観測度数をもつ占有格子地図による
長時間地図構築のためのデータ削減手法友納 正裕^{*1}Data reduction for long-term mapping
using occupancy grid map with observation frequencyMasahiro TOMONO^{*1}^{*1} Chiba Institute of Technology

2-17-1 Tsudanuma, Narashino, Chiba 275-0016, Japan

Received: 31 January 2018; Revised: 5 April 2018; Accepted: 8 May 2018

Abstract

This paper presents a method of data reduction for long-term robotic mapping using a 2D laser scanner. The proposed method introduces observation frequency into the occupancy grid map, and calculates the importance of each scan through the ray-casting algorithm. The importance of a scan is high if the scan observes many occupancy cells which are new in terms of the observation frequency. Also, the method calculates the coverage of a subset of scans to examine how the subset covers the map well. Then, the method determines the threshold of the scan importance according to the designated coverage, and removes the scans which are less important than the threshold. The reduced laser scans improve computation time and memory consumption for pose adjustment and map reconstruction in loop closure. Experiments using real-world data show the proposed method effectively reduces data size and computation time in robotic mapping.

Keywords : Mapping, SLAM, Occupancy grid map, Long-term autonomy, Map maintenance

1. 緒 言

ロボティクスにおいて地図構築は重要な研究分野であり、SLAM (Simultaneous Localization and Mapping) を中心に盛んに研究が行われている。地図構築における課題の1つに長時間活動がある。これは、ロボットが環境中を長時間動き回って、(作業の最中であっても) 地図を作り続けるという問題である (Konolige and Bowman, 2009)。環境が広大であれば、地図構築に長時間かかるのは当然であるが、ここでは、オフィスや家庭など比較的小規模な環境で同じ場所を何度も通るような長時間活動を考える。この場合、地図を作り続ける目的は、環境の変化を反映して地図を更新することであり、実時間で更新する場合や毎日定期的に更新する場合など、種々の形態が考えられる。

地図を長時間作り続ける際の問題として、処理時間とメモリ使用量の増加がある。後述のように、地図形状の歪みを減らして一貫性を保つには、ループ閉じ込みを行う必要がある。ループ閉じ込みをすると地図を作り直すのが、これには過去のデータをすべて用いるため、処理時間とメモリを消費する。

もう1つの問題として、地図更新がある。たとえば、同じ環境で毎日作業をするロボットは、日々のセンサデータから地図を更新することが望まれる。その際、同じ場所を何度も通るので、ループ閉じ込みは頻発する。しか

し、上記のように、ループ閉じ込みの後に地図を作り直すには過去のセンサデータが必要になるが、日々蓄積したセンサデータをすべて用いるのは量が多すぎて現実的でない。これに対処するには、有用な新しいセンサデータだけを選別する必要がある。

本稿では、レーザスキャナを用いた屋内の 2D 地図構築において、地図を長時間作り続けることができるように、センサからの入力データおよび地図の内部データを削減する手法を提案する。提案手法は、占有格子地図に観測度数を導入し、それにより情報が古くなって有用性の少なくなったセンサデータを選別して削除する。そして、センサデータの削減に合わせて、ループ閉じ込みで用いる内部データであるポーズグラフを縮約する。これらにより、ループ閉じ込みとその後の地図再構築に要する処理時間とメモリ使用量が削減される。

本稿では地図更新そのものは対象としないが、提案手法を用いてロボットが活動し続けると、有用な新しいセンサデータだけが残っていくので、地図更新への発展がしやすいと期待される。

以下、2 章で問題設定とそれに対するアプローチを述べ、3 章で占有格子地図の観測度数を用いたスキャンの選別手法を説明する。4 章で地図と処理時間の削減方法を述べ、5 章で実験結果を報告し、6 章で結言する。

2. 問題設定とアプローチ

2-1 前提

まず、本稿で用いる用語を説明する。ロボット位置は $(x, y, \theta)^T$ であり方位 θ を含む。レーザスキャナの 1 周期分のデータをスキャンと呼ぶ。本稿のシステムはスキャン単位で処理を行う。ループとはロボットが同じ場所を通る周回経路であり、ループを検出して地図の歪みを解消する処理をループ閉じ込みと呼ぶ。

本稿で対象とするシステムは、前稿 (友納, 2017) で提案した 2D レーザスキャナを用いた 2D-SLAM である。これは、ICP (Iterative Closest Points) (Besl and McKay, 1992) に基づくスキャンマッチングでロボット軌跡と点群地図を生成する。そして、ロボット軌跡のループを検出し、ポーズ調整 (Irie and Tomono, 2017) によりループ閉じ込みを行う。

ポーズ調整はロボット位置の拘束関係を表すポーズグラフを用いてロボット軌跡を修正する技術である (Olson et al., 2006)(Konolige et al., 2010)。ポーズグラフは、ロボット位置を表すノードと、ノード間の相対位置を表すアークから構成され、ロボットの推定位置と計測値とのずれを表すことができる。ポーズ調整はこのずれを最小化することで、ロボット軌跡の歪みを修正する。

本システムで生成する地図は、点群地図と占有格子地図である。点群地図は、マップ点と呼ばれる 2D 点から構成される (友納, 2017)。マップ点は、レーザスキャナで計測した点 (スキャン点) の集合の代表点であり、空間的および時間的な局所範囲でスキャン点をクラスタリングして生成する。マップ点の範囲を時間的に区切るのは、ループを一周して同じ場所に戻ったときに、一周前のマップ点と現在のマップ点を分離して、地図を修正できるようにするためである。点群地図は、ICP やループ検出など SLAM の処理本体で用いられる。占有格子地図は、本システムでは SLAM 処理には使われないが、障害物の表現に優れ、経路計画などに有用なので、SLAM で得たロボット位置を用いて、SLAM とは別に生成される。

2-2 長時間活動における問題

このような前提において、ロボットが長時間活動を行う場合の具体的な問題を以下にあげる。

- スキャン数の増加

前述のように、活動時間に比例してスキャン数は単調増加する。得られたスキャンをすべて保存すると、メモリやストレージの使用量が増える。

- ポーズ調整の処理時間

ポーズ調整の処理時間は、ポーズグラフのノード数にほぼ比例し、ループの数にも依存する。スキャンマッチングに基づく SLAM では、ロボット位置はスキャンごとに推定されるため、ロボット位置およびノードの個数はスキャン数に比例する。また、環境が小規模ならば同じ場所を何度も通ると考えられるので、軌跡が長くなると、ループの数も増えると想定される。したがって、スキャン数が増えればポーズ調整の処理時間も増える。

Fig. 1 The transition of cell values when adding scans to the occupancy grid map. black: occupied cells, gray: unknown cells, white/yellow/blue: free cells by scan1/scan2/scan3. (a) Ground truth map, (b) free cells (white) by adding scan1, (c) free cells (yellow) by adding scan2, (d) free cells (blue) by adding scan3. There remain only a few cells given by scan1. This implies that scan1 has less contribution to the map.

● 点群地図のサイズと再構築時間

ポーズ調整を行うとロボットの軌跡が修正されるため、それに合わせて点群地図を再構築する必要がある。前述のように、点群地図を構成するマップ点はスキャン点の時空間の局所範囲でまとめたクラスターであり、マップ点の個数はスキャン数にほぼ比例する。このため、スキャン数が増えれば、点群地図のメモリ量も再構築時間も増える。

● 占有格子地図の再構築時間

ポーズ調整を行うと占有格子地図も再構築する必要がある。占有格子地図はレイキャスティング法を用いて行うが、再構築の際は、全スキャンを使ってレイキャスティングをし直すことになる。したがって、再構築時間はスキャン数に比例する。また、再構築に備えて全スキャンを保存しておく必要があり、そのためのメモリ量はスキャン数に比例する。一方、対象環境の面積が変わらなければ、占有格子地図のサイズは固定であり、地図自体のメモリ量の増加はない。

2.3 アプローチ

前節で述べたように、長時間活動を行うと、多くの要素で処理時間とメモリ量の増加が生じるが、これらはすべてスキャン数の増加に起因する。したがって、スキャン数を削減できれば、問題が解決すると考えられる。

本稿の手法は、スキャンの時間関係を考慮しながら、有用な新しいスキャンを選別することでスキャン数を削減する。この選別は、図1のように、占有格子地図を用いて行う。

まず、占有格子地図の標準的な生成方法であるレイキャスティングを説明する。レイキャスティングでは、センサ中心からレーザビームを模擬した直線を伸ばし、ビームが通過したセルの値を更新する。図1で、レーザスキャナの計測範囲が円内であるとすると、ビームが物体に当たった位置にあるセルは占有 (値 = 1)、ビームが通過したセルは自由 (値 = -1) となる。計測範囲内に物体がない場合はビームは返ってこないため、未観測となり、通常はセル値を変えない¹。

占有格子地図のセル値は対数オッズで管理され、複数のスキャンによるセル値を合算して対数オッズを計算する。たとえば、あるセルの占有が3回、自由が2回あれば、対数オッズは $3 + (-2) = 1$ となる。この対数オッズから占有確率が計算される (Thrun et al., 2005)。ただし、図1では、簡単のため、セル値を単純に上塗りしている。

次に、図1を用いて提案手法の考え方を説明する。同図 (a) は占有格子地図の真値であり、実環境には壁と物体があるとする。同図 (b) は、ロボットがスキャン1を取得したときの占有格子地図のセル値である。ここでは、スキャン1による自由セルを白で表す。同様に、同図 (c), (d) は、スキャン2、スキャン3によるセル値であり、それらの自由セルをそれぞれ黄色と青で表す。占有セルはいずれも黒で表す。

この図を見ると、スキャン1が書き込んだセル値の多くは、その後のスキャン2とスキャン3によって上塗りされている。スキャン1の占有セル (黒) は、スキャン2とスキャン3の円内に含まれるので、すべて上塗りさ

¹多くのレーザスキャナは近接領域も見えないため、ごく近くに障害物がある可能性を排除できないからである

れる。スキャン1の自由セル（白）はわずかしかなかった。これは、この時点では、スキャン1が占有格子地図の生成にあまり寄与していないことを示している。この例では、セル値を単純に上塗りしているが、多くのスキャンから占有確率を計算する場合でも、スキャンが重複するにつれて同様の現象が生じる。

本稿の手法は、各スキャンが占有格子地図の生成に寄与したセル数からそのスキャンの影響度を計算し、影響度の小さいスキャンを削減する。図1からわかるように、スキャンの影響度は、時間の進行とともに変化する。一般には、古いスキャンほど影響度は小さくなる。ただし、時間的には古くても、その領域で新しいスキャンが取得されなければ、古いスキャンは生き残る。このように、本稿の方法は、相対的に新しいスキャンを残す性質をもつので、スキャンの削減と同時に、地図の更新も行いやすい。

2.4 関連研究

長時間活動のためにポーズグラフを削減する既存研究はいくつか存在する。(Konolige and Bowman, 2009)は、古い観測を削除してポーズグラフを削減するアルゴリズムを提案している。ただし、Visual SLAMを対象にしており、占有格子地図は用いていない。(Kretschmar and Stachniss, 2012)は、占有格子地図の生成に関して情報量が少ないレーザスキャンを捨ててポーズグラフを圧縮する。この方法では時間関係が考慮されていないので、情報量が高ければ、新しいスキャンよりも古いスキャンが残る可能性がある。このため、変化しない環境が対象となる。(Walcott-Bryant et al., 2012)は、2つの占有格子地図を比較して、古い点を記録しながら、削除すべきポーズグラフのノードを決める。この考え方は本稿の手法に近いが、本稿では観測度数を用いてレイキャスティング中に効率よく処理する点が異なる。

占有格子地図を工夫して動的環境に対処する研究もある。(Yang and Wang, 2011)は、占有格子地図に feasibility という属性をもたせ、静止物体と移動物体を区別する方法を提案している。移動物体の分離が主な目的であり、地図の削減とは異なる。(Meyer-Delius et al., 2012)は、各セルの時間軸におけるマルコフモデルを定義して環境変化をとらえる。ただし、パーティクルフィルタによる自己位置推定が目的であり、ポーズグラフ縮約や地図の削減は行わない。

3. 観測度数を用いたスキャンの削減

3.1 観測度数の導入

前述のように、占有格子地図では、各セルに対数オッズを持たせて占有確率の計算を行う。しかし、対数オッズだけでは、そのセルを何回観測したかの情報は残らない。たとえば、あるセル c_1 のオッズが -3 となるには、 c_1 をビームが3回通過して自由値 (-1) が3回加算されてもよいし、あるいは、 c_1 をビームが6回通過・3回到達して自由値 (-1) が6回と占有値 (1) が3回加算されてもよい。対数オッズだけでは、この区別はつかない。

占有格子地図を作る際、ノイズや誤差があっても安定した結果となるためには、複数回の観測によって、各セルに多くのビームが通って、統計的に処理することが望ましい。しかし、観測は多いほどよいわけではなく、古い観測は新しい観測によって塗り替えられて、必要なくなることも多い。たとえば、物体の位置が変わると、移動元のセルは自由状態となり、移動先のセルは占有状態となる。移動後の位置が確定すれば、古い観測は必要なくなる。これらのことから、観測した回数を明示的に管理し、適切な数の観測でセル値を計算すれば、無駄な計算を減らせると考えられる。

提案手法では、これを実現するために観測度数を導入する。各セルは観測度数をもち、そのセルを観測した回数を観測度数に記録する。そして、観測度数に基づいて最新の N 個の観測だけを残し、古い観測を捨てる。 N は対数オッズから占有確率を計算するのに十分な観測数である。通常の実用ならば、経験的には、1~2秒分の観測でよいと考えられる。たとえば、スキャンのフレームレートが10fpsならば、 $N = 10 \sim 20$ でよい。

3.2 スキャンの影響度

ここでは、観測度数を用いて、古くて情報量の少ないスキャンを見つけるための指標である影響度を提案する。まず、スキャン s_k に対して、有効自由セル数を f_k 、有効占有セル数を o_k とする。ここで有効とは、観測度数の閾値 $bthre$ より古くないことを意味する。 $bthre$ は前述の N に相当する。また、 i 番目のセルの対数オッズを v_i 、観

測度数を b_i とする. s_k のセンサ中心位置は SLAM によって得られているとする.

スキヤンの影響度 e_k を以下のように定義する.

$$e_k = f_k + A o_k \quad (1)$$

A は o_k に重みをつけるための定数である. 占有格子地図では, 一般に, 占有セルは, 自由セルに比べて数は少ないが, 障害物を示すので重要性は高い. このため, A によって占有セル数に重みを持たせ, 有効占有セルをもつスキヤンを優先する仕組みを入れておく. ただし, 本稿ではつねに $A = 1$ とする.

影響度は, レイキャスティングの最中に計算する. 影響度の計算を含めたレイキャスティングの手順を以下に示す.

- (1) 全セルについて, $v_i = 0, b_i = 0$ に初期化する.
- (2) 新しいスキヤンから降順に, 各スキヤン s_k に対して以下の処理をくり返す.
- (3) s_k を得たときのセンサ中心から, レーザビームを模した直線を計測距離だけ (物体に到達するまで) 伸ばし, 直線上の各セルについて以下の処理をくりかえす. これを s_k のすべてのビームについて行う.
  - (a) 直線が通過したセル (自由空間)

if $b_i < b_{thre}$ then $f_k = f_k + 1, v_i = v_i - 1, b_i = b_i + 1$.
  - (b) 直線が到達したセル (物体)

if $b_i < b_{thre}$ then $o_k = o_k + 1, v_i = v_i + 1, b_i = b_i + 1$.
- (4) f_k, o_k からスキヤンの影響度 e_k を求める.

この処理は次のことを意味する. スキヤン s_k のビームによってセル値が更新されるかどうかは観測度数で判定される. 上記の処理では, 新しいスキヤンから処理するので, セル c_i の観測度数が閾値 b_{thre} を超えれば, その先のスキヤン s_n ($1 \leq n \leq k-1$) は c_i に影響を与えない. s_n は, c_i に対しては古い観測であり, c_i は s_n の影響度に加算されない. そして, 大部分が古い観測であると判定されたスキヤンは, 影響度が小さくなり, 占有格子地図を更新する際に必要ないと判定される.

この方法の利点は以下である.

- レイキャスティングの過程で計算できるので, 計算コストが小さい.
- 新しい観測が少ない領域では, 古いスキヤンでも残る. 新しい観測が少なければ, 観測度数が閾値に満たなくなり, スキヤンは削除されず, 最後に観測した状態が保たれる.

なお, 上記手順のステップ (3) において, $b_i \leq b_{thre}$ にらず常に $v_i = v_i - 1$ あるいは $v_i = v_i + 1$ を行うようにしてもよい. すなわち,

- (a) 直線が通過したセル (自由空間)

$v_i = v_i - 1$. if $b_i < b_{thre}$ then $f_k = f_k + 1, b_i = b_i + 1$.
- (b) 直線が到達したセル (物体)

$v_i = v_i + 1$. if $b_i < b_{thre}$ then $o_k = o_k + 1, b_i = b_i + 1$.

こうすると, セル値の更新に b_{thre} を過ぎた古いビームも使われるようになり, セル値の安定性が増す. ただし, 環境変化に対するセル値の更新は遅くなるので, どちらを選ぶかは目的に応じて決めればよい.

3.3 スキヤンの削減

影響度に基づいてスキヤンを選別する最も簡単な方法は, 影響度の閾値 e_{thre} を用いて, $e_k < e_{thre}$ となるスキヤンを削除することである. しかし, 適切な e_{thre} は環境によって変わり, 設定が難しいという問題がある.

これに対処するため, スキヤン集合の網羅率を用いて e_{thre} を決める方法を提案する. ここで, スキヤン集合 S の網羅率とは, 次式のように, 現時点で残っている全スキヤン集合 S_0 の影響度の合計に対する, S による影響度の合計の比である.

$$C(S) = \frac{\sum_{s \in S} e(s)}{\sum_{s \in S_0} e(s)} \quad (2)$$

Fig. 2 Comparison of robot trajectories (red) with and without scan reduction

網羅率が大きいほど、 S による占有格子地図は S_0 による占有格子地図に近づく。このように、網羅率は占有格子地図の完成度合いを表すことができ、直感的に指定しやすい指標であると考えられる。スキャンを削減しつつ網羅率を高く保つには、影響度が大きいスキャンを優先して S に入れるとよい。このような考えにもとづき、網羅率を用いてスキャン選別の閾値 $ethre$ を決める手順を以下に示す。

- (1) 全スキャンの影響度の合計 $totalE$ を求める。
- (2) 全スキャンを影響度の順にソートする。
- (3) 各スキャン s_k について、影響度の大きい方から以下の処理を行う。 $coverage$ はユーザが指定した網羅率である。

- $sum = sum + e_k$
- $\frac{sum}{totalE} < coverage$ ならば、ステップ (3) をくり返す。

- (4) ステップ (3) 終了時のスキャンの影響度 e_k を閾値 $ethre$ とする。

3.4 スキャン削減の例

提案手法がどのように作用するか、簡単な実験で示す。図 2 は、本手法を適用しない場合 (a) と網羅率 0.95 で適用した場合 (b) を比較したものである。これらの図では、占有格子地図の上に、地図生成に用いたスキャンに対応するロボット位置を赤で表示している。同図 (a) に示すように、ロボットは A, B, ..., F を順にたどって走行した。同図 (b) では、スキャンが削減されているので、ロボット位置も断片的になっている。なお、ここでは、見やすさのために単純な短い経路をとっており、削減率は高くない。

同図 (b) を見ると、提案手法によって古いスキャンが削除されているのがわかる。たとえば、区間 AB の前半部は削除されている。その部分で取得したスキャン群は主に部屋の上部のセルの占有確率を決めるが、後に走行した区間 EF の後半部によってセル値が置換えられ、区間 AB の影響度が減ったためである。また、区間 CD も削除されているが、これは、区間 DE の残っている部分で置換えされたためである。なお、同図 (b) において、区間 CD の右上にセル値が未定の領域（灰色）があるが、これは網羅率 0.95 であるために生じた欠落である。

別の問題として、(a) では壁に対する占有セル（黒）が欠落なく出ているが、(b) では欠けている部分がある。これは、スキャンの歪みや誤差のため、最近の少数のスキャンだけでは、セル値が不安定になるためと考えられる。これは、セル値計算で多くのスキャンを使用すれば改善する。すなわち、観測度数はスキャン選択だけに用いて、3.2 節の最後に述べたように、対数オッズ v_i の計算を観測度数によらずに行えばよい。ただし、こうすると、環境変化を迅速に反映することはできなくなる。このように、セル値の安定性と環境変化への対応は相反する面がある。このような特性は占有格子地図に本質的なものと考えられるが、うまくバランスをとることは興味深い課題である。

4. 地図と処理時間の削減

4.1 地図の削減

前章の方法でスキャンを削減すると、占有格子地図と点群地図も削減することができる。

占有格子地図は、削減されて残ったスキャンを用いてレイキャスティングを行い、再構築する。スキャン数が減った分だけ、占有格子地図の生成時間も削減される。情報が古いスキャンは、レイキャスティングにおいて他のスキャンで上塗りされるため、その情報の大部分が最新のセル状態には反映されない。そのようなスキャンを削減しても、占有格子地図は、スキャンを削減しなかった場合とほとんど変わらないと期待される。

点群地図も、削減されて残ったスキャンを用いて再構築する。前述のように、点群地図はマップ点の集合であり、マップ点は時空間の局所範囲内にあるスキャン点のクラスターである。1個のマップ点に対するスキャン点群は、ロボットが移動しながら得た複数のスキャンをSLAMによって地図座標系に配置し、一定時間内で同一点と見なせるほど空間的に近いスキャン点を集めたものである。概念的には、局所範囲内に真の点 p があると想定し、その近傍のスキャン点群が p の計測データであるとみなして、そのスキャン点群から p を推定したものがマップ点である。多くの場合、そのスキャン点群の重心をマップ点とする。なお、実装では、時間範囲はロボットが5m走行する時間、空間範囲は半径5cmとしている。

このように、マップ点を作るスキャン点群は複数のスキャンに由来するので、1個のマップ点 p_j に対して複数のスキャンが対応しうる。その中の最初のスキャンを p_j の開始スキャン、最後のスキャンを p_j の終了スキャンと呼ぶ。開始スキャンの番号を $ssid_j$ 、終了スキャンの番号を $esid_j$ とすると、マップ点削減を以下のように行う。

- (1) 各マップ点 p_j について以下の処理を繰り返す。
- (2) 点 p_j の所属するスキャン $id(sid_j)$ を次のように求める。

$$sid_j = (ssid_j + esid_j) / 2 \quad (3)$$

- (3) sid_j のスキャンの影響度が $ethre$ 未満ならば、 p_j を削除する。

ステップ(2)において開始スキャンと終了スキャンの間の中のスキャンを採用しているのは、中間のスキャンの方が端のスキャンよりもマップ点に寄与するスキャン点数が多く、マップ点との関係が強いと考えられるからである。なお、後述のポーズグラフの縮約手順と同様に、開始スキャンから終了スキャンまでの全スキャンの影響度を $ethre$ で評価する方法も考えられる。しかし、マップ点の時間範囲は現在の実装で5m走行分であり、キーフレーム間隔よりも長いため、その区間のスキャン数が多くなり、処理量が増えたり、評価基準の設定が複雑になるなどの問題が生じるため、マップ点には採用しなかった。

4.2 ポーズグラフの縮約

スキャンを削減すればポーズグラフも削減できるが、ポーズグラフがもつ情報をできるだけ失わないように処理する必要がある。前述のように、ポーズグラフはノードとアークからなる。ノードはロボット位置を表し、アークはノード間の相対位置を表す。スキャンマッチングによるSLAMでは、各スキャンに対してロボット位置を求めるので、ノードとスキャンは1対1に対応する。ただし、ポーズ調整をキーフレームで行う場合は、キーフレーム単位でロボット位置を抽出してノードとする。

本稿では、前稿(友納, 2017)と同様に、キーフレーム単位でノードを生成する。ここで、キーフレームとは一定間隔で間引いたスキャンのことであり、並進距離または回転角が閾値(現状では、それぞれ0.4mと60度)を超えるたびに抽出する。そして、ノードに対応するキーフレーム間のスキャン群の中で、影響度が閾値 $ethre$ 以上のスキャンが1つでもあれば、そのノードを採用する。採用されないノードは削除する。この手順を以下に示す。

- (1) ポーズグラフの各ノード n_i に対して、以下の処理を行う。 n_i に対応するスキャンを s_{j_i} とする。 j_i は i 番目のノードに対応するキーフレームのスキャン番号である。
- (2) ポーズグラフの最初のノード n_1 と最後のノード n_M は採用する。
- (3) ノード n_i ($i > 1$)について、キーフレーム間に存在するスキャン $s_{j_{i-1}+1}$ から s_{j_i} の間に、その影響度が閾値 $ethre$ 以上のスキャンがあれば、 n_i を採用する。

Fig. 3 Pose graph reduction. (a) concatenating of two arcs connected with n_2 , (b) double concatenating of three arcs connected with n_2 , (c) merging of two parallel arcs generated by removing n_2 and n_5 .

ノードを削除するとアークの付け替えを行う必要がある。これは、従来研究 (Kretzschmar et al., 2010)(Thrun et al., 2005) と同様に行う。基本的には、直列につながったアークの統合 (concatenating), および、並列につながったアークの統合 (merging) の2種類の操作がある。図3にその様子を示す。 c_{ij} はノード間の相対位置 (定数) であり、ポーズ調整で拘束として使われる。 c_{ij} は正規分布 $N(\mu_{ij}, \Sigma_{ij})$ で表される。 μ_{ij} は相対位置, Σ_{ij} はその共分散行列である。

図3(a)は、直列につながったノード n_2 を削除した例である。 n_2 につながるアークを削除し、新たに n_1 と n_3 の間にアークを張る必要がある。このとき、 n_1 と n_3 の相対位置 c_{13} を、 c_{12} と c_{23} から次のように計算する。

$$\begin{aligned}\Sigma_{13} &= \Sigma_{12} + \Sigma_{23} \\ \mu_{13} &= \mu_{12} \oplus \mu_{23}\end{aligned}\quad (4)$$

$\oplus$ は compounding オペレータである (Smith and Cheeseman, 1986)。

同図(b)は、ノード n_2 がループの結節点となる場合である。 n_2 を削除すると、 n_1 と n_3 , および、 n_4 と n_3 の間に、それぞれ、アーク c_{13} と c_{43} ができる。これらは直列アークの統合であり、その相対位置は式(4)と同様に計算すればよい。

同図(c)は、ループが重なっている際に、ノード n_5 , n_2 を続けて削除した例である。 n_5 を削除すると、 n_4 と n_3 の間に新たにアークが張られる。その相対位置 c_{43a} は式(4)と同様に計算できる。 n_2 を削除すると、 n_4 と n_3 の間にもう1本のアーク c_{43b} ができる。 n_4 と n_3 の間の2本のアークは次式の merging 処理により1本にまとめる。

$$\begin{aligned}\Sigma_{43} &= (\Sigma_{43a}^{-1} + \Sigma_{43b}^{-1})^{-1} \\ \mu_{43} &= \Sigma_{43}(\Sigma_{43a}^{-1}\mu_{43a} + \Sigma_{43b}^{-1}\mu_{43b})\end{aligned}\quad (5)$$

5. 実 験

小型2Dレーザスキャナを用いて大学校舎内でデータを取得し、提案手法を評価した。このレーザスキャナは360度全周を計測でき、角度分解能は1度、最大計測距離は約6m、スキャン取得レートは10Hzである。実験に用いたシステムのCPUはcore i7-4910MQ, OSはWindows 7, 言語はVisual C++である。

使用したデータセットを表1に示す。dataset1では、ロボットが部屋の中を走り回り、同じ場所を不規則に何度も通り、重複領域は不規則で複雑である。dataset2では、廊下の小さなループをほぼ規則的に8周する。そのため、コースのほぼ全体が規則的に重複する。dataset3では、dataset2を含む広い廊下環境を走行する。ループは多種類あり、重複回数が多い領域と少ない領域が分かれる。どのデータセットも最大9,000スキャンまでを使用した。

すべての実験で、パラメータは $bthre = 20, A = 1$ とし、 $ethre$ は3.3節の方法で網羅率に応じて自動設定した。

Table 1 Datasets used in the experiments.

	type	size[m]	#loop	overlap
dataset1	room	5 x 7	1 ~ 8	many, complex
dataset2	corridor	10 x 10	8	many, regularly
dataset3	corridor	40 x 10	1 ~ 4	few, regularly

Fig. 4 Relationships between coverage and reduction ratio.

Fig. 5 The comparison of the maps reduced with coverage 0.6 and 0.9.

5.1 網羅率に対するスキンの削減率

網羅率によってスキンの削減率がどう変化するか調べた。本稿では、削減率 (reduction ratio) を、削減後のスキン数を削減前のスキン数で割った値で定義する。この定義では、削減後のスキンが少ないほど削減率は小さくなる。本稿では、これを「削減率が高い (良い)」と表現する。

各データセットに対して、スキン数を 3,000 個, 6,000 個, 9,000 個とした場合の結果を図 4 に示す。当然の傾向として、網羅率を高くするほど削減率は低くなっている。また、どのデータセットも、スキン数が多いほど削減率は高くなっている。これはスキン数が多いほど、重複して計測する面積が増えるためである。

データセットの違いの要因は面積である。同じスキン数ならば、面積が小さいほど計測の重複度が大きくなる。つまり、何度も同じ場所を計測していることになり、スキンの削減率が高くなるはずである。実際、dataset1 は最も面積が小さく、削減率は高くなっている。また、dataset3 は最も面積が大きく、削減率は低くなっている。

生成された地図の品質が網羅率によって変わる様子を図 5 に示す。これは、占有格子地図を網羅率 0.6 と 0.9 とで比較した例である。網羅率 0.6 では欠けた部分があるが、網羅率 0.9 だと欠けた部分はほぼ埋まっている。

ただし、dataset3 については、網羅率 0.9 でも欠けた領域が散見される。この 1 つの要因は、dataset3 の面積が広いことである。このため、スキンの重複が小さく、影響度の大きいスキンが多数存在する。もう 1 つの要因は長い廊下である。実験に用いたレーザスキャナの最大計測距離は 6m であるため、長い廊下の長手方向では計測不可となって、前後方向のかなりの領域が未観測になる。このため、廊下の中間部分のスキンは有効セル数が減って、影響度が小さくなる。この 2 つの要因が重なって、dataset3 では欠落部分が残ったと考えられる。網羅率を高くすれば欠落部分は解消するが、削減率は低くなる。この対処は今後の課題である。

5.2 スキン数に対する削減率

次に、スキン数に対して削減率がどう変化するか調べた。図 6 に入力スキン数と削減後のスキン数の関係を示す。どのデータセットにおいても、入力スキン数が増えても、削減後のスキン数はそれほど増えず、一定あるいは微増であることがわかる。これは、走行につれてスキン数が増えると同時に、重複部分も増えるた

Fig. 6 Relationships between #input scan and #reduced scans.

Fig. 7 Relationships between #input scan and #pose nodes.

Fig. 8 Relationships between #input scan and #pose arcs.

めである。

図7に入力スキャン数と削減後のノード数の関係を示す。点線はポーズグラフのキーフレーム間引きだけ行った場合のノード数である。キーフレーム間引きだけだと、ノード数はスキャン数に比例して増加する。提案手法を行うことで、ノード数はさらに削減され、しかも、スキャン数に対する増加率も抑えられていることがわかる。

図8に入力スキャン数と削減後のアーク数の関係を示す。点線はキーフレーム間引きだけ行った場合のアーク数である。アークもノードと同様の傾向があるが、アーク数はループの個数にも依存するので、増減の挙動はノードよりも複雑である。

5.3 処理時間とメモリ量の削減

表2に、スキャン9,000個、網羅率0.95におけるデータと処理時間の削減量を示す。たとえば、dataset1において、スキャン数が9,000個から965個、ポーズノード数が416個（キーフレーム間引き済）から83個、ポーズアーク数が1,352個から257個、点群地図の点数が45,030個から8,247個に削減された。また、ポーズ調整時間（PA time）は16[ms]から3[ms]、占有格子地図の生成時間（GM time）は267[ms]から35[ms]に削減された。点群地図の生成時間はもともと1[ms]程度で、高精度に時間測定できなかったため、この表には載せていない。

おおまかな傾向として、スキャン数の削減率に応じて、他の量も減っていることがわかる。ただし、dataset2のポーズ調整時間は他と比べて大きい。これは、dataset2が最もループが多く、ポーズアークが多数あるためと考えられる。また、dataset2とdataset3の地図点数がdataset1より少ないが、これは、前述のように、長い廊下において前後方向の計測セル数が減ったためと考えられる。

スキャンやポーズグラフの削減によってメモリ量も削減される。Windowsのリソースモニタを用いて、dataset1について、スキャン削減ありとスキャン削減なしとのメモリ使用量の差（削減量）を調べたところ、削減前のスキャン数が9,000個の場合の削減量は約16MB、27,000個の場合の削減量は約71MBであった。なお、27,000個で

Table 2 Computation time: gainRate=0.95, obthre=20, scanNum=9,000

Dataset	scan num	node num	arc num	PA time	GM time	pt num
dataset1	9,000 / 965	416 / 83	1,352 / 257	16 / 3	267 / 35	45,030 / 8,247
dataset2	9,000 / 1,753	671 / 231	2,532 / 644	28 / 6	306 / 69	34,612 / 6,642
dataset3	9,000 / 3,033	657 / 249	1,171 / 292	10 / 3	316 / 126	33,764 / 12,519

の削減後のスキャン数は 1086 個であった。スキャン 27000 個は 45 分間の走行にあたるが、数時間続けて走行させるような場合は、スキャン数を削減しなければ膨大なメモリを消費すると推測できる。

興味深い現象として、削減後の dataset2 のポーズノード数が、スキャン数との対比で見たとき、他よりも多いことがあげられる。スキャン数の削減率が約 0.19 なので、ポーズノード数は 671 個から 130 個程度に削減されるのが妥当と考えられるが、実際はそれより 100 個ほど多くなっている。4.2 節の手順では、ステップ (3) で、ポーズノード n_i に対して、 $s_{j_{i-1}+1}$ から s_{j_i} 間のどれかのスキャンの影響度が閾値を超えれば、 n_i が選択されることになっている。そこで、 n_i に直接対応するスキャン s_{j_i} のみに限定すると、dataset2 の削減後のポーズノード数は 132 個となり、上記の推定通りになった。このことは、 $s_{j_{i-1}+1}$ から s_{j_i} 間に影響度が閾値を超えたスキャンがまんべんなく存在したことを意味する。逆に、dataset3 ではそのようなスキャンが局在していたことになる。この理由として、dataset2 では経路のほとんどが多重ループで重複しており、影響度の分布が規則的になっていたことが考えられる。

5.4 削減された地図の定性評価

図 9 に、点群地図、ポーズグラフ、および、占有格子地図について、提案手法によるスキャンの削減前後の比較を示す。点群地図内に描かれた線図がポーズグラフである。網羅率は 0.95、スキャン数は 9,000 である。

点群地図では、点数が減るため、全体に薄くなっている。副次効果として、過去の移動障害物も削減される。原理的には、最新の観測分以外は削除される。

占有格子地図でも、最新の観測分以外は削除されるため、スキャンの削減前には全観測のオッズ集計によって消えた移動障害物が、最新の観測分に対応する部分だけ現れている。それに伴う副作用として、dataset1 では、左下の壁がやや薄くなっている。これは、スキャンの歪みや誤差のために少し位置がずれてセルの占有確率の計算を行ったためと推測される。なお、3.2 節の最後に述べたように、ステップ (3) での v_i の更新を $b_i < bthre$ によらず常に行うと、セルの占有確率が安定されて壁が薄くなくなる。

dataset1 と dataset2 では、網羅率 0.95 で占有格子地図は十分なものが得られている。しかし、5.1 節と同様に、dataset3 では一部に欠けた（未計測）領域が残っている。これは網羅率を 0.98 にすると解消される。

6. 結 言

本稿では、2D レーザスキャナを用いた長時間の地図構築において、センサデータおよび地図の内部データを削減する手法を提案した。本手法では、占有格子地図のセルに観測度数を導入して、レイキャスティングによる占有格子地図生成に対するスキャンの影響度を計算し、スキャン選別の基準とする。そして、指定した網羅率に応じてスキャンの影響度の閾値を自動設定し、スキャンを選別する。実環境データで実験を行い、本手法によりスキャンが削減され、それに伴って、ポーズグラフと点群地図のメモリ使用量が削減され、ポーズ調整の処理時間や占有格子地図の生成時間も削減されることを確認した。

今後の課題として、削減後の地図の完成度の保証があげられる。現状では、データセットによっては、高い網羅率を指定しても、生成された占有格子地図の一部が欠落する現象が見られるため、その改善が必要である。たとえば、環境変化やノイズに対して占有格子地図の安定性を高めるためには、観測時刻や最後の観測からの経過時間、レーザスキャナの有効計測距離などを考慮することが役に立つ可能性があり、この検討は今後の課題である。

また、大規模なデータに本手法を適用して、長時間地図構築の検証実験を行うことも今後の課題である。本手法は、新しい情報をもつスキャンを優先して残す性質があるため、地図更新と相性がよいと考えられるが、さまざまな種類の環境変化にそのまま対応できるわけではない。移動物体や準静止物体などの環境変化の要因をよく分析し、地図更新への適用を検討していきたい。

Fig. 9 Comparison of the maps with and without reduction. top: point cloud map and pose graph, bottom: occupancy grid map.

文 献

- Besl, P. J. and McKay, N. D., A method of registration of 3-D shapes, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 14, No. 2 (1992), pp. 239-256.
- Irie, K. and Tomono, M., A compact and portable implementation of graph-based SLAM, *Proceedings of ROBOMECH* (2017), 2P2-B01.
- Konolige, K. and Bowman, J., Towards lifelong visual maps, *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems* (2009), pp. 1156-1163.
- Konolige, K., Grisetti, G., Kummerle, R., Burgard, W., Limketkai, B. and Vincent, R., Efficient sparse pose adjustment for 2D mapping, *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems* (2010), pp. 22-29.
- Kretzschmar, H., Grisetti, G. and Stachniss, C., Lifelong map learning for graph-based SLAM in static environments, *Künstliche Intelligenz*, Vol. 24, Issue 3 (2010), pp. 199-206.
- Kretzschmar, H. and Stachniss, C., Information-theoretic compression of pose graph for laser-based SLAM, *International Journal on Robotics Research*, Vol. 31, Issue 11 (2012), pp. 1219-1230.
- Meyer-Delius, D., Beinhofer, M. and Burgard, W., Occupancy grid models for robot mapping in changing environments, *Proceedings of AAAI Conference on Artificial Intelligence* (2012), pp. 2024-2030.

- Olson, E., Leonard, J. and Teller, S., Fast iterative alignment of pose graphs with poor initial estimates, Proceedings of IEEE International Conference on Robotics and Automation (2006), pp. 2262–2269.
- Smith, R. and Cheeseman, P., On the representation and estimation of spatial uncertainty, International Journal on Robotics Research, Vol. 5, Issue 4 (1986), pp. 56–68.
- Thrun, S., Burgard, W. and Fox, D., Probabilistic robotics (2005), MIT Press. (邦訳) 上田隆一訳, 確率ロボティクス (2007), 毎日コミュニケーションズ.
- 友納正裕, 2D レーザスキャナによる SLAM における地図の点群表現とループ閉じ込み, ロボティクスシンポジウム予稿集 (2017), pp. 42–47.
- Walcott-Bryant, A., Kaess, M., Johannsson, H. and Leonard, J. J., Dynamic pose graph SLAM: long-term mapping in low dynamic environments, Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (2012), pp. 1871–1878.
- Yang, S. W. and Wang, C. C., Feasibility grids for localization and mapping in crowded urban scenes, Proceedings of IEEE International Conference on Robotics and Automation (2011), pp. 2322–2328.

References

- Besl, P. J. and McKay, N. D., A method of registration of 3-D shapes, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 14, No. 2 (1992), pp. 239–256.
- Irie, K. and Tomono, M., A compact and portable implementation of graph-based SLAM, Proceedings of ROBOMECH (2017), 2P2-B01.
- Konolige, K. and Bowman, J., Towards lifelong visual maps, Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (2009), pp. 1156–1163.
- Konolige, K., Grisetti, G., Kummerle, R., Burgard, W., Limketkai, B. and Vincent, R., Efficient sparse pose adjustment for 2D mapping, Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (2010), pp. 22–29.
- Kretzschmar, H., Grisetti, G. and Stachniss, C., Lifelong map learning for graph-based SLAM in static environments, Künstliche Intelligenz, Vol. 24, Issue 3 (2010), pp. 199–206.
- Kretzschmar, H. and Stachniss, C., Information-theoretic compression of pose graph for laser-based SLAM, International Journal on Robotics Research, Vol. 31, Issue 11 (2012), pp. 1219–1230.
- Meyer-Delius, D., Beinhofer, M. and Burgard, W., Occupancy grid models for robot mapping in changing environments, Proceedings of AAAI Conference on Artificial Intelligence (2012), pp. 2024–2030.
- Olson, E., Leonard, J. and Teller, S., Fast iterative alignment of pose graphs with poor initial estimates, Proceedings of IEEE International Conference on Robotics and Automation (2006), pp. 2262–2269.
- Smith, R. and Cheeseman, P., On the representation and estimation of spatial uncertainty, International Journal on Robotics Research, Vol. 5, Issue 4 (1986), pp. 56–68.
- Thrun, S., Burgard, W. and Fox, D., Probabilistic robotics (2005), MIT Press.
- Tomono, M., Point-cloud map representation and loop closure in 2D lidar SLAM, Proceedings of Robotics Symposia (2017), pp. 42–47 (in Japanese).
- Walcott-Bryant, A., Kaess, M., Johannsson, H. and Leonard, J. J., Dynamic pose graph SLAM: long-term mapping in low dynamic environments, Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (2012), pp. 1871–1878.
- Yang, S. W. and Wang, C. C., Feasibility grids for localization and mapping in crowded urban scenes, Proceedings of IEEE International Conference on Robotics and Automation (2011), pp. 2322–2328.