

明治大学大学院理工学研究科

2014年度

博士学位請求論文

**移動ロボットを用いた分散センサ群の位置推定に基づく**

**知能化空間の構成支援システムに関する研究**

Study on a support system for building intelligent spaces based on  
position estimation of distributed sensors using a mobile robot

学位請求者 電気工学専攻

橋川 史崇

## 目次

1	緒言	1
1.1	研究背景	1
1.2	本論文の構成	1
2	知能化空間	3
2.1	知能化空間のコンセプト	3
2.2	知能化空間の例	4
2.3	知能化空間の構築における課題	5
3	分散センサの位置推定支援システム	9
3.1	分散センサと移動ロボットの地図情報の比較	9
3.2	分散センサの地図生成	11
3.2.1	分散センサの動作	11
3.2.2	レーザ式測域センサ	12
3.2.3	分散センサによる地図生成手法	13
3.3	移動ロボットの自己位置推定と地図生成	17
3.3.1	移動ロボットの動作	17
3.3.2	ホイールオドメトリによる動作モデル	18
3.3.3	移動ロボットの地図生成	21
3.3.4	移動ロボットの SLAM	21
3.4	地図比較時の動作	23
4	分散センサと移動ロボットの地図情報の比較方法	25
4.1	座標変換による地図情報の比較	25
4.2	座標変換による手法の検証	26
4.3	座標系間の角度差による地図情報の比較	29
4.4	ICP マッチングによる地図比較	32
4.4.1	分散センサの位置指定	32
4.4.2	移動ロボットの SLAM 修正	36
5	地図比較手法の検証実験	38
5.1	使用機器と実験環境および実験方法	38
5.1.1	分散センサ	38
5.1.2	移動ロボット	40
5.1.3	実験環境	43
5.2	分散センサの地図生成における実験結果	45
5.3	移動ロボットの FastSLAM の実験結果	47
5.4	ICP マッチングを用いた位置推定と地図生成の結果	48

5.5 座標系間の角度差による手法.....	58
6 提案手法についての考察.....	62
6.1 分散センサの地図生成と移動ロボットの FastSLAM の考察.....	62
6.2 提案手法による実験結果の考察.....	62
7 結言.....	66
7.1 本論文のまとめ.....	66
7.2 今後の課題と展望.....	68
付録.....	70
A 移動ロボットによる空間認識手法と課題.....	70
B 車輪型移動ロボットの制御手法.....	73
B.1 移動機構.....	73
B.2 移動ロボットの軌道追従制御.....	76
C ワンボードマイコン制御によるセンサの小型化の例.....	81
参考文献.....	84
業績一覧.....	90
謝辞.....	92

## 図目次

図 2.1	知能化空間の概念図.....	3
図 3.1	提案手法の概要図.....	10
図 3.2	分散センサのシステム構成図.....	11
図 3.3	分散センサの座標系.....	12
図 3.4	レーザ式測域センサの使用例.....	13
図 3.5	格子地図の生成例.....	14
図 3.6	格子地図生成手法.....	15
図 3.7	対数オッズ算出手法.....	15
図 3.8	移動物体検知手法.....	16
図 3.9	移動ロボットのシステム構成図.....	17
図 3.10	移動ロボットの座標系.....	19
図 3.11	オドメトリ動作モデル.....	19
図 3.12	移動ロボットに搭載したレーザ式測域センサの仕様図.....	21
図 3.13	パーティクルフィルタによる移動ロボットの位置推定例.....	22
図 3.14	提案手法のシステム構成図.....	23
図 4.1	パーティクルフィルタによる分散センサの位置推定手法.....	25
図 4.2	システム概要図.....	26
図 4.3	座標重ね合わせ手法による移動ロボットの地図.....	27
図 4.4	分散センサ No.0 における対応の様子.....	28
図 4.5	分散センサ No.1 における対応の様子.....	28
図 4.6	分散センサ No.2 における対応の様子.....	28
図 4.7	分散センサ No.3 における対応の様子.....	29
図 4.8	傾き比較の手法.....	31
図 4.9	知能化空間の動作手法.....	32
図 4.10	移動ロボットの動作手法.....	33
図 4.11	ICP マッチングの概要図.....	35
図 4.12	SLAM における分散センサの局所地図利用方法.....	36
図 5.1	知能化空間を構成する分散センサ.....	38
図 5.2	Pioneer 3-DX.....	40
図 5.3	UTM-30LX.....	41
図 5.4	実験環境図.....	43

図 5.5	分散センサ No.0 による地図生成結果.....	45
図 5.6	分散センサ No.1 による地図生成結果.....	46
図 5.7	分散センサ No.2 による地図生成結果.....	46
図 5.8	分散センサ No.3 による地図生成結果.....	47
図 5.9	大域地図.....	47
図 5.10	FastSLAM による大域地図生成の推移.....	48
図 5.11	FastSLAM による局所地図.....	48
図 5.12	FastSLAM による距離データ.....	48
図 5.13	分散センサの位置推定結果.....	50
図 5.14	分散センサ No.0 における ICP 実装結果.....	50
図 5.15	分散センサ No.1 における ICP 実装結果.....	51
図 5.16	分散センサ No.2 における ICP 実装結果.....	51
図 5.17	分散センサ No.3 における ICP 実装結果.....	52
図 5.18	分散センサ No.0 における ICP 実装推移.....	52
図 5.19	分散センサ No.1 における ICP 実装推移.....	53
図 5.20	分散センサ No.2 における ICP 実装推移.....	53
図 5.21	分散センサ No.3 における ICP 実装推移.....	53
図 5.22	ICP を用いた SLAM の大域地図.....	54
図 5.23	ICP を用いた SLAM の大域地図生成の推移.....	54
図 5.24	ICP を用いた SLAM の局所地図.....	54
図 5.25	ICP を用いた SLAM の距離データ.....	55
図 5.26	SLAM における距離データと分散センサ No.0 との対応推移.....	55
図 5.27	SLAM における距離データと分散センサ No.1 との対応推移.....	55
図 5.28	SLAM における距離データと分散センサ No.2 との対応推移.....	56
図 5.29	SLAM における距離データと分散センサ No.3 との対応推移.....	56
図 5.30	SLAM における距離データと分散センサ No.0 との誤対応推移.....	57
図 5.31	SLAM における距離データと分散センサ No.1 との誤対応推移.....	57
図 5.32	SLAM における距離データと分散センサ No.2 との誤対応推移.....	57
図 5.33	SLAM における距離データと分散センサ No.3 との誤対応推移.....	58
図 5.34	ICP を用いた SLAM の失敗例.....	58
図 5.35	分散センサの位置推定結果.....	60
図 5.36	傾き比較による移動ロボットの地図.....	60
図 5.37	分散センサ No.0 における対応の様子.....	61
図 5.38	分散センサ No.1 における対応の様子.....	61
図 5.39	分散センサ No.2 における対応の様子.....	61
図 5.40	分散センサ No.3 における対応の様子.....	61

図 A. 1	移動ロボットの SLAM の概要図 .....	70
図 B. 1	車輪型動ロボットのステアリング例 .....	73
図 B. 2	ステアリング方式の分類.....	73
図 B. 3	ステアリング方式の例 .....	74
図 B. 4	移動ロボットの動作タイプ .....	74
図 B. 5	移動ロボットの旋回タイプ .....	75
図 B. 6	本研究の移動ロボットのステアリング .....	75
図 B. 7	移動ロボットの軌道追従制御の様子 .....	76
図 B. 8	座標系の変換.....	77
図 B. 9	リファレンス点の初期位置の決定法 .....	79
図 B. 10	走行時の車輪の回転による軌跡の様子.....	80
図 C. 1	ワンボードマイコン”Raspberry Pi” .....	81
図 C. 2	Raspberry Pi による移動ロボット制御ユニットの小型化 .....	81
図 C. 3	Raspberry Pi による制御ユニットの小型化例.....	82
図 C. 4	移動ロボット制御ユニット比較例 .....	83

## 表目次

表 5. 1	URG04-LX-UG01 の仕様表 .....	39
表 5. 2	RaspberryPi の仕様表 .....	39
表 5. 3	Pioneer 3-DX の仕様表 .....	41
表 5. 4	UTM-30LX の仕様表 .....	42
表 5. 5	ICP による分散センサ No.0 の位置推定結果 .....	49
表 5. 6	ICP による分散センサ No.1 の位置推定結果 .....	49
表 5. 7	ICP による分散センサ No.2 の位置推定結果 .....	49
表 5. 8	ICP による分散センサ No.3 の位置推定結果 .....	49
表 5. 9	角度差による分散センサ No.0 の位置推定結果 .....	59
表 5. 10	角度差による分散センサ No.1 の位置推定結果 .....	59
表 5. 11	角度差による分散センサ No.2 の位置推定結果 .....	59
表 5. 12	角度差による分散センサ No.3 の位置推定結果 .....	59
表 B. 1	ステアリング方式の名称と特性 .....	74
表 B. 2	リファレンス点の設置座標 .....	80

# 1 緒言

## 1.1 研究背景

近年の IT 産業やロボット工学の発展に伴い、我々の生活の中でも IT やロボット技術が数多く利用されてきている。例えば、建造物内に設置されている警備用の監視カメラや、生産工場などで製品の運搬を無人でおこなう産業用ロボットがあげられる。監視カメラは、建造物内の各階の天井隅などに設置され、映像から人の存在の有無を自動識別している。夜間などの人の出入りを禁止している時間帯に人の存在を検知した際には、自動で警報を鳴らすなどの無人での監視をおこなうことができる。産業用ロボットなどにも自動識別のためにセンサが搭載され、ロボットと周囲の人や物との位置関係を認識することで、自律行動を可能にしている。工場内を行き交う様々な運搬物や作業員が存在する空間内でも人が側を通過する際には、安全のために停止することができる。また、通路に運搬物が置かれている場合には、置かれた障害物を避けて通過することや、移動させることができる。いずれの例も我々人間がおこなうには労力や時間を費やす作業を、IT やロボット技術によって自動化することで効率よく正確におこなうことができる。上記の例の他にも IT やロボット技術は、我々の生活の様々な場面で利用されている。

今日でも様々な場面で利用するための IT やロボット技術の研究が多くおこなわれている。監視カメラに代表するような、ある空間の状態を識別する IT 技術の一例としては、知能化空間に関する研究がある[1]。知能化空間は、空間内に存在する人や物の位置を取得して、その空間内の状態を認識するシステムである。代表例として、空間内にロボットアームやセンサを設置して空間そのものを知能ロボットとして人を支援するシステムや[2]、センサから人の動作を読み取り空間内の状況を認識するシステム[3]がある。人や物の位置を取得するために、知能化空間として定めた空間内にセンサを天井などに分散的に配置している[4]。配置されたセンサは、互いに無線ネットワークなどにより接続してコンピュータ制御することで、各センサが取得した空間内の情報を統合して空間内の状態を認識する[5]。認識した空間情報を利用して、特定人物の動向追跡[6]や、ロボットへのナビゲーション[7]、ロボットによる人物追従 [8][9][10]などの支援をおこなうことができる。

## 1.2 本論文の構成

本論文の章構成を以下に示す。本章では、本研究における研究背景を記した。2 章では、まず、知能化空間の説明と実用例を示して、知能化空間構築における関連研究とその課題を紹介する。3 章では、本研究における分散センサの位置推定手法および知能化空間を構成

する分散センサと移動ロボットの動作手法をそれぞれ記す。分散センサと移動ロボットが独立しておこなう地図生成と位置推定手法を紹介する。また、提案手法における分散センサと移動ロボットの地図情報の比較方法の説明をおこなう。4章では、本研究の提案手法における分散センサと移動ロボットの全体の動作手法を記す。また、提案手法に至る前の事前実験として、分散センサと移動ロボットの地図の座標系を単純に座標変換する手法や ICP マッチング、座標系間の角度差に注目して地図同士を重ね合わせて位置の対応を図った原理や検証実験についても報告する。5章では、本研究の提案手法が有効であるかを確認するための評価実験の環境とその評価項目を示す。また、評価実験の実験結果も記し、4台の分散センサと1台の移動ロボットが独立して生成した地図と提案手法によって生成した地図の比較の様子や、推定された位置座標と実際の座標として計測した位置座標をまとめる。6章では、評価実験から得られた結果を基に、提案手法が分散センサの位置推定と移動ロボットの SLAM における推定精度の向上について適切であるかを考察する。7章では、本論文全体のまとめと、本研究の提案手法におけるまとめを示す。付録 A には、移動ロボットの SLAM の動作手法を紹介する。付録 B には、本論文で取り扱う車輪型移動ロボットの移動機構と移動ロボットが自律して設計した経路を追従することを目的とした軌道追従制御について記載した。ここでは、移動ロボットと目標経路上にリファレンス点を設けそのリファレンス点を結ぶ二次曲線を利用した方法により軌道追従制御を行う方法や障害物回避について述べる。付録 C には、ワンボードマイコンによる各センサの小型制御ユニットシステムを作成した例を紹介する。

## 2 知能化空間

### 2.1 知能化空間のコンセプト

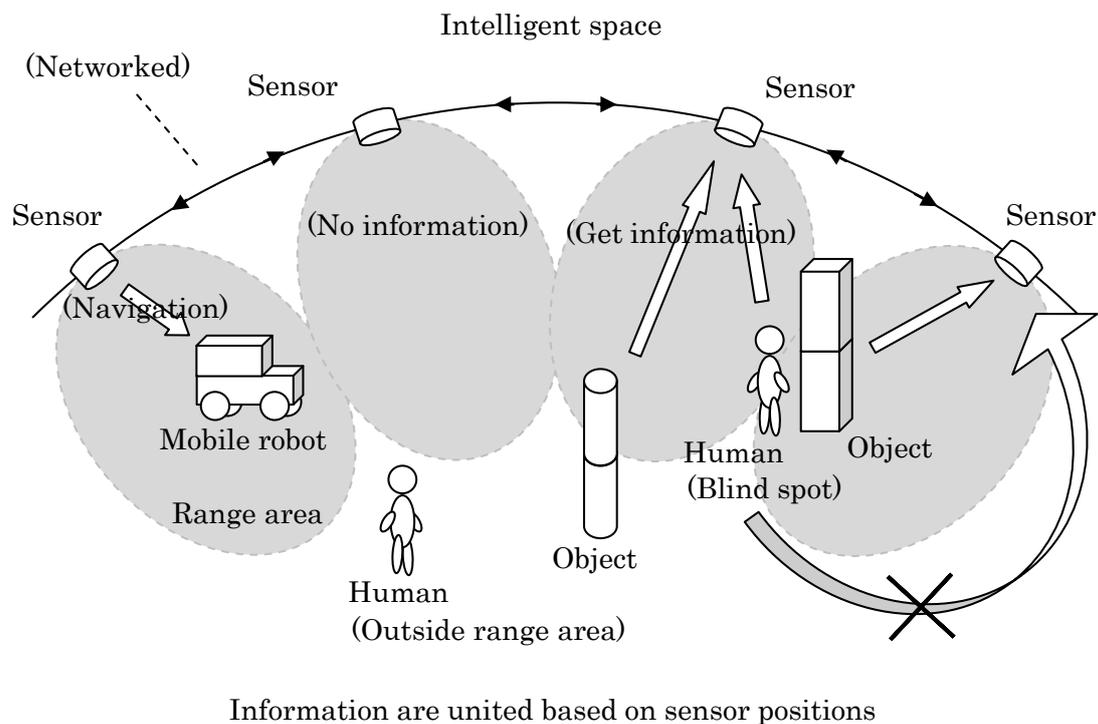


図 2.1 知能化空間の概念図

図 2.1 に知能化空間の概念図を示す。知能化空間は、空間に設置された様々なセンサから空間内の人や物の位置情報などを取得して、その空間内の状態を認識するシステムを示す。認識した空間情報は、人への道案内や空間内の防災システムへ利用するなど様々な利用方法が考えられる。図 2.1 では人や物の位置を取得するために、ある空間の周りにセンサを分散的に配置して、その空間を知能化空間として定めている。配置されたセンサは、それぞれから観測できる範囲の空間情報を取得している。知能化空間は一つのセンサの観測範囲ではまかなえない広範囲に及ぶ場合や、曲がり角や壁、障害物などにより死角が多く存在する場合も多いため、複数台のセンサが用いられる。各センサの空間情報を統合することで、知能化空間全体の空間認識をおこなうことができる。各センサは空間情報統合のために、ネットワーク接続されコンピュータでその空間情報を統合する。認識した空間情報をロボットや人へ伝えることで、ロボットへのナビゲーションや人への支援、センサやロボットによる特定人物の追従などをおこなうことができる。

## 2.2 知能化空間の例

ここでは、これまでに実現されてきた知能化空間の一例を紹介する。配置されるセンサには、特定の物体の識別のために色彩を認識情報として取得するネットワークカメラ[11][12]や人物追従のために人の足首などの特徴的な形状取得するために空間の形状を取得するレーザセンサ[13][14]を用いる例や、特定の対象物に搭載することで、その動作情報を取得することができる加速度センサ[15][16]など様々なセンサがそれぞれの目的毎に用いられている。様々なセンサにより空間内のロボットやコンピュータも知能化空間と統合することで、ロボットやコンピュータから得られる空間情報も統合して、知能化空間の機能を拡張することが可能になる[17][18][19][20]。

### ネットワークカメラを用いた例

#### 文献[11]

室内空間の天井にネットワークカメラを複数台設置して、各カメラが観測した領域内を知能化空間として定めている。各カメラの画像からターゲットして定めた移動物体の動きを認識するシステムが紹介されている。

#### 文献[12]

文献[11]に関連して、各カメラから得られた空間内の画像情報から人物追従をおこない、その空間内に存在する移動ロボットに追従対象となる人物の位置を伝えることで、移動ロボットによる人物追従システムをサポートしている。

### レーザセンサを用いた例

#### 文献[13]

複数台のレーザセンサを室内空間の床下に設置して、その空間を知能化空間として定めている。それぞれのセンサから人の足首を観測しており、観測された足首から得られた位置データをネットワーク接続されたセンサ間で情報共有することで、人物追従をおこなう手法を紹介している。

#### 文献[14]

人通りの多い通路などの天井に複数の3Dレーザセンサを設置して、その空間を知能化空間として定めている。3Dセンサによって観測された人の頭から肩にかけてのデータをセンサ間で共有することで、特定人物の進行方向などを認識して、追従をおこなっている。

### 加速度センサを用いた例

#### 文献[15]

特定人物の腰に加速度センサを搭載して、その人物の進行方向や位置情報が取得できる状態になっている。また、レーザセンサを室内空間の床下に設置して、その空間を知能化

空間として定める。センサから人の足首を観測しており、観測された足首から得られた位置データをネットワーク接続された加速度センサとの間で情報共有することで、複数の人物が存在する空間における特定の人物追従をおこなう手法を紹介している。

#### 文献[16]

複数の人物に加速度センサを搭載して、人と壁などの静止物体を見分ける。各加速度センサはネットワーク接続されており、複数の人物の進行方向や位置情報を取得して統合することで知能か空間内に存在する複数の人の認識や追従が可能になることを示している。

### その他ロボットやコンピュータを用いた例

#### 文献[17]

室内空間に人感センサを設置して移動物体の存在を検知した際に照明を自動点灯させるシステムや、音声認識によって窓などを自動開閉させるシステムを室内に構築している。家という空間全体をロボットと見なして、知能化空間を構築している。

#### 文献[18]

人が出入りするドアの手前に、2台のレーザセンサを向かい合わせて設置している。それぞれのレーザセンサは、人の腰の高さ程に設置されており、各センサから人の位置と進行方向を推定している。その推定された位置情報に基づき、ドアを自動開閉するシステムを紹介している。

#### 文献[19]

知能化空間を構成する各レーザセンサや CCD カメラ、超音波センサから得られた空間情報を統合することで、知能化空間内に存在する移動ロボットをその空間情報に基づき制御する方法を紹介している。各センサ情報から移動ロボットの位置推定をおこない、移動ロボットをナビゲーションすることで、移動ロボットを介して空間内の人へサポートを提案している。

#### 文献[20]

室内空間に設置したネットワークカメラから人物の位置を取得して、移動ロボットにその位置情報を伝えている。人物の位置情報を取得した移動ロボットに、その移動する人の座標を追跡させることで、知能か空間から移動ロボットが人物検知をおこなうサポートをしている。

## 2.3 知能化空間の構築における課題

知能化空間によって空間内の状態認識をおこなうには、知能化空間を構成する分散センサが取得した人や物の位置を統合する必要がある。各分散センサの空間情報を統合して知能化空間内全域の空間情報とするためには、知能化空間を構成する各分散センサの設置位

置や設置方向を知る必要がある。各分散センサの設置位置や方向を知ることができれば、各分散センサが取得した空間内の物の位置関係を統合して空間全域の物の位置関係を知ることができる。また、センサの観測領域を超えるような広い空間や曲がり角などの死角が多い空間の知能化空間の構築には、センサ 1 台の観測領域だけでは効果的な空間情報を取得することができない場合が多い。そのため、複数のセンサを設置する必要がある。複数のセンサが空間内に設置された場合、知能化空間構築のためには、全ての分散センサの設置位置と方向を知る必要がある。しかし、広域に多数のセンサが設置された場合、全ての分散センサに対して設置位置と方向を知るとは煩雑で困難な作業になる。センサが 1、2 台の場合は、設置位置を計測するためにコンベックス等の計測機器を用いてセンサ 1 台 1 台の設置位置を手動で測定しても、多くの時間がかかる作業ではない。しかし、センサが複数台設置される場合は、手動での測定は時間と労力のかかる作業になる。また、設置されたセンサの位置が計測基準点となる座標から遠距離にあるほど、計測器の微小な設置方向の誤差などが、計測した設置位置に大きな誤差を与えてしまうことが考えられる。以上のことから、知能化空間の構築を容易にするためには、設置した分散センサの位置を自動的に安定して推定するシステムが求められる[21][22]。本節では、知能化空間を構成する分散センサの位置推定のために提案された代表的な手法を紹介する。

#### 文献[23],文献[24]

分散レーザ測域センサの位置推定システムを提案している。このシステムでは、複数台の分散レーザ測域センサを各々の観測領域が重なるように空間に配置している。その重複領域内において、各センサが同一の移動物体の位置をそれぞれ観測して、その位置の対応関係からセンサ相互の設置位置を推定している。

#### 文献[25],文献[26]

文献[23],文献[24]に関連して、移動ロボットを各分散センサにより観測して、その位置の対応関係から各センサの設置位置を推定している。移動ロボットから分散センサを検知する手法として、円柱形の柱を分散センサに設置して移動ロボットのレーザセンサから観測した形状が常に一定の形状であれば、それを分散センサと判断することとしている。

#### 文献[27]

あらかじめ位置のわかっている固定ランドマークを各センサから検出することで、その位置を基準とした分散センサの位置推定をおこなっている。位置推定された分散センサを新たな固定ランドマークとすることで、各々の観測領域が重なっている分散センサの位置推定を可能にしている。

#### 文献[28]

複数台のネットワークカメラにより構成した知能化空間において、Bayesian filter を用いて特定の移動物体を観測することにより、各ネットワークカメラの位置座標を推定する手法を提案している。各ネットワークカメラで観測した同一時刻の同一移動物体から得られ

る視覚特徴を特徴量として、その位置の対応点に基づいてカメラ間の位置関係を推定している。

#### **文献[29]**

室内空間に超音波センサを複数台設置して、センサ間の重複観測領域内に存在する特定の移動物体までの距離を各センサから計測することにより、センサ同士の位置を推定する手法を提案している。

#### **文献[30]**

超音波センサの受講強度を利用した、センサの位置推定方法を紹介している。センサ間の重複観測領域内に存在する特定の移動物体までの距離計測に超音波センサの受講強度を用いて、センサ同士の位置を推定する手法を提案している。

#### **文献[31]**

屋外環境において超音波センサを複数台設置して、それぞれの観測領域が重なるように設置する。その重複観測領域内に侵入した移動物体までの距離を取得して、センサ間の位置関係から各センサの位置推定をおこなう。

#### **文献[32]**

分散センサとして、超音波センサを用いている。各超音波センサの受講強度を利用して移動物体の追跡をおこなうことで、センサ間の位置関係を計算する。各センサの位置関係から全センサの位置を求める。

#### **文献[33]**

固有の無線周波数を発するセンサをそれぞれの観測領域が重なるように設置して、その空間内に存在する特定の移動物体までの距離を取得する。その取得した位置関係から各センサの位置を推定する手法を提案している。

#### **文献[34]**

2台のカメラで観測した共通の物体を基にカメラ間の位置関係を求めることで、それらのカメラと重複した観測領域を持つカメラにおいて、共通の特徴点を観測している他のカメラの位置推定をおこなう手法を提案している。

#### **文献[35]**

ガウス分布に基づく2台のカメラの位置に基づく地図の対応関係から、位置推定する手法を提案している。同様に、位置推定されたカメラとその他のカメラの対応も求めることで、各カメラの位置推定をおこなっている。

#### **文献[36]**

特定の移動物体の追従により、観測領域を持たないカメラの位置を推定する手法を提案している。移動物体を観測したカメラは、直前に移動物体を観測していたカメラ付近に配置されていると判断することで、移動物体が画像上に進入してきた方向などからカメラの大まかな位置と角度の推定をおこなっている。

これまでに提案されてきた分散センサの位置推定の手法は、特定の移動物体を特徴量としており、実環境のような移動物体が複数存在するような場合には、対象とする移動物体を誤認識する可能性がある。また、分散センサの設置条件に関しては、同一の移動物体を同時に測定するため、それぞれの観測領域が重なるように配置する必要がある手法が主である。センサ位置推定が実現できる程度に観測領域を重複させることで、センサ配置の自由度が低下して密集したセンサ配置になるため、特に広範囲に及ぶ空間にセンサが密集して分散配置する場合は、知能化空間の構築が煩雑になる可能性がある。

## 3 分散センサの位置推定支援システム

### 3.1 分散センサと移動ロボットの地図情報の比較

2.3 節に記載した関連研究の多くには、移動ロボットを分散センサから追従する対象として定め、その移動ロボットの位置情報を利用した分散センサの位置推定をおこなう手法があった。しかし移動ロボットにも空間認識をさせることで、自己位置の情報だけではなく知能化空間内の空間情報を取得させることができると考えられる。このことから、移動ロボットが取得した知能化空間の空間情報を分散センサの位置推定に利用できる可能性がある。移動ロボットは知能化空間内を走行しながら知能化空間全体の広域な空間情報を取得でき、それぞれの分散センサは知能化空間内の任意の場所に設置されているため、位置を変えずにセンサ周辺の空間情報を取得することができると考えられる。そのため、移動ロボットが知能化空間内を走行しながら取得した広域な空間情報と分散センサの局所的な空間情報を比較することで分散センサの位置推定が可能になると考えられる。分散センサの局所的な空間情報が、移動ロボットの大域的な空間情報のどの位置に対応するのかを求めることで、分散センサの位置推定が可能になると考えられる。そこで本研究では、複数の分散センサの設置位置を移動ロボットの空間情報を利用して求めるシステムの提案と実装をおこなう。移動ロボットによる空間認識に関する研究では既に **SLAM (Simultaneous Localization and Mapping)** [37][38] と呼ばれる地図生成と自己位置推定を同時におこなう手法がある。本研究でも分散センサの位置推定に有効な手法として用いるが、**SLAM** による移動ロボットの位置推定は、推定誤差を免れることが難しい。**SLAM** の詳細は付録 A で述べる。

そこで、本研究は知能化空間を構成する分散センサと移動ロボットの位置推定を同時におこなう手法を提案する。位置推定には、分散センサと移動ロボットが生成した地図を用いる。分散センサと移動ロボットは、初めは自己位置や地図情報が無い状態を想定しており、それぞれ独立して地図生成をおこなう。移動ロボットには地図生成のために、分散センサと同様のレーザ式測域センサを 1 台搭載している。本研究の提案手法は、知能化空間を構成する複数台の分散センサと 1 台の移動ロボットにより構成される。

図 3.1 には、本研究の提案手法のシステム構成の概要図を示す。分散センサは自己位置座標を基準にした地図生成を独立しておこなっている。生成される地図は、知能化空間内に設置された各分散センサの設置位置周辺の局所的な地図である。本論文ではこの分散センサの地図を局所地図と呼ぶ。局所地図の座標系は、分散センサを基準とした座標系で表わされる。移動ロボットは、知能化空間内全域を走行しながら **SLAM** を基に地図生成と自己位置推定を独立しておこなっている。生成される地図は、知能化空間内全域の大域的な地図である。本論文ではこの移動ロボットの地図を大域地図と呼ぶ。また、大域地図は移動

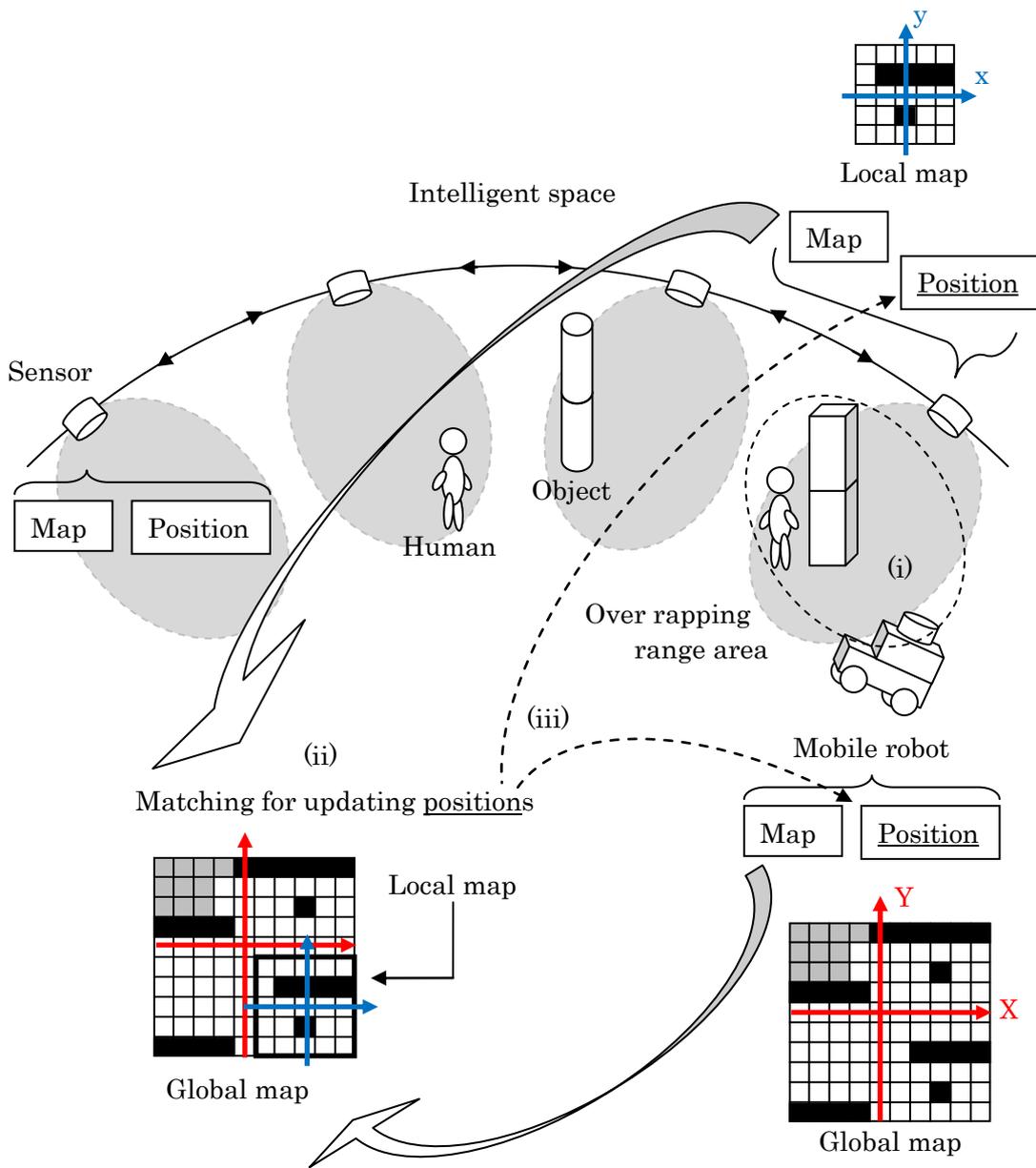


図 3.1 提案手法の概要図

ロボットの初期位置を原点とした座標系で表わされ、これを世界座標系と呼ぶ。分散センサおよび移動ロボットが生成する地図は、全て二次元の平面地図である。本研究の提案手法では、位置推定をおこなうために各分散センサと移動ロボットの地図を比較する。各分散センサの観測領域は、互いに重複しておらず独立している。地図の比較は、分散センサと移動ロボットが類似した形状の地図を生成しているときにおこなうことが望ましい。そこで、移動ロボットが各分散センサの観測領域内に存在するときに地図の比較をおこなう。分散センサと移動ロボットが生成した地図に同一の物体が存在すれば、その物体をランドマークとして分散センサと移動ロボットの位置関係を求めることができる。そのため、移

動ロボットは、各分散センサの観測領域内を通過する経路を走行する。

図 3.1 の (i)から(iii)に、提案手法の流れを示す。(i)は、初期位置から地図を生成しながら走行する移動ロボットが、分散センサの観測領域に進入した様子を示す。このとき、分散センサは移動ロボットとの地図比較を開始する。(ii)は、移動ロボットによる大域地図と分散センサによる局所地図の比較に基づいて、世界座標系における大域地図上での分散センサの位置推定をおこなう様子を示している。(iii)では(ii)の結果を用いて、分散センサの推定位置を更新している。移動ロボットは、大域地図を生成しながら、分散センサの観測領域内を通過するたびに(i)から(iii)を繰り返し、各分散センサの位置を更新する。また、更新した分散センサ位置に基づき更新した局所地図を、移動ロボットによる大域地図の生成にも反映させる。SLAM における距離データと地図の対応関係において、移動ロボットが生成する地図だけでなく、世界座標系に変換された分散センサの局所地図との対応関係の結果も評価する。双方の地図により良く対応する世界座標系の位置を、SLAM による移動ロボットの自己位置推定値として採用していく。このように、空間に設置された分散センサによる地図も利用することで、移動ロボットの SLAM における自己位置および地図の推定精度も向上することが期待できる。

## 3.2 分散センサの地図生成

### 3.2.1 分散センサの動作

本研究で使用する分散センサは、距離データを基にした平面地図を生成するために、レーザ式測域センサを用いる[39]。レーザ式測域センサは、センサから照射したレーザが周辺物体までの距離と方向を取得することで周辺空間の形状を計測することができる。

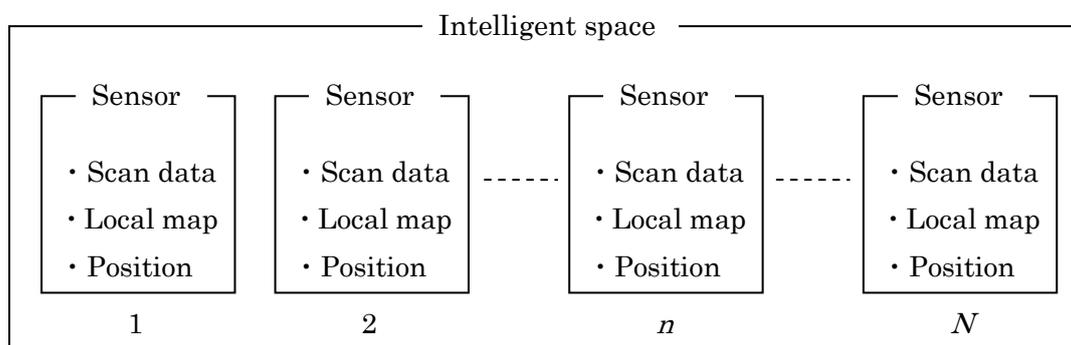


図 3.2 分散センサのシステム構成図

図 3.2 に分散センサのシステム構成図を示す。分散センサは、任意の空間内に複数台設置される。設置された分散センサによって状態認識される空間を知能化空間として構築する。図 3.2 では  $N$  台の分散センサが、知能化空間を構成する図を示している。初期値として、分散センサには世界座標系における任意の座標値 **Position** が与えられており、地図は無い状態である。各センサは、周辺物体までの距離データを **Scan data** として取得する。次に、取得した **Scan data** を基に地図を生成する。生成された地図は、分散センサ周辺の形状を表した局所地図を **Local map** とする。**Local map** は、世界座標系における分散センサの位置を基準にした座標系で表わされる。各分散センサの地図生成は、**Scan data** が取得されるたびに、それぞれ独立しておこなわれる。各分散センサの自己位置 **Position** は、分散センサが独立して動作している間は更新されない。各分散センサの生成した地図と位置座標は、知能化空間全域の空間情報として一括して管理する。

### 3.2.2 レーザ式測域センサ

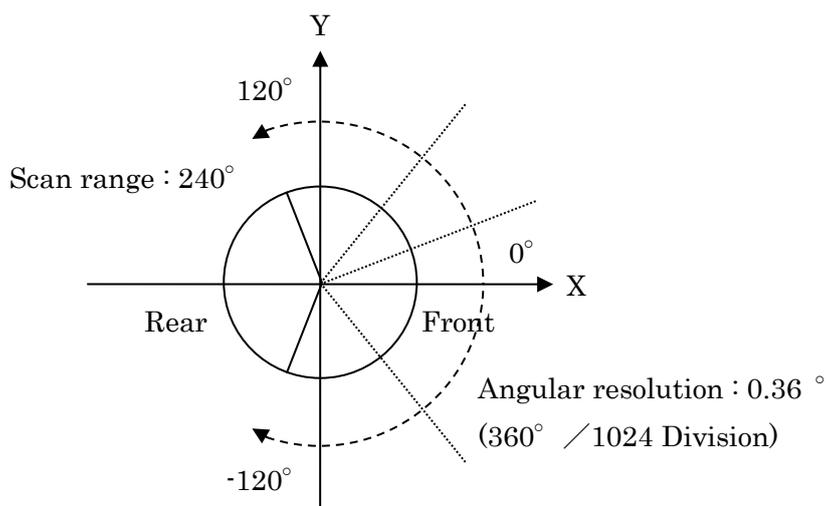


図 3.3 分散センサの座標系

本節では、分散センサによる地図生成手法を記す。本手法で使用する地図は全て平面地図であり、平面地図の生成のために分散センサとしてレーザ式測域センサを用いる。図 3.3 には、本研究の分散センサとして使用したレーザ式測域センサの仕様を記す。図 3.3 は、レーザ式測域センサを上部から見た図である。センサの平面座標系は、センサを原点としてセンサの前方を X 軸、左方を Y 軸として定める。本研究の分散センサは、レーザ光により水平面状の空間を 0.36 度ピッチで 240 度スキャンし、検出体との距離と方向を検出できる。

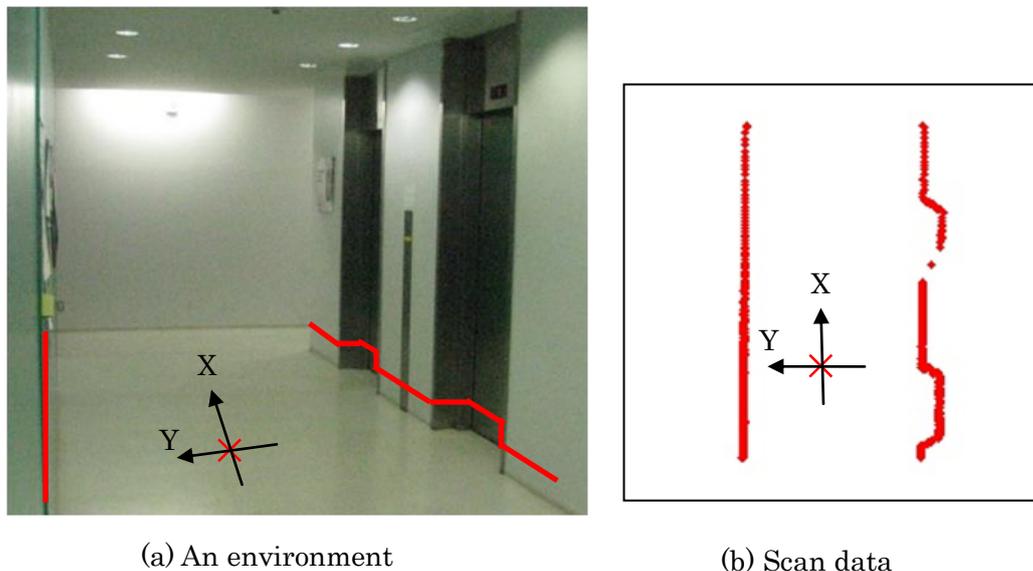


図 3.4 レーザ式測域センサの使用例

図 3.4 には、本研究で使用するレーザ式測域センサの使用例を示す。図 3.4 (a)には、センサを設置した実際の室内空間の写真である。x 印は、センサを設置した位置を示している。センサの設置方向は座標系に示すとおりである。写真に赤色直線で示した部分がセンサによってスキャンされる箇所である。図 3.4 (b)は、図 3.4 (a)のセンサが実際に観測した周辺物体までの距離と方向を、距離データとして座標変換して平面座標上に表示したものである。表示には、オープンソースのコンピュータビジョン用ライブラリである OpenCV を用いた[40][41]。図 4.2 (a)と同様に、x 印は、センサを設置した位置を示しており、センサの設置方向は座標系に示すとおりである。図 3.4 (a)と 図 3.4 (b)を比較するとセンサの距離データから適切な室内空間の形状が得られていることが確認できる。

### 3.2.3 分散センサによる地図生成手法

分散センサが地図生成をおこなう際は、上記のレーザ式測域センサの距離データを基にする。本研究で用いる地図は、占有格子確率を利用した格子地図である。図 3.5 に格子地図の一例を示す。図 3.5 (a)はある室内空間の距離データをセンサにより取得した様子である。図 3.5 (b)は図 3.5 (a)の距離データを基に生成した格子地図を示している。距離データが観測した室内空間の壁の部分を黒色で示している。占有格子確率による格子地図を用いることで、距離データで観測した移動物体を除去した地図生成をおこなうことができる。図 3.5 (a)の距離データには丸枠線で囲った部分に示すように、移動物体である歩行者の足首が観測されている。しかし、格子地図を用いると、図 3.5 (b)に丸枠線で囲った部分に示すよう

に歩行者の足首を除去した地図生成ができる。

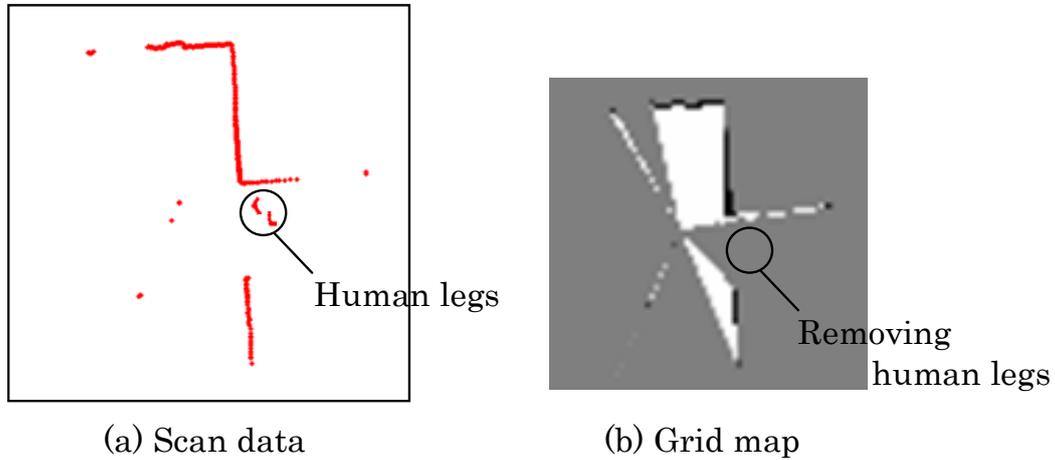


図 3.5 格子地図の生成例

距離データは、センサを基準とした座標系に、センサにより取得した周辺物体までの距離と方向を座標変換した観測物体の座標である。格子地図は、地図を格子状に分割して、各格子に物体の存在する確率を表す占有格子確率を求めることで生成される。センサを基準とした座標系を格子状に分割して、その座標系に座標変換したセンサの距離データを基に各格子の物体の存在確率を算出する。各格子の物体の存在確率は、占有格子確率  $p(m_i|z_{1:t}, x_{1:t})$  を用いて表される[37][38]。占有格子確率は、対数オッズ  $l_{t,i}$  を用いて式 (3.1) のように求められる。対数オッズ  $l_{t,i}$  は全ての実数で表わされる。

$$p(m_i|z_{1:t}, x_{1:t}) = 1 - 1/(1 + \exp(l_{t,i})) \quad (3.1)$$

ここで  $m_i$  は格子地図の各格子であり、 $z_{1:t}$  は時刻  $t$  までの計測値、 $x_{1:t}$  は時刻  $t$  までのセンサの位置である。占有格子確率  $p(m_i|z_{1:t}, x_{1:t})$  は 0 から 1 までの実数値で表され、時刻  $t$  までの距離データを座標が対応する格子ごとに蓄積することで各格子の物体の存在の有無を表現することができる。式 (3.1) の計算を図 3.5 (b) の様な格子地図の格子すべてに対しておこなう。対数オッズ  $l_{t,i}$  の算出方法を含む地図生成方法の詳細は、図 3.6、図 3.7 と併せて説明する。

```

1: Algorithm occupancy_grid_mapping ( $\{l_{t-1,i}\}, x_t, z_t$ )
2:   for all cells  $m_i$  do
3:     if  $m_i$  in perceptual field of  $z_t$  then
4:        $l_{t,i} = l_{t-1,i} + \text{inverse\_sensor\_model}(m_i, x_t, z_t)$ 
5:     else
6:        $l_{t,i} = l_{t-1,i}$ 
7:     end if
8:   end for
9:   return  $\{l_{t,i}\}$ 

```

図 3.6 格子地図生成手法

```

1: Algorithm inverse_range_sensor_model ( $m_i, x_t, z_t$ )
2:   Let  $x_i, y_i$  be the center-of-mass of  $m_i$ 
3:    $r = \sqrt{(x_i - x)^2 + (y_i - y)^2}$ 
4:    $\phi = \text{atan2}(y_i - y, x_i - x) - \theta$ 
5:    $k = \text{argmin}_j | \phi - \theta_{j,sens} |$ 
6:   if  $r > \min(z_{max}, z_t^k + \alpha/2)$  or  $| \phi - \theta_{k,sens} | > \beta/2$  then
7:     return  $l_o$ 
8:   if  $z_t^k < z_{max}$  and  $|r - z_t^k| > \alpha/2$ 
9:     return  $l_{occ}$ 
10:  if  $r \leq z_t^k$ 
11:    return  $l_{free}$ 
12:  end if

```

図 3.7 対数オッズ算出手法

図 3.6 は占有格子地図を生成する手法である。図 3.6 は、全ての格子  $i$  に対して繰り返し演算をおこない、計測  $z_t^k$  の範囲内にある格子について対数オッズ  $l_{t,i}$  を更新する。範囲内にある格子に対しては、図 3.6 の 4 行目の  $\text{inverse\_sensor\_model}(m_i, x_t, z_t)$  関数で占有値を更新する。それ以外の格子は、図 3.6 の 6 行目にあるようにそのままにされる。図 3.7 は図

3.6 の対数オッズ  $l_{t,i}$  を更新する手法である。格子  $m_i$  に含まれる距離データのインデックス (ピッチ)  $k$  と距離  $r$  を特定することで求められる。センサの座標を  $x_t = (x, y, \theta)$ 、各格子の中心の座標を  $(x_i, y_i)$  とする。この計算は図 3.7 の 2 行目から 5 行目でおこなわれている。 $\theta_{j,sens}$  は測域センサにより観測された物体までの角度、 $z_t^k$  は測域センサにより観測された距離データである。 $\alpha$  は障害物の厚さ、 $\beta$  はセンサの観測幅である。7 行目では、格子が計測範囲から外れているか、計測された距離  $z_t^k$  よりもさらに  $\alpha/2$  遠い場合、対数オッズで表現された値を返す。格子が計測された距離から  $\pm \alpha/2$  離れていなければ、9 行目で  $l_{occ}$  が返される。格子がそれより近ければ、11 行目で  $l_{free}$  が返される。

また、取得した距離データと生成した格子地図を用いて簡易な移動物体の検知が可能である。以下にレーザ式測域センサによる簡易移動物体検知の方法を記す。距離データを取得して地図生成をおこなったときに、距離データと格子地図を比較して格子地図に存在しない距離データがあれば、移動物体が存在すると判断する。距離データはセンサの周辺物体までの距離データであり、壁のような静止物体や人間のような移動物体の距離データを含んでいる。しかし格子地図は、占有格子確率により移動物体を除去した地図生成をおこなうため、静止物体のみが反映された地図が生成される。そこで距離データと格子地図を比較することで、距離データに含まれる移動物体が差異として求められる。

```

1: Algorithm moving_object_detection( $p(m_i|z_{1:t}, x_{1:t}), x_t, z_t$ )
2:   for all cells  $m_i$  do
3:     if  $m_i$  in perceptual field of  $z_t$  then
4:       if  $p(m_i|z_{1:t}, x_{1:t}) > occ$ 
5:         return 0
6:       else
7:         return 1
8:       end if
9:     end if
10:  end for

```

図 3.8 移動物体検知手法

図 3.8 に移動物体検知の手法を記す。格子地図の全ての格子  $m_i$  において、距離データの観測値  $z_t$  の座標値が含まれる格子の占有格子確率を評価する。占有格子確率が  $occ$  より大きければ移動物体存在の判断値に 0 を返す。占有格子確率が  $occ$  以下であれば移動物体存在の判断値に 1 を返す。 $occ$  は、占有格子確率における物体の存在の有無を判断する閾値であ

る。図 3.5 の地図生成の例と併せて移動体検知を説明すると、まず、図 3.5 (a) の距離データを図 3.5 (b)の格子地図の座標系へ変換して、照らし合わせる。次に、距離データが存在するが占有格子確率が  $occ$  以下の格子がある場合は、その距離データは移動物体と判断する。この比較を全ての距離データに対しておこない、いずれかの格子に移動体検知の結果として 1 が返された場合は、移動物体が存在すると判断する。全ての格子に対して 0 が返される場合は、移動物体が存在しないと判断する。1 が返される場合は、距離データと格子地図の対応に一致しない箇所があり、0 が返される場合は、距離データと格子地図の対応が一致することを示している。

### 3.3 移動ロボットの自己位置推定と地図生成

#### 3.3.1 移動ロボットの動作

本研究で使用する移動ロボットは、基本的な動作機構である車輪によって移動をおこなう。車輪型移動ロボットは、二足歩行型移動ロボットなどに比べ、車体の重心の移動が少ないため走行制御が容易である。自己位置については、車輪の回転を基にしたホイールオドメトリによって簡易的な推定ができる。

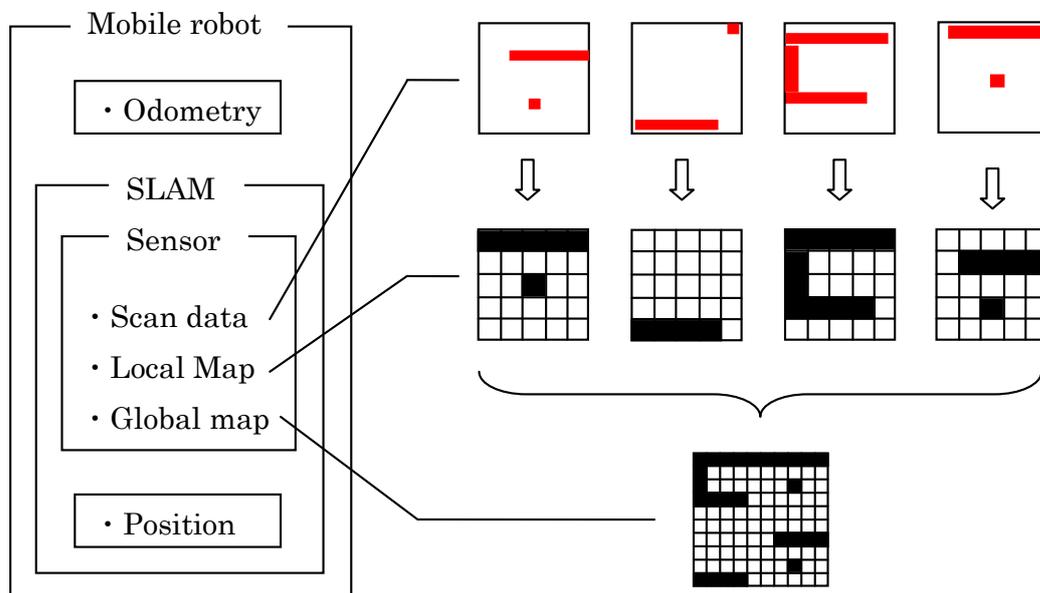


図 3.9 移動ロボットのシステム構成図

図 3.9 には、本研究で用いる移動ロボットのシステム構成図を示す。初期値として、移動ロボットには自己位置座標（原点座標： $(0, 0, 0)$ ）のみが与えられ、地図は無い状態である。

移動ロボットは、ホイールオドメトリによる自己位置 **Odometry** を随時取得できる。ホイールオドメトリには推定誤差が含まれるため、**SLAM** による位置推定をおこなう。**SLAM** は、移動ロボットが独立して地図を生成することで、周囲の物体と移動ロボットの位置関係から移動ロボットの自己位置を推定する手法である。移動ロボットは、地図生成のためにレーザ式測域センサを搭載しており、分散センサとして用いるレーザ式測域センサと同様のセンサである。移動ロボットに搭載したセンサが取得するデータは、分散センサと同様に周辺物体までの距離データ **Scan data** である。取得した **Scan data** を基に地図を生成する手法についても分散センサと同様の手法によって、局所地図 **Local map** を生成する。移動ロボットは走行しながら局所地図を生成するため、地図生成のたびに移動ロボットの位置が変化する。計測基準位置が地図生成のたびに異なるため、ほとんどの場合、分散センサの局所地図とは異なり、局所地図の形状は生成されるたびに異なる。生成された局所地図には、それぞれの地図が生成されたときの移動ロボットの位置が記録されている。**SLAM** では、生成されたときの移動ロボットの位置が記録された全ての局所地図と、ホイールオドメトリによる誤差を含む現在の移動ロボットの位置を基準にした **Scan data** を比較する。**Scan data** と対応する形状を多く含む局所地図が生成されたときの移動ロボットの位置を考慮して、現在の局所地図の生成と現在位置 **Position** の推定をおこなう。また、移動ロボットが生成した形状の異なる全ての局所地図を統合することで、移動ロボットが走行した全経路における大域地図 **Global map** を生成することができる。形状の異なる局所地図には、それぞれの地図が生成されたときの移動ロボットの位置が保存されている。そのため、各局所地図の計測基準点がわかるので、全ての局所地図を統合した大域的な座標系へ座標変換することで全局所地図を統合した大域地図として表すことができる。この大域地図の座標系は、移動ロボットの初期位置を原点とした世界座標系としている。

### 3.3.2 ホイールオドメトリによる動作モデル

本研究で用いる移動ロボットは、基本的な移動機構の一つである車輪型移動ロボットである[42]。車輪型移動ロボットの中でも二輪速度差方式は、構造が単純で制限が比較的容易である。本研究では、二輪速度差方式の移動ロボットを用いておこなう。本研究で用いる移動ロボットのステアリング方式や動作制御手法の詳細は付録 A に記す[43][44]。

二輪速度差方式の移動ロボットの座標系を図 3.10 のように設定する。本実験では、移動ロボットの座標系には、移動ロボットの位置を原点とした局所的な座標系と移動ロボットが走行を開始した位置を原点とした大域的な座標系の二種類がある。座標系は、二次元の平面座標系であり、大域的な座標系を世界座標系とする。世界座標系上の移動ロボットの位置座標を  $(x, y, \theta)$  とする。

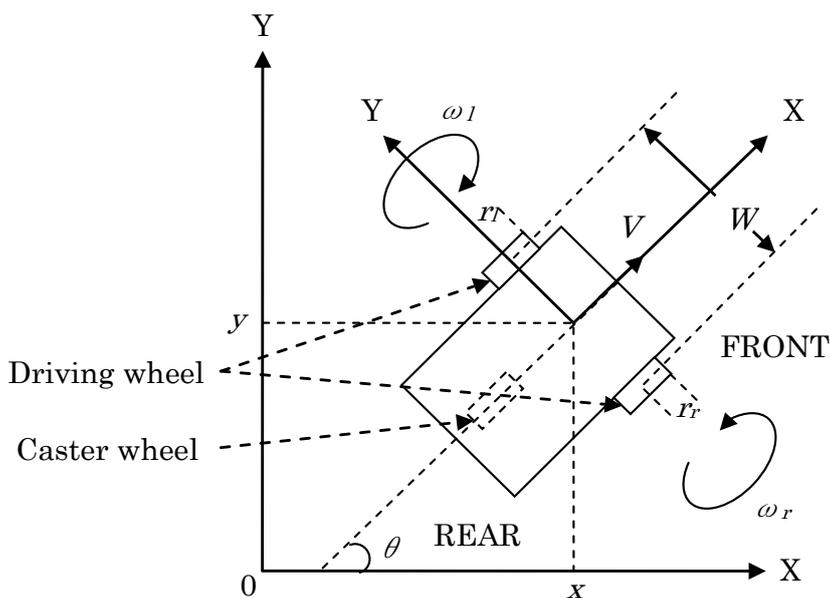


図 3.10 移動ロボットの座標系

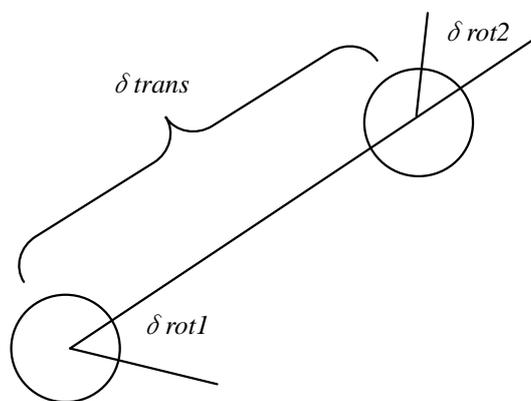


図 3.11 オドメトリ動作モデル

図 3.11 にホイールオドメトリによる動作モデルを示す。移動モデルを定式化することで、移動ロボットのホイールオドメトリによる位置推定と走行制御をおこなう。 $\delta rot1$  は時刻  $t-1$  における移動ロボットの状態変数  $\theta_{t-1}$  を表しており、以下の式 (3.2) のように求められる。

$$\delta rot1 = atan2(y_t - y_{t-1}, x_t - x_{t-1}) \quad (3.2)$$

$\delta trans$  は時刻  $t-1$  から  $t$  迄に進んだ距離を表しており、以下の式 (3.3) のように求められる。

$$\delta trans = \sqrt{(x_t - x_{t-1})^2 + (y_t - y_{t-1})^2} \quad (3.3)$$

$\delta rot2$  は時刻  $t$  における移動ロボットの状態変数  $\theta_t$  を表しており、以下の式 (3.4) のように求められる。

$$\delta rot2 = \theta_t - \theta_{t-1} - \delta rot1 \quad (3.4)$$

$x_t, y_t, \theta_t$  は時刻  $t$  における移動ロボットの  $x, y, \theta$  座標を表しており、以下の式 (3.5) のように求められる。

$$\begin{pmatrix} x_t \\ y_t \\ \theta_t \end{pmatrix} = \begin{pmatrix} x_{t-1} \\ y_{t-1} \\ \theta_{t-1} \end{pmatrix} + \begin{pmatrix} \delta trans \cos(\mu_{t-1, \theta} + \delta rot1) \\ \delta trans \sin(\mu_{t-1, \theta} + \delta rot2) \\ \delta rot1 + \delta rot2 \end{pmatrix} \quad (3.5)$$

式(3.2)から(3.5)を用いて、移動ロボットの位置座標をホイールオドメトリにより求めることができる。また、付録 A の走行制御方法に基づき、求められたホイールオドメトリの位置を用いて走行制御のための目標速度を算出することができる。移動ロボットの位置から目標地点までに必要な速度、角速度を求め、式(3.6)、式(3.7)より走行制御をおこなう。式(3.6)より移動ロボットの左右の車輪の角速度から移動ロボットの速度  $v$ 、角速度  $\omega$  を求める。また、式(3.7)より、移動ロボットの速度  $v$ 、角速度  $\omega$  から目標地点まで移動するために必要な左右の車輪の速度を求めることができる。ホイールオドメトリによる移動ロボットの動作確認の詳細は付録 A に記す。

$$\begin{pmatrix} v \\ \omega \end{pmatrix} = \begin{pmatrix} \frac{r_r}{2} & \frac{r_l}{2} \\ \frac{r_r}{W} & -\frac{r_l}{W} \end{pmatrix} \begin{pmatrix} \omega_r \\ \omega_l \end{pmatrix} \quad (3.6)$$

$$\begin{pmatrix} v_r \\ v_l \end{pmatrix} = \begin{pmatrix} 1 & \frac{W}{2} \\ 1 & -\frac{W}{2} \end{pmatrix} \begin{pmatrix} v \\ \omega \end{pmatrix} \quad (3.7)$$

### 3.3.3 移動ロボットの地図生成

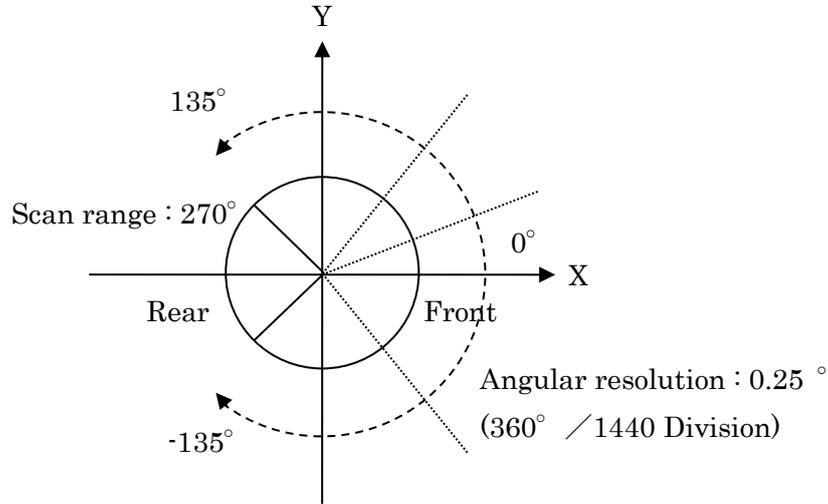


図 3.12 移動ロボットに搭載したレーザ式測域センサの仕様図

移動ロボットは、自律して地図生成をおこなうためにレーザ式測域センサを 1 台搭載する。図 3.12 に移動ロボットに搭載したレーザ式測域センサの仕様図を示す。センサの原点は移動ロボットの基準点と一致させることで位置と方向を移動ロボットの座標系と統一させる。なお、移動ロボットの角度表記範囲は  $0 < \theta \leq 180$ 、 $-180 < \theta \leq 0$  である。搭載したセンサの仕様と地図生成手法は、知能化空間を構成する分散センサで用いるものと同様である。図 3.6 と図 3.7 に記した分散センサの地図生成方法におけるセンサの座標  $x_t = (x, y, \theta)$  において、移動ロボットの位置座標に置き換えた手法が、移動ロボットの地図生成方法になる。生成される地図は、各時々における移動ロボットの位置座標を基準にして生成した局所地図である。本研究では、移動ロボットの走行開始時の座標系を世界座標系としている。移動ロボットが走行しながら生成した各局所地図を世界座標系に変換して統合した地図が、移動ロボットが生成した全領域の大局地図として生成される。

### 3.3.4 移動ロボットの SLAM

本研究では、移動ロボットは Particle Filter を用いた FastSLAM[45]により地図生成と自己位置推定をおこなっており、その手法を紹介する。SLAM は、初めに移動ロボットが環境の地図情報がなく、移動ロボットの位置も分からない場合に、移動ロボットが自律して地図生成と位置推定をおこなう手法である。SLAM では、移動ロボットに搭載したセンサからの距離データと移動ロボットのホイールオドメトリによる位置座標のみから地図生成と自己位置推定をおこなう。SLAM には、これまでに様々な手法が提案されており、歴史的に最も初期の SLAM は拡張カルマンフィルタ (EKF : Extended Kalman Fiter)を基に

した EKF-SLAM[46]があるが、本研究では、比較的高精度かつ高速に動作するパーティクルフィルタを利用した FastSLAM を用いる[37][38]。パーティクルフィルタは、標本の集合で分布を表現する手法である。この手法はガウス分布などよりも広範囲の分布を表現できる。しかし、パーティクルフィルタは、次元によってガウス分布よりも計算量が増大してしまう。パーティクルフィルタは、地図を記述するために数多くの変数が必要なため、単純に SLAM を実装すると困難な場合がある。SLAM において、ランドマークの対応関係があらかじめ分かっている場合は、移動ロボットの姿勢を考慮することで、地図中で連結していない任意の特徴量すべての位置を互いに求めることができる。そのため特徴量の位置推定は、移動ロボットの姿勢が不確かなときに生じる。この従属関係を用いて、パーティクルフィルタを SLAM に適用する。パーティクルフィルタでは、いくつかの変数に対する事後確率分布だけがパーティクルフィルタで表現され、他の変数の表現にはガウス分布やその他の確率関数が用いられる。

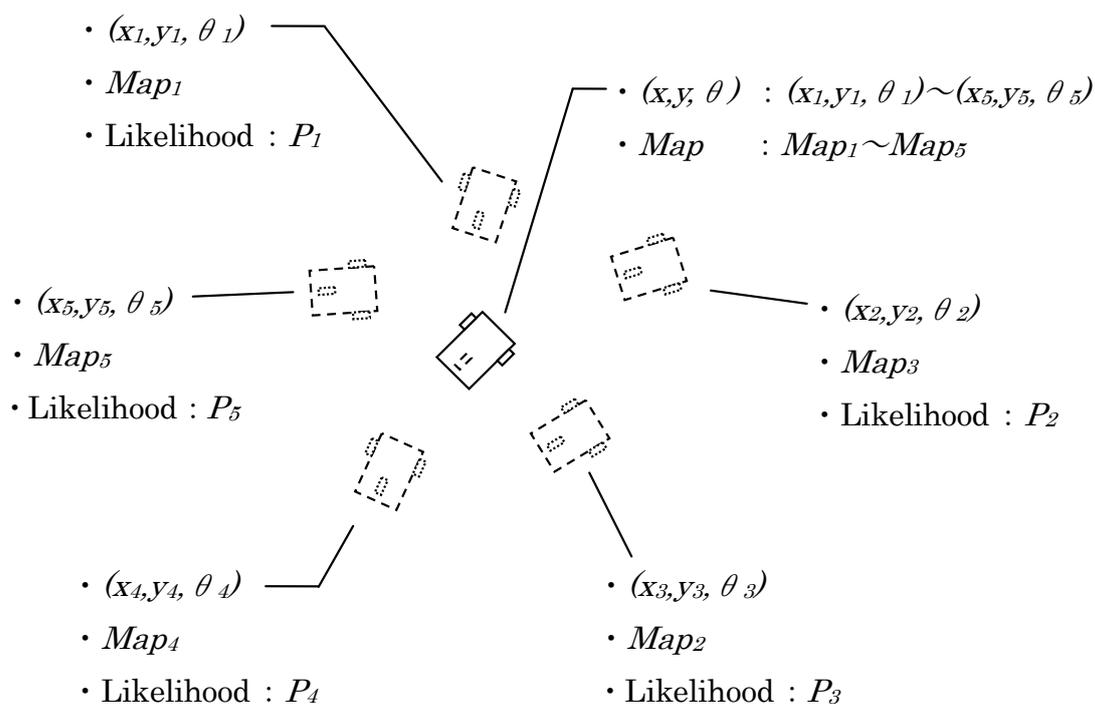


図 3.13 パーティクルフィルタによる移動ロボットの位置推定例

FastSLAM では、移動ロボットの軌跡の推定にパーティクルフィルタが用いられる。図 3.13 に移動ロボットにおけるパーティクルフィルタの適用例を示す。図はパーティクル数を 5 個としたときの例である。移動ロボットの推定値は、図中央の実線で描かれた移動ロボットとして示し、その周囲に分布する 5 つの破線で描かれた移動ロボットは各パーティクルの推定値を示す。位置推定  $(x_i, y_i, \theta_i)$  ( $i = 0, 1, 2, 3$ ) と地図生成  $Map_i$  はパーティクル毎に多数

の別々な特徴量として分けられ、それらを基にパーティクルの尤度  $P_i$  が求められる。移動ロボットの推定値は各パーティクルの尤度を考慮して求められる。**FastSLAM** はデータ対応付けの判断がパーティクル 1 つ 1 つに対しておこなわれるため、フィルタでは、1 つではなく複数のデータ対応付けに対する事後確率が記録され続ける。データの対応付けをおこなうたびに、1 つのデータ対応付けしか追従しないのではなく、複数に分布させたデータ対応付けをおこなうことで、**FastSLAM** は最尤な対応付けをおこなう。そうすることでデータ対応付けの事後確率分布を近似できる。複数の対応付けを同時に追跡できる能力があるため、**FastSLAM** には、逐次的な最尤データ対応付けに基づいた手法よりもロバスト性がある。一方、パーティクルフィルタは非線形な移動ロボットの動作モデルに対応している。これは、移動ロボットの運動の非線形性が大きいときや姿勢の不確かさが比較的大きいときに有効である。パーティクルフィルタではデータ対応付けのたびに、一回の姿勢推定をおこなうため **FastSLAM** はオンラインにも対応することができる。また、**FastSLAM** を距離センサに適用すると、占有格子地図を用いた SLAM になる。なお、実装では **FastSLAM** の基盤として、オープンソースである **OpenSLAM** の **GridSLAM** を用いた[47]。

### 3.4 地図比較時の動作

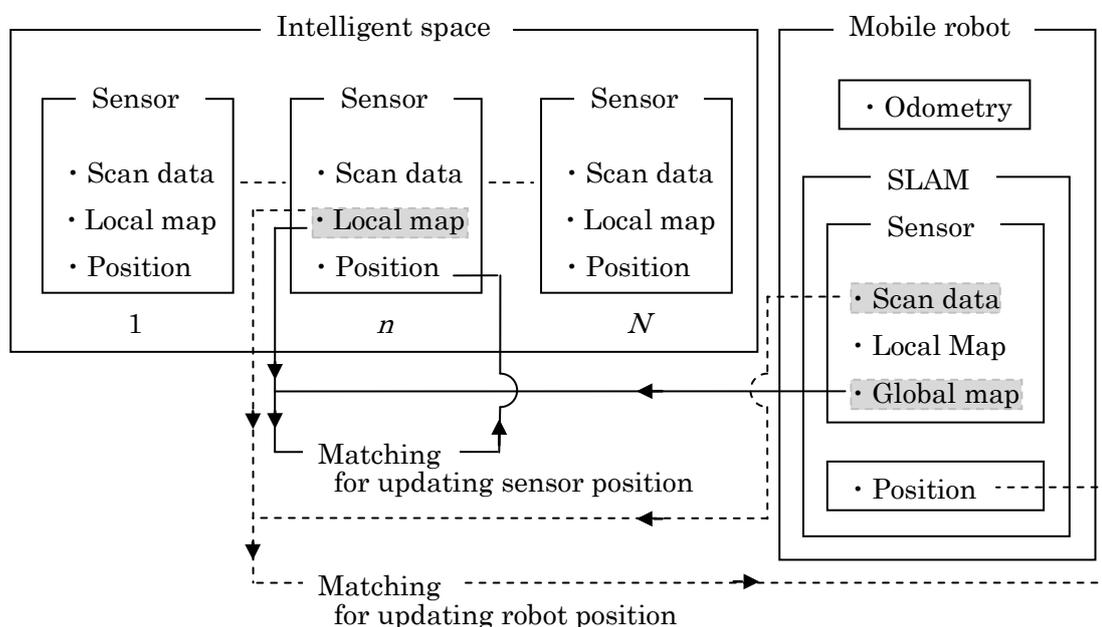


図 3.14 提案手法のシステム構成図

図 3.14 に本研究の提案手法のシステム構成図を示す。本研究の提案手法は、移動ロボッ

トがいずれかの分散センサの観測領域内に存在するときにおこなう。分散センサの観測領域内に移動ロボットが存在するかどうかを判断するために、本研究では、簡易的な移動ロボット検知方法を用いる。全分散センサの位置座標と移動ロボットの位置座標との直線距離が最小の分散センサに注目して、その直線距離が分散センサの観測最大距離内であり、その分散センサが移動物体を検知した際に、移動ロボットが分散センサの観測領域内に存在すると判断する。分散センサと移動ロボットの位置座標の直線距離が分散センサの最大観測距離以内であれば、座標上では移動ロボットが分散センサの観測領域内に存在すると判断できると考えられるためである。

図 3.14 では  $n$  台目の分散センサの観測領域内に移動ロボットが存在する場合の例を示している。まず、図中の実線矢印に示す流れのように、分散センサと移動ロボットは、地図の比較から分散センサの位置座標を推定する。このとき、分散センサの局所地図と移動ロボットの大局地図を比較する。分散センサの局所地図の座標系は、分散センサの位置 **Position** を基に移動ロボットの大局地図の世界座標系へ座標変換される。座標系を統一した地図同士を比較することで、分散センサの局所地図の形状と類似した部分が、移動ロボットの大局地図中のどこにあるのかを探索する。地図の形状探索から、分散センサの世界座標における設置位置 **Position** を推定値として更新する。次に、移動ロボットの **SLAM** の推定精度を向上させるために、位置推定した分散センサの局所地図を利用する。移動ロボットの大局地図との比較により分散センサの位置座標が更新されるため、分散センサの局所地図の座標も更新される。更新された分散センサの局所地図を **SLAM** の地図生成に利用することで、**SLAM** の推定値を修正する。分散センサの局所地図は、図中の破線矢印に示す流れのように、移動ロボットの **SLAM** の **Scan data** と **Local map** の比較の際に用いる。通常の **SLAM** では、移動ロボットの **Scan data** と移動ロボットの **Local map** の比較から移動ロボットの **Position** を推定している。この際に、移動ロボットの **Scan data** と世界座標系に座標変換された分散センサの **Local map** の比較の結果も考慮することで、位置推定された分散センサの局所地図を反映させた移動ロボットの地図生成と自己位置推定をおこなう。この提案手法は、分散センサの観測領域内で移動ロボットが **SLAM** をおこなうたびに繰り返しおこなわれる。分散センサと移動ロボットの地図情報を比較することによって、分散センサでは、お互いに重複した観測領域がなく共通のランドマークを観測できない状態であったが、移動ロボットとの地図比較により共通のランドマークを観測して、自己位置の推定をおこなうことができる。移動ロボットでは、独立しておこなう **SLAM** による自己位置推定には推定誤差は避けられないものであったが、分散センサの地図を用いることで自己位置推定値の誤差を修正することができる。

## 4 分散センサと移動ロボットの地図情報の比較方法

### 4.1 座標変換による地図情報の比較

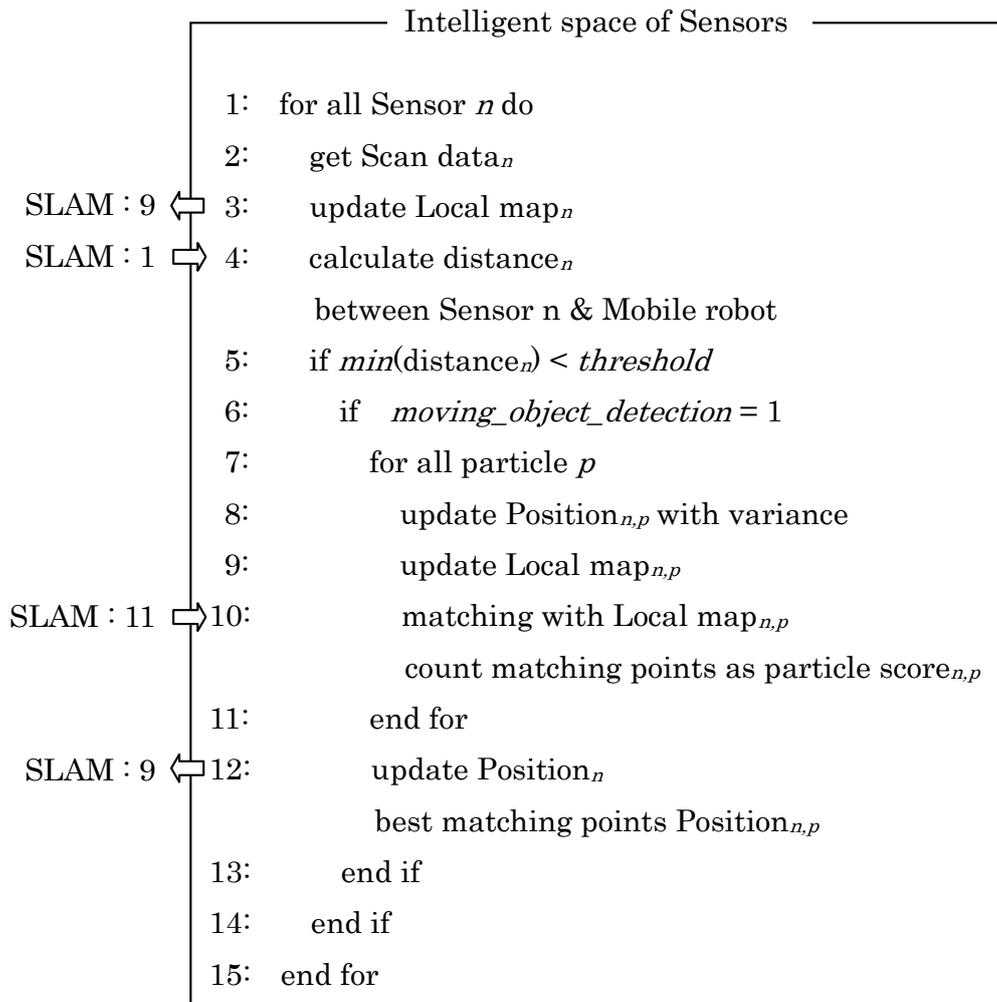


図 4.1 パーティクルフィルタによる分散センサの位置推定手法

本節では、提案手法を評価する事前の検証実験において、分散センサと移動ロボットの地図の比較の際に ICP マッチングや座標系間の角度差を用いずに、分散センサと移動ロボットの座標系を合わせて比較することで、地図の対応点を求める。この事前実験のシステムには RTM[48][49]を用いて実装した。本節の手法における分散センサの位置推定には、移動ロボットの SLAM と同様に、パーティクルフィルタを用いる。パーティクル毎の分散センサの局所地図を移動ロボットの世界座標系に変換して、分散センサと移動ロボットの

地図を比較する。対応点数を分散センサのパーティクルの尤度とする。移動ロボットの SLAM の修正では、位置推定された分散センサの局所地図を移動ロボットの世界座標系に変換して、移動ロボットの各パーティクルの距離データと比較する。対応点を SLAM のパーティクルの尤度へ加算する。

図 4.1 は、パーティクルフィルタを用いた分散センサの位置推定手法を示している。1 行目から 6 行目にかけては、3.1 で解説した分散センサの動作手法と同様である。7 行目以降では分散センサの位置推定に移動ロボットの SLAM と同様のパーティクルフィルタを用いている。8 行目では、分散を与えた分散センサの位置座標を各パーティクルの位置として更新している。9 行目では、更新された位置座標を基に局所地図の座標系を変換している。10 行目では、移動ロボットの局所地図と座標系を重ね合わせて、対応点数を各パーティクルの尤度に行している。12 行目では、最高尤度のパーティクルの位置座標を分散センサの推定位置として更新している。移動ロボットの SLAM の動作手法は 3.4 と同様である。

## 4.2 座標変換による手法の検証

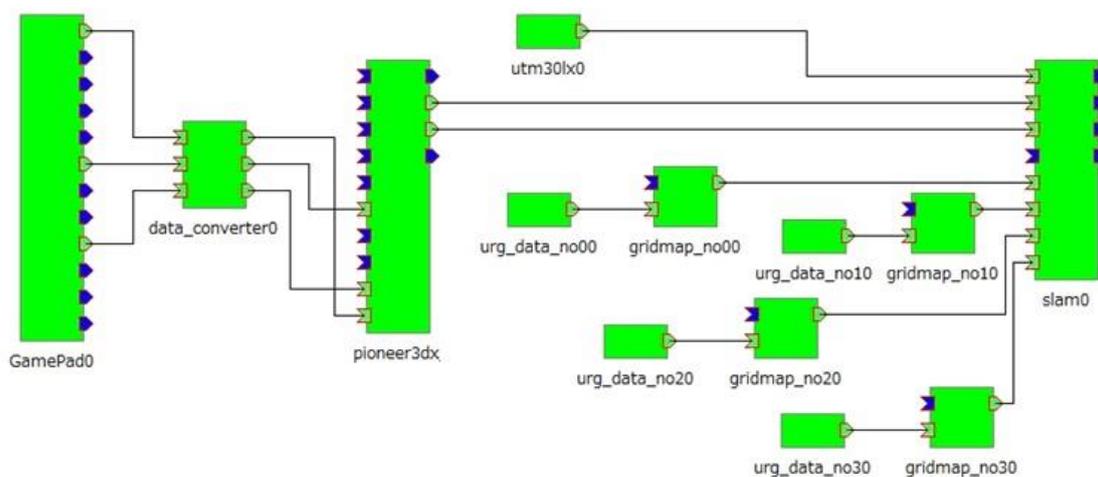


図 4.2 システム概要図

図 4.2 には、本節の手法によるシステムをオブジェクト化した図を示す。システムのオブジェクト化には OpenRTM を用いた。OpenRTM は、独立行政法人産業技術総合研究所が開発をおこなっているロボット技術用のミドルウェアである。ロボットの技術要素であるアクチュエータ、センサや動作制御の手法などを一つのコンポーネントとみなし、これを RT コンポーネント (RT-Component, RTC) と呼ぶ。RT コンポーネントは他の RT コンポーネントとデータ通信するためのポートがあり、複数の RT コンポーネントをポート接続す

ることで、複数のロボット技術から構成されるシステムを実装することができる。RT ミドルウェアは、システムを役割毎にコンポーネントとして分散して構築することができる。

図 4.2 は、事前実験を実装するために、コントローラで動作制御した移動ロボットにより知能化空間内を走行するシステムである。GamePad コンポーネントは移動ロボットの動作制御をおこなうためのコントローラコンポーネントである。data\_converter コンポーネントにより GamePad から移動ロボット用に変換されたデータが、移動ロボットの制御コンポーネントである pioneer3dx コンポーネントへ送信される。pioneer3dx はオドメトリデータを slam コンポーネントへ送信している。utm30lx コンポーネントは移動ロボットに搭載されたレーザ式測域センサであり、随時スキャンデータを取得している。urg\_data\_nox コンポーネントは各分散センサでありスキャンデータを gridmap\_nox コンポーネントへ送信している。gridmap\_nox コンポーネントは格子地図を生成して slam コンポーネントへ送信している。slam コンポーネントでは受信した移動ロボットと分散センサのデータを用いて図 4.1 に示したそれぞれの位置推定をおこなっている。

図 4.3 は本節の手法により移動ロボットが生成した地図を示す。実装環境は図 5.4 に示す。

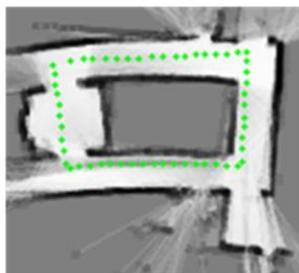


図 4.3 座標重ね合わせ手法による移動ロボットの地図

図 4.4 から図 4.7 には、本節の手法による分散センサの地図と移動ロボットの距離データとの対応の様子を示す。(i)から(iv)は比較をおこなった順序における対応点の様子を表している。分散センサの設置位置は、図 5.4 に示したとおりである。分散センサの格子地図において、座標変換した移動ロボットの距離データを赤色の点で表わしている。また、青色の点は分散センサの格子と移動ロボットの距離データが重なった対応点を表している。

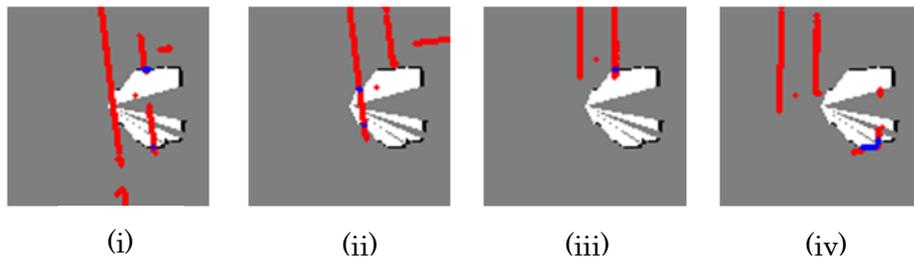


図 4.4 分散センサ No.0 における対応の様子

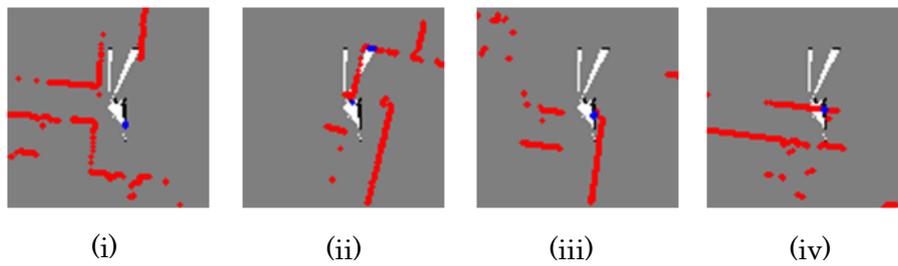


図 4.5 分散センサ No.1 における対応の様子

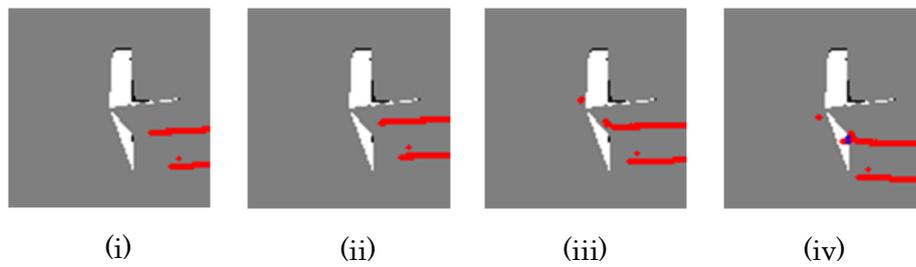


図 4.6 分散センサ No.2 における対応の様子

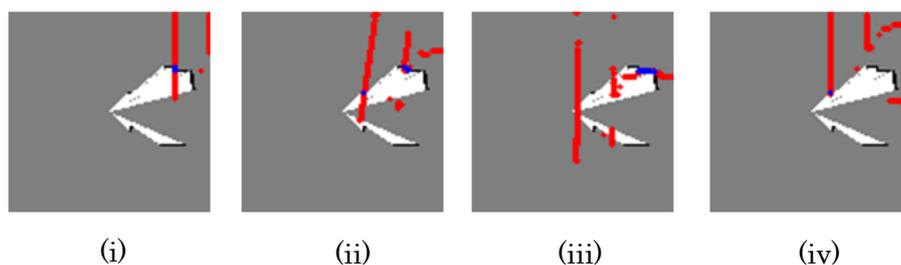


図 4.7 分散センサ No.3 における対応の様子

図 4.3 の地図は、図 5.9(a)移動ロボットが独立しておこなう FastSLAM による地図と比べるとおよそ同じ生成精度であることが確認された。これは図 4.4 から図 4.7 をみると、分散センサの地図と移動ロボットの距離データとの対応点がほとんど得られていないため、SLAM におけるパーティクルフィルタの尤度に加算される対応点数がなく、結果として SLAM へ及ぼす影響がないと考えられる。そのため、移動ロボットが独立しておこなう FastSLAM とあまり変わらない地図生成精度となったと考えられる。各センサとも移動ロボットの距離データとの対応点がほとんど取れていないため、初期値から適切な推定値に更新されない状況も見受けられた。そのことから、繰り返し比較をおこなっても対応点を得られることが少なかったと考えられる。座標の違いにより対応点を得られないことが確認できるが、図 4.5(iii)に示すように、座標同士が近くても座標系の角度の違いによって対応点を得られない場合があることも確認できた。

### 4.3 座標系間の角度差による地図情報の比較

次に、上記の座標系を単純に重ね合わせ対応点を求めた実験の結果を基に座標系の角度の差を考慮した対応点の求め方を考案した。本節では、提案手法を評価する事前の検証実験において、分散センサと移動ロボットの地図の比較方法において、座標系の角度の差を修正することで対応点を増加させる手法を提案する。分散センサと移動ロボットの座標系を変換して単純に位置合わせしただけでは、地図の比較方法において、対応点がほとんど得られない結果がたびたび確認された。対応点を得られない原因は、座標位置の違いよりも、評価実験から分散センサと移動ロボットの座標系の角度の違いによって起こっている様子が見られた。そこで本節の手法では、対応点の数ではなく座標系の推定角度の差に注目した位置推定方法を考案した。

本節の手法では分散センサの位置推定には、移動ロボットの SLAM と同様に、パーティ

クルフィルタを用いる。分散センサの位置推定では、パーティクル毎の分散センサの局所地図を移動ロボットの大局座標系に変換して、分散センサと移動ロボットの地図を比較する。分散センサと移動ロボットの座標系の角度差を分散センサのパーティクルの尤度とする。移動ロボットの SLAM の修正では、分散センサの局所地図を移動ロボットの大局座標系に変換して、分散センサの局所地図と移動ロボットのスキャンデータを比較する。対応点を SLAM のパーティクルの尤度へ加算する。

分散センサと移動ロボットの座標系の角度の算出には、各地図中の代表点 2 点を選出することによる直線の式の傾きを用いる。分散センサの局所地図の代表点 2 点の選出は、占有格子確率が高く、局所地図の原点である分散センサの位置から直線距離で最短の格子と最長の格子の中心座標とする。格子地図において、壁などの静止物体の占有格子確率を求めると、計測開始時から位置座標に変化がほとんど無い。そのため、物体が存在する格子には、ほとんど同等の占有格子確率が算出されることになる。占有格子確率が高い上位 2 点を代表点として選出することは難しいため、さらに観測地点からの距離を制約条件とした。移動ロボットの局所地図の代表点 2 点の選出は、占有格子確率が高く、分散センサの局所地図の代表点と直線距離が最も近い格子の中心座標をそれぞれ代表点とする。

図 4.8 には座標系の角度差を評価する分散センサの位置推定手法を示す。図 4.8 では、パーティクルフィルタによる分散センサの位置推定方法の部分を抜粋して示している。1 行目では、分散を与えた分散センサの位置座標を各パーティクルの位置として更新している。3 行目では、更新された位置座標を基に局所地図の座標系を変換している。4 行目から 8 行目にかけては、更新した局所地図の各格子に対して、局所地図の原点である分散センサの位置から占有格子確率が  $occ$  より大きい格子までの直線距離を求めている。9 行目では、求めた直線距離が最小距離の格子位置を選出している。10 行目では、求めた直線距離が最大距離の格子位置を選出している。11 行目では、求めた最小距離と最大距離の格子の中心座標を通る直線の式の傾きを算出している。12 行目から 17 行目では、移動ロボットの局所地図の占有格子確率が  $occ$  より大きい格子に対して、移動ロボットの座標系に変換した分散センサの局所地図の代表点 2 点との直線距離を求めている。18 行目、19 行目では、求めた直線距離が最小距離の格子位置を選出している。20 行目では、求めた最小距離の格子の中心座標を通る直線の式の傾きを算出している。21 行目では、分散センサと移動ロボットの直線の式の傾きの差を算出して、この差を各パーティクルの尤度としている。移動ロボットの SLAM の動作手法は 3.3 と同様である。

Map matching method with slope

```

1: for all particle  $p$ 
2:   update Position $_{n,p}$  with variance
3:   update Local map $_{n,p}$ 
SLAM : 11 ⇒ //matching with Local map $_{n,p}$ 
                //calculate matching points as particle score $_{n,p}$ 
                //Local map $_{n,p} \Leftrightarrow$  Local map of Mobile robot
                //select 2 representative points in Local map $_{n,p}$ 
4:   for all cell  $m_j$  Local map $_{n,p}$ 
5:     if  $p(m_i|z_{1:t}, x_{1:t}) > occ$ 
6:        $r_i = \sqrt{(x_i - x)^2 + (y_i - y)^2}$ 
7:     end if
8:   end for
                //highest probability points with min & max  $r$ 
9:    $i_{min} = \min(r_i)$ 
10:   $i_{max} = \max(r_i)$ 
                //calculate slope of Local map $_{n,p}$ 
11:   $slope_i = slope(i_{min}, i_{max})$ 
                //select 2 nearest points in Local map of Mobile robot
12:  for all cell  $m_j$  Local map of Mobile robot
13:    if  $p(m_j|z_{1:t}, x_{1:t}) > occ$ 
14:       $r_{j_{min}} = \sqrt{(x_j - x_{i_{min}})^2 + (y_j - y_{i_{min}})^2}$ 
15:       $r_{j_{max}} = \sqrt{(x_j - x_{i_{max}})^2 + (y_j - y_{i_{max}})^2}$ 
16:    end if
17:  end for
                //nearest point by  $(x_{i_{min}}, y_{i_{min}})$ 
18:   $j_{min} = \min(r_i)$ 
                //nearest point by  $(x_{i_{max}}, y_{i_{max}})$ 
19:   $j_{max} = \min(r_i)$ 
                //calculate slope of Local map of Mobile robot
20:   $slope_j = slope(j_{min}, j_{max})$ 
                //calculate slope sub as particle score $_{n,p}$ 
21:   $slope\_sub_p = |slope_i - slope_j|$ 
22: end for

```

図 4.8 傾き比較の手法

## 4.4 ICP マッチングによる地図比較

### 4.4.1 分散センサの位置指定

図 4.9 には、提案手法における分散センサの動作手法を示す。図 4.10 には、提案手法における移動ロボットの SLAM の動作手法を示す。

分散センサと移動ロボットは、提案手法によって地図の比較をおこなうとき以外はそれぞれ独立して動作している。提案手法により地図の比較をおこなう際は、分散センサは図 4.9 の 3 行目、4 行目、7 行目、8 行目に矢印で示す様に、移動ロボットと地図データの比較をおこなう。移動ロボットは図 2.1 の 1 行目、9 行目、13 行目に矢印で示す様に、分散センサと地図データの比較をおこなう。地図の比較は、移動ロボットが SLAM をおこないながら、分散センサの観測領域内を通過するたびに繰り返され、各分散センサの位置を更新する。また、更新された分散センサの推定位置を、移動ロボットの地図生成にも反映させる。移動ロボットが生成する地図だけでなく、更新位置に基づく分散センサの地図とも対応する位置を、SLAM による移動ロボットの自己位置推定値として更新する。

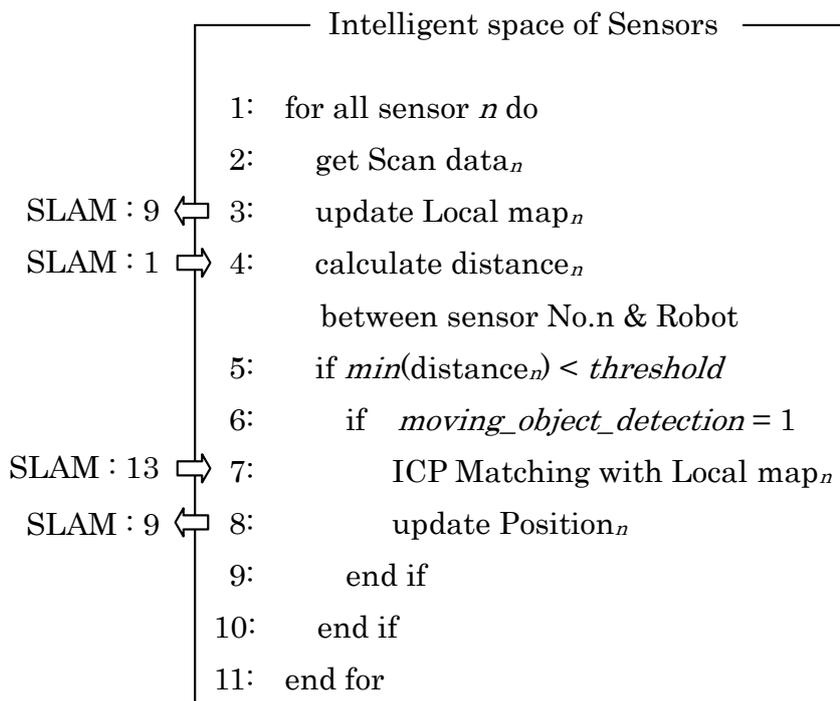


図 4.9 知能化空間の動作手法

図 4.9 において、分散センサは、2 行目、3 行目でそれぞれ距離データから局所地図を生成している。各分散センサの位置座標は、初期値として任意の値が与えられている。生成

した局所地図と分散センサの位置座標は、移動ロボットの SLAM の精度向上へ利用される。4 行目では各分散センサと移動ロボットの位置座標の直線距離を求める。5 行目、6 行目では求めた距離が最小のセンサ No. $n$  において、移動物体が分散センサの観測領域内に存在するかどうかを判断する。7 行目、8 行目では、世界座標系に変換した分散センサの局所地図と移動ロボットの全域地図を比較して、分散センサの位置推定値を更新する。地図の比較には、7 行目に示すように ICP マッチングを用いる[50][51]。ICP による地図比較の際、移動ロボットの全域地図は、分散センサの局所地図に比べ広範囲な地図のため、局所地図との比較に必要な全域地図の領域を切り取る。切り取る領域の決定方法は、分散センサの位置 Position を中心として、局所地図と同じサイズに切り取った領域とする。

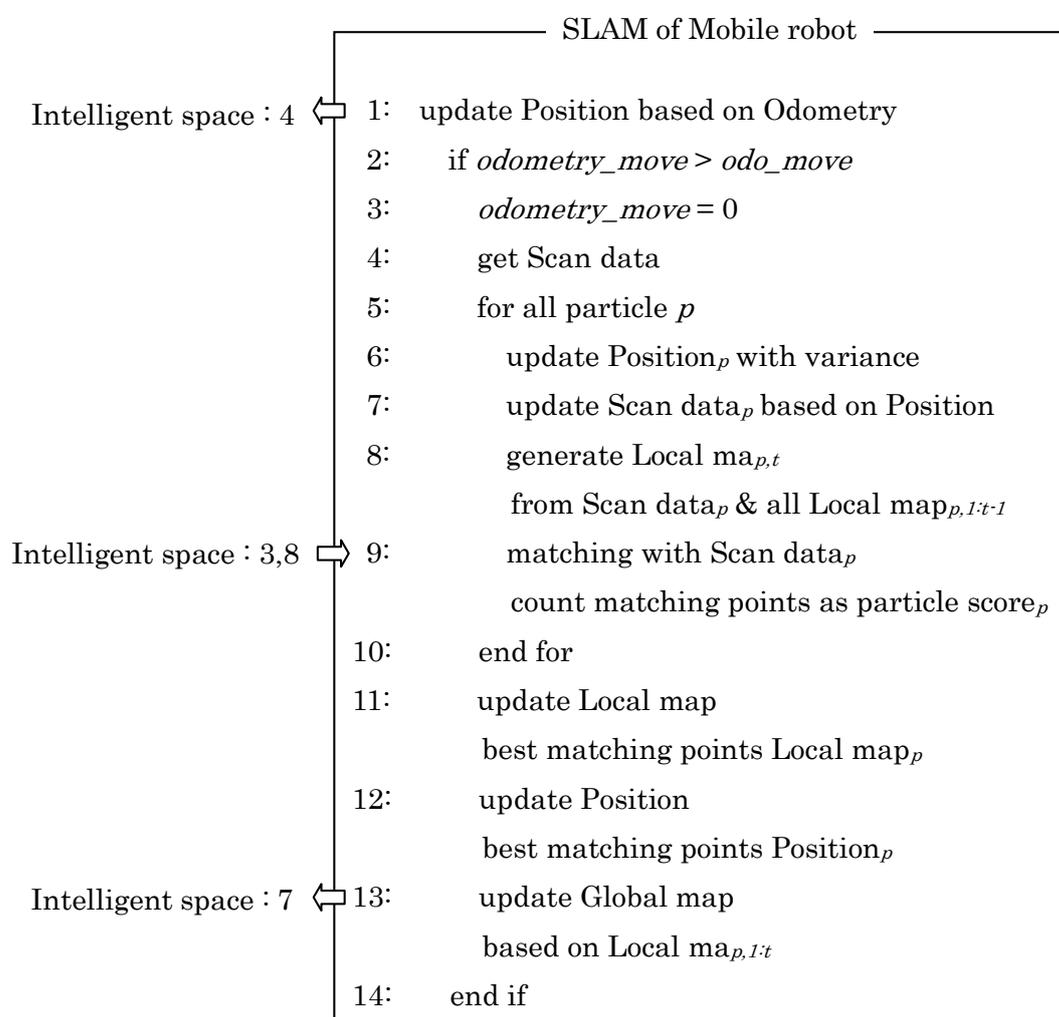


図 4.10 移動ロボットの動作手法

図 4.10 において、移動ロボットの初期位置座標は、 $(x, y, \theta) = (0, 0, 0)$  を与え、これを全域地図の世界座標系として定める。移動ロボットの位置推定は、1 行目に示す様にホ

ホイールオドメトリにより更新されていく。更新された位置座標は分散センサとの直線距離の計算へ用いられる。2行目、3行目では、ホイールオドメトリによる移動距離により SLAM をおこなうかどうかを判断している。ホイールオドメトリによる移動距離が *odo\_move* を超えたときに、位置推定の誤差を修正するために SLAM をおこなう。本研究の SLAM はパーティクル毎に地図生成をおこなうため、SLAM をおこなう周期が短いと、データ量が増えてしまう。時間経過とともにデータ量が増大するにつれて、計算量が増加してしまうことを避けるため、本研究の SLAM は、移動ロボットが一定距離 (*odo\_move*=500mm) をホイールオドメトリで走行した後におこなう。SLAM をおこなう際は、4行目に示す様に、まず移動ロボットに搭載したセンサによる距離データを取得する。次に、6行目、7行目では、ホイールオドメトリと取得した距離データを用いて、パーティクルフィルタにおけるパーティクル毎の Position と Scan data を算出する。ホイールオドメトリに分散を与えた値を各パーティクルの位置座標として更新する。更新された位置座標を基に座標変換した Scan data をそのパーティクルの Scan data とする。8行目では、パーティクル毎の Scan data とこれまでに生成した局所地図の対応関係から、現在の局所地図を生成する。9行目では生成した地図と距離データを比較することで対応点を求め、その対応点の数をパーティクルの尤度とする。移動ロボットが分散センサの観測領域内に存在する場合は、移動ロボットの距離データと世界座標系に変換した分散センサの局所地図との対応点数も求め、その対応点の数をパーティクルの尤度へ加算する。11行目では、最高尤度のパーティクルの地図を移動ロボットの局所地図として更新している。12行目では、最高尤度のパーティクルの位置座標を移動ロボットの推定位置として更新している。13行目では、最高尤度のパーティクルがこれまでに生成した局所地図を移動ロボットの全域座標系へ統合して、全域地図を生成している。生成された全域地図は、分散センサの位置推定のための ICP マッチングによって利用される。

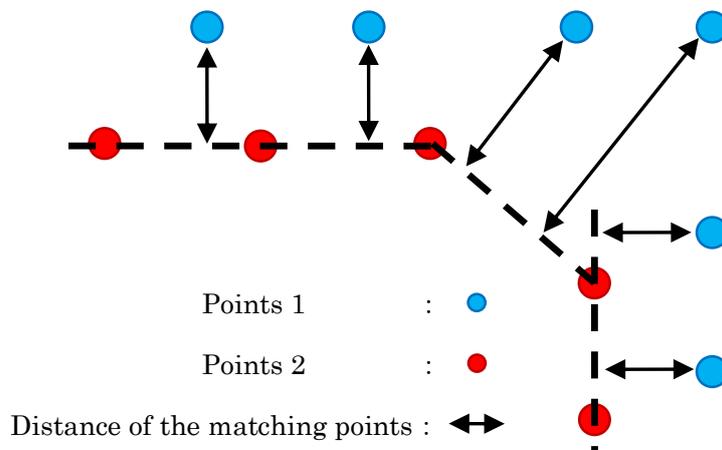


図 4.11 ICP マッチングの概要図

本システムでは、図 4.11 に示すような、ICP マッチングを利用した地図比較手法を適用する。適用した ICP マッチングは、2つの点群と点群間の相対距離を初期値として、点群 1 の各点に対して近接する 2 点を点群 2 の中から選択し、その 2 点を結ぶ直線を対応線とする。そして点群 1 の点と対応線との距離の総和が最小となる相対位置を繰り返し計算により求める手法である。分散センサと移動ロボットの間でより点群間の距離が小さい地図上の位置を探索することで位置・姿勢の差を補償することができ、大域地図上での分散センサの位置を、より適した位置に更新できる。

地図の比較には、4.1 から 4.2 に示す事前の検証実験の原理と結果を考慮して、推定精度に優れた ICP マッチングと座標系間の角度差に注目した手法を用いた。事前実験においても地図の比較がおこなわれるまでは、分散センサと移動ロボットの位置や角度は独立して推定されている。そのため、切り出した移動ロボットの大域地図に、分散センサの局所地図を投影してその対応点を求めても、ほとんど対応点が得られないことがたびたび確認された。このように単純な対応点の比較ではほとんど対応点が取れないため、本システムでは、図 4.8 や図 4.11 に示すような、座標系間角度差や ICP マッチングを利用した地図比較手法を適用する。

#### 4.4.2 移動ロボットのSLAM修正

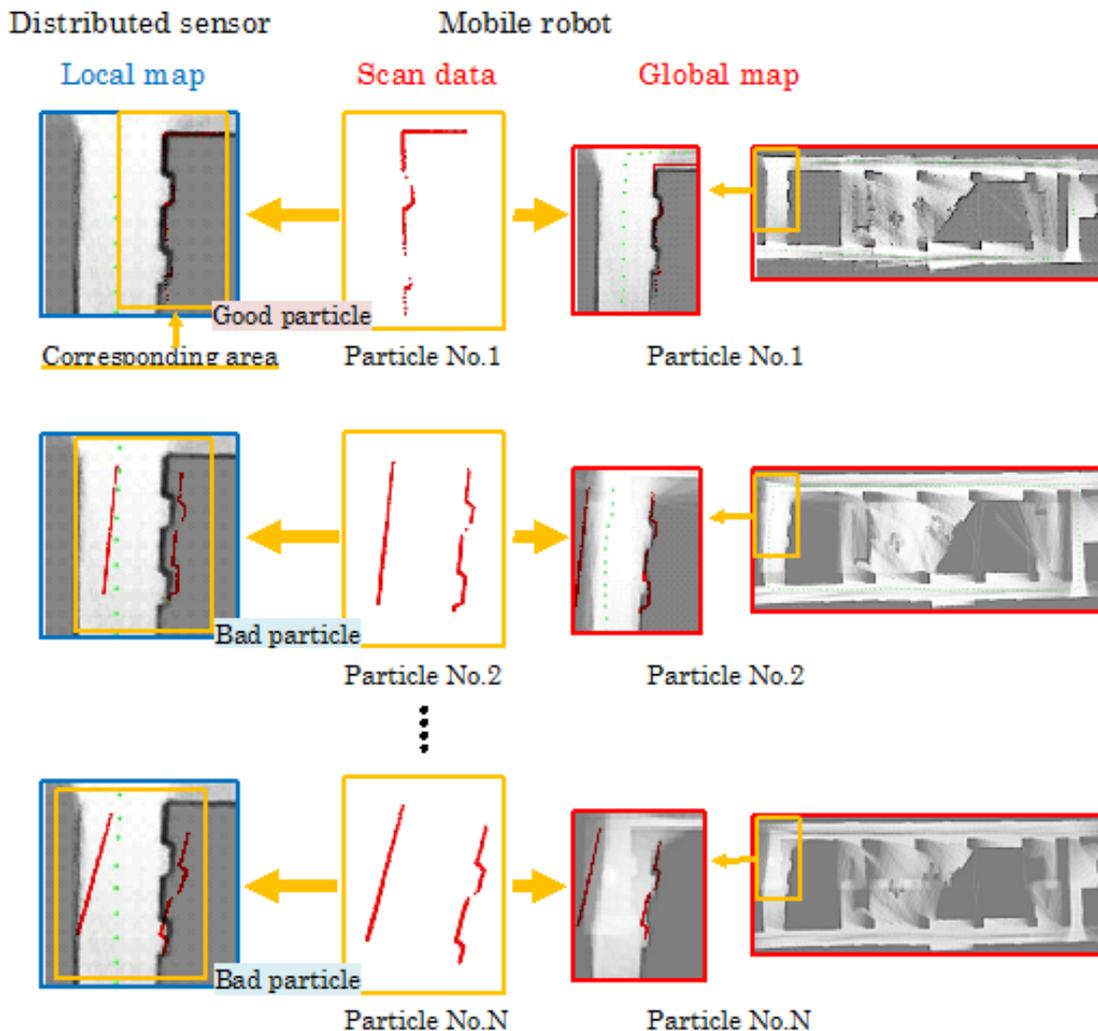


図 4.12 SLAM における分散センサの局所地図利用方法

移動ロボットの全域地図においても、分散センサによる情報は有益である。通常、移動ロボットがSLAMを行なう場合は、それまでに生成した全域地図と、新たな距離データの位置対応により、ロボット位置と地図を更新していく。提案システムでは、新たな距離データを分散センサにおいて生成された局所地図とも比較する。移動ロボットが新たに取得した距離データは、それまで移動ロボット自身が生成した全域地図および、分散センサの推定位置に基づいて世界座標系に変換した分散センサの局所地図との対応関係が考慮される。FastSLAMを構成するパーティクルフィルタの枠組みにおいて、双方の地図によりよく対応する位置のパーティクルを高尤度のパーティクルとする。世界座標系上の移動ロボット位置は、各パーティクルの尤度と自己位置座標を考慮した重み付き平均値を採用していく。これにより、移動ロボットの走行に応じた地図生成に、環境に固定された分散センサの地

図による拘束を加えることとなり、生成される大域地図の精度向上にもつながると考えられる。

図4.12は、SLAMに分散センサの局所地図との位置対応を考慮した流れを示している。通常、移動ロボット側のSLAMでは、パーティクルフィルタにおける各パーティクルで、推定位置からの距離データとそれまでに生成した大域地図との対応関係から尤度評価をおこなうことで、新たな推定位置と大域地図を更新していく。この尤度評価および更新に、図4.22で述べた方法で位置修正をおこなった分散センサの局所地図との比較も加える。移動ロボットは、搭載するレーザセンサにより距離データを取得する。また、パーティクル毎に異なる世界座標上の位置および地図を保持している。そのため、距離データを各パーティクルの保持する位置に基づいて各大域地図に投影した際は、それぞれ異なる位置に投影されることになる。この大域地図に投影された各パーティクルの距離データが、大域地図および対応する分散センサの局所地図と比較される。まず、移動ロボットは、図4.12の右側に示す様に各パーティクルにおいて、移動ロボットがそれまでに生成した大域地図と、前述した距離データの対応格子数を位置対応の結果とする。これは通常のSLAMと同じ流れである。実際には、大域地図中の黄色の枠線に示す様に、距離データの周辺領域のみを抜き出して比較する。一方、分散センサ側では、図4.11にて更新した分散センサの位置に基づいて、図4.12の左側に示す様に各大域地図の座標系に変換した局所地図と、移動ロボットの各パーティクルに対応する距離データを比較し、その対応格子数を求める。この二つの比較による対応格子数を加算したものに基づいて、移動ロボットのパーティクルの尤度を算出する。例えば、図4.12では最上部に示すNo.1のパーティクルの比較は他のパーティクルに比べ、局所地図および大域地図双方とよく対応しているので、良いパーティクルと評価されることになる。この処理は、前述したとおり分散センサの地図および移動ロボットによる距離データ双方で推定した地図に近いパーティクルを重要視することを意味している。最終的には、この結果に基づいて移動ロボットの位置と大域地図を更新する。

## 5 地図比較手法の検証実験

### 5.1 使用機器と実験環境および実験方法

#### 5.1.1 分散センサ

本研究の実験では、知能化空間を構築する分散センサとしてレーザ式測域センサを使用する。図 5.1 に示すセンサは北陽電機社製の URG04-LX-UG01 である。制御機器はワンボードマイコンである Raspberry Pi によっておこなう[52][53][54]。Raspberry Pi への電源供給はモバイルバッテリーからおこない、URG04-LX-UG01 への電源は、USB 接続により供給される。

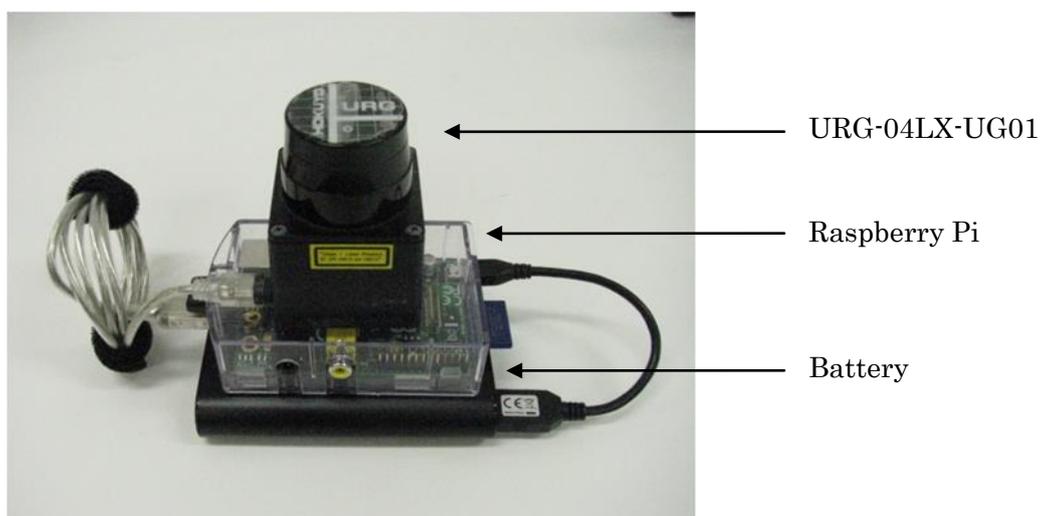


図 5.1 知能化空間を構成する分散センサ

表 5.1 URG04-LX-UG01 の仕様表

電源電圧	DC 5v ±5% (USB から供給)
消費電流	500mA以下 (但し、電源投入時は 800mA)
検出距離及び検出体	精度範囲：60～4,095mm (白ケント紙 70mm□以上) * 測距範囲：20～5,600mm 測距精度 0.06～1m : ±30mm*、1～4m : 距離の 3%以下* (但し、検出体は白ケント紙 70mm□以上)
測距分解能	1mm 単位
走査角度	240 度
角度分解能	約 0.36 度 (360° / 1024 分割)
走査時間	100 ms / scan
インターフェース	USB Ver2.0 FS モード (12Mbps) SCIP2.0
質量	約 160g
外形寸法	50mm×50mm×70mm(W×D×H)

表 5.2 RaspberryPi の仕様表

CPU	700 MHz / ARM1176JZF-S コア (ARM11 ファミリ)
メモリ (SDRAM)	256MB (GPU 共有)
USB 2.0 ポート	2 (統合 USB ハブ)
ストレージ	SD メモリーカード / MMC / SDIO カードスロット
ネットワーク	10/100 Mbps イーサネット (RJ45)
電源	700 mA (3.5 W)
電源ソース	5V / microUSB または GPIO
大きさ	85.0 mm × 56.0 mm (3.35 in × 2.20 in) (ただし、コネクタ部の突起は含まない)。
OS	Debian, Fedora, Arch Linux

表 5.1 に URG-04LX-UG01 の仕様表を示す。表 5.2 に Raspberry Pi の仕様表を示す。分散センサは、知能化空間として定めた空間に固定して設置する。そのため、分散センサの制御機器をノート PC でおこなうと、狭い場所など空間の状況次第ではノート PC を設置することが困難な場合がある。地図生成のために設置したノート PC が、設置場所の形状を変化させてしまうことも考えられる。様々な形状の空間で有効に動作する知能化空間の構築をおこなうために、ノート PC を介して制御をおこなっていた分散センサを、ワンボードマイコンによる制御へ変更することで小型化した。ディスプレイやキーボードなどを持たない処理機能のみのコンパクトなコンピュータを分散センサに導入したことで、レーザ式

測域センサを多様な環境に設置可能にした。また、無線通信によりネットワーク化して、環境にセンサを配置した際の設置場所の形状変化も軽減した。付録 C には、本研究で Raspberry Pi を用いてシステム開発したいくつかの小型化したデバイスを紹介する。

### 5.1.2 移動ロボット

本研究の実験では、移動ロボットとして二輪駆動の車輪型移動ロボットを使用する。図 4.2 に示す移動ロボットは、知能機動型ロボットシステムを有するリバスト社製小型人工知能ロボット、Pioneer 3-DX である[55]。Pioneer 3-DX の仕様を表 5.3 にまとめる。Pioneer 3-DX には制御用ノート PC 搭載している。前輪には駆動輪 2 輪、後輪には固定輪 1 輪を有する。移動機構の詳細と制御方式は付録 B に記載する。Pioneer 3-DX の角度に関するエンコーダは前方を  $0^\circ$  とし、左右に  $\pm 180^\circ$  を採っている。座標系に関する詳細は図 4.1 に示す。また、通信方法はユーザ I/O ユニットのポートで制御され、ARIA を通じてアクセス可能。ARCOS ソフトウェアが入出力ユニットをポートでクライアント PC に転送し、制御コマンドが返送される。RedHat 環境または任意の C/C++ 実装 Win32 環境のカスタムプログラムとクライアント PC からロボットの起動が可能。また、移動ロボットは地図を生成するために、レーザ式測域センサを 1 台搭載している。移動ロボットに搭載したレーザ式測域センサ UTM-30LX を図 5.3 に示す。



図 5.2 Pioneer 3-DX

表 5.3 Pioneer 3-DX の仕様表

サイズ	45x44x24.5cm
最低地上高	6.5cm
重量 (標準バッテリー含)	9kg
公称電圧	12v
駆動方式	2WD・デフ・バランス用リアキャスター
車輪分類	フォームフィルドナイロンおよびハードキャスター
車輪直径	20cm
車輪幅	5cm
後端旋回半径	32cm
最高速度	1.2m/s
通過できる段差 (最大)	2.5cm
通過できる窪み (最大)	5cm
勾配登坂能力	25%
走行地形条件	車いすと同等
インターフェース	6 USB 2.0 (3 external) 4 serial
マイコン追加シリアルポート	3



図 5.3 UTM-30LX

表 5.4 UTM-30LX の仕様表

電源電圧	DC 12V ±10%
消費電流	パワーON 時 max 1A 定常時 0.7A 以下
検出距離及び検出体	検出保障値 0.1~30m (白ケント紙) *2 : 最大検出距離 60m(出力限界値) 最小検出物 130mm (10m) : 距離により変動する
測距精度	0.1~10m : ±30mm , 10~30m : ±50mm (白ケント紙) *2 周囲照度 3000lx 以下 白ケント紙 0.1~10m : ±30mm*1 周囲照度 100000lx 以下 白ケント紙 0.1~10m : ±50mm*1
測距分解能及び繰返し精度	1mm 単位 0.1~10m : $\sigma < 10\text{mm}$ , 10~30m : $\sigma < 30\text{mm}$ (白ケント紙) *2 周囲照度 3000lx 以下 : $\sigma < 10\text{mm}$ (白ケント紙 10m まで) 周囲照度 100000lx 以下 : $\sigma < 30\text{mm}$ (白ケント紙 10m まで)
走査角度	270 度
角度分解能	約 0.25 度 (360° / 1440 分割)
走査時間	25ms (モータ回転数 2400rpm)
インターフェース	USB Ver2.0 FS モード (12Mbps)
質量	約 210g
外形寸法	60mm×60mm×85mm(W×D×H)

表 5.4 には、レーザ式測域センサ UTM-30LX の仕様表を示す。センサへの電源は、移動ロボットに搭載したバッテリーから供給される。本研究では移動ロボットに搭載する UTM30-LX と分散配置測域センサの URG04-LX で使い分けているが、検出距離や走査角度、角度分解能の違いのみで基本的な動作は同様である。

### 5.1.3 実験環境

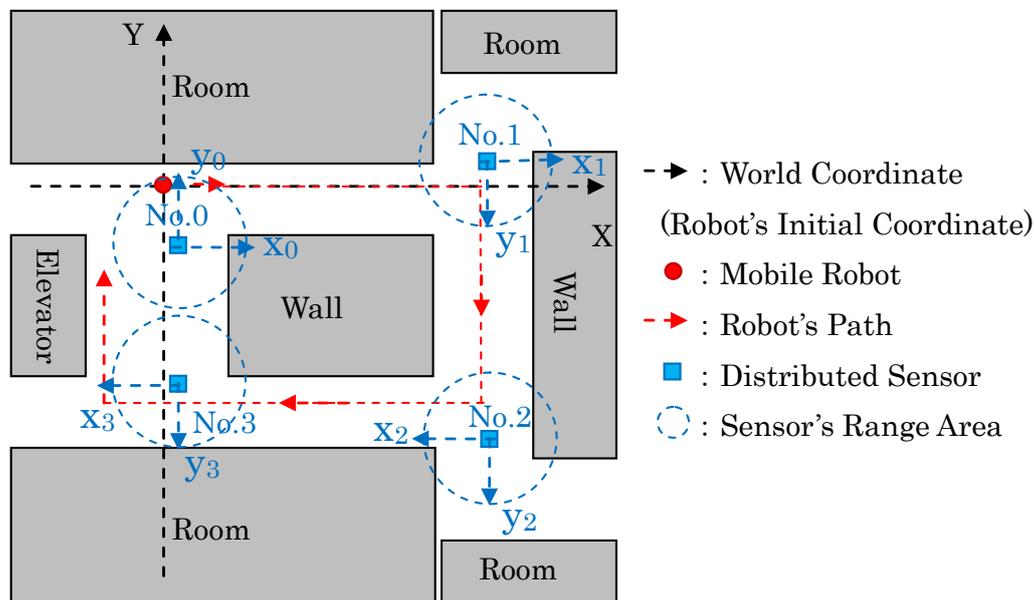


図 5.4 実験環境図

図 5.4 に本研究でおこなった実験の実験環境図を示す。図中の青色の四角印は分散センサの設置位置を示す。分散センサは No.0 から No.3 までの計 4 台のレーザ測域センサを設置した。各センサの座標系は青色の  $x_i$ - $y_i$  ( $i=0,1,2,3$ ) で表わされ、 $x_i$  軸方向がセンサの観測領域の中心方向である。分散センサの相互の観測領域の重なりは考慮せず、知能化空間の構築にあたって、人間や物の位置情報などの取得のために意味があると思われる、通路の交差点付近に、適当に配置している。図中に示した X-Y 座標軸は大域地図の座標系である。赤丸は移動ロボットの走行開始地点であり、赤色の破線は、移動ロボットの走行経路である。経路の全長は 1 周でおよそ 40m である。今回の実験では、移動ロボットは全分散センサの観測領域をそれぞれ一度通過し、走行開始地点へ戻る経路とした。また、本実験で用いる地図は、高さを持たない二次元地図である。分散センサと移動ロボットに設置するセンサは、床面からの高さをそろえ、同じ高さの地図情報を比較することが望ましい。しかし、今回の実験環境の大部分は壁のみにより構成されており、センサの設置高さの違いは構成される占有格子地図にほとんど影響しない。そのため、本実験では設置高さは考慮せず、分散センサを床面に設置した。

移動ロボットは自己位置推定のためにパーティクルフィルタを用いる。今回の実験では、パーティクルの数を 30 個とした。また、構築する格子地図の格子サイズは、移動ロボットの SLAM と分散センサ両方ともに  $100 \times 100 [\text{mm}^2]$  とした。分散センサは  $5000 [\text{mm}]$  離れた距離まで物体を測定するようにしている。分散センサを Raspberry Pi に接続したものを 1

セットとして環境に配置し、距離データを取得するたびに格子地図を生成、更新してファイルに保存する。移動ロボットは制御用ノート PC が搭載され、センサ情報の取得をおこなう。分散センサおよび移動ロボットは、全て無線 LAN によりネットワーク接続されている。分散センサでは、移動ロボットの地図情報報が記載されたファイルを読み込み、移分散センサの位置推定および地図更新をおこなう。移動ロボットでは、分散センサの地図情報が記載されたファイルを読み込み、移動ロボットの位置推定および地図更新をおこなう。本論文で提案する知能化空間における分散センサと移動ロボットの地図情報比較の手法が、分散センサの位置推定と移動ロボットの SLAM における推定精度の向上に有効であることを示すための評価実験をおこなった。その実験の評価項目 1 から評価項目 4 を以下に示す。

### **評価項目 1：分散センサの位置推定評価**

提案手法により世界座標系に対する分散センサの位置・姿勢が適切に推定されているかを評価する。ICP マッチングと座標系間の角度差により推定した各分散センサの位置を任意に与えた初期値およびコンベックスにより計測した計測値と比較して、推定された位置座標を評価する。ICP マッチングは、文献[56]と同様の方法により実装した。一般的に ICP マッチングは推定結果が初期位置に依存する手法であり、与えた初期位置によっては、正しい対応関係が得られない場合もある。本実験では、分散センサの初期位置として、分散センサの初期位置を、位置の実測値を中心に分散センサの最大観測距離以内(半径 5000mm)に設定することで、ある程度真値に近い初期値から対応関係を求めている。

### **評価項目 2：分散センサと移動ロボットの地図のマッチングの様子**

移動ロボットが分散センサの観測領域に進入した際、分散センサの局所地図と大域地図のマッチングの様子から、提案した比較の手法が適切であることを確認する。ICP における分散センサと移動ロボットの地図の比較により、その対応点数を評価する。

### **評価項目 3：移動ロボットの SLAM による地図生成精度の向上評価**

移動ロボットの SLAM における位置と地図の推定に分散センサの地図情報を反映させ、推定値の修正をおこなうプロセスが適切に機能しているのかを確認する。移動ロボットが SLAM により独立して生成した地図と提案手法により生成した地図を比較して提案手法が SLAM による地図生成へ及ぼす影響を確認することで、提案手法が適切に動作しているのかを評価する。

### **評価項目 4：分散センサの地図と移動ロボットのスキャンデータのマッチングの様子**

移動ロボットが分散センサの観測領域に進入した際、分散センサの局所地図と移動ロボットのスキャンデータの対応点の様子から、推定値の修正をおこなうプロセスが適切に機

能しているのかを確認する。

## 5.2 分散センサの地図生成における実験結果

図 5.5 から図 5.8 に図 5.4 に示した各分散センサの設置位置とその格子地図を示す。図中 (a) には設置位置の様子を、(b) にはその格子地図を示す。(a) の白線矢印は設置した分散センサの向きを示し、黄線矢印は移動ロボットが移動した経路を示している。(b) の赤色矢印は、設置した分散センサの向きを示している。各分散センサの地図は、移動ロボットが走行を開始するより前にある程度に生成した。これは、分散センサの地図が無い状態から分散センサが地図を生成するにはある程度時間が必要であり、本研究の提案手法における分散センサと移動ロボットの地図比較のためには、移動ロボットが走行を開始したときに分散センサの地図がある程度生成されている必要があるからである。

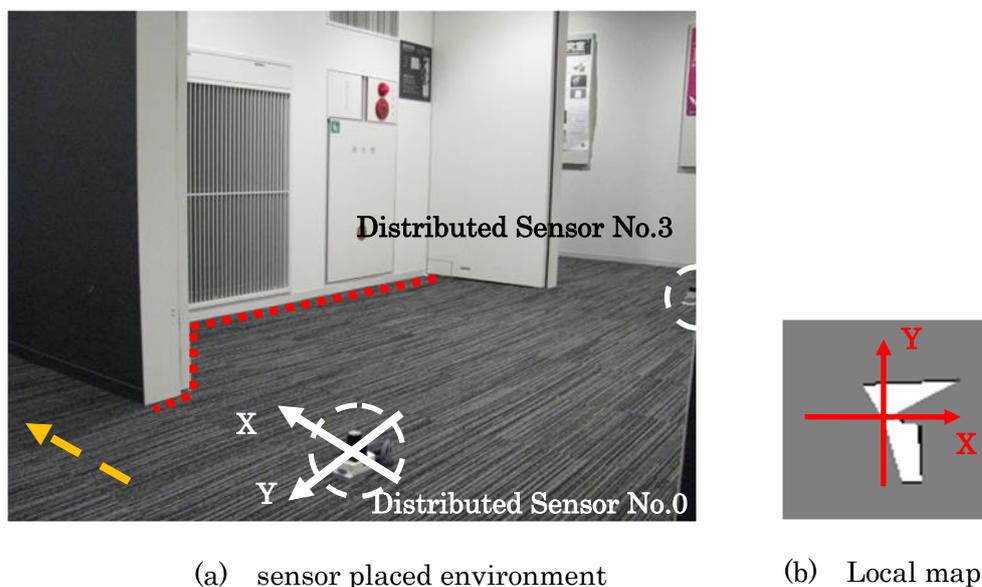
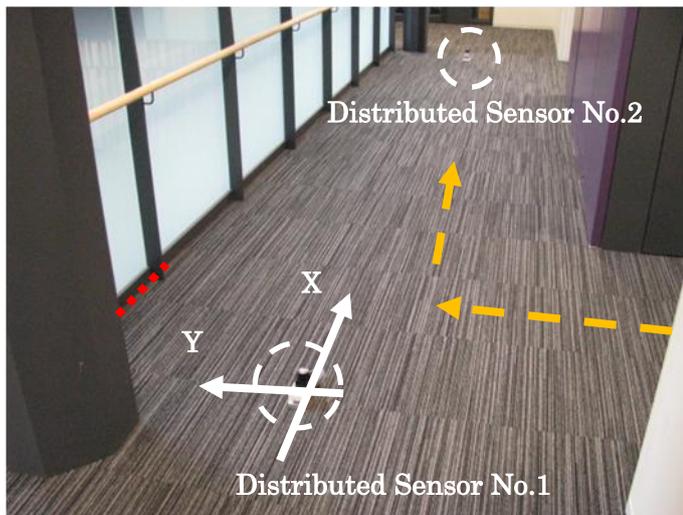
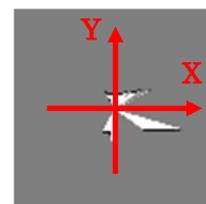


図 5.5 分散センサ No.0 による地図生成結果

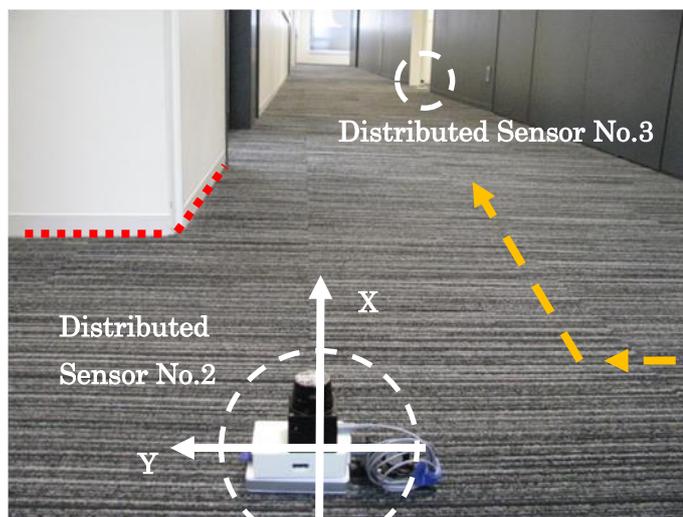


(a) sensor placed environment

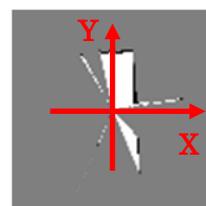


(b) Local map

図 5.6 分散センサ No.1 による地図生成結果



(a) sensor placed environment



(b) Local map

図 5.7 分散センサ No.2 による地図生成結果

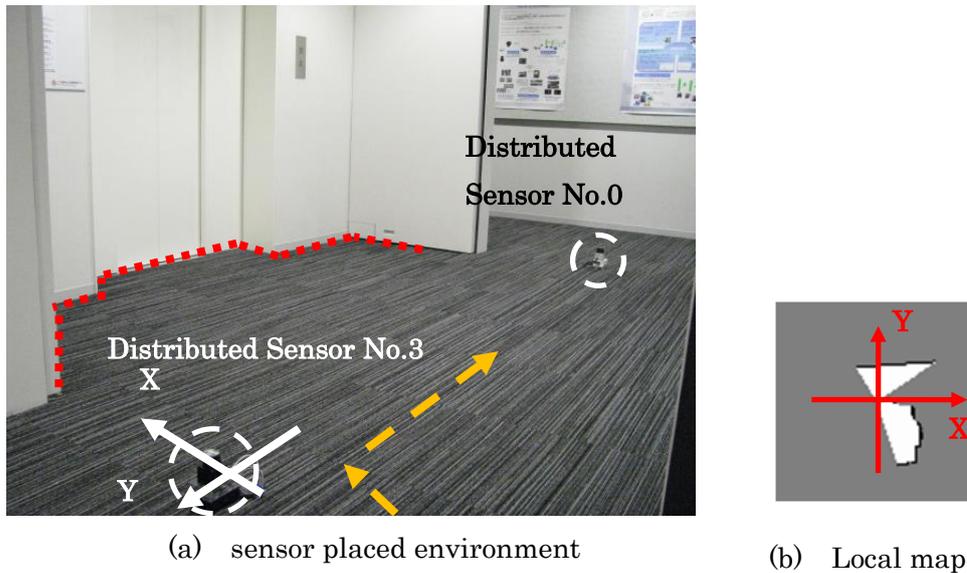


図 5.8 分散センサ No.3 による地図生成結果

### 5.3 移動ロボットの FastSLAM の実験結果

図 5.9 は図 5.4 において移動ロボットが独立して生成した地図を示している。図中(a)は FastSLAM により生成した地図を示し、(b)はホイールオドメトリにより生成した地図を示している。図 5.10 は FastSLAM により大域地図が更新されていく様子を示している。図 5.11 は大域地図が更新されたときの局所地図の様子を示している。赤色の点は、距離データを示している。図 5.12 は局所地図が更新されたときの距離データの様子を示している。各図中の(i)から(iv)は時間推移における各様子を示している。

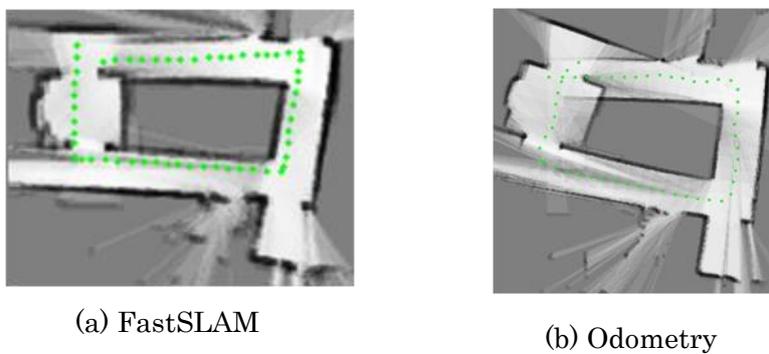


図 5.9 大域地図

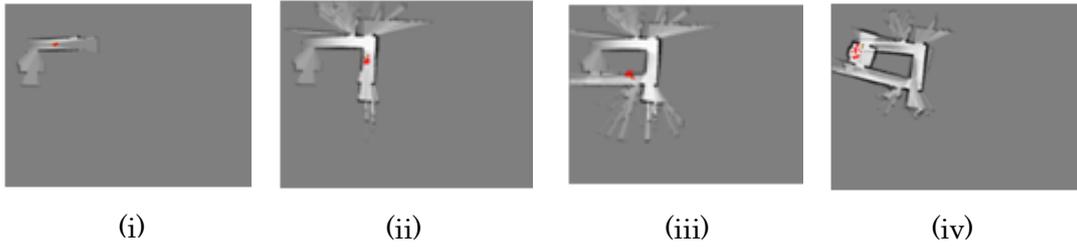


図 5.10 FastSLAM による大域地図生成の推移

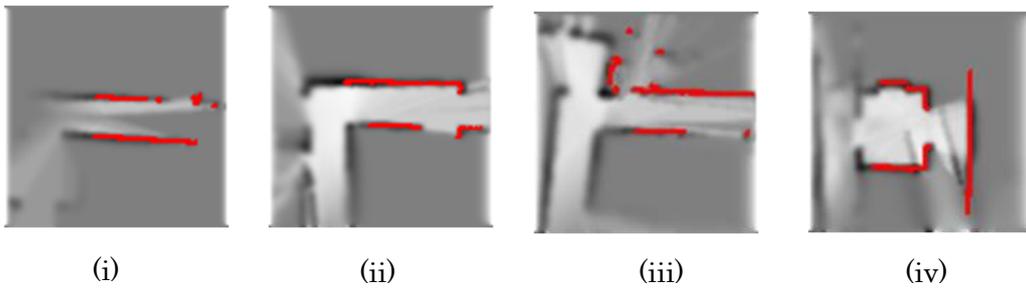


図 5.11 FastSLAM による局所地図

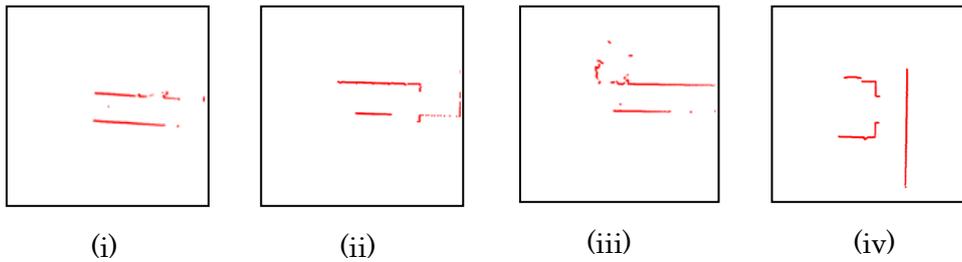


図 5.12 FastSLAM による距離データ

#### 5.4 ICP マッチングを用いた位置推定と地図生成の結果

表 5.5 から表 5.8 には、ICP マッチングによる各分散センサの位置推定結果を示す。Initial position は分散センサに与えた初期座標値、Measured position はコンベックスにより計測した分散センサの座標、Estimated position は本研究の提案手法による位置推定結果を示している。図 5.13 には、各分散センサの初期位置からの位置更新の様子を示している。青色

の点を初期値、緑色の点をコンベックスにより実測した計測値、赤色の点を推定値として表記した。

表 5.5 ICP による分散センサ No.0 の位置推定結果

No.0	X [mm]	Y [mm]	$\theta$ [deg]
Initial position	0.00	0.00	0.00
Measured position	0.00	-500	0.00
Estimated position	-192.80	-393.30	0.98

表 5.6 ICP による分散センサ No.1 の位置推定結果

No.1	X [mm]	Y [mm]	$\theta$ [deg]
Initial position	8000.00	0.00	-90.00
Measured position	8105.00	500	-90.00
Estimated position	8196.53	479.56	-91.01

表 5.7 ICP による分散センサ No.2 の位置推定結果

No.2	X [mm]	Y [mm]	$\theta$ [deg]
Initial position	8000.00	-6000.00	180.00
Measured position	8105.00	-6430.00	180.00
Estimated position	8083.61	-6674.12	179.24

表 5.8 ICP による分散センサ No.3 の位置推定結果

No.3	X [mm]	Y [mm]	$\theta$ [deg]
Initial position	0.00	-6000.00	180.00
Measured position	0.00	-5930.00	180.00
Estimated position	-279.73	-5103.73	178.53

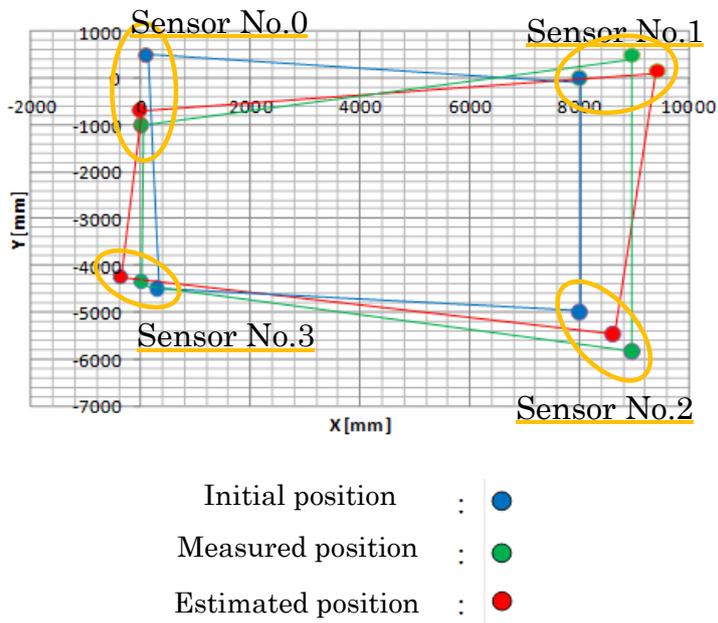


図 5.13 分散センサの位置推定結果

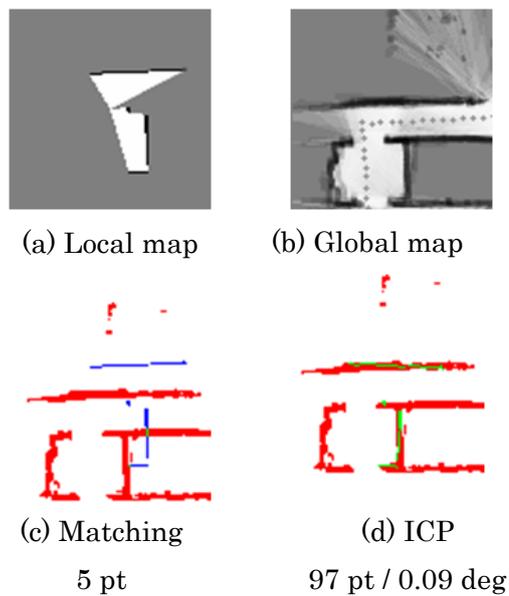


図 5.14 分散センサ No.0 における ICP 実装結果

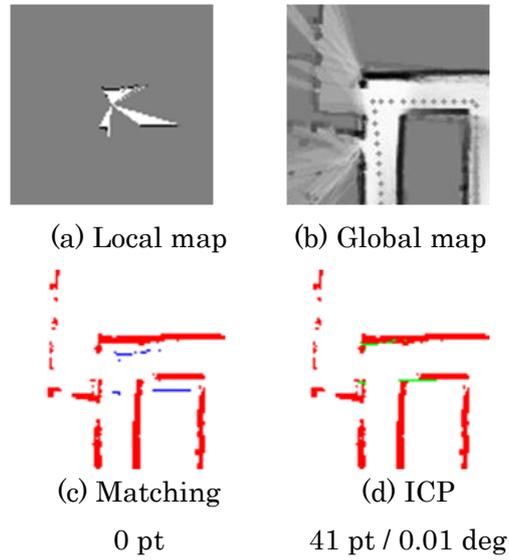


図 5.15 分散センサ No.1 における ICP 実装結果

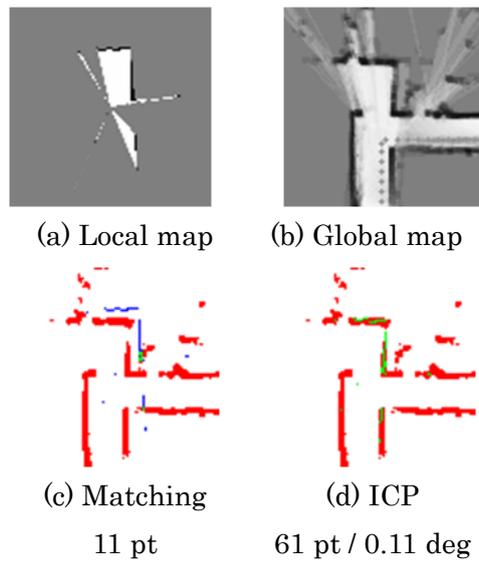


図 5.16 分散センサ No.2 における ICP 実装結果

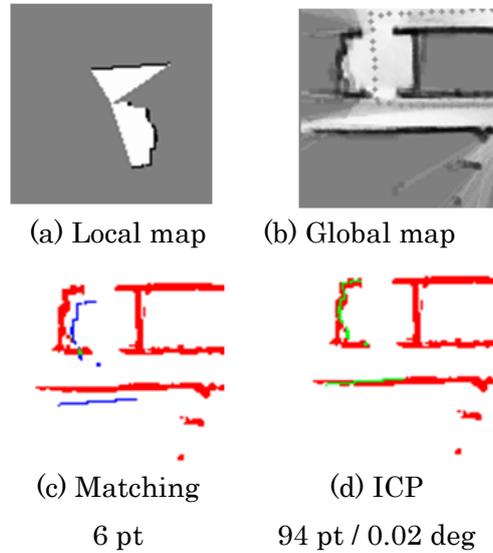


図 5.17 分散センサ No.3 における ICP 実装結果

図 5.14 から図 5.17 は ICP マッチングにより分散センサと移動ロボットの地図を比較した様子である。各図の(a)は分散センサの格子地図を示している。(b)は移動ロボットの全域地図において、地図比較のために分散センサの位置から分散センサの地図と同サイズに切り出した地図である。(c)は分散センサの地図と切り出した移動ロボットの地図において、単純に座標系を合わせて比較した結果である。青色の点は分散センサの地図中で占有格子確率が 0.5 よりも大きい格子の座標を示しており、赤色の点は切り出した移動ロボットの地図中で占有格子確率が 0.5 よりも大きい格子の座標を示している。緑色の点は、双方が座標上対応している点である。対応点数を、(c)の下に pt として示している。(d)は ICP マッチングにより位置推定された分散センサの地図と切り出した移動ロボットの地図との対応の様子を示しており、表記方法は(c)と同様である。

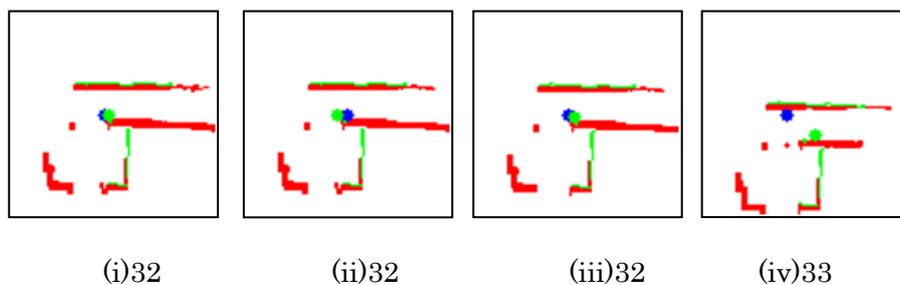


図 5.18 分散センサ No.0 における ICP 実装推移

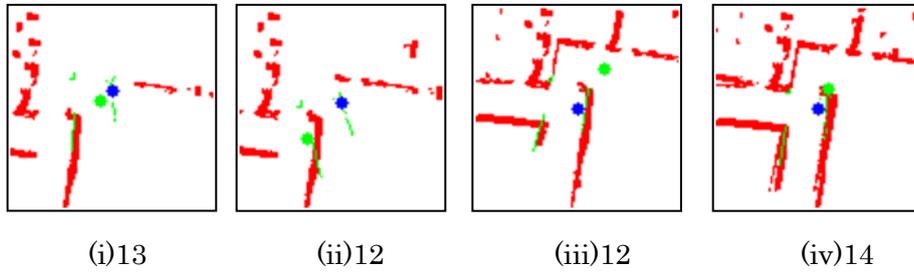


図 5. 19 分散センサ No.1 における ICP 実装推移

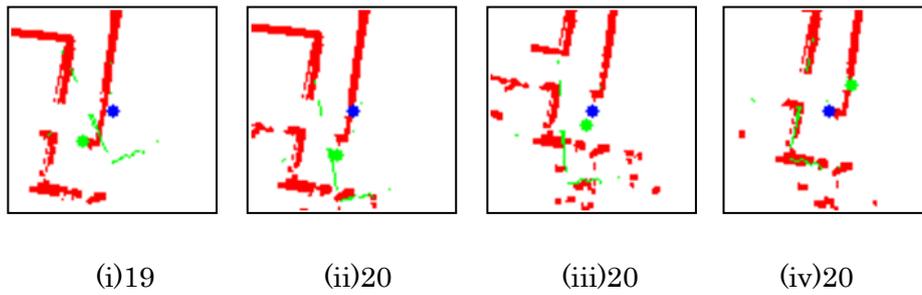


図 5. 20 分散センサ No.2 における ICP 実装推移

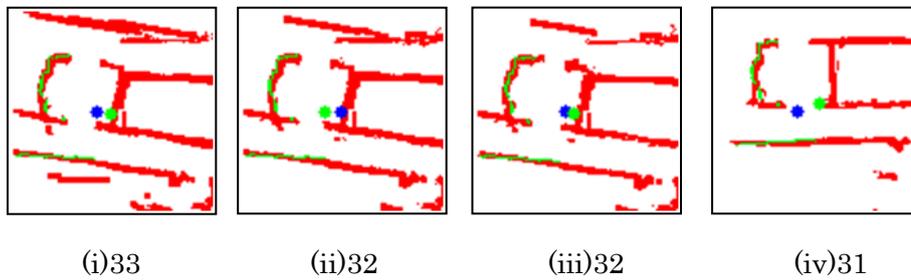


図 5. 21 分散センサ No.3 における ICP 実装推移

図 5.18 から図 5.21 は、SLAM において更新されていく移動ロボットの大域地図と各分散センサの地図との比較を繰り返しおこなった際の対応の様子である。表記方法は図 5.14 から図 5.17 の(d)と同様である。(i)から(iv)は繰り返しICPをおこなった推移を示しており、その隣の数是对応点数を示している。

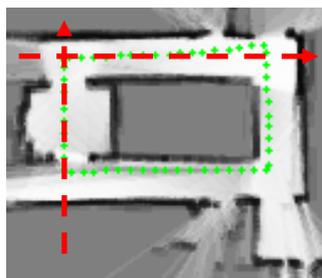


図 5.22 ICP を用いた SLAM の大域地図

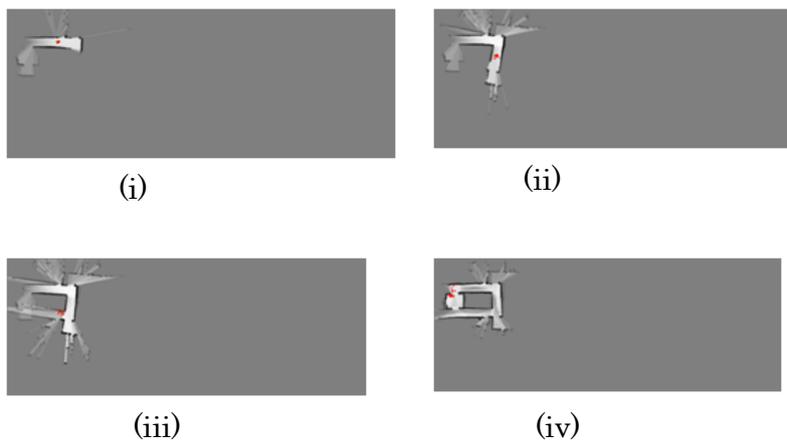


図 5.23 ICP を用いた SLAM の大域地図生成の推移

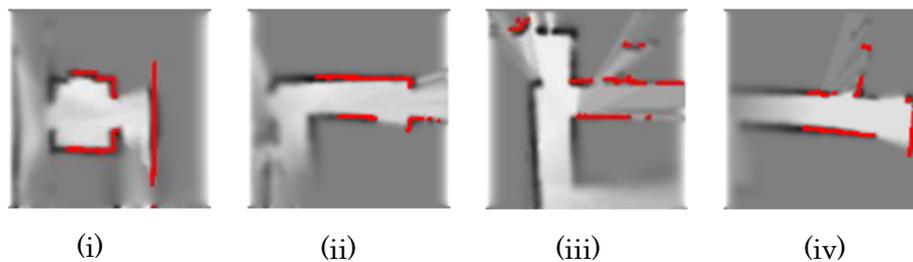


図 5.24 ICP を用いた SLAM の局所地図

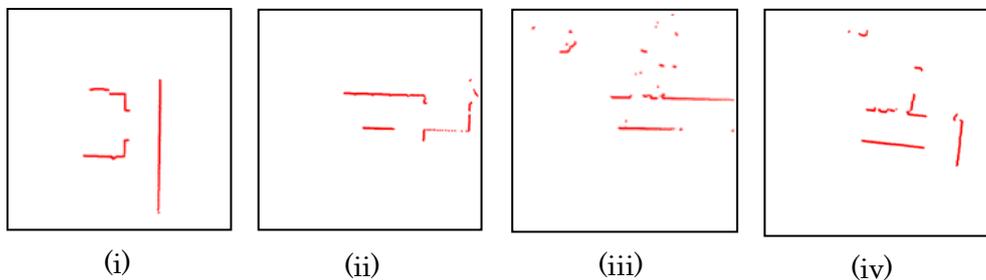


図 5.25 ICP を用いた SLAM の距離データ

図 5.22 は図 5.4 において、移動ロボットが本研究の提案手法に基づいて生成した大域地図を示している。図 5.23 は提案手法により大域地図が更新されていく様子を示している。図 5.24 は大域地図が更新されたときの局所地図の様子を示している。図 5.25 は局所地図が更新されたときの距離データの様子を示している。

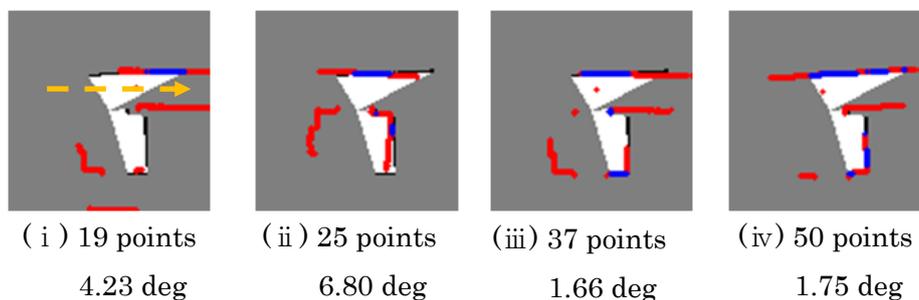


図 5.26 SLAM における距離データと分散センサ No.0 との対応推移

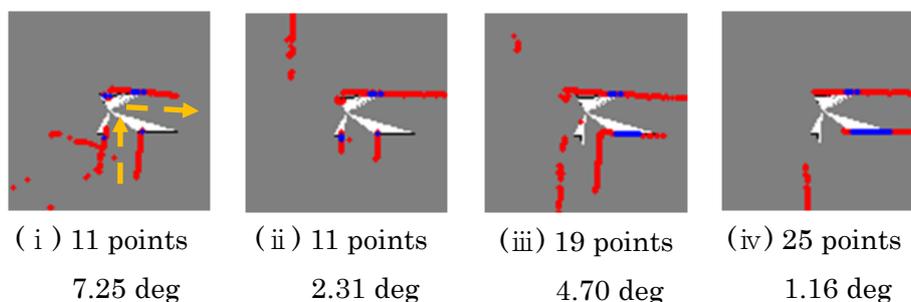


図 5.27 SLAM における距離データと分散センサ No.1 との対応推移

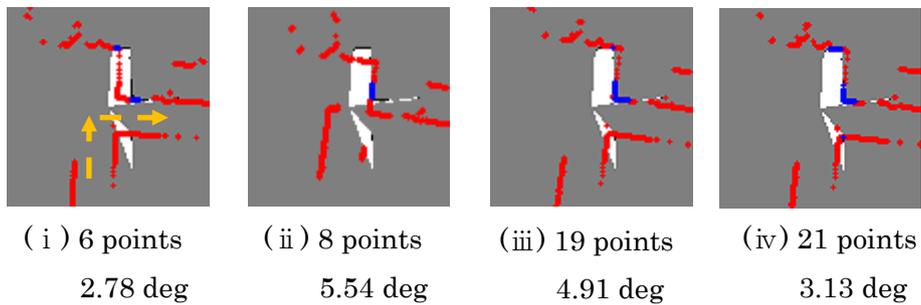


図 5.28 SLAM における距離データと分散センサ No.2 との対応推移

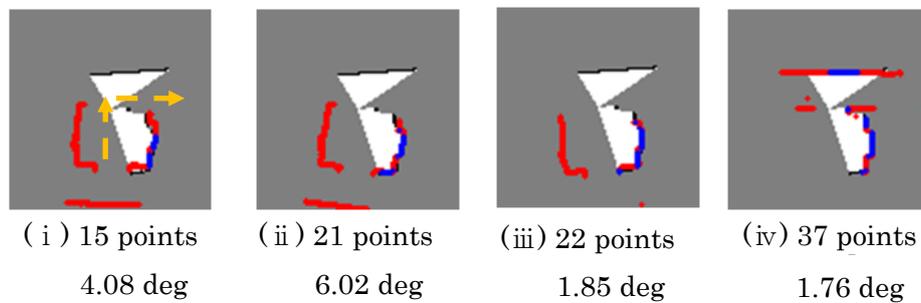


図 5.29 SLAM における距離データと分散センサ No.3 との対応推移

ICP により位置推定された分散センサの位置座標を基に座標変換した分散センサの局所地図と、SLAM による最高尤度のパーティクルにおける移動ロボットのスキャンデータとの対応点の様子を図 5.26 から図 5.29 へ示す。各図の赤色の点は距離データを示し、青色の点は局所地図と対応している格子を示している。各図の(i)から(iv)は比較を繰り返した推移を示しており、各分散センサの地図と移動ロボットの距離データの対応点数と座標系間の角度差を記している。黄色の破線矢印は、移動ロボットが通過した方向を示している。

以下には本提案手法において、推定が適切におこなわれない場合の結果を紹介する。図 5.30 から図 5.33 は、SLAM における最高尤度のパーティクルの距離データと、分散センサの地図の対応点の様子である。表記方法は、図 5.26 から図 5.29 と同じである。図 5.34 は本節の手法により移動ロボットが生成した地図を示す。実装環境は図 5.4 に示す。

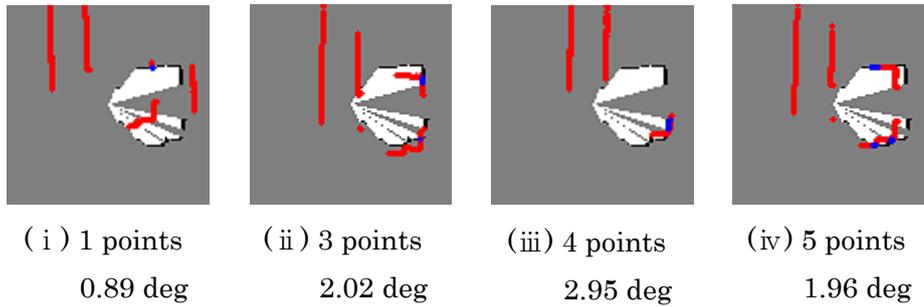


図 5.30 SLAM における距離データと分散センサ No.0 との誤対応推移

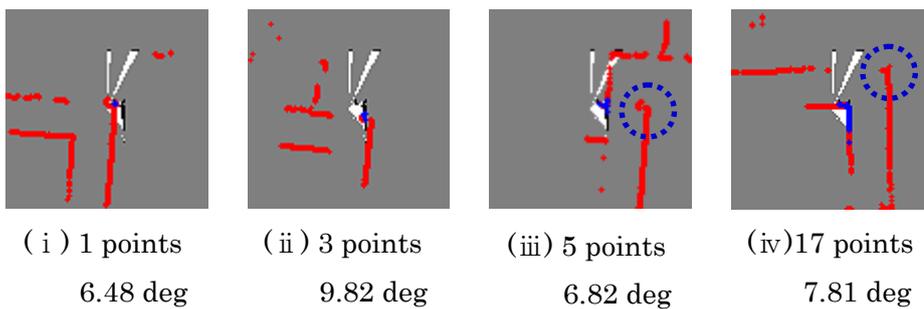


図 5.31 SLAM における距離データと分散センサ No.1 との誤対応推移

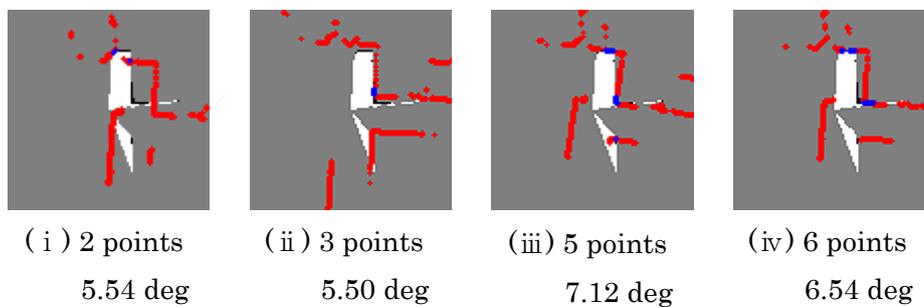


図 5.32 SLAM における距離データと分散センサ No.2 との誤対応推移

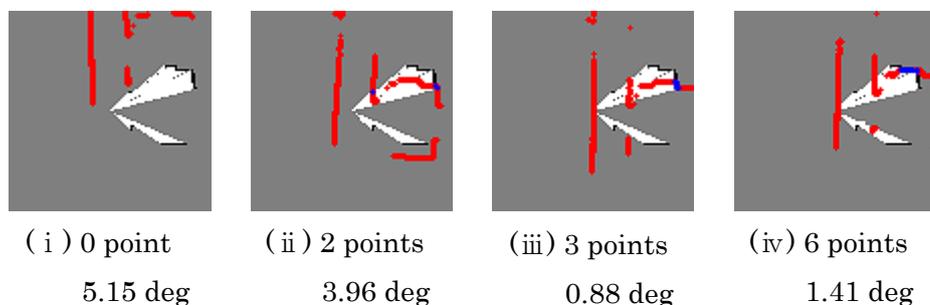


図 5.33 SLAM における距離データと分散センサ No.3 との誤対応推移

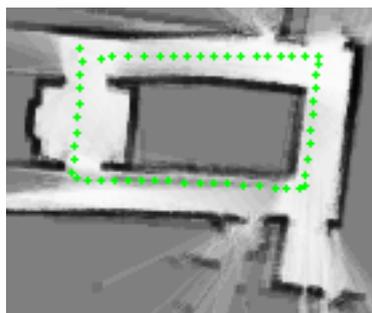


図 5.34 ICP を用いた SLAM の失敗例

## 5.5 座標系間の角度差による手法

表 5.9 から表 5.12 には、座標系間の角度差による各分散センサの位置推定結果を示す。Initial position は分散センサに与えた初期座標値、Measured position はコンベックスにより計測した分散センサの座標、Estimated position は本研究の提案手法による位置推定結果を示している。図 5.35 には、各分散センサの初期位置からの位置更新の様子を示している。青色の点を初期値、緑色の点をコンベックスにより実測した計測値、赤色の点を推定値として表記した。

表 5.9 角度差による分散センサ No.0 の位置推定結果

No.0	X [mm]	Y [mm]	$\theta$ [deg]
Initial position	300.00	-800.00	-90.00
Measured position	0.00	-500.00	-90.00
Estimated position	143.30	-475.71	-89.02

表 5.10 角度差による分散センサ No.1 の位置推定結果

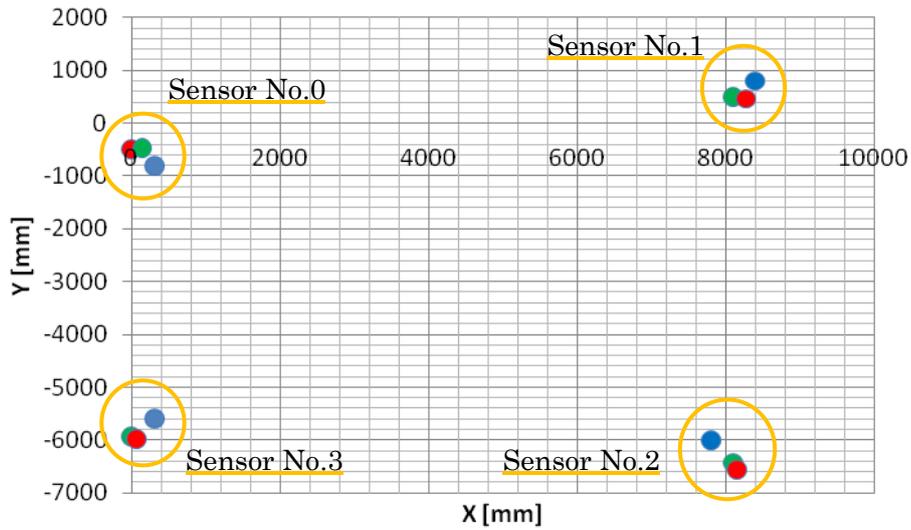
No.1	X [mm]	Y [mm]	$\theta$ [deg]
Initial position	8400.00	800.00	0.00
Measured position	8105.00	500.00	0.00
Estimated position	8270.74	462.59	-1.01

表 5.11 角度差による分散センサ No.2 の位置推定結果

No.2	X [mm]	Y [mm]	$\theta$ [deg]
Initial position	7800.00	-6000.00	180.00
Measured position	8105.00	-6430.00	180.00
Estimated position	8155.44	6556.56	179.24

表 5.12 角度差による分散センサ No.3 の位置推定結果

No.3	X [mm]	Y [mm]	$\theta$ [deg]
Initial position	300.00	-5600.00	90.00
Measured position	0.00	-5930.00	90.00
Estimated position	61.98	-5980.18	88.53



Initial position	:	●
Measured position	:	●
Estimated position	:	●

図 5.35 分散センサの位置推定結果

図 5.36 は傾き比較の手法により移動ロボットが生成した地図を示す。実装環境は図 5.4 に示す。

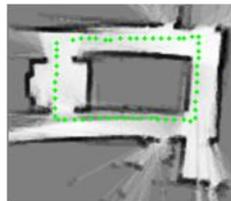


図 5.36 傾き比較による移動ロボットの地図

図 5.37 から図 5.40 には、本節の手法による分散センサの地図と移動ロボットのスキャンデータとの対応の様子を示す。分散センサの格子地図において、座標変換した移動ロボットのスキャンデータを赤色の点で表わしている。また、青色の点は分散センサの格子と移動ロボットのスキャンデータが重なった対応点を表している。

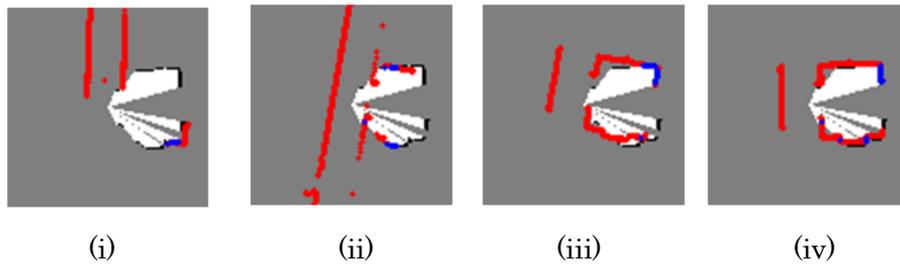


図 5.37 分散センサ No.0 における対応の様子

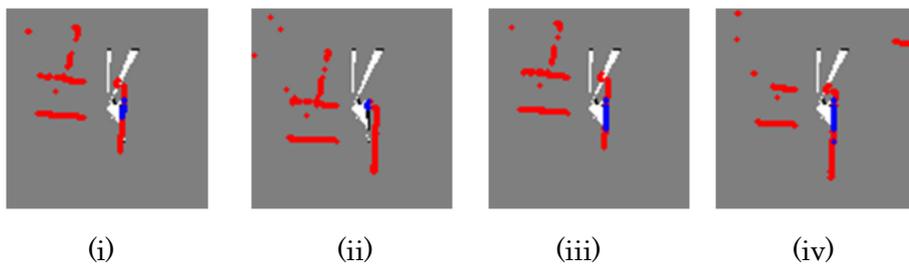


図 5.38 分散センサ No.1 における対応の様子

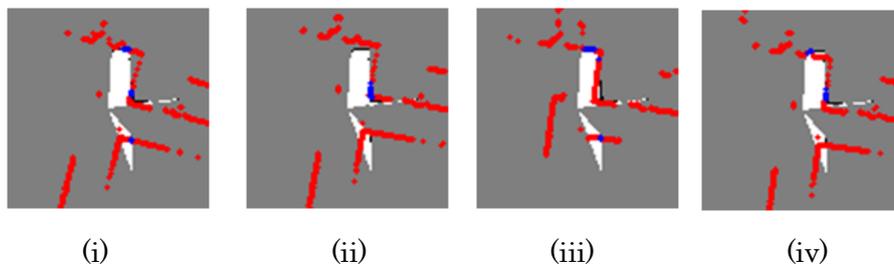


図 5.39 分散センサ No.2 における対応の様子

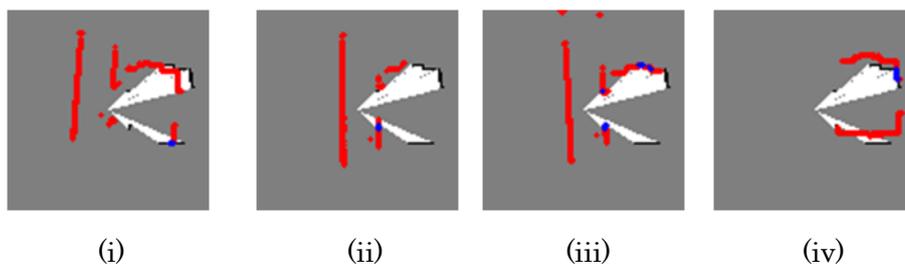


図 5.40 分散センサ No.3 における対応の様子

## 6 提案手法についての考察

### 6.1 分散センサの地図生成と移動ロボットの FastSLAM の考察

分散センサの地図生成の考察について、移動ロボットが走行を開始する前の分散センサによる地図生成は、分散センサの観測領域内にも人などの移動物体が存在する状態でおこなった。分散センサの局所地図生成結果である図 5.5 から図 5.8 を見ると、移動物体が存在する環境でも占有格子確率により移動物体の影響を軽減した地図生成ができています。

移動ロボットの FastSLAM の考察については、図 5.9 (a)には FastSLAM による地図生成の様子を確認したが、図 5.4 の見取り図と比較すると地図が歪んでいるように見える。図 5.10 に示す大域地図生成の推移の結果からも、地図生成するにつれて、地図が歪んできていることが確認できる。図 5.11 の局所地図とその距離データを示す図 5.12 を見ると、双方の対応は比較的良く対応していることがわかる。FastSLAM において、距離データと局所地図の対応関係が得られているにもかかわらず、大域地図が歪んでしまうということの原因の 1 つには、ホイールオドメトリの推定誤差を SLAM が修正しきれていないことが考えられる。しかし、図 5.9 (b)に示すホイールオドメトリのみを用いて生成した地図と比較すると、地図生成におけるホイールオドメトリの位置推定誤差がある程度修正されていることがわかった。本節の FastSLAM では、移動ロボットが過去に通過して生成した地図中の経路が、周回して同一の経路の地図を生成したときに重ならず歪んだ地図が生成されることが多く確認された。

### 6.2 提案手法による実験結果の考察

#### 評価項目 1

評価項目 1 では、分散センサの位置推定評価をおこなった。本手法による分散センサの位置・姿勢推定を評価するために、ICP マッチングと座標系間の角度差により推定した各分散センサの位置を初期値と計測値を示して、推定された位置座標が、初期値から計測値近くへと推定されていく様子を確認する。表 5.5 から表 5.8 には ICP マッチング、表 5.9 から表 5.12 には座標系間の角度差を用いた各分散センサの世界座標系に対する設置位置と設置角（姿勢）の推定結果を示す。手動では設置角度を厳密に測定することは困難であるため、計測値は求めていない。推定結果により各センサの設置位置と設置角度が修正されていることが確認できる。図 5.13 には ICP マッチング、図 5.35 には座標系間の角度差による各分散センサの初期位置からの位置更新の様子を示している。青色の点を初期値、緑色の点をコンベックスにより実測した計測値、赤色の点を推定値として表記した。それぞ

れの手法において、各分散センサ位置として適当に与えた初期位置関係から、提案システムにより実際の推定値に接近した位置関係に更新されていることが確認できる。

## 評価項目 2

評価項目 2 では、ICP マッチングによる分散センサの局所地図と移動ロボットの全域地図との対応関係から、提案した比較手法が適切であるかを確認した。座標変換をしてそのまま重ね合わせた場合と ICP マッチングによって分散センサと移動ロボットの地図の比較をおこなった場合の対応点数を比べ、ICP マッチングが適切におこなわれているか確認する。図 5.14 から図 5.17 には図 5.4 に示した実験環境において、ICP マッチングによる地図の対応結果を示している。各図の(a) には分散センサにより得られた実際の局所地図を示す。(b) は、全域地図から移動ロボットの推定位置周辺のみを、局所地図との比較のために切り出したものである。全域地図および局所地図における黒色の点は、空間内の構造物を示している。図の(c) は分散センサの初期位置と移動ロボットの推定位置を基準として(a) と(b) を重ね合わせ対応点を取得した様子である。青色の点が分散センサの地図であり、赤色の点が移動ロボットの地図である。また、緑色の点は双方の格子が対応している箇所を占めている。(d) は、ICP マッチングによって得られた対応結果であり(c)と同様の表記方法である。各分散センサとも単純に双方の地図を重ね合わせただけの(c)では十分な対応は得られず、(d) による ICP マッチングが明らかに多くの対応点を得られている様子が確認できた。また、図 5.18 から図 5.21 は、SLAM により更新されていく全域地図と分散センサの局所地図に対して ICP をおこなった結果を示している。(i)から(iv)のどの結果も一定の対応点数が得られていることが確認できる。以上のことから、提案したシステムは適切に動作しているといえる。

## 評価項目 3

評価項目 3 では、移動ロボットの SLAM による地図生成精度の評価をおこなった。SLAM のパーティクルにおける距離データと分散センサの地図の対応点数を尤度に加え、推定値の修正をおこなうプロセスが適切に機能しているのかを確認した。移動ロボットが SLAM により独立して生成した地図よりも提案手法による SLAM の地図が精度よく生成されていることを確認する。図 5.22 に示す ICP により生成された地図と図 5.9 (a)に示す FastSLAM により生成された地図を確認すると、FastSLAM が生成した地図に見られた地図の歪みが提案手法により修正されていることが確認できる。また、図 5.23 に示す全域地図の推移については、(ii)、(iii)においては推定誤差により地図が歪んでいるが、(iv)においては図 5.24 と図 5.25 における FastSLAM の様子は変わらないが、提案手法により歪みが修正されていることが確認できる。本研究の提案手法は、FastSLAM が生成した歪みを含む地図を分散センサの地図との位置関係を求めることで FastSLAM の地図の歪みを修正したと考えられる。分散センサとの地図情報の比較をおこなった場合は、移動ロボットが生成した地図に

において、設定した世界座標系の座標軸と比較して歪みが少なくなっており、各分散センサの位置姿勢が適切に推定された結果と考えられる。また、図 5.36 の地図は、座標系間の角度差による大域地図であり、移動ロボットが独立しておこなう FastSLAM による地図と比べるとループクローズは閉じられる傾向があるが、歪みのある地図が生成されることが確認された。これは、後に示す図 5.37 から図 5.40 をみると、分散センサの地図と移動ロボットのスキャンデータとの対応点が地図の比較を繰り返すと得られていく場合があることが確認できる。しかし、未だにわずかな角度の違いにより対応点が少ししか得られていない状況である。そのため、SLAM におけるパーティクルフィルタの尤度に及ぼすに影響が少ないことがわかった。結果として、移動ロボットが独立しておこなう FastSLAM と同程度の地図や、FastSLAM よりも歪んだ地図が生成されることがあると考えられる。

#### 評価項目 4

評価項目 4 では、分散センサの局所地図と移動ロボットの距離データの対応関係の様子を確認する。分散センサの局所地図と SLAM におけるパーティクルの距離データとの対応点数から、推定値の修正をおこなうプロセスが適切に機能しているのかを確認した。

図 5.26 から図 5.29 に示す ICP による分散センサの地図と移動ロボットのスキャンデータの対応点数が、比較をおこなう度に増加していく様子が確認できる。また、座標系間の角度差も少なくなっていることが確認できる。この対応点数は SLAM の尤度へ加えられるため、移動ロボットの距離データだけではなく、位置推定された分散センサの地図にも対応する大域地図の生成がおこなわれたと考えられる。一方、図 5.37 から図 5.40 は、座標系間の角度差による対応点の推移である。全体的に対応点数が少ししか得ることができず、分散センサの位置推定は初期値からほとんど変化せず適切におこなわれていない場合があるため、対応点を得るためには、座標系間の傾き以外に地図の形状に注目する必要があると考えられる。このことから、地図の形状を探索する ICP マッチングの方が座標系間の角度差を評価する手法に比べて分散センサの位置推定に有効であると考えられる。

また、以下には ICP マッチングによる手法で確認された問題点について述べる。本研究の ICP マッチングにおける問題点は、図 5.30 から図 5.34 に示した。提案手法では、ICP マッチングより分散センサの位置を、分散センサの地図が移動ロボットの地図の形状に近い場所に推定する。そのため、移動ロボットの地図生成精度にも左右されるが、おおよそ安定して対応点が取れる。しかし、移動ロボットが分散センサの地図を用いて SLAM による位置推定をおこなう場合、パーティクルフィルタにおける分散をパーティクルの距離データへ与えるため、どのパーティクルの距離データも分散センサの地図と対応点が適切に取れない場合が起きることが確認できた。評価実験において推定が適切におこなわれない場合の結果を図 5.30 から図 5.33 に示した。分散センサの地図と移動ロボットの距離データとの対応点が地図の比較を繰り返すと徐々に対応点が得られていく様子が確認できる。し

かし、未だに角度の違いにより対応点が少ししか得られない場合や、図 5.31(iii)(iv)の青色破線円内に見られるように、対応点が得られていても実際に対応するべき箇所が異なっている場合が見られた。そのため、SLAM におけるパーティクルフィルタに加算される尤度が適切でなくなる場合があることがわかった。結果として、提案手法においても移動ロボットが独立しておこなう FastSLAM のように歪んだ地図が生成されることがある。図 5.34 の地図は、移動ロボットが独立しておこなう FastSLAM による地図と比べるとループクローズは閉じられる傾向があるが、歪みのある地図が生成されることが確認できる。

上記の問題は、図 5.30 から図 5.33 に示した様に、わずかな座標の違いにより対応点が得られないことが原因の一つであると確認できたため、SLAM においてパーティクルを適切に分散させる手法を導入することや、パーティクルフィルタを用いずに ICP マッチングを導入するなどを検討する必要がある。また、Michael Milford 氏らが提案している RatSLAM[57]での画像マッチング方法も参考にできると考えられる。RatSLAM では、カメラを搭載した移動ロボットがある経路を繰り返し走行して、カメラ画像を空間情報として保存していく。その保存された画像と現在取得した画像をマッチングすることで、自己位置推定をおこなう。しかし、マッチングの際には、光による画像のコントラストや倍率が影響して、適切にマッチングがおこなわれない場合がある。これを緩和するためにマッチングをおこなう画像の解像度を低くすることで、対応する画素の位置に曖昧さをもたせる。曖昧さにより対応点を取りやすくなり、結果として適切なマッチングが可能になる効果があると示している[58][59]。RatSLAM により屋内外様々な環境で、移動ロボットが独立して自己位置推定を適切におこなう様子が述べられている。

## 7 結言

### 7.1 本論文のまとめ

ここでは、本論文における各章のまとめを記す。

#### 1 緒言

1.1 では、IT 産業やロボット工学の分野における技術が、人の生活場面で利用される例を記して、その技術の中で近年盛んに研究されている知能化空間に関する研究背景を記した。1.2 では本論文の構成を記した。

#### 2 知能化空間

2.1 では、知能化空間のコンセプトを記した。2.2 では、現在までに実現されている実際の知能化空間の実装例を記した。また、2.3 では、関連研究として様々なセンサを用いた知能化空間の構築例を紹介した。

#### 3 分散センサの位置推定支援システム

3.1 では、分散センサの位置推定手法の概要を記した。3.2 では、分散センサの地図生成の推定手法を記した。3.3 では、移動ロボットの地図生成と自己位置の推定手法を記した。また、3.4 では、提案手法による分散センサと移動ロボットの地図比較におけるシステム全体の動作概要を記した。

#### 4 分散センサと移動ロボットの地図情報の比較方法

本研究の提案手法における分散センサと移動ロボットの地図比較手法を記した。ここでは、4.1 と 4.2 に示した座標変換による手法、4.3 に示した角度差による手法、4.4 に示した ICP マッチングによる手法の 3 通りの地図比較手法を紹介した。以下にそれぞれの要約を記す。

- 座標変換による手法

分散センサと移動ロボットの地図の座標系を変換して重ね合わせることでより地図比較をおこなう方法。

- 角度差による手法

双方の座標系間の角度差が小さくなるように、地図を重ね合わせて比較する方法。

- ICP マッチングによる手法

双方の地図の形状に注目した ICP マッチングを用いて、地図を比較する方法。

#### 5 地図比較手法の検証実験

5.1 では、本研究の提案手法が有効であるかを確認するための評価実験の環境とその使用機器、実験における評価項目を示した。5.2 では、分散センサの地図生成結果を記した。5.3 では、移動ロボットが SLAM による地図生成の様子を記した。5.4 で

は、ICP マッチングにおける評価項目について、5.5 では角度差における評価項目についての実験結果を記した。4 台の分散センサと 1 台の移動ロボットが独立して生成した地図と提案手法によって生成した地図の比較結果や、推定された位置座標と実際の位置として計測した位置座標をまとめた。

## 6 提案手法についての考察

6.1 では分散センサが独立して生成した地図や、移動ロボットが独立しておこなった SLAM における地図生成結果についての考察を記した。6.2 では、評価項目における実験結果を基に、提案手法が分散センサの位置推定と移動ロボットの SLAM における推定精度の向上について適切であるかを考察した。4 章で紹介した手法の中で、ICP マッチングによる位置推定が最も適切に位置推定できることを示した。

## 7 結言

7.1 では、本論文の各章のまとめを記す。7.2 では、今後の課題と展望を記す。

また、以降に記載する付録の内容は、以下に記す。

### 付録

#### A 移動ロボットによる空間認識手法と課題

車輪型移動ロボットにおける SLAM の手法を紹介した。ホイールオドメトリによる自己位置推定では誤差を免れないことや、SLAM を用いることで推定された位置をある程度修正できることを述べた。

#### B 車輪型移動ロボットの制御手法

本研究で取り扱った車輪型移動ロボットの移動機構や自律移動における経路設計方法について記した。ここでは、移動ロボットと目標経路上にリファレンス点を用いた軌道追従制御をおこなう方法を紹介した。

#### C ワンボードマイコン制御によるセンサの小型化の例

ワンボードマイコンによる各センサの小型制御ユニットシステムを作成した例を紹介した。制御ユニットを小型化することで、室内空間などの狭い空間でのロボット制御に有効であることを示した。

本論文では、提案システムが智能化空間の構築をおこなう上で有効であることを述べ、智能化空間が様々な場面で用いられるようになるための智能化空間構築支援システムになる可能性があることを示した。空間内に設置された複数の分散センサにより構成される智能化空間は、分散センサの設置位置を知ることが重要であるが、一般的にその作業は煩雑である。そのため、本稿では、その位置を自動的に推定するシステムを提案した。具体的には、レーザ式測域センサを分散的に配置して構成される智能化空間を対象とし、空間内で移動ロボットを走行させ、SLAM により大域地図を生成する。分散センサが生成した局所地図との比較に基づいて、分散センサの位置を推定するシステムを提案した。地図の比

較には ICP マッチングを導入し、双方の座標系の対応関係を求めることで、分散センサの位置を更新した。

評価実験では、提案手法を実装した知能化空間を実際に構築して、移動ロボットを走行させて分散センサの位置推定が可能になることを示した。また、位置が更新された分散センサの局所地図を移動ロボットが走行中におこなう SLAM にも反映させることで、移動ロボットの全域地図の歪みも修正でき、地図のループクローズにも効果があることを示した。

## 7.2 今後の課題と展望

今後の課題として、SLAM におけるパーティクルのスキャンデータと分散センサの地図の対応点をより適切に求める手法の提案を挙げた。パーティクルを適切に分散させる手法の導入や、パーティクルフィルタの代わりに ICP マッチングを実装するなどを検討する。また、現システムでは分散センサの初期位置をあらかじめ与えている。分散センサに与えた初期値が実際の値と大きく離れている場合には、地図比較の初期段階での適切な比較ができず、結果として分散センサ位置の修正がおこなわれない。しかし、文献[60][61]で提案した検知方法のように、移動ロボットに再帰性反射板を取り付け、分散センサにおいてレーザ受光強度を評価することで、移動ロボットの接近を知ることができる。分散センサにより移動ロボットを検出した際の位置関係から、実際の位置に近い大まかな分散センサの初期位置を与えることが可能である。その後には本研究の提案手法をおこなうことで ICP マッチングの初期値問題を解消できる可能性が高い。

本論文で提案した知能化空間を構成する分散センサの位置座標推定方法は分散センサが広域かつ多量に空間に設置される場合に有効であると考えられる。今後は本研究の提案手法の活躍の場を広げるために、ワンボードマイコンにより制御された分散センサおよび移動ロボットの RT コンポーネント化[62][63]などをおこない、様々な環境でのシステムの実装と評価を容易におこなうことのできる柔軟なシステム構成も検討していく。柔軟なシステム構成により、知能化空間内に新たに分散センサが追加された場合も、移動ロボットがその分散センサ付近を通過するだけで、分散センサの位置推定ができるようになる。同様に、位置推定された分散センサが移動させられてしまった場合なども、移動ロボットによって移動後の分散センサの位置推定をおこなうことができる。また、知能化空間外に分散センサを設置した場合などには、移動ロボットを知能化空間外の分散センサ付近に向かわせることで、その分散センサの位置推定をおこなうことができる。知能化空間外に設置されていた分散センサを位置推定された知能化空間内のセンサとして用いることで、知能化空間を拡張することも可能になる。

今後このような分散センサの位置推定システムが拡張され、広まることで、様々な場所に広域・大量に分散配置されたセンサの位置推定が移動ロボットの空間情報から求められる

るようになると考えられる。分散センサを適当に巻いた空間を移動ロボットが通過するだけで簡単に、どこにでも知能化空間のような空間認識システムを構築することができるようになると期待できる。

## 付録

### A 移動ロボットによる空間認識手法と課題

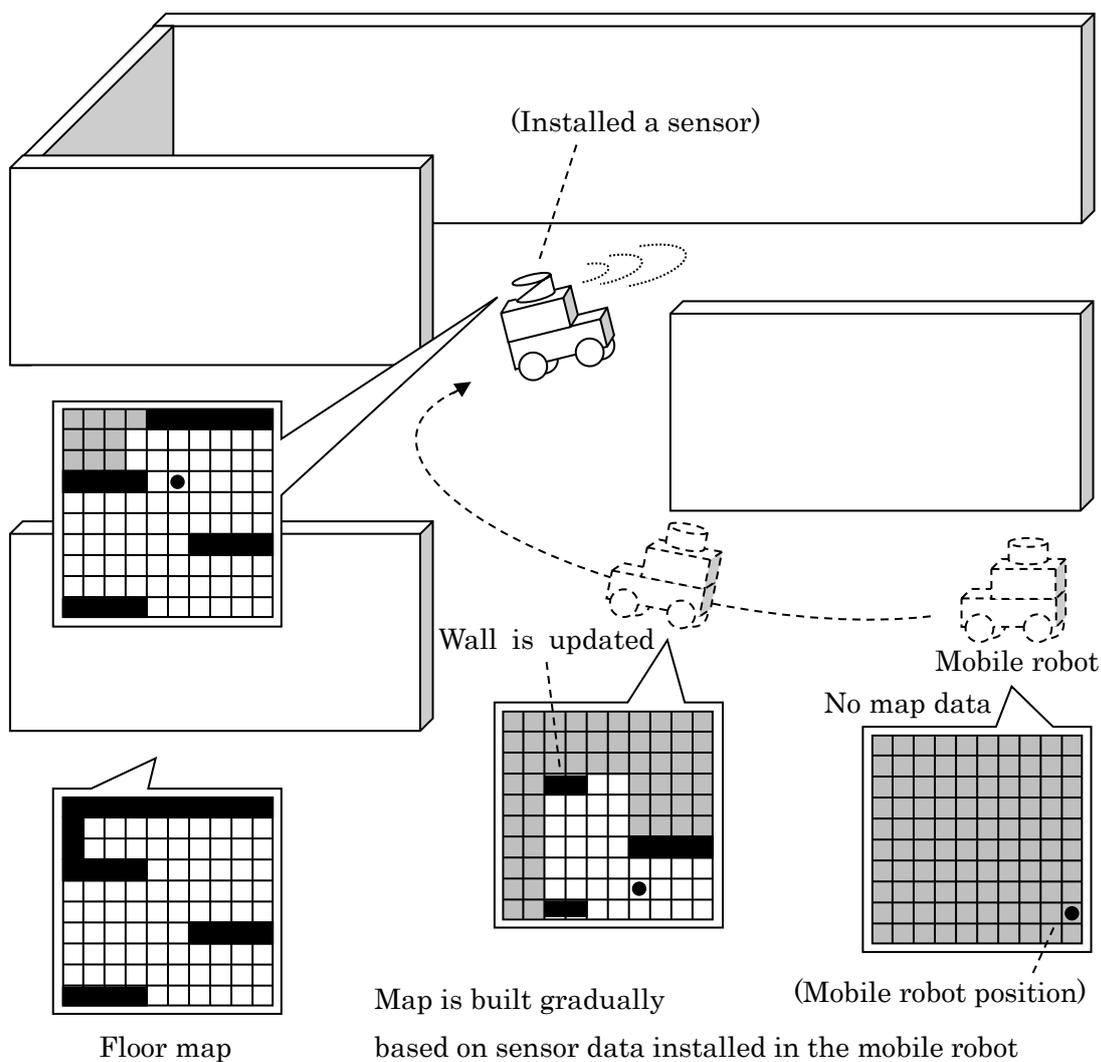


図 A.1 移動ロボットの SLAM の概要図

以下には、移動ロボットによる空間認識に関する研究背景を紹介する。移動ロボットが空間内に存在する物の位置関係を把握するためには、まず移動ロボット自身の位置を把握する必要がある。移動ロボットの位置を取得する方法は、そのロボットが有する移動機構によって異なる。近年では、二足歩行ロボットに代表する足を移動機構とする移動ロボッ

トもあるが、移動ロボット本体の重心の変化が影響するため比較的制御が高度である[64][65]。基本的な移動機構のひとつとしては、車輪がある[66]。車輪の回転から推定したホイールオドメトリのみによる自己位置では、路面との摩擦やスリップによる誤差を含むことが多く実用的ではない。そのため、ホイールオドメトリによる推定誤差を修正する方法がこれまでに提案されている[67][68]。修正方法は主に移動ロボットにセンサを搭載することで、周辺物体の位置を考慮しながら自己位置推定をおこなう[69][70]。周辺物体の位置は、移動ロボットの位置を基準として、周辺物体までの距離と方向を計測することで求めることができる。移動ロボットは作業をおこなう際に自律して移動する機会が多いため、移動ロボットの位置は逐次変化することが考えられる。空間認識のためには、逐次変化する移動ロボットの位置を把握することが必要である。しかし、上記の手法は主にホイールオドメトリに基づいているため、推定値の修正には限度がある[71][72]。

ホイールオドメトリによる位置推定の誤差を修正する手法として、**SLAM (Simultaneous Localization and Mapping)** と呼ばれる手法がある[37][38][73]。SLAM は、移動ロボットの空間認識において広く用いられており、ナビゲーションなどのために移動ロボットが自律して自己位置推定と地図生成を同時におこなう手法である。SLAM において、移動ロボットは自律して現在位置や周囲の物の位置情報を取得して地図生成することで空間認識をおこなう[74]。移動ロボットに搭載されるセンサには、近年ではネットワークカメラを用いた手法なども多く研究されている[75][76][77]が、レーザセンサを用いた手法はデータ量も少なく基本的とされている[37][38]。レーザセンサによる距離データを用いた基本的な地図生成方法の1つとして、平面地図がある。地図には平面地図のほかに三次元地図があるが、3次元地図は平面地図に比べ立体的な地図表現ができるがデータ量が膨大のため、平面地図に比べてリアルタイム処理には時間を要するというデメリットがある。リアルタイムで高速に動作させるシステムを考える場合、平面地図が適している。地図生成は、未知の環境では特に周囲環境の地図が必ずしも存在するとは限らないため重要である。このことから、まず地図生成を移動ロボットにおこなわせることが必要になる。

図 A.1 に平面地図を用いた移動ロボットの SLAM の概要図を示す。初めに移動ロボットは、自己位置と地図の情報がない状態である。周囲環境の平面地図を生成するためには、移動ロボットが走行中に計測した周辺の物体までの距離とそのときの移動ロボットの位置を取得する。そのため移動ロボットには、周辺の物体までの距離を計測するセンサを搭載している。計測した距離データは、地図情報として位置合わせして統合する。位置合わせには、ホイールオドメトリによる移動ロボットの現在位置を用いる。SLAM による自己位置は、移動ロボットが走行しながら生成したその時々々の平面地図を現在計測した距離データと比較することで推定される。平面地図にはその時々々の移動ロボットの自己位置も記録されているため、現在の距離データと位置合わせすることができる。各時点で保存した地図データと現在の距離データとの間に同じ領域があれば、その領域における地図の形状の対応関係から、現在の移動ロボットの位置を修正することができる。しかし、長時間や長距離の

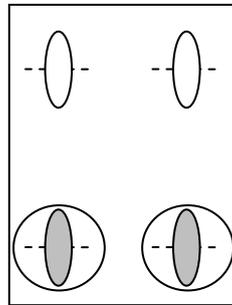
走行、路面の状態による誤差の蓄積の問題が完全に解消されるわけではなく、周回経路などでは、その誤差はループクローズとして顕著に表れる[78]。そのため、適切に推定誤差を修正する SLAM の手法が必要とされている [79]。

## B 車輪型移動ロボットの制御手法

車輪型移動ロボットは、構造が単純で制限が比較的容易であることから最も実用に近いと考えられている。本章では、車輪移動機構と移動タイプおよび動作制御手法を記す。

### B.1 移動機構

ここでは文献[58][59]を参考にして、車輪型移動機構を以下のステアリング方式（進行方向を変える方式）によって分類する。



[凡例]

図 B.1 車輪型移動ロボットのステアリング例

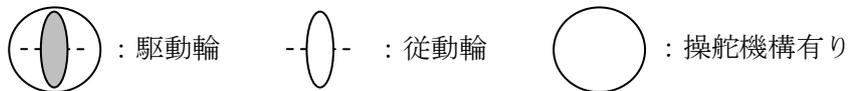


図 B.2 ステアリング方式の分類

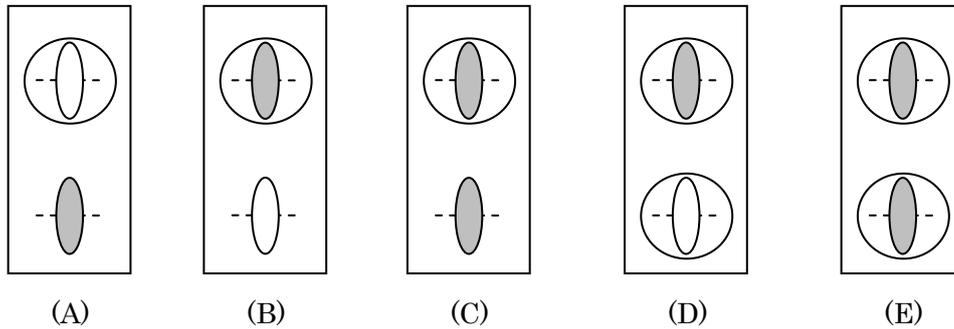


図 B.3 ステアリング方式の例

次に、各ステアリング方式の名称と特性を以下にまとめる。

表 B.1 ステアリング方式の名称と特性

	ステアリング方式の名称	直進	旋回	信地旋回	その場旋回	斜行	横行
凡例	二輪速度差方式	○	○	○	○	×	×
A	前輪ステアリング方式1	○	○	×	×	×	×
B	前輪ステアリング方式2	○	○	×	×	×	×
C	前輪ステアリング方式3	○	○	×	×	×	×
D	総輪ステアリング方式	○	○	○	×	○	○
E	独立ステアリング方式	○	○	○	○	○	○

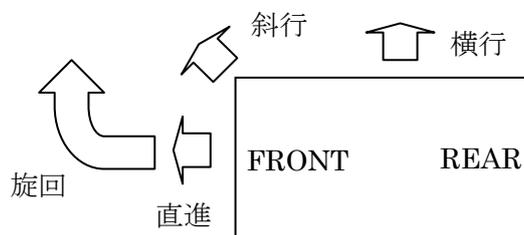


図 B.4 移動ロボットの動作タイプ

次に、本研究で使用した移動ロボットのステアリング方式とその動作を以下に示す。

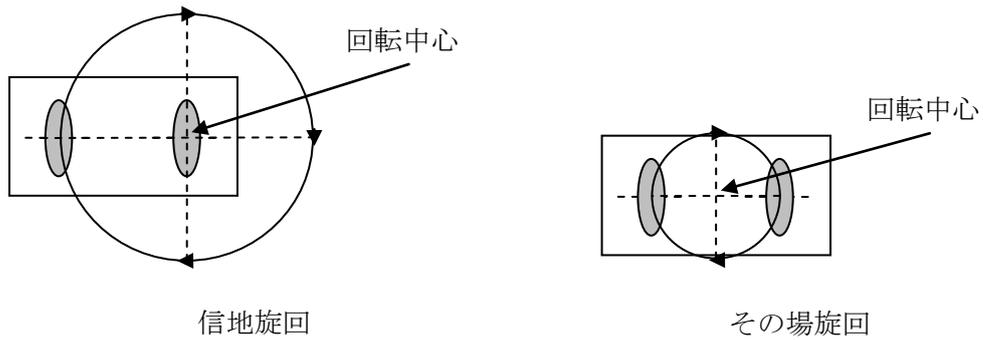
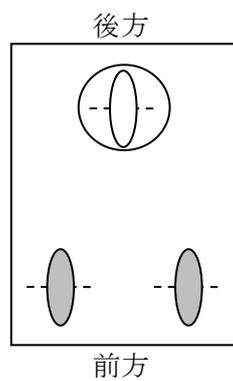


図 B. 5 移動ロボットの旋回タイプ



[本研究で用いる移動機構]

図 B. 6 本研究の移動ロボットのステアリング

## B.2 移動ロボットの軌道追従制御

産業用ロボットの作業経路などのように目的地までの走行経路が定められている場合や障害物が存在する環境下などでの経路を生成する場合に、移動ロボットがその経路上を走行するように制御することが軌道追従制御の目的である[15]。本章では移動ロボットの位置を基に、所定経路上の目標点を移動ロボットの座標値を取得する毎に与えていく軌道追従制御の手法を紹介する[58][59]。

### B.2.1 目標点追従制御

本研究でおこなう手法は、移動ロボットが常に所定経路上のある一定距離前方を目標点としてその目標点に至る二次曲線を走行するように制御することで、与えられた経路に追従させる。経路生成に二次曲線を用いるのは、例えば曲線の次数を更に高くしたとしても速度目標値の計算が複雑になる割には軌道追従制御の精度の向上が期待できないためである。また、低次数の曲線を利用することにより、短い時間で高速に制御できるためである。以下にその手法を示す。但し、経路上の目標点のことをリファレンス点とする。

- (1) 移動ロボットと経路上のリファレンス点がある一定距離を保つように、経路上でリファレンス点を動かす。
- (2) 移動ロボットとリファレンス点を結ぶ二次曲線を生成する。
- (3) 移動ロボットが二次曲線上を走行するような速度目標値（並進速度、角速度）を算出する。

以上の操作を移動ロボットの座標値を取得する毎におこなう。本手法による移動ロボットの軌道追従制御の様子を図 B.7 に示す。

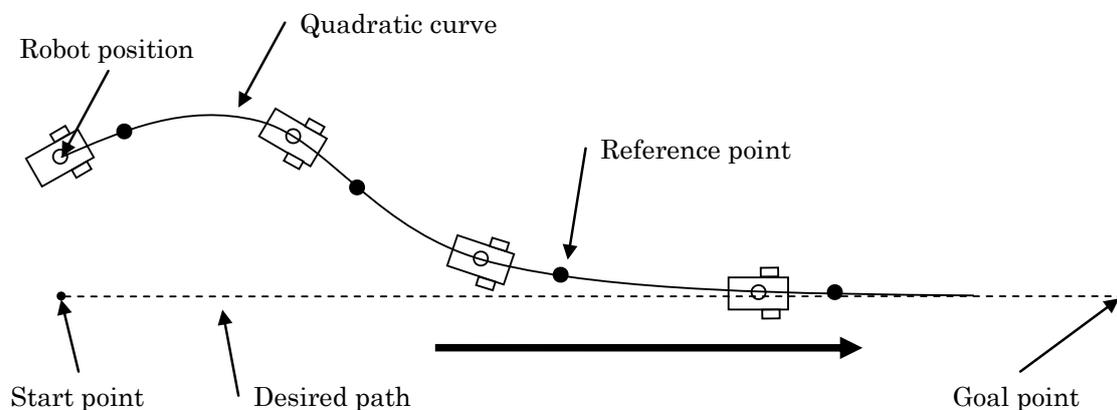


図 B.7 移動ロボットの軌道追従制御の様子

## B.2.2 速度目標値

移動ロボットから経路上のリファレンス点に至る二次曲線を計算し、さらにその二次曲線上を走行させるための速度目標値を計算する方法について記述する。図 B.8 には移動ロボットにおけるリファレンス点の座標系の変換図を示す。

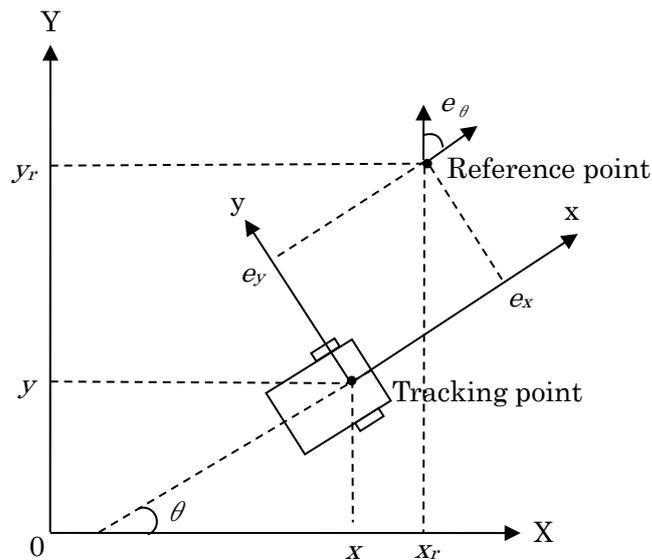


図 B.8 座標系の変換

移動ロボットの状態ベクトル

$$(x \ y \ \theta)^T \quad (\text{B.1})$$

リファレンス点の状態ベクトル

$$(x_r \ y_r \ \theta_r)^T \quad (\text{B.2})$$

車両座標系におけるリファレンス点の座標

$$(e_x \ e_y \ e_\theta)^T \quad (\text{B.3})$$

大局座標系からロボット座標系への変換行列は以下のように求められる。

$$\begin{pmatrix} e_x \\ e_y \\ e_\theta \end{pmatrix} = \begin{pmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_r - x \\ y_r - y \\ \theta_r - \theta \end{pmatrix} \quad (\text{B.4})$$

ここで、移動経路となる二次曲線を

$$y = Ax^2 \quad (\text{B.5})$$

$$A = \text{sign}(e_x) \frac{e_y}{e_x^2} \quad \text{sign}(e_x) = \begin{cases} 1 & (e_x > 0) \\ 0 & (e_x = 0) \\ -1 & (e_x \leq 0) \end{cases}$$

とする。式(B.5) は移動ロボットの進行方向とその変化量を示す。また、移動ロボットがこの二次曲線上を走行するための並進速度  $v$  と角速度  $\omega$  をそれぞれ距離微分、角度微分から求めると

$$v = \text{sign}(e_x) \sqrt{\dot{x}(1 + 4A^2x^2)} \quad (\text{B.6})$$

$$\omega = \frac{2Ax^3}{v^3} \quad (\text{B.7})$$

と求めることができ、時刻  $n\Delta t \leq t < (n+1)\Delta t$  において、

$$x = K_n(t - n\Delta t) \quad K_n = \text{sign}(e_x) \frac{\alpha}{1 + |A_n|} \quad (\text{B.8})$$

とすると、位置座標取得時間の間隔  $\Delta t$  が微小であることから、式(B.6) と式(B.7) はそれぞれ、

$$v_n \cong K_n \quad (\text{B.9})$$

$$\omega_n \cong 2A_n K_n \quad (\text{B.10})$$

と近似でき、これを時刻  $n\Delta t$  における速度目標値として与える。また、式(B.8) の  $\alpha$  については、以下のようにする。

$$\alpha = \begin{cases} v_{max} & : \text{通常モード} \\ v_{max} d_s / d_{s0} & : \text{停止モード} \end{cases} \quad (\text{B.11})$$

$d_s$  : 移動ロボットと目的地の距離

$d_{s0}$  :  $d_s$  が  $d_{s0}$  になった時、移動ロボットの並進速度を減速

次にリファレンス点の移動速度  $v_{rn}$  について以下のように設定する。

$$v_{rn} = \begin{cases} v_n d_0 / d_n & (e_x \geq 0) \\ 0 & (e_x < 0) \end{cases} \quad (\text{B.12})$$

$d_n$  : 移動ロボットとリファレンス点の距離

$d_0$  :  $d_n$  が距離感覚  $d_0$  を保つようにリファレンス点を移動

また、リファレンス点の初期位置は移動ロボットを中心とする半径  $d_{max}$  の円と経路の交点とする。図 B.9 には、リファレンス点の初期位置の決定法の概要図を示す。

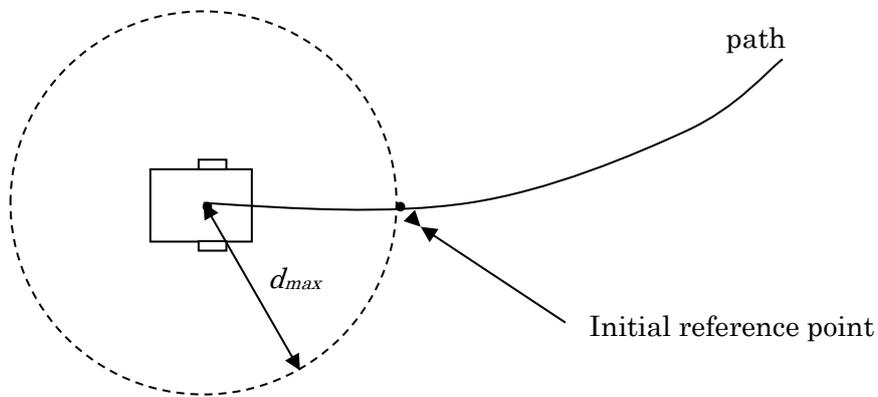


図 B.9 リファレンス点の初期位置の決定法

### B.2.3 軌道追従制御の実装

以下に本研究で実現した軌道追従制御の実装内容とその結果を示す。実装では移動ロボットの初期状態ベクトルを

$$(x, y, \theta)^T = (500, 500, -90)^T \quad (\text{B.13})$$

とし、原点(0,0) を中心とした半径 500 [mm] の円形経路を走行させた。  
 実際には表 A.2 に示すリファレンス点を中継地点として通過する経路移動をさせ、それに沿った二次曲線の経路を追従できていることが図 B.10 からわかる。  
 また、中継地点の切替えは移動ロボットが各中継地点から半径 500 [mm] の範囲に侵入したときに次の中継地点への切替えをおこなっている。

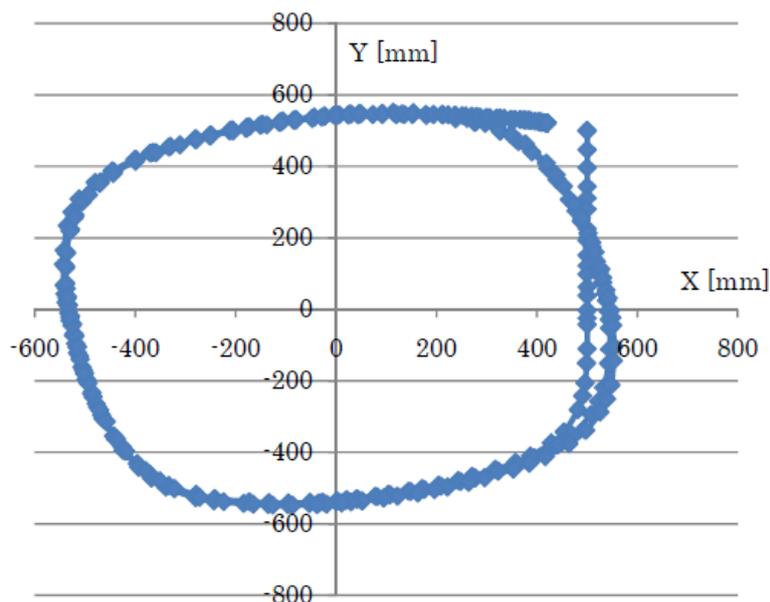


図 B.10 走行時の車輪の回転による軌跡の様子

表 B.2 リファレンス点の設置座標

中継点 1	中継点 2	中継点 3	中継点 4
500,-500	-500,-500	-500,500	500,500

#### B.2.4 まとめ

本章では、移動ロボットのホイールオドメトリを用いて状態変数を取得した。前述の速度目標値の算出結果を位置座標が取得される毎に制御値として送り、適当な制御結果を得ることができた。しかし、実際に長距離の経路を走らせると車輪と地面との摩擦によるエンコーダーの誤差が時間経過とともに蓄積して、結果的に実際の位置とは大きく誤差が出てしまうという問題が想定される。以上のことから長距離走行のためには蓄積する誤差を修正することや自己位置の推定方法が重要な課題となる。

## C ワンボードマイコン制御によるセンサの小型化の例

本研究における分散センサと同様に、他のセンサや移動ロボットの制御にもワンボードマイコンを導入した。図C.1に示す様にワンボードマイコンはノートPCよりも小さいため、コンパクトな形状で移動ロボットやセンサの制御をおこなうことができる。

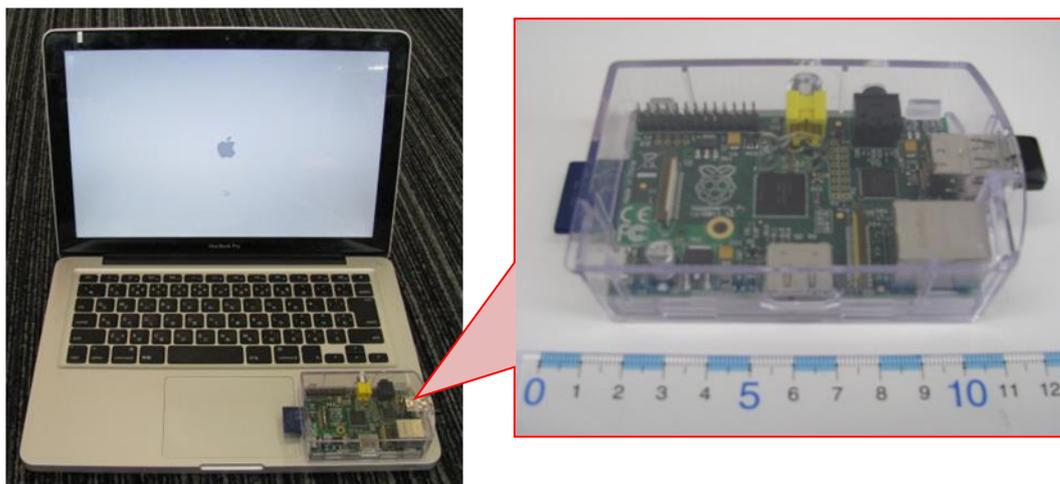


図 C. 1 ワンボードマイコン”Raspberry Pi”



(a)小型移動ロボット+Raspberry Pi

(b)小型移動ロボット+ノート PC

図 C. 2 Raspberry Pi による移動ロボット制御ユニットの小型化



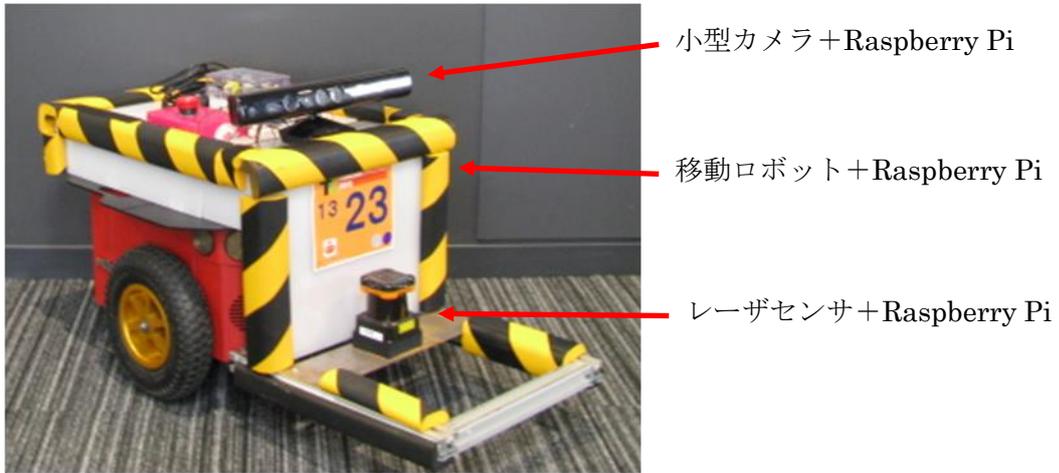
(a)小型カメラ+Raspberry Pi



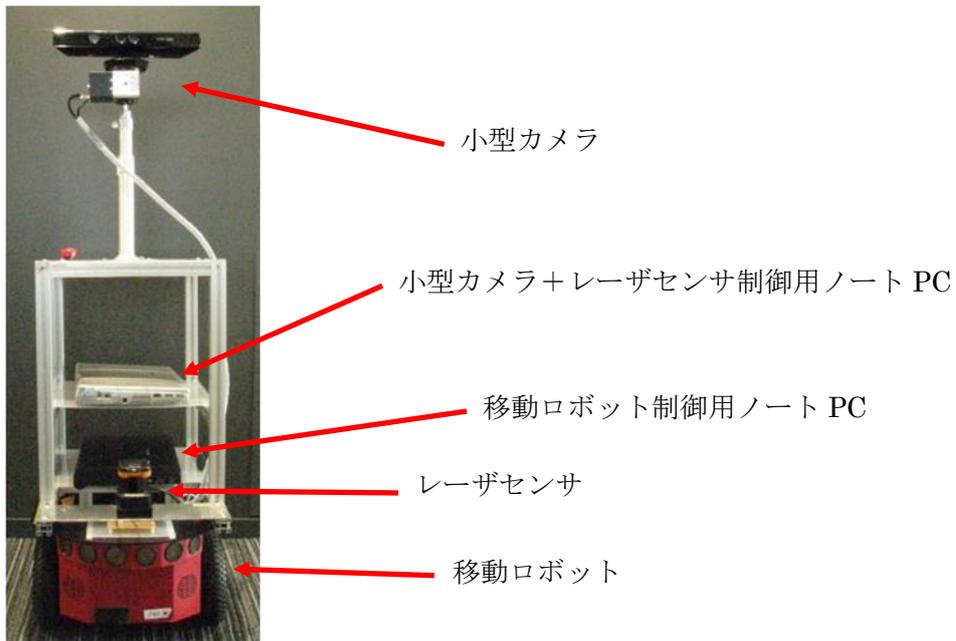
(b)ワイヤレスコントローラー+Raspberry Pi

### 図 C. 3 Raspberry Pi による制御ユニットの小型化例

図 C.2 に示すように、移動ロボットに搭載する制御 PC を小型化することで、積載重量面などでノート PC 搭載が困難な小型移動ロボットの制御も可能にした。また、図 C.3 に示す様に、各種のセンサ制御 PC も小型化した。これらの小型化したセンサと移動ロボットを組み合わせ、図 C.4 に示すように移動ロボット制御ユニットを小型することで、屋内のような狭い空間での実装を可能にした。



(a)Raspberry Pi による小型カメラ+レーザセンサ+移動ロボット制御ユニット



(b) ノート PC による小型カメラ+レーザセンサ+移動ロボット制御ユニット

図 C. 4 移動ロボット制御ユニット比較例

## 参考文献

- [1] Joo-Ho Lee, Hideki Hashimoto, “Intelligent Space - concept and contents”, *Advanced Robotics*, Vol.16, No.3, pp.265-280, 2002
- [2] T Sato, T Harada, T Mori, “Environment-Type Robot System “Robotic Room” Featured by Behavior Media, Behaviour Contents, and Behavior Adaptation.” *IEEE/ASME Trans. on Mechatronics*, Vol.9, No.3, pp.529-534, ISSN 1083-4435, 2004
- [3] Y. Nishida, T. Hori, T. Suehiro, and S. Hirai, “Sensorized environment for self-communication based on observation of daily human behavior,” in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, pp.1364–1372, 2000
- [4] TaeSeok Jin, Primoz Podrzaj, Hideki Hashimoto, “Position Uncertainty Reduction of Mobile Robot Based on DINDs in Intelligent Space” *Industrial Electronics, IEEE International Symposium on (Volume:4 )*, Vol.4, pp.2603 – 2607, 2006
- [5] David Herrero, Humberto Mart´inez, “Fuzzy Mobile-Robot Positioning in Intelligent Spaces Using Wireless Sensor Networks”, *Sensors (Basel)*, doi:10.3390/s111110820, pp.10820–10839, November 2011
- [6] 宜保他, “全方位視覚センサとレーザ距離センサによる人間の位置トラッキング”, *電気学会電子・情報・システム部門大会, MC3-22P2-I07*, pp. 578-579, 2002
- [7] Batalin M, Sukhatme G, Hattig M, “Mobile Robot Navigation Using a Sensor Network” In *Proceedings of International Conference on Robotics and Automation, New Orleans, LA, USA*, pp. 636-641, May 2004
- [8] Kazuyuki Morioka, Szilveszter Kovacs, Joo-Ho Lee, Peter Korondi, “A Cooperative Object Tracking System with Fuzzy-Based Adaptive Camera Selection”, *International Journal on Smart Sensing and Intelligent Systems*, Vol. 3, No. 3, pp.338-358, 2010
- [9] Kazuyuki Morioka, Joo-Ho Lee, Hideki Hashimoto, "Human-following mobile robot in a distributed intelligent sensor network", *IEEE Transactions on Industrial Electronics*, Vol.51, No.1, pp.229-237, 2004
- [10] 森岡一幸, 李周浩, 橋本秀紀, “知能化空間における移動ロボットの人間追従制御”, *日本ロボット学会誌*, Vol.22, No.1, pp.103-111,2004
- [11] TaeSeok Jin and Hideki Hashimoto, “Motion Estimation of Moving Target Using Multiple Images in Intelligent Space”, *Motion Planning*, Xing-Jian Jing (Ed.), ISBN: 978-953-7619-01-5, InTech, 2008
- [12] Tae-Seok Jin, Jang-Myung Lee, and Hideki Hashimoto, “Position Control of Mobile Robot for Human-Following in Intelligent Space with Distributed Sensors”, *International Journal of Control, Automation, and Systems*, Vol. 4, No. 2, pp. 204-216, April 2006
- [13] Yuichi Nakamura, Kazuyuki Morioka, “Human tracking based on direct communication among

- networked laser range scanners in an intelligent space”, The 7th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI2010), p534-535, 2010
- [14] Drazen Brscic, Takayuki Kanda, Tetsushi Ikeda, Takahiro Miyashita, “Person tracking in large public spaces using 3-D range sensors, IEEE Transactions on Human-Machine Systems”, Vol. 43, No. 6, pp. 522-534, 2013
- [15] Kazuyuki Morioka, Fumitaka Hashikawa, Tomoyasu Takigawa, “Human Identification Based on Walking Detection with Acceleration Sensor and Networked Laser Range Sensors in Intelligent Space”, International Journal on Smart Sensing and Intelligent Systems, Vol.6, No.5, pp.2040-2053, 2013
- [16] Xu Y, Ouyang Y, Le Z, Ford J, Makedon F, “Mobile Anchor-Free Localization for Wireless Sensor Networks”, Distributed Comput, Sensor Syst, 4549, pp.96-109, 2007
- [17] 大場 光太郎, 大原 賢一, “ユビキタス・ロボティクス”, 日本ロボット学会誌, Vol. 25, No. 4, pp. 505-508, 2007
- [18] Jae Hoon Lee, Yong-Shik Kim, Bong Keun Kim, Kohtaro Ohba, Hirohiko Kawata, Akihisa Ohya, Shin'ichi Yuta, “Security Door System Using Human Tracking Method with Laser Range Finders”, Proc. of IEEE International Conference on Mechatronics and Automation, pp.2060-2065, 2007
- [19] Dražen Brščić, Hideki Hashimoto, “Mobile Robot as Physical Agent of Intelligent Space”, CIT. Journal of Computing and Information Technology, Vol.17 No.1 Ožujak, pp81-94, 2009
- [20] Enriquez, G.; Hashimoto, S. “Wireless Sensor Network-Based Navigation for Human-Aware Guidance Robot”, In Proceedings of International Conference on Robotics and Biomimetics, Bangkok, Thailand, 21–26, pp. 2034-2039, February 2008
- [21] Mark Paskin, Carlos Guestrin, Jim McFadden, “A robust architecture for distributed inference in sensor networks”, IPSN '05 Proceedings of the 4th international symposium on Information processing in sensor networks, No. 8, IEEE Press Piscataway, NJ, USA, table of contents ISBN:0-7803-9202-7, 2005
- [22] Lazos L, Poovendran R, SeRLoc, “Robust Localization for Wireless Sensor Networks”, ACM Trans. Sensor Networks, Vol.1, pp.73-100, 2005
- [23] Takeshi Sasaki, Hideki Hashimoto: “Auto-Calibration of Distributed Laser Range Finders Based on Object Tracking”, The 25th Annual Conference Of The Robotics Society Of Japan, 1C21(1)-1C21(3), 2007
- [24] 佐々木毅, 橋本秀紀, “移動体トラッキングに基づく分散レーザレンジファインダの自動キャリブレーション” 第25回日本ロボット学会学術講演会, 1C21(1)-1C21(3), 2007
- [25] Takeshi Sasaki, Hideki Hashimoto, “Calibration of Distributed Laser Range Finders using Mobile Robots”, ROMOBEC2007, 2A2-K01, 2007
- [26] 佐々木毅, 橋本秀紀, “移動ロボットを用いた分散レーザレンジファインダのキャリブ

- レーション”, ロボティクス メカトロニクス講演会 2007, 2A2-K01, 2007
- [27] David Herrero, Humberto Mart´inez, “Fuzzy Mobile-Robot Positioning in Intelligent Spaces Using Wireless Sensor Networks”, *Sensors (Basel)*, doi:10.3390/s111110820, pp.10820-10839, 2011
- [28] Stanislav Funiak, Carlos Guestrin, Mark Paskin, Rahul Sukthankar, “Distributed Localization of Networked Cameras” *Information Processing in Sensor Networks, IPSN The Fifth International Conference on*, pp.34-42, 2006
- [29] Christopher Taylor, Ali Rahimi, Jonathan Bachrach, "Simultaneous Localization, Calibration, and Tracking in an ad Hoc Sensor Network ", *Information Processing in Sensor Networks IPSN 2006. The Fifth International Conference on*, pp27-33, 2006
- [30] NICULESCU, D. AND NATH, B. 2003. “Ad hoc positioning system (aps) using aoa”, In *Proceedings of INFOCOM’03. Vol. 3.* 1734-1743, 2003
- [31] N. Bulusu, D. Estrin, J. Heidemann “GPS-less Low Cost Outdoor Localization For Very Small Devices,” *IEEE Pers. Commun*, 7(5), pp.28-34, October 2000
- [32] Letchner J, Fox D, LaMarca A, “Large-Scale Localization from Wireless Signal Strength”, In *Proceedings of International Conference on Artificial Intelligence, Las Vegas, NV, USA*, pp. 15–20, June 2005
- [33] Menegatti E, Zanella A, Zilli S, Zorzi F, Pagello E, “Range-Only SLAM with a Mobile Robot and a Wireless Sensor Networks”, In *Proceedings of International Conference on Robotics and Automation, Kobe, Japan*, pp. 1699-1705, May 2009
- [34] Mantzel W.E, “Distributed Camera Network Localization Signals, Systems and Computers”, *Conference Record of the Thirty-Eighth Asilomar Conference on (Volume:2 )*, pp.1381-1386, 2004
- [35] Alexander T, Ihler Randolph, L Moses, “Nonparametric Belief Propagation for Self-Localization of Sensor Networks”, *IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS*, Vol. 23, No. 4, pp.809-819, 2005
- [36] A Rahimi, B Dunagan, T Darrel, “Simultaneous calibration and tracking with a network of non-overlapping sensors”, In *CVPR*, volume 1, pp. 187-194, 2004.
- [37] Sebastian Thrun, Wolfram Burgard, Dieter Fox, “Probabilistic Robotics” *Massachusetts Institute of Technology*, pp1-483, 2007
- [38] Sebastian Thrun, Wolfram Burgard, Dieter Fox 著, 上田隆一 訳, “確率ロボティクス” *Massachusetts Institute of Technology*, pp1-577, 2007
- [39] 北陽電気株式会社“スキャナ式レンジセン(測域センサ)”,  
[http://www.hokuyo-aut.co.jp/02sensor/07scanner/utm\\_30lx.html](http://www.hokuyo-aut.co.jp/02sensor/07scanner/utm_30lx.html)
- [40] Open CV リファレンスマニュアル, <http://opencv.jp/opencv-1.0.0/document/>
- [41] 奈良先端科学技術大学院大学 Open CV プログラミングブック制作チーム著, “ Open CV

- プログラミングブック“ pp.1- 413, 2007
- [42] 株式会社リバスト “ 知能機動型ロボットシステム Pioneer 3-DX “ ,  
<http://www.revast.co.jp/service/chinou/type02.html>
- [43] Kouichi Yoshizawa, Hideki Hashimoto, Masayoshi Wada, Shunji Mori, "Path Tracking Control of Mobile Robots Using a Quadratic Curve", 1996 IEEE Intelligent Vehicles Symposium, p.58-63, 1996
- [44] 吉沢浩一, “ ネットワーク型移動ロボットの階層的障害物回避に関する研究”, 東京大学修士論文, pp.1-31, 1997
- [45] D. Haehnel, D. Fox, W. Burgard, S. Thrun, “A highly efficient FastSLAM algorithm for generating cyclic maps of large-scale environments from raw laser range measurements”, In Proc. of the Conference on Intelligent Robots and Systems (IROS), pp.1-6, 2003
- [46] Bailey Tim, Nieto J, Guivant J, Stevens M, “Consistency of the EKF-SLAM Algorithm, Intelligent Robots and Systems”, 2006 IEEE/RSJ International Conference on, pp. 3562-3568, 2006
- [47] OpenSLAM, <https://www.openslam.org/>, Copyright and V.i.S.d.P.: Dirk Haehnel; Dieter Fox; Wolfram Burgard; Sebastian Thrun
- [48] OpenRTM, <http://www.openrtm.org/openrtm/ja>, Copyright c 2005-2010 National Institute of Advanced Industrial Science and Technology, Japan. All Rights Reserved
- [49] 長瀬雅之, 中本啓之, 池添明宏著, “はじめてのコンポーネント指向ロボットアプリケーション開発 ミドルウェア超入門編”, pp.1-213, 2008
- [50] F.Lu, E.Milios, “Robot Pose Estimation in Unknown Environments by matching 2D Range Scans”, Journal of Intelligent and Robotic Systems”, Vol.18, pp.249-275, 1997
- [51] Javier Minguez, Luis Montesano, Florent Lamiroux, “Metric-Based Iterative Closest Point Scan Matching for Sensor Displacement Estimation” IEEE TRANSACTIONS ON ROBOTICS, Vol. 22, No. 5, pp.1047-1054, October 2006
- [52] Raspberry pi, <http://www.raspberrypi.org/>
- [53] 福田和宏 著, “これ1冊でできる！ラズベリー・パイ超入門”, ソーテック社, pp.1-272, 2014
- [54] Matt Richardson 著, Shawn Wallace 著, 船田巧 訳, “Raspberry Pi をはじめよう”, オライリージャパン, pp.1-208, 2013
- [55] ARIA API (開発ソフトウェア) ARIA overview,  
<http://www.ai.rug.nl/vakinformatie/pas/content/Aria-manual/main.html>,  
Copyright 2002,2003,2004,2005 ActivMedia Robotics, LLC. All rights reserved,  
Copyright 2006 MobileRobots, Inc. All rights reserved
- [56] Satoshi Yamanaka, Kazuyuki Morioka, “Mobile Robot Navigation Using Hybrid Simplified Map with Relationships between Places and Grid Maps” 10th IFAC Symposium on Robot

- Control, pp.616-621, 2012
- [57] Milford M, Wyeth Gordon, Prasser D, “Efficient Goal Directed Navigation using RatSLAM, Robotics and Automation”, ICRA 2005. Proceedings of the 2005 IEEE International Conference on, pp.18-22, 2005
- [58] Ruth Schulz, Michael Milford, David Prasser , Wyeth Janet Wiles, “Learning spatial concepts from RatSLAM representations, Robotics and Autonomous Systems”, Vol 55(5), From Sensors to Human Spatial Concepts, pp. 403-410, 2007
- [59] Milford, Michael, Wyeth, Gordon, Prasser, David, “Simultaneous Localisation and Mapping from Natural Landmarks using RatSLAM”, 2004 Australasian Conference on Robotics and Automation, Australian Robotics and Automation Association Inc Canberra, pp.1-9, 2004
- [60] 粕谷勇輔, 橋川史崇, 森岡一幸 “移動ロボットと分散センサ群の柔軟なコミュニケーションに基づく知能化空間の構成支援システム”, 第15回 計測自動制御学会 システムインテグレーション部門講演会, 東京, pp0514-0517, 2014
- [61] 粕谷勇輔, 橋川史崇, 森岡一幸 “RT コンポーネントを利用した知能化空間を構成する分散センサ群の簡易位置推定システム”, 日本機械学会ロボティクス・メカトロニクス講演会 2014 富山市, pp.238, 2014
- [62] Noriaki Ando, Shinji Kurihara, Geoffrey Biggs, Takeshi Sakamoto, Hiroyuki Nakamoto, "Software Deployment Infrastructure for Component Based RT-Systems", Journal of Robotics and Mechatronics, Vol.23, No.3, pp.350-359, 2011
- [63] Noriaki Ando, Ken'ichi Ohara, Takashi Suzuki, and Kohtaro Ohba, "RTC-Lite: Lightweight RT-Component for Distributed Embedded Systems", SICE Journal of Control, Measurement, and System Integration (SICE JCMSI), Vol. 2, No.6, pp. 328-333, November 2009
- [64] Buschka, P.; Saffiotti, A.; Wasik, Z. Fuzzy “Landmark-Based Localization for a Legged Robot”, In Proceedings of International Conference on Intelligent Robotic Systems, Takamatsu, Japan, pp. 1205-1210, November 2000
- [65] Herrero-Pérez D, Martínez-Barberá H, Saffiotti A, “Fuzzy Self-Localization Using Natural Features in the Four-Legged League”. In Proceedings of RobuCup’2004, Lisbon, Portugal, pp. 110-121, July 2004
- [66] 光山宣史, 高島俊, “車輪式自律移動ロボットの制御”, 日本機械学会 [No.00-2]ロボティクス・メカトロニクス講演会 2000 講演論文集 2P1-41-058, 2000
- [67] Teddy N. Yap, Jr. and Christian R. Shelton, “Simultaneous Learning of Motion and Sensor Model Parameters for Mobile Robots”, 2008 IEEE International Conference on Robotics and Automation Pasadena, CA, USA, pp. 2091-2097, May 2008
- [68] N. Roy and S. Thrun, “Online self-calibration for mobile robots”, in Proc. IEEE ICRA’99, pp. 2292–2297, May 1999
- [69] 阪東茂, 油田信一, 坪内孝司, “屋内自律移動ロボットのための環境の特徴的方位の検出

- 手法”, 日本ロボット学会誌, vol31, No3, pp310-318, 2013
- [70] Leonard, John J, Durrant-Whyte, Hugh F, “Directed sonar sensing for mobile robot navigation”, Series The Springer International Series in Engineering and Computer Science, Vol. 175, XXI, pp.1-183, 1992
- [71] 渋谷奈保, 高橋友和, 井手一郎, “車載レーザスキャナによる距離データマップの構築と高精度自車位置推定”, 電子情報通信学会論文誌. D, 情報・システム / 電子情報通信学会 編, Vol92, No2, pp.215-225, 2009
- [72] 渋谷奈保, 佐藤准嗣, 高橋友和, 井手一郎, 村瀬洋, 小島祥子, 高橋新, “レーザレーダデータ間の対応付けによる自車位置情報の精度向上の検討,” 2006 年電子情報通信学会総合大会, D-12-50, pp.1-6, March 2006
- [73] Juan-Antonio Fernández-Madrigo, José Luis Blanco Claraco, “Simultaneous Localization and Mapping for Mobile Robots Introduction and Methods”, pp.1-499, 2012
- [74] 森岡博史, 李想揆, Tongprasit Noppharit, 長谷川修, “人の多い混雑な環境下での SLAM による移動ロボットのナビゲーション”, 第 28 回日本ロボット学会学術講演会予稿集 2010 年 pp.1-4, 2010
- [75] M.J Milford, D Prasser, G Wyeth, “Experience mapping: producing spatially continuous environment representations using RatSLAM”, in Australasian Conference on Robotics and Automation, Sydney, Australia, pp.1-11, 2005
- [76] David Prasser, Gordon Wyeth, Michael Milford, ”Experiments in Outdoor Operation of RatSLAM, In Proceedings of the 2004 Australasian Conference on Robotics and Automation, pp.1-6, 2004
- [77] 根岸善郎, 三浦純, 白井良明, ”全方位ステレオとレーザレンジファインダの統合による移動ロボットの地図生成”大阪大学 大学院 工学研究科 日本ロボット学会誌 Vol,21, No.6, pp.690-696, 2003
- [78] Cyrill Stachniss, Dirk Hähnel, Wolfram Burgard, “Exploration with Active Loop-Closing for FastSLAM, Intelligent Robots and Systems”, IROS 2004, Proceedings. 2004 IEEE/RSJ International Conference on (Volume:2 ), pp. 1505-1510 ,2004
- [79] 上田隆一, 新井民夫, 浅沼和範, 梅田和昇, 大隅久, “パーティクルフィルタを利用した自己位置推定に生じる致命的な推定誤りからの回復法” 日本ロボット学会誌, Vol.23, No.4, pp.466-473, 2005

## 業績一覧

### 学術論文（査読あり）

1

2015年4月 「Convenient position estimation of distributed sensors in intelligent spaces using SLAM for mobile robots」 共著（共著者：Fumitaka Hashikawa, Kazuyuki Morioka）, Journal of Robotics and Mechatronics

2

2015年3月

「An assistance system for building intelligent spaces based on map sharing among a mobile robot and distributed sensors」 共著（共著者：Fumitaka Hashikawa, Kazuyuki Morioka）, International Journal On Smart Sensing and Intelligent Systems

3

2014年7月

「SLAM-Based Sensor Calibration System for Easy Construction of Intelligent Spaces」 共著（Fumitaka Hashikawa, Kazuyuki Morioka）, International Conference on Advanced Intelligent Mechatronics (AIM2014 France), pp.1741-1746

4

2012年11月

「Mobile Robot SLAM Interacting with Networked Small Intelligent Sensors Distributed in Indoor Environments」  
共著（Fumitaka Hashikawa, Kazuyuki Morioka, Noriaki Ando） Simulation, Modeling, and Programming for Autonomous Robots 2012 (SIMPAN 2012 Japan) pp.275-286

5

2012年1月

「Calibration of Networked Sensors in an Intelligent Space Based on Interactive SLAM」 共著（Fumitaka Hashikawa, Kazuyuki Morioka） Proc. of the 17th International Symposium on Artificial Life and Robotics 2012 (AROB Japan) pp.1171-1174

6

2011年10月

「Mobile Robot Navigation Based on Interactive SLAM with an Intelligent Space」 共著（Fumitaka Hashikawa, Kazuyuki Morioka） Proc. of The 8th International Conference on Ubiquitous Robotics and Ambient Intelligence (KROS Korea) pp.55-56

7

2011年5月

「Human Tracking Based on Integration of Laser Range Scanners and Cameras in Intelligent Space」 共著 (Kazuyuki Morioka, Shintaro Kuroiwa, Fumitaka Hashikawa, Tomoyasu Takigawa) Proc. of 2011 8th Asian Control Conference (Asian Control Association Taiwan Republic of China) pp.719-724

## 学会発表

1

2013年6月

「移動ロボットと分散配置センサノードの地図情報共有に基づく知能化空間構築の簡易化」  
共同 (発表代表者: 橋川史崇) 日本機械学会ロボティクス・メカトロニクス講演会 2013 茨城

2

2012年9月

「知能化空間における分散センサと移動ロボットの協調 SLAM システム」 共同 (発表代表者: 橋川史崇) 日本ロボット学会 第30回記念学術講演会 2012 北海道

3

2011年12月

「移動ロボットの SLAM を利用した知能化空間におけるレーザ測域センサの位置姿勢推定」  
共同 (発表代表者: 橋川史崇) 計測自動制御学会 SI 部門講演会 2011 京都府

4

2010年6月

「分散配置した測域センサを利用した移動ロボットの SLAM」 共同 (発表代表者: 橋川史崇) 日本機械学会ロボティクス・メカトロニクス講演会 2010 北海道

## その他の研究活動

独立行政法人産業技術総合研究所での技術研修と共同発表(2012年04月01日～2013年03月31日)、オープンラボなどのイベント参加

## 謝辞

研究をおこなうにあたり、指導教員である森岡一幸先生には常に適切な御教授をいただきました。進学後の予定を見通した研究計画についても御指導いただきました。長い学生生活の中で、大学外部の研究機関での技術研修、共同研究の機会等もくださり、様々な方と関わり合いながら広い視野を以て研究をおこなう事ができました。ここに深く御礼申し上げます。また、これまでに研究に関するシステムにおける手法の考え方など数々の助言をしてくださった同研究室の皆様にも御礼申し上げます。研究のみならず皆様方には大学生活においても大変お世話になりました。本論文の審査の際には、副査をお引き受けいただきました森啓之先生、嶋田総太郎先生からの的確な御意見、御指導をいただくことで自分に足りない考え方、見聞を得ることができました。心より感謝申し上げます。

ありがとうございました。今後ともお会いする機会がございましたら、何卒宜しく願い申し上げます。