移動ロボットによる手書き地図を活用した屋内向け地図構築手法の提案

Mobile Robot-based Accurate Indoor Map Construction from Hand-Drawn Maps

精密工学専攻　37 号　鈴木龍紀

Ryuki Suzuki

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．自律移動ロボットの運用においてSLAM（simultaneous localization and mapping）技術は必要不可欠である．SLAMとは文字通り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．地図構築の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きた．森らは建物の内装地図などの概ね形状が一致している簡易地図を事前地図として活用しロボットの自己位置推定を行った(1)．町中らはGoogleマップから得た情報を元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．またこれらの手法はいずれも，自己位置推定にとどまっており，地図構築はできない．

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MCL（Monte Carlo localization）(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の手法がいくつか提案された(5)(6)(7)．しかし，これらの手法は地図構築における手書き地図の活用方法ではなく，移動ロボットの経路生成や手書き地図上での自己位置推定にとどまっている．

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地図構築手法を提案する．この手法では，Bahramらの手法(4)を用いて手書き地図上におけるロボット位置を推定した後，実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．また，手書き地図を地図構築に活用することで，高精度な地図構築が可能になるだけでなく，地図生成を行う環境を事前に把握することが可能であることから，地図構築時に人間が行う経路計画やロボット制御全てを自動化できる利点がある．

2. 手法概要

本手法は 2 次元距離センサ搭載車輪型移動ロボットを前提としている．提案手法の全体的なプロセスを Fig.1 に示す．本手法は，手書き地図上における処理と実環境上における処理の 2 つに分けられる．

まず手書き地図における処理について説明する．初めに手書き地図から直線情報を抽出する．本論文では直線抽出に確率的ハフ変換を用いる．次に，オドメトリを用いて，手書き地図上における MCL(4)を行い，手書き地図上のロボット位置を推定する．この時の MCL(4)は一般的な MCL(3)と同様にパーティクルを活用し，ロボットの位置推定を行う他，状態変数に地図のスケールも加え手書き地図のスケールも同時推定する．この時，手書き地図上のロボット位置が走行不可領域であった場合，直前のフレームにて算出した直線の観測確率を用いてロボット位置の再推定を行う．直線の観測確率の算出に関しては後述する．続いて，レイトレーシングを用いて疑似 2 次元距離センサデータを点群として取得する．この時，疑似 2 次元距離センサデータの各点が，手書き地図から抽出した直線のうちどの直線に属しているかも同時に算出する．

次に，実際環境上における処理について説明する．初めに，オドメトリによるロボットの位置更新を行う．次に，ロボットに搭載した 2 次元距離センサで観測した点群から RANSAC（random sample consensus）を用いて直線を抽出する．続いて，手書き地図上における処理で取得した疑似 2 次元距離センサ点群と実環境にて取得した 2 次元距離センサ点群の対応を求めることで，手書き地図における直線と実際に観測した直線の対応を算出し，確率密度関数による直線の観測確率を求める．その後，直線の観測確率がしきい値を超えた直線を用いて，回転成分の自己位置推定を行う．また，スキャンマッチングを用いた並進方向の自己位置推定を行う．

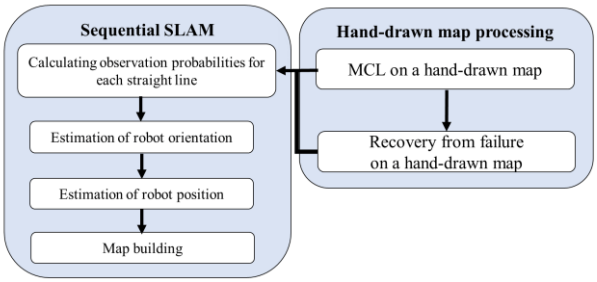


Fig.1 System overview

3. 直線観測確率算出

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活用する．

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

ただし，r_i^t と $\widehat{r_i^t}$ はそれぞれ t フレームで取得した点群の i 番目の点と t フレームにおける事前地図上のロボット位置から取得した仮想点群の i 番目の点である．また，σは2次元距離センサの不確かさである．$f(r)$は各直線の存在確率である．

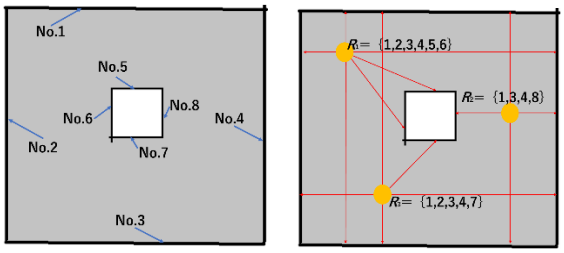
直線観測確率は対応関係にある手書き地図上にて疑似 2 次元距離センサから得られた点群の各点と，実際に得られた点群の各点の点間ユークリッド距離に基づいて算出される．上記の各点の対応関係は，点群のインデックスで求める．すなわち，式(1)の直線観測確率がしきい値を超えた時に直線は観測されたとみなす．

4. 手書き地図上におけるロボット位置の再推定

手書き地図上における MCL[4]によって走行不可領域にロボットの自己位置推定がされた時，直前のフレームで観測されたと見なされた直線を活用しロボット位置の再推定を行う．再推定は，手書き地図の事前処理と，MCL[4]を実行するための観測点の算出，MCL[4]の実行の 3 段階の処理に分けられる．

事前処理では，候補となる観測点 R_i をランダムに分布させる．そして，Fig.2 に示すように各候補観測点から観測できる手書き地図上の直線のインデックスを求める．ここで，黒色は壁，灰色は非占有領域をそれぞれ表す．黄色い点は候補観測点，赤い矢印は疑似観測データである．

続いて，観測点を決定するために直前のフレームにて手書き地図上のどの直線が観測されたかを求める． これは，MCL[4]が破綻したとしても直前に観測した直線を観測できる地点近辺にロボットがいる可能性が高いためである．よって，ロボットが直前のフレームに観測した手書き地図上の直線を観測できる観測点群を求め，それら周辺にパーティクルを散布することでロボット位置の再推定を行うことができる．



(a) Indices of lines (b) Observation points

Fig.2 Determination of observation points.

5. 自己位置推定

5.1 回転方向自己位置推定

3 章にて得られた手書き地図上における観測された直線と実環境で観測された直線を用いて，次式のように回転方向の自己位置推定を行う．

$$\theta = \frac{1}{N}\sum_{i=1}^{M}(\theta_{k_i} - \theta_i) \tag{2}$$

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事前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．

5.2 並進方向自己位置推定

回転方向の自己位置推定終了後，我々の以前の手法[9]と同様に point-to-plane ICP（iterative closest point）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．

$$E = \sum_{i=1}^{R}\left\|\boldsymbol{n}^T\left(\boldsymbol{y}_{u_i} - (\boldsymbol{x}_i + \boldsymbol{T})\right)\right\|^2 \tag{3}$$

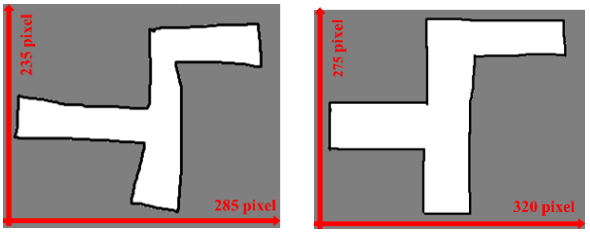
ただし，$\boldsymbol{x}$ と $\boldsymbol{y}$ はそれぞれ前のフレームでの既存スキャン点群（以下，ターゲット点群）の点と現在フレームでスキャンした点群（以下，ソース点群）の点であり，$\boldsymbol{n}$ は法線ベクトルである．また，R と u_i はそれぞれソース点群の点の数（繰り返し計算の試行回数）とターゲット点群中の i 番目の点に対応するソース点群のインデックスである．$\boldsymbol{T}$ は変換行列の並進成分ベクトルである．

6. シミュレーション

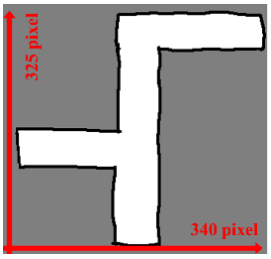
6.1 概要

本手法の検証のため Fig.3(a)と Fig.4(a)に示すシミュレーション環境にてシミュレーション実験を行った．手書き地図は，Windows 10 内蔵ソフト「ペイント」を使用して簡単に作成した．作成した手書き地図での 1 pixel は 10 cm として処理される．Fig.3 (a)の環境を用いた実験にて使用した手書き地図を Fig.3(b)～(d)に示す．Fig.3 (b)は乱雑に書いたため曲線が目立つ．Fig.3 (c)は実験環境に比べて左側の形状の横比率にズレがある．Fig.3(d)は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ズレがある．また，Fig.4(a)の環境を用いた実験にて使用した手書き地図を Fig.4 (b)～(d)に示す．与えた手書き地図の大きさは Fig.4 (a)が最も大きく Fig.4 (c)が最も小さい．仮想 2 次元距離センサの最大測域距離は 50 m とし，仮想エンコーダの誤差は正規分布の乱数を用いて表現した．

(a) Simulation environment 1

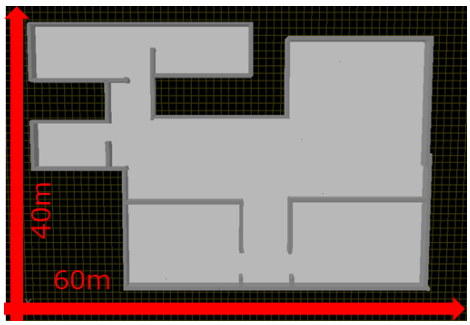


(b) Hand-drawn map A (c) Hand-drawn map B

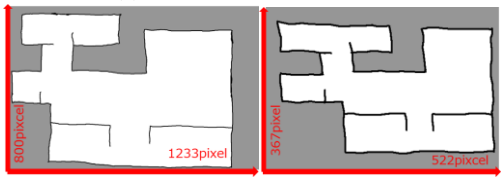


(d) Hand-drawn map C

Fig.3 Simulation1



(a) Simulation environment 2



(b) Hand-drawn map D (c) Hand-drawn map E

(d) Hand-drawn map F

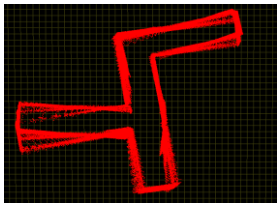
Fig.4 Simulation2

6.2 実験結果

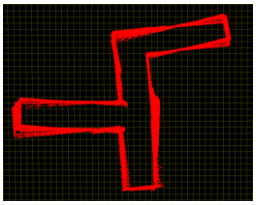
実験結果を Fig.5 と Fig.6，Table 1 と Table 2 に示す．Fig.5 と Fig.6 は構築された地図を示し，Table 1 と Table 2 は構築した地図の真値とのずれの平均を示している．これらの結果より，手書き地図を事前地図として活用することで地図構築の精度が向上したことが分かる．

Fig.3(a)に示すシミュレーション環境における実験では，Fig.3(d)の手書き地図を用いた場合は途中，手書き地図上における MCL[4]がうまくいかず手書き地図上のロボット位置を見失ったものの，直前のフレームにて計算した手書き地図上の直線観測確率を用いてロボット位置の再推定を行った．途中で手書き地図上のロボット位置を見失った原因として，使用した手書き地図上の通路形状の縦横比が場所により大きく異なっている事が考えられる．実際，本手法で使用した MCL について詳細に書かれている文献[4]では，実環境と手書き地図上の縦横比が大きく異なる場合，自己位置推定が正しくなされないと明記されている．提案手法では手書き地図上における大まかな自己位置が分かれば地図が構築できるため，文献[4]に比べ縦横比の変化には対応できるものの，急激な縦横比の変化により大まかな自己位置推定が成されず，地図構築が不可能であった．一方 Fig.3(b) の手書き地図は縦横比に大きなズレがあるものの，Fig.5(c)と Table1 に示すように高い精度で地図構築が成された．これは地図全体の大きさは違うものの，地図の縦横比に大きな違いがないためである．

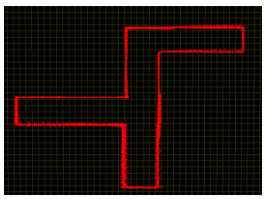
一方，Fig.4(a)に示すシミュレーション環境における実験では，Fig.4(c) (d)に示す手書き地図を活用した場合はシステムが破綻してしまった．これらはロボットが方向転換を行った際に走行不可領域にロボットが進んだためロボット位置の再推定が成されたが，元々大幅に手書き地図上のロボットの自己位置がズレていたため，ロボット位置の再推定が正確に成されず，手法が破綻してしまった．また，Fig.4(c) (d)に比べて Fig.4(b) が破綻しなかった理由として，与えた手書き地図のサイズが大きかったことが挙げられる．破綻してしまう例として，大幅に手書き地図上のロボット位置の推定がズレている状態でロボット位置の再推定を行うケースがある．一方，手書き地図上の走行可能領域が広く書かれている場合は走行不可領域にロボットが進むケースは少なくなり，取得した観測データが以前のフレームと大きく変化したタイミングで MCL[4]により正確な手書き地図上のロボット位置の推定がされる．以上の考察から，システムに与える手書き地図は走行可能領域を意図的に広く書くと，システムが破綻しにくくなることが分かった．



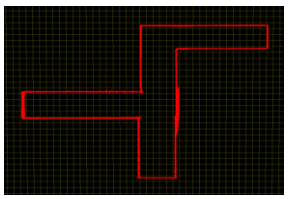
(a) Odometry



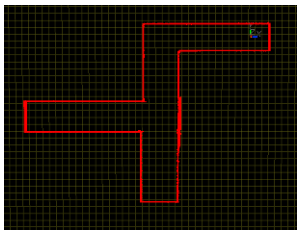
(b) ICP-SLAM



(c) Hand-drawn map A

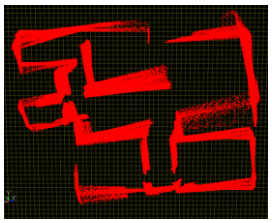


(d) Hand-drawn map B

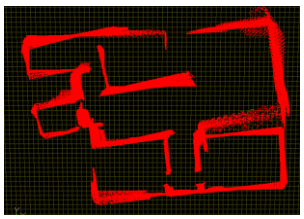


(e) Hand-drawn map C

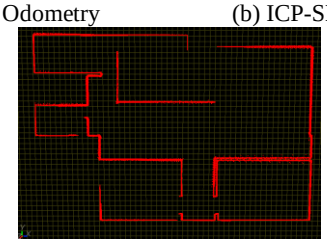
Fig.5 Map of simulation1



(a) Odometry



(b) ICP-SLAM



(c) Hand-drawn map D

Fig.6 Map of simulation2

Table 1 Recognition rate for each position of simulation1

methods	measurement error [m]
Odometry	6.157
ICP-SLAM	2.720
Hand-drawn map A	0.270
Hand-drawn map B	0.297
Hand-drawn map C	0.300

Table 2 Recognition rate for each position of simulation2

methods	measurement error [m]
Odometry	1.102
ICP-SLAM	2.165
Hand-drawn map D	0.189
Hand-drawn map E	-
Hand-drawn map F	-

7. 結論

本論文では，手書き地図を事前情報として与えることで高精度な地図構築を行う手法を提案した．提案手法では，手書き地図上における直線の観測確率及び実環境との対応関係をロボットの自己位置推定に用いることで高精度な地図構築を行った．しかし，複雑な環境下では手書き地図上の走行可能領域の広さの違いにより，システムが破綻しまうケースが見られた．よって今後の展望としては，乱雑な手書き地図にも対応できる手法の構築を目指す．

また，現状では直線環境にのみ対応しているため今後は曲線フィッティング等を用いてカーブ等を含む環境にも対応させる．

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 2011，2011.

[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] Frank Dellaert, Dieter Fox, Wolfram Burgard, and Sebastian Thrun, “Monte Carlo localization for mobile robots,” In Proc. of the 1999 IEEE International Conference on Robotics and Automation (ICRA1999), pp. 1322-1328 ,1999.

[4] Bahram Behzadian, Pratik Agarwal, Wolfram Burgard, and Gian Diego Tipaldi, “Monte Carlo localization in hand-drawn maps,” In Proc. of the 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS2015), pp. 4291-4296, 2015.

[5] Federico Boniardi, Abhinav Valada, Wolfram Burgard, and Gian Diego Tipaldi, “Autonomous indoor robot navigation using a sketch interface for drawing maps and routes,” In Proc. of the 2016 IEEE International Conference on Robotics and Automation (ICRA2016), pp. 2896-2901, 2016.

[6] Federico Boniardi, Bahram Behzadian, Wolfram Burgard, and Gian Diego Tipaldi, “Robot navigation in hand-drawn sketched maps,” In Proc. of the 2015 IEEE European Conference on Mobile Robotics (ECMR2015), pp. 1-6, 2015.

[7] Kevin Chen, Marynel Vázquez, and Silvio Savarese, “Localizing Against Drawn Maps via Spline-Based Registration,” In Proc. of the 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS2020), pp. 8521-8526, 2020.