

## L0 ノルム最小化と多重解像度法を用いた 室内向け自律移動システムの設計と開発

中沢 実<sup>†</sup> 杉下 大河<sup>†</sup> 阿部 拓真<sup>†</sup>

**概要:** 平成 26 年現在, 車椅子利用に関わる死傷事故が年間 200 件以上発生している[1]. そのうち 9 割以上が 65 歳以上の高齢者で占めており, 5 割以上が電動車椅子を利用した際のものである. 本研究では最終的に, 電動車椅子利用における事故の低減を目的としている. 車椅子利用時の事故は車椅子利用者の判断能力, 危険予知能力の衰えからくる誤操作や反応遅れなどによるものである. 本論文は, この部分を補助するとともに, 目的地まで自律的に走行するシステムを報告する. 自律走行時の自己位置推定や障害物回避に軽量なアルゴリズムとして L0 ノルム最小化と多重解像度法を使用した. これにより組込用途の小型計算機でも運用可能であり, 即応性が高いリアルタイムな移動支援システムを実現できる.

**キーワード:** 電動車椅子, 自律移動システム, L0 ノルム最小化, 多重解像度法,

### A Study of Inside Navigation System using L0 Norm Minimization Method and Multi Resolution Method

MINORU NAKAZAWA<sup>†</sup> TAIGA SUGISHITA<sup>†</sup>  
TAKUMA ABE<sup>†</sup>

**Abstract:** More than 200 a year casualties when doing wheelchair use, occur in 2014. More than 90 percent occupy it by a senior citizen beyond 65 years old soon, and more than 50 percent are something when using an electric wheelchair. Finally, we aim in this research purpose which is reduced to an accident in electric wheelchair. An accident at the time of wheelchair use is chosen as the wrong operation and the reaction delay reeled off a decline of the judgment ability of the wheelchair use person and the dangerous forecast ability. This paper describes that help automatic operation as well as report the system run in self-controlled way to the destination. Using L0 norm minimization and Multiresolution algorithm for obstacle avoidance and self-localization for autonomous driving. By using this method, including small computer became operational. Further, we realize real time mobile support system

**Keywords:** Electronic Wheel Chair, Autonomous Driving System, L0 Norm minimization method and Multi resolution method

#### 1. はじめに

平成 26 年現在, 車椅子利用に関わる死傷事故が年間 200 件以上発生している. そのうち 9 割以上が 65 歳以上の高齢者で占められており, 5 割以上が電動車椅子を利用した際のものである. 車椅子利用時の事故は車椅子利用者の判断能力, 危険予知能力の衰えからくる誤操作や反応遅れなどによるものである. また, 平成 25 年に超高齢化社会となった日本においては, 益々, 人の移動を支援する必要がある.

屋内外を自律的に移動するロボットとして, 無人搬送車 (AGV)などが実用化されている[2]. しかし, 製造工場内などの人が立ち入ることが少ない特殊な環境下で運用されることが多い. 人が移動する手段として, ジョイスティックなどで操作する電動車いすは多く存在するが, そのジョイスティックを操作することができない人向けとなると, その

実用例はほとんど存在しない. そこで本研究では, 始めて訪れる環境であっても, 場所さえ指定することができればその場所まで自律的に移動する車いすロボットを開発する. また, 高齢化や身体障害などにより運動機能が低下した人々の多くは自立を望んでいる. そこで自律的に移動しながらも利用者の意志を反映できるシステムとするため, 完全自律移動と併用しつつ, 移動を補助することが可能な仕組みが必要である.

以前の研究では, 過去に障害者向けの自律移動車椅子について研究が行われた[3]. この研究では脳波活動より, 目的地を推定した後, 目的地まで完全自律走行するというものであった. しかし, 位置推定に用いたアルゴリズムの計算コストが大きく, 高性能な計算機を必要とした. また, 自律走行する車いすの挙動は搭乗者には予測できず, 極端に壁面寄りを行ったり, 障害物を回避するものの, 対応が遅かったりと, 搭乗者にとって恐怖を覚える問題があっ

<sup>†</sup> 金沢工業大学  
Kanazawa Institute of Technology

た。

本研究では、これら問題点を解決するために、計算コストが低く、十分な精度で自己位置を推定できるアルゴリズムを用いる。また、ジョイスティックなどのユーザインターフェイスで自律走行時の速度や走行方法を自在に変更できる機能の開発を目標とした。

従来手法では、ロボットは走行時に LRF より取得する環境の概形を、環境地図とマッチングすることにより自己位置を推定する。本研究では、マッチングによく用いられる高コストな ICP アルゴリズムやパーティクルフィルタなどを使用せず、L0 ノルム最小化法と多重解像度法を合わせて用いる。L0 ノルムとはマッチする地図の点の個数と定義され、L0 ノルム最小化法は動的な変化にロバストな手法として注目されている。しかし L0 ノルムを最小化する際には、任意の姿勢に回転と平行移動を行う総探索でマッチ点を探索することが多く、計算コストがかかる。そこで、この探索に、多重解像度法を用いる。あらかじめ複数の解像度な環境地図を生成し、自律走行時に生成する局所環境地図も同様に複数生成する。比較するマッチ点数の少ない低解像度地図では高速に L0 ノルム最小化を計算することができるが、空間分解能が低いため、誤差を含む。この誤差を含んだ自己位置を次の段階の環境地図の初期位置として引き継ぎ、最終的に高解像度な環境地図で高精度な自己位置を求める。本研究では環境地図の 1 ピクセルが実環境の 200 mm, 100 mm, 50 mm に対応する 3 組の環境地図を用いた。

## 2. 自律移動型車椅子

### 2.1 解決すべき問題のためのタスクの定義

本研究において車いすロボットとは、主に屋内環境で、歩行者などが行き交う通常の環境を走行し、予め設定した目的地へ人を自律的に移送するものである。また、対象を屋内環境と限定している。これを全ての環境とすると、地球規模での大域的な自己位置推定が必要となり、起こりうる事象をすべて想定することが困難となる。そのため、屋内環境とあるが、後述の自己位置推定のための環境地図が生成可能な環境であれば屋外(建物の外で屋根がなく開けた地)でも走行可能であるとする。

開発した車いすロボットの一連の動作について述べる。移動の支援が必要な人が車いすロボットに搭乗し、行き先指定を行う。目的地が指定されると車いすロボットは目的地へ向け走行を開始する。走行中は、経路中にある障害物は自動に検出して回避し、緊急時は非常停止する。途中、ゆっくり又は早く目的地へ向かいたい等の際に、走行速度をジョイスティック操作により調節可能で、通路の中央や壁際を走行したいなどの走行方法も自由に操作することができる。そして目的地まで走行することでタスクは完了となる。

## 2.2 自律走行

車いすロボットには、目的地へ向けて自律的に走行する機能が必要である。そのためには車いすロボット自身が、現在どの位置に存在し、どの方向を向いているのかという姿勢を認識する必要がある。これを自己位置推定という。また、人間の存在する環境を走行するため、人間に危害を加えたり、恐怖を覚えさせたりしてはならず、安全性が求められる。そのため、人や障害物を自動で検出して、回避行動や一次停止などの行動をとる必要がある。

## 2.3 走行環境

図 1 に想定する走行環境を示す。走行環境は金沢工業大学八束穂リサーチキャンパス情報科学研究所 67 号館 3 階全域であり、環境中には目的地の候補となる各部屋やエレベータ、トイレなどが存在する。

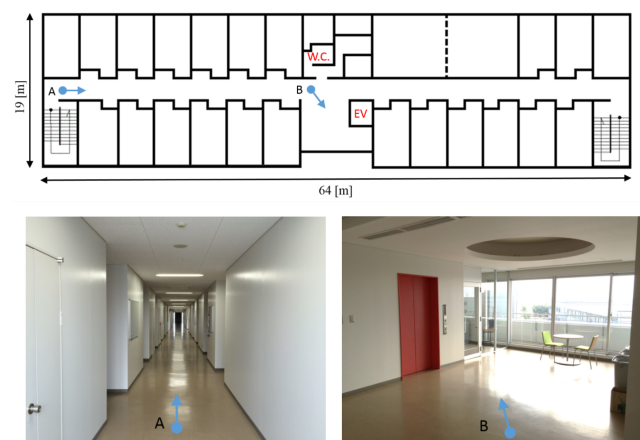


図 1: 走行環境

Figure 1: Driving Environment

## 3. システムならびにハードウェア構成

図 2 に本研究で構築しているシステムのハードウェア構成を示す。本システムでは、移動ベース(電動車椅子)[4]、各種センサ、走行制御 PC、車体制御用マイクロコントローラから構成される。

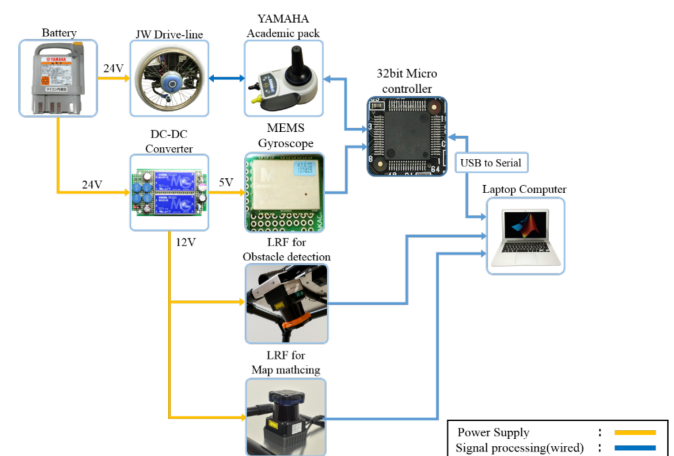


図 2: ハードウェア構成

Figure 2: Configuration of Hard-Ware

ロボットに用いるセンサは大きく分けて内界センサと外界センサに分けられる。外界センサは、ロボットの外界の情報を知るためのものであり、カメラや超音波センサなどがある。カメラには画像として周囲の情報を取得し、超音波センサ[6]などは周囲の物体までの距離情報を取得することができる。対して内界センサはロボット内部の情報を得るためのセンサである。これは角速度を測定するジャイロセンサや、車輪の回転数を測定するロータリーエンコーダなどがある。本研究で用いるセンサは、JW アクティブのホイールモーター内蔵のロータリーエンコーダと、車体の旋回中心に設置した1軸ジャイロセンサ、センサ架台に設置したマップマッチ用の測域センサ1台と障害物検出用の測域センサ1台である。

#### 4. 自己位置推定

本研究では、車いすロボットの自律走行に環境地図ベースの自己位置推定を用いる。環境地図とはロボットを運用する環境の壁や障害物の位置関係を記述したものであり、本研究では占有格子地図を用いた。

占有格子とは、環境を量子化した領域に障害物が占有する可能性を確率的に表したものである。これを可視化した際、輝度が高く、白く見える領域ほど障害物の占有する可能性が低いことを示す。車椅子ロボットが自律走行する際は、環境地図と LRF から得られる実際の環境の概形をマッチングさせることにより自己位置を推定する。このマッチング工程には ICP アルゴリズム[9]やパーティクルフィルタなど様々な手法が提案されているが、いずれも計算負荷が高く、処理に高性能な計算機を必要とする。

そこで本研究では  $L_0$  ノルム[10]を最小化するマッチング手法と多重解像度法[11]を併用することで、高速かつ動的環境に頑健なマッチング手法を実現する。

##### 4.1 既存手法

ノルム(norm)とは距離を意味し、ICP アルゴリズム等は  $L_2$  ノルム(2乗距離)を最小化することでマッチングを実現することが多い。ここで予め生成した大域環境地図の点群を  $\{p_i\}$ 、自律走行時に取得する局所環境地図の点群を  $\{q_j\}$  と仮定する。 $\{p_i\}$  と  $\{q_j\}$  には位置は異なるものの、対応点が存在する。ロボットの自己位置は  $\{q_j\}$  を  $\{p_i\}$  に重ね合わせる事により推定される。以下の評価関数  $E(R,t)$ (式 1)を最小化することで最適解を求めることができる。ここで、 $\{q\}$  の定義を式(2)に示す。

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

$$\{q\} = \sum_{j=1}^n q_j \quad (2)$$

$R$  は  $2 \times 2$  の回転行列(式 2)であり、 $t$  は 2次元のベクトル

(式 3)であり、 $n$  は点群の総数である。それぞれが 2次元平面でのロボットの回転と平行移動に対応する。 $\{p_i\}$  中のある点  $p_i$  が、 $\{q_j\}$  中のある点  $q_j$  に対応すると仮定すると、式 (1)における関数  $f$  は式 4 で表される。式 4 の関数  $f$  は  $p_i$  と  $q_j$  のユークリッド距離を意味し、ICP アルゴリズムではこの距離の最小化を繰り返し、重ね合わせのためのパラメータを求める。

$$R = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix} \quad (3)$$

$$t = \begin{pmatrix} x \\ y \end{pmatrix} \quad (4)$$

$$f(p_i, \{q_j\}) = \|p_i - q_j\|^2 \quad (5)$$

##### 4.2 $L_0$ ノルム最小化マッチング

$L_2$  ノルムを用いたアルゴリズムでは点間の距離を評価の対象とするため、動的物体が多い実環境でのロボットには適さない。また、解が収束するまで反復計算を行う ICP アルゴリズムでは、計算量が不定である。そのため、リアルタイム性が求められるロボットでは、ロボットの制御周期内に計算が完了する保証が無いため適さない。そこで動的物体に対して頑健である手法として、 $L_0$  ノルムを最小化する手法を用いる。

$L_0$  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個数として定義される。この対応とは近傍点の有無である。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  $E(R,t)$  は定義されるが、関数  $f$  は異なり、以下ようになる。

$$f(p_i, \{q_j\}) = \begin{cases} 0 & (\exists j, \|p_i - q_j\| \leq \epsilon) \\ 1 & (\text{otherwise}) \end{cases} \quad (6)$$

ここで  $\epsilon$  は距離の閾値であり、任意に定める実数である。この  $\epsilon$  の値は、後ほど述べる地図の解像度に対応する。 $L_0$  ノルムを算出する際は、お互いの点の ID を必要とせず、 $\pm\epsilon$  以下の点の有無のみで、対応点の存在を求める。マッチングには  $\{q_j\}$  を任意の姿勢に回転と平行移動を行い、 $L_0$  ノルムを算出する。このとき  $\pm\epsilon$  以下の最近傍点の探索に最も計算コストがかかり離散最適化問題に陥るため、解像度と探索範囲を限定して総探索することがよく用いられる。

##### 4.3 多重解像度法による高速化

最近傍点探索には、最近傍候補点を探索して、その点に対するユークリッド距離を求めて評価を行う。総当たり法での最近傍候補点の個数を  $m$  とすると、探索に必要な計算コストは  $O(m)$  である。対して ICP アルゴリズムでよく用いられる k-d 木法では  $O(\log m)$  となる。そこで本研究では、多重解像度法を用いて高速化を図る。多重解像度法とは、複数の解像度の環境地図を用いる手法である。低解像度地

図では、最近傍候補点の個数が少なく、 $L_0$  ノルム最小化における、総当たり法を用いても高速に探索できる。

しかし、地図の空間分解能が低いため、実環境との誤差は大きい。そこで低解像度地図で求めた自己位置 $(x,y,\theta)$ を高解像度地図へ引き継ぎ、高解像度地図で狭い範囲を細かい間隔で探索することにより、全体的な計算量を抑えつつ高速に解を求める。本研究では、環境地図の1グリッドを実環境の200mm<sup>2</sup>、100mm<sup>2</sup>、50mm<sup>2</sup>に対応した3組の解像度が異なる環境地図を予め生成する(以下、それぞれ200mm<sup>2</sup>地図、100mm<sup>2</sup>地図、50mm<sup>2</sup>地図とする)。解像度が最も低い環境地図では、x方向、y方向とも-100mm~100mmの範囲を20mm間隔で、姿勢(方向)は-45deg~45degの範囲を5deg間隔で探索した。この際の誤差はx方向y方向に±100mmであり、姿勢は±2.5degとなるため、次の段階の解像度の環境地図ではこの誤差の範囲内で探索を行う。最終的にはx方向、y方向に±25mm、姿勢±0.625degの精度で探索することができた。この3段階の環境地図を用いた探索に要する時間は17msほどであった。図2に多重解像度法使用のイメージを示す。

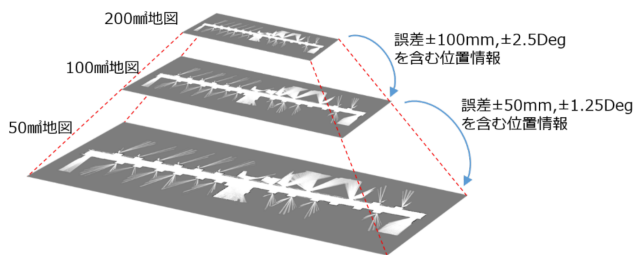


図 2: 自己位置推定のための多重解像度法

Figure 2: Multi Resolution Method for Self Location Estimate

#### 4.4 自己位置推定法の評価

以上で述べた手法で評価を行う。図3、図4、図5にそれぞれの解像度ごとのマッチング前とマッチング後の結果を示す。点群 $\{p_i\}$ と $\{q_j\}$ を各解像度で生成してマッチングした。初期位置としてx方向、y方向共に1m、-45degの誤差を持ったマッチング点群 $\{q_j\}$ を与えたところ、マッチングは成功し、想定誤差範囲内に収まることが分かった。200mm<sup>2</sup>地図にて大まかに地図がマッチングされているが、誤差が残っていることが分かる。100mm<sup>2</sup>地図でのマッチングでほぼマッチングされ、50mm<sup>2</sup>地図では回転成分のみのマッチングとなった。

200mm<sup>2</sup>地図のマッチングに要した時間は約9ms、100mm<sup>2</sup>地図のマッチングに要した時間は約5ms、50mm<sup>2</sup>地図のマッチングに要した時間は4msであった。

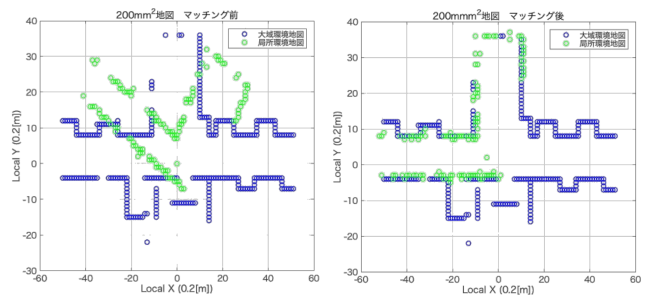


図 3: 200mm<sup>2</sup>地図におけるマッチング結果

Figure 3: Matching Result of 200mm<sup>2</sup> map

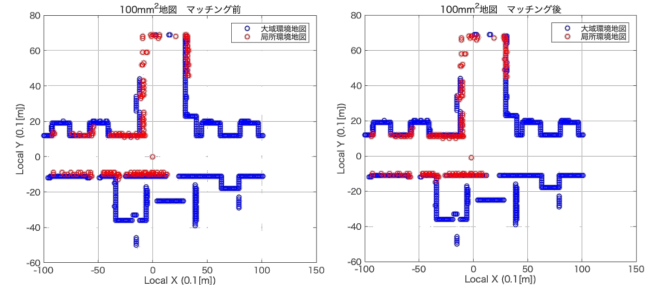


図 4: 100mm<sup>2</sup>地図におけるマッチング結果

Figure 4: Matching Result of 100mm<sup>2</sup> map

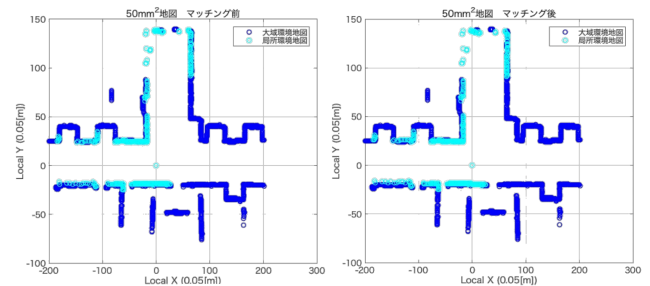


図 5: 50mm<sup>2</sup>地図におけるマッチング結果

Figure 5: Matching Result of 100mm<sup>2</sup> map

#### 4.5 ジャイロオドメトリ

本節では内界センサであるロータリーエンコーダ値とジャイロセンサを用いた自己位置推定法を述べる。移動ロボットの現在の位置や姿勢を推定するのによく用いられる本手法は、車体の速度を積分するものである。ここでロボットの移動開始時間を $t_0$ とする。その後の時刻 $\tau$ でのロボットの進行方向の速度を $v(\tau)$ 、回転角速度を $\omega(\tau)$ とする。すると、以下の式にてロボットの現在位置 $(x,y)$ と姿勢 $(\theta)$ を求めることができる。

### 5. 環境地図

本研究では環境地図ベースの自律走行を行うことは述べた。ここで、ロボットを運用する環境の全域を環境地図としたものを大域環境地図とする。車いすロボットが自律走行する際に作成する環境地図を局所環境地図とする。以降はこの定義で述べる。

### 5.1 局所環境地図の作成

図6に局所環境地図の生成方法を示す。局所的な占有格子地図を生成するため、グリッドの初期状態を中央値で初期化する。高い重みのグリッドがレーザ通過可能な範囲であり、低い重みのグリッドであるほど障害物が占有する可能性が高い事を示す。使用したLRFは視野角270度、方向分解能が0.25度であり、1080点の距離を一度に測距可能である。1秒間あたり43,200点の距離情報を取得することが可能であるが、ロボットの制御周期内に扱えるデータ量とするために、この点群をフィルタ処理して削減する。

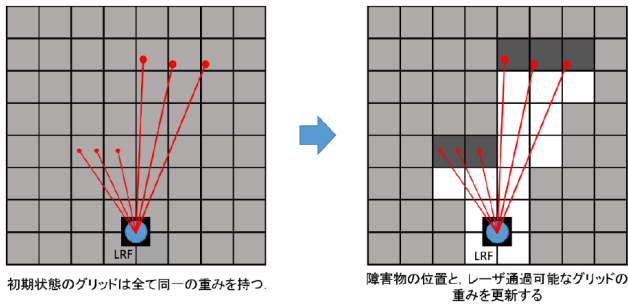


図 6: 局所環境地図の生成

Figure 6: Local Environment Map

LRFの特性上、点群はLRFから遠ければ遠いほど疎になり、また赤外線レーザのスポットが広がる。また、距離の誤差の影響を多く受けるため、LRFから任意の距離R以下の点群のみ抽出する処理を行う。本実験ではRを10mとした。点群が密すぎると、非常に近い範囲の距離情報を持ったデータが多くなるため、2次元空間をグリッド状に区切り、同一グリッドの代表値を使用する。図7に実際の環境で生成した局所環境地図の1例を示す。

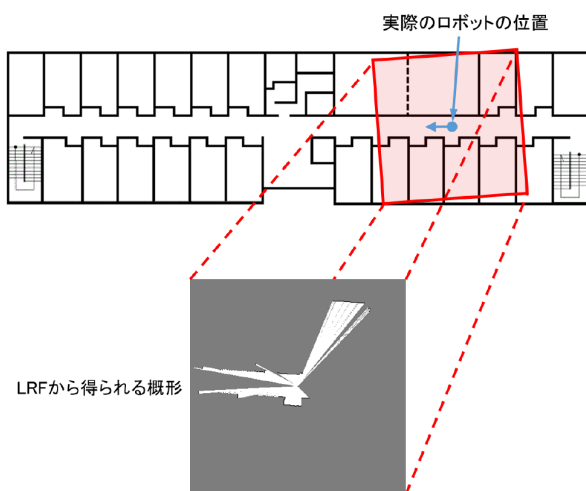


図 7: 実環境と局所環境地図の対応

Figure 7: Mapping between real environment and local environment map

### 5.2 大域環境地図の生成

大域環境地図は、局所環境地図を連続的に合成することで生成する。あらかじめ、車いすロボットを手動にて操作

し、走行環境の局所環境地図を生成する。図8に示すように時刻 $t-1$ に取得した環境地図を $m_{t-1}$ 、時刻 $t$ に取得した環境地図を $m_t$ とする。この二つの局所環境地図を $L_0$ ノルム最小化法と多重解像度法で処理すると、モデルデータとなる $m_{t-1}$ に、 $m_t$ を重ね合わせる事のできる変換パラメータ $(R,t)$ が求められる。求めたパラメータで $m_t$ を変換すると、 $m_{t-1}$ には無い領域が $m_t$ に含まれる。これを新しい環境地図として追加し、環境地図全体を更新していく。以降は更新された環境地図をモデルデータとして更新を繰り返し、走行環境全体の環境地図を生成する。また、各マッチングの場面で求められる変換パラメータ $(R,t)$ を累積していくことで、自己位置を推定することができる。

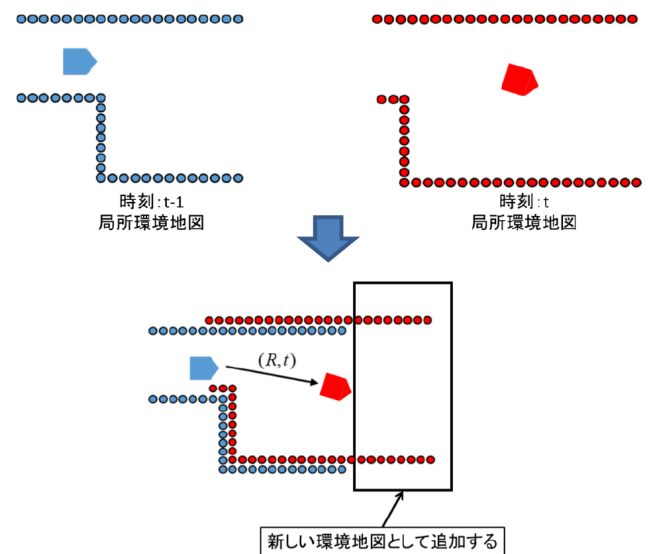


図 8: 環境地図の更新

Figure 8: update environment map

4.3章で述べたように、多重解像度法で用いる低解像度地図は1グリッドが $200\text{mm}^2$ である。そこで、ジャイロオドメトリより、車いすロボットの位置が $\pm 200\text{mm}$ 以上、又は姿勢が $\pm 5\text{deg}$ 以上変化したときに上記の自己位置推定を行うようにした。図9に実環境と、生成した環境地図を示す。

この環境地図は $50\text{mm}^2$ 地図であり、この地図を縮小することで $100\text{mm}^2$ 地図と $200\text{mm}^2$ 地図を生成する。この際、単純に地図画像を縮小しては、地図の幾何学的特徴が失われることがある。そこで $50\text{mm}^2$ 地図における障害物のグリッドの座標値のみを0.5倍又は0.25倍し、床関数を用いてダウンサンプリングを行う。最後に、再び環境地図画像に変換した。これにより幾何学的特徴量の消失を最低限に防いだ。

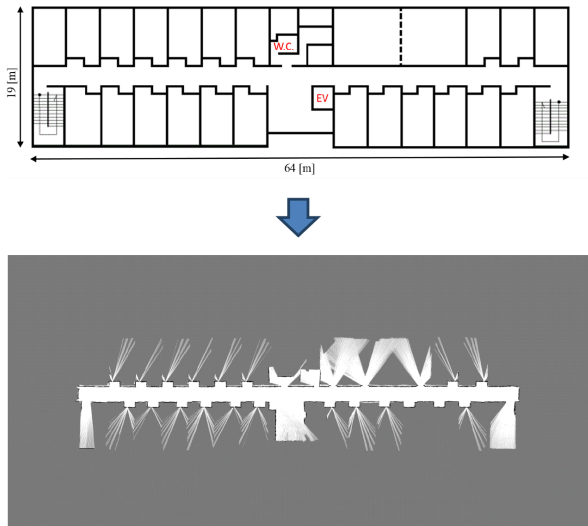


図 9: 大域環境地図

Figure 9: Global Environment Map

## 6. 自律走行

本研究にて車いすロボットの自律走行処理の流れを図10に示す。なお、自律走行時には、環境地図の生成と目的地までの経路設計が完了しているものとする。

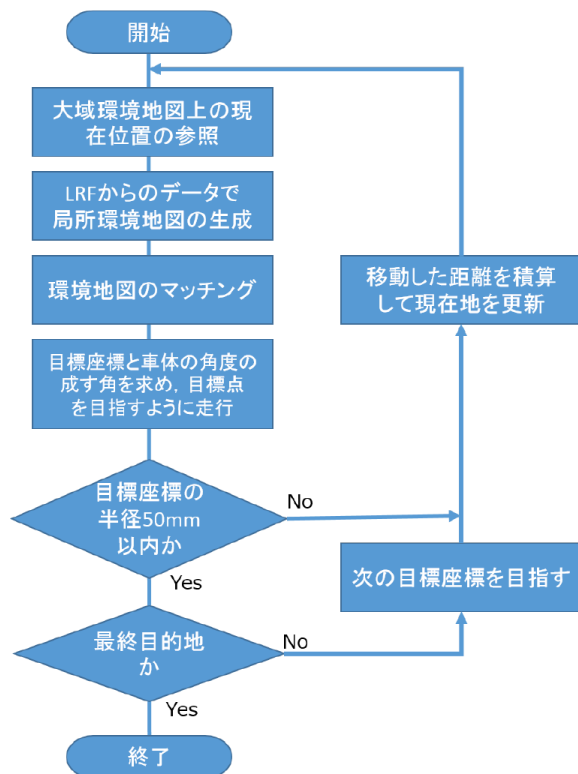


図 10: 自律走行処理の流れ

Figure 10: Flow of Autonomous Navigation

### 6.1 車いすロボットの誘導

車いすロボットの誘導は、環境地図上で定めた経路候補点を追従するように誘導する。経路設計では、環境地図上

にて初期位置、目的位置、初期位置から目的位置に至るまでの通過目標の座標点を設定する。本研究では自己位置と目的座標から目的座標の方向を求め、車いすロボットの姿勢と目的座標の方向との差を無くすように誘導する方式を採用した。ここで車椅子ロボットの座標を  $(x, y)$ 、設計した経路候補点の座標を  $(x_i, y_i)$ 、グローバル座標上でのロボットの姿勢を  $\theta_{GL}$  とすると、目的座標までの角度  $\lambda$  は次式で示される。

$$\lambda = \tan^{-1} \frac{y_i - y}{x_i - x} + \theta_{GL}$$

$\lambda$  が正のとき左車輪を多く回転させて右旋回し、負のときは右車輪を多く回転させて左旋回することにより、目的座標を目指して走行する。しかしこのように単純に両輪の速度制御をしては、ハンチング(目的値付近でのふらつき)が起こり、常に蛇行しながら走行することとなる。そこで車輪の制御にPI制御を用いる。

## 7. 自律走行の評価実験

### 7.1 自己位置推定精度の評価

5章で生成した環境地図を元に、自律走行時の自己位置推定の精度を評価する。図11に手動で設置した経路候補点と、それらを直線で結んだものを示す。エレベータホールから一室への経路とした。

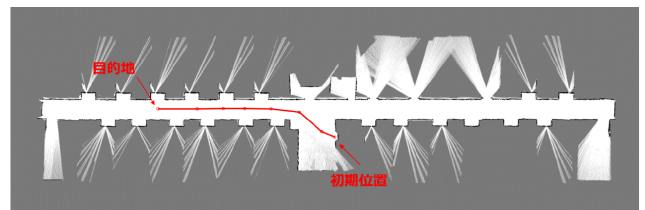


図 11: 評価に用いた走行経路

Figure 11: Driving Route for Evaluation

続いて、図11の経路追従するように、自己位置推定しながら走行した。結果を図12に示す。赤線で示すものが事前に設定する経路であり、青線で示すものが実際の走行軌跡である。なお、青線の太さは実環境での想定誤差に合わせて縮尺を設定している。想定誤差範囲内で自己位置推定が実現しているため、設定した経路と実際の走行軌跡がほぼ重なり合うようにマッチしている。

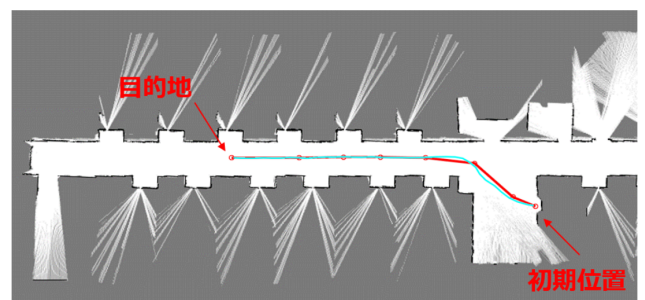


図 12: 静的環境時の走行軌跡

Figure 12: Driving Route (Static Environment)

図 13 に自己位置推定結果の一部を拡大したものを示す。この結果から、安定した自己位置推定が行われていることが分かる。また、空間中にロボットを停止させ、繰り返し停止精度を計測した。結果、想定誤差範囲である  $\pm 25\text{mm}$  以内と、良好な結果となった。

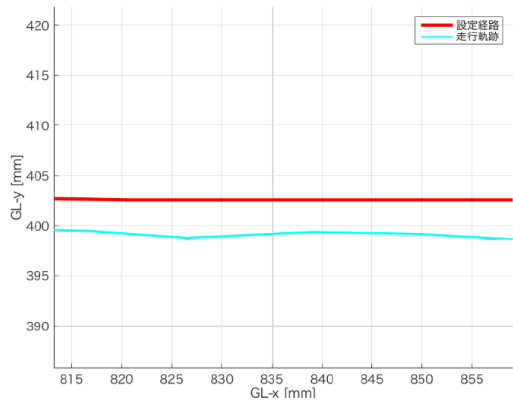


図 13: 自己位置推定結果  
 Figure 13: Result of Self Location Estimation

## 7.2 動的環境下における自己位置推定の評価

5 章で生成した環境地図と同一の環境にて車椅子ロボットを自律走行させた。この際、動的環境とするために、7 名程を、車椅子ロボットの周辺をランダムに歩行させ、動的障害物が存在する環境での自己位置推定の評価を行った。経路設定等は、静的環境で行った方式と同一とした。

図 14 に経路設定と走行軌跡を示す。静的環境を走行した軌跡に対して、動的障害物となった人が影響して自己位置に乱れが生じていることが分かる。しかし概ね自己位置を推定できており、自律走行には問題が無いことがわかる。

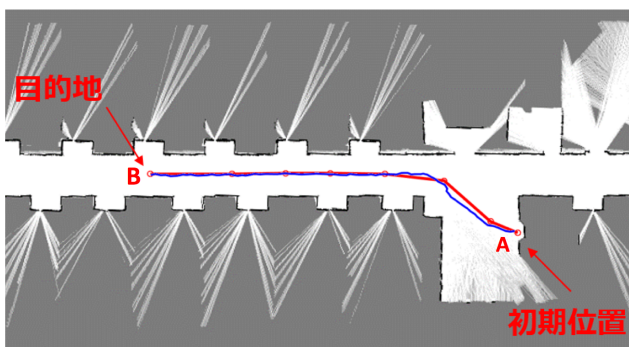


図 14: 動的環境時の走行軌跡  
 Figure 14: Driving Route (Dynamic Environment)

次に動的環境時と静的環境時のマッチング点数について比較する。マッチング点数とは、 $L_0$  ノルム最小化法と多重解像度法を組み合わせた処理で、マッチしたと判断する点群のペアの個数である。この個数が多いほど精度の高いマッチングと言える。図 15 にそれぞれの結果を示す。この結果より、両環境とも、走行開始直後の開けたエレベータホールでは、全体的なマッチング点数が高いことが分

かる。また、動的環境では、動的物体の影響でマッチング点数が大きく上下している。静的環境走行時に対して全体的なマッチング点数は低いものの、自己位置推定が行えている。この結果より、 $L_0$  ノルム最小化法が、動的環境に頑健であるとわかる。

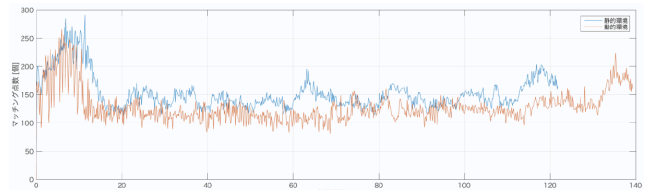


図 15: マッチング点数の比較  
 Figure 15: Comparison number of matching

## 8. 障害物回避

車いすロボットの障害物回避には、センサ架台に設置した LRF を用いる。この LRF は地面から 1.5m の位置に、地面に対して  $40\text{deg}$  下向きに俯角を持たせて固定している。そのため LRF の設置高度以下の障害物を検出することが可能である。また、下り勾配などの走行不可能領域も検出可能である。回避方法は非常に単純で、図 16 右に示すテーブルにて地面と判断される値から外れた領域に障害物があると判断する。LRF をこのように角度を付けて設置することで、高さを持つ障害物を検出することができる。しかし、一瞬しかスキャンされないため、一定時間障害物と検出した FS 座標系で把握し、回避するよう走行する。図 17 に障害物を検出した様子を示す。

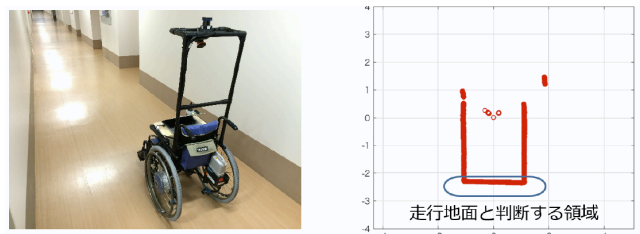


図 16: 障害物を検出していない時の LRF 情報  
 Figure 16: LRF when there is non obstacle



図 17: 障害物を検出した時の LRF 情報  
 Figure 17: LRF when there are some obstacle

## 9. おわりに

本研究では、屋内環境において、目的地まで自律的に走行する車いすロボットの開発を行った。開発した車いすロ

ボットは環境地図上で指定された目的地までの自律走行と、その経路間での障害物回避をタスクとしている。

本論文では開発した車いすロボットのハードウェア構成、自律走行のための自己位置推定手法、実際の自律走行時の各種精度について述べた。ハードウェア構成では、既存の電動車椅子を外部から制御できるロボットへと改造するために搭載した機器や、各種センサ類の搭載方法を述べた。また、ジャイロセンサと車輪内蔵のロータリーエンコーダから自己位置を推定する手法であるジャイロオドメトリをマイクロコントローラ内で実現した。その他、電源装置やセンサ設置架台を開発した。

自律走行のための自己位置推定手法では、 $L_0$  ノルム最小化法に多重解像度法を適応し、高速かつ高精度で、動的環境に頑健なマッチングを行うことに成功した。また、障害物検出と回避機能により、安全な自律走行システムを開発した。

今後の課題として、自律走行までに環境地図を生成し、経路を手動で設定するなど、多くの手作業を自動化することがあげられる。現状では自律走行する環境ごとに時間をかけ、環境地図の微修正を行い、目的地候補となる部屋や施設への経路をあらかじめ設定しておかなければならない。そこで環境地図を生成することに特化したロボットを自律的に動作させ、経路設計を自動的に行うシステムなどを開発することが考えられる。また、環境地図生成と自己位置推定を同時に行う手法である Simultaneous Localization and Mapping (SLAM)を取り入れる必要がある。

## 参考文献

- [1] 最近の交通事故の実態：警察庁  
<[https://www.npa.go.jp/koutsuu/kikaku12/shi\\_04jikojitai.pdf](https://www.npa.go.jp/koutsuu/kikaku12/shi_04jikojitai.pdf)>
- [2] AGV 自律走行型無人搬送車：株式会社シコウ  
<<http://www.shiko.biz/agv/index.html>>
- [3] 鷹箸幸典：“脳波による遠隔操作ロボットの制御に関する研究”，平成 26 年度 金沢工業大学大学院 情報工学専攻 修士論文
- [4] ヤマハ発動機株式会社：JW アクティブ  
<<http://www.yamaha-motor.co.jp/wheelchair/lineup/active/>>
- [5] 米田完，坪内孝司，大隅久：“はじめてのロボット創造設計 改訂 2 版” 講談社，(p.165)，2013。
- [6] 北陽電機株式会社：UTM-30LX-EW  
<<http://www.hokuyo-aut.co.jp/search/single.php?serial=146>>
- [7] Microinfinity Co.,Ltd.：R1350N  
<<http://minfinity.com/eng/page.php?Main=1&sub=1&tab=2>>
- [8] ルネサスエレクトロニクス：SH7125  
<<http://japan.renesas.com/products/mpumcu/superh/shtiny/sh7125/index.jsp>>
- [9] 八木康史，斎藤英雄：“コンピュータビジョン最先端ガイド 3”，アドコムメディア，(p.33)，2010。
- [10] 日永田祐介，末永剛，竹村健太郎，高松淳，小笠原司：“ $L_0$  ノルム最小化を利用した動的な混雑環境下に適応可能なリアルタイム SLAM”，「画像の認識・理解シンポジウム (MIRU2011)」2011 年 7 月
- [11] 榎修一，松本高斉，正木良三，谷口素也：“位置同定コンポーネントの開発と自律移動ロボット Lapi への応用”，電子情報

報通信学会 CNR2011-26(2011-12)

- [12] 阿部拓真，中沢実：“自律搬送車における効率的経路制御の設計と実験”，平成 26 年度電気関連学会北陸支部連合大会講演論文集
- [13] Chang, C. L. and Lee, R. C. T.. Symbolic Logic and Mechanical Theorem Proving. Academic Press, 1973, 331p.