

屋内環境における手書き地図を活用した全自動 SLAM の直線観測確率を用いた移動量補正

○平尾至韻（中央大） Pathak Sarthak（中央大）

池勇勳（北陸先端大） 梅田和昇（中央大）

本研究では、実環境の形状と手書き地図の形状との対応関係を用いて屋内環境の地図を構築する方法を提案する。また、手書き地図と実環境の形状の差が大きい場合に直線観測確率を用いた移動量補正を行う手法を提案し、手法の有用性を検証する。

1. 序論

現在、自律移動ロボットによる人間の業務代替が注目されている。工場やショッピングモール、空港などの屋内施設に自律移動ロボットが導入され、移動や誘導の業務代替を行うケースが増加しており、これらのロボットの運用においては、走行環境の詳細な地図が必要不可欠である。

そこで、走行環境の地図構築に Simultaneous Localization and Mapping (SLAM) 技術が用いられる。SLAMとは地図構築とロボットの自己位置推定を同時に行うことでロボットの走行環境の地図を生成する技術のことである。この地図構築精度を向上させるべく、事前情報を活用した手法が考案されており[1]、誰もが利用でき、汎用性の高い事前情報を用いた新たな手法が求められる。町中らはGoogleマップから得た情報を基にロボットの軌跡地図を作成し自己位置推定に活用した[2]。しかし、これらはいずれも事前情報の準備やパラメータの調整に時間がかかる。一方、事前情報の中でも人の事前知識による手書き地図の準備は容易に可能である。BahramらはMonte Carlo Localization (MCL)[3]による自己位置推定手法を手書き地図上でも可能にした[4]。これにより、手書き地図を用いた自律移動ロボットの運用と手法がいくつか提案された[5][6][7]。また、Suzukiらは実環境と手書き地図の直線間の対応関係を地図構築に活用した[8]。しかし、これらの手法は移動ロボットの経路生成や手書き地図を活用した地図構築にとどまっており、ロボットのみで地図構築に必要な作業を全て自動化することはできない。

そこで、本論文では手書き地図を事前情報として活用する屋内環境向け全自動地図構築SLAMを提案する。本論文において全自動とは、「ロボットによる計測や移動制御、経路計画、地図構築といった地図構築に必要な全作業をロボットのみで行うこと」と定義する。

本研究では、まず与えられた手書き地図から環境情報を全て計測できる観測地点の決定とその地点の探索順序を求める。これ以降は上記で求めた地点と順路、環境の直線特徴を基に、手書き地図を活用したロボットの位置姿勢推定と経路計画、動作制御を繰り返すことで地図構築を行う。この際に、直線観測確率の値に応じて移動量に補正を行うことで、

与える手書き地図と実環境の形状の差が大きい場合でも頑健に動作する手法を提案する。

2. 手法概要

本研究で用いる手書き地図とは、専門知識を必要とせず誰でも作成可能な地図とする。ただし、本研究では直線環境の環境のみへの対応であり、手書き地図と実際の環境の縦横比が過度に異なることを前提とする。

提案手法の全体的なプロセスを図1に示す。

最初に手書き地図において、地図生成に必要な環境情報を全て計測できるロボットの最少の観測地点群を求める。この観測地点を探索する順序を巡回セールスマン問題によって算出する。これにより、地図生成に必要な環境の探索計画が求まる。これ以降は探索計画を基にロボットの移動、計測及び地図構築を繰り返す。まずオドメトリを用いて手書き地図上におけるMCL[4]を行い、ロボットの位置を推定する。次に、上記で求めた探索順路に沿って手書き地図上での経路計画を行う。その後、上記経路計画を基に実環境においてロボットの動作を制御する。このとき、直線観測確率の値に応じて移動量を補正する。移動後に計測を行い、環境において観測した直線と手書き地図上における直線間の対応関係を用いてロボットの自己位置推定を行うことで地図構築を行う。ここで、直線抽出は実環境では取得したセンサ情報に対して Random Sample Consensus (RANSAC) を用いる。手書き地図上では確率的ハフ変換を用いて直線情報を抽出する。また、両直線の対応関係は、手書き地図上でレイトレーシングを用いて取得する疑似2次元距離センサ点群の対応を求めることで算出する。

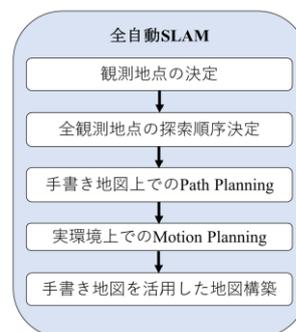


図1 全自動 SLAM の流れ

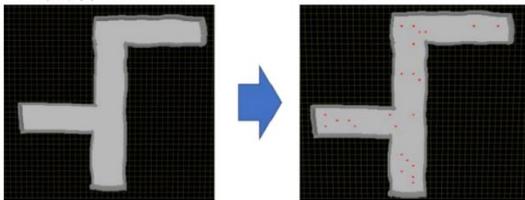
3. 観測地点および探索順路の決定

3.1 観測地点の決定

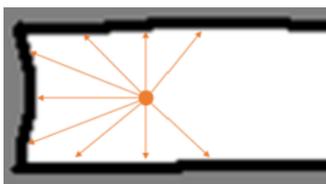
本手法では、最初に、Randomized Art-Gallery Algorithm for Sensor Placement[9]に基づく観測地点の散布及び決定を行う。このアルゴリズムは、与えられた地図内の環境全てを計測するために必要なセンサやロボットの観測地点群を求めるアルゴリズムである。本手法ではこの観測地点の算出にパーティクルを用いる。具体的な処理手順を以下に記す。

- 1) 図 2(a) に示すように、与えられた手書き地図上に観測点候補 (パーティクル) をランダムに散布する。
- 2) 図 2(b) に示すように、レイトレーシングを用いて、各候補点から計測できる地点を算出する。
- 3) Step2 の処理を Step1 で求めた全ての候補点 (パーティクル) に対して行い、各候補点から計測可能な地点を登録する。
- 4) Step3 で求めた候補点に対して、貪欲法を用いて与えられた地図内のすべての計測対象が計測できる観測地点候補の組み合わせを算出する。この際、より多くの計測対象インデックスが登録されている候補点から順に算出し、すべての計測対象が登録された時点で終了とする。

以上の処理によって、ロボットが通るべき地点の集合を求めることができる。ここで求めた観測地点にて計測を行うことで、与えられた地図内に存在する壁などの観測対象を全て計測することができる。



(a) 観測地点の決定



(b) 計測可能点群の計算

図 2 観測地点の決定

3.2 探索順路の決定

3.1 節で求めた手書き地図上の観測地点に対して、巡回セールスマン問題の解法を適用することで与えられた地図内の環境全てを計測する探索順路を決定する。この際、壁などの障害物も考慮する。図 2(a) に示す観測地点から求まる探索順路を図 3 に示す。なお、一部環境外に線が描画されているのは、実際の移動経路ではなく進む順序を点で結んだものを図示しているためである。

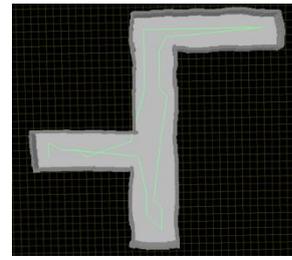


図 3 探索順序の決定

4. 手書き地図を活用した SLAM

本手法では、実環境での計測データを手書き地図上に反映させた上で、手書き地図上における経路計画[5]を行う。

まず手書き地図上における MCL[4]によって自己位置推定を行い、同時に手書き地図のスケール s も推定する。次に推定したスケール s を用いて、実環境での計測データを補正し、計測データを手書き地図上に反映させる。続いて、手書き地図上で最急降下法を用いた経路計画を行う。ここで、上記で取得した計測データ中に障害物が存在する場合には、障害物付近に移動コストを高く散布し、図 4 に示すように回避経路を生成する。これにより、手書き地図上にない障害物が生じる場合でも、障害物を回避した経路計画が可能である。最後に、この経路計画で取得した手書き地図上のロボットの移動量を、上述したスケール s を用いて実環境上での移動量に変換する。

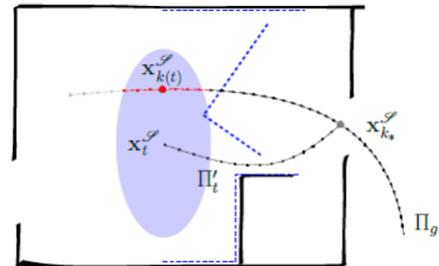


図 4 手書き地図上における経路生成

5. 環境における動作計画

本手法では、4 章にて求めたロボットの移動量を基に Dynamic Window Approach(DWA)[10]を用いて実際の動作制御に必要な車輪速度を求め、環境でロボットを移動させる。ここで、DWA は、移動ロボットの制御手法の一つである。DWA における時刻 t での出力速度は、左車輪速度 v_{left} と v_{right} の 2 変数をベクトルにまとめて、 $\mathbf{V}_t = [v_{left} \ v_{right}]^T$ と表される。DWA では、候補となる \mathbf{V}_t ごとに評価値を与える評価関数を定義し、その評価値が最も高い速度を探索して、その結果を出力速度とする。そのため、DWA により求まる車輪速度を用いて環境でのロボットの移動量を算出し、動作計画を行う。以上の処理によって、移動経路に沿ったロボットの動作が可能になる。

6. 手書き地図を活用した SLAM

6.1 直線観測確率と移動量補正

提案手法では、次節で述べるロボットの回転方向の自己位置推定に手書き地図上における直線観測確率を活用する。直線観測確率とは、現在フレームにて手書き地図上のどの直線が観測されているかを示す。本手法では、手書き地図上の各直線が観測されたかの判別に次式の直線観測確率を用いる。

$$f(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left\{-\frac{(x_i^t - s_t \widehat{x}_i^t)^2}{2\sigma^2}\right\} \quad (1)$$

$$f(y) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left\{-\frac{(y_i^t - s_t \widehat{y}_i^t)^2}{2\sigma^2}\right\} \quad (2)$$

ただし、 i は直線のインデックス、 $x_i^t(y_i^t)$ と $\widehat{x}_i^t(\widehat{y}_i^t)$ はそれぞれ t フレームで取得した実環境上における各直線までの距離と手書き地図上における各直線までの距離である。 σ は2次元距離センサで得られる距離の不確かさである。

式(1)と式(2)の直線観測確率が閾値を超えたときに直線は観測されたとみなす。また、直線観測確率がしきい値を下回ったとき、直線観測確率がしきい値以上になるように移動量を補正する。概念図を図5に示す。 $f(y)$ はしきい値以上、 $f(x)$ はしきい値以下であった場合、ロボットは x 軸正方向に一定の移動量を補正する。

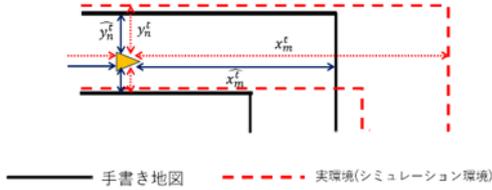


図5 直線観測確率

6.2 自己位置推定

6.2.1 回転方向自己位置推定

4.1 節で得られた手書き地図上で観測された直線と実環境で観測された直線を用いて、次式より回転方向のロボット姿勢の修正量を求める。これにより回転方向の自己位置推定を行う。

$$\theta = \frac{1}{N} \sum_{i=1}^M (\theta_{k_i} - \theta_i) \quad (3)$$

ただし、 θ は回転方向のロボットの修正量であり、 k_i は i 番目の観測直線に対応する手書き地図上の直線インデックスである。また、 M は対応が算出された直線の数である。

6.2.2 並進方向自己位置推定

回転方向の自己位置推定を行った後、point-to-plane ICP(iterative closest point)[11]を用いて並進方向の自己位置推定を行う。この時の並進方向の移動量は次式の距離の二乗和 E を評価値として算出する。

$$E = \sum_{i=1}^R \|n^T(y_{u_i} - (x_i + T))\|^2 \quad (4)$$

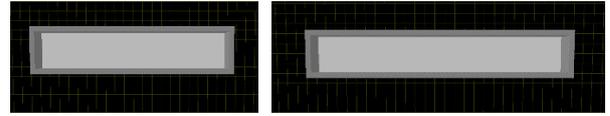
ただし、 x と y はそれぞれ前のフレームでの既存スキ

ャン点群(以下、ターゲット点群)の点と現在フレームでスキャンした点群(以下、ソース点群)の点であり、 n は法線ベクトルである。また、 R と u_i はそれぞれのソース点群の点の数(繰り返し計算の試行回数)とターゲット点群中の i 番目の点に対応するソース点群のインデックスである。 T は並進成分のベクトルを表す変換行列である。

7. 実験

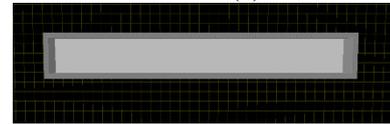
7.1 概要

本手法の検証のため、手書き地図を事前情報として図6(a)~(d)と図7に示すシミュレーション環境内で全自動SLAMを行う。手書き地図は、専門知識が不要なWindows 10内蔵ソフト「ペイント」を使用して作成し、手書き地図での1pixelは10cmとした。図6(a)~(d)の環境を用いた実験にて使用した手書き地図を図8に示す。図8は図6(b)とほぼ同じ比率であり、縦横比が大きくなったときに、直線観測確率に応じた移動量補正の有無によってその手法が有用であるか検証する。次に図7の環境を用いた実験にて使用した手書き地図を図9(a)(b)に示す。図9(b)は実験環境に比べて左の形状の横比率及び上に続く経路の縦比率にそれぞれ差があり、直線環境よりも複雑な環境で動作するか検証した。また、仮想2次元距離センサの最大測域距離は50mとした。

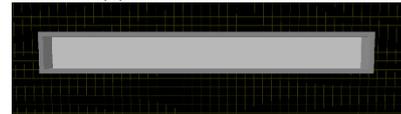


(a) 縦4m×横20m

(b) 縦4m×横25m



(c) 縦4m×横30m



(d) 縦4m×横35m

図6 シミュレーション環境1

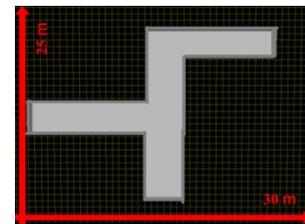


図7 シミュレーション環境2

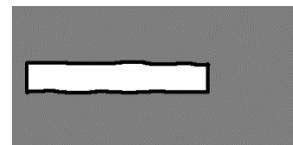
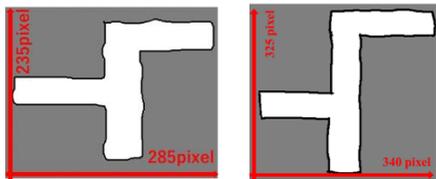


図8 手書き地図1



(a) 手書き地図2 (b) 手書き地図3
図9 手書き地図

7.2 実験結果

環境1の実験において、真の地図の点群と得られた地図点群のそれぞれ対応する点間距離の平均を図10に示す。図6(b)を用いた場合には手書き地図とシミュレーション環境の縦横比がほぼ等しいため、移動量補正の有無で誤差に差はほとんどない。図6(a)を用いた場合には縦横比が1倍の場合と誤差に違いは見られないが、縦横比が1.2倍の図6(c)を用いた場合は差が大きくなっている。これは、シミュレーション環境が広がったことで計測誤差が大きくなった。縦横比が1.4倍の図6(d)を用いた場合には、移動量補正の有無によって実験結果に差がみられた。これにより、縦横比が大きくなった場合でも安定して動作する手法だと考えられる。

環境2実験では図9(a),(b)の手書き地図を用いてシミュレーションを行った。図9(a)を用いた場合には移動量補正の有無に関わらずシミュレーションを動作することができた。縦横比が大きく異なる図9(b)において、移動量補正が無い場合には自己位置推定の累積誤差によって環境内での動作計画が不可になってしまった。一方で、移動量補正があった場合には直線部分の累積誤差を低減してシステムを最後まで動作することができた。したがって、本システムは縦横比が大きく異なる場合であっても有用なものであると考える。

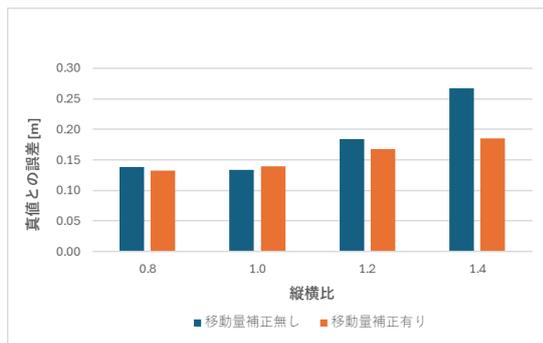
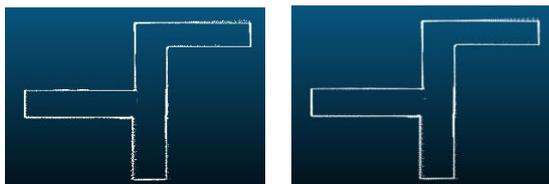
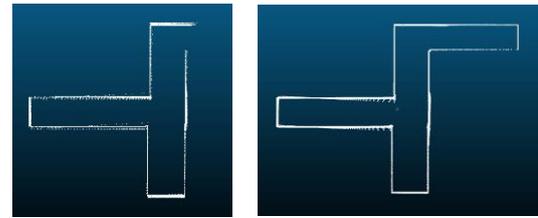


図10 実験結果と真値との誤差



(a) 移動量補正無し (b) 移動量補正有り



(c) 移動量補正無し (d) 移動量補正有り
図11 実験結果

8. 結論

本論文では、手書き地図を事前情報として与える屋内環境向けのSLAMにおいて、移動量補正によって頑健に動作する手法を提案した。移動量補正の有無による実験結果を比較することで手法の有用性を示した。今後の展望として、曲線環境に対応した手法の構築や現実世界での実装を目指す。

参考文献

- [1] 森武俊, 栗原誠, 黒田藍子, 野口博史, 田中雅行, 福井類, 下坂正倫, 佐藤知正: “パーソナルモビリティのための簡易マップを手がかりとする自己位置同定と詳細マップの生成 (移動ロボットの自己位置推定と地図構築),” ロボティクス・メカトロニクス講演会講演概要集, 2011.
- [2] Masaro Kimba, Noriaki Machinaka, and Yoji Kuroda, “Edge-Node map Based Localization without External Sensor Data,” IFAC-PapersOnLine vol. 51, pp. 203-208, 2018.
- [3] Dellaert, F., Fox, D., Burgard, W., and Thrun, S., “Monte Carlo localization for mobile robots,” In Proc. of ICRA, pp. 1322-1328, 1999.
- [4] Behzadian, B., Agarwal, P., Burgard, W., and Tipaldi, G.D., “Monte Carlo localization in hand-drawn maps,” In Proc. of IROS, pp. 4291-4296, 2015.
- [5] Boniardi, F., Valada, A., Burgard, W., and Tipaldi, G.D., “Autonomous indoor robot navigation using a sketch interface for drawing maps and routes,” In Proc. of ICRA, pp. 2896-2901, 2016.
- [6] Boniardi, F., Behzadian, B., Burgard, W., and Tipaldi, G.D., “Robot navigation in hand-drawn sketched maps,” In Proc. of ECMR, pp. 1-6, 2015.
- [7] Chen, K., Vázquez, M., and Savarese, S., “Localizing Against Drawn Maps via Spline-Based Registration,” In Proc. of the IROS, pp. 8521-8526, 2020.
- [8] Suzuki, R., Ji, Y., Pathak, S., and Umeda, K., “Indoor SLAM based online observation probability using a hand drawn map,” In Proc. SII, pp.695-698, 2022.
- [9] Héctor, G.B., “A randomized art-gallery algorithm for sensor placement,” In Proc. of the seventeenth annual symposium on Computational geometry, 2001.
- [10] Fox, D., Burgard, W., and Thrun, S., “The dynamic window approach to collision avoidance,” IEEE Robot. Autom. Mag., vol.4, no.1, pp.23-33, 1997
- [11] Low, K.L., “Linear Least-Squares Optimization for Point-to-Plane ICP Surface Registration,” Technical Report TR04-004, University of North Carolina, 2004.