

人混み中での移動ロボットの自己位置推定に有効な 測域センサデータ処理法

-自己位置推定における提案手法の実験的効果検証と評価-

藤井 祐介^{†1}, 大矢 晃久^{†1}, 坪内 孝司^{†1}

Effective LIDAR Data Processing Method for Mobile Robot Localization in Crowded Environments

-Evaluation through Experiments-

Yusuke FUJII^{‡1}, Akihisa OHYA^{‡1}, Takashi TSUBOUCHI^{‡1}

Abstract—The purpose of this study is to enable a mobile robot to localize itself in crowded environments. In this study, we have proposed a method to obtain effective LIDAR data for mobile robot localization in crowded indoor environments by the process of scan data accumulation and far point extraction. We examined effectiveness of our method through evaluation experiment.

1. はじめに

現在、人が生活する空間で自律的に移動し、活躍するロボットの開発が進んでいる。博物館や美術館に展示物をガイドしながら巡回するロボットを導入する事例もあり、その開発が進められている [1]。博物館では来場者によって混雑することもあり、障害物が多数存在する環境でもロボットは目的地まで迷わず案内できることなどが求められる。このタスクを実現できる案内ロボットを構成するためには、地図作成、経路設定、自己位置推定、障害物検出、動作計画など、考慮すべき要素が多数ある。そのうち、案内ロボットが自律走行する際に人混みに大きく影響を受ける要素は、自己位置推定と障害物検出、動作計画である。その中で本研究では自己位置推定を取り扱う。

移動ロボットが自己位置を推定するためには、周辺環境の情報を取得する必要がある。取得できる情報は使用するセンサによって様々だが、現状多くのロボットにおいて測域センサが搭載されることが多い。測域センサは水平方向の2次元形状情報を取得することができ、かつカメラなどで問題となる照明条件の影響がないことが利点として挙げられる。測定範囲も広いこと、センサー一つで十分な情報量が手に入ることから、本研究では水平方向の測域センサデータを用いた自己位置推定を考える。

測域センサを用いた移動ロボットの自己位置推定の方法の一つとしてスキャンマッチングがある。これは既知の物体の位置や壁面の形状が記された地図データと、周囲の環境を測域センサで観測して得られるスキャンデータとを照らし合わせて自己位置を求める方法である [2]。しかし、人が密集している場所では人に遮られてセンサから周囲の環境が見えず、かつ周囲の見え方の変化も激しいために、スキャンマッチングでは自己位置推定に大きな誤差が生じる問題がある。

筆者らは、この問題に対して有効な測域センサデータ処理法を提案し、人混み中でも人の点群を削減し壁の点群のみに近いスキャンデータを取得できることを確認した [3]。本稿で

は、本手法による処理データを用いて自己位置推定までを行い、位置推定に対して実際にどのような効果があるのか、人混みを実験的に再現しながら検証した結果について報告する。

2. 関連研究

人混み環境での自己位置推定に関する先行研究は盛んであり、測域センサを用いたさまざまな方法が提案されている。D.wolf らの手法 [4] では、動いている物体を認識して SLAM の処理上で除外し、精度の向上を図っている。しかし、このような手法は動いていない障害物を認識、除外することが難しいといった課題がある。Masuda らの手法 [5] では、ICP アルゴリズム [6] において地図データとスキャンデータの対応点間距離の平均値と標準偏差を用いた外れ値除去の要素を入れた方法を提案している。この手法は、地図データと同じ形状のスキャンデータがある程度取得できないとはずれ値の閾値設定が難しく、誤対応点の影響が残ってしまう懸念がある。日永田らの手法 [7] では、対応点間距離の2乗和で表される L_2 ノルムの代わりに、対応があると判別された点の数をを用いた L_0 ノルム最小化を用いた SLAM を提案している。この手法では動的環境下においても正確な地図作成と自己位置推定を可能としているが、 L_0 ノルムの最適化計算に時間がかかること、インライアの点群がアウトライアの点群より少ないといううまくいかない、という課題が挙げられている。

データ処理を工夫する関連研究としては、Fox らの手法 [8] が挙げられる。これは人の密度が高い動的な環境において、フィルタによって地図と異なっているセンサの値を検出し、ロボットが正しい位置にいることの信頼度を上げるスキャンデータのみを取り出す方法である。しかし、この手法は事前に正確な地図データを作成する必要や、精度を重視すると計算量が多くなり処理時間が長くなるトレードオフの関係といった課題が挙げられる。

これらに対して本研究で提案している手法は、人混みの状況や動作に影響されず、計算時間を抑えて人混み中でも周辺環境の形状を取得できるという特徴がある [3]。

3. 蓄積と遠方点抽出によるスキャンデータ処理

人混み中で自己位置推定に失敗する原因は、オクルージョンなどにより地図データに記されている壁などのものが観測で

^{†1} 筑波大学 大学院 システム情報工学研究科

^{‡1} Graduate School of Systems and Information Engineering,
University of Tsukuba

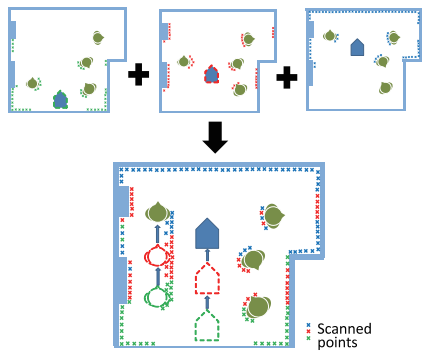


Fig. 1 Accumulation of the scan data obtained at each point

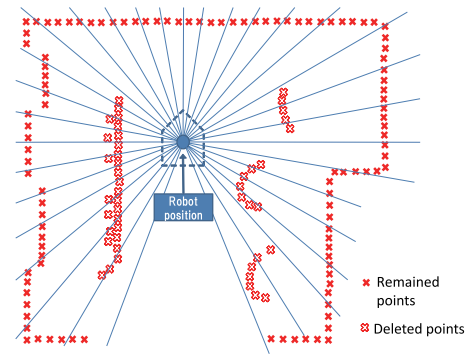


Fig. 2 Far point extraction

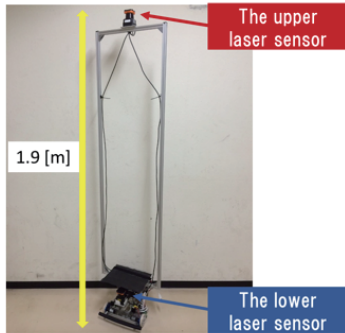


Fig. 3 Mobile robot used in the experiment



Fig. 4 A scene of experiment

きないことと、地図データに記されていない人などのスキャンデータが得られてしまうことの2つが考えられる。筆者らが提案したスキャンデータ処理法は、この2つの対策を同時に行うことができる。

まず、人混み中で人などの障害物に遮られて周囲の環境の一部しかセンサから観測できない問題には、スキャンデータの蓄積が効果的である。Fig. 1のように、ロボットの走行中それぞれの瞬間に周囲の壁が一部ずつしか見えていなくても、その一部ずつ見えているデータを集めることで壁全体の形状を取得できる。そこでデータ処理の始めとして、まずロボットが動きながら一定距離を走行する間、測域センサのスキャンデータをロボットのオドメトリにより世界座標系に変換したデータを蓄積する。

次に、人などの障害物が観測されてしまう問題には、遠方点抽出の処理を行うことで対策できる。屋内などの壁に囲まれた部屋の中などでは、人や障害物と地図データとなる壁などの位置関係に注目すると、必ずロボットから見て障害物は手前に位置し、地図となる壁は最も遠くにある。この関係性に注目し、先ほどの蓄積したスキャンデータの中から、ロボットから見て最も遠くにある点だけを取り出すことで壁のみのスキャンデータを得ることができる。そこで次のデータ処理として、Fig. 2のように、蓄積したデータに対してロボットの現在位置を中心に、微小角度ごとに放射状に区切る。最後に、微小角度の範囲内でロボットの位置から見て最も遠い距離にあるデータだけを抽出する。この3つのデータ処理手順により、人混み中においても人がいない時と同等のスキャンデータを取得することができる。

4. 評価実験

提案するスキャンデータ処理の有無による、人混み中での自己位置推定の結果を比較した。自己位置推定の手法は様々な存在するが、本稿では屋外などの実環境下においてもロバーストに自己位置推定が可能であるとされるMCLを対象として評価を行った。

4.1 方法

4.1.1 データ取得

Fig. 3に、実験で使用したロボットの外観を示す。ロボットには、地上から高さ1.9[m]の上部と地上から高さ0.2[m]の下部に測域センサUTM-30LXを搭載した。自己位置推定の評価は、まずロボットが走行しながら上部測域センサで人混みの影響を受けずに自己位置を推定し、この位置を真値とする。同時に下部測域センサで人混み中での自己位置を推定し、両者の位置推定結果の差を誤差として評価を行った。実験場所は7[m]四方程度の屋内の部屋の中とし、Fig. 4に示すような部屋の左右両側の壁際に人の列ができていた状況で行った。列に並ぶ人々には5秒間隔で1歩ずつ前進し、列の先頭の人には反対側の列に移動してもらった。壁際の障害物は、外れ値除去による除外がしにくく、マッチングのずれに影響を与えやすいことを踏まえ、この設定とした。この状況の中で、部屋の中央に四角い経路を設定してロボットを走行させてデータを取得した。

4.1.2 データ処理による壁抽出

データの蓄積距離を0.5[m]と設定し、その区間での蓄積データと遠方点抽出結果をFig. 5に示す。1回のスキャンデータだけでは、壁際の人によってセンサから左右両側の壁が部分的にしか見えず、壁の直線を全体的に取得できていない。

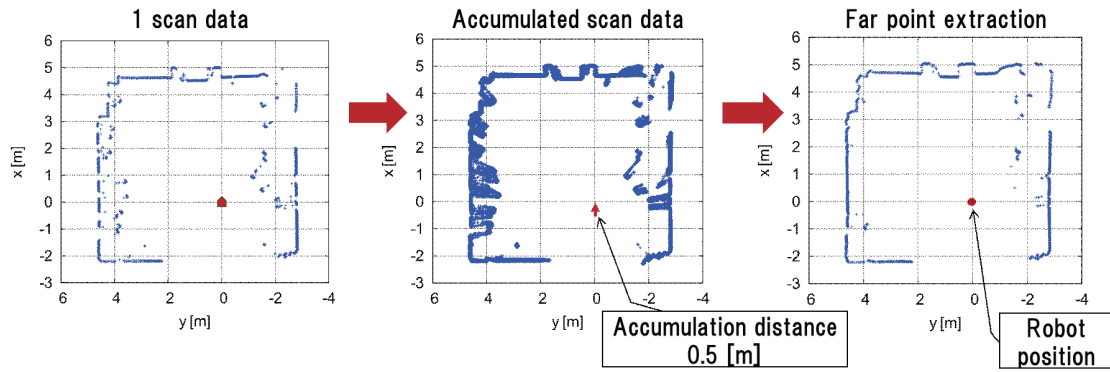


Fig. 5 Result of data processing

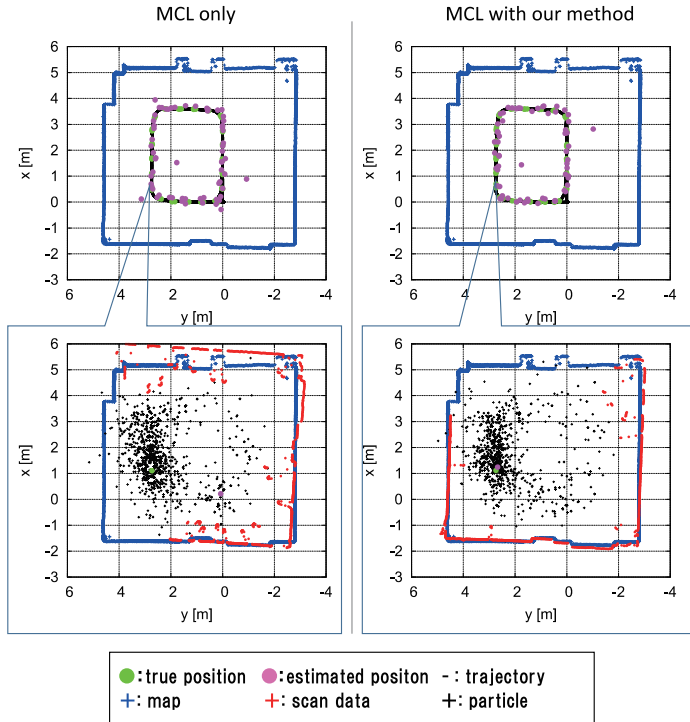


Fig. 6 Upper row:Reference positions(green) and estimated positions(purple), Lower row:An example of MCL matching result(sampling parameter = 0.6)

しかし、0.5 [m] 走行している間のスキャンデータを蓄積することで、蓄積中に一部ずつ見えていた壁の形状の全体像を取得できている。そこからロボットの位置を基準に遠方点抽出を行った結果、壁の取得範囲が増加し、壁際の人点群が削減された壁のみに近いスキャンデータを取得することができている。この実験ではロボットは0.15 [m/s] で走行し、0.5 [m] 走行するまでに約 10 万点のスキャン点を蓄積していた。しかし、この点群数に対して一回の遠方点抽出の処理は約 0.06 [s] で行うことができているため、処理時間に関しては問題なく行える。

また、蓄積距離の設定に関しては、人混みの程度や障害物の大きさなどに合わせて調節可能である。蓄積距離を増やすほど壁が見える可能性が増えて壁形状を取得しやすくなる。しかし、蓄積距離を増やし過ぎるとオドメトリで重ねあわせていることから少しずつゆがんでいく事も考えられるため、自己位置推定の頻度やオドメトリの精度などに合わせて設定す

る必要がある。

4.1.3 自己位置推定

自己位置推定手法には、最適化計算に基づいた方法と確率論に基づいた方法の 2 種類に分類される。ICP スキャンマッチングなどに代表される最適化手法は、マッチング時に外れ値などの影響を大きく受けやすく、人混み中での自己位置推定は不得意という特徴がある。一方、確率論に基づく手法の一つである Monte Carlo Localization(MCL) は、パーティクルフィルタを用いたアルゴリズムであり、人混みなどの動的環境下においてもロボストに自己位置推定ができることで知られている。本研究ではもともとロボスト性が高い MCL に注目し、確率ロボティクス [9] を参考にして実装を行った。MCL にはパーティクルのばら撒き方や計測モデルのパラメータ値など調整する部分が複数あり、これを適切に行うことでロボストな自己位置推定を維持することができる。逆に、パラメータの調節が適切でないと人混み中などでは位置推定の誤差が大

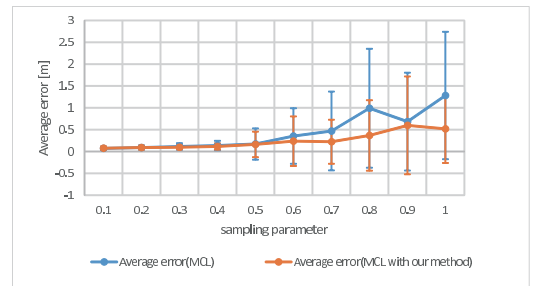


Fig. 7 A comparative evaluation of average localization errors between MCL only(blue) and MCL with our method(orange) (by sampling parameter)

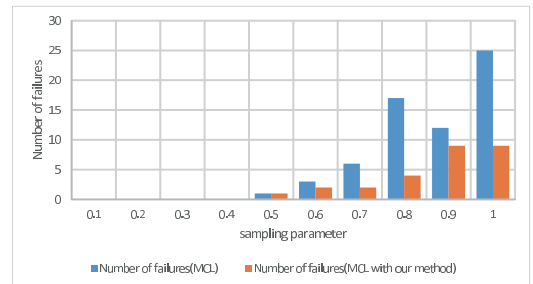


Fig. 8 A comparative evaluation of number of failures between MCL only(blue) and MCL with our method(orange) (by sampling parameter)

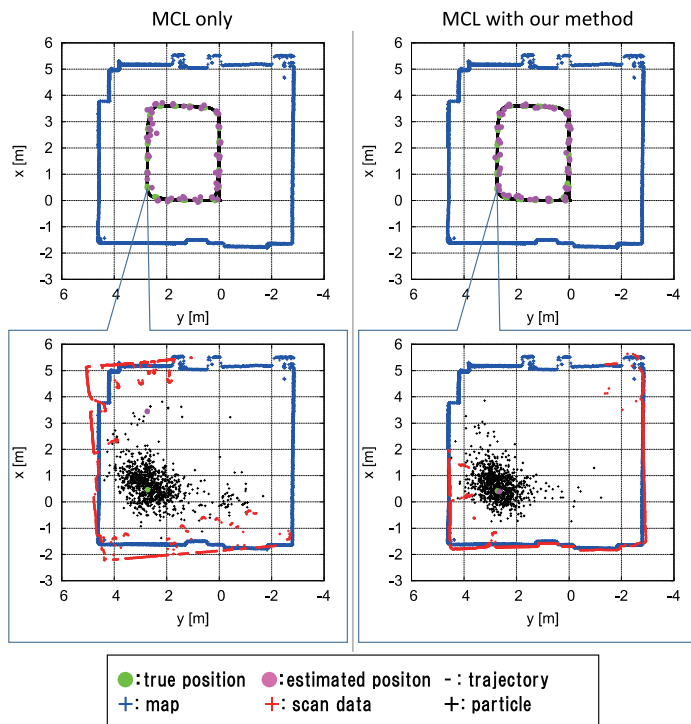


Fig. 9 Upper row: Reference positions (green) and estimated positions (purple), Lower row: An example of MCL matching result ($hit = 0.8$)

きくなったり、失敗したりする可能性が高くなる。そこで、提案手法を MCL に組み込んだとき、提案手法がない時と比べて位置推定の誤差がどの程度変化するか、また位置推定に失敗する回数などがどの程度変化するか検証した。

4.2 結果

4.2.1 サンプリングに関するパラメータ設定への影響

パーティクルフィルタでは、パーティクルを真値付近に散布することで精度を保ちつつロバストな位置推定が可能になる。パーティクルをサンプリングするときの動作モデルには、オドメトリ動作モデルを採用した [9]。このオドメトリ動作モデルにはパーティクルをサンプリングするときの並進量と回転量に関するパラメータが存在し、ロボットのオドメトリ特性や自己位置推定の間隔などに合わせて調節することが重要となる。このパラメータは、値を小さくするとサンプリング範囲が狭まり、値を大きくするとサンプリング範囲が広がることを意味する。人混みなどの障害物が多数存在する環境の場合、スキャンデータの形状が地図と大きく異なるため、パーティクルのサンプリング範囲が広いと予期しない位置の尤度が高くなってしまふ可能性が出てくる。逆に、パーティクルのサンプリング範囲を狭くしすぎると、精度が出やすくなるがロボットの位置が大きすぎてしまった時に復帰しにくくなる傾向がある。そこで、提案手法のデータ処理の有無によって、パーティクルのサンプリング範囲を変化させた時の人混み中での自己位置推定の誤差や、位置推定に失敗する変化を検証した。そこで、提案手法のデータ処理により、パーティクルのサンプリング範囲を変化させた時の人混み中での自己位置推定にどのような効果があるのか検証した。この時、パーティクルのサンプリング数は 1000 個で固定した。

Fig. 6 の上段に、サンプリングパラメータの値を 0.6 にしたときの自己位置推定の結果を、下段にある箇所でのマッ

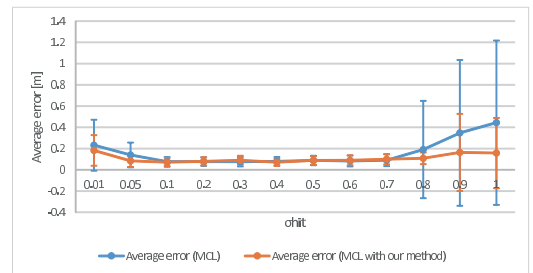


Fig. 10 A comparative evaluation of average localization errors between MCL only (blue) and MCL with our method (orange) (by hit)

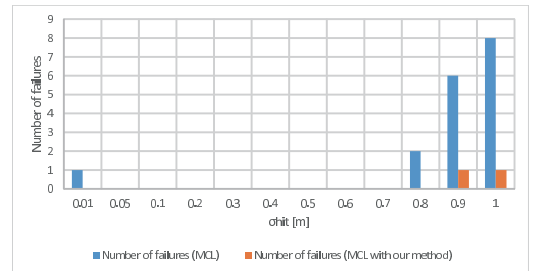


Fig. 11 A comparative evaluation of number of failures between MCL only (blue) and MCL with our method (orange) (by hit)

ングの様子を示す。また、Fig. 6 の左側が MCL のみの時の結果、右側が提案手法を MCL に組み込んだときの結果である。今回の実験では、ロボットが経路に沿って部屋を 3 周走行する間、0.5 [m] 走行ごとに計 63 回自己位置推定が行われた。MCL のみの場合、人の点群やパーティクルの回転量が大きい影響で、位置に関しては真値 (黄緑) と推定結果 (紫) が 3 [m] 以上、角度も地図データ (青) とスキャンデータ (赤) が 90 [deg] 程度に大きくずれてマッチングしてしまい位置推定に失敗することがあった。しかし、提案手法を組み込んで人の点群を削減したことにより、同じ位置において位置も角度も大きくずれることなくマッチングできている。Fig. 7 に、サンプリングパラメータを 0.1 から 1.0 まで変化させたときの x-y 座標系における自己位置推定の位置の平均誤差と標準偏差の関係、Fig. 8 に失敗回数との関係を示す。位置の平均誤差と標準偏差は、ロボットの走行中に行われた 63 回分の上部測域センサによる真値と下部測域センサによる位置推定結果の差の合計から求めた。また、失敗回数とは、位置推定の結果が基準となる真値から 1 [m] 以上ずれた回数である。Fig. 7 においてサンプリングのパラメータを 0.6 以上の大きな値にしたとき、MCL のみ (青) では平均誤差が大きくなっているが、提案手法あり (橙色) では全体的に誤差の増加を抑えることができている。人などのアウトライア一点群を除去したため、精度良くマッチングしているパーティクル以外の尤度が低くなったと考えられる。Fig. 8 においても、サンプリングパラメータを 0.6 以上にしたときに提案手法なしのときの失敗回数 (青色) に比べて提案手法ありの時の失敗回数 (橙色) のほうが少なくなっている。この実験結果から、自己位置推定の誤差や失敗回数が小さいときのパラメータ範囲が広がっているため、提案手法によって適切なサンプリングパラメータの設定がしやすくなっていると考えられる。

4.2.2 計測モデルのパラメータ設定に対する影響

MCL の計測モデルとしては、正規分布を基準としたレンジファインダのビームモデルを採用した [9]。この中の合成尤度関数のパラメータの一つに、正規分布の標準偏差を表す hit がある。 hit はセンサの計測誤差を考慮するパラメータで、環境やセンサの特性に合わせて適切に調節する必要がある。また、このパラメータは尤度計算時のマッチングの緩さにも関係し、 hit が大きいと少しずれたマッチングでも尤度が高くなりやすく、逆に小さくすると少しでもマッチングがずれた場合尤度が極端に低くなりやすくなる。計測モデルのパラメータに関しては、最尤推定 [9] などを用いて尤度が最大になるように自動的にパラメータを求める方法もあるが、最尤推定を行うためには適切な hit の初期値を設定する必要がある。厳密なパラメータの微調整を必要とせず、適当に設定したパラメータでも人混みなどで安定した自己位置推定ができることが理想的である。そこで、提案手法の有無によって、 hit を 0.1 から 1.0 の間で変化させたときの、MCL による自己位置推定の誤差や位置推定に失敗する回数の評価を行い、提案手法の計測モデルに関するパラメータ設定への影響を検証した。この実験ではパーティクルのサンプリングパラメータは 0.1 に設定し、パーティクルの個数を 1000 で固定とした。

Fig. 9 の上段に $hit = 0.8$ のときの自己位置推定の結果を、下段にある箇所のマッチングの様子を示す。また、Fig. 9 の左側の図が MCL のみの時の結果、右側の図が提案手法に MCL を組み込んだ時の結果である。MCL のみの場合、真値の位置 (黄緑) に対して推定結果 (紫) が 3 [m] 以上の誤差が生じ、角度も地図データ (青) とスキャンデータ (赤) が重ならずずれの場合が計 63 回の自己位置推定のうち 2 回発生した。一方で MCL に提案手法を組み込むと、人の点群が削除されて角度がずれている場合マッチングの尤度が適切に低くなり、真値付近のパーティクルが採用されやすくなっている。計 63 回の自己位置推定のうち、大きな位置推定の誤差は一度も発生しなかった。

Fig. 10 に、MCL のみの時と MCL に提案手法を組み合わせた時の hit の値に対する x-y 座標系における自己位置推定結果の位置の平均誤差と標準偏差の関係、Fig. 11 に失敗回数との関係を示す。MCL のみの場合、 hit が 0.8 以上になると平均誤差 (青) が増加し位置推定に失敗する回数 (青) も多くなっていた。一方で、提案手法を MCL に組み合わせたときの平均誤差 (橙色) や失敗回数 (橙色) は、MCL のみのときと比べて大きく減らすことができた。この実験結果から、提案手法を組み合わせることで平均誤差や失敗回数が少ないときの hit の値、つまり適切な尤度関数の標準偏差の幅が広がっているため、計測モデルのパラメータの設定がしやすくなっていると考えられる。

5. まとめ

本研究では、実環境下で活躍する自律移動ロボットの実現のために、現状困難とされている人混み中での自己位置推定について取り組んでいる。人混み中での自己位置推定を可能にするために、地図データと同様の形状をしたスキャンデータを人混み中でも得ることを目的とした。そこで、人混み中でも人の隙間から壁が一部ずつ見えること、また屋内の壁に

囲まれた環境下では障害物は手前に存在し壁が最も遠くにあるという位置関係に注目し、スキャンデータ蓄積と遠方点抽出による測域センサのデータ処理方法を提案した。実際に人混み中において提案手法を用いると、ただの 1 回のスキャンデータに比べてアウトライアーである人の点群を削減し、インライアーである壁の点群を増加させたスキャンデータを取得することができた。次に、提案手法の有無によって人混み中での自己位置推定結果や性能がどのように変化するか、ロバスト性が高い自己位置推定手法の一つである MCL を用いて評価実験を行った。提案手法を組み合わせると、自己位置推定の位置の平均誤差や大きく位置推定がずれる失敗回数が MCL のみの時と比べて少なくなり、位置推定の精度が良い時の MCL のサンプリングパラメータやビームモデルの尤度関数に用いられる標準偏差パラメータの範囲が広がる結果となった。このことから、提案手法によって適切に設定する必要がある MCL のパラメータの設定がしやすくなることがわかった。

一方で今回の実験環境下では、MCL のみでも自己位置推定結果の平均誤差が小さいパラメータ範囲が存在していた。今後は、MCL のみのパラメータ調節だけでは自己位置推定の平均誤差や失敗回数が少なくなる人混み環境を実験的に再現し、その環境下で提案手法を組み合わせるとどこまで位置推定できるか実験を行う。また、提案手法によってどの程度の人混みまで自己位置推定が可能になるかの評価も行いたいと考えている。人混みの密度や配置、またはインライアの点群が見えている割合などを基準に、提案手法がどのような条件下まで有効性をもつのか、さらに検証していきたい。

参考文献

- [1] D. Rodriguez-Losada, F. Matia, R. Galan, M. Hernando, J. M. Montero, and J. M. Lucas: "Urbano, an Interactive Mobile Tour-Guide Robot", *Advances in Service Robotics*, Ed. H. Seok. In-Teh, pp. 229–252, 2008.
- [2] Y. Hara, H. Kawata, A. Ohya, S. Yuta: "Mobile Robot Localization and Mapping by Scan Matching using Laser Reflection Intensity of the SOKUIKI Sensor", *Proc. of IECON'06*, 2006.
- [3] 藤井祐介, 大矢晃久, 坪内孝司: "人混み中での移動ロボットの自己位置推定に有効な測域センサデータ処理法の提案", *ROBOMECH2014*, 2014.
- [4] D. Wolf, G. S. Sukhatme: "Online Simultaneous Localization and Mapping in Dynamic Environments", *Proc. of ICRA2004*, 2004.
- [5] T. Masuda, K. Sakae, N. Yokoya: "Registration and Integration of Multiple Range Images for 3-D Model Construction", *Proc. ICPR*, vol. 1, pp. 879–883, 1996.
- [6] 増田 健: "ICP アルゴリズム", 情報処理学会研究報告 *CVIM-168*, no. 23, 2009.
- [7] 日永田 佑介: " L_0 ノルム最小化を利用した動的な混雑環境下に適用可能なリアルタイム SLAM", 画像の認識・理解シンポジウム (*MIRU2011*) 論文集, 2011.
- [8] Dieter Fox, Wolfram Burgard, Sebastian Thrun, Armin B. Cremers: "Position Estimation for Mobile Robots in Dynamic Environments", *J. of Artificial Intelligence Research*, vol. 11, pp. 391–427, 1999.
- [9] Sebastian Thrun, Wolfram Burgard, Dieter Fox, 上田隆一 (訳): 確率ロボティクス, 2015.