

未知障害物環境に対応するための モンテカルロ自己位置推定における観測範囲の選択

Observation Range Selection in Monte Carlo localization for Unknown Obstacle Environments

○学 池邊龍宏 (千葉工大) 正 林原靖男 (千葉工大)
正 上田隆一 (千葉工大)

Tatsuhiko IKEBE, Chiba Institute of Technology,
Yasuo HAYASHIBARA, Chiba Institute of Technology
Ryuichi UEDA, Chiba Institute of Technology

We modify Monte Carlo localization as it can eliminate sensor measurements derived from unknown obstacles. Each particle in the modified MCL has its own observation range which is smaller than that of the sensor. In the calculations of Bayes theorem and resampling, the directions of the observation are changed as particles do not observe unrecorded obstacles on the map for self-localization. We have examined this method in a simulator. The result suggests that a restriction of the observation range makes MCL stable.

Key Words: Mobile robot navigation, Monte Carlo localization, LiDAR, Noise immunity

1 緒言

屋外での自律移動ロボットの自己位置推定は、Monte Carlo localization (MCL) [1] と、LiDAR の組み合わせで行われることが多い。MCL は確率的な自己位置推定手法で、センサの情報をベイズの定理で位置の情報に変換する。具体的には、数百程度のロボットの位置と向きの候補 (パーティクル) をデータとして持ち、センサの情報と一致するパーティクルを残していくことで、尤もらしい位置と向きを求める。LiDAR (Light Detection and Ranging) は2次元あるいは3次元のレーザースキャナのことを指し、ロボットから壁などの障害物までの距離を、1点ではなく平面状、あるいは立体的に計測できる。

この組み合わせの場合、ロボットには LiDAR のセンシング対象となる、壁などの障害物の位置を記録した地図を持たせることになる。MCL は LiDAR からの距離の情報と地図の障害物の位置を比較し、ロボットの位置と向き (位置と向きをあわせて、以後「姿勢」と呼ぶ) を求める。

この方式の場合、地図に記載のない障害物 (未知障害物と呼ぶ) が雑音となる。屋外では歩行者や走行中、停車中の自動車、自転車などが、未知障害物となる。例として、図1に、つくばチャレンジ [2] の様子を示す。つくばチャレンジというのは、実際に人や自動車を通る横断歩道、公園において、ロボットを約 2[km] 自律走行させる技術チャレンジである。図1を撮影したときは、ロボットを見物する人がスタート地点に押し寄せており、LiDAR の示す障害物と地図上の障害物が一致しない状況であった。特に2次元の LiDAR を用いる場合、このような状況に対し、MCL の側でなんらかの対策をしないと、位置推定が破綻しやすくなる。

この問題への対策の例として、富沢ら [3]、赤井ら [4] の研究事例がある。富沢ら [3] は、LiDAR の値から周囲の地

図を作成し、パーティクルの姿勢ごとにロボットの持つ地図と照合して不一致度を求め、ロボット周辺が地図と異なる状況になっていることを検知した。一方、赤井ら [4] は、ロボットの姿勢とセンサ観測のクラスを同時に推定し、地図上に存在する障害物から得られた観測のみを使用する自己位置推定法を提案した。センサ観測のクラスとは、得られた観測が地図上にある障害物であるかを表すものである。

これらの手法は、移動ロボットでも有効性が検証されている一方、計算量が通常の MCL より大きくなる。そこで筆者らは、「未知障害物に当たったレーザを無視」することを考えた。逆に言うと、センサから送られてくる情報から、自己位置推定に有効なものだけを選別して利用するという考え方となる。この方法の場合、MCL へ入力されるセンサ情報は減るので、選別アルゴリズムの計算量が小さければ、むしろ全体の計算量は減少する。

そこで本稿では、未知障害物に当たったであろうレーザを無視する簡素な手法をひとつ示し、この手法を評価する。この評価により、自己位置推定に有効なセンサ情報を選別する手法の可能性を議論する。2章では、実装した未知障害物への対策方法、3章では実験の方法と結果について説明する。4章で本稿をまとめる。

2 観測範囲を制限した MCL

実装する手法は、ひとつのパーティクルがベイズの定理を適用される際に使われる LiDAR の観測範囲を制限し、パーティクルごとに、その範囲を変えるというものである。たとえば2次元 LiDAR で、ロボット前方を $0[\text{deg}]$ としたときに左右 $\varphi_{\max}[\text{deg}]$ の範囲が計測できるとする。このとき、MCL で各パーティクルの重みを変えると、 $\varphi^{(i)} \pm \varphi'[\text{deg}]$ の範囲の距離計測値しか用いない。ここで i は i 番目のパーティクルという意味である。つまり、パーティクルによって使う距離計測値の範囲を変える。また、 $-\varphi_{\max} \leq \varphi^{(i)} - \varphi'$,

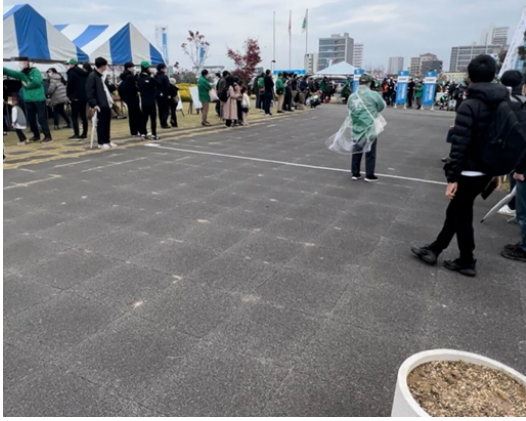


Fig.1 Crowds at the starting point of Tsukuba Challenge 2022

$\varphi^{(i)} + \varphi' \leq \varphi_{\max}$ とする。

このように観測範囲を限定し、パーティクルごとに可変とする狙いは、未知障害物に観測範囲が向いているパーティクルを減らすことである。たとえば自己位置推定が安定している状態でロボットが直線状の通路を前進していると、右の路肩に地図には記載のない乗用車が停車してあったときの状況を考える。このとき、パーティクルのうち、観測範囲が乗用車に向いているものは、地図と LiDAR からの値の整合性がとれず、リサンプリングの際に消去されやすい。この際、生き残ったパーティクルは、複製される。このような淘汰が進むと、パーティクルが未知障害物の観測を避けるようになる。

この方法は、未知障害物への対策になっていると同時に、各パーティクルにベイズの定理を適用する際の計算が $\Delta\varphi^{(i)}/\varphi_{\max}$ 倍に減少する。一方、LiDAR から MCL へ取り込む情報が減少することや、パーティクルの観測する向きが一方に揃ってロボットのバスタ性がなくなる問題の発生が懸念される。後者については、リサンプリングの際、 $\beta[\%]$ のパーティクルの観測範囲をランダムに変更することで対処する。次章の実験では、 $\beta = 10[\%]$ としている。前者については、次章で検証する。

3 実験

3.1 実験環境

実験は、つくばチャレンジの環境を再現したシミュレータ環境で行う。図 2 は、シミュレータ内で図 1 の場所、状況を再現した部分である。図中にある壁のようなものはロボットが持つ地図中に記載のある障害物（既知障害物）で、実機に搭載した 2 次元 LiDAR で得たデータをもとに配置している。同じく図中に見られるマネキンと箱は、図 1 を参考に、未知障害物として配置した。これらの未知障害物は、静止している。

シミュレータ内で走行させるロボットは、図 4 のような差動二輪型のロボットである。地面から高さ 0.7[m] の位置に、観測範囲を 360[deg]、角度分解能を 1[deg] の 2 次元

LiDAR を搭載している。また、シミュレータ内の既知、未知障害物との距離は、この LiDAR で検知できるように設定されている。

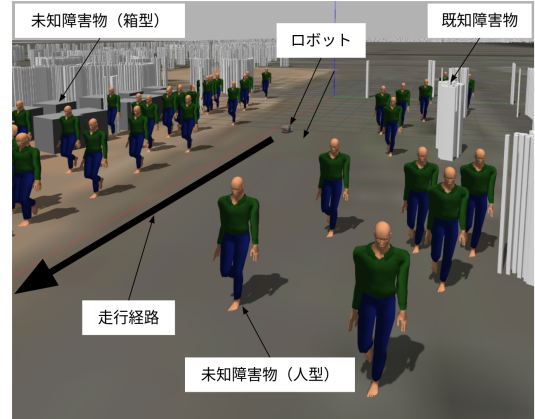


Fig.2 A scene in the simulator imitating Fig. 1

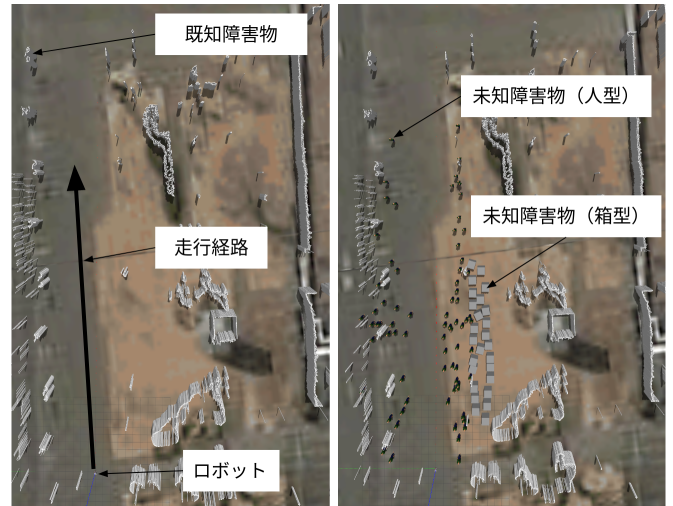


Fig.3 Environments without (left) and with (right) unknown obstacles

3.2 完走率の比較

まず、様々な観測範囲 φ' でロボットを自律走行させて、完走率を調査した。図 5 左に示すスタート地点からゴール地点までを走行できれば完走とする。このナビゲーション自体は簡単なものであるが、MCL が破綻すると、完走できない。また、未知障害物の配置や数については、後述の結果のように、通常の MCL では推定が破綻するように調整した。パーティクル数は 10,000 とした。

観測範囲を変えて、それぞれの観測範囲で 100 回試行した結果を表 1 に示す。センサを用いない場合 ($2\varphi' = 0[\text{deg}]$) や観測範囲に制限を設けない場合 ($2\varphi' = 360[\text{deg}]$) は、完走した試行が一度もなかった。一方、 $2\varphi' = 1[\text{deg}]$ と極端に観測範囲を狭めた場合の完走率は 28[%]、 $2\varphi' = 10[\text{deg}]$ の場合については 54[%] となった。

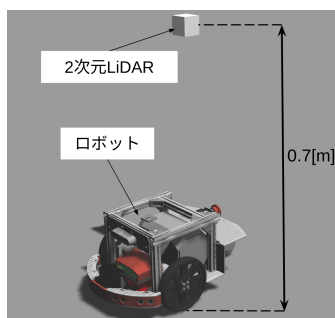


Fig.4 The robot running in the simulator

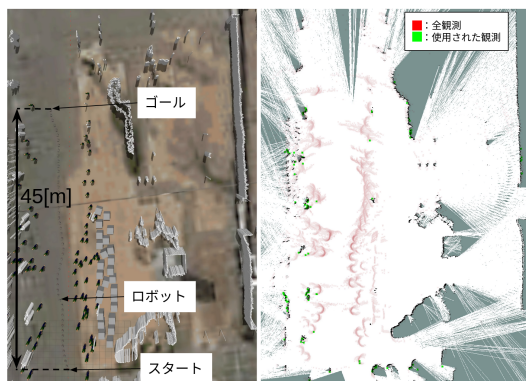


Fig.5 Navigation from start to finish (left) and visualization of observations used in the observation update (right)

図5右は、観測範囲が $2\varphi' = 1[\text{deg}]$ のときに、観測対象となった距離計測値と、ならなかった距離計測値を、障害物上にプロットしたものである。観測対象となった距離計測値は、MCLの周期たびに、全てのパーティクルで使用されている観測範囲の最頻値として求めたものである。緑色が観測対象となった距離計測値で、ほとんどが既知障害物を計測したものとなっている。2章の手法で、未知障害物を計測した値を除去することが可能であることが分かる。

4 結言

本稿では、MCLとLiDARを組み合わせて自己位置推定するときに、未知障害物を計測したセンサ値を省く仕組みを提案し、評価した。具体的な方法は、各パーティクルごとに2次元LiDARの観測範囲を制限し、観測する方向をランダムに変えるというものであった。シミュレータ内

に構築した未知障害物の多い環境で、数種類の観測範囲を試したところ、 $360[\text{deg}]$ 観測可能なLiDARに対し、 $1[\text{deg}]$ や $10[\text{deg}]$ など、極端に観測範囲を制限したほうが良好な結果が得られた。

今後は、観測範囲を制限することで起こる悪影響はないかを確認するために、実環境で実験する。また、観測範囲を制限したことで計算量が減少しているはずであるが未検証なので、この点についても調査する。

参考文献

- [1] Jens-Steffen Gutmann and Dieter Fox, “An Experimental Comparison of Localization Methods Continued,” Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 454-459, 2002.
- [2] 油田信一, “つくばチャレンジ: 市街地における移動ロボットの自律走行の公開実験 —11年の経緯と成果—”, 第23回ロボティクスシンポジウム講演論文集, (2018), pp. 59-66.
- [3] 富沢哲雄, 村松聡, 平井雅尊, 佐藤晶則, 工藤俊亮, 末廣尚士, “グリッドマップのマッチングに基づく未知障害物にロバストな自己位置推定”, 日本ロボット学会誌, Vol. 30, No. 3, pp. 280-286, 2012.
- [4] 赤井直紀, モラレスルイス洋一, 平山高嗣, 村瀬洋, “幾何地図上での観測物体の有無を考慮した自己位置推定”, 計測自動制御学会論文集, Vol. 55, No. 11, pp. 745-753, 2019.

Table 1 Relation between the observation range and the number of success trials

$2\varphi'[\text{deg}]$	完走率
0	0.00
1	0.28
10	0.54
45	0.04
90	0.00
360	0.00