

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

日永田 佑介[†] 末永 剛[†] 竹村 憲太郎[†] 高松 淳[†] 小笠原 司[†]

[†] 奈良先端科学技術大学院大学 情報科学研究科 〒630-0192 奈良県生駒市高山町 8 9 1 6 - 5

E-mail: †hieida.yusuke01@is.naist.jp

あらまし 本稿では動的混雑環境下に適用可能な L_0 ノルム最小化によるリアルタイム SLAM を提案する。提案手法は LSH と重点的サンプリングを採用し、大きな計算量を要求する L_0 ノルム最小化の高速化を実現した。本稿で提案した、 L_0 ノルム算出に特化した LSH は、最悪なケースでも計算量を削減することができる。また、バイズ推定を用いた手法より簡便な、MAP 推定を用いても L_0 ノルム最適化では誤差に頑健なため、結果として高速化に貢献している。実際に、提案手法を実装し、動的混雑環境下で実験を行った。既存の L_2 ノルム最小化手法や、その改良手法である M-estimator を用いた手法と比較をし、提案手法の有用性を示した。

キーワード 移動ロボット, SLAM, LSH, 重点的サンプリング, 動的環境

1. はじめに

未知環境下で移動を行う自律移動ロボットは *Simultaneous Localization and Mapping* (SLAM) [1] に関する問題を解決する必要がある。SLAM の解決を行う大部分のシステムは環境が静的であるという仮定の下、時系列順に観測を位置合わせすることによって地図生成を実現している。既存手法である *Iterative Closest Point* (ICP) による手法 [2] とパーティクルフィルタ [3] を用いた手法は、静的な環境下において高い性能を有す [4]。

しかし、この静的な環境であるという仮定は、実環境下ではほとんど成り立たない。動的で混雑した環境下での計測の重ね合わせは、静的な物体（以降インライアとする）、と移動物体や時系列変化を起こす動的な物体（以降アウトライアとする）が同時に存在するため、Fig. 1 のように、既存の SLAM 手法では地図生成ができない、もしくはその精度を著しく低下させてしまう、といった問題が発生する。この様な環境下で、頑健な位置合わせを行うには、アウトライアへの考慮が要求される。

1.1 関連研究

動的環境下に対応した SLAM 手法を大別すると、大きく 2 つのアプローチに分類される。1 つはインライアやアウトライアを特徴抽出によって検出しトラッキングなどをする手法、もう一方のアプローチは、位置合わせの際、アウトライアの尤度を推定し、徐々にそれらの影響を軽減する手法である。

前者のアプローチとしては、Wolf と Sukhatme らが壁による直線や、角など静的環境によく現れるランドマークを使用する手法 [5] を提案している。また Hähnel らは確率的なデータアソシエーションフィルタを採用したサンプリングベースのアウトライア物体の追跡手法を提案

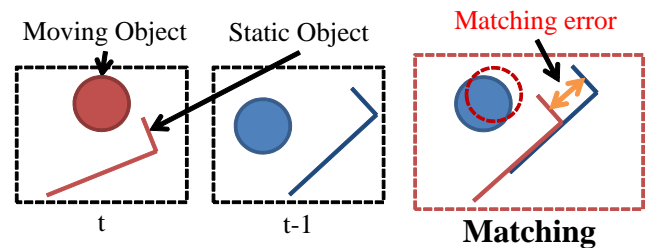


Fig. 1 Erroneous estimation of displacement due to dynamic environment. The circular object moves rightward from time $t-1$ (middle) to time t (left), while the L-shape object is fixed. Alignment only using the L-shape object outputs the correct displacement.

している [6]。しかし、これらの手法の性能は、どれだけ正確にインライアとアウトライアを判別できるのか、という点に強く依存する。

後者のアプローチとしては Hähnel らが *expectation-maximization* (EM) アルゴリズムを利用した手法 [7] を提案している。E-step ではアウトライアの尤度を評価し、M-step ではその尤度に基づいて重み付けしながら、位置合わせを行う。EM アルゴリズムの結果は一般的に初期の推定に強く依存する。そして、これをリアルタイムの推定に用いることは困難である。

アウトライア検出の観点から考えると、M-estimator [8] などのロバスト統計の使用は解決策の 1 つであるといえる。M-estimator を用いた最適化手法は単純であり、かつ EM アルゴリズムよりも高速であるが、初期位置に依存し、局所最小解に陥りやすいという問題点がある。

1.2 提案手法

本稿では、動的環境に適用可能なリアルタイム実行が可能な SLAM システムを提案する。提案手法では、リア

リアルタイムで2次元平面上の距離情報を計測可能なLIDARセンサを用いる。

先行研究で用いられているSLAMシステムが一般的に対応付けの評価指標に L_2 ノルムを用いるのに対し、提案手法では L_0 ノルムを評価指標として用いる。従来、 L_0 ノルムを用いた最適化は離散最適化問題であるため、非常に時間がかかるという傾向があった。しかし、提案手法ではLocality Sensitive Hashing (LSH) [9]と重点的サンプリングを利用する手法を採用することにより、高速な最適化を実現することができる。それにより、リアルタイムでの地図生成を可能にする。

本稿の主な貢献は以下の3つである。

(1) L_0 ノルム最小化によるSLAMが動的環境下において有用であることを実験的に立証する。 L_0 ノルム最小化は、顔認識[10]やインペインティング[11]など様々な手法で有用性を示されている。 L_0 ノルム最小化をSLAMの位置合わせに適用したのは我々が知る限り提案手法が初めてである。

(2) LSH[9]を用いることによって近似的に L_0 ノルム算出を高速化する手法を提案する。 L_2 ノルム最小化と違って、ある距離内に他の点が存在するかどうか、つまり r 近傍点探索を行えば良い。そのため、近傍点がないと判断し早い段階で計算が打ち切れ、近傍点のIDも L_0 ノルムの計算に必要なない。これにより、計算時間を安定させ、各点の位置情報を取り除くことによってメモリ消費を抑えることができる。

(3) L_0 ノルム最小化に重点的サンプリングを利用する手法を提案する。 L_0 ノルムがアウトライアに頑健であるため、過去のステップにおける位置合わせの尤度を用いるベイズ推定は必要ではない。そのため、maximum-a-posteriori (MAP) 推定を利用し、最適解1つのみを探すだけで地図を生成することができる。

提案手法はロバスト推定量を使用する手法[12]と似通っている部分がある。しかし、既存手法が解決できなかった問題についても、(2)及び(3)によって、提案手法は解決することができる。

2. L_0 ノルム最小化によるリアルタイムSLAM

2.1 ノルムの最小化

ノルム (norm) というのは、距離を意味し、本稿では位置合わせの評価指標に利用する。一般的によく用いられているのは、距離の2乗和である L_2 ノルムである。2つの点群 $\{p_i\}$ と $\{q_j\}$ を仮定する。これらが同じ環境下における微小に異なった位置からの測定の場合、点群 $\{p_i\}$ と $\{q_j\}$ のそれぞれの点に対して、互いに座標値は異なるが対応する点が存在することになる。ロボットの位置推定はこの2つの点群を重ね合わせることによって推定される。各点の対応が与えられており、環境が静的であれば、その場合のノイズは等方性の正規分布に基づいて発生するとみなせる。そのため位置推定の際、式(1)の

評価関数 $E(R, t)$ を最小化することで、最適解とすることができる[1]。

$$E(R, t) = \sum_{i=1}^n f(Rp_i + t, \{q_j\}) \quad (1)$$

ここで、 n が点群 $\{p_i\}$ の点の総数であり、 R は 2×2 の行列で t は2次元のベクトルであり、それぞれロボットの平面上の移動における回転と平行移動を表す。

点群 $\{p_i\}$ のある点 p_i に対応する点群 $\{q_j\}$ の点 q_j とすると、関数 f は以下のように定義される。

$$f(p_i, \{q_j\}) = \|p_i - q_j\|_2$$

この関数 f は p_i と q_j とのユークリッド距離を表し、これを L_2 ノルムと呼ぶ。関数 $E(R, t)$ が連続的であり、制御入力やオドメトリから近似解を得ることができるので、解の最小化を行う際に通常勾配ベースの最適化がよく採用される[12]。

しかし、一般的な実環境の場合、この点群の対応は不明である。そこで、近似解を得るために、ICP[2]では最近傍点を対応点と見なす。この対応点探索と最小化は、解に収束するまで繰り返し行われる。

2.2 L_0 ノルムを用いたSLAM

2.1節で述べたように L_2 ノルムは計測ノイズが正規分布に基づいて発生するという仮定に依存する。しかし、アウトライアによるノイズはそのような分布に基づいて発生し得ない[1]。そのため、動的環境におけるSLAMにおいて L_2 ノルムを使用することは理論的に問題があるといえる。

そこで、近年、コンピュータビジョンにおいて盛んに研究されている、アウトライアに対しロバストな手法として、 L_0 ノルム最小化[13]を用いる手法を提案する。 L_0 ノルムは最小化に時間がかかる傾向がある。そこで、本稿ではSLAMのために設計したLSHと、重点的サンプリングに基づいた最適化を提案し、 L_0 ノルムの最小化の高速化を実現する。

2.3 L_0 ノルムの定義

L_0 ノルムは2つの点群において、対応があると判別された点の数として、定義される。ここにおける対応は、近傍点の存在の有無に基づいている。ここで2つの点群、 $\{p_i\}$ と $\{q_j\}$ において、式(1)同様、評価関数 $E(R, t)$ によって L_0 ノルムは定義されるが、 f は異なり以下のようになる。

$$f(p_i, \{q_j\}) = \begin{cases} 0 & (\exists j, \|p_i - q_j\| \leq \epsilon) \\ 1 & (\text{otherwise}) \end{cases} \quad (2)$$

距離の閾値である ϵ は、計測誤差に基づいて設定される任意の実数である。 L_0 ノルムの算出は、探索対象の点

群において半径 ϵ に点が存在するかどうかを決定する際、明白な点の対応を必要としない。文献 [9] で用いられている用語に基づき、本稿ではこれらの点を r 近傍 (r -near neighbors) と呼ぶ。

L_0 ノルムに基づく評価関数は非凸状であり、連続的ではない。LIDAR の性質上、センサに近い点群と遠い点群の密度が異なるため、セグメント化し点群密度の均一化を行う。そして、任意の姿勢に回転並行移動を行い、最近傍点が半径 ϵ 以内に存在するか否かを求め、 L_0 ノルムを算出する。最近傍点探索に brute-force 法、及び kd-tree 法を用いた場合、 L_0 ノルム算出の計算時間は、 L_2 ノルム算出と比べ、大差ない。しかし、 L_0 ノルム最小化を行う場合、離散最適化問題に陥ってしまうため、解像度と探索範囲を設定し、総探索することによって最適解を求められることが多く、一般的な L_2 ノルム最小化手法に比べ、大きな時間がかかってしまう。そのため、この最適化手法の高速化を実現する必要がある。

2.4 Locality Sensitive Hashing による高速化

一般的に r 近傍はクエリ点に対し最近傍点を求めて、その点とのユークリッド距離を算出する、という 2 つの手順で決定される。brute-force 法では最近傍点を探索する場合、探索対象の点の数を m とすると (本稿では $\{q_j\}$ の点の数である)、探索コストは $O(m)$ かかる。この計算を kd-tree 法 [14] や Voronoi 図 [15] 等を用い高速化した場合、探索コストは $O(\log m)$ となる。

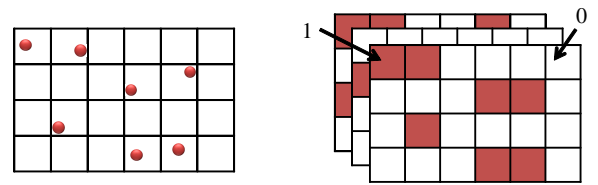
しかし、本稿では r 近傍探索を再検討し、計算時間が $O(1)$ である Locality Sensitive Hashing (LSH) [9] を用いる手法を提案する。LSH は、ベクトルデータをハッシュ値に変換するアルゴリズムである。ベクトルデータの距離が近ければ近いほど、確率的に同じハッシュ値を取る傾向のあるハッシュ関数を用い、探索範囲を制限することで高速化をはかる。

LSH による検索は、複数のハッシュテーブルを用意し、ランダムに決定されるパラメータを持つ複数のハッシュ関数を用いることによって、ハッシュ関数による検索範囲の過剰な絞り込みの可能性を緩和させる。一般的に、ハッシュ関数は以下の式で定義される。

$$h(q) = \left\lfloor \frac{a \cdot q + b}{w} \right\rfloor \quad (3)$$

ここで w はハッシュ関数によって設定されるハッシュグリッドのサイズ、 a は安定分布によって、 b は一様分布によって 0 から w の範囲でランダムに決定されるパラメータである。

LSH は一般的に最近傍点探索に用いられるが、 r 近傍探索に LSH を適用する場合、2 つの利点がある。すべての探索点がクエリ点と異なったハッシュ値である場合、最近傍点探索においてはクエリ点にもっとも近い点を探索し続ける必要がある。しかし、 r 近傍探索の場合、 r 近



(1) Creating hash tables

(2) Accessing hash bin

Fig. 2 Several Boolean-array hash tables are generated. Each bin represents the existence of r -near neighbors, and thus the existence is decided by accessing the corresponding bin in $O(1)$.

傍点は存在しないと判断し、計算を終了することができる。また提案する SLAM 手法の場合、 r 近傍点がどの点であるかということを考慮する必要は無く、 r 近傍点の有無のみを考慮するだけでよい。

本稿では LSH を用いた手法を SLAM 問題に改良し適用した。本稿で設定した問題では、2 次元平面上に測定点が存在するため、以下の 2 対のハッシュ関数を用いる。1 つの関数 H_x は x 軸の写像であり、もう一方の H_y は y 軸の写像を表す。

$$H_x(q) = \frac{aq_x + b_x}{\epsilon}, \quad H_y(q) = \frac{aq_y + b_y}{\epsilon} \quad (4)$$

パラメータである a , b_x と b_y は一般的な LSH と同じくランダムに求められる。クエリ点のハッシュ値を評価関数を用いて計算する前に、探索点群のすべての点のハッシュ値が計算され、Fig. 2 のように対応する 2 次元のハッシュテーブルのビンが書き換えられていく。そして、クエリ点の r 近傍点の有無はクエリ点のハッシュ値に対応するハッシュテーブルの値を参照することによって探索される。

2.5 重点的サンプリングを用いた最小化手法

SLAM では位置推定に 2 つの異なるタイプの情報を利用することができる。1 つは車輪などの回転から検出されたオドメトリ、もしくは制御入力から得られた事前確率の確率密度関数であり、もう一方はセンサを用いて得られた計測に対する評価関数である。評価関数を最小にすることが、システムの目的であり、それを最小化する際、事前確率は非常に大きな役割を持っている。

提案手法では Fig. 3 に示すように、重点的サンプリング (importance sampling) と粗密探索に基づいた探索により、評価関数の最小化を行っている。

(1) 位置の事前確率に基づき、ランダムに n_1 パターンの位置姿勢のサンプル A を生成する。

(2) 各サンプルの重み w_i を以下の式に基づいて計算する。

$$w_i = \exp\left(\frac{m - E(R, t)}{k}\right) \quad (5)$$

ここでパラメータ k は評価関数においてどれだけ重みを

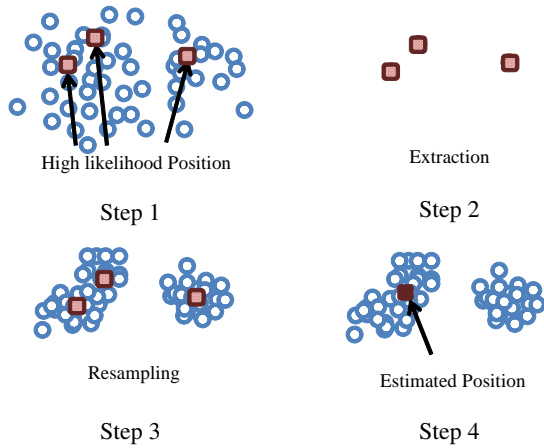


Fig. 3 Minimization by importance sampling and coarse-to-fine search. By gradually changing the distribution σ of sampling range, the samples converge on the minimum in the L_0 -norm.

影響させるか、といった度合いを表している。

(3) 計算された重みに基づいて、サンプル A からランダムに n_2 パターンの位置姿勢のサンプル B を抽出する。

(4) $(\sigma_x, \sigma_y, \sigma_\theta)$ の正規分布に基づいて、上記で生成したサンプル B の周囲に n_3 パターンの位置姿勢を生成する。サンプリングされる位置姿勢の数は各パーティクルの重みに基づいている。

(5) (2) から (4) を数回繰り返す。

提案手法における最小化は文献 [3] のような、ベイズ推定ではなく *maximum-a-posteriori* (MAP) 推定のように、最適解のみを次のステップの最適化に用いる。前者のような推定手法は、測定の信頼性がアウトライアによって一時的に乏しくなってしまう L_2 ノルムを用いた位置推定の頑健性を高めるために用いられる。しかし、 L_0 ノルムはそもそもがアウトライアに対して頑健である。そのため、 L_0 ノルムを用いた SLAM では、MAP 推定で十分であるといえる。結果として、最小化を加速でき、また探索するサンプル数の抑制にも繋がる。

3. 実験

3.1 実験環境

LIDAR LMS100 (Fig. 4 (a)) は角度方向にそって等間隔に距離を計測する。これを電動車いす型移動ロボット (Fig. 4 (b)) に取り付ける。計測時間と推定精度を考慮し、点群の重ね合わせの際、ローテーション処理を行う計測点 $\{p_i\}$ はセグメンテーションされ、セグメント内では点群が一定間隔で分布されるよう間引き処理を行う。他方、計測点 $\{q_j\}$ はセグメント内の隣り合う点の距離が式 (2) で設定したパラメータである閾値 ϵ 以下になるよう、点を補間する。この節の実験では、LIDAR の計測精度に基づいて ϵ は $1[cm]$ に設定した (Table 1 を参照)。なお LSH のハッシュテーブルは 5 つ用いた。



(a) LIDAR: SICK LMS100 (b) Mobile robot

Fig. 4 Equipment used in this experiment.

Table. 1 Specification of SICK LMS100.

| | |
|--------------------|-----------|
| Viewing angle | 270 [deg] |
| Angular resolution | 0.5 [deg] |
| Accuracy in depth | 1.2[cm] |
| Maximum depth | 20[m] |
| Frequency | 30 [Hz] |



(a) Appearance (b) Experimental environment

Fig. 5 Experimental environment. (a) Appearance of the inside of the building. (b) Actual experimental environment crowded by people.

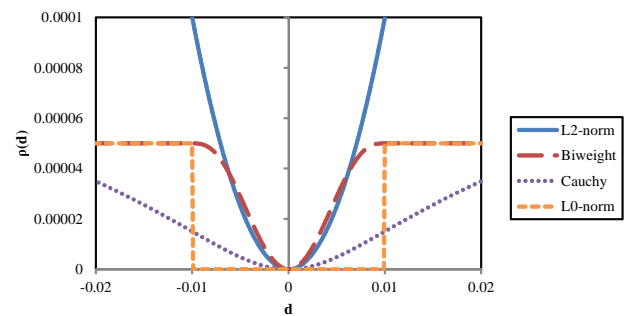


Fig. 6 The relationship between the distance d and the function $\rho(e)$.

本稿では主に 3 種類の実験を行った。前述したとおり、提案手法はロバスト推定を用いた手法と関連性が強い。そのため、最初の実験では ICP [16] や M-estimator を導入した ICP [12] と比較することによって、動的な環境下における L_0 ノルムの有用性について検討を行った。次の実験では計算時間とその精度に関して LSH の有効性について検討を行う。さらに 3 番目の実験では実環境で実験を行い、提案手法の有用性を示す。

3.2 L_0 ノルムの有用性の検証

Fig. 5 に示すように、幅が 5 [m]、長さが 20 [m] の環

境で実験を行った。25人程度が、同時にロボットの周囲を移動した。ロボットはその間環境内で停止し、環境を測定し続ける。このような実験環境でロボットの位置推定を行った場合、 x 、 y と θ をロボットの x 軸方向、 y 軸方向と回転方向の移動量とすれば、この (x, y, θ) はすべて0になるはずである。

初期位置を考慮せずに、各手法に対するアウトライアの影響について調べるために、本実験では総当たり法を用いることで最適解の探索を行った。探索範囲 x 、 y 、 θ は $[-5[cm], 5[cm]]$ 、 $[-5[cm], 5[cm]]$ 、 $[-0.3 [rad], 0.3 [rad]]$ とし、探索時の解像度は、 x 軸方向、 y 軸方向は $0.5[cm]$ 、 $0.5[cm]$ 、回転方向は $0.01 [rad]$ ($\cong 0.57 [deg]$)とし、合計24,000通りをサンプルし探索した。

本節における実験では、M-estimatorで用いられる関数 $\rho(d)$ としてCauchy関数と、Biweight関数を用いた。式(1)における $f(Rp_i + t, \{q_j\})$ をM-estimatorにより設定される $\rho(d)$ の d に置き換えることで、実装を行った。Fig. 6に今回M-estimatorで用いた関数と L_2 ノルム、 L_0 ノルムのグラフが描かれている。

Cauchy関数は式(6)のように定義される。

$$\rho(d) = \frac{C^2}{2} \log \left(1 + \left(\frac{d}{C} \right)^2 \right) \quad (6)$$

ここで、 C は半値反復を決定するパラメータであり、正の実数である。

一方、Biweight関数は式(7)で示され、これは閾値の設定によって、 L_0 ノルムのようにアウトライアの影響を完全に除去することができる。Fig. 6を見ると、Biweightのグラフが $|d| \geq 0.01$ のとき、平坦になっているのがわかる。

$$\rho(d) = \begin{cases} \frac{B^2}{2} & (d \geq B) \\ \frac{B^2}{2} \left(1 - \left(1 - \left(\frac{d}{B} \right)^2 \right)^3 \right) & (d < B) \end{cases} \quad (7)$$

ここで B は閾値を決定するパラメータであり、正の実数である。なお、本実験では B と C はセンサの計測誤差を考慮し、 $1 [cm]$ とした。

上記で説明した通り、ロボットの推定移動量がすべて0であるべきなので、その値を真値とし、平均二乗誤差 (root mean squared error (RMSE)) を比較することで、定量的に各手法の精度を比較することができる。Table 2にこの結果を示す。 L_0 ノルムを用いた手法とBiweight関数を用いた手法は他の2つの手法に比べ、高い精度で位置推定を行っていることがわかる。Cauchy関数を用いた手法は L_2 ノルムを用いた位置推定法と比べ、高い精度だが、他の2つの手法と比べるとそれほど正確ではない。この結果により、 L_0 ノルムを用いることによって、事前に設定した閾値に基づいてアウトライアによる影響が除去されていることがわかり、 L_0 ノルムを用いた手法が有効であることがわかった。また、Biweight関数を用

Table. 2 RMSE of the estimated localization: L_2 -norm, M-estimator (Cauchy and Biweight functions) and L_0 -norm.

| Method | RMSE | | |
|-------------|----------|----------|--------------|
| | $x [cm]$ | $y [cm]$ | $\phi [rad]$ |
| L_2 -norm | 8.93 | 8.06 | 0.0497 |
| Cauchy | 1.54 | 0.0039 | 0 |
| Biweight | 0.579 | 0 | 0 |
| L_0 -norm | 0.576 | 0 | 0 |

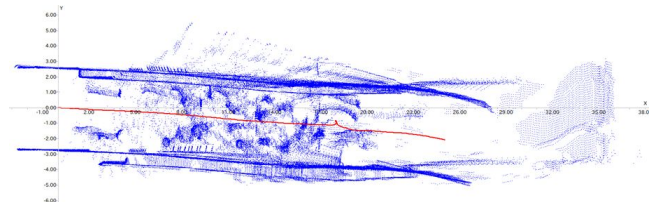


Fig. 7 The map estimated by the L_2 -norm minimization. The red line indicates robot's displacement and the blue points indicate measured points on the global coordinates.

いた手法についても同様に、性能が高いことがいえる。しかしこの結果は、精度が高かったこの2つの手法について、点群を重ね合わせする際に、同じく離散最適化問題に陥ってしまうことを示している。

Fig. 7からFig. 10は、 L_2 ノルム最小化、Cauchy関数、Biweight関数、 L_0 ノルム最小化 (Fig. 10) の4つの手法を用いて、それぞれ推定された地図である。各地図は、位置推定によってセンサによるすべての計測を重ね合わせるによって構成される。赤い線はロボットの推定移動軌跡を示す。しかし、本実験では移動ロボットは動かししていないため、本来現れるべきでない。外側の青い境界は建物の静的な壁を示し、その内側の点は移動している人を表す。地図が適切に再構成されている場合、外側の境界はぶれなく重ね合わされるはずである。これらの図は L_0 ノルムが動的環境下において有効であることを示している。

3.3 LSHの有用性の検証

本実験では提案したLSHによる高速化と、brute-force法とkd-tree法[17]の比較を行う。環境としては3.2節の実験と同じであるが、本実験では実際に移動ロボットが動いている環境で実験を行う。

Table 3に平均計算時間、1 [sec]あたりで L_0 ノルム算出によって計算されたフレーム数、また算出された L_0 ノルムのRMSEを示す。ここで、brute-force法とkd-tree法は、LSHを用いた近似近傍点探索と違って、常に正しい解を出すため、RMSEを算出する際の L_0 ノルムの真値とした。計算時間は、5倍ほどkd-tree法よりも速いが、精度はやや劣っている。RMSEを見ると、 L_0 ノルムと真値とのRMSEは14.4点になっている。

L_0 ノルムを近似的に算出することによる地図精度の

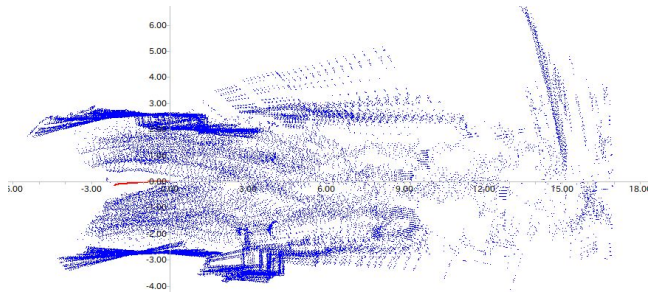


Fig. 8 The map estimated by the Cauchy function.

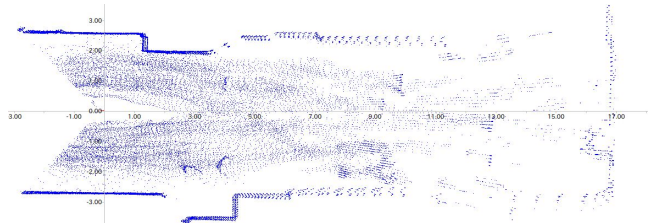


Fig. 9 The map estimated by the Biweight function.

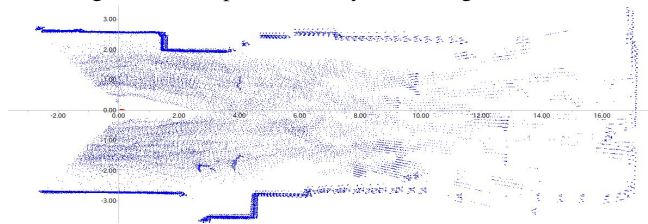


Fig. 10 The map estimated by the L_0 -norm minimization.

Table. 3 Comparison among three kinds of r -near neighbor search: Brute-force, kd-tree and LSH.

| method | time [s] | # of localizations per second | RMSE of L_0 -norm [points] |
|-------------|----------|----------------------------------|---------------------------------|
| brute-force | 126 | 190 | - |
| kd-tree | 24 | 1000 | - |
| LSH | 5.4 | 4436 | 14.4 |

劣化の有無を評価するために、実際に LSH を用いて地図生成を行う。Fig. 11 に静的環境下で計測したデータを利用し、RBPF-SLAM [18] によって生成した地図を示す。この地図を真値と見なす。Fig. 12 に同じ計測点において LSH を用いた L_0 ノルム最小化を利用し生成した地図を示す。ただし、この地図は動的環境下で生成されている。この環境は Fig. 13 に示すように、一般的な L_2 ノルム最小化手法では地図生成ができないほど動的である。LSH は L_0 ノルムを近似的に算出しているが、その誤差は目視で判断する限り、許容範囲内に収まっている。

3.4 実際の使用における検証

本実験では提案手法である LSH による近似 L_0 ノルム算出法と重点的サンプリングによる L_0 ノルム最小化手法を用いて、実際に地図生成を行う。位置推定は、Intel Core2Duo T7500 (2.2[GHz]) と 2[GB] メモリを搭載した PC を用い、約 5[fps] の速度での推定が可能である。また、各パラメータは $n_1 = 300$, $n_2 = 20$, $n_3 = 15$, $k =$

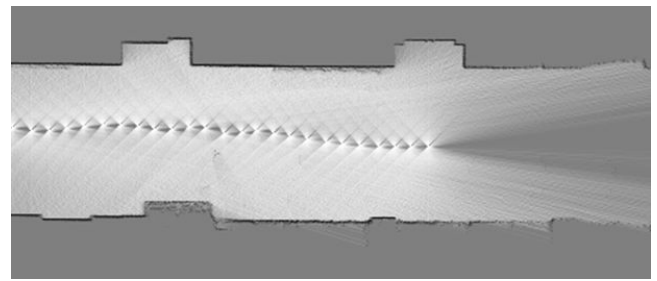


Fig. 11 The map estimated by the RBPF-SLAM method [18] under the environment without moving objects.

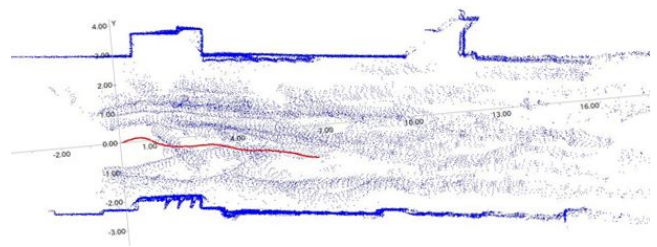


Fig. 12 The map estimated by L_0 -norm under the same environment as Fig. 11, but including moving objects.

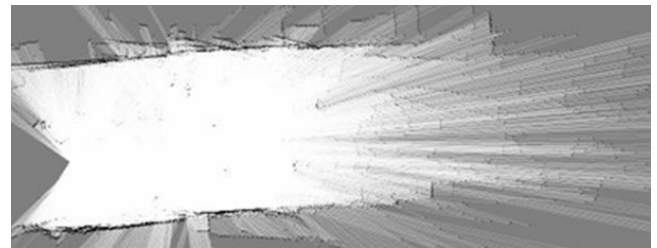


Fig. 13 The map made by L_2 -norm under a dynamic environment.

20.0 と設定している。Steps (2) から (4) は 2 回繰り返した。つまり 1 時間ステップあたり 900 回 L_0 ノルムを評価することにより最適化を行っている。その際、第 1 ステップの分散は、 $(\sigma_x, \sigma_y, \sigma_\theta) = (0.03, 0.03, 0.025)$ と設定し、第 2 ステップの分散は $(\sigma_x, \sigma_y, \sigma_\theta) = (0.01, 0.01, 0.01)$ とした。

最初の実験は、約 50[m] × 30[m] の広さの屋内環境で行った。25 人程の人が Fig. 14 に示されるようにアウトライアとしてロボットの周りを移動した。300 フレームの計測を利用した。Fig. 15 に推定した点地図を示す。また Fig. 16 に占有格子地図を示す。これは SLAM のオープンソースライブラリである *Mobile Robot Programming Toolkit* (MRPT) [19] による占有格子地図生成アルゴリズムと提案手法を用いて生成したものである。Fig. 15 から、アウトライアが除去できている様子がわかる。なお、ロボット移動時に、位置推定のみではなく、地図生成もリアルタイムで行っている。生成した地図の精度を評価するためにこの建物の間取り図を重ねた。間取り図の橙色と重なっている部分からも、十分に精度の良い地図が生成できることがわかる。

次の実験では屋外環境と屋内環境が混在する環境で地図生成を行った。Fig. 17 に提案手法で生成した地図を示

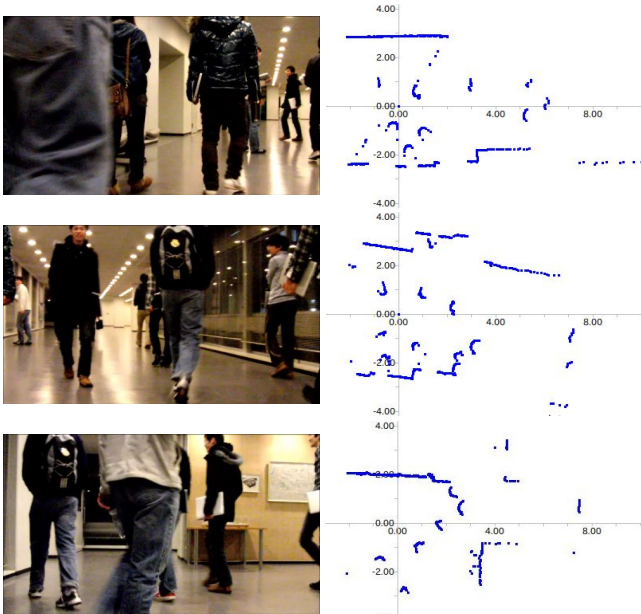


Fig. 14 Snapshots of images obtained from digital camera (left column) and SICK LMS100 (right column) installed in the robot.

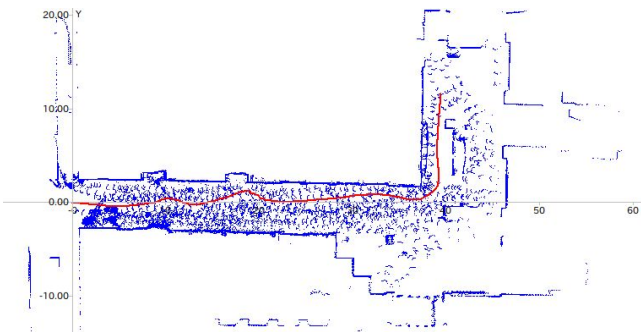


Fig. 15 The points map estimated by the L_0 -norm minimization.

し、また、RBPF-SLAMを用いて同じ環境で生成した地図を Fig. 18 に示す。実験で用いた環境は約 $140[m] \times 80[m]$ の広さを持っている。今回は地図を生成するために 700 フレームの計測を用いた。また、地図の精度を評価するために、Google Maps [20] からダウンロードした俯瞰写真を用い、生成した地図に重ねた。推定された経路は、実際に移動ロボットが通過した経路に沿っていた。対して、RBPF-SLAM で生成した地図は大きくずれており、最終的には $60[m]$ 程の推定誤差になった。また、計算時間も提案手法に比べ 10 倍以上かかった。

4. 考 察

4.1 L_0 ノルムの優位性

L_0 ノルムと Biweight 関数の両方には、同様の利点と欠点が存在する。利点は B あるいは ϵ を小さくすることで、アウトライアに対して頑健になるという点であり、欠点は ϵ を小さくしすぎることによって、両者とも分散最適化問題に陥ってしまい実時間での計算が困難になる点である。しかし、Biweight 関数と異なり、LSH は r 近傍

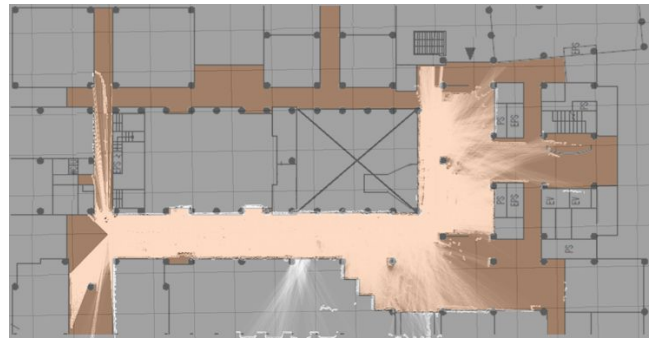


Fig. 16 The occupancy grid map obtained by the open SLAM software *MRPT* [19] and the proposed method. We overlay the floor plan of this building to evaluate the accuracy.



Fig. 17 The map estimated by the proposed method. We overlay the corresponding aerial image from the *Google Maps* [20] to evaluate the accuracy.



Fig. 18 The map estimated by the RBPF-SLAM method.

探索を用い、点の ID を無視することによって L_0 ノルム算出を高速化することができる。また、 L_0 ノルムを用いることで、近傍点の位置座標など細かいパラメータを無視するが、精度が維持されることが本稿の実験で明らかになっている。そのため提案手法は有効であるといえる。本節で提案した手法とさらに Biweight 法による M-estimator をハイブリッド化することによって、計算時間は増大しリアルタイムでの実行は困難になるが、精度は向上する可能性はある。

4.2 地図生成の限界

すべての点群を速度に基づいて分類し、その中でインライアが優勢であれば、提案手法は問題なく動作する。しかし、ロボットが複数人に完全に取り囲まれてしまうケースにおいては、地図生成の失敗が起こる傾向がある。



(a) The map estimated

(b) Scene

Fig. 19 The map estimated from the data when the robot could not get inlier data. Compared to the result in Fig. 16, due to the severe situation, several errors are shown.

Fig. 19 はそのようなケースで生成した地図である。地図の左端の末端で、推定誤差が発生していることがわかる。このケースでは壁を全く検出することができなかったため、静止する物体の点群よりも動的な物体群のほうが優位になってしまった。また、既存手法と同様、インライアの形状特徴が著しく乏しい場合における位置合わせも、困難であると言える。より動的環境に対して頑健にするためには、このようなケースでも正確な位置合わせが行えるよう実装を検討する必要がある。EM アルゴリズム [7] と L_0 ノルム最小化の組み合わせは可能な解決策の 1 つである可能性があるが、計算速度を犠牲にするという問題点がある。

また提案手法ではフレーム間の位置合わせを行っているため、環状の形状をもつ場所の地図生成を行う場合、微小な誤差が累積し、地図に大きなずれが生じる場合がある。それを軽減するためには *Loop-closing* [22] 等のマッチング手法の適用を検討する必要がある。

5. さいごに

本稿では、動的環境下におけるリアルタイム SLAM を提案した。一般的な L_2 ノルム最小化手法やロバスト推定手法と精度を比較をし、その性能について検討を行った。そして、実際にいくつかの動的環境下に提案手法を適用し、実際に地図を生成することで、提案手法の有用性を示した。

本稿で提案した手法はシンプルかつ汎用性が高いため、さまざまなシステムへの応用が可能であると考えられる。今後は、提案手法の頑健性と汎用性の高さを生かし、環境モデルの取得を必要とする様々なアプリケーションへの適用が望まれる。

また提案手法は理論上、3 次元の環境モデルにも適用可能である。ただし、提案手法を 3 次元に適用する場合、自由度が 3 つ増えるため、最適解導出に用いるパーティクルの数が増大し、位置推定に計算時間を要する。リアルタイム処理を行うためには、計算を並列化するなどして、最適化をより高速にする必要があるだろう。

文 献

- [1] S. Thrun, W. Burgard, and D. Fox, *Probabilistic robotics*, The MIT Press, 2005.
- [2] P. J. Besl and N. D. McKay, "A method for registration of

3-D shapes," *IEEE Transactions on Pattern Analysis and Machine Intelligence(PAMI)*, Vol. 14, No. 2, pp. 239–256, 1992.

- [3] M. Isard and A. Blake, "CONDENSATION-conditional density propagation for vision tracking," *Int. Journal of Computer Vision (IJCV)*, Vol. 29, No. 1, pp. 5–28, 1998.
- [4] S. Thrun, "Simultaneous mapping and localization with sparse extended information filters: Theory and initial results," *Algorithmic Foundations of Robotics V*, Vol. 7, pp. 363–380, 2003.
- [5] D. F. Wolf and G. S. Sukhatme, "Mobile robot simultaneous localization and mapping in dynamic environments," *Autonomous Robots*, Vol. 19, No. 1, pp. 53–65, 2005.
- [6] D. Hähnel, D. Schulz, and W. Burgard, "Map building with mobile robots in populated environment," In *Proc. of Int. Conf. on Intelligent Robots and Systems (IROS)*, pp. 496–501 vol.1, EPFL, Switzerland, Sept. 2002.
- [7] D. Hähnel, R. Triebel, W. Burgard, and S. Thrun, "Map building with mobile robots in dynamic environments," In *Proc. of Int. Conf. on Robotics and Automation (ICRA)*, pp. 1557–1563 vol.2, Taipei, Taiwan, Sept. 2003.
- [8] P. J. Huber, *Robust statistics*. Wiley-Interscience, 1981.
- [9] A. Andoni, M. Datar, N. Immorlica, P. Indyk, and V. Mirrokni, "Locality-sensitive hashing scheme based on p -stable distributions, nearest neighbor methods in learning and vision," MIT Press, 2006.
- [10] J. Wright, A. Yang, A. Ganesh, S. Sastry, and Y. Ma, "Robust face recognition via sparse representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence(PAMI)*, Vol. 31, No. 2, pp. 210–227, 2009.
- [11] L. Mancera and J. Portilla, "L0-norm-based sparse representation through alternate projections," In *Proc. of Int. Conf. on Image Processing (ICIP)*, pp. 2089–2092, Atlanta, USA, Jan. 2006.
- [12] 阪野貴彦, 池内克史, "移動型レンジセンサによる歪み補正 ICP アルゴリズム," In *Meeting on Image Recognition & Understanding*, IS3-36, Sendai, Japan, July 2006.
- [13] G. C. Calafiore, "On 3-D point set matching with MAE and SAE cost criteria," *IEEE Transactions on Systems, Man, and Cybernetics, Part A*, Vol. 38, No. 2, pp. 443–450, 2006.
- [14] J. L. Bentley, "Multidimensional binary search trees used for associative searching," *Communications of the ACM*, Vol. 18, pp. 509–517, 1975.
- [15] F. Aurenhammer, "Voronoi diagrams: a survey of a fundamental geometric data structure," *ACM Computing Surveys*, Vol. 23, pp. 345–405, Sept. 1991.
- [16] S. Kaneko, T. Kondo, A. Miyamoto, and S. Igarashi, "Robust ICP registration algorithm extended by m-estimation," *Journal of the Japan Society of Precision Engineering*, Vol. 67, No. 8, pp. 1276–1280, 2001.
- [17] M. Greenspan and M. Yurick, "Approximate K-D tree search for efficient ICP," In *Proc. of Int. Conf. on 3-D Digital Imaging and Modeling (3DIM)*, pp. 442–448, Banff, Alberta, Canada, Oct. 2003.
- [18] G. Grisetti, C. Stachniss, and W. Burgard, "Improved techniques for grid mapping with rao-blackwellized particle filters," *IEEE Transactions on Robotics*, Vol. 23, pp. 34–46, 2007.
- [19] *Mobile Robot Programming Toolkit*. <http://www.mrpt.org/>.
- [20] *Google Maps*. <http://maps.google.co.jp/>.
- [21] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, "Numerical Recipes in C," Cambridge University Press, Cambridge, UK, 2nd edition, 1992.
- [22] C. Stachniss and W. Burgard, "Exploration with active loop-closing for FastSLAM," In *Proc. of Int. Conf. on Intelligent Robots and Systems (IROS)*, pp. 1505–1510 vol.2, Sendai, Japan, Sep. 2004.