

推薦論文

# $L_0$ ノルム最小化と多重解像度法を用いた 室内向け自律移動車椅子の開発

中沢 実<sup>1,a)</sup> 阿部 拓真<sup>1</sup> 杉下 大河<sup>1</sup> 佐野 渉二<sup>1</sup>

受付日 2017年2月27日, 採録日 2017年9月5日

**概要:** 平成 28 年現在, 車椅子利用に関わる死傷事故が年間 260 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られており, 7 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にしている. 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  $L_0$  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,  $L_0$  ノルム最小化, 多重解像度法

## A Study of Inside Navigation System Using $L_0$ Norm Minimization Method and Multi Resolution Method

MINORU NAKAZAWA<sup>1,a)</sup> TAKUMA ABE<sup>1</sup> TAIGA SUGISHITA<sup>1</sup> SHOJI SANNO<sup>1</sup>

Received: February 27, 2017, Accepted: September 5, 2017

**Abstract:** In 2016, more than 260 accidents involving the use of wheelchairs occur annually. More than 90% of them are elderly people aged 65 and older, and more than 50% are made when using electric wheelchairs. Finally, in this research, we aim to reduce accidents in electric wheelchair use. The accident at the time of using the wheelchair is due to the judgment ability of the user, the erroneous operation resulting from the decline of the danger prediction ability, and the reaction delay. In this paper we report a system that supports this part and runs autonomously to the destination.  $L_0$  norm minimization and multiple resolution method were used as a lightweight algorithm for self position estimation and obstacle avoidance during autonomous driving. As a result, it is possible to operate even a compact computer for embedded applications, realizing a real-time mobility support system with high responsiveness.

**Keywords:** Electronic Wheel Chair, Autonomous Driving System,  $L_0$  Norm minimization method and Multi resolution method

### 1. はじめに

平成 28 年現在, 車椅子利用に関わる死傷事故が年間 260 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, 7 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 [1]. また, 平成 25 年に超高齢化社会となった日本においては, ますます, 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 (Auto Guided Vehicle) などが実用化されている [2]. しか

<sup>1</sup> 金沢工業大学  
Kanazawa Institute of Technology, Nonoichi, Ishikawa 921-8501, Japan

<sup>a)</sup> nakazawa@infor.kanazawa-it.ac.jp

本論文の内容は 2016 年 5 月の第 167 回マルチメディア通信と分散処理研究会にて報告され, マルチメディア通信と分散処理研究会主査より情報処理学会論文誌ジャーナルへの掲載が推薦された論文である.

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椅子は多く存在するが、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、初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椅子ロボットを開発することを目的とする。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、障害者向けの自律移動車椅子について研究が行われた [4]。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、自律走行する車椅子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た。

これらの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た。具体的には、利用者が自律移動車椅子に搭乗しつつ、人の歩く速度 1m/s において、50 msec 以内に自己位置推定処理を完了し、その自己位置精度を、 $\pm 25$  mm 以内であることを目標とした。

また、ジョイスティックなどのユーザインタフェー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 (Laser Range Finder) 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 [5]。

本研究では、マッチングによく用いられる高コストな ICP (Iterative Closest Point) アルゴリズム [6] やパーティクルフィルタ [7] などを使用せず、 $L_0$  ノルム最小化法と多重解像度法を合わせて用いる。 $L_0$  ノルムとはマッチする地図の点の個数と定義され、 $L_0$  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  $L_0$  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  $L_0$  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<sup>2</sup>, 100 mm<sup>2</sup>, 50 mm<sup>2</sup> 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

本研究において車椅子ロボットとは、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、あらかじ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。これをすべての環境とすると、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。

開発した車椅子ロボットの一連の動作について述べる。

- (1) 移動の支援が必要な人が車椅子ロボットに搭乗し、行き先指定を行う。
- (2) 目的地が指定されると車椅子ロボットは目的地へ向け走行を開始する。
- (3) 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。
- (4) ゆっくりまたは早く目的地へ向かいたいなどの際に、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とする。
- (5) 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

車椅子ロボットには、目的地へ向けて自律的に走行する機能が必要である。そのためには車椅子ロボット自身が、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、回避行動や一時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東穂リサーチキャンパス情報技術研究所 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 (a) の地図上にある矢印と記号が、図 1 (b) の画像は車椅子ロボットからの視線となる。

## 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。本システムは、移動ベース（電動車椅子）、各種

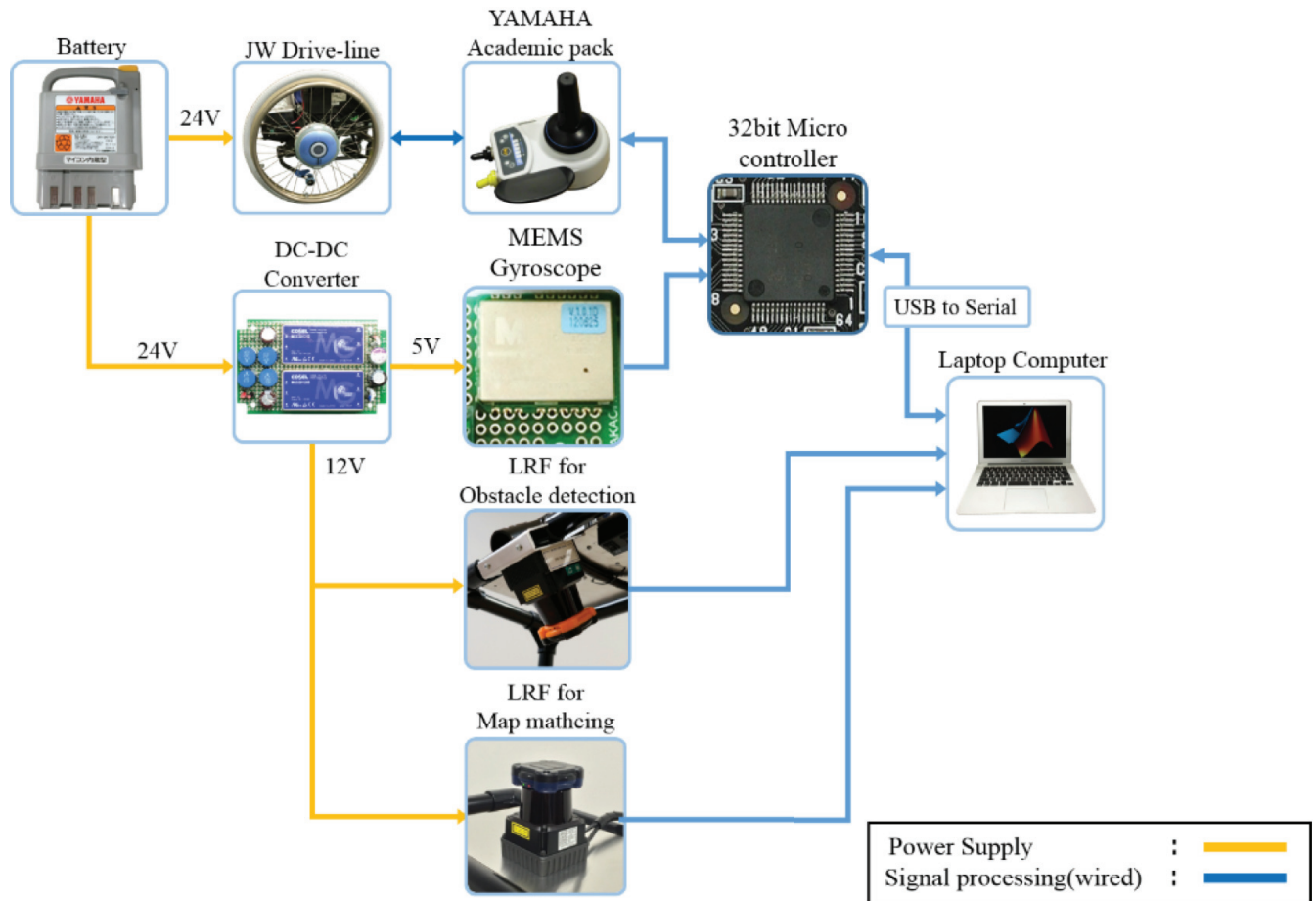


図 2 ハードウェア構成  
Fig. 2 Configuration of hardware.



図 1 走行環境  
Fig. 1 Driving environment.

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。

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，電動車椅子である JW アクティブ [3] のホイールモータ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 [8]。

これらのハードウェアは，1 章で述べた目標を達成するために，各種センサ情報（測域センサ，ロータリーエンコーダ，1 軸ジャイロセンサ）からの情報を取得後，後述する車体制御用マイクロコントローラを経て，位置推定と移動制御用のコマンドを生成するために，50 msec 以内となるように構成しており，自己位置精度についても，測域センサの精度でもある  $\pm 25$  mm 以内を保証できるように選定した。

### 3.1 車体制御用マイクロコントローラ

車椅子ロボットに搭乗する人の体重や，走行環境の床の材質により，駆動モータの負荷が異なる際に，送信した走行速度は同一でも，実際の速度が異なることを実験により確認している。



これより、JW アクティブ内では、モータ回転数を指定速度となるようなフィードバック制御をしていないと推測される。そのため、安定した走行制御を行うため、JW アクティブより得られる現在速度（モータ回転数）をもとに、指定した速度となるようモータ回転数を制御する必要がある。制御 PC でモータ回転数を制御することは可能だが、後述の自己位置推定や障害物回避のために制御 PC のシステムリソースを温存したい。そこで本論文ではマイクロコントローラでモータ回転数を制御する。さらに、ジャイロセンサから得られる角速度と、車椅子から得られる車輪回転数を用いることで、オドメトリを実現する。オドメトリとは、車輪の回転数などにより、ロボットの自己位置を推定する手法であり、本論文ではジャイロからの角度速度を用いるジャイロオドメトリを実現する。一般的にオドメトリの計算には、実時間処理が重要であり、リアルタイム制御が可能なマイコンであることが都合が良い。ジャイロオドメトリに関しては 4.1 節で述べる。

#### 4. 自己位置推定

本研究では、車椅子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 [6], [9] やパーティクルフィルタ [7] 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 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、以下の式で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。

$$\theta(\tau_N) = \omega(\tau_N)\Delta t + \theta(\tau_{N-1}) \quad (1)$$

$$x(\tau_N) = v(\tau_N) \cos(\theta(\tau_N))\Delta t + x(\tau_{N-1}) \quad (2)$$

$$y(\tau_N) = v(\tau_N) \sin(\theta(\tau_N))\Delta t + y(\tau_{N-1}) \quad (3)$$

この計算は 3 章のシステム構成で述べた車体制御用マイクロコントローラ内で処理する。JW アカデミックバックから取得できる車椅子ロボットの車輪回転速度は 50 ms 間隔であるので、以上の式の  $\tau$  の計算は 50 ms 間隔での実行となる。この時間内において自己位置の推定を実現することで、高精度かつ安定性の高い自己位置推定システムが実現できる。この時間を超えてしまう場合は、車椅子の位置座標がずれてしまい復帰に時間がかかってしまい、目的地まで自律的に到達できない可能性が高くなってしまふことが考えられる。

また、姿勢を求める式 (1) の  $\theta$  を左右の車輪速度差と、車輪のトレッド（両車輪間の間隔）から回転成分を求めており、これでジャイロオドメトリを実現している。

##### 4.2 既存手法

ノルム (norm) とは距離を意味し、ICP アルゴリズムなどは  $L_2$  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。ここであらかじ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  $p_i$  に重ね合わせるにより推定される。

以下の評価関数  $E(\mathbf{R}, \mathbf{t})$  (式 (4)) 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、 $\{q_j\}$  の定義を式 (5) に示す。

$$E(\mathbf{R}, \mathbf{T}) = \sum_{i=1}^n f(\mathbf{R}p_i + \mathbf{t}, \{q_j\}) \quad (4)$$

$$\{q_j\} = \sum_{i=1}^n q_{ij} \quad (5)$$

ここで、 $n$  は、点群  $\{p_i\}$  の点の総数となり、 $\mathbf{R}$  は、 $2 \times 2$  の回転行列 (式 (6)) である。さらに、 $\mathbf{t}$  は 2 次元のベクトルである (式 (7))。そして、 $f(p_i, \{q_j\})$  は  $L_2$  ノルムの計算となる (式 (8))。

$$\mathbf{R} = \begin{pmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{pmatrix} \quad (6)$$

$$\mathbf{t}_i = \begin{pmatrix} x_i \\ y_i \end{pmatrix} \quad (7)$$

$$f(p_i, \{q_j\}) = \|p_i - q_j\|^2 \quad (8)$$

##### 4.3 $L_0$ ノルム最小マッチング

$L_2$  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、4.1 節で解説したように、リアルタイム性が求められるロボットで

は、車椅子ロボットの制御周期 (50 ms) 内に計算が完了する保証がないため適さない。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 (4) 同様、評価関数  $E(\mathbf{R}, t)$  は定義されるが、関数  $f$  は異なり、以下のようになる (式 (9))。

$$f(p_i, \{q_j\}) = \begin{cases} 0, & (\exists j, \|p_i - q_j\| \leq \epsilon) \\ 1, & (\text{otherwise}) \end{cases} \quad (9)$$

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させる。

$L_0$  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を行う必要が生じる。

#### 4.4 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。しかしながら、 $k-d$  木法では、多くの最近傍候補点が存在する場合には、演算量が不定となるため、リアルタイム性が必要となる人物搭載形電動車椅子では、安定した結果が得られない。

そこで本研究では、総当たり法を使用しつつ、最近傍候補点が限定できる多重解像度法を用いて処理の高速化を図る。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。下記に述べるように、今回の実装においては、多重解像度法を用いることにより最近傍候補点が100個という環境を維持することによりリアルタイム性を確保しつつ精度の維持を行っている。

しかし、地図の空間分解能が低い場合、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。図3に多重解像度法使用のイメージを示す。本研究では、環境地図の1グリッドを実環境の  $200\text{mm}^2$ ,  $100\text{mm}^2$ ,  $50\text{mm}^2$  に対応した3組の解像度が異なる環境地図をあらかじめ生成する (以下、それぞれ  $200\text{mm}^2$  地図,  $100\text{mm}^2$  地図,  $50\text{mm}^2$  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

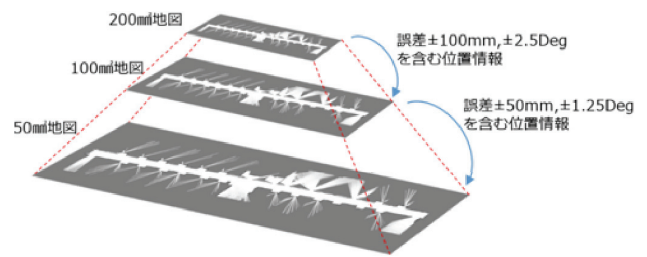


図3 自己位置推定のための多重解像度法

Fig. 3 Multi resolution method for self location estimation.

も  $-1\text{m} \sim 1\text{m}$  の範囲を  $200\text{mm}$  間隔で、姿勢 (方向) は  $-45\text{deg} \sim 45\text{deg}$  の範囲を  $5\text{deg}$  間隔で探索した。この際の誤差は  $x$  方向  $y$  方向に  $\pm 100\text{mm}$  であり、姿勢は  $\pm 2.5\text{deg}$  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  $\pm 25\text{mm}$ 、姿勢  $\pm 0.625\text{deg}$  の精度で探索することができることになる。

## 5. 環境地図

本論文では環境地図ベースの自律走行を行っていることは述べた。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。車椅子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。以降でこの定義内容について述べる。

### 5.1 局所環境地図の生成

図4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ことを示す。使用した LRF は視野角  $270\text{deg}$ 、方向分解能が  $0.25\text{deg}$  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。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フィルタ処理を行った。

本実験では以下の理由により、 $R$  を  $10\text{m}$  とした。

$L_0$  ノルム最小化法では、LRF センサから取得された点群を、上述したように  $50\text{mm}$ ,  $100\text{mm}$ ,  $200\text{mm}$  単位の2次元空間をグリッド状に区切り、重複した数値を削除し、同一グリッドの代表値をする。そこから、事前に作成されている大域環境地図と AND 演算を行ってマッチ数を求める。したがって、センサ側のデータ数が少なければ少ないほど、AND 演算の回数を減らし、計算を高速化することが可能となる。今回の走行環境 (2.3 節) において実験を実施し、センサから  $10\text{m}$  以内であれば、地図のマッチン

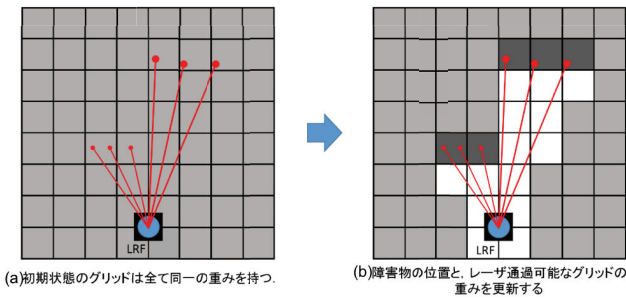


図 4 局所環境地図の生成  
 Fig. 4 Create location environment map.

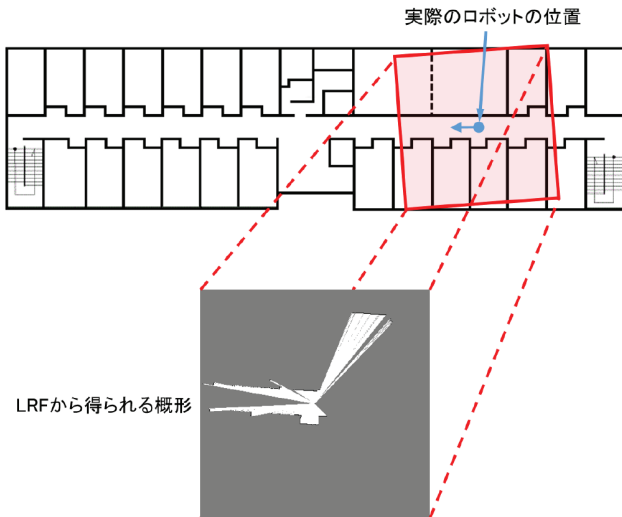


図 5 実環境と局所環境地図の対応  
 Fig. 5 Mapping between real environment and local environment map.

グに十分な特徴量を含めることができると考える。

図 5 に実際の環境で生成した局所環境地図の一例を示す。

### 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、車椅子ロボットを手動で操作し、走行環境の大域環境地図を生成する。図 6 に示すように時刻  $t-1$  に取得した局所環境地図を  $m_{t-1}$ 、時刻  $t$  に取得した局所環境地図を  $m_t$  とする。

この 2 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、 $m_t$  を重ね合わせることでの変換パラメータ  $(R, t)$  が求められる。求めたパラメータで  $m_t$  を変換すると、 $m_{t-1}$  にはない領域が  $m_t$  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、走行環境全体の大域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。

4.4 節で述べたように、多重解像度法で用いる低解像度

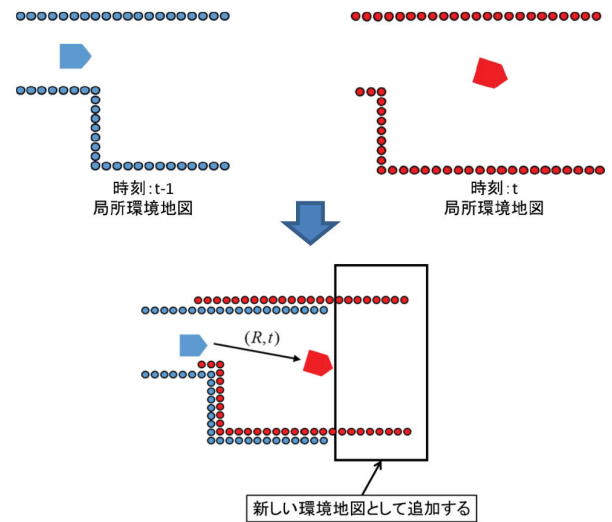


図 6 環境地図の更新  
 Fig. 6 Update environment map.

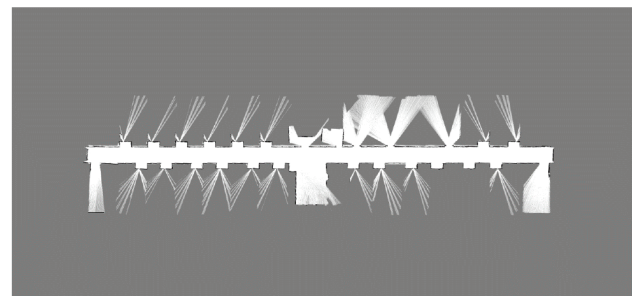
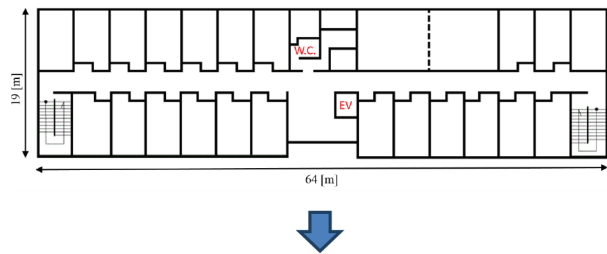


図 7 大域環境地図

Fig. 7 Global environment map.

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、車椅子ロボットの位置が  $\pm 200 \text{ mm}$  以上、または姿勢が  $5 \text{ deg}$  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 7 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のみでは、地図上における壁を含む障害物の幾何学的特徴が失われることがある。

そこで  $50 \text{ mm}^2$  地図における壁を含む障害物のグリッドの座標値のみを 0.25 倍にダウンサンプリングを行い、最後に再び環境地図画像に変換した。これにより縮小地図画像における壁を含む障害物の幾何学的特徴量の消失を最低限に防いでいる。



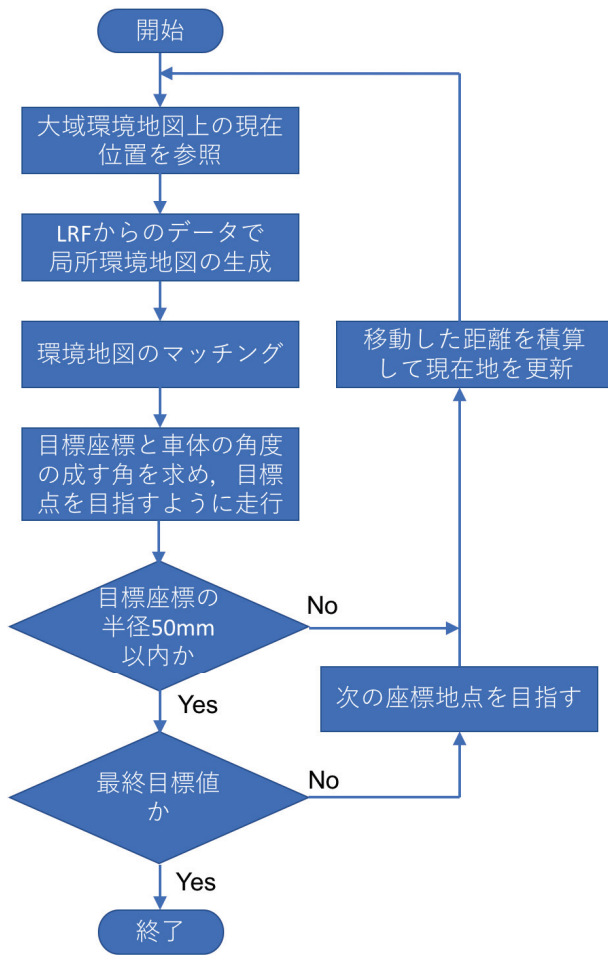


図 8 自律走行処理の流れ  
Fig. 8 Flow of autonomous navigation.

## 6. 自律走行

4章で述べた自己位置推定および5章で述べた2種類の環境地図に基づき、本研究における車椅子ロボットの自律走行処理の流れを図8に示す。なお、今回は、自律走行時には、大域環境地図の生成が完了しているものとする。

### 6.1 ロボットの誘導

車椅子ロボットの誘導は、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、環境地図上で初期位置、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必要となる。そこで、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、車椅子ロボットの姿勢と目的座標の方向との差をな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。

$$\lambda = \tan^{-1} \frac{y_i - y}{x_i - x} + \theta_{GL} \quad (10)$$

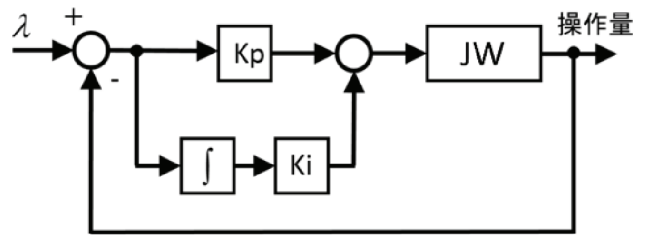


図 9 制御ブロック図  
Fig. 9 Control block.

$\lambda$  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）が起これ、つね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。図9に、制御ブロック図を示す。目標座標点への姿勢の偏差を  $\lambda$  とし、求めた操作量を左右の車輪の速度に加減算することで制御する。

## 7. 自律走行の評価実験

4章から6章までに述べた手法と、2章で述べた解決すべきタスクと環境下での評価を行う。本論文における主たる評価目的は以下のとおりである。

- 多重解像度法による性能速度と推定精度の検証
- $L_0$  ノルム最小化による頑健性の検証

上記の検証により、自律走行とマニュアル走行の併存が可能となるリアルタイム性と高精度な自己位置推定能力の結果が期待される。

### 7.1 自己位置推定法の性能速度評価

多重解像度法に関する評価を行うために、実環境下における自己位置推定の性能速度を評価する。

図10、図11、図12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、 $y$  方向ともに  $1\text{m}$ 、 $-45\text{deg}$  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。200  $\text{mm}^2$  地図で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。100  $\text{mm}^2$  地図でのマッチングではほぼマッチングされ、50  $\text{mm}^2$  地図では回転成分のみのマッチングとなった。

なお、200  $\text{mm}^2$  地図のマッチングに要した時間は約9ms、100  $\text{mm}^2$  地図のマッチングに要した時間は約5ms、50  $\text{mm}^2$  地図のマッチングに要した時間は約4msとなる。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。したがって、4.1節で示したJWアカデミックパックの制御周期（50ms）内で確実に計算可能となっている。

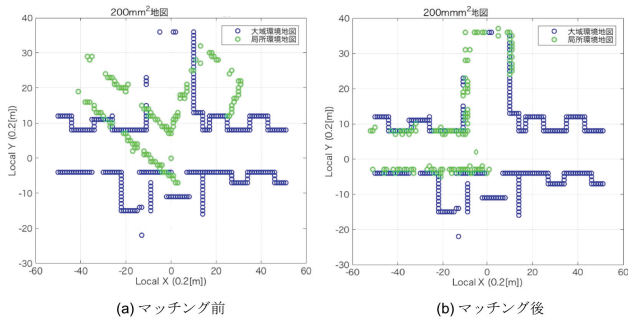


図 10 200 mm<sup>2</sup> 地図におけるマッチング効果  
 Fig. 10 Matching effective of 200 mm<sup>2</sup> map.

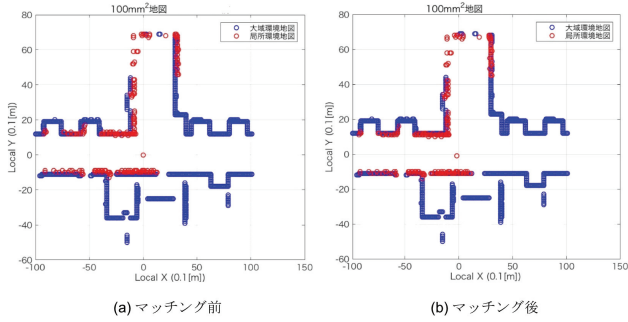


図 11 100 mm<sup>2</sup> 地図におけるマッチング効果  
 Fig. 11 Matching effective of 100 mm<sup>2</sup> map.

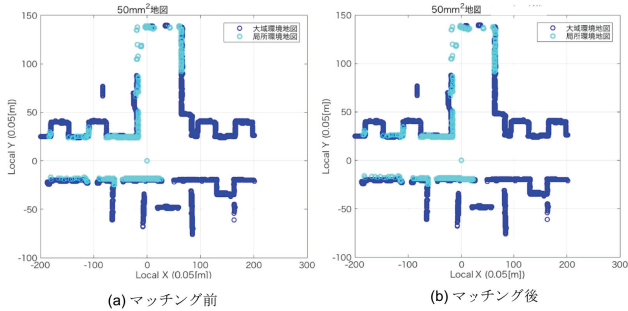


図 12 50 mm<sup>2</sup> 地図におけるマッチング効果  
 Fig. 12 Matching effective of 50 mm<sup>2</sup> map.

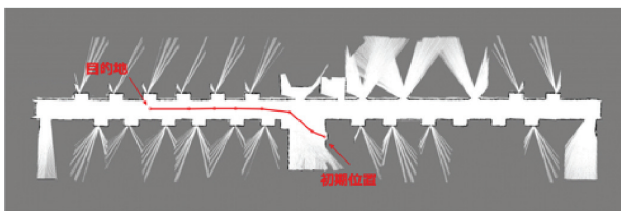


図 13 評価に用いた走行経路  
 Fig. 13 Driving route for evaluation.

## 7.2 自己位置推定精度の評価

5章で生成した環境地図をもとに、自律走行時の自己位置推定の精度を評価する。図13に文献[12]中に示した自動経路設計に基づき、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。評価実施内容は、エレベータホールから一室への経路とした。

続いて、図13の経路追従するよう、自己位置推定しな

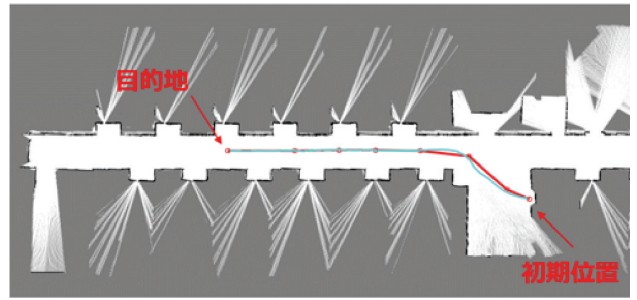


図 14 静的環境時の走行軌跡  
 Fig. 14 Driving route (static environment).

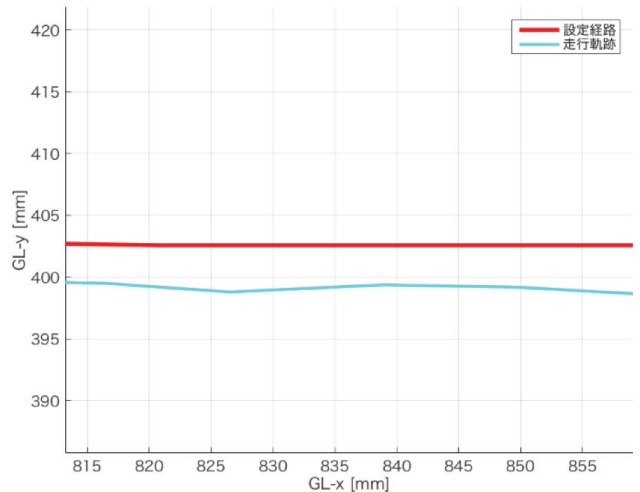


図 15 自己位置推定結果  
 Fig. 15 Result of self location estimation.

から走行した結果を図14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15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±25 mm以内と、4.4節に示したとおりの結果となった。

## 7.3 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で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名ほどに、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(文献[13]の動画参照)。経路設定などは、静的環境で行った方式と同一とした。

図16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



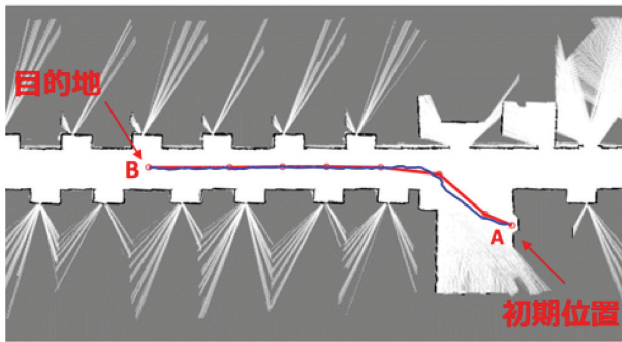


図 16 動的環境時の走行軌跡

Fig. 16 Driving route (dynamic environment).

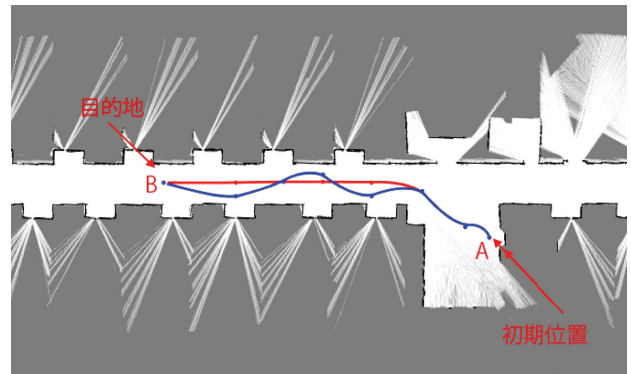


図 18 ユーザ操作による経路変更

Fig. 18 Change way-point by user operations.

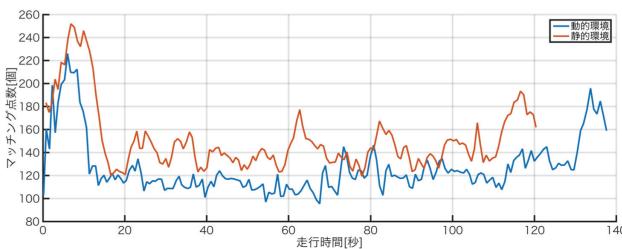


図 17 マッチング点数の比較

Fig. 17 Comparison number of matching.

置に乱れが生じていることが分かる。しかしおおむね自己位置を推定できており、自律走行には問題がないことが分かる。

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いえる。図 17 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分かる。

さらに、車椅子に装備されているジョイスティックを用いて、車椅子の走行速度を変更させたり、走行途中において、経路の変更を試みた。方式としては、自動経路設定時に作成される経路候補点 (way-point) の座標をジョイスティック操作により変更させることで、最終経路を守りつつ、途中経路の変更を行う評価を実施した。図 18 に初期経路設定時の軌跡 (赤) と、ユーザによる経路変更 (左右) を行ったときに経路軌跡 (青) を示す。ユーザは、本実験では、あえて、直線経路において、ジグザクに経路をとったところ、経路候補点を障害物に対して安全な場所まで配置させることによって、車椅子の経路を変更できていることを示している。ただし、安全面を考慮し、短時間に何度

もジョイスティック操作を入れられないように 200 msec に一度の受付制御としており、大幅な経路変更は実施できないようにしている。

## 8. まとめ

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椅子ロボットの開発を行った。開発した車椅子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椅子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。今後の課題として、現状では自律走行する環境ごとに地図の作成を行う必要がある点があげられる。また、目的地候補となる部屋や施設への経路に対して、経路自動作成とジョイスティックによる手動制御に対する感性評価とその改良を行う必要がある。さらに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、最適経路設計を行うシステムなどを開発することが考えられる。同時に、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) を取り入れることも考えられる。

## 参考文献

- [1] 警視庁：最近の交通事故の実態，警視庁（オンライン），入手先 ([https://www.npa.go.jp/koutsuu/kikaku12/shi\\_04jikojittai.pdf](https://www.npa.go.jp/koutsuu/kikaku12/shi_04jikojittai.pdf)) (参照 2017-01-06)。
- [2] 株式会社シコウ：AGV 自律走行型無人搬送機，シコウ

- (オンライン), 入手先 (<http://www.shiko.biz/agv/index.html>) (参照 2017-01-06).
- [3] ヤマハ発動機株式会社: JW アクティブ, ヤマハ (オンライン), 入手先 (<https://www.yamaha-motor.co.jp/wheelchair/lineup/active-plus/>) (参照 2017-01-06).
- [4] 中沢 実, 鷹箸孝典, 阿部拓真: 脳波からの指示による電動車椅子制御システムの実装と評価, 情報処理学会研究報告マルチメディア通信と分散処理 (DPS), Vol.2015-DPS-163, No.14, pp.1–8 (2015).
- [5] OpenSLAM.org: OpenSLAM, available from (<https://openslam.org/>) (accessed 2017-01-06).
- [6] Rusinkiewicz, S. and Levoy, M.: Efficient variants of the ICP algorithm, *Proc. 3rd International Conference on 3-D Digital Imaging and Modeling*, pp.145–152 (2001).
- [7] 山下 遼他: パーティクルフィルタを用いた全方向移動ロボットの自己位置同定における効率的な探索法, 日本機械学会ロボティクス・メカトロニクス講演会講演論文集, 1A2-E27 (2010).
- [8] 北陽電機株式会社: UTM-30LX, 北陽電機株式会社 (オンライン), 入手先 (<http://www.hokuyo-aut.co.jp/search/single.php?serial=146>) (参照 2017-01-06).
- [9] 増田 健: ICP アロゴリズム, コンピュータビジョン最先端ガイド 3, pp.33–62, アドコムメディア (2010).
- [10] 永田祐介, 永末 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), pp.1256–1263 (2011).
- [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, 電子情報通信学会 CNR2011-26, pp.15–19 (2011).
- [12] 阿部拓真, 中沢 実: 自律搬送車における効率的な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 (2014).
- [13] 実験動画, 入手先 (<https://youtu.be/4GkuU-JMezw>) (参照 2017-01-06).

## 推薦文

本研究は車いす利用者の死傷事故への対処という必要性の高い課題に対して, 高齢者の自律的移動を補助するシステムを開発しています。十分な評価を行っており, 考察も深く行っています。今後, システムのさらなる自動化が期待されます。以上の理由により, 本論文を推薦いたします。

(マルチメディア通信と分散処理研究会主査 重野 寛)



中沢 実 (正会員)

1993 年金沢工業大学大学院工学研究会情報工学専攻博士前期課程修了。同年富士通研究所入社。1996 年金沢工業大学助手。1999 年同大学講師, 2002 年同大学助教授, 2011 年同大学教授となり, 現在に至る。博士 (工学)。自律分散システム, ロボティクス, 信号処理等に関する研究に従事。2016 年情報処理学会山下記念研究賞受賞。電子情報通信学会, 日本ロボット学会, 人工知能学会, IEEE 各会員。本会シニア会員。



阿部 拓真

2013 年金沢工業大学工学部情報工学科卒業。2015 年同大学大学院工学研究科情報工学専攻博士前期課程修了。在学中, ロボット制御に関する研究に従事。現在, 株式会社 NS・コンピュータサービスに勤務。



杉下 大河

2016 年金沢工業大学工学部情報工学科卒業。2017 年同大学大学院工学研究科情報工学専攻博士課程在学。ロボット制御に関する研究に従事。



佐野 渉二 (正会員)

2006 年神戸大学大学院自然科学研究科博士前期課程修了。2012 年同大学大学院自然科学研究科博士後期課程修了。神戸大学工学研究科学術推進研究員, 公立ほこだて未来大学特任研究員, 神戸大学工学研究科研究員を経て, 2016 年金沢工業大学講師となり, 現在に至る。博士 (工学)。実世界コンピューティング, ヒューマンコンピュータインタラクション, 公共交通システムの研究に従事。人工知能学会, 日本工学教育協会, サービス学会, ACM, IEEE の各正会員。