点を用いて重ね合わせを行うことで正確に重ね合わせを行い、画像特徴点からループの検出を行うことで従来手法の問題を解決している。しかし、RGBD-ICP アルゴリズムは、RGB-D カメラのデータのみで重ね合わせを行うため、生成した3次元地図を利用する場合、実世界の位置と3次元地図座標のキャリブレーションが必要になる。環境の2次元地図が既に存在する場合には、2次元地図と対応付けられた3次元地図を生成することで、地図のキャリブレーションが必要なくなる。

そこで、本研究では、位置推定可能な小型無人移動体を用いてデータの収集を行い、2次元環境地図と対応付けられた3次元地図の生成を行う手法を提案する。

我々の研究室では、我々の生活する実世界と情報の世界をより密接に関係づけることを目的として小型無人移動体 (SUV: Small Unmanned Vehicle) を開発している。SUV は、無人で位置推定可能な小型ロボットであり、事前に環境内を走行することで2次元環境地図を生成する機能を持ち、その地図を利用して位置推定を行うことができる。本研究では、SUV に RGB-D カメラを搭載してデータの収集を行うことで、RGB-D カメラのデータとデータ収集時の位置情報を記録する。RGBD-ICP アルゴリズムを、3次元の点群と画像特徴点だけでなく、環境地図との対応も利用して重ね合わせを行うように拡張し、SUV の位置情報をセンサーデータの初期位置とすることで、環境地図と対応の取れた3次元地図を生成する。環境地図と対応の取れた3次元地図を生成することで、既存の位置推定の仕組みをそのまま利用して、3次元地図をナビゲーションシステムなどに応用することができる。

ただし、詳細な3次元地図は、空間情報を正確に表現することができるが、データ量が大きくなり扱いづらくなるばかりでなく、ユーザーに必要以上の情報を与え、本当に必要な情報が分からなくなり、混乱を招くことになると考えられる。そのため、3次元地図を利用したシステムでは、必要に応じてアイコンやテキスト、画像や音声、動画などマルチメディアコンテンツを3次元地図上にアノテーションすることでユーザーに分かりやすく情報を提示する必要がある。例えば、ユーザーの障害状態に応じて必要なバリア情報をアイコンやテキストを用いて3次元地図上に表示するもの[23]や、3次元地図上に Flash などのコンテンツへのリンクをアノテーションとして表示するもの[26]がある。本研究で生成した3次元地図は、点の集合であり、データ量が非常に大きく扱いづらい。そこで、応用システムへ利用するための準備として、点群の3次元地図から平面を検出し、ポリゴン

²Kinect は米国 Microsoft Corporation の米国およびその他の国における登録商標です。

化する仕組みを実現する。

3次元地図を作り直す必要がある場合、収集したデータが不十分だった場合、データの収集を手動で行うのは非常に手間のかかる作業である。そこで、本研究では、SUVに自動でデータ収集を行わせる仕組みを実現し、さらなるコストの低減を目指す。SUVには自律走行を行う機能があり、設計した経路に沿って、環境内を走り回ることができる。そこで、SUVがデータ収集時に走行する必要がある経路を生成し、その経路に沿ってSUVがデータ収集をしながら走行することで、データを自動的に収集する仕組みを実現する。

以下に本論文の構成を示す。第2章では、小型無人移動体SUVについて、そのコンセプトと機能について詳しく述べる。第3章では、環境地図と対応の取れた3次元地図生成について、点群データから平面を抽出し3次元地図をポリゴン化する方法について、また、データ収集のための走行経路を生成する仕組みについて述べる。第4章で、本研究で生成した3次元地図の応用例を含む展望について述べる。第5章で関連研究について述べ、最後の第6章で、まとめと今後の課題について述べる。

第2章 小型無人移動体 (SUV)

筆者の所属する研究室では、我々の生活する実世界と情報の世界をより密接に関係づけるため、自律走行可能な小型無人移動体 SUV (Small Unmanned Vehicle) を開発している。

SUV は事前に環境内を走行することにより、SUV が環境を認識するために必要な環境地図を生成する機能を持ち、その環境地図を利用して自身の姿勢 (位置・向き) の推定を行う。リアルタイムで自身の姿勢を認識することができるため、設計した経路に沿った柔軟な自律走行が可能である。また、地図と現在のセンサーの値を比較することで、人間などの地図には存在しない障害物を認識することが可能である。

本章では、SUV のコンセプト、SUV の構成、SUV が持つ機能である環境地図生成、自己位置推定、自律走行、移動障害物の検出について説明する。

2.1 SUV のコンセプト

SUV は、自身を取り巻く環境を認識して人間に情報を提示することができ、また、環境内を探索することで様々なデータを蓄積し、そのデータを再利用することで高度なサービスを提供することを目的とした、無人で自律走行可能な小型のロボットである。

例えば、人は一人で歩くよりも、誰かの後ろについていく方がより安全であり、安心することができる。そこで、SUV に前方を走行させることにより、後ろを歩く人の安全性と安心感を向上させることができる。

人が移動する上で最も注意すべきなのは死角から突然現れる移動障害物である。そこで、人がより安全に移動するためには死角から接近している移動障害物を事前に察知することが必要である。しかし、人が実際に死角の情報を得ることは難しい。そこで、いかに環境側に周囲の情報を収集し伝達する仕組みを実現するか、ということが重要になってくる。

しかし、複雑で数多く存在する歩道の全域に渡ってそのようなインフラを整備することはコストの面で実現が困難である。そこで、上記の問題をインフラを整備する以外の方法で解決したい。まず、歩行者の安全に大きく関わる環境の情報は現在地から離れた場所ではなく、歩行者の進行方向前方に集中していると考え

られる。そこで、SUV は歩行者の前方を走行することで、道案内をすると同時に、歩行者から見て死角となる領域のセンシングを行い移動障害物を事前に検知することができる。そうして、人は SUV について歩くことで道に迷うこともなく安全に安心して目的地まで移動することができる。

また、SUV は環境内を任意に動き回ることにより様々なデータを記録、蓄積し、それらのデータを再利用して環境の情報をより詳細に解析することで、高度なナビゲーション等のサービスを提供することができる。本研究の目標である 3 次元地図の自動生成はここに位置付けられる。

2.2 システム構成

SUV は、ベースとなる対向二輪型の移動機構、環境地図生成と自己推定に利用するレーザーレンジセンサー、3次元地図生成に利用する RGB-D (カラーおよび深度) カメラ、そして、それらを制御するためのノート PC から構成されている (図 2.1 を参照)。



図 2.1: SUV の構成

次に二輪走行機構、レーザーレンジセンサー、RGB-D カメラについて説明する。制御については次節以降で詳しく説明する。

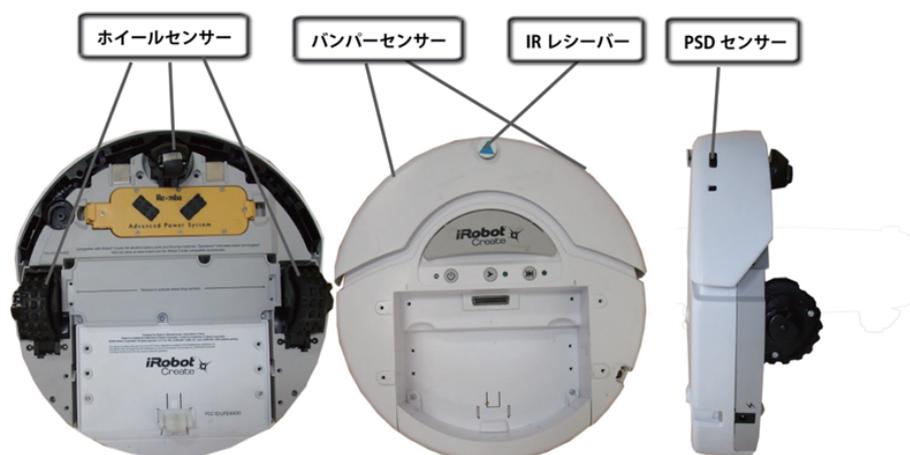


図 2.2: iRobot Create のセンサー

2.2.1 二輪走行機構と本体内蔵のセンサー

SUV はベースとなる移動体として iRobot 社の iRobot Create¹を使用している。iRobot Create は、お掃除ロボットとして知られる Roomba¹をベースとした、教育・開発者向けのプログラム可能なロボットである。iRobot Create はサーボモーターによって独立に動く二つの車輪を持ち、これにより前後移動・その場回転・カーブ移動が可能である。速度は-500 ~ 500mm/s まで出すことが可能である。また、二つのタイヤの回転数を内部で計算することによって、走った距離と回転角度 (オドメトリ情報)を知ることができる。しかし、このオドメトリ情報は必ずしも正しい値ではなく、誤差を含む値である。モーターの回転数が高出力の時と、低出力の時の誤差は特に大きく、出発地点からの相対位置を知るためには不十分な精度である。これについての問題解決の方法として、環境地図とレーザーレンジセンサーを用いた自己位置推定を行っている。詳しくは 2.4 節で述べる。また、iRobot Create 本体には様々なセンサーが標準で搭載されている。図 2.2 に示すバンパーセンサー、ホイールセンサー、PSD センサー、IR レシーバーである。このうち、バンパーセンサーは SUV の前面の左右にそれぞれ持ち、進行中に右のバンパーセンサーが反応したら左へ回転し軌道を修正するなど、このセンサーの値によって SUV の動作を決定することができる。また、バンパーは壁と衝突した時の衝撃を吸収してくれる。ホイールセンサーはタイヤが地面に着いているかを調べるセンサーである。タイヤが地面に着いていない時は、制御 PC の信号を無視して必ずモーターの回転を停止する機能を持つ。PSD センサーは赤外線を使って iRobot Create の側面の壁との距離を測ることができるが、SUV はレーザーレンジセンサーによって、広

¹iRobot Create および Roomba は米国 iRobot Corporation の米国およびその他の国における登録商標です。

範囲の障害物までの距離を測定可能なため、本研究ではPSDセンサーは使用しない。IR レシーバーは赤外線を受信することができるセンサーである。これにより iRobot Create に付属するリモコンから命令を送信することができる。

2.2.2 レーザーレンジセンサー

SUV は自分から見た障害物までの距離と角度を知ることができるレーザーレンジセンサーを搭載している。このレンジセンサーは物体にレーザー光をあて、その反射光を受光して物体までの距離を算出する。これを一定角度間隔で距離を測定することで、平面上に扇状の距離情報を最大 30m、角度 240 度の範囲で得ることができる (図 2.3 を参照)。レーザーレンジセンサーは SUV の目として、環境地図の生成や現在地の推定、障害物の検出などに利用される。

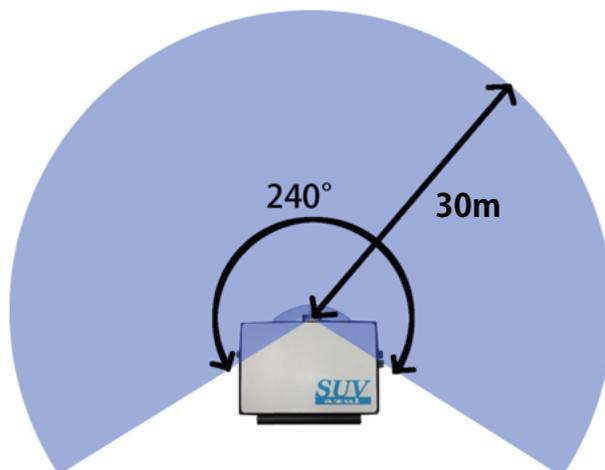


図 2.3: レーザーレンジセンサーの取得可能範囲

2.2.3 RGB-D カメラ

SUV は RGB-D カメラとして Kinect を搭載している。Kinect とは Microsoft から発売されている家庭用ゲーム機 Xbox360 用の入力デバイスであり、プレイヤーの動きや姿勢を認識することで、コントローラを使わずに操作が可能になるというものである。

Kinect は RGB カラーセンサー (カメラ) に加えて、カメラから見た物体までの距離が計測できる深度センサーを持ち、平面上の障害物までの距離が分かるレーザーレンジセンサーと違い、物体の 3 次元スキャンを行うことが可能である。この深度センサーは赤外線レーザーで特徴のあるパターンを照射し、赤外線カメラ

でそのパターンを解析することで奥行きを計算することができる。Kinect は水平視野 57 度、垂直視野 43 度、センサー範囲は 1.2m ~ 3.5m の範囲を撮影することができる。図 2.4 は左が Kinect のカメラの画像で、右が深度センサーの画像である。RGB 画像は 640x480、Depth (深度) 画像は 320x240 で共に 30 フレーム/秒で取得できる。この二種類のデータを使って実際に 3 次元形状復元したものを図 2.5 に示す。本研究ではこの機能を利用して 3 次元地図の生成を行っている。詳しくは第 3 章で述べる。

また、Kinect は専用のソフトウェアを動作させるプロセッサが内蔵されており、このソフトウェアにより人の検出、トラッキングすることが可能である。この機能は 2.6 節の移動障害物の検出に利用している。



図 2.4: Kinect のセンサーデータ

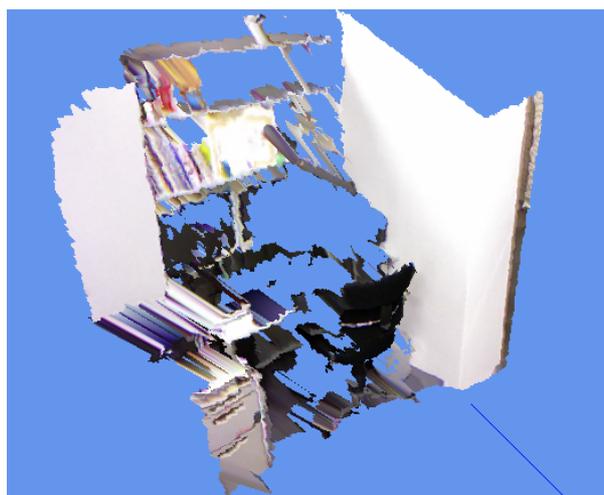


図 2.5: 3 次元形状の復元

2.3 環境地図生成

SUV が現在地を把握したり、目的地を決めてその場所まで自律走行するためには SUV が走行する環境を認識していなければならない。SUV が環境を認識するためには走行する環境の地図が必要となる。そこで、SUV は事前に環境内を走行することで、レーザーレンジセンサーの計測記録から環境地図を作成する。移動体の地図生成は、自己位置推定と地図生成を同時に行う必要があり、これは SLAM (Simultaneous Localization and Mapping) 問題として有名な問題である。SUV は ICP (Iterative Closest Point) アルゴリズムによりセンサーデータの重ね合わせを行い SLAM 問題を解決している。

SUV は環境地図として 2 次元の占有格子地図を使用する。占有格子地図とは、等間隔の格子に配置された確率変数で表現される地図のことであり、確率変数の値が高いほどその領域が物体によって占められている可能性が高いことを意味する。占有格子地図の求め方は次の通りである。この地図 \mathbf{m} を、時刻 t までの姿勢 $\mathbf{x}_{1:t}$ とセンサーの計測値 $\mathbf{z}_{1:t}$ から求めるには、すべての \mathbf{m} の可能性について $p(\mathbf{m}|\mathbf{z}_{1:t}, \mathbf{x}_{1:t})$ を計算し、尤度が最大となる \mathbf{m} を求めればよい。しかし、すべての \mathbf{m} について $p(\mathbf{m}|\mathbf{z}_{1:t}, \mathbf{x}_{1:t})$ を求めることは計算量的に不可能であるので、 \mathbf{m} のインデックス i を示す格子を m_i とし、

$$p(\mathbf{m}|\mathbf{z}_{1:t}, \mathbf{x}_{1:t}) = \prod_i p(m_i|\mathbf{z}_{1:t}, \mathbf{x}_{1:t})$$

と近似することで、各格子のバイナリ値を推定する問題に分解する。レーザーレンジセンサーの値から $p(m_i|\mathbf{z}_{1:t}, \mathbf{x}_{1:t})$ を求める方法は `inverse_range_sensor_model` [19] を使用している。

実際に計測したデータから地図を求める手法について説明する。各時刻において、尤度最大となる姿勢を探索し、その尤もらしい姿勢において地図を更新していくのが基本的な方法である。具体的な手順として、SUV を環境に走らせることによって得られるオドメトリ情報 $\mathbf{u}_{1:t}$ と、レーザーレンジセンサーの情報 $\mathbf{z}_{1:t}$ から以下の方法で求める。

1. x_0 を原点として、 $p(\mathbf{m}|\mathbf{z}_0, \mathbf{x}_0)$ を求める。
2. 時刻 $i (1 \leq i \leq t)$ に対して 3 を実行
3. ICP アルゴリズムにより $p(\mathbf{x}_i|\mathbf{m}, \mathbf{z}_i, \mathbf{x}_{i-1}, \mathbf{u}_i)$ が最大となる \mathbf{x}_i を探索する。
4. 2 で求めた \mathbf{x}_i によって $p(\mathbf{m}|\mathbf{z}_i, \mathbf{x}_i)$ を求め、 \mathbf{m} を `inverse_range_sensor_model` により更新する。

以上の方法で作った占有格子地図の例を図 2.6 に示す。画像の各画素の色の濃さが占有格子地図の各格子の確率変数の値の大きさを表す。黒は障害物有り、白は障

害物無し、灰色が初期状態を表している。各格子の間隔は 50mm, SUV が 300mm 進むごとに記録したデータから作成した。

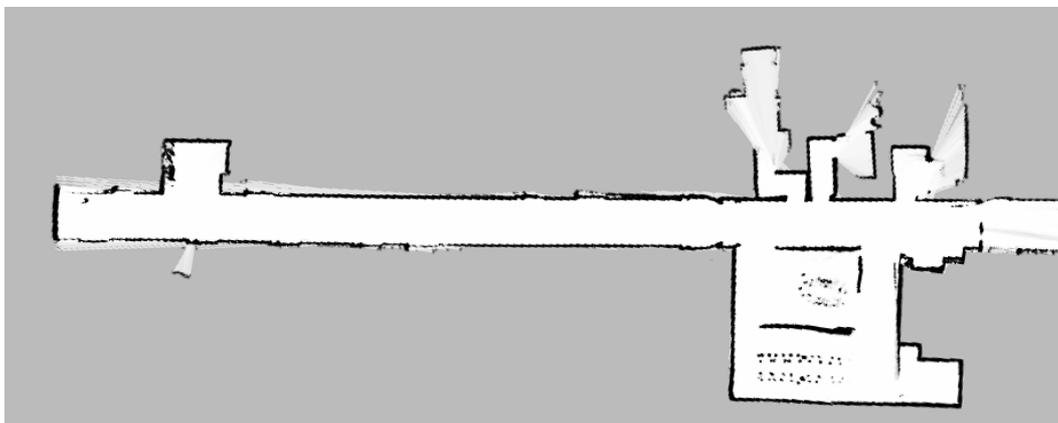


図 2.6: 生成された占有格子地図

2.4 自己位置推定

設計した経路に沿った柔軟な自律走行を行うためには SUV は環境地図上での自身の位置を常に知っていなければならないため、自己位置推定を行う必要がある。ここでいう位置とは、2次元のグローバルな座標系、つまり地図の座標系において、移動体の姿勢 $\mathbf{a}_t = (x_t, y_t, \theta_t)$ のことである。姿勢は移動体の位置 x, y と向き θ によって決まる値である。

前節で生成した環境地図と現在のレーザーレンジセンサーの値を比較することで、環境地図上での位置を推定する。環境地図生成と同様にレーザーレンジセンサーの値を利用するので、2.3節で説明した地図生成アルゴリズムのステップ3のように、ある時刻 t において $p(\mathbf{a}_t | \mathbf{m}, \mathbf{z}_t, \mathbf{a}_{t-1}, \mathbf{u}_t)$ が最大となるような \mathbf{a}_t を求めることで位置を推定することができる。しかし、実際の環境では物の配置が変わるなど、環境地図作成時と自己位置推定時でその場の環境が同一であるという保証がない。そのため、ここでは \mathbf{a}_t の精度よりも、ある程度動的な環境でも一定の誤差以内で \mathbf{a}_t を推定し続けることが重要となるので、確率で位置の推定を行うパーティクルフィルタ [6] を用いている。

パーティクルフィルタ (または逐次モンテカルロ法) とは、 N 個のサンプル点 (パーティクル) の集合を近似的に事後確率の分布関数として扱う方法である。ここでは事後確率 $p(\mathbf{a}_t | \mathbf{m}, \mathbf{z}_t, \mathbf{a}_{t-1}, \mathbf{u}_t)$ の分布関数をパーティクルフィルタで求めることで位置推定を行う。パーティクルフィルタによる位置推定は次の3ステップで行われる。

予測

各パーティクルについて、次の運動モデルに従って次状態の \mathbf{a}_t を予測する。

$$\mathbf{a}_t = \mathbf{a}_{t-1} + \mathbf{R}_{t-1}(\mathbf{u}_t + \mathbf{w}_t)$$

\mathbf{R}_t : \mathbf{a}_t の向きへの回転行列

\mathbf{w}_t : オドメトリ誤差を表す正規分布 $N(0, \Sigma_W)$ に従った乱数 (ここでの Σ_W は経験的に求めている。また乱数はボックスミュラー法 [3] により計算する)

尤度計算

各パーティクルについて尤度 $p(\mathbf{z}_t | \mathbf{a}_t, \mathbf{m})$ を個々の計測値に対する尤度の積

$$p(\mathbf{z}_t | \mathbf{a}_t, \mathbf{m}) = \prod_{k=1}^K p(\mathbf{z}_t^k | \mathbf{a}_t, \mathbf{m})$$

で求める。個々の尤度は以下の計測モデルを使って求める。

$$p(\mathbf{z}_t^k | \mathbf{a}_t, \mathbf{m}) = \begin{pmatrix} \mathbf{z}_{\text{hit}} \\ \mathbf{z}_{\text{short}} \\ \mathbf{z}_{\text{max}} \end{pmatrix} \cdot \begin{pmatrix} p_{\text{hit}}(\mathbf{z}_t^k | \mathbf{a}_t, \mathbf{m}) \\ p_{\text{short}}(\mathbf{z}_t^k | \mathbf{a}_t, \mathbf{m}) \\ p_{\text{max}}(\mathbf{z}_t^k | \mathbf{a}_t, \mathbf{m}) \end{pmatrix}$$

$\mathbf{z}_{\text{hit}}, \mathbf{z}_{\text{short}}, \mathbf{z}_{\text{max}}$: $\mathbf{z}_{\text{hit}} + \mathbf{z}_{\text{short}} + \mathbf{z}_{\text{max}} = 1$ となるように決めた重み定数 (経験的に求めた)

$p_{\text{hit}}(\mathbf{z}_t^k | \mathbf{a}_t, \mathbf{m})$ 、 $p_{\text{short}}(\mathbf{z}_t^k | \mathbf{a}_t, \mathbf{m})$ 、 $p_{\text{max}}(\mathbf{z}_t^k | \mathbf{a}_t, \mathbf{m})$ はそれぞれ正しい計測時の局所的な計測雑音、想定外の物体、計測失敗の確率分布を表す。それぞれの分布図のグラフを図 2.7 に示す。

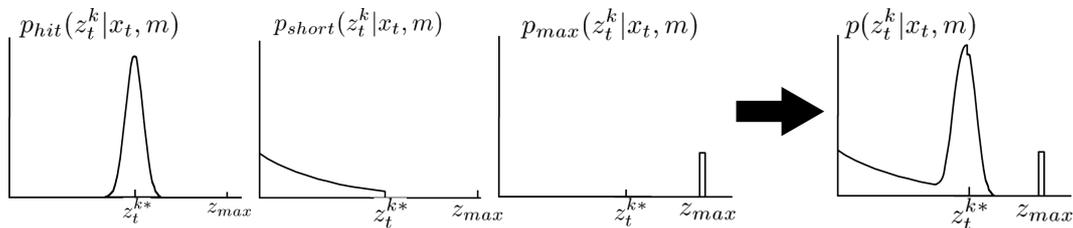


図 2.7: 確率分布図

リサンプリング

前の状態の各パーティクルの尤度を元にパーティクルを選び直す。高い尤度が与えられたパーティクルを多くのパーティクルにコピーし、尤度の低いパーティクルを消去することにより、パーティクル数を一定に保つ。

位置推定はオンタイムで推定を行う必要があるため、処理速度とロバスト性が求められる。そこで、処理速度を向上させるために地図とのマッチングを行うセンサーの分解能を15度に設定し、ロバスト性を上げるためにパーティクルを多めの500個用意して位置推定を行っている。1回の位置推定にかかる処理時間は10msec以下で、これを1秒間に4回繰り返すことで、オドメトリの誤差に対してロバストな位置推定を行っている。

パーティクルフィルタによる位置推定が、オドメトリのみで位置を計算した場合に比べてどの程度正しく位置を推定できているかを示したものが図2.8と図2.9である。これはリモコンによるマニュアル操作で同じコースを6周したときの様子である。図2.9のグラフから分かるように、位置推定なしのオドメトリ情報のみでは誤差が累積されており、パーティクルフィルタによる位置推定ありの結果が1周目から6周目まで一定の誤差範囲内に留まっている。図2.9の軌跡を見ても位置推定なしの場合は誤差の累積のため現在地情報と実世界の現在地の対応が全く取れていないが、位置推定ありの方は安定して位置推定ができていることが分かる。

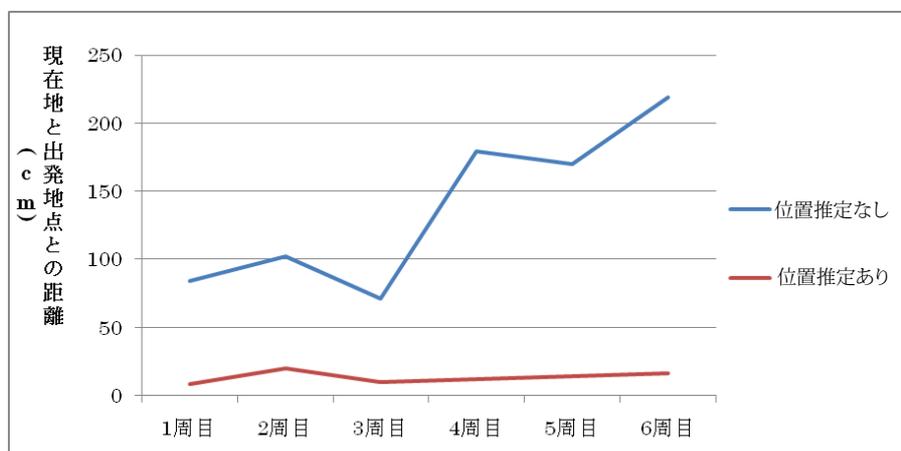


図 2.8: 周回後の現在地情報と出発点との距離

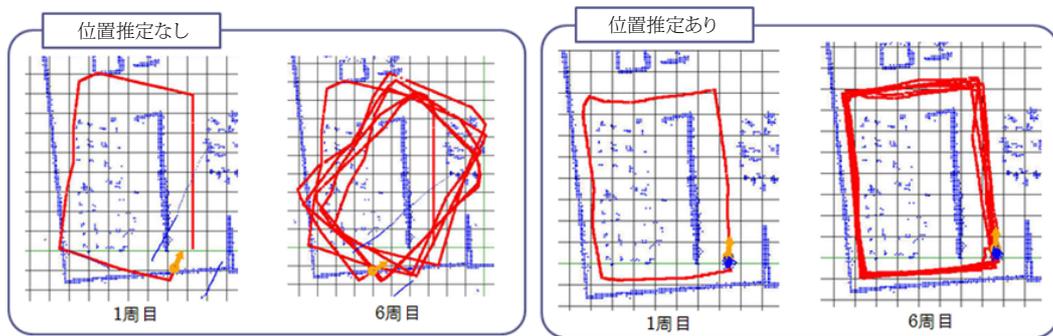


図 2.9: 位置推定なし・ありの場合のそれぞれの軌跡

2.5 SUV の自律走行

SUV は目的地が設定されると、現在の位置から指定された任意の座標に対して走行可能な経路を生成し、自律走行を行う機能を持つ。本節ではその手法について説明する。環境地図に用いた占有格子地図は地図上の格子ごとに物体に占められている確率を表現しているが、一定の閾値以下の領域を走行可能領域として、走行可能領域に経路生成のためのグラフ構造を自動で生成する。生成手法は、次の通りである。まず、図 2.10 の黒い線の部分で表されているような障害物から一定距離内の領域を塗りつぶし、その境界線にノードを一定間隔で追加する。さらに、障害物から一定距離外の領域を塗りつぶし、その境界線にも同様にノードを追加する。図 2.10 の点はこの手法で生成されたノードである。生成されたそれぞれのノードについて一定距離内にあるノードをつなげることによってグラフ構造を作る。図 2.11 は実際に生成されたグラフ構造である。

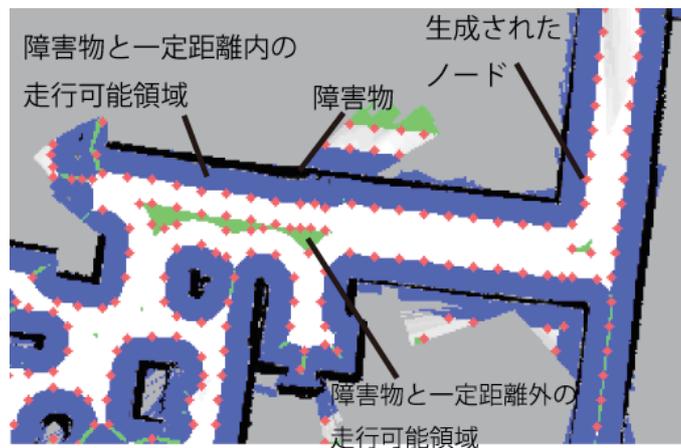


図 2.10: 地図上の障害物とその境界線

次にグラフ構造から経路を生成する手法を説明する。環境地図の走行可能領域で任意の場所が目的地として指定された時、現在地と目的地のそれぞれの最近傍のノード間を A* アルゴリズムで探索することで経路を生成する。A* はダイクストラ法の推定値が小さいものから順に探索し、経路が見つかった時点で探索を終了するアルゴリズムである [11]。このアルゴリズムは目的地までの推定値をユークリッド距離とすることで最短経路を探索できることが保証されている。図 2.11 に示す赤色の線は実際に生成された経路である。また、占有格子地図上では通行可能な領域ではあるが、人間の都合によって移動体に通って欲しくない状況や、地図作成後に障害物が置かれたことによって通行不可能となる状況が考えられる。これに対しては、前者は GUI 上から簡単に手動でグラフ構造を編集できる機能、後者はレーザーレンジセンサーから取得した値で地図上に矛盾が生じた際にグラフ構造を更新する機能を用意することで柔軟な経路生成を可能としている。

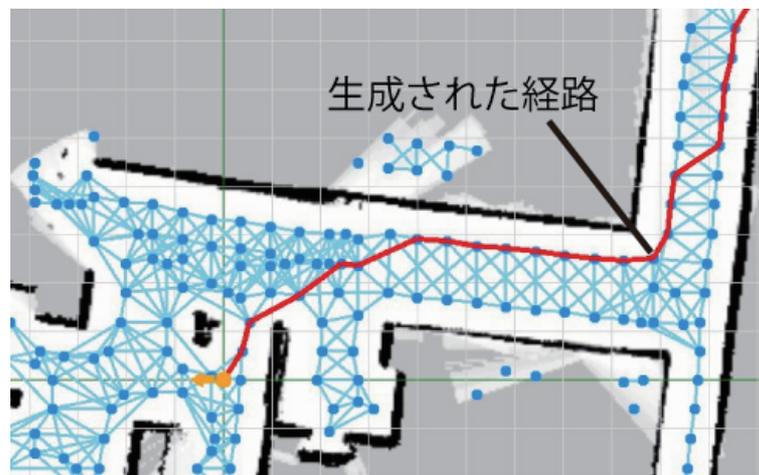


図 2.11: 地図に付与されたグラフ構造と生成された経路

この手法では、点と点を線で結ぶように経路を生成したため、SUV が実際に経路を走行する際に経路を忠実にトレースしようとする、各ノード上で一時停止し、次のノードの向きへその場回転を行い、直進するという行動を取る。しかしそのような行動は、その場回転を行うため効率が悪く、位置推定の誤差や車輪の回転のずれにより忠実に経路をトレースすることが難しい。そのため、対向二輪型の移動機構を持つ SUV にあった軌道計画を行う必要がある。また、ノードを目標点とするのではなく、現在の SUV の位置を基に経路上に一時的な目標地点を設定し、常に目標となる点を更新することで、ずれに対して強い制御を行うことができる。SUV は、旋回半径と移動速度を指定することで走行を行う。そこで、図 2.12 の青色の線のように、現在の SUV の位置から一定距離先の経路上の点を一時的な目標地点として設定し、SUV の位置と目標地点を通り、SUV の進行方向を接

線とする円の半径を旋回半径に指定することで、SUV は図 2.12 に示す赤色の線に沿うように走行する。また、SUV の位置が経路から大きく外れてしまった場合や U ターンのような急な方向転換を行う必要がある場合には、その場回転を行い経路に復帰する。位置推定を行う度に目標地点を更新することで、生成された経路にできるだけ沿うように、スムーズに移動することができる。

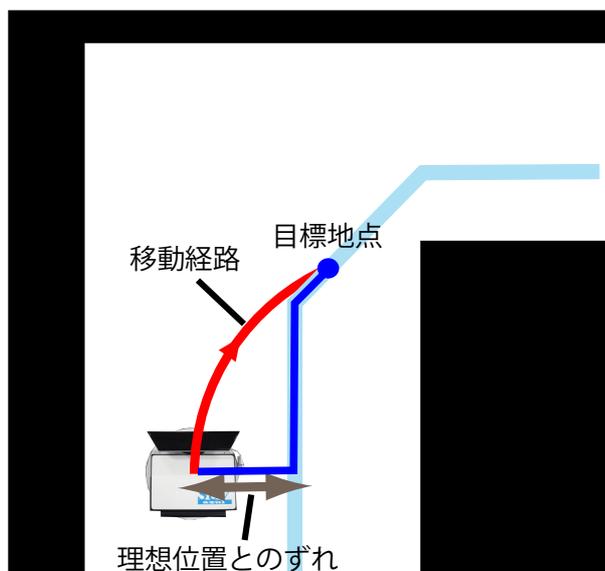


図 2.12: 自律走行のアルゴリズム

2.6 移動障害物の検出

SUV は安全に走行するため、自律走行中に移動障害物を検出したら移動を停止し、衝突の危険がなくなったら移動を再開するという機能がある。本節ではその仕組みについて説明する。

SUV はレーザーレンジセンサーと Kinect を同時に用いた移動障害物の検出を行う。レーザーレンジセンサーによる検出方法は、センサーの値と地図を比較することで地図にない障害物を一定時間ごとに発見し、この情報からオプティカルフローを用いることで、動いている障害物を検出する方法である。Kinect は距離画像に基づいて人間を検出する機能を有しているのをそれを用いて人間を検出する。レーザーレンジセンサーは検出距離が長く、距離の誤差が少ないため移動障害物の進行方向を計算するのに適しているが、動いている障害物にしか反応せず、SUV が移動している場合にはオプティカルフローを正確に計算できない。一方、Kinect は検出距離が短い、静止している人間を検出でき、SUV が移動していても人間を検出することができる。このため、レーザーレンジセンサーと Kinect の

検出結果を統合することでより確実に移動障害物（特に人間）を検出することができる。ここでは、検出時点で静止している人間も移動障害物と見なしている。実際に検出した時の様子を図 2.13 に示す。左上の画像が Kinect で取得した RGB 画像で、左下が Kinect で取得した障害物までの深度と人間の検出結果である。右の環境地図上に検出した人間の位置がピクトグラムで表示される。

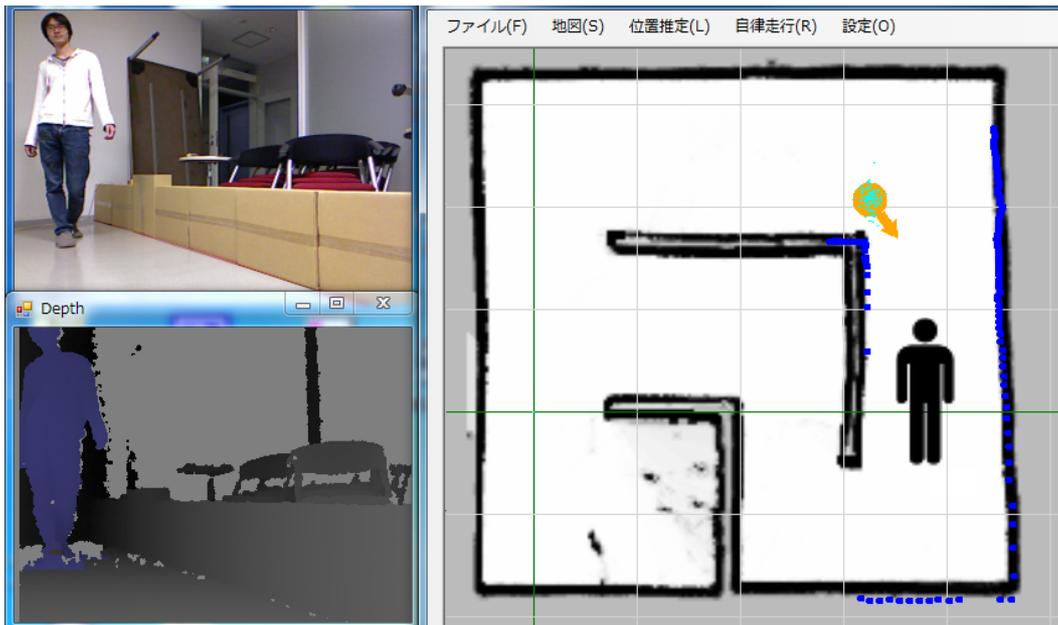


図 2.13: SUV の移動障害物の検出

次章では、SUV を利用した 3 次元地図の自動生成について詳しく説明する。

第3章 3次元地図の自動生成

本研究ではナビゲーションなどのインタフェースとしての利用が期待される3次元地図を、自動的に生成する手法を提案する。具体的には、前章で説明したSUVに環境中を自律走行させることでデータの収集を行い、収集したデータから3次元地図を生成する。ここで言うデータとはRGB-Dカメラ(カラーおよび深度センサー)によって取得できるRGB画像とDepth画像およびSUVの位置情報を指す。SUVが位置推定および自律走行をするための手がかりとして前章で説明した環境地図(占有格子地図)を事前に作成しておく。

3次元地図を生成するためには、環境地図生成と同様に収集したデータの重ね合わせを行う必要がある。本研究ではSUVが事前に環境地図を持っていることを利用し、従来のセンサーデータのみから重ね合わせを行う手法を拡張した、センサーデータだけでなく環境地図とも重ね合わせを行う手法を提案する。これについて3.1節で詳しく説明する。センサーデータは点の集合であり、センサーデータを重ね合わせて生成する3次元地図は膨大な数の点の集合となるためデータ量が非常に大きくなってしまう。そのような3次元地図を実際に利用することは非常に困難であるため、3次元地図の構造を単純化する必要がある。そこで、建物の内装を構成する主要要素である壁や床といった平面に着目し、3次元地図中の平面を抽出しポリゴン化する機能を実現した。これについては3.2節で詳しく説明する。3.3節ではデータの収集を自動化するため、SUVがデータ収集時に走行する経路を生成する機能について説明する。

3.1 3次元地図生成

収集したデータ(RGB画像、Depth画像、SUVの位置情報)から3次元地図生成を行う。RGB-Dカメラのデータから3次元形状の復元を行い、3次元形状モデルの重ね合わせを行うことで3次元地図を生成する。環境地図生成の際にもセンサーデータの重ね合わせを行い地図を生成したが、ここでは3次元地図をより利用しやすくするために環境地図と同じ座標系で3次元地図を生成する必要がある。そのため3次元地図生成ではセンサーデータだけでなく環境地図とも重ね合わせを行う。収集したデータにはSUVの環境地図上での位置情報も含まれるため、重ね合わせを行う際に初期値としてSUVの位置情報を利用することで、ある程度環

境地図と対応の取れた位置にセンサーデータを配置することができ、環境地図と重ね合わせを行うことができる。しかし、SUVが取得できる位置情報は x, y 座標および左右方向の向き(ヨー)のみで3次元的な位置および角度情報は取得することができない。そのためセンサーが必ず水平に固定されていないと取得したセンサーデータおよび位置情報のみで環境地図の座標系と対応するように3次元的な重ね合わせを行うことは難しい。しかし、実際にはSUVは移動しながらデータを収集するため、センサーが常に水平に固定されているとは言えない。そこで、取得したセンサーデータから3次元的な位置情報であるSUVの z 座標および上下方向の向きと進行方向に対しての回転角(ピッチ、ロール)の推定を行うことでSUVの3次元的な位置情報を取得する。

3次元的な位置情報は、取得したセンサーデータから床の検出を行い、床が水平面と一致するような z 座標およびピッチ、ロールの推定を行うことによって求められる。これについては3.1.1項で詳しく説明する。センサーデータの重ね合わせには、画像特徴点のマッチングとICPアルゴリズムを組み合わせたRGBD-ICPアルゴリズム[12]を、環境地図ともマッチングできるように拡張したアルゴリズムを用いて行う。このアルゴリズムについては3.1.2項で詳しく説明する。3.1.3項では、生成した3次元地図を環境地図と対応付けを行うことで、3次元地図の精度の検証を行う。

3.1.1 床の検出と位置推定

RGB-DカメラはSUVに固定されているため、床の検出はRGB-Dカメラが床に対してある程度水平になっていることを前提として行う。床は面の方程式 $ax + by + cz - 1 = 0$ で表すことができるのでRANSAC(Random Sample Consensus)アルゴリズム[7]を用いてもっとも床らしいパラメータ a, b, c を求める。RANSACアルゴリズムとは、外れ値を含むデータから尤もらしいパラメータをロバストに推定するアルゴリズムである。全てのデータからランダムにサンプルを取り出し、サンプルからパラメータを求める。求めたパラメータに全データを当てはめ、観測されたデータとの誤差を計算し、誤差が許容範囲内であればそのパラメータに投票を行う。これを繰り返し行い得票数が最も多いパラメータに投票したデータを抽出する。抽出したデータをもとに再度パラメータを求めることで尤もらしいパラメータを推定する。以下にRANSACアルゴリズムで床のパラメータ a, b, c を求める手順について説明する。

1. 全ての点からランダムに3点 P_1, P_2, P_3 を取り出す
2. 3点を含む面がある程度水平でなければ手順1に戻る
RGB-D カメラは床と水平に設置されているため、3点を含む面がある程度水平でなければ床ではないと見なし手順1からやり直す。水平かどうかの判断は、水平面と3点を含む面のずれを計算することで行う。面のずれは垂直上向きベクトルと3点を含む面の法線ベクトルの内積によって求める。面の法線ベクトルは $\overrightarrow{P_1P_2}$ と $\overrightarrow{P_1P_3}$ の外積を求めることで計算する。
3. 3点を含む面を計算する
面の方程式 $ax + by + cz - 1 = 0$ と3点 P_1, P_2, P_3 の連立方程式を解いてパラメータ a, b, c を求める。
4. 求めた面と全ての点の誤差を計算し投票を行う
求めた面と点の誤差は面と点の距離として計算を行う。本論文では面と点の距離が1cm以下の場合はそのパラメータに対して投票を行った。
5. 手順1-4を繰り返し実行する
6. 得票数が最も多いパラメータに投票した点を抽出する
7. 抽出した点から尤もらしいパラメータを求める
抽出した点と面との距離の総和が最小になるような最小二乗問題を、特異値分解を用いて解くことで尤もらしいパラメータ a, b, c を求める。

ただし、データによってはデータ中に床の領域が含まれず、正しく床が検出されない場合がある。そのため手順5で抽出された点が極端に少ない場合は床の検出に失敗したものとして、そのデータを除外する。

RANSAC アルゴリズムは繰り返し回数が多く、センサーデータすべての点を用いてアルゴリズムを適用するとかなりの計算時間がかかってしまうため、センサーデータからサンプリングしたのに対してアルゴリズムを適用する。本論文ではセンサーデータからサンプリングした1000個の点を全データとし、繰り返し回数を1000回としてアルゴリズムを適用した。実際に床を検出した様子を図3.1に示す。床の検出は左図のように3次元に復元されたものを利用して検出する。床として検出された面に含まれる領域を赤色で示している。右上および右下の図は検出された領域を Depth 画像および RGB 画像に反映したものである。

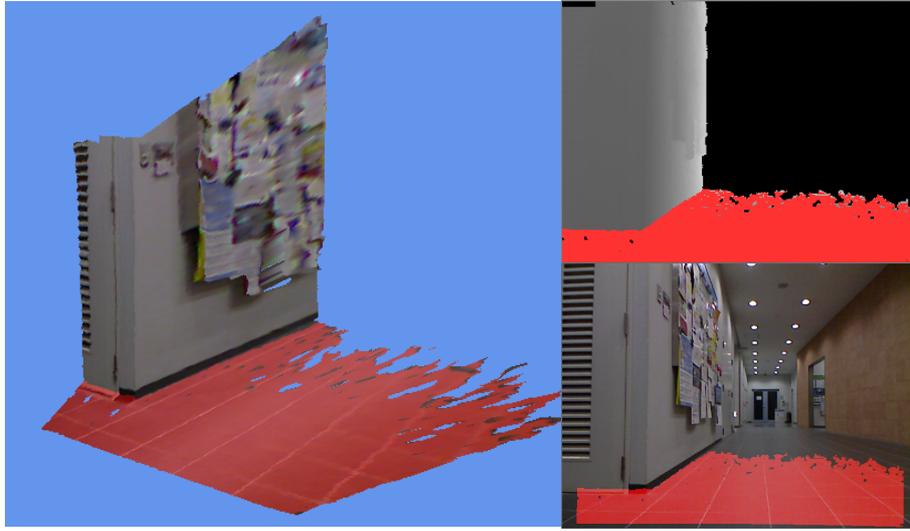


図 3.1: 床の検出

次に、検出した床を利用して SUV の 3 次元的な位置情報 (高さ p_z , ピッチ θ_p , ロール θ_r) の推定を行う。床が水平であると仮定すると、床上の点 p を水平面上の点 p' に変換する $p' = R \cdot p + T$ となる回転行列 R と平行移動行列 T から各パラメータを計算することができる。回転行列 R および平行移動行列 T は水平面と検出した床との対応点 3 点から求める。その手順について説明する。RGB-D カメラの向きは y 軸正方向とし、水平面上の 3 点は $P_1(0, 0, 0), P_2(0, 1, 0), P_3(1, 0, 0)$ とする。

1. 点 P_1, P_2, P_3 と対応する検出した床上の点 Q_1, Q_2, Q_3 を求める。点 P_1, P_2 を検出した床に垂直に降ろした点を点 Q_1, Q'_2 とする。 $\overrightarrow{Q_1 Q_2} = k \overrightarrow{Q_1 Q'_2}$ かつ $|\overrightarrow{Q_1 Q_2}| = 1$ となる点を点 Q_2 とし、点 Q_1 から x 軸方向に $|\overrightarrow{Q_1 Q_3}| = 1$ となる点を点 Q_3 とする。

2. 以下のようにベクトル x_1, x_2, x_3 と y_1, y_2, y_3 を求める。

$$\begin{aligned} x_1 &= \overrightarrow{P_1 P_2}, x_2 = \overrightarrow{P_1 P_3}, x_3 = x_1 \times x_2 \\ y_1 &= \overrightarrow{Q_1 Q_2}, y_2 = \overrightarrow{Q_1 Q_3}, y_3 = y_1 \times y_2 \end{aligned}$$

3. $X = \begin{bmatrix} x_1 & x_2 & x_3 \end{bmatrix}$, $Y = \begin{bmatrix} y_1 & y_2 & y_3 \end{bmatrix}$ に対して $X = RY$ が成り立つので、両辺に Y^{-1} をかけて $R = XY^{-1}, T = P_1 - RQ_1$ により回転行列 R および平行移動行列 T を求める。

4. x 軸に対する回転行列を R_x , y 軸に対する回転行列を R_y とすると回転行列 R は $R = R_x(\theta_p) \cdot R_y(\theta_r)$ と表すことができ、ピッチ θ_p およびロール θ_r を求めることができる。高さ p_z は平行移動行列 T の z 成分から求まる。このとき平行移動行列の x 成分および y 成分は 0 になるはずである。

実際に床を水平に修正したものを図 3.2 に示す。修正前では上の図で奥方向、下の図では右にわずかに傾いているが、修正後は水平になっていることが分かる。

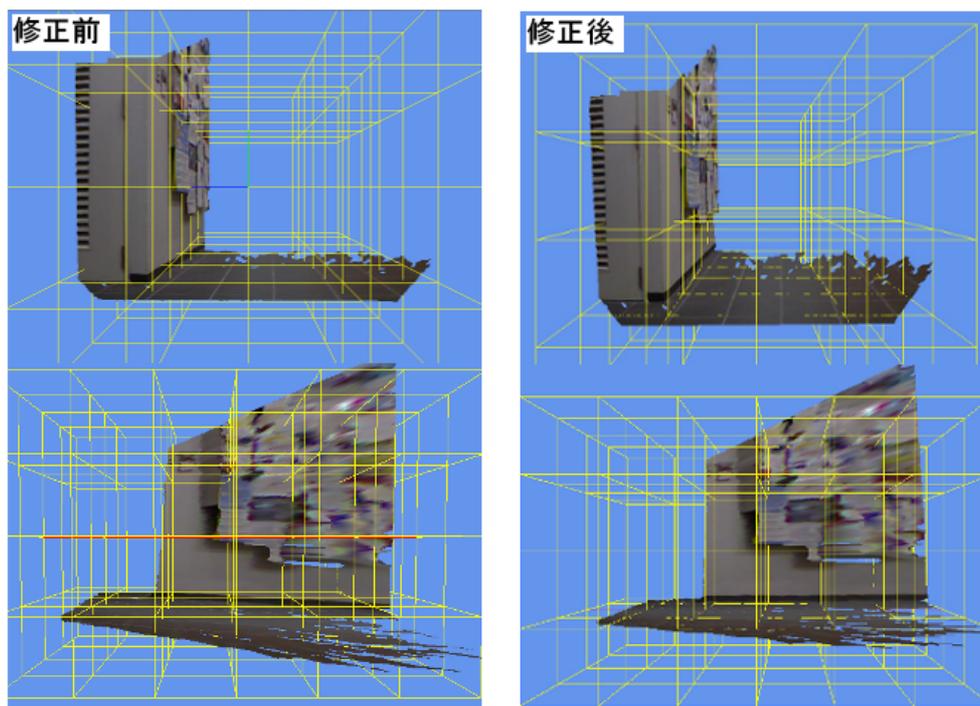


図 3.2: 床の向き修正前後の比較

3.1.2 センサーデータの重ね合わせ

前項で求めた SUV の 3 次元的な位置情報とセンサーデータをもとに復元した 3 次元モデルの重ね合わせを行う。本研究では画像特徴点のマッチングと 3 次元形状モデルの位置合わせを行う ICP アルゴリズムを組み合わせた RGBD-ICP アルゴリズムをさらに環境地図ともマッチングできるように拡張したアルゴリズムを用いる。重ね合わせを行うには回転および平行移動を行う剛体変換行列 A を求める必要がある。変換行列 A は次式によって SUV の位置情報から計算することができる。

$$A = \text{TR}_z(\theta_y)\text{R}_x(\theta_p)\text{R}_y(\theta_r)$$

本手法では、床が水平になるように SUV の位置情報を修正しているため、床が水平であると仮定すると SUV の z 座標およびピッチ θ_p とロール θ_r は修正する必要がない。そのため SUV の xy 座標と左右方向の向き θ_y のみ重ね合わせにより推定

を行う。本手法は従来のICPアルゴリズムと同じく、点の選択、マッチング、重み付け、不要な対応点の除去、誤差量の計算、誤差量の最小化のステップで行われる。それぞれについて以下で説明する。

1. 点の選択

重ね合わせを行う入力フレームの点群から特徴点の集合 F_s 、環境地図とのマッチングに用いる点の集合 M_s 、点群のマッチングに用いる点の集合 P_s を抽出する。 F_s はSURF (Speeded Up Robust Features)[9]により画像の局所特徴量を計算して抽出を行う。局所特徴量とは画像の局所的な領域から特徴量を抽出する手法であり、SURFは同様に局所特徴量であるSIFT特徴量[15]の計算を高速化した手法である。SURFおよびSIFT特徴量はスケーリングや回転に対して不変な特徴量を計算することができる。しかし、RGB画像から抽出した特徴点是对応する点が点群中には存在しない可能性がある。そのような点は重ね合わせに利用できないため点群中に対応する点が存在する点だけを F_s とする。実際にSURFを抽出した様子を図3.3に示す。左の図の赤色の円一つ一つが特徴量が抽出された領域であり、円の中心が特徴点となる。 F_s を3次元モデル上に表示した様子を右の図に示す。環境地図とのマッチングに用いる点の集合 M_s には点群のうち変換後の z 座標がSUVのレーザーレンジセンサーの高さあたりになる点を選択する。点群に含まれるすべての点を用いて重ね合わせを行うことは、かなりの計算量が必要になり現実的ではないため、点群からマッチングに用いる点の集合 P_s をサンプリングする必要がある。本研究では、処理時間とマッチング精度を考慮して、元の点群のおよそ4分の1の点を一様にサンプリングした。

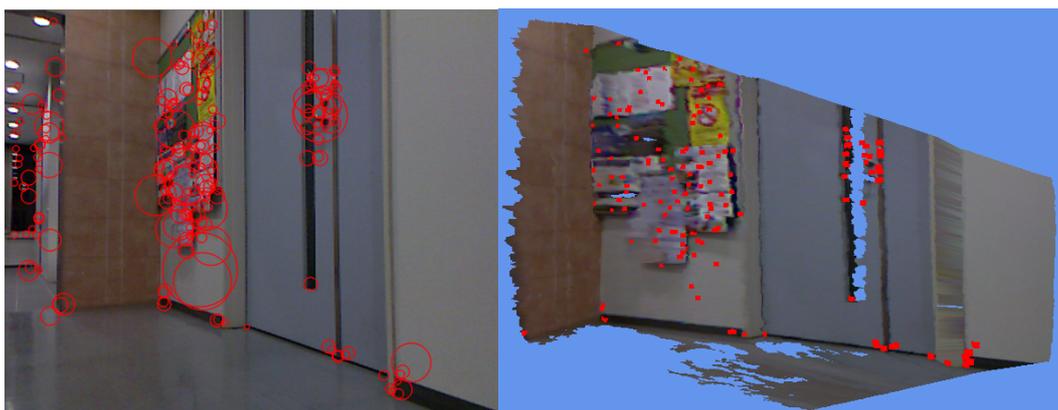


図 3.3: SURF 特徴点の抽出

2. 重ね合わせ先となるデータの選択

重ね合わせ先となるターゲットフレームは入力フレームと重なりやすいものである必要があるため、入力フレームと画像特徴点が最も多くマッチングするフレームを選択する。ただし、全てのフレームに対して計算することは困難であるため、入力フレームの直前の数フレームに対してこの処理を行う。入力フレームとマッチングできるフレームが存在しない場合は入力フレームの直前のフレームをターゲットフレームとする。画像特徴点の対応点は特徴ベクトルのユークリッド距離を計算し、距離が短いものを選択することで求める。図3.4に画像特徴点のマッチングを行い、マッチングした2点間に黄色の線を引いたものを示す。



図 3.4: 画像特徴点のマッチング

3. マッチングと重み付け

手順1で選択した点の集合 F_s, M_s, P_s と対応する点をそれぞれ求め、 F_s とそれに対応する点の集合を A_f 、 M_s とそれに対応する点の集合を A_m 、 P_s とそれに対応する点の集合を A_p とする。 A_f は入力フレームとターゲットフレームの画像特徴点のマッチングを計算することで求める。このとき点 f_s に対応する点 f_t が存在しない場合は点 f_s を F_s から削除する。変換後の座標は、セ

ンサーデータ上の点 p 、剛体変換行列を t として、 $t \cdot p$ で求めることができる。 A_m は環境地図上の点を3次元座標に変換したもののうち点 $t \cdot m_s$ に最も近い点を点 m_t として求める。 A_p はターゲットフレームの点群から点 $t \cdot p_s$ に最も近い点を点 p_t として求める。 M_s および P_s の最近傍の探索は k-d 木 [2] を用いて行った。剛体変換行列 t は SUV の位置情報から求める。重みは全て均一とした。

4. 外れ値の除去

外れ値が存在すると誤差量の最小化に悪影響が出るため A_f , A_m , A_p からそれぞれ外れ値の除去を行う。 A_f では前項でも説明した RANSAC アルゴリズムを用いて正解データを抽出し、それ以外の点を外れ値として除去する。 A_m および A_p では一定距離以上離れた点を外れ値として除去する。また、 A_p では点 p_t が3次元形状モデルの境界になっている点も外れ値として除去する。Depth 画像の隣接するピクセルにデータが存在しないピクセルがある場合はその点を3次元モデルの境界とする。

5. 誤差量の計算と最小化

A_f は2点間の距離の2乗、 A_p および A_m は点と平面の距離の2乗から誤差量を求める。誤差量は次式によって計算する。法線ベクトルは点 p_t 周辺の点群および点 m_t 周辺の環境地図上の点の主成分分析によって求める。本論文では $\alpha = 0.3, \beta = 0.3, \gamma = 0.4$ とした。

$$\begin{aligned}
 e = & \alpha \left(\frac{1}{|A_f|} \sum_{i \in A_f} w_i |t(f_s^i) - f_t^i|^2 \right) \\
 & + \beta \left(\frac{1}{|A_p|} \sum_{j \in A_p} w_j |(t(p_s^j) - p_t^j) \cdot \mathbf{n}_j|^2 \right) \\
 & + \gamma \left(\frac{1}{|A_m|} \sum_{k \in A_m} w_k |(t(m_s^k) - m_t^k) \cdot \mathbf{n}_k|^2 \right) \\
 & (\alpha + \beta + \gamma = 1)
 \end{aligned}$$

w: 重み, t: 剛体変換行列, n: 法線ベクトル

誤差量の最小化には Levenberg-Marquardt 法 [14, 16] を用いる。Levenberg-Marquardt 法とは Gauss-Newton 法の一つで次式の2乗和の形の最小値を求める方法である。誤差量 e の式を次式に変形し Levenberg-Marquardt 法を適用する。最適化するパラメータ \mathbf{x} は x, y 座標と左右方向の向き θ_y である。

$$E(\mathbf{x}) = \frac{1}{2} \sum_{i=1}^n (f_i(\mathbf{x}))^2$$

$\mathbf{f}(\mathbf{x}) = [f_1(\mathbf{x}), \dots, f_n(\mathbf{x})]^T$ 、 $\mathbf{f}(\mathbf{x})$ のヤコビ行列を \mathbf{J} として \mathbf{x}_n を次式で更新して最適化する。

$$\mathbf{x}_{n+1} = \mathbf{x}_n - [\mathbf{J}^T \mathbf{J} + \lambda \mathbf{I}]^{-1} \mathbf{J}^T \mathbf{f}(\mathbf{x}_n)$$

λ : スカラー量, \mathbf{I} : 単位行列

手順 3-5 を誤差量が十分小さくなるか繰り返し回数が一定回数を超えるまで繰り返し行う。

3.1.3 環境地図との対応付け

SUV が一定間隔進むたびに RGB-D カメラの値と SUV の位置情報を記録し、走行終了後にセンサーデータの重ね合わせを行うことで地図を生成した。実際に生成した地図を図 3.5 に示す。生成された地図がどの程度正確か、環境地図と比較することで検証した。3次元地図中のレーザーレンジセンサーの高さ付近のデータを環境地図にマッピングしたものを図 3.6 に示す。この図から、生成した3次元地図が環境地図と正確に重なっていることが分かる。

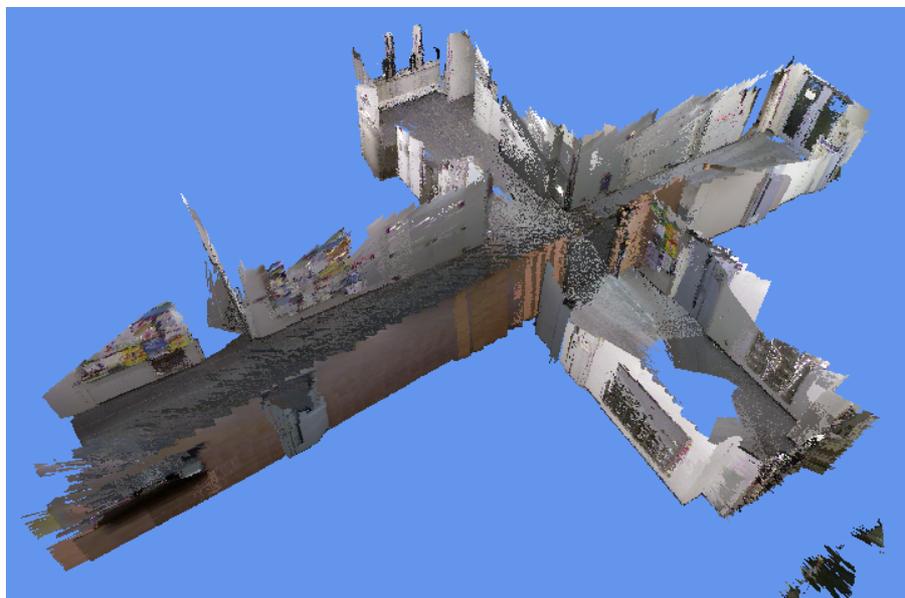


図 3.5: 生成された 3次元地図

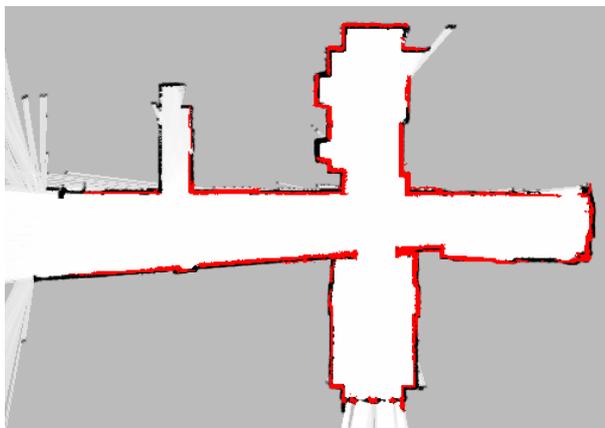


図 3.6: 環境地図と生成された3次元地図の対応付け

その他の手法で生成した3次元地図と比較するため、その他の手法で生成した3次元地図を同様に環境地図上にマッピングしたものを図3.7に示す。①は前項で説明した重ね合わせを行わずSUVの位置情報だけを基に3次元地図を生成したものである。SUVの位置情報に誤差があるため環境地図とうまく重なっていないことが分かる。②は環境地図との対応点だけで重ね合わせを行った3次元地図である。環境地図とは正確に重なっているが、点群同士の重ね合わせを行っていないため④の提案手法と比べ赤い線の幅が太くなり点群同士がしっかりと重なっていないことが分かる。また、②の手法では、床を事前に検出し補正することで2次元的な重ね合わせだけで地図生成を行ったが、3次元的な重ね合わせを行う場合、環境地図との重ね合わせだけでは2次元的な対応しか表現できないため正確に重ね合わせることができないことが予想される。提案手法では特徴点や点群とも重ね合わせを行うため最適化するパラメータを変更することで3次元的な重ね合わせも行うことができる。③はRGBD-ICPを用いて生成した3次元地図を示している。環境地図との重ね合わせを行っていないため環境地図とのずれが目立つ。本研究ではSUVの位置を初期値としたため環境地図から大きくずれてしまうことはないが、環境地図とずれた点群とマッチングしてしまうとマッチングに失敗するまで徐々に誤差がたまっていってしまう。以上の結果から、事前に環境地図が生成されており、データ収集時の環境地図上での位置情報が取得できている状況において、本提案手法は有効であるといえる。

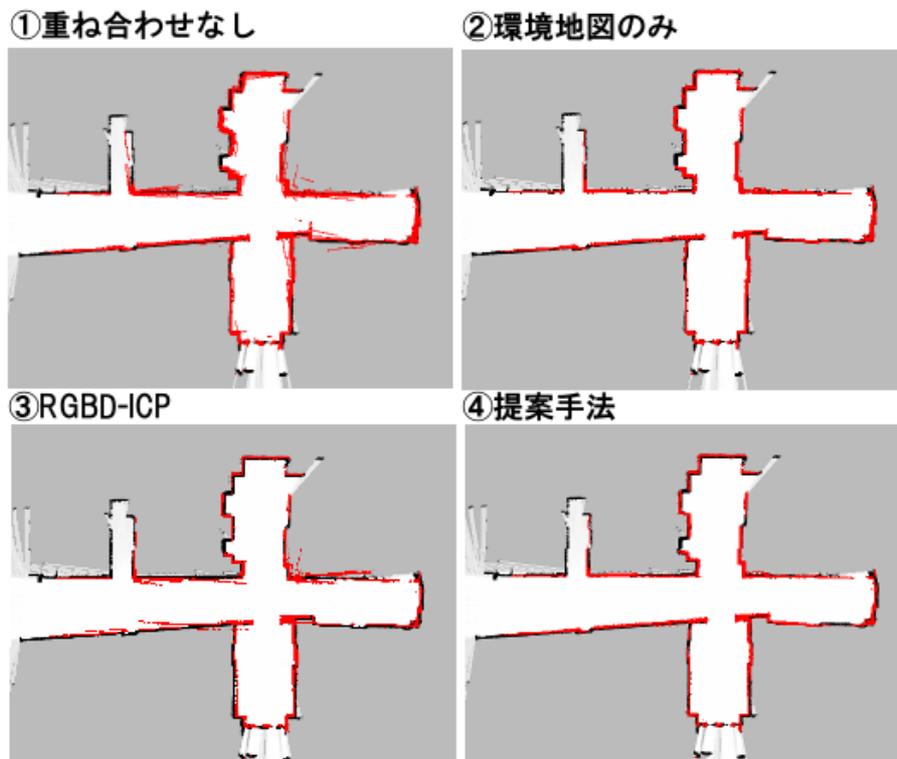


図 3.7: 各手法の比較

3.2 3次元地図のポリゴン化

前節で説明した3次元地図は膨大な数の点の集合であり、データ量が非常に大きく扱うのが非常に困難である。そこで点の集合である3次元地図から面を検出しポリゴン化することで3次元地図の構造の単純化を行う。本研究では、特に建物を構成する主要な要素である壁や床に着目し、ある程度の大きさをもった面だけをポリゴン化する。面を膨大な数の点の集合である3次元地図から直接検出することは困難であるため、一つ一つのセンサーデータから面を抽出し、抽出した面の重ね合わせを行うことで、3次元地図のポリゴン化を行う。

3.2.1 面の検出とテクスチャの作成

面の検出は3.1.1項の床の検出と同様のアルゴリズムで行う。ただし、ここでは面の向きを指定しないので手順2は行わない。また、複数の面を検出するため点群から検出された面に含まれる点を取り除き、面が検出されなくなるまで繰り返し面の検出を行う。実際に検出した様子を図3.8に示す。図では床と壁と扉の3つの面が検出されている。

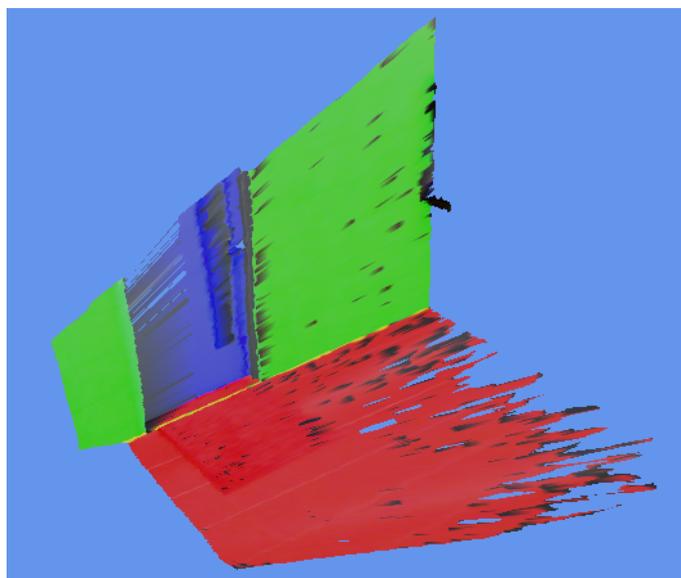


図 3.8: 面の検出

次に、ポリゴンを描画するため、面のテクスチャを生成し描画範囲を指定する手順について説明する。図 3.9 は図 3.8 の床のテクスチャを生成する様子を示している。まず、面に含まれる点を平面上にマッピングする (①)。このままでは点が離散的であり描画範囲を指定することができないので、モルフォロジーフィルタ [18] を用いて画像を平滑化する (②)。画像の輪郭線を抽出し描画する必要がある領域を求める (③)。赤色の線が輪郭線であり、輪郭線の内側を描画する必要がある。ポリゴンを描画するには描画範囲を 3 角形に分割する必要があるが、凹凸のある輪郭線を 3 角形に分割するのは難しい、そこで輪郭線の凸包を求め、この凸包を一つの面として描画する (④)。この時、輪郭線の内側の面積が小さいものはノイズとして無視する。図 3.10 の左側の図はセンサーデータから復元した 3 次元形状モデルであり、右側の図は検出した面をポリゴン表示したものを示す。画像の平滑化を行っているため画像が粗くなってしまっているが、検出した面のテクスチャが作られていることが分かる。本研究では、点群を面にマッピングしたものをテクスチャとして利用したため、画像の平滑化を行う必要があったが、カメラ画像を用いて 3D ポリゴンへのテクスチャマッピングを行う手法が研究されており [21]、そのような手法を取り入れることで、画像の平滑化を行わず、より詳細なテクスチャを生成することが今後の課題である。

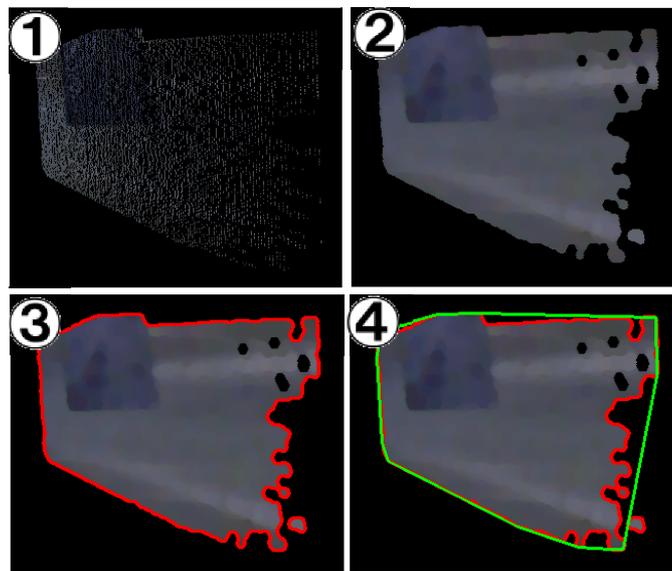


図 3.9: テクスチャの生成と描画範囲

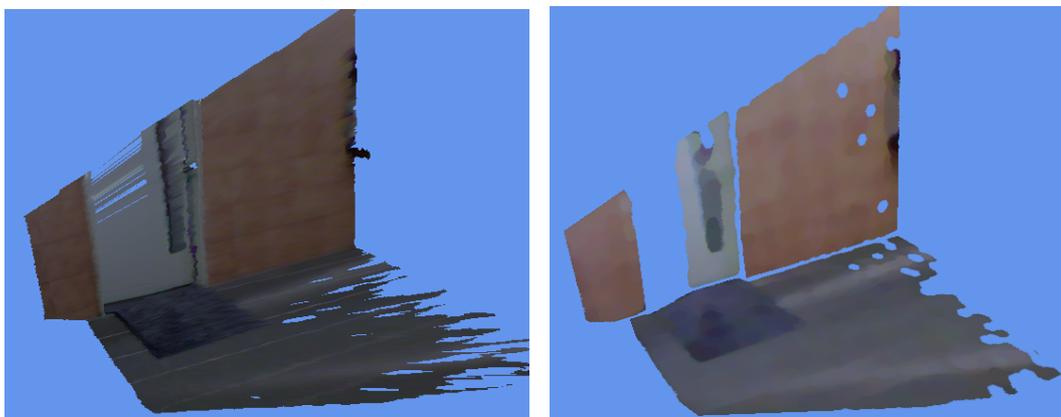


図 3.10: 3次元モデルとポリゴン表示の比較

3.2.2 面の重ね合わせ

前項で求めた面と一致する面を探し、一致する面が存在する場合は面の重ね合わせを行う。一致する面が存在しない場合はその面を新しい面として登録する。一致する面の検出は次のように行う。入力面とターゲットとなる面の向きのずれをそれぞれの面の法線ベクトルの内積から求め、ずれが閾値以下のものを検出する。

さらに、入力面の重心とターゲットとなる面の距離を求め、この距離が一定値以下で最も短くなる面を一致する面として検出する。面の重ね合わせは二つの面から新しい面を作ることで行う。入力面とターゲット面の法線ベクトルをそれぞれの面の面積で重み付けして平均を求めたものを重ね合わせてできる面の法線ベクトルとする。面のテクスチャは前項のテクスチャの生成方法と同様に入力面とターゲットとなる面それぞれに含まれる3次元上の点を新しい面に垂直に降ろした位置に点をマッピングすることでテクスチャを生成する。ただし、全ての点に対してマッピングを行うのは非常に時間がかかるため、入力面とターゲットとなる面でそれぞれ基準となる3点のみマッピングを行い、残りの点はテクスチャ上の2次元アフィン変換を求めることで新しいテクスチャを生成する。テクスチャの重ね合わせの様子を図3.11に示す。

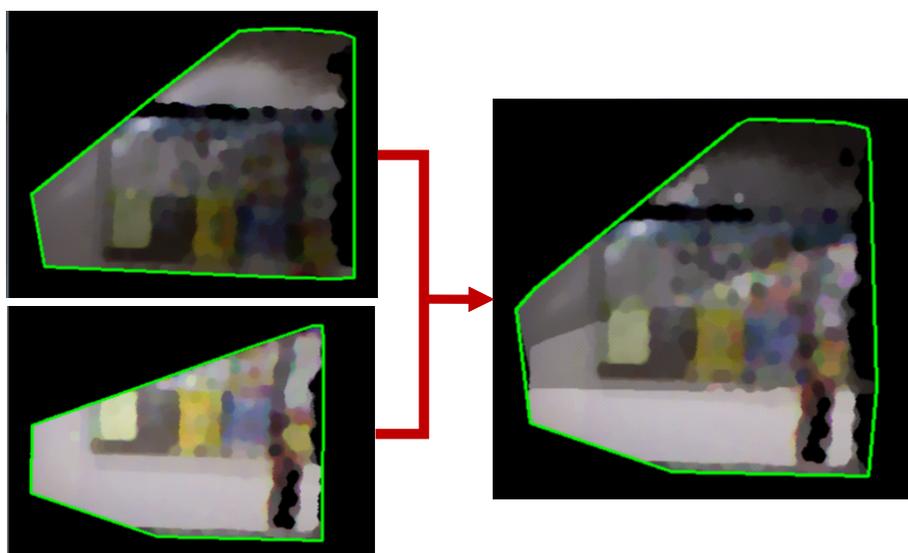


図 3.11: 面の重ね合わせ

図3.5の点群から成る3次元地図をポリゴン化した地図を図3.12に示す。点群から成る3次元地図は約1500万点の頂点データから構成されていたが、ポリゴン化した3次元地図では面の枚数が34枚、頂点数は767点とかなり単純化することができた。本研究では、問題を単純化するため、ポリゴンを表示する際に凸包の描画を行ったが、ポリゴンを描画する順番によっては透過処理が適切に行われず裏側にある面が見えなくなってしまうという問題が発生してしまった。そのため凸包を描画するのではなく、輪郭線の内側だけを描画することが今後の課題である。また、データが抜けている部分はデータの取り直しを行う必要があるが、そのような場所を人が探すのは非常に手間がかかる。そこで、明らかにデータが抜けている部分を自動的に補完、検出するという仕組みも今後の課題である。

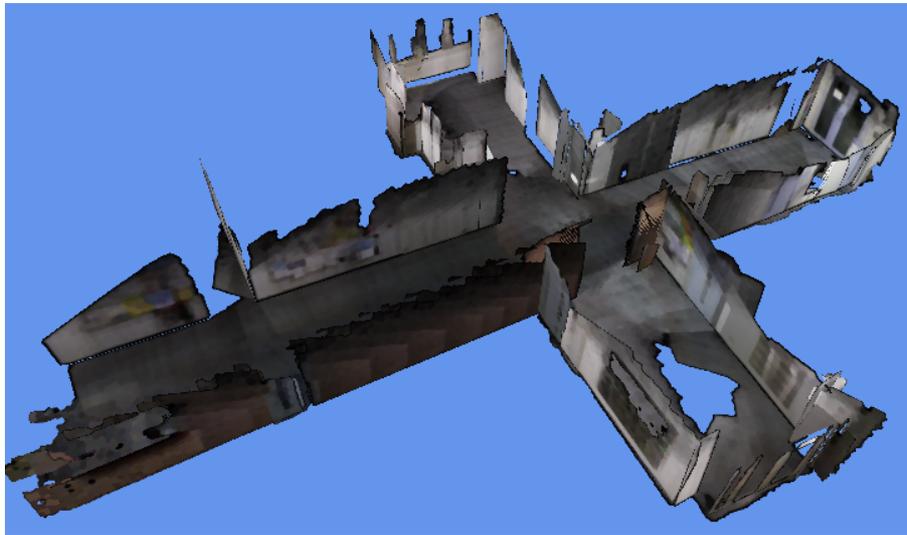


図 3.12: ポリゴン化した地図

3.3 経路生成

3次元地図生成のために必要なデータを自動的に収集するため、SUVが走行するデータ収集経路を事前に生成する必要がある。3.1節で説明したように3次元地図生成では画像特徴点を利用して3次元形状モデルの重ね合わせを行っている。そのため、収集するデータには画像特徴点が多く含まれることが理想的である。屋内において画像特徴点が発見されやすい領域は掲示物や扉などが含まれる壁沿いである。逆に床は特徴的な部分が少なく画像特徴点発見されにくい。図 3.13 に、取得したRGB画像の特徴点が発見したものを示す。この図から、壁沿いに特徴点が集まり、床には特徴点が少ないことが分かる。特徴点を多く含むようなデータを収集するには、壁だけのデータを取るのが好ましいように思えるが、3次元地図を生成するためには、特徴点を含まない床のデータも収集する必要がある。そのため、3次元モデルをうまく重ね合わせるためには、特徴点が多く含まれる壁と特徴点が少ない床のデータを同時に取得することができる経路を生成する工夫が必要となる。そこで、壁と床のデータを同時に取得するためにSUVが壁から一定距離の位置を壁に沿って走行する経路を生成する。壁からの距離は経験的におよそ60cmとする。ただし、道幅が120cm以下の狭い通路では通路の中央を走行する。

3次元地図を生成するためには環境中をくまなく探索する必要があるが、壁沿いを走行するだけでは環境中のデータをくまなく収集することはできない。そこで、道幅が広く壁沿いからでは道の中央付近のデータが収集できない通路や開けた場所では、データを収集するために通路の中央付近も走行することでデータを収集

するようにする。また、生成した経路を一周するだけでは物陰になってしまいデータが取れない領域ができてしまう可能性があるため、経路を一周した後、経路を逆向きにもう一周することでくまなくデータを収集する。

SUVは2.5節で説明したように、ノードのグラフ構造を利用して自律走行することが可能であるので、まずは経路生成の基準となるノードを生成する。これについては3.3.1項および3.3.2項で詳しく説明する。3.3.3項では、このノードを利用した経路生成のアルゴリズムについて説明する。3次元地図を作成したくない場所や、特定の場所だけの3次元地図を作成したい場合が考えられるので、3次元地図を生成する範囲を限定できる機能を実現した。これについて3.3.4項で説明する。3.3.5項では、生成した経路の評価を行う。

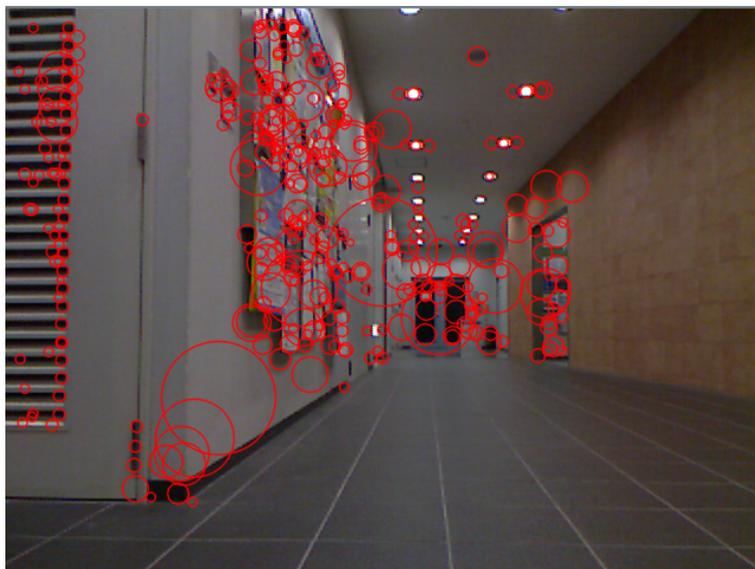


図 3.13: 画像特徴点を検出した様子

3.3.1 ノードの生成

SUVは、ノードとノードの間のつながりによるグラフ構造から経路を生成し、自律走行することが可能である。そこで、ここではデータ収集経路の基準となるノードの生成方法について説明する。ここでは、ノードの生成条件の違いから、ノードを壁から一定距離のノード、狭い通路の中央のノード、狭い通路と広い通路の交差点のノードの3タイプに分け、それぞれをノードA,B,Cとする。

1. 走行可能領域の壁からの距離を計算する
最近傍の計算は k-d 木を使って行う。環境地図の黒色の点つまり障害物有りの点から k-d 木を作成し、走行可能領域つまり白色の点からの最近傍とその距離を計算する。この時、手順 3 で利用するため近傍とその距離は 10 近傍まで計算を行う。
2. 壁からの距離で走行可能領域を分割する
SUV は壁から 60cm の位置を走行し、広い通路および開けた場所ではそこからさらに 60cm 間隔ごとに走行しデータの収集を行う。そのためのノード生成の手がかりとするため、走行可能領域を壁からの距離が 60cm になる位置で分割する。図 3.14 の①は分割した領域を色で表している。壁から近いほど濃く、遠くなるにつれて薄い青になる。
3. 道幅が狭い通路の中央を探す
道幅が狭い通路では通路の中央を SUV が走行するため、通路の中央にノード生成の手がかりを作る必要がある。そこで、最近傍点と手順 1 で計算した 10 近傍をそれぞれ比較し、最近傍点と k 近傍点の位置が大きく離れていた場合その走行可能領域は道の中央にあると判断する。ただし、最近傍までの距離と k 近傍までの距離の差が環境地図の格子間隔の 2 倍以上だった場合はノイズとして除去する。図 3.14 の②の赤色の線が道幅の狭い通路の中央を表している。
4. グリッド線を引きノード A およびノード B を生成する
一定間隔でグリッド線を引き、手順 2 で分割した領域の境界線および、手順 3 で検出した通路の中央線とグリッド線が交差する点にノードを生成する。図 3.14 の③は 60cm 間隔でグリッド線を引いたものである。図 3.14 の④の赤色の点がノード A、緑色の点がノード B である。このとき、一定距離内にノードが複数生成される場合、ノード A とノード B をそれぞれ一つ生成する。ただし、ノード A においてノード生成位置の境界線の向きが違う場合はノードを複数生成する。図 3.15 の赤色の円で囲った部分は境界線の向きが同じであるためノードを一つだけ生成し、黄色の円で囲った部分は境界線の向きが違うため両方のノードを生成している。
5. 狭い通路と別の通路の交差点にノード C を生成する
手順 2 の境界線と手順 3 の中央線が交差する点にノード C を生成する。ノード C を生成するのはノードの接続時にノード A とノード B の間に壁が入ってしまいノードの接続ができなくなることを防ぐためである。図 3.14 の④の黄色の点がノード C を表している。

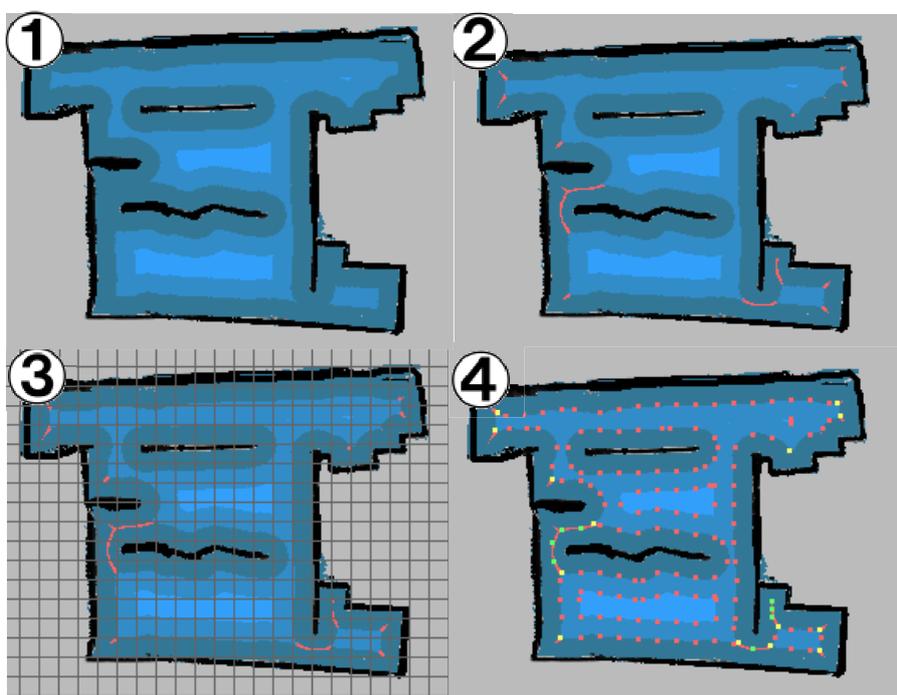


図 3.14: ノード生成アルゴリズム

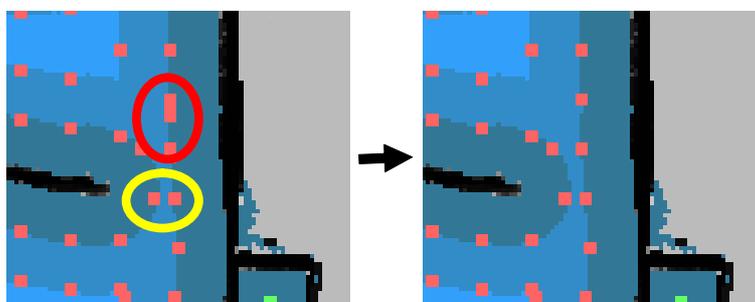


図 3.15: 密集したノードの除去

3.3.2 ノード間の接続

前項で述べた手法で生成したノード同士を接続しグラフ構造を生成する。壁沿いを走行する経路を生成するため、ノード間の関係を隣接するノードと到達可能なノードの2種類定義する。隣接するノードはノードAおよびノードCで同じ境界線に生成された隣り合ったノード同士、もしくはノードBおよびノードCで隣り合ったノード同士の関係である。到達可能なノードは隣接したノード以外で近くにあり間に障害物のないノード同士の関係である。近くにあるノードは基準のノードの周りを矩形で囲み矩形の内側にあるノードを探すことで求める。隣接するノードは基準のノードから境界線をたどっていくことで見つける。ノード間の障害物の有無は、ノードとノードの間にある程度の太さを持った線を引き、線が障害物と重なるかどうかで判定する。線の太さはSUVの直径が約30cmであるので位置推定の誤差を考慮しておよそ40cmに相当する太さとする。これをすべてのノードで実行することでノード同士を接続しグラフ構造を生成する。実際にグラフ構造を生成したものを図3.16に示す。赤色の線が隣接するノードであり、青色の線が到達可能なノードを表している。

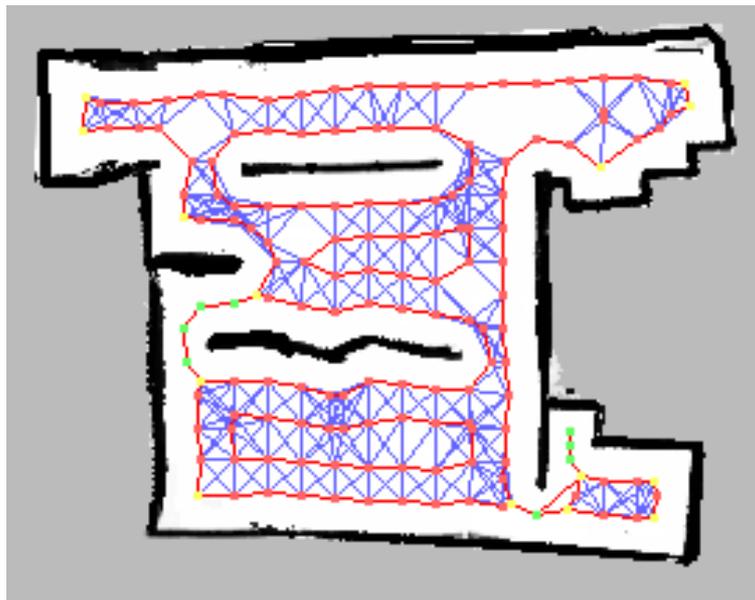


図 3.16: ノード間のグラフ構造

3.3.3 経路生成のアルゴリズム

生成したグラフ構造を利用して経路の生成を行う。基本的には壁沿いを一周し、出発地点に戻ってくる経路を生成する。以下のようなアルゴリズムで経路を生成した。

1. スタート地点を設定する
スタート地点はSUVの現在地から最も近いノードAまたはノードCを選択する。
2. 指定されたノードから隣接するノードをたどって1周する経路Rを生成する
隣接するノードをたどって一周する経路Rを生成する。この時通過したノード $R_i (0 \leq i \leq n)$ に到達可能ノード N_m が存在し、ノード N_m が未探索の場合、 $R_i \rightarrow N_m$ を探索候補に追加していく。また、 $N_j = R_i$ となる探索候補が存在する場合それを削除する。
3. 未探索のノードを通る経路R'を生成し経路Rに挿入する
探索候補のうち、 $R_i \rightarrow N_m$ の距離が最も短くなるものを選び、ノード N_m をスタート地点として手順2と同様の操作で経路R'を生成する。経路Rの $i+1$ 番目に経路R'を挿入する。このとき、滑らかな経路を生成するために $R_{(i-1)} \rightarrow R'_1$ が到達可能であれば R'_0 を削除する。図3.17の左側の図の緑色の経路が経路R、赤色の経路が経路R'、青色の線が $R_i \rightarrow N_m$ を示す。右側の図は経路Rに経路R'を挿入したものを示している。

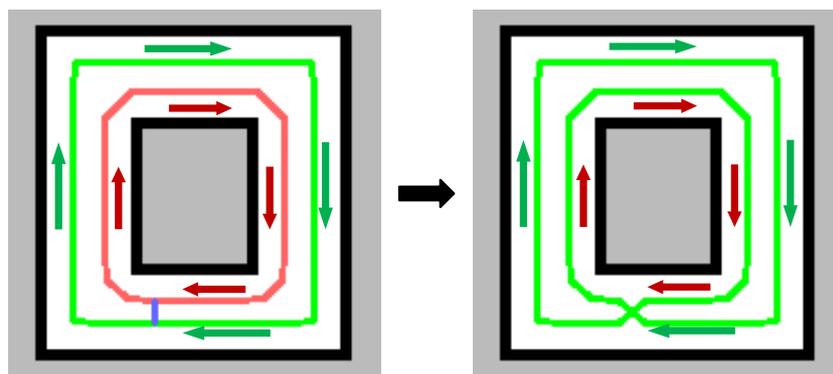


図 3.17: 未探索ノードを通る経路

4. 探索候補がなくなるまで手順3を繰り返し実行する
5. 経路上のすべてのノードCに対して手順6を実行する

6. 狭い通路を通る経路 R'' を生成し経路 R に挿入する

ノード C からノード B へ進む経路 R'' を生成する。経路 R'' の経路 R への挿入方法は経路 R'' の終端であるノード R''_n によって3つの場合に分かれる。それぞれの場合について以下で説明する。図 3.18 にそれぞれの場合で経路が生成される様子を示す。赤色の経路がそれぞれ経路 R'' を示す。

(a) R''_n がノード B の場合

経路 R'' を反転し経路の0番目と n 番目を削除したものを経路 R'' の末尾に追加し、経路 R の i 番目に経路 R'' を挿入する。

(b) R''_n が未探索のノード C の場合

R''_n をスタート地点として手順 2-4 を実行し経路 R''' を生成する。図 3.18 の b の青色の経路が経路 R''' を示す。経路 R'' の0番目と n 番目を削除し反転したものを経路 R''' の末尾に追加し、経路 R'' の n 番目を削除したものを経路 R''' の先頭に挿入する。経路 R の i 番目に経路 R''' を挿入する。

(c) R''_n が探索済みのノード C の場合

経路 R_{ij} を反転し、経路 R'' の0番目と n 番目を削除したものを経路 R の j 番目に挿入し、経路 R の i 番目に経路 R'' を挿入する。ただし、経路 R の i 番目から j 番目までの経路を経路 R_{ij} とし、 $R_j = R''_n$ とする。図 3.18 の c の青色の経路は経路 R_{ij} を示す。

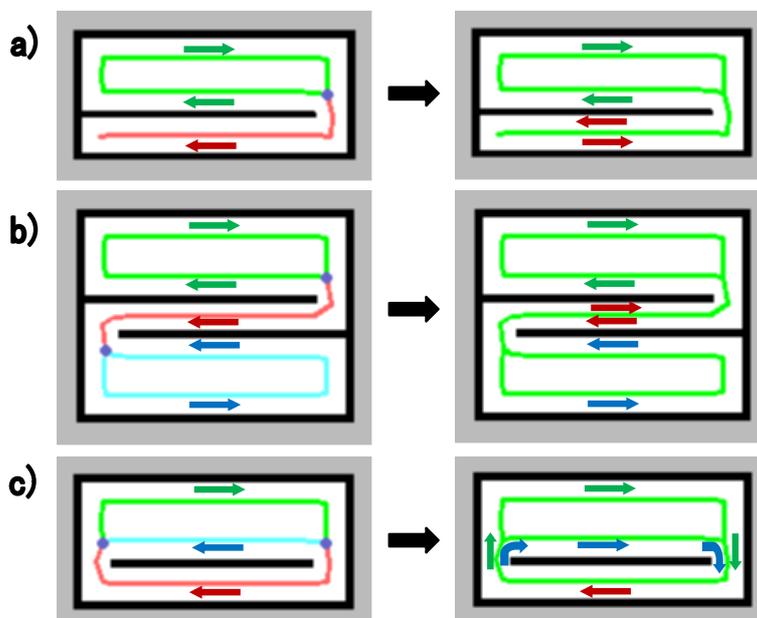


図 3.18: 狭い通路を通る経路

仮想的な壁や障害物と見なし経路生成に利用する。3次元地図を生成したい範囲を直線で囲むことで、直線を仮想的な壁と見なし、囲まれた範囲のデータを収集する経路を生成する。環境地図上に正確な形状が現れない机やいすが存在する位置に、矩形や楕円形の図を配置することで、経路生成時には配置された図形を障害物と見なし、障害物情報を考慮した経路を生成する。実際に環境地図に情報を付与したものを図 3.20 に示す。これらの情報は、経路生成時のみ利用され、位置推定には影響しない。

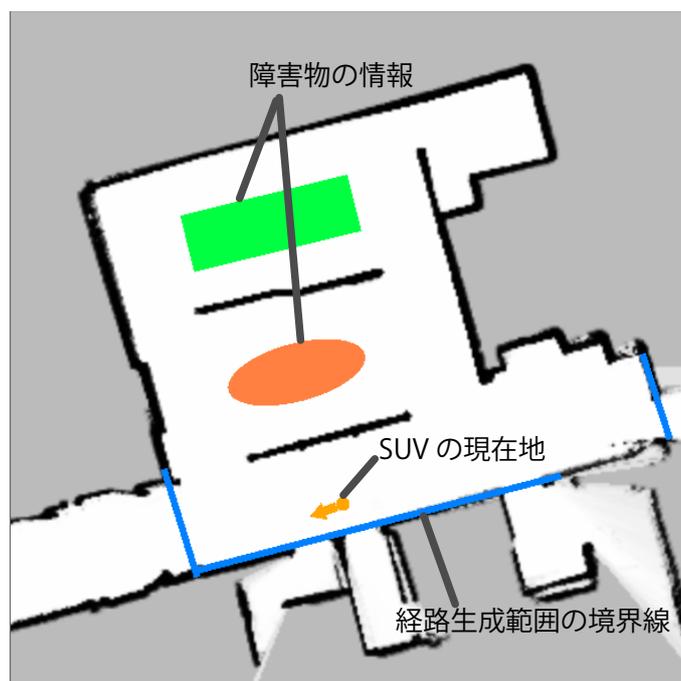


図 3.20: 経路生成範囲の指定と障害物情報の付与

3.3.5 生成した経路の評価

生成した経路が実際にデータを収集できるか検証を行う。3次元地図の生成のためには、3次元的なデータの収集を行う必要があるため、環境中をくまなく探索できているかという点と、データが十分に収集できているかという点から経路の網羅性を判断しなければならない。しかし、3次元的なデータが十分に取れているかをシミュレータ上で検証することは難しく、実際にSUVが走行する際には、位置推定の誤差やSUVの制御の問題から、生成した経路を正確にトレースすることは難しい。そのため、シミュレータ上で網羅性が保証されても、実環境でも同様に網羅性が保証されるとは考えづらい。そこで、本研究では、問題を単純化し、2次元

的な網羅性を考える。少なくとも一度はデータの収集を行っている領域はデータが十分に収集できているものとして、データが収集できると思われる領域を2次元の環境地図上にマッピングし網羅性の検証を行う。

経路の網羅性の検証を行うため、SUVが走行する経路を基に、RGB-Dカメラのセンシング領域を計算し、環境地図を塗りつぶす機能を実現した。実際にセンシング領域を塗りつぶした様子を図3.21に示す。緑色の線で示す経路を走行した際にセンシングできる領域が赤色で塗りつぶされている。赤色で塗りつぶされた領域が、データを十分収集することができたと思われる領域である。

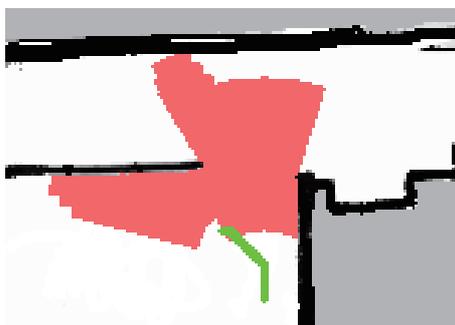


図 3.21: センシング領域の塗りつぶし

図3.22の左側に示す経路を生成し、網羅性の検証を行った。SUVが経路を走行した際に、センシング可能だと思われる領域を塗りつぶしたものが図3.22の右側の図である。この図から、ほぼすべての領域のデータを収集することができていることが分かる。しかし、地図の左下に一部データを収集できていない領域が存在する。これは、建物の構造が入り組んでいるため、その領域のデータを収集する経路が生成されなかったことが原因だと思われる。

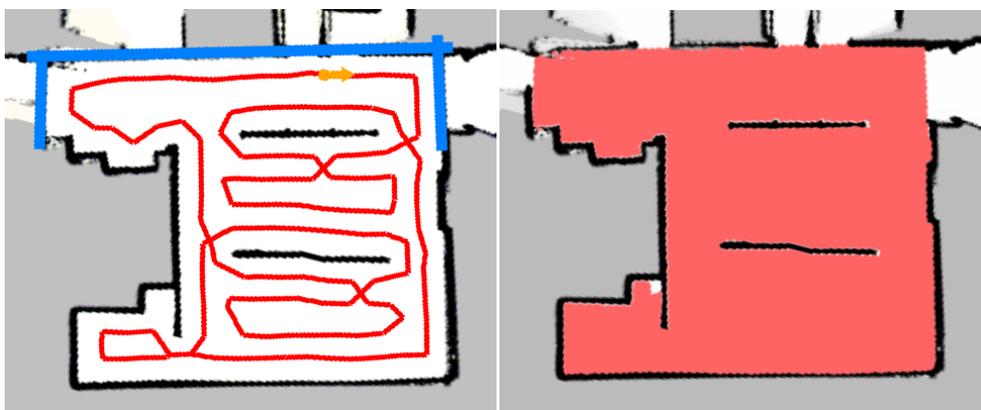


図 3.22: 生成した経路と網羅性の検証

次に、図 3.22 の経路を実際に SUV に走行させ、収集したデータから生成した 3 次元地図を図 3.23 に示す。センシング領域を求めた際に、データを取得することができなかった領域では、3 次元地図が大きく欠けてしまっていることが分かる。これは上記の理由に加えて、RGB-D カメラは距離が近すぎるとデータが取得できないことと、生成した経路と実際に走行した経路のずれが原因でデータが収集できなかったものと思われる。また、3 次元地図の上部でも床が欠けている部分が複数見られる。これは、センサーデータの重ね合わせの失敗や、データが十分に取れていなかったことが原因だと考えられる。これについては、センサーデータの重ね合わせの収束条件の変更や、サンプリング点の選択方法の変更によって 3 次元地図の精度を改善することや、欠けている領域を自動的に検出し補完する機能を実現することで解決することができると考えられる。以上から、本研究で生成した経路は、入り組んだ場所では生成する経路を工夫する必要があるが、それ以外の場所では十分に環境を網羅できていると言える。



図 3.23: 生成された 3 次元地図

センシング領域を塗りつぶす機能は、厳密にセンシング領域を計算するものではないが、データを収集できない領域をある程度検出することが可能であることが確認できた。そのため、実際にデータを収集する前に、この機能を利用して経路の妥当性を評価することが可能であると思われる。データを収集することができない領域が検出された場合には、データが収集できない領域を考慮して、経路の再計算を行うことで、より網羅性の高い経路を生成する事ができる。また、先にも述べたように、この機能はセンシング領域を厳密に計算するものではないため、データ収集後にデータ不足が判明する事も考えられる。そのような場合には、

データが不足した領域を検出し、その領域のデータのみを収集する経路を生成し、データの再収集を行う仕組みが必要である。このような仕組みを実現する事も今後の課題の一つである。

第4章 展望

本研究では、SUV を、前章で述べた手法で生成した経路に沿って走行させることで、実空間のデータを自動的に収集し、収集した RGB-D カメラのデータを基に 3 次元地図の生成を行った。しかし、実際に 3 次元地図を利用したシステムの開発などの 3 次元地図の応用には着手できていない。そのため、今後は生成した 3 次元地図の利用法について考えていく必要がある。そこで、本章では 3 次元地図や SUV についての展望を述べる。

生成した 3 次元地図は、地図上に存在する施設の場所やその情報などの、地図に関する情報を持っておらず、地図としての利用価値がまだあまり高くない。そこで、3 次元地図に関連情報をアノテーションできる仕組みを提供することで、より利用価値の高い 3 次元地図を作ることができる。アノテーションとして付与する情報には、より高解像度なテクスチャ画像や利用できる施設の情報、ユーザーによるコメントなどの情報などが考えられる。また、それらの情報を 3 次元地図上に表示することでユーザーに分かりやすく情報を提示することができる。これについて 4.1 節で詳しく説明する。地図に付与された情報の検索などを可能にすることで疑似体験システムやナビゲーションシステムへの応用が考えられる。4.2 節では SUV を道案内ロボットとしたナビゲーションシステムや、3 次元地図を我々の研究室で開発している個人用知的移動体 AT (Attentive Townvehicle) [25, 22] のインタフェースとして利用する方法について説明する。アノテーション情報の提示方法には 3 次元地図上に表示するだけでなく、拡張現実感 (Augmented Reality、以下 AR) を用いてカメラ画像にアノテーション情報を重畳表示するなどの方法がある。また、死角になる領域の 3 次元地図を重畳表示することで疑似的に壁を透過するといった利用法も考えられる。これについて 4.3 節で詳しく説明する。また本研究では、SUV を用いてデータを自動的に収集する機能を実現した。4.4 節ではその機能を応用し、建物内を常に複数の SUV が走り回っているような状況で、3 次元地図を定期的に更新する仕組みや、SUV が施設の混雑具合や障害物の情報などのデータを収集し、それらの情報を 3 次元地図上に反映することでリアルタイム性の高い情報を利用する仕組みについて説明する。

4.1 様々な情報の提示

生成した3次元地図は地図上に存在する施設などに関する情報を持たず、テクスチャも不鮮明で分かりにくいいため、生成した3次元地図そのままではナビゲーションシステムなどの応用には適していない。そこで、テクスチャ画像の貼り付けやアノテーションを行うことで地図に情報を付与する必要がある。まずは3次元地図へのテクスチャ画像の貼り付けについて説明する。鮮明なテクスチャ画像を貼り付けることは、扉や階段の位置、ショッピングモールなどでは店舗の外観など、地図にとって非常に重要であるだけでなく、人が現在地を認識するためにも役立つと考えられる。画像データは、SUVに高解像度なカメラを搭載し、3次元地図生成用のデータを収集する際に同時に収集する。そうすることで撮影した時のSUVの位置情報からある程度自動的にテクスチャを作成できると考えられる。また、テクスチャ画像の貼り付けは内装のシミュレーションなどにも利用できるため、自動で行うだけでなくユーザーが任意の画像を貼り付ける仕組みも必要である。図4.1に手動でテクスチャ画像を貼り付けたものを示す。



図 4.1: テクスチャ画像が貼り付けられた3次元地図

次に3次元地図へのアノテーションについて説明する。地図に含まれる施設や店舗などの位置や詳細情報をアノテーションする。地図上の施設などの情報を自動的に地図にアノテーションすることは困難であるため手動で簡単にアノテーションできる仕組みを提供する必要がある。アノテーション情報は地図上にはアイコンとして表示され選択することで詳細情報を閲覧することができる。また、地図に含まれる施設の情報だけでなく、地図の利用者が店の評価やコメントなどを自由にアノテーションすることで地図がより高度になり利用価値が高くなると考えら

れる。これらの情報を地図利用者が携帯端末などで閲覧できるようにすることで環境の認識や情報の検索が容易になり、3次元地図を疑似体験システムやナビゲーションシステムに利用できるようになると考えられる。図4.2に情報を付与した3次元地図を示す。エレベーターやトイレなどの一般的な施設はアイコンで表示し、掲示板の位置や部屋番号などのラベル情報を付与し、3次元地図上にマーカーを表示する。マーカーを選択することで選択したマーカーに関する詳細情報がポップアップ表示される。



図 4.2: アノテーションが付与された3次元地図

4.2 ナビゲーションへの利用

地図に様々な情報を付与したことで3次元地図をナビゲーションに利用することができるようになる。ここではSUVを道案内ロボットとしたナビゲーションシステムについて説明する。図4.2に示したような、ラベル情報やアイコン、詳細情報などのアノテーション情報が付与された地図を提示することで、ユーザーは建物に関する様々な情報を入手することができ、これらの情報を基に目的地を決定することで、ユーザーが本当に行きたい場所を目的地として決定することが容易になる。また、3次元地図上で目的地を決定することにより、3次元地図と対応のとれた環境地図上でも目的地を把握することができ、環境地図を持つSUVが、現在地から目的地までの経路を探索して自動走行を行うことができ、利用者はSUV

の後ろをついていくことで目的地まで到達することができる。また、この仕組みは、ナビゲーションとしての利用法以外にも荷物運びロボットとしての利用も考えられる。SUVは障害物や人の検出を行うことが可能であるため、経路の途中で通路が障害物によりふさがれ通行不能であった場合には、通行不能な通路を迂回するルートへと経路を変更することができ、移動障害物を検知した場合には回避行動をとるなどの動作が可能である。ただし、SUVを道案内ロボットとする場合にはRGB-Dカメラを後方に向けるなどの工夫をして利用者がついて来ているかを確認する仕組みを考える必要があると思われる。図4.3に3次元地図を利用したナビゲーションシステムの画面例を示す。



図 4.3: 3次元地図を利用したナビゲーションシステムの画面例

次に、自動走行可能な乗り物のインターフェースとしての3次元地図について説明する。我々の研究室ではAT (Attentive Townvehicle) と呼ばれる自動走行可能な個人用知的移動体の研究開発をしている。ATの外観を図4.4に示す。ATは、搭乗者である人間や自身を取り巻く環境に適応し、周囲の環境や他の移動体との通信によって協調的に動作することが可能な個人用の乗り物である。ATの位置情報に関するインターフェースに3次元地図を利用することで様々な関連情報を搭乗者に提示できるようになるほか、実世界のATの位置と連動して3次元地図上の視点が移動することで、搭乗者が自身の位置を容易に把握することができ、目的地を容易に決定することができる。また、ATにはSUVと連携走行を行う機能がある(図4.5を参照)。これは、ATが自動走行をする際に、SUVがATの前方を走行しATの死角になる領域をSUVがセンシングするという、SUVをATの拡張センサーとして利用する仕組みである[22]。この機能を利用し、SUVが撮影したRGB-DカメラのデータをリアルタイムにATのインターフェースである3次元地図上に反映

し、死角領域の情報を搭乗者に知らせることが可能になる。



図 4.4: 個人用知的移動体 AT



図 4.5: AT と SUV の連携走行

4.3 ARへの応用

3次元地図やそれに付与されたアノテーションを位置情報を基に実世界に重畳表示する仕組みについて説明する。カメラの位置情報を基にカメラ画像にアノテーション情報や3次元地図を重畳表示することでARを実現することができる。カメラの位置情報はカメラ画像と3次元地図から生成した画像の重ね合わせやオプティカルフローを計算することで推定することができると考えられる。一般的にARはオブジェクトを表示したい位置にマーカーを設置する必要があり、このマーカーの設置の手間がARの問題点の一つであった。しかし、我々の仕組みは位置推定を行うことができるので、マーカーを設置する必要がなく、環境側に特別なインフラを整備する必要がないことが利点の一つである。利用者に表示する情報は、3次元地図へのアノテーション情報だけでなく、死角となる領域の3次元地図をカメラ画像に重畳表示することで、壁の裏側に何があるかなどを見ることができる。図4.6に壁の裏側にある机とイスをカメラ画像に重畳表示した画面例を示す。このように死角領域の情報をユーザーに提示することで、ユーザーが建物の構造を把握するのに役立つと考えられる。また、SUVによるリアルタイムな情報を重畳表示することで、死角領域に人がいないかなどを確認することができ、より安全に移動することができる。この仕組みはヘッドマウントディスプレイ (Head-Mounted Display、以下HMD) を用いて利用する場合に特に効果的であると考えられる。最近ではシースルー型のHMDも登場し、将来的には自分の視界に直接デジタル情報が重畳表示されているように見えるようになると思われる。



図 4.6: 死角領域の AR 表示

4.4 複数 SUV の連携

これまで説明してきたように、SUV には様々な役割がある。将来的には図 4.7 に示すように、環境内に複数の SUV が存在し、それぞれの SUV がそれぞれの役割を持って行動していると考えられる。そのような環境では、複数の SUV 同士が連携することは、データ収集の効率化や障害物情報の共有などにおいて非常に重要な課題である。SUV 同士が連携するためには全ての SUV をサーバーで管理する必要がある。サーバーで管理することにより SUV 同士が情報を共有できるようになり、より質の高いサービスを提供できると考えられる。具体的には、施設の混雑状況を共有し 3 次元地図上に反映することでサービス利用者が目的地の混雑状況を把握できたり、障害物情報を共有することで走行不能な通路が検出された場合に全ての SUV がその場所まで行かなくても事前に経路を変更することができる。また、ある SUV からは死角になってしまう領域を別の SUV がカバーし、その情報を 3 次元地図上に反映することで、より安全なナビゲーションシステムを実現することができると考えられる。また、データ収集時には、複数の SUV でデータの収集を行う経路を生成し、複数の SUV が共同でデータを収集することによって、より効率的にデータの収集を行うことができる。

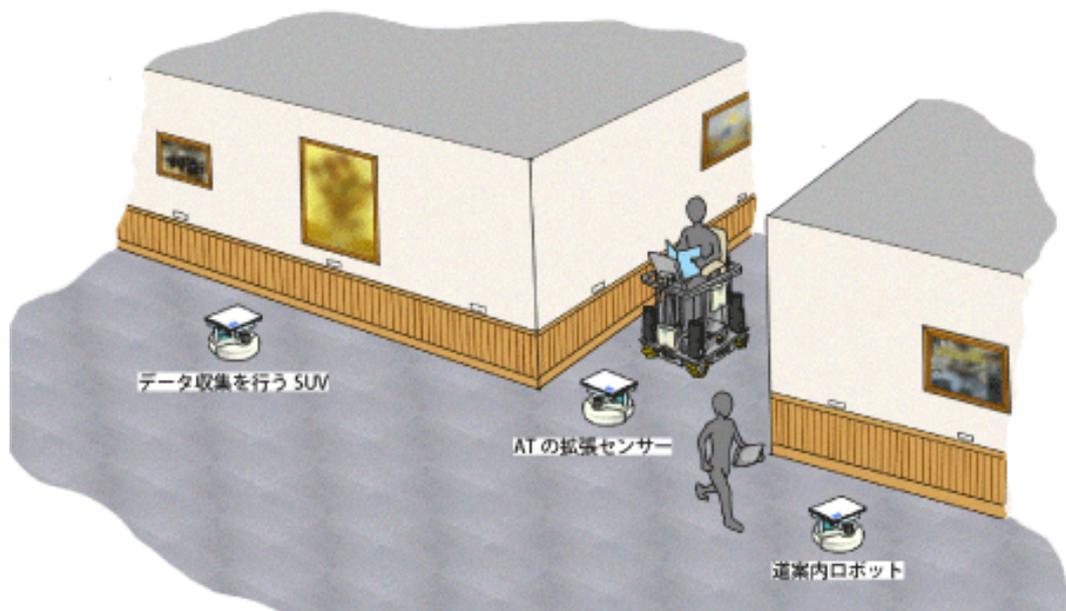


図 4.7: 様々な役割を持つ SUV

第5章 関連研究

本研究では、小型無人移動体を用いてデータの収集を行い、収集したデータの重ね合わせによって3次元地図を自動的に生成する手法を実現した。関連研究として、5.1節ではRGB-Dカメラを用いた3次元地図の生成に関する研究、5.2節ではレーザーレンジセンサーを搭載した移動ロボットを用いて3次元地図の生成を行う研究、5.3節では3次元地図を用いて情報提示を行う研究、5.4節では格子点配置を用いて自律移動ロボットの走行経路を生成する研究について紹介し、それぞれについて本研究との違いを述べる。

5.1 RGB-D Mapping: Using Depth Cameras for Dense 3D Modeling of Indoor Environments

Henryらは、屋内環境の精密な3次元地図の生成にRGB-Dカメラがどのように利用できるか調査を行い、RGB-Dカメラを用いて3次元地図を生成する手法を提案している[12]。RGB-Dカメラによって取得できる3次元の点群は、フレーム間の重ね合わせに適しており、3次元構造の再構成をすることができるが、画像に含まれる様々な情報を考慮していない。一方、RGB画像は様々な視覚情報を持ち、Simultaneous Localization and Mapping (SLAM)におけるループの検出に適している。しかし、カメラ画像のみから3次元構造を再構成することは困難である。そこで、提案手法では、フレーム間の画像特徴点の対応点から初期位置を決定し、3次元の点群と画像特徴点にICPアルゴリズムを適用し重ね合わせを行う。これにより、ICP (Iterative Closest Point) アルゴリズムが局所解に収束してしまう可能性を抑えることができる。また、画像特徴点からループの検出を行い、TORO (Tree-based Network Optimizer)[8]を用いてループ解決を行うことで3次元地図の生成を行う。

本研究では、SUVを用いてデータの収集を行うため、環境地図とデータ収集時のSUVの位置情報を知ることができるという点がこの研究と大きく異なる点である。環境地図を事前に持っているため、環境地図と対応の取れた3次元地図を生成することが可能である。本研究では、この研究の提案手法であるRGBD-ICPアルゴリズムを、環境地図との対応も利用するように拡張した重ね合わせ手法を提案した。環境地図を利用して重ね合わせを行うため、この研究のようにループ解

決を行う必要がないことも本提案手法の利点である。また、この研究では重ね合わせには連続したデータが必要になるが、本研究ではSUVの位置情報を利用することで、連続的なデータでなくても扱うことができる。これにより、データを一度にすべて収集する必要がなくなるほか、地図の一部分だけを更新することが可能になる。

5.2 レーザレンジファインダ搭載移動ロボットによる動的環境の3次元地図生成

中本らは、レーザレンジファインダ (Laser Range Finder、以下 LRF) を搭載した移動ロボットを用いて動的環境の2次元地図および静的環境の3次元地図の生成を行っている [24]。具体的には LRF を移動ロボットに2台搭載し、周囲の環境における2次元情報と3次元情報を取得する。LRF は、1台は水平に、もう1台は傾けて搭載する。水平に搭載した LRF は SUV における LRF (レーザレンジセンサー) と同様の役割を果たし、傾けて搭載された LRF は RGB-D カメラに相当する。この研究では、2つ目の LRF の設置する角度を下向きと上向きの2通りに変えて、2回同じ場所を計測することで、環境全体の3次元地図を生成する。LRF は平面のセンシングを行うため、3次元的な重ね合わせを行うことができず、位置推定の結果とセンサーの設置角度から、センサーデータの3次元的な位置を決定する。そのため、正確な位置推定が難しい環境では、3次元地図を生成する事が困難であると思われる。本研究では、RGB-D カメラを用いることで、センサーデータの重ね合わせを可能とし、環境地図と対応付けを行うため、位置推定に多少の誤差が含まれる環境でも問題なく3次元地図を生成する事ができる。しかし、本研究では、RGB-D カメラの傾きを変えると、角度によっては床を検出できなくなるため、RGB-D カメラの向きが固定されており、センサーの届かない位置、特に上方はデータを取得することができないという問題がある。そのため、センサーを傾けたり、向きを変えたりする仕組みは、今後の課題の一つである。

5.3 施設におけるバリアフリー情報の3次元地図としての視覚化

和泉らは、バリアフリー情報を3次元地図として提供する研究を行っている [23]。具体的には、施設の構造を表現する基本地図データとバリア情報、施設の3次元モデルから成る3次元バリアフリーマップシステム (3DBFMS) の構築を行った。基本地図データとは施設内の区画や部屋などが地図要素として定義されたものを指す。この研究では、「詳細な3次元モデルは、バリア情報を正確に表現できるが、

ユーザーに必要以上の情報を与えることになり、バリア情報の取得を妨げる」として、3次元モデルは建物構造の把握に利用し、アイコンやテキストによりバリア情報の提供を行っている。これらの情報をブラウザ上で閲覧することができ、施設全体の構造やバリア情報を把握するための俯瞰表示や、目的地までのウォークスルーシミュレーションを行うことができる。この研究では、3次元モデルの作成に多くのコストがかかることを問題点として挙げているが、本研究で提案した3次元地図生成手法を利用することで、このようなシステムに利用できる簡略化された3次元地図を容易に作成できる。また、本研究では地図生成時に実環境のデータ収集を行っているため、収集したデータの分析を行うことで、ある程度自動的にバリア情報を地図上に反映することができると考えられる。また、施設内をSUVが動き回ることによって、障害物情報などのリアルタイム性の高い情報が利用できるようになる。このような3次元地図を利用したシステムを開発することは、本研究の今後の課題として重要なテーマである。

5.4 格子点配置を用いた自律移動ロボットによる環境掃引経路計画

ロボットのセンシング範囲のような、ある一定の領域により作業環境全体を覆い尽くす作業は掃引作業 (Continuous Area Sweeping) と呼ばれる。計画行動型の基本手法として Contour Parallel (沿輪郭) 型と Direction Parallel (蛇行) 型が挙げられる。これらは、線分によって環境を掃引する手法であるが、点によって環境を被覆する手法として、深澤らは、格子点配置を用いて自律移動ロボットの環境掃引経路計画を行う手法の提案をしている [27]。具体的には、掃引対象領域に周囲を観測すべき地点を配置し、それらの点群を結ぶ経路を求めることで掃引経路計画を実現している。この研究では、ロボットのセンシング範囲を、ロボットを中心とした円形の範囲として、観測点の配置を行っている。しかし、本研究では、センシング領域は円形ではなく、RGB-D カメラのセンシング領域であるため、観測点は位置だけでなく、ロボットの向きも考慮する必要がある。そのため、問題が非常に複雑になり、観測点を求めることが困難である。本研究では、センシング領域を考慮せず、主に壁沿いを走行することで環境中を探索する経路を生成し、実際にデータの収集を行った。しかし、センシング領域を考慮しなければ、環境中をくまなく探索する事は困難であるため、センシング領域を考慮した経路の生成も、本研究の今後の課題に含まれる。

また、この研究では、未知障害物に遭遇した場合のオンラインでの経路再計画手法についても提案している。具体的には、すでに掃引した領域と、新たに発見された障害物が占める領域を掃引領域から除き、再定義された掃引対象領域に観測点の再配置を行う。再配置には反応拡散方程式を利用して、すでにある観測点

を移動させることにより、掃引対象領域の変動が起きた境界付近の観測点のみが移動し、ほかの観測点は移動しない。そこで、移動しなかった点同士を結ぶ経路情報はそのまま利用することで、経路生成に要する計算量を低減する。この研究では、未知障害物は静止障害物であるという仮定の下、経路の再計画を行っているが、実際の環境では移動障害物の検出も行う必要がある。特に屋内において、移動障害物は人である可能性が高く、本研究では、RGB-Dカメラを用いて人の検出を行い、データ収集中に人を検出した場合にはデータの収集を一時中断する機能を実現した。

第6章 まとめと今後の課題

6.1 まとめ

本研究では、小型無人移動体を用いて自動的に実環境に関するデータの収集を行い、収集したデータから環境地図と対応の取れた3次元地図を生成する仕組みを実現した。

その仕組みは、筆者の所属する研究室で研究開発された小型無人移動体 SUV (Small Unmanned Vehicle) を利用する。SUV は、事前に環境内を走行することにより、SUV が環境を認識するために必要な2次元環境地図を生成する機能を持ち、その環境地図を利用して自身の位置推定を行い、設計された経路に沿った柔軟な自律走行が可能である。

本研究では、SUV に RGB-D カメラを搭載し、3次元地図生成のためのデータを収集する機能を実現した。3次元地図の生成には、SUV が持つ環境地図と RGB-D カメラのセンサーデータ、データ収集時の SUV の位置情報が必要になる。SUV の自律走行機能を利用して、環境地図を基にデータを収集する経路を生成する機能を追加し、SUV が生成した経路に沿って走行しながらデータを収集することで、実際に3次元地図を生成できることを確認した。

また、地図の精度を向上させるため、収集したセンサーデータの重ね合わせを行う前処理として、RANSAC (Random Sample Consensus) アルゴリズムを用いてセンサーデータから床の検出を行い、床が水平になるようにデータの修正を行う機能を実現した。センサーデータの重ね合わせに関して、SUV の位置情報を初期値として利用し、RGBD-ICP (Iterative Closest Point) アルゴリズムを拡張して、センサーデータ間の3次元の点群と画像特徴点の対応点だけでなく、センサーデータと環境地図との対応点も利用することで、環境地図と対応の取れた3次元地図を生成する機能を実現した。

センサーデータの重ね合わせに基づいて生成された3次元地図は膨大な点の集合であり、そのままでは利用が困難であるため、点群から平面を検出しポリゴン化する機能を実現した。3次元地図全体から平面を検出することは困難であるため、それぞれのセンサーデータから平面を検出し、検出した面の重ね合わせを行うことで3次元地図のポリゴン化を行う仕組みを実現した。

6.2 今後の課題

6.2.1 地図精度の向上

本研究では、重ね合わせの際の条件として、床を水平に固定することで、収束させるパラメータの数を減らし、地図精度の向上を図った。しかし、実際には床の検出に失敗する場合や、床がスロープになっているなどの可能性が考えられる。そのような場合、現在のアルゴリズムでは、フレームごとノイズとして除去してしまうか、スロープになっている床を水平に修正してしまうという問題がある。そこで、床を水平に固定しなくても良いように重ね合わせのアルゴリズムの改善を行うか、SUVにジャイロセンサーを搭載し、傾きを計測するなどの解決策が考えられる。

6.2.2 3次元オブジェクトの配置

本研究では、建物を構成する主要素として壁と床に着目し、3次元地図から平面の検出を行い、壁や床をポリゴン化した3次元地図を生成する機能を実現した。しかし、実際の環境では、机やイスなどの3次元オブジェクトが環境中に存在することが考えられる。そのため、複雑な面で構成される3次元オブジェクトをどのように扱うか考えなければならない。3次元オブジェクトのポリゴン化はそれだけで多くの計算量を必要とする。そのため、3次元地図生成時に精密な3次元オブジェクトのポリゴン化を行うのは困難である。しかし、データの収集は行っているため、そこに3次元オブジェクトが存在するという事は容易に検出できる。そこで、オブジェクトのおおよその大きさを検出し、同じくらいの大きさの直方体や円柱のオブジェクトを3次元オブジェクトが検出された位置に換わりに配置するという処置が考えられる。また、必要に応じてユーザーが任意の3次元オブジェクトを自由に配置できる機能を実現し、自動で配置された簡易オブジェクトをユーザーがより精密なオブジェクトに置き換えて、より高度な地図を作成するというやり方も考えられる。

6.2.3 データ収集方法の改善

データ収集に関する改善点として、センシング領域を考慮した経路生成、未知の障害物との遭遇時の行動計画、複数台のSUVの連携が考えられる。以下でそれぞれについて説明する。

本研究では、データ収集用に生成する経路は、壁からの距離を考慮するだけで、SUVのセンシング領域を考慮していない。そのため、データが収集できない領域や、収集したデータが3次元地図の生成には十分でない領域ができてしまう可能

性がある。この問題を解決するためには、SUVのセンシング領域を考慮した経路生成が必要になる。しかしながら、RGB-Dカメラのセンシング領域は視野角60度と狭く、そのままではセンシング領域を考慮した経路を生成するのは非常に困難であると思われる。そこで、RGB-Dカメラを回転させ、SUVの周囲360度をセンシング可能にするという方法が考えられる。360度をセンシング可能にすることで、センシング領域がSUVを中心とした円形になり、データを収集する必要がある地点を求めることができる。ただしこの場合、SUVは観測点で一時停止し、360度データを取り終えてから次の観測点に移動することになるため、データが連続的でなくなり、センサーデータの重ね合わせを連続するフレームで行うと重ね合わせに失敗するフレームが多くなると予想される。そこで、SUVの位置情報を利用し、うまく重ね合わせが行える可能性が高いフレームを選択するという手法が考えられる。SUVの座標と向きから、SUVがその地点で収集した領域を計算し、その領域が多く重なるフレームほど重ね合わせはうまくいく可能性が高いと考えられる。また、この手法はこの場合に限らず有効であると考えられるため、今後優先して実現する必要があると考える。

また本研究では、データ収集中に人を検出した場合、その場で一時停止をするという機能を実現した。しかしながら、実際の環境では、障害物は人だけとは限らない。また、未知の障害物と遭遇した場合の対処法についても考えなければならない。未知の障害物に対処法の違いから、移動障害物と静止障害物に分けられる。移動障害物である場合、対象が近付いていれば回避行動を、遠ざかっていればデータ収集の一時中断を行う必要がある。静止障害物である場合は、走行経路上に存在するのであれば経路の変更を行う必要がある。

本研究では、1台のSUVが全てのデータを収集するという前提で経路の生成や機能の実現を行ったが、大規模な環境の3次元地図を作りたい場合や、より効率的にデータを収集する手段として、複数台のSUVが連携してデータを収集するということが考えられる。また、SUV同士の連携はデータ収集だけでなく、SUVを道案内ロボットや見回りロボットとして利用する場合にも重要になる。

6.2.4 3次元地図を利用した応用システムの開発

本研究では、ナビゲーションシステムなどの基礎となる3次元地図の生成を行った。そのため、今後は生成した3次元地図を利用した応用システムの開発を行う必要がある。3次元地図を利用したシステムについては第4章で述べたが、主に以下のような応用が考えられる。

まず第一に、3次元地図に様々な情報を付与し、それらの情報を検索、閲覧する事ができるシステムが挙げられる。生成した3次元地図は地図上に存在する施設などに関する情報を持たず、テキストも不鮮明で分かりにくいいため、ナビゲーションなどの応用には適していない。そこで、3次元地図に関連情報をアノテーション

できる仕組みを提供することで、より利用価値の高い3次元地図を作る。アノテーションとして付与する情報には、より高解像度なテクスチャ画像や利用できる施設の情報、ユーザーによるコメントなどの情報などが考えられる。これらの情報を地図に付与する事により、3次元地図を疑似体験システムやナビゲーションシステムへ利用する事が可能となる。

ナビゲーションシステムの一例として、筆者の所属する研究室で研究開発されている個人用知的移動体 AT (Attentive Townvehicle) のインタフェースとして利用する事が考えられる。ATは目的地を指定することで、目的地まで自動走行が可能な個人用の乗り物である。ATの位置情報に関するインタフェースに3次元地図を利用することで様々な関連情報を搭乗者に提示できるようになるほか、実世界のATの位置と連動して3次元地図上の視点が移動することで、搭乗者が自身の位置を容易に把握することができ、目的地を容易に決定することができるようになると思われる。

これらの応用システムを開発する事は、今後の課題の中でも特に重要なものである。

謝辞

本研究を進めるにあたり、指導教員である長尾確教授には、研究に対する姿勢や心構えといった基本的な事から、研究に関する貴重な御意見、論文執筆に関するご指導等を頂き、大変お世話になりました。心より御礼申し上げます。

松原茂樹准教授には、ゼミなどを通じて、本研究に関する的確な御指摘や有益となるような御意見を頂き、大変お世話になりました。心よりお礼申し上げます。

大平茂輝助教には、研究に関する事から技術的なことまで幅広く御指導、御意見を頂き、大変お世話になりました。心より御礼申し上げます。

石戸谷顕太郎さんには、本研究に対するアドバイスや、基礎的な知識や研究に対する考え方等の様々な御指導を頂きました。また、研究室の良い雰囲気づくりをして頂きました。ありがとうございました。

ATプロジェクトのメンバーである渡邊賢さん、矢田幸大さんには、本研究を進めるにあたり不可欠であるSUVに関する技術的なご指導や、研究やプログラミングに関する様々なアドバイスを頂き、様々な面でお世話になりました。ありがとうございました。

高橋勲さん、棚瀬達央さん、川西康介さんには、ゼミ等で貴重な意見を頂いたことに加え、研究室における様々な活動の中で大変御世話になりました。ありがとうございました。

長尾研究室秘書の鈴木美苗さんには、学生生活ならびに研究生生活の様々な面で御支援をして頂きました。ありがとうございました。

以上の方々を始め、本研究を進めるにあたり、御支援、御協力頂いた全ての方々に、深く感謝の意を表します。

最後に、日々の生活を支えて頂いた両親にも最大限の感謝の気持ちをここに表します。ありがとうございました。

参考文献

- [1] A. Akbarzadeh, J.-M. Frahm, P. Mordohai, B. Clipp, C. Engels, D. Gallup, P. Merrell, M. Phelps, S. Sinha, B. Talton, L. Wang, Q. Yang, H. Stewenius, R. Yang, G. Welch, H. Towles, D. Nister, and M. Pollefeys. Towards Urban 3D Reconstruction from Video. In *Proc. of the Third International Symposium on 3D Data Processing, Visualization, and Transmission (3DPVT'06)*, pp. 1–8, 2006.
- [2] Jon Louis Bentley. Multidimensional binary search trees used for associative searching. *Communications of the ACM*, Vol. 18, pp. 509–517, 1975.
- [3] G E P Box and M E Muller. A Note on the Generation of Random Normal Deviates. *The Annals of Mathematical Statistics*, Vol. 29, No. 2, pp. 610–611, 1958.
- [4] Yan Cui and Didier Stricker. 3D shape scanning with a Kinect. In *ACM SIGGRAPH 2011 Posters*, pp. 57:1–57:1, 2011.
- [5] Paul Doliotis, Alexandra Stefan, Christopher Mcmurrough, David Eckhard, and Vassilis Athitsos. Comparing Gesture Recognition Accuracy Using Color and Depth Information. *Proc. of the International Conference on Pervasive Technologies Related to Assistive Environments (PETRA)*, 2011.
- [6] Austin I. Eliazar and Ronald Parr. DP-SLAM 2.0. In *Proc. of the International Conference on Robotics and Automation*, 2004.
- [7] Martin A. Fischler and Robert C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, Vol. 24, pp. 381–395, 1981.
- [8] Giorgio Grisetti, Cyrill Stachniss, and Wolfram Burgard. Nonlinear constraint network optimization for efficient map learning. *IEEE Transactions on Intelligent Transportation Systems*, Vol. 10, pp. 428–439, 2009.
- [9] Bay H. Speeded-Up Robust Features (SURF). *Computer Vision and Image Understanding*, Vol. 110, No. 3, pp. 346–359, 2008.

- [10] D. Hähnel, W. Burgard, and S. Thrun. Learning Compact 3D Models of Indoor and Outdoor Environments with a Mobile Robot. *Robotics and Autonomous Systems*, Vol. 44, No. 1, pp. 15–27, 2003.
- [11] P E Hart, N J Nilsson, and B Raphael. A Formal Basis for the Heuristic Determination of Minimum Cost Paths. *IEEE Transactions on Systems Science and Cybernetics*, Vol. 4, No. 2, pp. 100–107, 1968.
- [12] Peter Henry, Michael Krainin, Evan Herbst, Xiaofeng Ren, and Dieter Fox. RGB-D Mapping: Using Depth Cameras for Dense 3D Modeling of Indoor Environments. In *Proc. of the International Symposium on Experimental Robotics (ISER)*, 2010.
- [13] Davison A. J. MonoSLAM : real-time single camera SLAM. *IEEE Trans. Pattern Anal. Mach. Intell.*, Vol. 6, pp. 1052–1067, 2007.
- [14] K. Levenberg. A method for the solution of certain non-linear problems in least squares. *Quarterly Journal of Applied Mathematics*, Vol. 2, No. 2, pp. 164–168, 1944.
- [15] David G. Lowe. Object Recognition from Local Scale-Invariant Features. In *Proc. of the International Conference on Computer Vision*, Vol. 2, pp. 1150–1157, 1999.
- [16] Donald W. Marquardt. An Algorithm for Least-Squares Estimation of Non-linear Parameters. *SIAM J. on Applied Mathematics*, Vol. 11, pp. 431–441, 1962.
- [17] Iason Oikonomidis, Nikolaos Kyriazis, and Antonis Argyros. Efficient model-based 3D tracking of hand articulations using Kinect. In *Proc. of the British Machine Vision Conference*, pp. 101.1–101.11, 2011.
- [18] Jean Serra. *Image Analysis and Mathematical Morphology*. Academic Press, 1982.
- [19] S. Thrun, W. Burgard, and D. Fox. Probabilistic Robotics, The MIT Press, 2005. 上田隆一 (訳): 確率ロボティクス. 毎日コミュニケーションズ, 2007.
- [20] S. Thrun, S. Thayer, W. Whittaker, C. Baker, W. Burgard, D. Ferguson, D. Hähnel, D. Montemerlo, A. Morris, Z. Omohundro, C. Reverte, and W. Whittaker. Autonomous exploration and mapping of abandoned mines. *IEEE Robotics and Automation Magazine*, Vol. 11, No. 4, pp. 79–91, 2004.

- [21] 桑野寛人, 河野英昭, 入江徹, 前田博, 生駒哲一. 車載カメラ画像を用いた3Dポリゴン地図へのテクスチャマッピング. ファジィシステムシンポジウム講演論文集, 2008.
- [22] 渡辺賢, 井上泰佑, 長尾確. 小型無人移動体による個人用知的移動体のセンシング領域の拡張. 情報処理学会第73回全国大会, 2011.
- [23] 和泉信生, 古谷洋一郎, 石村俊幸, 吉田隆一. 施設におけるバリアフリー情報の三次元地図としての視覚化. 情報処理学会研究報告. グラフィクスとCAD研究会, Vol. 2004, No. 121, pp. 97–102, 2004.
- [24] 中本琢実, 山下淳, 金子透. レーザレンジファインダ搭載移動ロボットによる動的環境の3次元地図生成. 電子情報通信学会技術研究報告. WIT, 福祉情報工学, Vol. 106, No. 144, pp. 25–30, 2006.
- [25] 安田知加, 井上泰佑, 岸佳奈恵, 長尾確. 自動走行可能な移動体によるミュージアムでの鑑賞体験の個人化. 情報処理学会第72回全国大会, 2010.
- [26] 永松明, 中里祐介, 神原誠之, 横矢直和. 屋内環境におけるモバイルプロジェクション型AR案内システム. 日本バーチャルリアリティ学会論文誌, Vol. 14, No. 3, pp. 283–293, 2009.
- [27] 深澤佑介, Chomchana Trevai, 太田順, 湯浅秀男, 新井民夫, 浅間一. 格子点配置を用いた自律移動ロボットによる環境掃引経路計画. 計測自動制御学会論文集, Vol. 39, No. 11, pp. 1054–1060, 2003.