

## 測域センサを用いた移動ロボットの環境地図構築及び歪み補正に関する検討 Mapping and Correction Method using Mobile Robot with LRF

○横山和成\*, 石井雅樹\*

○Kazuaki Yokoyama, Masaki Ishii

\*秋田県立大学

\*Akita Prefectural University

キーワード : SLAM(Simultaneous Localization and Mapping), オドメトリ(Odometry),  
ICP アルゴリズム(Iterative Closest Point Algorithm),  
モンテカルロ法(Monte Carlo Method), LRF(Laser Range Finder)

連絡先: 〒015-0055 秋田県由利本荘市土谷字海老ノ口 84-4

秋田県立大学 システム科学技術学部 機械知能システム学科

石井 雅樹, Tel: (0184)27-2220, Fax: (0184)27-2188, Email: ishii@akita-pu.ac.jp

### 1. はじめに

近年, ロボットの利用分野は拡大しつつあり, 人間の生活環境内で利用されるロボットの開発が活発になっている. 環境中においてロボットが作業を行うためには, 周囲環境について障害物の位置情報等の幾何学的特徴により構成される環境地図が必要である. 従来の工場内等の環境では人間が環境地図を構築し, ロボットに与えてきた. しかし, 人間の生活環境といった多様な環境において事前に環境地図を構築することは困難であり, ロボットによる自律的な環境地図構築手法が求められている.

現在研究されている環境地図構築手法に SLAM(Simultaneous Localization and Mapping)がある. これはロボットが取得したセンサデータから, 自身のグローバル座標系における位置と傾き(以下, 姿勢とする)を推定し, その地点での外界センサによるスキャンデータ(以下, スキャンとする)と合わせて環境地図に登録することで地図を構築する.

SLAM を処理のタイミングにより大別すると, センサデータの取得と同時に逐次的に処理を行うオンライン手法と, センサデータを収集

した後にまとめて処理を行うオフライン手法に分けることができる. 前者は逐次型であるため処理の過程で環境地図が得られるという利点を持つが, 環境地図の精度はオフライン手法に比べ劣る. 一方, 後者は処理の段階で全てのセンサデータを用いることができるため精度を高めることが可能であるが, センサデータの収集と同時に環境地図を構築することができない. 特に, センサデータ取得間隔以内に処理が完了するオンライン SLAM をリアルタイム SLAM と呼ぶ.

以上より, 事前の環境地図構築が困難な環境において自律的に環境地図を構築するためには, 高精度に姿勢を推定し, かつリアルタイムに処理が可能な SLAM が必要であると考えられる. そこで本研究では, 多様な屋内環境を移動するロボットに搭載することを想定し, 一定の間隔で取得したセンサデータをもとに, 高精度かつリアルタイムに環境地図構築が可能な SLAM の開発を目的とする. なお, 本研究では多様な屋内環境として, 幾何学的特徴の多い環境(室内)と, 幾何学的特徴の乏しい環境(廊下)の 2 種類を扱う. 本稿では, 初めに複数のオンライン

ン SLAM 手法についてその処理手順を述べ、次に実際の屋内環境で取得したデータを用いてそれぞれの手法により構築した環境地図の精度を比較検討する。さらに、センサ誤差やモデル化誤差の蓄積により生じる地図の歪み補正処理について検討し、実験により提案手法の有用性を評価する。

## 2. SLAM(Simultaneous Localization and Mapping)

SLAM において環境地図はスキャンと姿勢の集合であると定義される。ここで時刻  $t$  におけるスキャンを  $o_t$ 、姿勢の推定値を  $\hat{s}_t$  で表すと、環境地図  $m_t$  は以下の式で表される。

$$m_t = \left\{ \left\{ o_t, \hat{s}_t \right\} \right\}_{t=0, \dots, t} \quad (1)$$

高精度の環境地図を構築するためには、高精度のスキャンと正確な姿勢が必要である。しかし、姿勢を直接求めることは不可能であり、ロボットが取得できる情報から自己の姿勢を求める必要がある。このような限定された情報からロボットの姿勢を推定する問題は自己位置推定問題と呼ばれ、地図構築問題は自己位置推定問題を内包している。一般に、より良い自己位置推定を行うためには内界センサデータの累積による自己位置推定に加え、環境地図と現在のスキャンとのスキャンマッチングにより自己位置を推定する必要がある。ここで、前者をオドメトリによる自己位置推定、後者をスキャンマッチングによる自己位置推定と呼ぶ。スキャンマッチングにより推定された姿勢の精度は環境地図の精度に依存し、一方で環境地図の精度は姿勢の精度に依存するという交差した関係が起こる。これは SLAM 問題と呼ばれる。

この問題を解決するために、環境地図と姿勢の同時確率を定義し、これを最尤化させる姿勢を求める手法が提案されている<sup>23)</sup>。これは姿勢と環境地図を条件とするスキャンの確率分布と、制御量と前回姿勢を条件とする姿勢の確率分布を、マルコフ性を仮定した上でベイズアンフィルタを用いて融合する手法である。ここで前者を知覚モデル(Perceptual Model)、後者を運動モデル(Motion Model)と呼ぶ。静的な環境を仮定するとき、時刻  $t$  における姿勢と環境地図の同時確率は以下の式で表される。

$$P(s_t, m | o^t, a^t) = \eta P(o_t | s_t, m) \int P(s_t | a_t, s_{t-1}) P(s_{t-1}, m | o^{t-1}, a^{t-1}) ds_{t-1} \quad (2)$$

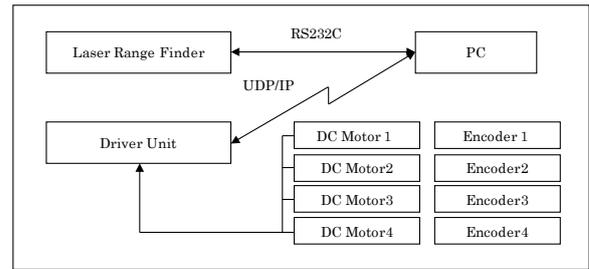


Fig. 1 システム構成



Fig. 2 ロボット外観

ここで  $P(o_t | s_t, m)$  は知覚モデル、 $P(s_t | a_t, s_{t-1})$  は運動モデルであり、 $o^t$ 、 $a^t$  はそれぞれスキャンデータとオドメトリの時系列を表す。SLAM では式(2)を最大化させる最尤姿勢  $\hat{s}_t$  を求め、式(1)により地図を更新する処理を繰り返すことで地図を構築する。

## 3. 屋内移動ロボット

### 3.1 ロボットの構成

本研究で使用するロボットに求められる仕様を以下に挙げる。

- ①平坦な床面での移動能力を有する
- ②ロボットの移動量を算出するための内界センサを持つ
- ③周辺環境の幾何学的特徴を得るための外界センサを持つ
- ④各センサ・アクチュエータを制御し、SLAM 処理を行うことができる。

そこで本研究では、三菱電機特機システム株式会社製小型全方位ロボット「MDT-RO-03」をベースに用いたロボットを設計した。システム構成図を Fig.1、ロボットの外観を Fig.2 に示す。このロボットは全方位移動機構を持ち、4軸をそれぞれ独立に制御することにより、あらゆる平面移動が可能である。各車軸にはロータリーエンコーダが挿入されており、車軸回転量を取得することが可能である。また、SLAM 処理およびコマンド送受信に PC、外界センサとしてレーザレンジファインダ(LRF)を搭載している。各モータおよびロータリーエンコーダはモータドライバに接続されている。PC

ーモータドライバ間は UDP/IP による通信が行われ、モータドライバは送られてきたコマンドによりモータ出力を制御し、同時に現在のモータ出力とロータリーエンコーダの積算値を送信する。PC-LRF 間は RS232C による通信が行われ、LRF が取得したスキャンデータを PC に送信する。

### 3.2 全方位移動機構

全方位移動機構は 90 度毎に配置された 4 個のオムニホイールにより構成され、それぞれを独立に駆動することが可能である。各車輪の回転速度を  $c_1 \sim c_4$  とすると、ロボット座標系におけるロボットの速度  $r\dot{x}$ ,  $r\dot{y}$ ,  $r\dot{\theta}$  は以下の式により求められる

$$\begin{pmatrix} r\dot{x} \\ r\dot{y} \\ r\dot{\theta} \end{pmatrix} = \begin{pmatrix} 0 & -\frac{d}{4} & 0 & \frac{d}{4} \\ -\frac{d}{4} & 0 & \frac{d}{4} & 0 \\ -\frac{d}{4T} & -\frac{d}{4T} & -\frac{d}{4T} & -\frac{d}{4T} \end{pmatrix} \begin{pmatrix} c_1 \\ c_2 \\ c_3 \\ c_4 \end{pmatrix} \quad (3)$$

ここで  $d$  はオムニホイールの直径、 $T$  はトレッドを表す。

### 3.3 内界センサ

ロボットは内界センサとして各車輪軸にロータリーエンコーダを持ち、その分解能は 150[ppr]である。ロータリーエンコーダのパルスはモータドライバに搭載されたバイナリカウンタによってカウントされ、電源投入時からの積算値を出力する。時刻  $t$  における各車輪軸の回転速度は以下の式により算出される。

$$c_t = \frac{2\pi}{N} \cdot \frac{\Delta p_t}{\Delta T} \quad (4)$$

ここで、 $N$  はエンコーダの分解能、 $\Delta p_t$  は時刻  $t$  におけるパルス積算値の差分値、 $\Delta T$  はサンプリング間隔である。 $\Delta p_t$  はバイナリカウンタの境界値を考慮して決定される。

### 3.4 外界センサ

外界の幾何学的特徴を取得することが可能なセンサを測域センサと呼び、その代表としてレーザレンジファインダ(LRF:Laser Range Finder)がある。レーザレンジファインダはレーザを扇状に照射し、反射のあった点の角度および距離を取得することができる。本研究では北陽電機株式会社製 URG-04LX(以下、URG とする)、UTM-30LX(以下、UTM とする)の 2 種類のレーザレンジファインダを使用する。

Fig.3(a)に URG の外観、同図(b)に UTM の外



(a) URG (b) UTM

Fig. 3 LRF の外観

Table 1 URG の仕様

|       |              |
|-------|--------------|
| 測定距離  | 60~4095[mm]  |
| 走査角度  | 240[deg]     |
| 距離分解能 | 10[mm]       |
| 角度分解能 | 0.34[deg]    |
| 走査時間  | 100[ms/scan] |

Table 2 UTM の仕様

|       |                   |
|-------|-------------------|
| 測定距離  | 0.1~30[m]         |
| 走査角度  | 270[deg]          |
| 距離分解能 | 0.1~10[m] ±30[mm] |
|       | 10~30[m] ±50[mm]  |
| 角度分解能 | 0.25[deg]         |
| 走査時間  | 25[ms/scan]       |

観を示す。また、URG の仕様を Table1、UTM の仕様を Table2 にそれぞれ示す。

## 4. SLAM 手法

### 4.1 オドメトリ

内界センサデータの累積による自己位置推定手法はオドメトリと呼ばれ、定数時間で処理が可能であるという利点を持つ一方で、モデル化誤差やスリップによる誤差の累積を修正できないという欠点を持つ。しかし、短区間での精度は比較的高く、小規模な環境で使用されるロボットの自己位置推定や、その他の自己位置推定手法の初期値として多く利用されている。

オドメトリによる SLAM の処理手順を以下に示す。

- ① スキャンおよび車輪回転速度の取得
- ② ロボット座標系におけるロボットの速度の算出
- ③ 世界座標系への座標変換
- ④ 前回姿勢との足し合わせによる姿勢の更新

ここで、車輪回転速度は式(4)、ロボットの速度は式(3)より算出される。

## 4.2 ICP スキャンマッチング

外界センサから得られるスキャンと環境地図をマッチングさせ、ロボットの姿勢を推定する手法をスキャンマッチングという。スキャンマッチングの代表的な計算法に ICP アルゴリズム(Iterative Closest Point Algorithm)がある<sup>5)</sup>。ICP アルゴリズムは 2 つの点群間で対応点を求め、対応点間の距離の二乗和が最小となるような姿勢を求める手法であり、その処理は以下の手順を繰り返すことにより行われる。

- ① マッチングさせるスキャンの初期姿勢の設定
- ② 対応点の探索
- ③ 移動量の推定
- ④ 繰り返し判定

ここで、初期姿勢は一般にオドメトリによる姿勢推定値を用いる。

## 4.3 モンテカルロ法

センサデータにより推定されたロボットの姿勢は不確かさを含んでいる。ここで不確かさを尤度として定義し、知覚モデルと運動モデルの尤度を融合する手法として式(2)が提案されている。しかし、この式を直接計算することはできず、多くの近似手法が提案されている。確率分布の仮定を設けない手法としてはモンテカルロ法があり、姿勢の確率分布をパーティクルの分布として近似することで任意の確率分布を離散的に表現可能である<sup>6)</sup>。以下にモンテカルロ法の処理手順を示す。

- ① リサンプリング
- ② 予測
- ③ 重み付け
- ④ 状態推定

## 4.4 提案手法

モンテカルロ法を基に 3 種類のオンライン SLAM 手法を組み合わせ環境地図の構築を行う。以下に処理の流れを示す。

- ① リサンプリング
- ② 予測
  - オドメトリによる自己位置の移動
  - ICP スキャンマッチング
  - 自己位置推定
- ③ 重み付け
- ④ 状態推定

より正確な環境地図を得るためには正確な自己位置を得る必要がある。本研究では正確な自己位置に注目し、自己位置推定を複数回行うことで正確な自己位置の獲得を行う。すなわち、予測ステップにおいて、はじめにオドメトリに

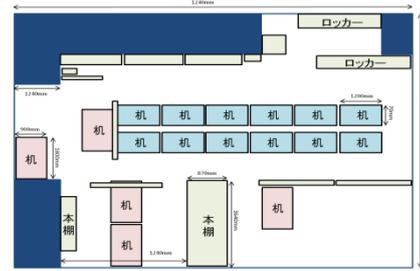


Fig. 4 室内環境

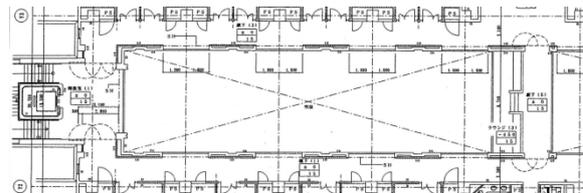


Fig. 5 廊下環境

よる自己位置の移動を行い、次に ICP スキャンマッチングにより地図の位置合わせを行う。その後 ICP スキャンマッチングにより移動した際の自己位置を推定することでより正確な自己位置の獲得を行う。

## 5. 環境地図構築実験

### 5.1 実験概要

本研究では前項で説明した SLAM 手法を実環境へ適用し、実装実験を行う。

実験に用いるスキャンデータ、オドメトリの時系列データは、Fig.2 に示す全方位移動ロボットを操作して 200[ms]の間隔で収集したセンサデータを使用する。

実験環境は、幾何学的特徴の多い環境である秋田県立大学ロボット工学研究室 GII-314(室内環境)と、幾何学的特徴の少ない環境である学部棟 II の 3 階廊下(廊下環境)を扱う。室内環境は机、椅子、仕切りといった障害物が多くあり、それらの幾何学的特徴により比較的精度よく環境地図の構築ができると考えられる。一方、廊下環境は柱およびドア付近の突起以外は目立った障害物が無く、幾何学的特徴の少ない環境である。実験環境の概略図をそれぞれ Fig.4, 5 に示す。

### 5.2 オンライン SLAM により構築した環境地図の精度の比較

オドメトリにより構築した室内および廊下の環境地図を Fig.6 に示す。ここで、原点はロボットのスタート位置を表し、ロボット正面が Y 方向、側面が X 方向である。黒の実線で描かれた矢印は時系列データ取得終了時のロボ

ットの姿勢を表す。また青い点は環境地図に登録されたスキャンを表し、赤い点はロボットが現在取得しているスキャンを表す。Fig.6より、オドメトリにより構築された環境地図はモデル化誤差やスリップなどのオドメトリ誤差が原因となり崩壊してしまっている。そのため、動作計画等への利用は困難である。

ICP スキャンマッチングにより構築した室内および廊下の環境地図を Fig.7 に示す。LRF は URG を使用した。なお、対応点の探索処理(誤った対応点の除去処理)で用いる初期許容距離および初期許容角度については、予備実験で最良の結果が得られた 500[mm]および 5[deg]を採用した。

構築した環境地図はオドメトリによる手法と比べて精度が高く、オドメトリ誤差を修正できていることがわかる。しかし、オドメトリ誤差を完全に除去することはできず、広大なループ構造を持つ廊下環境では誤差の累積により正確な環境地図構築ができていない。

モンテカルロ法により構築した室内および廊下の環境地図を Fig.8 に示す。ICP スキャンマッチングと同様に、予備実験より対応点除去処理に用いる許容距離および許容角度の値はそれぞれ 50[mm], 1[deg]とした。また、パーティクルは 1000 個とし、重み付け処理の尤度計算に用いるパラメータ  $\sigma$  の値は 0.0015 を採用した。

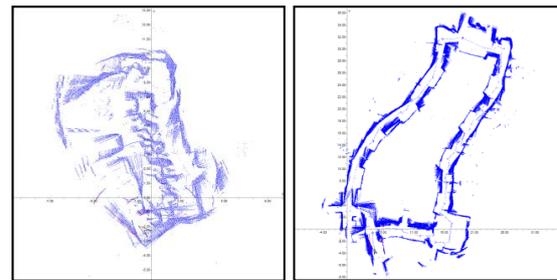
構築した環境地図は ICP スキャンマッチングに次ぐ精度であり、オドメトリ誤差の修正がきている。しかし、ICP スキャンマッチングと同様にループ構造の解決ができていない。

以上より、環境地図の精度は ICP スキャンマッチングによる SLAM が最良の結果となり、次いでモンテカルロ法、オドメトリの順であった。以降の実験では、ICP スキャンマッチングによる SLAM で構築した環境地図を用いて評価を行う。

### 5.3 LRF の違いによる環境地図の精度の比較

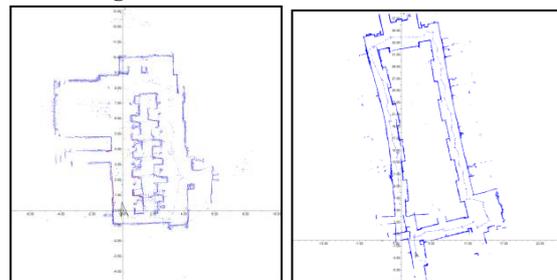
使用する LRF を UTM に変更し、ICP スキャンマッチングにより構築した環境地図を Fig.9 示す。

Fig.7 と比較すると、UTM を使用した場合は廊下環境の地図の歪みが小さくなっていることがわかる。最大測定距離が 4[m]から 30[m]へと広がったことにより、取得できる特徴点が増え、直線部の正確なマッチングが行えたためと考えられる。しかし、誤差により一部マッチングに失敗し、それが原因となり環境地図の後半部にブレが生じている。



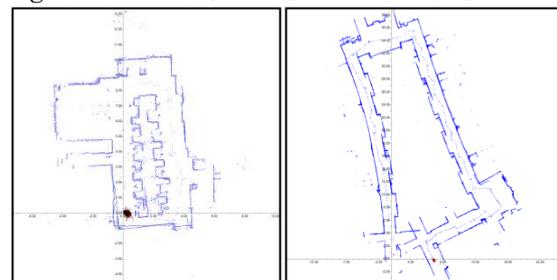
(a) 室内環境 (b) 廊下環境

Fig. 6 オドメトリによる SLAM



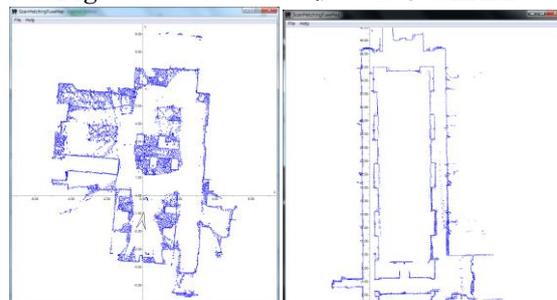
(a) 室内環境 (b) 廊下環境

Fig. 7 ICP スキャンマッチングによる SLAM



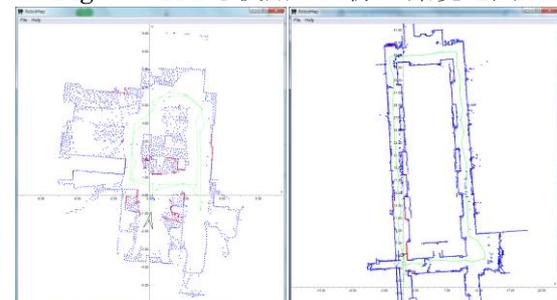
(a) 室内環境 (b) 廊下環境

Fig. 8 モンテカルロ法による SLAM



(a) 室内環境 (b) 廊下環境

Fig. 9 UTM を使用した際の環境地図



(a) 室内環境 (b) 廊下環境

Fig. 10 提案手法による SLAM

## 5.4 提案手法により構築した環境地図の精度の比較

提案手法により構築した環境地図を Fig.10 に示す. ここで緑の点はロボットの軌跡を表す. Fig.9 と比較すると, 室内環境では ICP スキャンマッチングにより構築した地図と同様に高い精度で地図が構築されている. 幾何学的特徴が多く, 比較的短い距離を走行した場合, 誤差の蓄積により環境地図の構築に失敗することは少ないと言える.

廊下環境では環境地図の後半に生じていたブレが解消されている. これは, 自己位置推定を複数回行うことで, より正確な自己位置を獲得したためと考えられる. これにより正確な環境地図の構築が行われ, 精度が向上していると考えられる. しかし, 誤差を完全に消しきることはできず, わずかな歪みが残っている.

## 6. 広域な環境における歪み補正処理

広域な環境で地図の構築を行った場合, センサ誤差やモデル化誤差の蓄積により地図に歪みが生じる. そのため, ループ環境では実際には元の位置に戻ってきているにもかかわらず, 以前計測した物体が別の物体として地図へ登録されるといった問題が生じる. そこで本研究では, 地図の歪み補正処理について検討を加えた.

### 6.1 歪み補正処理

Fig.11 に環境地図構築時の ICP スキャンマッチングおよびオドメトリの移動量を示す. ここで, 横軸はデータ No.(200[ms]間隔で取得)である. また, 青い棒グラフが ICP スキャンマッチングによる移動量, 赤い棒グラフがオドメトリによる移動量を示している. ICP スキャンマッチングによる移動量が増加している点は, 蓄積したオドメトリ誤差が大幅に修正された箇所を意味している.

本研究では ICP スキャンマッチングによる移動量に閾値を設け, 移動量が閾値以上となった場合に歪み補正処理を実行する. 処理の流れを以下に示す.

- ①移動量の算出および閾値による判定
- ②移動量の差分計算
- ③各座標の補正量計算
- ④座標更新

誤差は時間経過に伴って増加する. したがって, 各座標の補正量は探索開始地点における補正量が零になるように(5)式の線形関数を用いて計算した.

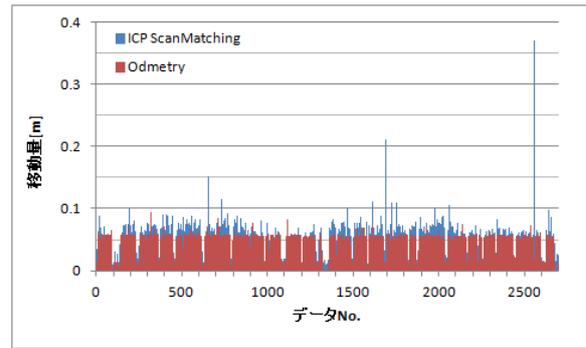
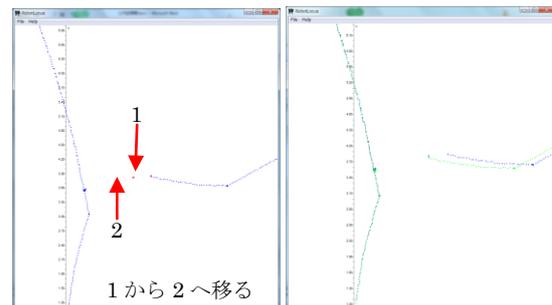


Fig. 11 ICP スキャンマッチングとオドメトリの移動量



(a) 処理前の軌跡 (b) 処理後の軌跡

Fig. 12 ロボットの軌跡に対する補正処理



(a) 処理前の軌跡 (b) 処理後の軌跡

Fig. 13 C 点の拡大図

$$E_i = \frac{diff}{N} i \quad (5)$$

ここで,  $E_i$  はデータ No.  $i$  番目における補正量,  $diff$  はスキャンマッチングおよびオドメトリ移動量の差分,  $N$  は現在までのデータ数である.

### 6.2 実験および考察

実験環境は広域かつ幾何学的特徴の乏しい廊下(ループ構造)を対象とした. 提案手法の有用性を評価するために, 環境地図を構築した際のロボットの軌跡に対して歪み補正処理を行った. なお, ICP スキャンマッチングの移動量に関する閾値を 0.15[m], 0.20[m], 0.25[m] と変化させ実験を行った.

Fig.12(a)は補正処理前のロボットの軌跡を

示し、同図(b)は閾値を 0.15[m]とした場合の補正処理後のロボットの軌跡を示したものである。また、Fig.13(a)は補正処理前の C 点の拡大図を示し、同図(b)は補正処理後の C 点の拡大図を示している。Fig.12(a)の A, B の点では y 軸方向へ歪みが生じ、C 点では x 方向へ歪みが生じている。閾値を 0.20, 0.25[m]とした場合、補正処理は C 点で 1 回実行された。この場合、C 点は正確に補正されたものの、A, B 点は x 軸方向へ補正されてしまい誤った補正が行われた。一方、閾値を 0.15[m]とした場合、補正処理は各点で 1 回ずつ、計 3 回実行された。この場合、A, B, C 点とも歪みの生じた軸方向へ修正され、補正処理による軌跡全体の歪みも認められなかった。

## 7. おわりに

本研究では静的な環境を移動するロボットの SLAM 手法として、3 種類のオンライン手法による環境地図構築を行い、その精度を比較した。さらに、3 種類の SLAM 手法を組み合わせた手法を提案し、環境地図の構築を行った。その結果、室内では高い精度で地図を構築することができた。また、廊下環境では ICP スキャンマッチングにより構築した環境地図と比較し、精度を向上させることができた。

また、広域な環境で構築した地図の歪み補正処理について検討を行い、実験による評価を行った。その結果、ICP スキャンマッチングによる移動量の閾値を 0.15[m]とした場合において最も精度よく歪み補正を行うことができた。

今後は、ロボットの軌跡に対して行った歪み補正処理を構築した環境地図へ実装し、改善を行う予定である。また、動的な環境に対して SLAM 手法を適用し、その問題点を検証する予定である。

## 参考文献

- 1) 日本機械学会編:「機械工学便覧 γ 編 メカトロニクス・ロボティクス」, 日本機械学会, (2008)
- 2) S. Thrun:「Robotic Mapping: A Survey」, Exploring Artificial Intelligence in the New Millenium, Morgan Kaufmann, (2002)
- 3) S. Thrun, W. Burgard, D. Fox:「Probabilistic Robotics」, The MIT Press, (2005)
- 4) 前山 祥一, 大矢 晃久, 油田 信一:「移動ロボットの屋外ナビゲーションのための

オドメトリとジャイロのセンサ融合によるデッドレコニング・システム」, 日本ロボット学会誌, vol.15, no.8, pp.84-91,(1997)

- 5) P. J. Besl and N. D. McKay:「A Method for Registration of 3-D Shapes」, IEEE Trans. On PAMI, vol.14, no.2, pp.239-256, (1992)
- 6) 北側 源四郎:「モンテカルロ・フィルタおよび平滑化について」, 統計数理, vol.44, no.1, pp.31-48, (1996)