

平成 27 年度 修士論文

自律移動ロボットの自己位置推定評価

所属 三重大学大学院 工学研究科 機械工学専攻
研究室 メカトロニクス研究室

平成 26 年度入学 414M128

氏名 平 裕太郎

目次

1	はじめに	1
1.1	研究背景	1
1.2	従来研究	8
1.2.1	自己位置推定	8
1.2.2	従来手法と問題点	8
1.3	MieC の自律移動の課題とその解決	11
1.4	研究目的	14
2	自己位置推定精度の解析	15
2.1	概要	15
2.2	提案手法:L 字角周辺における自己推定精度のモデルに基づいた数学的解析	18
2.2.1	前提条件	18
2.2.2	自己位置推定精度について	18
2.2.3	L 字角の LRF 計測をモデル化	19
2.2.4	モデルから誤差 Δx , $\Delta \varphi$ を計算	21
2.2.5	レーザー 1 本 1 本がもつ誤差の行列表現	23
2.2.6	偶然誤差 n が発生させるずれの算出	24
2.2.7	ずれ推定楕円式の算出	28
2.3	実験	30

2.3.1	使用する計測機器	30
2.3.2	実験条件	31
2.3.3	実験から取得したデータ	31
2.4	ずれ推定楕円と実験のずれ比較	35
2.4.1	実験で取得した誤差の処理	35
2.4.2	データ点を間引く前の比較結果	35
2.4.3	データ点を間引いた後の比較結果	38
2.5	ずれ推定楕円による自己位置推定精度評価	42
2.6	自己位置推定精度解析のまとめと今後の展望	43
3	マッチング取捨選択	44
3.1	概要	44
3.2	LRF を用いたの自己位置推定	45
3.2.1	マッチングの基礎技術について	45
3.2.2	画像マッチング, マッチング率について	45
3.3	自己位置推定破綻防止のためのマッチング取捨選択	48
3.3.1	提案手法:距離データ点列からマッチング閾値設定	48
3.3.2	サポートベクターマシン (SVM) を使用した点列の分類	52
3.4	事前実験:マッチング率の差異の検証	53
3.5	SVM による判別罅ワ臙譽瀋蟻存	55
3.6	マッチング取捨選択のまとめと今後の展望	60
4	結びに	61
	付録	62
	参考文献	67

目 次

1.1	Autonomous mobile robot developed by Panasonic used only in the limited environment	2
1.2	Autonomous mobile vehicle developed by Google Inc.	3
1.3	Autonomous mobile vehicle developed by Nissan Co.,Ltd.	3
1.4	ROPITS, the small-sized autonomous mobile vehicle developed by Hitachi, Ltd.	4
1.5	Utilization rate of netshopping in each generation	6
1.6	Rate of purchase way	6
1.7	Small-sized autonomous mobile drone, developed by Amazon, will be used for transportation.	7
1.8	MieC LM developed by our laboratory	7
1.9	LRF UTM-40LX produced by HOKUYO AUTOMATIC	9
1.10	NS chart of removal unstable features and select a correnct matching	13
2.1	There are many L-shaped edges in outdoor environment.	15
2.2	The binary image of L-shaped edge	16
2.3	This figure shows the model of measurement. Δx , $\Delta \varphi$ are the slip of L-shaped edge.	17
2.4	This figure shows the model of measurement. Δx , $\Delta \varphi$ are the slip of LRF. . .	17

2.5	This figure shows when position x slips Δx , how a distance which one of lazer measures changes.	20
2.6	This figure shows when angle φ slips $\Delta\varphi$, how a distance which one of lazer measures changes.	20
2.7	This figure shows when \mathbf{E} becomes a smallest.	28
2.8	The landscape of the experiment focus on L-shaped edge	31
2.9	Standard diviations are calculated by using the results of the experiment. Then they are plotted in three dimensions.	32
2.10	This figure shows l - θ plane of Fig. 2.9.	33
2.11	This figure shows the approximate curved surface based on the standard diviation obtained from the experiment.	34
2.12	Ellipse obtained at 1.0[m] before thining out the data	36
2.13	Ellipse obtained at 2.0[m] before thining out the data	36
2.14	Ellipse obtained at 3.0[m] before thining out the data	36
2.15	Ellipse obtained at 4.0[m] before thining out the data	36
2.16	Ellipse obtained at 5.0[m] before thining out the data	37
2.17	Ellipse obtained at 6.0[m] before thining out the data	37
2.18	Ellipse obtained at 7.0[m] before thining out the data	37
2.19	Ellipse obtained at 8.0[m] before thining out the data	37
2.20	Ellipse obtained at 9.0[m] before thining out the data	38
2.21	Ellipse obtained at 10.0[m] before thining out the data	38
2.22	Ellipse obtained at 1.0[m] after thining out the data	39
2.23	Ellipse obtained at 2.0[m]after thining out the data	39
2.24	Ellipse obtained at 3.0[m] after thining out the data	40
2.25	Ellipse obtained at 4.0[m]after thining out the data	40

2.26	Ellipse obtained at 5.0[m] after thining out the data	40
2.27	Ellipse obtained at 6.0[m]after thining out the data	40
2.28	Ellipse obtained at 7.0[m] after thining out the data	41
2.29	Ellipse obtained at 8.0[m]after thining out the data	41
2.30	Ellipse obtained at 9.0[m] after thining out the data	41
2.31	Ellipse obtained at 10.0[m] after thining out the data	41
2.32	This figure shows the area of slip estimating ellipse.	42
3.1	Picture of the landscape LRF data are taken	47
3.2	left:Original bynary image, rigth:Data bynary image	47
3.3	The relationship between matching ratio and correct matching percentage .	49
3.4	The threshold of matching in each environment	49
3.5	The relationship between matching ratio and correct matching percentage in indoor and outdoor environment	50
3.6	Make a group from 5 data points	50
3.7	How to make roughness feature	51
3.8	Support Vector Machine	52
3.9	Picture and binary image of indoor environment	54
3.10	Picture and binary image of outdoor environment	54
3.11	Picture and binary image of bycycle	55
3.12	Example of experiment	55
3.13	The order and position of experiment	56
3.14	The landscape of experiment near the stairs	57
3.15	The template of indoor passage	58
3.16	NS chart of autonomous moving applied proposed method	59

4.1	Experiment data of indoor environment	66
4.2	Experiment data of outdoor environment	66

第1章

はじめに

1.1 研究背景

自律移動ロボットは、人間のように自ら状況を判断し、出発地からある目的地まで人間の手を借りずに走行するロボットのことを指す。自律移動の技術を車に適用すれば、人が運転しなくても目的地まで運んでくれる自律移動車の実現が、ロボットに適用すれば災害発生時、人間が作業するには危険が及ぶ場所での作業を代替させることが可能である。現在実用化されている自律移動ロボットは Fig. 1.1 のように人間によって整備、限定された場所でのみ実用化されている [1] が、上記のように人間の生活する環境での運用も始まっている。

自律移動ロボットの 1 つである自律移動車の研究には、様々な企業が力を注いでいる。代表的なものとして、米 Google 社の自律移動車は、Fig. 1.2 の外観から分かるように既存の自動車にレーザーセンサ (velodyne 社製 LIDAR) を複数、加えて GPS やエンコーダも装備している。日本の自動車メーカーでは日産自動車が同社開発の電気自動車、リーフ (Fig. 1.3) にカメラ 12 台、レーザースキャナ 4 台を搭載し、自動運転デモや実験を進めている。運転支援、準自動運転のように、完全な自動運転ではなく運転の一部を補助するものも開発され、実際に自動車に搭載されることも多くなっている。このように様々なセンサを組み合わ



Fig. 1.1 Autonomous mobile robot developed by Panasonic used only in the limited environment

せることで自己位置を推定し、自動運転を行うことが一般的となっている。内閣府のSIP(戦略的イノベーション創造プログラム)の自動走行システム研究開発計画[2]では、自動運転レベルを1から4の4段階に定め、2020年代後半以降には完全自動走行システムの市場化を目指すとしている。現在ではレベル2の準自動走行システムを達成するシステムが開発されてきている段階である。また米高速道路交通安全局(NHTSA)[3]で決められた定義によると、現在達成されている自動化レベルとしては、レベル2,3の部分的、条件付きの自動化とされている。具体的な研究開発の基準、計画が作成されていることから分かります。国、企業が連携して研究開発に注力しており、今後より高いレベルの自動運転が実現されていくことが考えられる。また、日立の自律移動モビリティであるFig. 1.4はつくば市のつくばモビリティロボット実験特区で実証実験を行っており、様々な企業の取り組みからも関心が高い分野であることが分かる。

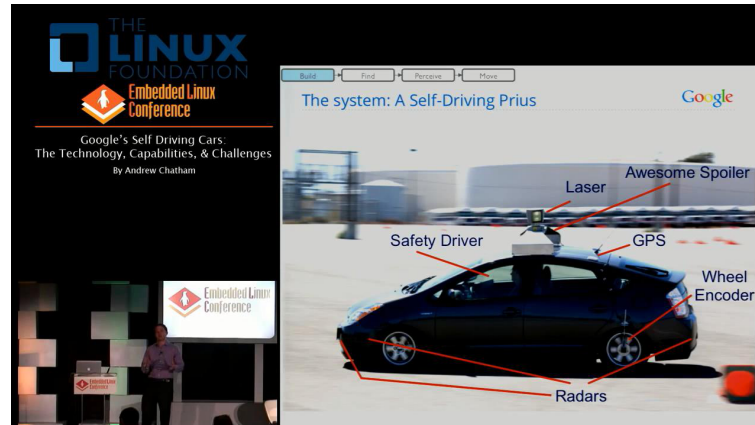


Fig. 1.2 Autonomous mobile vehicle developed by Google Inc.



Fig. 1.3 Autonomous mobile vehicle developed by Nissan Co.,Ltd.



Fig. 1.4 ROPITS, the small-sized autonomous mobile vehicle developed by Hitachi, Ltd.

しかし、日本において原動機付き自律移動ロボットの実験実証特区は上述のつくばモビリティロボット実験特区のみであり、実験特区内でさえもまだ完全自律移動ロボットの走行は認可されておらず、ハンドル付き搭乗型のロボットでないと走行実験すら出来ない状況である。実験特区が整備されたとしても、自動運転車、自律移動ロボット自体が接触事故を起こす可能性は 0 にはならない。最も実験が進んでいると思われる Google の自動運転車でさえ過去 6 年間で 11 件の事故を起こしたと発表されている。ほとんどが相手側に責任がある事故であるということであるが、実際の自動車で試験をすることは危険が伴うことは明確である。そこで、大型自律移動ロボットより危険性の低い比較的小型の自律移動ロボットの研究が現実的であり、小型自律移動ロボットを用いて研究を進め、その技術を徐々に大型に適用することで安全に自律移動ロボット、自動運転車を実現できると考える。小型ロボットの自律移動公開実験として、日本ではつくばチャレンジが前述のモビリティロボット実験特区で毎年開催されている [4]。小型自律移動ロボットそのものも利用価値が高く、例えば荷物配送サービスの自動化が挙げられる。Fig. 1.5 で示されるように、近年インターネットの普及によって、自宅にいても簡単に物を購入できるネットショッピングが大幅に普及してきている [5]。全体の 27.8% にも上り、私達の生活にも密接に関係してきている。内訳を詳しく見てみると、Fig. 1.6 から小型家電（タブレット・PC 周辺機器・ドライヤーなど）、PC、本・CD/DVD/BD 類、切符/チケットをネットショッピングで購入する人が多いと分かる。このカテゴリーに分類される商品の一つ一つは他カテゴリー商品と比べても比較的小形、軽量なものが多い。また、衣類（靴・アクセサリを含む）や小型家具、薬・化粧品もネットショッピングで購入する人が多く、これらの商品も前述の特徴が当てはまる [6]。現状ではこういった商品は配送目的地の最寄りの集積地から人の手で目的地まで届けられている。この「集積地から目的地までの輸送」という部分を小型自律移動ロボットに実行させることが小型自律移動ロボットの有用性を活かした有用な例の一つであると考えられる。自動配達ロボットの具体例としては Fig. 1.7 のドローンを使用した配達サービス Amazon Prime Air の実用化に向けた動きがあるが、この機体では屋内に侵入することは困難である。そのため本研究

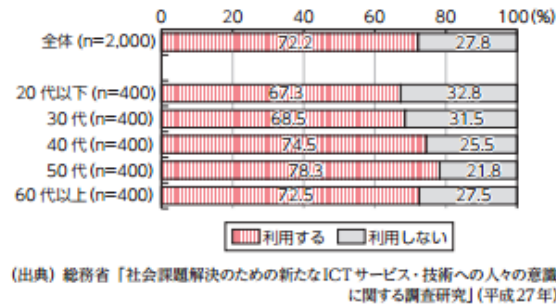


Fig. 1.5 Utilization rate of netshopping in each generation

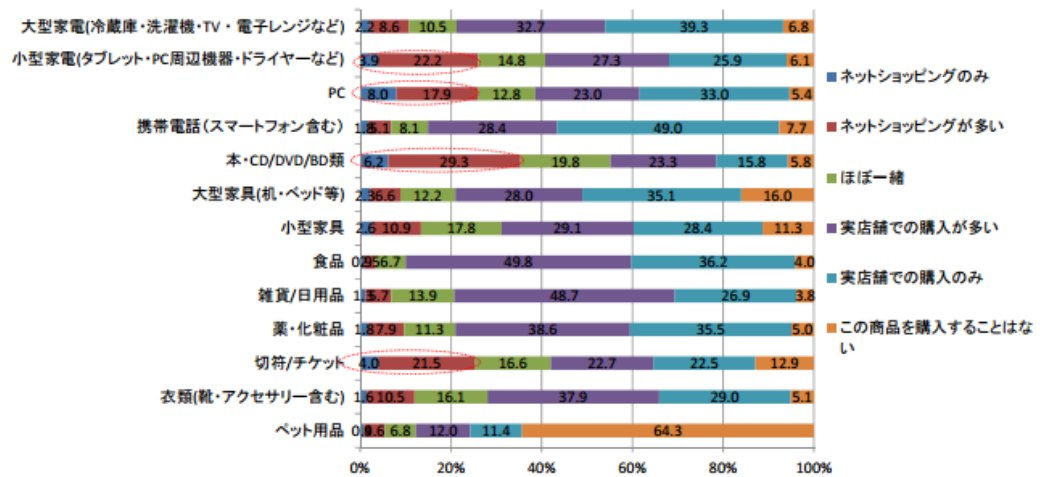


Fig. 1.6 Rate of purchase way

室では地上での自動配達を想定し，屋内にも侵入可能な Fig. 1.8 の MieC LM の研究を進めている．



Fig. 1.7 Small-sized autonomous mobile drone, developed by Amazon, will be used for transportation.

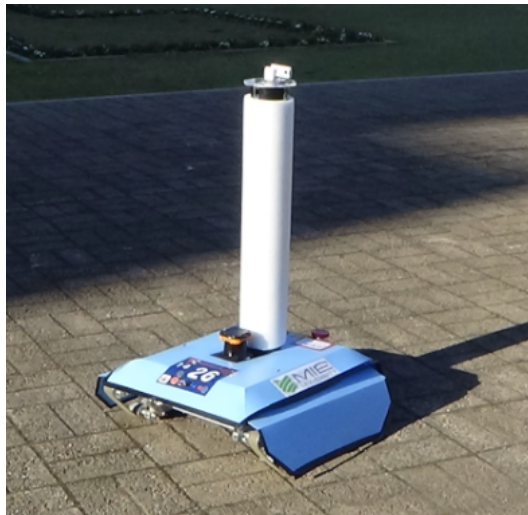


Fig. 1.8 MieC LM developed by our laboratory

1.2 従来研究

1.2.1 自己位置推定

自律移動で最も重要となるのは、現在ロボットがどこにいるのか分かっているかどうか、つまり自己位置を推定できているかどうかである。自己位置推定が可能であれば、どこまでが直進か、どの角を右左折するかを判断できる。この自己位置推定のために、種々のセンサを使用することになる。現在の主流は、ロボットの内部情報と、外界環境の情報を両方利用して自己位置を推定する手法である。内界センサとは、ロボットの内部情報を測定するセンサであり、ロボットの車輪の回転数を測定するロータリーエンコーダや、加速度からロボットの姿勢を測定するジャイロセンサなどがある。これら内界センサのみの自己位置推定(デッドレコニングと呼ばれる)のみで自律移動が可能ならばよいのだが、ロボットのために整備されていない私たちが生活する実環境では車輪のスリップなど様々な要因による誤差の発生、エンコーダやジャイロセンサが微分値で取得されることから生じる誤差の蓄積などによって自己位置推定が破綻してしまう。これは長い距離を走行すればするほど顕著に表れ、自律移動をする際の大きな問題の一つである。そこで、内界センサの他に外界センサを利用し、外界環境から自己位置を推定し、内界センサの誤差を解消する。現在非常によく使用されているのが Fig. 1.9 で示すような LRF(レーザーレンジファインダ)であり、周囲の物体との距離を高精度で測定することができる。この LRF を使用して環境地図を作成し、自律走行時利用する SLAM [7], Fast SLAM [8] といった手法が自己位置推定の主流となっている。このほかにも GPS やカメラによる自己位置推定も研究されている [9][10]。

1.2.2 従来手法と問題点

これまでの従来研究の中で、比較的安定な自己位置推定を実現している手法としては大きく分けて 3 種類ある。まずは LRF を用いた 2 次元データのみによる自己位置推定法である。最も基本的な手法としては、自律移動時得られた距離データと事前に取得しておいた距離



Fig. 1.9 LRF UTM-40LX produced by HOKUYO AUTOMATIC

データを比較して自分の位置を推定するもの、物体の形状を検出してその形状比較をして位置の補正をするもの、そして本研究室の MieC が採用している距離データから作成した周辺環境の画像を事前に生成しておいた画像とマッチングすることによって位置補正するものである [11][12]。この手法をデッドレコニングと組み合わせるものが一般的であるが、データ処理量が少なく済む代わりに、一度自己位置推定が破綻してしまうと復帰が困難であるという問題がある。

2次元データを使用するものとして頻繁に使用される手法はモンテカルロ法やパーティクルフィルタと呼ばれる手法である [13]。自律移動の際に今現在自分がいる場所周辺の情報を LRF で取得し(カレントマップと呼ばれる)推定した場所周辺にパーティクル(候補点)を散布する。このパーティクル一つ一つがロボットが存在する位置方向を持つ。事前に LRF で取得してある地図(グローバルマップと呼ばれる)とカレントマップを比較することによってパーティクルに存在確率の重みづけを行い、最も存在確率の高く計算されたパーティクルにロボットを動かす。存在確率が低いと判断されたパーティクルは消去され、候補点が絞られる。ある程度パーティクルが減少したらパーティクルを散布し直し、同じ手順で自己位置推定をしていく。直近のつくばチャレンジ 2015 においては Robot Operating System(ROS)を使用した自律移動ロボットが飛躍的に増えた。この ROS の自律移動プログラムの一つにパーティクルフィルタが採用されており、多くのチームがこのパーティクルフィルタを使用して長距離自律移動を実現していた。上記 LRF による自己位置推定の問題として、グロー

バルマップに存在しなかった観測点がカレントマップに現れたり、逆にグローバルマップに存在したはずの観測点がカレントマップに存在しなかった場合などに、自己位置推定が不安定となるというものがある。これは静止物体でないもの、例えば人間の足であったり、自転車などをマッチングに利用してしまうからであると考えられる。パーティクルフィルタを使用した手法では連続的に自己位置推定し続けるため、1 回の自己位置推定で誤った結果が出たとしても、次の自己位置推定で正しい自己位置推定ができればすぐに補正できることが強みである。そのためある程度の誤マッチングは問題が無いが、連続で誤マッチングが続いてしまう、例えばつくばチャレンジ 2012 のスタート地点のように人に囲まれ環境が以前と全く異なる状況が続くような場合であると自己位置推定が破綻してしまう。

こういったモンテカルロ法、パーティクルフィルタの弱点を補うべく、その他のセンサを併用する方法が 2 種類目の方法である。例としては江口らの DGPS を組み合わせたセンサフュージョンによる自己位置推定などがある [14]。赤井らは開空間、静的物体の少ない環境下においてもロバストな自己位置推定が可能であるように、上記の 2 次元モンテカルロ法と、磁気センサによる姿勢補正を組み合わせた方法を提案した [15]。この提案手法を用い、つくばチャレンジ 2012、2013 で課題の達成を果たしている。また、この論文内でモンテカルロ法を用いた場合、磁気センサを用いた場合、デッドレコニングのみを用いた場合、提案手法を用いた場合の 4 通りで走行させた結果、最も精度よく走行できたという結果も報告している。このようにセンサフュージョンを適切に施せばロバストな自律移動をすることが可能であるが、パーティクルフィルタによる自己位置推定には計算コストが相応にかかってしまうということが問題である。

もう 1 つの手法として、3-D LRF、2-D LRF を傾けることによって 3 次元のデータを取得できるようにした 3-DLRF 法によるマッチングがある [16][17][18]。この方法では上方特徴は短期間では変化しづらいということから、安定してマッチングを行うことが可能である。また、物体を判別することも可能であり、2-D ではできなかったことを可能にする。前述した Google Car もこの 3-D LRF を採用しており、車、歩行者などの判別が可能となってい

る．2-D LRF の 3-D データ取得については，2-D LRF を傾けて設置しミラーに反射させることで，傾けた部分の 3-D 情報とミラーに反射させて取得した 2-D 情報を一度に取得する手法も提案されている [19]．3-D 情報を使用する方法の問題点として，2-D データとは比較にならないほどの膨大なデータを処理しなくてはならないということがある．Google 開発の自動運転車は毎秒約 1GB のデータを処理していると言われており，小型自律移動ロボットに載せることができる CPU ではそこまでの高精度物体認識は難しい．また，マッチングは確かに安定しやすいが，不安定な特徴点が入ればマッチングが安定しなくなってしまうことは 2-D でマッチングする際と同様であり，根本的な解決になっていない．

問題点をまとめると，上記手法では内界センサと外界センサを組み合わせるとはいえ，ドレコニングに依存する部分が大きいので 1 度自己位置推定が破綻すると復帰が困難である．この問題の解決のため，マッチング点，マッチング回数を増やす手法が多く研究され，それに伴い自律移動が占める CPU コストは増大してしまった．そこで本研究では，マッチング 1 つ 1 つをより詳細に評価し，正しいデータのみを使用することによって CPU コストの増大を抑えながら自律移動を達成することを着眼点とし，研究を進める．

1.3 MieC の自律移動の課題とその解決

MieC は自己位置推定手法に局所画像マッチングを使用している．この自己位置推定手法は，中部圏の大学が中心となって開催された「つくばチャレンジへのチャレンジ」において，2014 年の三重大大学，2015 年の名古屋大学，大阪工業大学コースを走破する能力は有しているが，安定して走行できているとは言い難い．自己位置推定の破綻は多くは誤マッチングによるものであり，対処法としては，マッチングする前に防止する方法と，マッチング結果の評価によって防止する方法がある．事前データに無かった物体が入る等して誤マッチングを起こす問題の解決として，不安定物体を識別し画像上から取り除く必要がある．この問題に関しては当研究室の中林らによって SVM を用いた不安定物体識別法が提案されている [21]．現在は人間の足を取り除くにとどまっているが，学習を進めていくことで環境中の不安定物

体を正確に取り除くことが可能になる．LRF データ取得によるマップ生成時，そもそも自己位置推定に有効なデータであるかどうか定量的に評価することができないという問題もある．現在はオペレータがここならマッチングできるはずと判断した地点のデータが取得されるが，経験的なものであり，自律移動を不安定とする原因になる．以上はマッチング前にデータ処理をして誤マッチングを防ぐものである．マッチング結果の評価については，従来手法 [11][12] ではマッチング評価値を用いて画像マッチングによる自己位置推定をしている．つくばチャレンジにおいてマッチング率を 0.3 という一定の閾値で評価し，閾値を上回った場合だけマッチング結果を使用していた．また，環境毎に経験的に閾値を決定するとされている．しかし，マッチング率が高くても誤マッチングをしている場合も考えられ，反対に低くても誤マッチングしていない可能性もある．たとえば，屋外走行中に生垣を使用してマッチングを取ることは頻繁に行われ，植物の葉は安定してある場所に静止している物体とは断言できないがマッチングには使用される．この場合必ずしもマッチング率は高く出ないかもしれない．しかし補正としては利用価値のある値が出る可能性がある．反対に，建物の壁や角といった動かず，直線が多い特徴点を使用する場合はマッチング率は木々や生垣でのマッチング率より高くなるはずであり，マッチング率が一定の値よりも高いというだけで実際に適切な補正を得られているかどうかは分からない．マッチング前のデータ処理，マッチング後の評価という 2 つの課題を考慮した NS チャートを Fig. 1.10 に示す．このチャートのように 1 回のマッチングを丁寧なすることで，より安定した自律移動が可能となる．

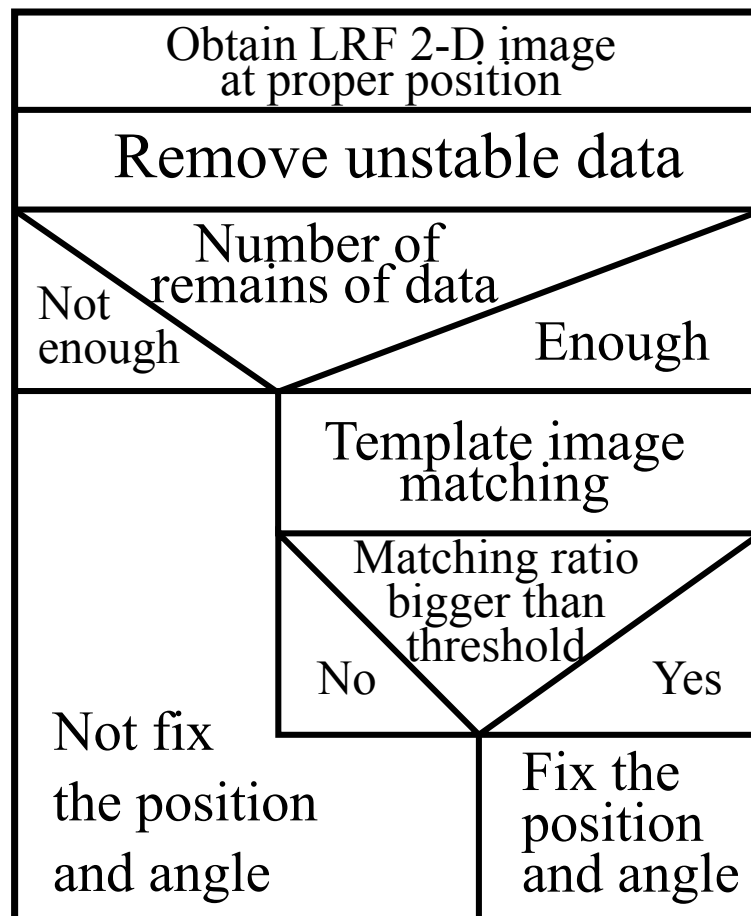


Fig. 1.10 NS chart of removal unstable features and select a correnct matching

1.4 研究目的

MieC の最終的な目標は、自動で学内のコンビニエンスストアで買い物をして戻ってくるというものである。出発地点から目標地点まで荷物を搬送することは前述の配送の自動化と同様である。そのため、自律移動はできるだけ省コストで実現したい。そこで、データ処理コストを増加させずにマッチング回数が少ない自律移動手法で自律移動を達成することを本研究の目的とする。この目的達成のためには、少ないマッチングデータで上述の 2 つの課題を解決することが求められる。マッチングによる補正回数が少ない自律移動では、1 回 1 回のマッチング精度が高い必要がある。この解決のために 2 章では「自己位置推定精度の解析」について述べる。また、局所マップマッチング法において誤マッチングは自己位置推定の破綻に直結する。誤マッチングが起きた場合、一旦外界センサによる補正は行わず、内界センサのみで次のマッチング点まで走行する方が自己位置推定の破綻を回避できる可能性が高いことが学内実験やつくばチャレンジ 2014, 2015 の参加を通して分かってきた。そこで、3 章では「マッチング結果の取捨選択」について述べる。

第2章

自己位置推定精度の解析

2.1 概要

マッチング回数を減らすことで自律移動のコストは削減できるが、同時に高精度の自己位置推定が要求される。このため、自己位置推定精度を解析し、精度を定量的に示す。本研究ではレーザーレンジファインダ (LRF) を使用して自己位置推定をする。自律移動をする際には様々な特徴点をこの LRF で取得してマッチングに使用することとなるが、その中でも頻繁に使用される代表的な特徴点として Fig. 2.1 のような L 字角に重点を置く。取得される

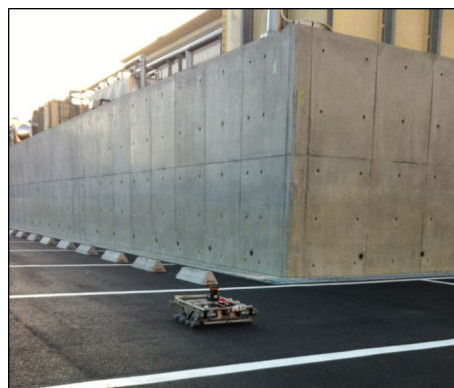


Fig. 2.1 There are many L-shaped edges in outdoor environment.

データは LRF から対象物体までの距離であり，レーザー角度とその角度において取得された距離を 2 値画像に加工して保存し，マッチングに使用する (Fig. 2.2)．しかし，計測データには誤差が発生するため真値データを取得することは難しく，誤差によって対象物体の位置姿勢にずれが発生する (Fig. 2.3)．そのため，マッチングが成功しても元のデータが大きく誤っているため正しく補正することができない可能性がある．一般的に計測は計測機器

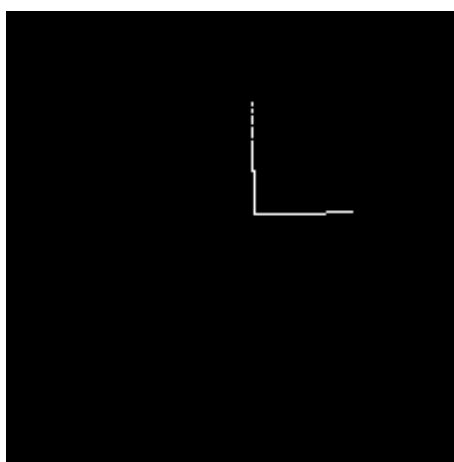


Fig. 2.2 The binary image of L-shaped edge

が固定されているものとして考える．しかし，計測機器ではなく対象物体が固定されているものとする，計測誤差によって計測機器である LRF の位置姿勢，すなわち自律移動ロボットの位置姿勢がずれていると考えることもできる (Fig. 2.4)．このずれがロボットの真の位置と推定位置の相違となるため，このずれを計測モデルから定量的に示し，自己位置推定精度を解析する．

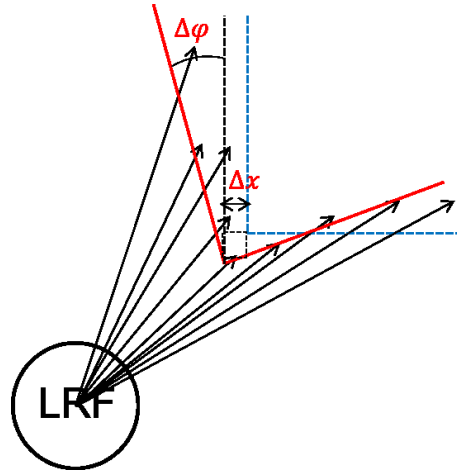


Fig. 2.3 This figure shows the model of measurement. Δx , $\Delta\varphi$ are the slip of L-shaped edge.

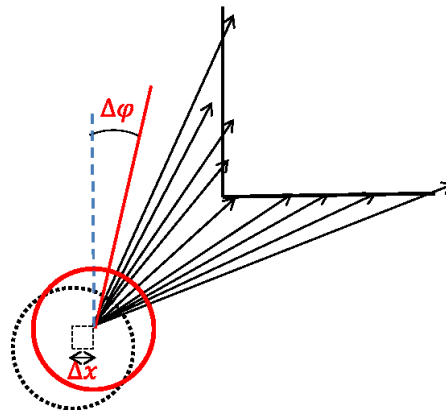


Fig. 2.4 This figure shows the model of measurement. Δx , $\Delta\varphi$ are the slip of LRF.

2.2 提案手法:L 字角周辺における自己推定精度のモデルに基づいた数学的解析

概要でも述べたとおり，屋外環境で安定した特徴点として使用することの多い Fig. 2.1 のような L 字角を特徴点の代表点として計測のモデル化を行い，L 字角周辺における自己推定精度を解析する．

2.2.1 前提条件

L 字角を検出する際，L 字の各面に等しい本数のレーザーが照射される時が L 字角を最も正確に検出できると考えられるので，L 字角の先端から 45° の直線上に LRF は位置するものとする．実環境において無限に広がる L 字は存在しないが，L 字の二面が同じ長さを持つような場合もこの 45° 地点に LRF を設置するのが適当であると考えられる．同じ長さを持っていない場合でもレーザーの本数を絞ることでマッチングに用いる L 字面の長さは等しくすることが可能であり，同じ状況を作り出すことが可能であるため，この条件下で実験する．

2.2.2 自己位置推定精度について

使用する 2-D LRF は，1 本のレーザーを回転させて計測するタイプのものである．前方 270° の距離を計測し，角度分解能は 0.25° であるため，1081 点の距離データを取得している．今回対象とするのは自律移動の際マッチングに頻繁に使用される建物などの L 字角であり，取得されるデータは LRF とこの L 字角までの各レーザー角度における距離データである．この距離データから LRF の位置のずれを算出することで自己位置推定の精度が解析できる．計測された距離データには真値という誤差が乗らない値が存在するが，計測機器を使用する際の誤差によって真値を得ることは不可能である．そこで，誤差を最小にするように近似する方法が様々存在する．今回の計測で発生する誤差は計測ごとにランダムに変化する誤差，偶然誤差である．計測データ点の誤差が最小になるような L 字角をフィッティングす

る時、偶然誤差を最小にするようにフィッティングすることになり、この場合に適切な近似法は最小二乗法である。最小二乗法で誤差が小さく真値に近い L 字角を算出することは可能であるが、誤差を完全に取り除くことはできない。各位置において最小二乗法を適用し、L 字角のずれが小さくなる点でマッチングすることが精度のよいマッチングであると言える。本研究では、Fig. 2.3 のモデルのように、真値付近に LRF を置いて計測した場合を想定し、系統誤差は較正をして除いた上で、偶然誤差のみについて考える。その偶然誤差によって最小二乗法で求められたロボットの推定位置が真値と $\Delta x, \Delta \varphi$ ずれることになる。この真値とのずれがどのように分布するか考察していく。

2.2.3 L 字角の LRF 計測をモデル化

以下の二図のように L 字角をモデル化する。モデルの LRF は実際に真値に設置されているものとする。自己位置推定を行う際、LRF で計測するとレーザー 1 本 1 本に偶然誤差が存在するため、ずれが発生する。したがって、その誤差によって、真値に設置されているはずであるのに、 $\Delta x, \Delta \varphi$ だけ自己位置推定がずれてしまうと考えられる。実際のレーザーの計測距離を \tilde{l} 、モデルから求められた推定距離を \hat{l} とすると、Fig. 2.5, Fig. 2.6 を描くことができる。この \tilde{l} と \hat{l} の誤差が実測時の偶然誤差によってどうなるかを見ることでマッチング精度を算出することができる。

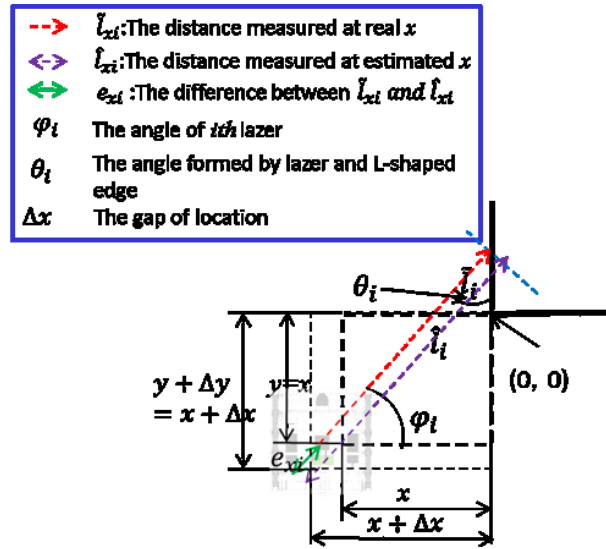


Fig. 2.5 This figure shows when position x slips Δx , how a distance which one of laser measures changes.

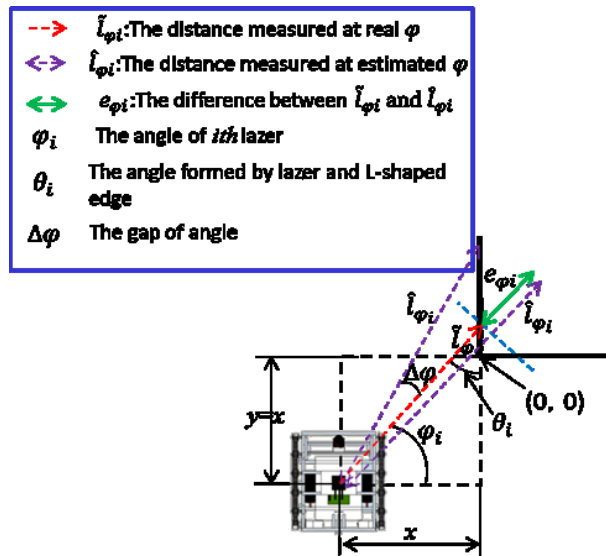


Fig. 2.6 This figure shows when angle φ slips $\Delta \varphi$, how a distance which one of laser measures changes.

2.2.4 モデルから誤差 Δx , $\Delta\varphi$ を計算

LRF の位置ずれ Δx による誤差

LRF の位置がずれた場合のモデルは Fig. 2.5 である．L 字角のロボットの進行方向と平行な面か，垂直な面のどちらに当たるかで場合分けをする必要がある．

i) $0 < \varphi_i < \pi/4$ のとき (L 字角のロボットの進行方向と垂直な面に当たる場合) の計測値 \tilde{l}_{x_i}

$$\tilde{l}_{x_i} = \frac{x}{\sin \varphi_i} + n_{x_i} \quad (2.1)$$

推定値 \hat{l}_{x_i}

$$\hat{l}_{x_i} = \frac{x + \Delta x}{\sin \varphi_i} \quad (2.2)$$

となるので，誤差 e_{x_i} は以下ようになる．

$$e_{x_i} = \tilde{l}_{x_i} - \hat{l}_{x_i} = -\frac{1}{\sin \varphi_i} \Delta x + n_{x_i} \quad (2.3)$$

ii) $\pi/4 \leq \varphi_i < \pi/2$ のとき (L 字角のロボットの進行方向と平行な面に当たる場合) の計測値 \tilde{l}_{x_j}

$$\tilde{l}_{x_j} = \frac{x}{\cos \varphi_j} + n_{x_j} \quad (2.4)$$

推定値 \hat{l}_{x_j}

$$\hat{l}_{x_j} = \frac{x + \Delta x}{\cos \varphi_j} \quad (2.5)$$

となるので誤差 e_{x_j} は以下ようになる．

$$e_{x_j} = \tilde{l}_{x_j} - \hat{l}_{x_j} = -\frac{1}{\cos \varphi_j} \Delta x + n_{x_j} \quad (2.6)$$

今回の実験では Fig. 2.5 , Fig. 2.6 を見ても分かるように，LRF の右斜め前方に L 字角があるとして LRF の右斜め前方 90° 分のレーザーを使用する．また， 0° , 90° に近いデータに関しては計測することができないので， $10^\circ \sim 80^\circ$ の間のデータを用いることとし，281 点の

データ点が対象となる．以降では全データが $2h$ である一般式を与える． Δx による誤差をベクトルでまとめて表すと以下ようになる．

$$E_x = - \begin{pmatrix} \frac{1}{\sin \varphi_1} \\ \vdots \\ \frac{1}{\sin \varphi_h} \\ \frac{1}{\cos \varphi_{h+1}} \\ \vdots \\ \frac{1}{\cos \varphi_{2h}} \end{pmatrix} \Delta x + \begin{pmatrix} n_{x_1} \\ \vdots \\ n_{x_h} \\ n_{x_{h+1}} \\ \vdots \\ n_{x_{2h}} \end{pmatrix} \quad (2.7)$$

LRF の角度ずれ $\Delta\varphi$ による誤差

LRF の角度がずれた場合のモデルは Fig. 2.6 である．こちらもどちらの面に当たるかで場合分けをする必要がある．

i) $0 < \varphi_i < \pi/4$ (L 字角のロボットの進行方向と平行な垂直な面に当たる場合) のときの計測値 \tilde{l}_{x_i}

$$\tilde{l}_{\varphi_i} = \frac{x}{\sin \varphi_i} + n_{\varphi_i} \quad (2.8)$$

推定値 \hat{l}_{φ_i}

$$\hat{l}_{\varphi_i} = \frac{x}{\sin(\varphi_i + \Delta\varphi)} \quad (2.9)$$

となるので，誤差 e_{φ_i} は以下ようになる．

$$e_{\varphi_i} = \tilde{l}_{\varphi_i} - \hat{l}_{\varphi_i} = \frac{x \cos \varphi_i}{\sin^2 \varphi_i} \Delta\varphi + n_{\varphi_i} \quad (2.10)$$

ii) $\pi/4 \leq \varphi_i < \pi/2$ のとき (L 字角のロボットの進行方向に平行に平行な面に当たる場合) の計測値 \tilde{l}_{φ_j}

$$\tilde{l}_{\varphi_j} = \frac{x}{\cos \varphi_j} + n_{\varphi_j} \quad (2.11)$$

推定値 \hat{l}_{φ_j}

$$\hat{l}_{\varphi_j} = \frac{x}{\cos(\varphi_j + \Delta\varphi)} \quad (2.12)$$

となるので誤差 e_{φ_j} は以下ようになる .

$$e_{\varphi_j} = \tilde{l}_{\varphi_j} - \hat{l}_{\varphi_j} = -\frac{x \sin \varphi_j}{\cos^2 \varphi_j} \Delta\varphi + n_{\varphi_j} \quad (2.13)$$

LRF の位置ずれ Δx による誤差と同じように , 誤差をまとめてベクトル表記すると以下のようになる .

$$\mathbf{E}_{\varphi} = - \begin{pmatrix} -\frac{x \cos \varphi_1}{\sin^2 \varphi_1} \\ \vdots \\ -\frac{x \cos \varphi_h}{\sin^2 \varphi_h} \\ \frac{x \sin \varphi_{h+1}}{\cos^2 \varphi_{h+1}} \\ \vdots \\ \frac{x \sin \varphi_{2h}}{\cos^2 \varphi_{2h}} \end{pmatrix} \Delta\varphi + \begin{pmatrix} n_{\varphi_1} \\ \vdots \\ n_{\varphi_h} \\ n_{\varphi_{h+1}} \\ \vdots \\ n_{\varphi_{2h}} \end{pmatrix} \quad (2.14)$$

2.2.5 レーザー 1 本 1 本がもつ誤差の行列表現

小節 2.2.4 においてモデルから幾何学的に求めた Δx , $\Delta\varphi$ によるレーザー誤差をまとめて行列で表現すると次のように書くことができる .

$$\mathbf{E} = - \begin{pmatrix} \frac{1}{\sin \varphi_1} & -\frac{x \cos \varphi_1}{\sin^2 \varphi_1} \\ \vdots & \vdots \\ \frac{1}{\sin \varphi_h} & -\frac{x \cos \varphi_h}{\sin^2 \varphi_h} \\ \frac{1}{\cos \varphi_{h+1}} & \frac{x \sin \varphi_{h+1}}{\cos^2 \varphi_{h+1}} \\ \vdots & \vdots \\ \frac{1}{\cos \varphi_{2h}} & \frac{x \sin \varphi_{2h}}{\cos^2 \varphi_{2h}} \end{pmatrix} \begin{pmatrix} \Delta x \\ \Delta\varphi \end{pmatrix} + \begin{pmatrix} n_1 \\ \vdots \\ n_h \\ n_{h+1} \\ \vdots \\ n_{2h} \end{pmatrix} \quad (2.15)$$

ここで ,

$$\Delta \mathbf{r} = \begin{pmatrix} \Delta x \\ \Delta\varphi \end{pmatrix} \quad (2.16)$$

$$J = \begin{pmatrix} \frac{1}{\sin \varphi_1} & -\frac{x \cos \varphi_1}{\sin^2 \varphi_1} \\ \vdots & \vdots \\ \frac{1}{\sin \varphi_h} & -\frac{x \cos \varphi_h}{\sin^2 \varphi_h} \\ \frac{1}{\cos \varphi_{h+1}} & \frac{x \sin \varphi_{h+1}}{\cos^2 \varphi_{h+1}} \\ \vdots & \vdots \\ \frac{1}{\cos \varphi_{2h}} & \frac{x \sin \varphi_{2h}}{\cos^2 \varphi_{2h}} \end{pmatrix} \quad (2.17)$$

$$\mathbf{n} = \begin{pmatrix} n_1 \\ \vdots \\ n_h \\ n_{h+1} \\ \vdots \\ n_{2h} \end{pmatrix} \quad (2.18)$$

とすると， E を以下の式に書き換えることができる．

$$E = \mathbf{n} - J \Delta \mathbf{r} \quad (2.19)$$

ここで， J はヤコビ行列で，位置，姿勢誤差 $\Delta \mathbf{r}$ をレーザー 1 本 1 本の誤差に変換する行列となっている．また \mathbf{n} は偶然誤差行列である．

この E が 0 に近くなるほど，真値に近い値となるが，偶然誤差 \mathbf{n} によって完全に真値と一致することはない．この偶然誤差 \mathbf{n} によって真値からどれくらい計測位置姿勢がずれるかを考察する．

2.2.6 偶然誤差 \mathbf{n} が発生させるずれの算出

小節 2.2.2 でも述べた，最小二乗法を計測距離データに適用し，計測データ点から L 字角を推定する際のずれ Δx ， $\Delta \varphi$ を算出する．式 (2.19) がレーザー誤差の式となっているので，

この式に最小二乗法を適用する，すなわち

$$\frac{\partial^T \mathbf{E} \mathbf{E}}{\partial \Delta \mathbf{r}} = \mathbf{0} \quad (2.20)$$

となるように $\Delta \mathbf{r}$ を決定する．展開すると

$$\begin{aligned} \frac{\partial^T \mathbf{E} \mathbf{E}}{\partial \Delta \mathbf{r}} &= \frac{\partial(T(\mathbf{n} - \mathbf{J} \Delta \mathbf{r})(\mathbf{n} - \mathbf{J} \Delta \mathbf{r}))}{\partial \Delta \mathbf{r}} \\ &= \frac{\partial(T\mathbf{n} - {}^T\Delta \mathbf{r}^T \mathbf{J})(\mathbf{n} - \mathbf{J} \Delta \mathbf{r})}{\partial \Delta \mathbf{r}} \\ &= \frac{\partial^T \mathbf{n} \mathbf{n}}{\partial \Delta \mathbf{r}} - \frac{\partial^T \mathbf{n} \mathbf{J} \Delta \mathbf{r}}{\partial \Delta \mathbf{r}} - \frac{\partial^T \Delta \mathbf{r}^T \mathbf{J} \mathbf{n}}{\partial \Delta \mathbf{r}} + \frac{\partial^T \Delta \mathbf{r}^T \mathbf{J} \mathbf{J} \Delta \mathbf{r}}{\partial \Delta \mathbf{r}} = \mathbf{0} \end{aligned} \quad (2.21)$$

となる．以降では式 (2.21) の各項の計算を行う． $\Delta \mathbf{r}$ は Δx , $\Delta \varphi$ を成分に持つベクトルであるから，偏微分は次のように書きかえることができる．

$$\frac{\partial^T \mathbf{E} \mathbf{E}}{\partial \Delta \mathbf{r}} = \begin{pmatrix} \frac{\partial^T \mathbf{E} \mathbf{E}}{\partial \Delta x} & \frac{\partial^T \mathbf{E} \mathbf{E}}{\partial \Delta \varphi} \end{pmatrix} \quad (2.22)$$

式 (2.21) の右辺の初項 ${}^T \mathbf{n} \mathbf{n}$ は Δx , $\Delta \varphi$ を含まないので 0 となる．

$$\frac{\partial^T \mathbf{n} \mathbf{n}}{\partial \Delta \mathbf{r}} = \mathbf{0} \quad (2.23)$$

第 2 項 ${}^T \mathbf{n} \mathbf{J} \Delta \mathbf{r}$ は，以下のように計算される．

$$\begin{aligned} {}^T \mathbf{n} \mathbf{J} \Delta \mathbf{r} &= (n_1 \cdots n_h \ n_{h+1} \cdots n_{2h}) \begin{pmatrix} J_{x_1} & J_{\varphi_1} \\ \vdots & \vdots \\ J_{x_h} & J_{\varphi_h} \\ J_{x_{h+1}} & J_{\varphi_{h+1}} \\ \vdots & \vdots \\ J_{x_{2h}} & J_{\varphi_{2h}} \end{pmatrix} \begin{pmatrix} \Delta x \\ \Delta \varphi \end{pmatrix} \\ &= \Delta x \sum n_i J_{x_i} + \Delta \varphi \sum n_i J_{\varphi_i} \end{aligned} \quad (2.24)$$

$$\frac{\partial^T \mathbf{n} \mathbf{J} \Delta \mathbf{r}}{\partial \Delta \mathbf{r}} = (\Sigma n_i J_{x_i} \quad \Sigma n_i J_{\varphi_i}) \quad (2.25)$$

ここで, J_x, J_φ は式 (2.17) の各成分である. 第 3 項 ${}^T \Delta \mathbf{r}^T \mathbf{J} \mathbf{n}$ の計算は,

$$\begin{aligned} {}^T \Delta \mathbf{r}^T \mathbf{J} \mathbf{n} &= \begin{pmatrix} \Delta x & \Delta \varphi \end{pmatrix} \begin{pmatrix} J_{x_1} & \cdots & J_{x_h} & J_{x_{h+1}} & \cdots & J_{x_{2h}} \\ J_{\varphi_1} & \cdots & J_{\varphi_h} & J_{\varphi_{h+1}} & \cdots & J_{\varphi_{2h}} \end{pmatrix} \begin{pmatrix} n_1 \\ \vdots \\ n_h \\ n_{h+1} \\ \vdots \\ n_{2h} \end{pmatrix} \\ &= \Delta x \Sigma^T J_{x_i} n_i + \Delta \varphi \Sigma^T J_{\varphi_i} n_i \end{aligned} \quad (2.26)$$

$$\frac{\partial^T {}^T \Delta \mathbf{r}^T \mathbf{J} \mathbf{n}}{\partial \Delta \mathbf{r}} = (\Sigma J_{x_i} n_i \quad \Sigma J_{\varphi_i} n_i) \quad (2.27)$$

第 4 項 ${}^T \Delta \mathbf{r}^T \mathbf{J} \mathbf{J} \Delta \mathbf{r}$ の計算は以下ようになる.

$$\begin{aligned} {}^T \Delta \mathbf{r}^T \mathbf{J} \mathbf{J} \Delta \mathbf{r} &= \begin{pmatrix} \Delta x & \Delta \varphi \end{pmatrix} \begin{pmatrix} J_{x_1} & \cdots & J_{x_h} & J_{x_{h+1}} & \cdots & J_{x_{2h}} \\ J_{\varphi_1} & \cdots & J_{\varphi_h} & J_{\varphi_{h+1}} & \cdots & J_{\varphi_{2h}} \end{pmatrix} \begin{pmatrix} J_{x_1} & J_{\varphi_1} \\ \vdots & \vdots \\ J_{x_h} & J_{\varphi_h} \\ J_{x_{h+1}} & J_{\varphi_{h+1}} \\ \vdots & \vdots \\ J_{x_{2h}} & J_{\varphi_{2h}} \end{pmatrix} \begin{pmatrix} \Delta x \\ \Delta \varphi \end{pmatrix} \\ &= \Delta x^2 \Sigma J_{x_i}^2 + 2 \Delta x \Delta \varphi \Sigma J_{x_i} J_{\varphi_i} + \Delta \varphi^2 \Sigma J_{\varphi_i}^2 \end{aligned} \quad (2.28)$$

$$\frac{\partial^T {}^T \Delta \mathbf{r}^T \mathbf{J} \mathbf{J} \Delta \mathbf{r}}{\partial \Delta \mathbf{r}} = (\Delta x \Sigma J_{x_i}^2 + 2 \Delta \varphi \Sigma J_{x_i} J_{\varphi_i} \quad 2 \Delta x \Sigma J_{x_i} J_{\varphi_i} + 2 \Delta \varphi \Sigma J_{\varphi_i}^2) \quad (2.29)$$

以上の計算結果を式 (2.21) に代入すると次のようになる .

$$\begin{aligned} \frac{\partial^T \mathbf{E} \mathbf{E}}{\partial \Delta \mathbf{r}} &= \mathbf{0} - (\Sigma n_i J_{x_i} \quad \Sigma n_i J_{\varphi_i}) - (\Sigma J_{x_i} n_i \quad \Sigma J_{\varphi_i} n_i) \\ &\quad - (\Delta x \Sigma J_{x_i}^2 + 2\Delta \varphi \Sigma J_{x_i} J_{\varphi_i} \quad 2\Delta x \Sigma J_{x_i} J_{\varphi_i} + 2\Delta \varphi \Sigma J_{\varphi_i}^2) = \mathbf{0} \end{aligned}$$

さらに計算を進めると以下の式が得られ , この式は偶然誤差 n から生じるずれを表している .

$$\begin{aligned} {}^T \mathbf{n} \mathbf{J} &= {}^T \Delta \mathbf{r}^T \mathbf{J} \mathbf{J} \\ {}^T \Delta \mathbf{r} &= {}^T \mathbf{n} \mathbf{J} ({}^T \mathbf{J} \mathbf{J})^{-1} \end{aligned} \tag{2.30}$$

2.2.7 ずれ推定楕円式の算出

小節 2.2.6 で偶然誤差によるずれを算出した．計測し，偶然誤差を取得すればずれを算出することは可能であるが，計測していない地点のずれももてるから推定可能であるはずである．本小節ではずれを推定する楕円を求める．このずれ推定楕円が小さければ小さいほど自己位置推定の精度は高くなる．ずれ推定楕円が最小となるのは E が最小となる時である． $E = n - J\Delta r$ において， n は $2h$ 次元をもつ誤差ベクトル， $J\Delta r$ はその $2h$ 次元中に存在する超平面に拘束されている．この E が最も小さくなる時は，この超平面上に垂直に n が投影された時である．図示すると，次の Fig. 2.7 となる．この図を見てわかるとおり， E は 0

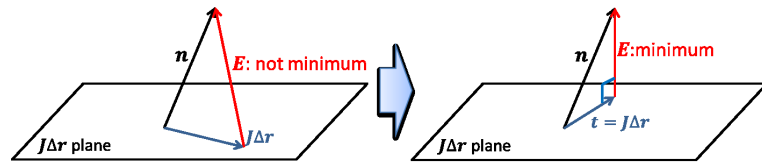


Fig. 2.7 This figure shows when E becomes a smallest.

とならないこともある．今回求めるものは E が最小となるような Δx ， $\Delta \varphi$ であり，それは Fig. 2.7 で示す平面上に垂直に n が投影される時である． E は平面に投影されて t となるので，この平面上のベクトル t と， $J\Delta r$ が同じ大きさとなる時， E が最小となる Δx ， $\Delta \varphi$ が求まる． t は n を平面に投影したベクトルである．ここで，偶然誤差は標準正規分布に従うことが分かっているため，これを利用してずれを推定する．1 次元正規分布に従う場合， 3σ のうちに 99.7% が含まれることが一般的に知られている．今回は二次元であるため 98.9% の $(\Delta x, \Delta \varphi)$ が含まれることになる．今， $|t| = 3\sigma$ とすると，偶然誤差であれば 98.9% がこの範囲内に存在し， $|t| = |J\Delta r|$ とするので， $|J\Delta r| = 3\sigma$ である．この $|J\Delta r| = 3\sigma$ の両辺を二乗すると，左辺は ${}^T\Delta r {}^T J J \Delta r$ となる．この時， ${}^T J J$ は 2×2 行列となるため，簡

略化のために，次のように置き換える

$${}^T J J = \begin{pmatrix} a & b \\ b & c \end{pmatrix} \quad (2.31)$$

また，右辺の σ^2 は，標準正規分布であれば 1 となるため，以下のように計算できる．

$$\begin{aligned} |J \Delta r| &= 3\sigma \\ |J \Delta r|^2 &= 9\sigma^2 \\ {}^T \Delta r J J \Delta r &= 9 \\ a\Delta x^2 + 2b\Delta x\Delta\varphi + c\Delta\varphi^2 &= 9 \\ a/9\Delta x^2 + 2b/9\Delta x\Delta\varphi + c/9\Delta\varphi^2 &= 1 \end{aligned} \quad (2.32)$$

これは偶然誤差を Δx - $\Delta\varphi$ 平面に投影した結果を表しており，傾いた楕円の方程式となっている．このずれ推定楕円によって，実際に計測していない地点のずれの分布を推定可能となるが，この推定楕円がずれを推定することができるかどうかを次節 3.5 の実験でデータを取得し，次々節 2.4 で比較確認する．

2.3 実験

提案手法がずれを推定できているかどうかを確認するため，実際に計測し，偶然誤差，標準偏差を取得した．

2.3.1 使用する計測機器

本研究ではデータ取得のため LRF を使用する．使用した LRF は二次元平面距離データを取得できる北陽電機株式会社製の UTM-30LX(Fig. 1.9) であり，

Table 2.1 The specification of UTM-30LX

検出距離及び検出体	検出保障値 0.1 ~ 30[m](白ケント紙)：最大検出距離 60[m](出力限界値) 最小検出物 130[mm] (10[m]):距離により変動する
測距精度	0.1 ~ 10[m]: ± 30 [mm] , 10 ~ 30[m]: ± 50 [mm] (白ケント紙) 周囲照度 3000lx 以下 白ケント紙 0.1 ~ 10[m] : ± 30 [mm] 周囲照度 100000lx 以下 白ケント紙 0.1 ~ 10[m] : ± 50 [mm]
測距分解能及び 繰り返し精度	1[mm] 単位 0.1 ~ 10[m]: $\sigma < 10$ [mm] , 10 ~ 30[m]: $\sigma < 30$ [mm] (白ケント紙) 周囲照度 3000lx 以下 $\sigma < 10$ [mm](白ケント紙 10[m] まで) 周囲照度 100000lx 以下 $\sigma < 30$ [mm](白ケント紙 10[m] まで)
走査角度	270°
角度分解能	約 0.25° (360°/1440 分割)

一般的に平面の距離データを取得する LRF では，1 本のレーザーを回転させて距離を計測しており，本研究で使用する 2-D LRF もこのような仕組みとなっている．

2.3.2 実験条件

Fig. 2.8 のように実際に LRF を L 字角の指定位置に設置し，LRF と L 字角の距離データを取得した．実験は三重大学看護棟の駐車場にある L 字で行った．L 字の材質はコンクリートである．

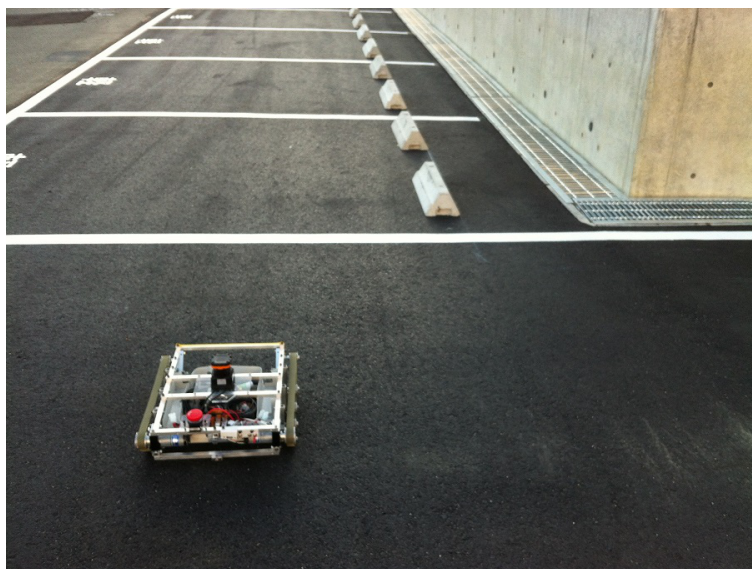


Fig. 2.8 The landscape of the experiment focus on L-shaped edge

今回の実験では Fig. 2.5 , Fig. 2.6 で示す x が 1.0 , 2.0 , 3.0 , 4.0 , 5.0 , 6.0 , 7.0 , 8.0 , 9.0 , 10.0[m] となるように LRF を設置し，L 字に向けて水平 $10^{\circ} \sim 80^{\circ}$ の範囲を 1000 回ずつ計測した．LRF の設置高さは地上から約 0.2[m] である．

2.3.3 実験から取得したデータ

計測から得られた距離データから標準偏差 (σ) を求め，計測距離 [m]，レーザーと対象物体の面がなす角度 [$^{\circ}$]，標準偏差 [m] を 3 次元プロットすると，Fig. 2.9 となった．

この結果から，ある計測距離以上，角度以下となる時，標準偏差が大きくなっており，Tab.2.1 の仕様に記載されている繰り返し測距精度と比較して約 10 倍ほどの大きさの違い

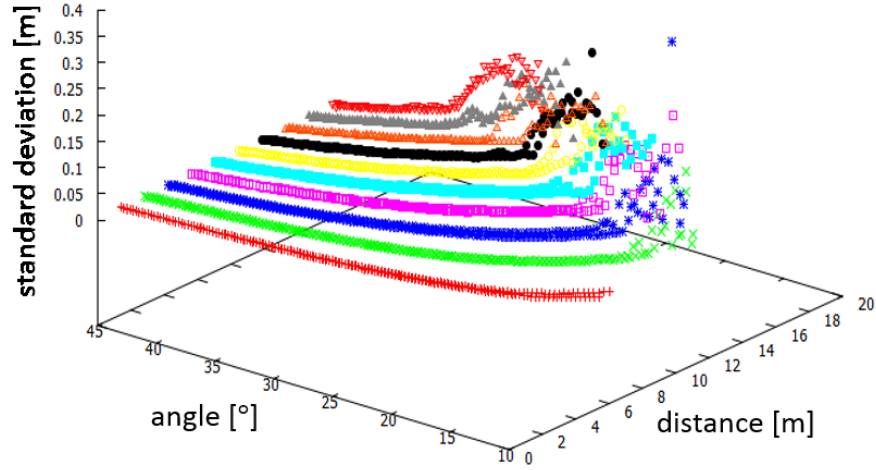


Fig. 2.9 Standard deviations are calculated by using the results of the experiment. Then they are plotted in three dimensions.

がある．また，このように標準偏差が大きくなるだけでなく，計測範囲を超えた，もしくは計測できなかったため，1000 回の計測中値が返ってこない場合も見られた．これは計測精度が距離となす角度に影響されていることを示している．Fig. 2.9 の全ての点から近似曲面を求めると，信頼度の低いデータも使用されてしまうことになる．そのため，標準誤差が大きすぎる距離，角度条件では正しく計測ができないものとして，使用しないこととする．距離データが返ってこなくなる時，標準偏差 σ が $0.035[\text{m}]$ より大きくなるという傾向が見られたため， σ が $0.035[\text{m}]$ 以下になるように条件式を以下のように決定した．

$$\theta = 4.0l - 27.0 \quad (2.33)$$

σ が $0.035[\text{m}]$ 以上となる点を消去し，計測距離 l ，なす角 θ 平面上にこの式をデータプロットとともに表示すると Fig. 2.10 となる．

条件式を適用した点から平均，分散，標準偏差を求める．この標準偏差を LRF の持つ偶然誤差 (σ) として扱う．求めた標準偏差，計測距離 l ，なす角 θ を 3 次元にプロットし，3 次

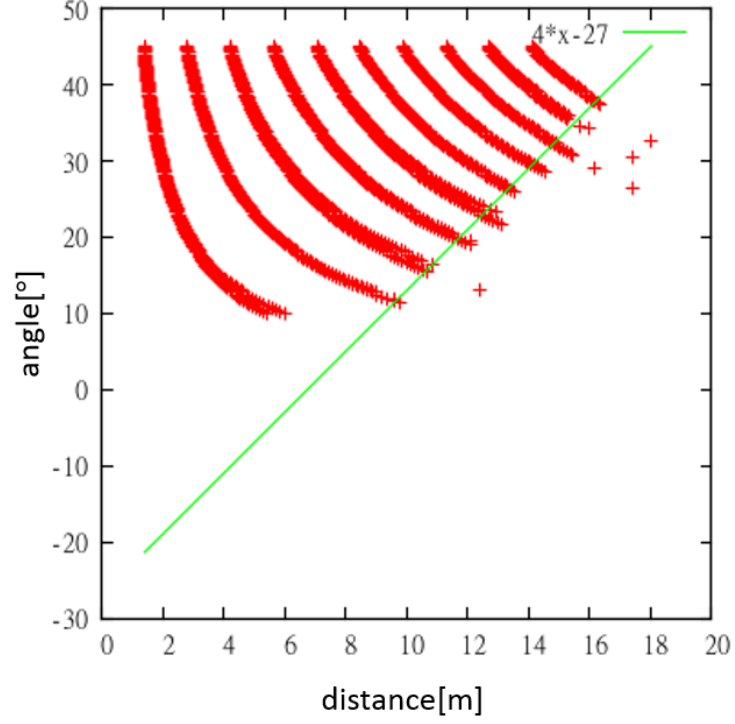


Fig. 2.10 This figure shows l - θ plane of Fig. 2.9.

曲面を最小二乗法によってフィッティングすることで以下の式を得る．

$$\begin{aligned}
 \sigma = & 8.807555 \times 10^{-3} + 1.560033l_i \times 10^{-3} - 6.188378\theta_i \times 10^{-4} + 7.683863l_i^2 \times 10^{-5} \\
 & - 7.555609l_i\theta_i \times 10^{-5} + 2.862907\theta_i^2 \times 10^{-5} + 2.206010l_i^3 \times 10^{-6} - 1.969974l_i^2\theta_i \times 10^{-6} \\
 & + 1.212346l_i\theta_i^2 \times 10^{-6} - 3.898655\theta_i^3 \times 10^{-7}
 \end{aligned} \tag{2.34}$$

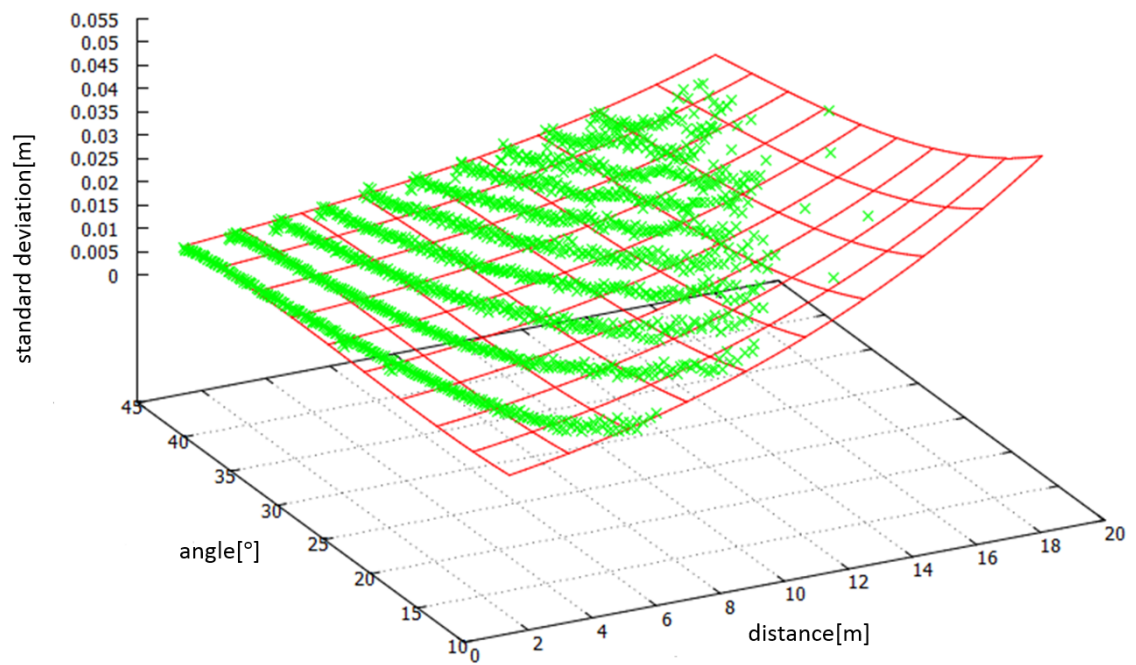


Fig. 2.11 This figure shows the approximate curved surface based on the standard deviation obtained from the experiment.

2.4 ずれ推定楕円と実験のずれ比較

実験は 1000 回 L 字角をスキャンしているため、1000 個の Δr が得られる。この 1000 点がずれ推定楕円に 98.9% 近く含まれることが確認できれば、実験から得られたずれとずれ推定楕円が一致していると言える。

2.4.1 実験で取得した誤差の処理

小節 2.2.6 で導出した式 (2.30) から、実験で得られた誤差を使用してずれを算出することができる。誤差には系統誤差と偶然誤差があるが、実験で得られた距離データの平均値を使用することでキャンセルした。式 (2.30) の偶然誤差は標準正規分布に従う。実験で得られた誤差は正規分布に従うはずであるが、標準正規分布に従うように正規化が必要がある。正規化には標準偏差と平均が必要となるため、小節 2.3.3 で得られた標準偏差 σ を使って正規化し、ずれを算出する。

2.4.2 データ点を間引く前の比較結果

0.25° ずつ取得した全ての距離データを使用して比較した場合の比較結果である。 $x=1.0[\text{m}]$ でモデルから算出されたずれ推定楕円内に含まれる Δr は、63.2% であった (Fig. 2.12)。

$x=2.0[\text{m}]$ でモデルから算出されたずれ推定楕円内に含まれる Δr は、69.7% であった (Fig. 2.13)。

$x=3.0[\text{m}]$ でモデルから算出されたずれ推定楕円内に含まれる Δr は、79.2% であった (Fig. 2.14)。

$x=4.0[\text{m}]$ でモデルから算出されたずれ推定楕円内に含まれる Δr は、84.3% であった (Fig. 2.15)。

$x=5.0[\text{m}]$ でモデルから算出されたずれ推定楕円内に含まれる Δr は、87.4% であった (Fig. 2.16)。

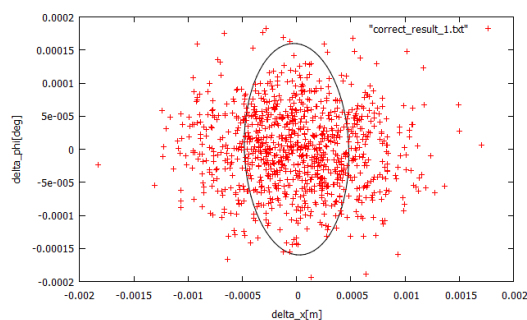


Fig. 2.12 Ellipse obtained at 1.0[m] before thinning out the data

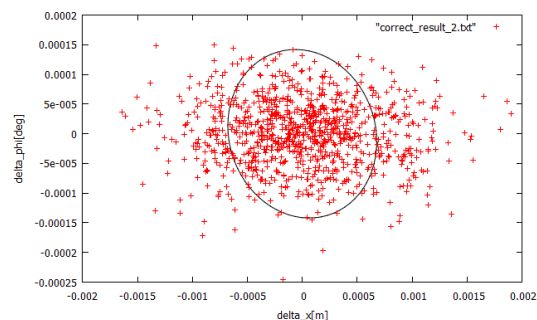


Fig. 2.13 Ellipse obtained at 2.0[m] before thinning out the data

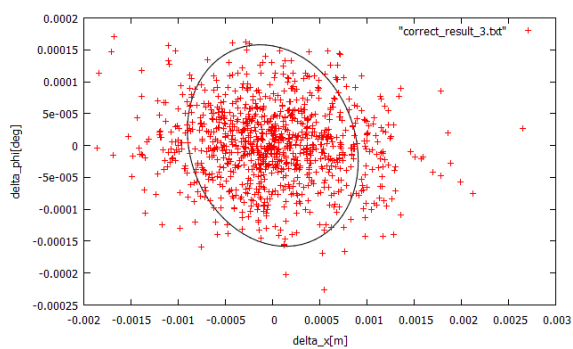


Fig. 2.14 Ellipse obtained at 3.0[m] before thinning out the data

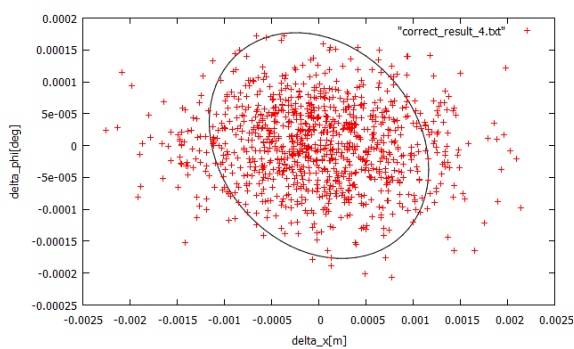


Fig. 2.15 Ellipse obtained at 4.0[m] before thinning out the data

$x=6.0$ [m] でモデルから算出されたずれ推定楕円内に含まれる Δr は , 91.5% であった (Fig. 2.17) .

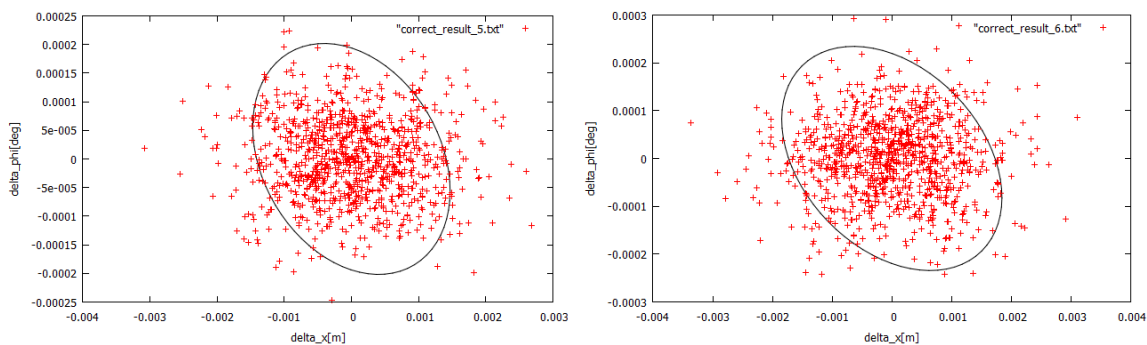


Fig. 2.16 Ellipse obtained at 5.0[m] before thinning out the data **Fig. 2.17** Ellipse obtained at 6.0[m] before thinning out the data

$x=7.0$ [m] でモデルから算出されたずれ推定楕円内に含まれる Δr は , 94.6% であった (Fig. 2.18) .

$x=8.0$ [m] でモデルから算出されたずれ推定楕円内に含まれる Δr は , 95.9% であった (Fig. 2.19) .

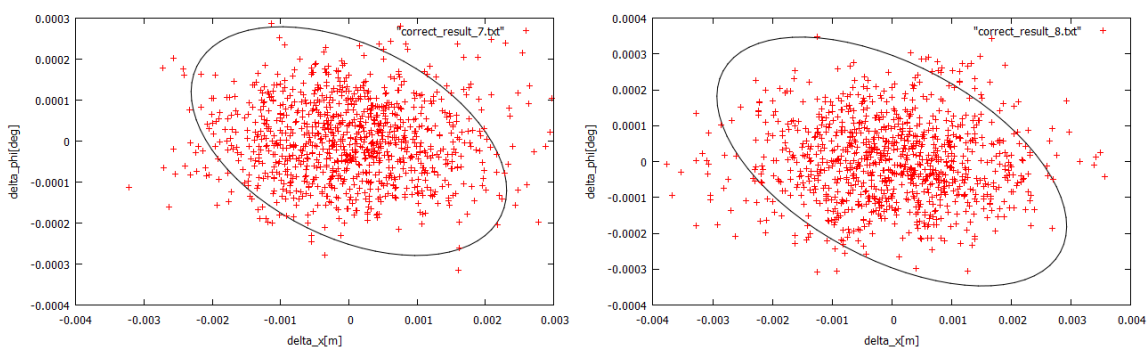


Fig. 2.18 Ellipse obtained at 7.0[m] before thinning out the data **Fig. 2.19** Ellipse obtained at 8.0[m] before thinning out the data

$x=9.0$ [m] でモデルから算出されたずれ推定楕円内に含まれる Δr は , 96.5% であった (Fig.

2.20) .

$x=10.0[\text{m}]$ でモデルから算出されたずれ推定楕円内に含まれる Δr は , 97.5% であった (Fig. 2.21) .

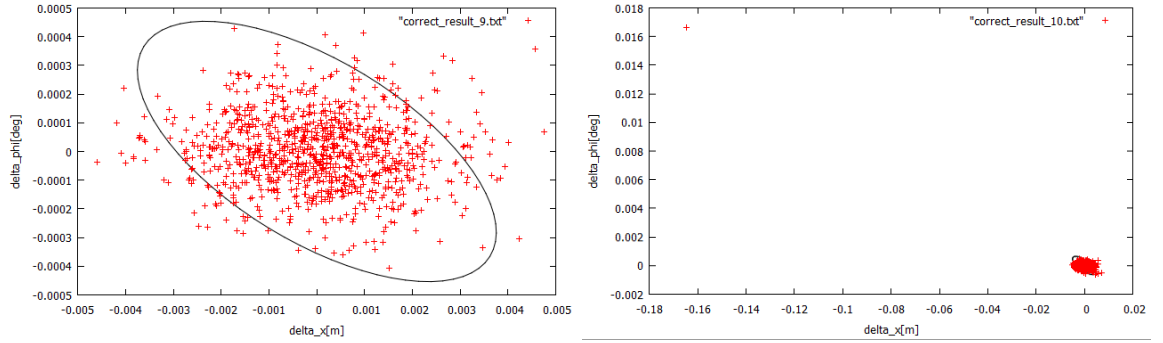


Fig. 2.20 Ellipse obtained at 9.0[m] before thinning out the data **Fig. 2.21** Ellipse obtained at 10.0[m] before thinning out the data

以上の結果から , L 字角に近い位置でのずれ推定楕円と実験から算出したずれが一致していないことが分かった . この理由として , レーザー 1 本を回転させて 0.25° ずつという非常に細かくデータ取得しているため , 求めた偶然誤差が独立したものとなっていなかったことが考えられる . 実際に共分散行列を求めると , 理想的な偶然誤差では対角成分以外は 0 となるのに対し , 実験から得られたデータでは対角成分以外にも大きな値が存在している状態であったことが確認できた .

この解消のため , 距離データを 6 点に 1 点だけ使用することとする . こうすることで対角成分以外はほぼ 0 となるデータとして扱うことができた .

2.4.3 データ点を間引いた後の比較結果

上記問題を解決するためにデータを間引いて同様の比較をした結果を以下に示す .

$x=1.0[\text{m}]$ でのモデルから計算したずれ推定楕円式は以下となる .

$$1 = (6.4793\Delta x^2 \times 10^6 - 2.2046\Delta x\Delta\varphi \times 10^5 + 5.9841\Delta\varphi^2 \times 10^7)/9 \quad (2.35)$$

このずれ推定楕円内に含まれる Δr は , 89.7% であった (Fig. 2.22) .

$x=2.0$ [m] でのモデルから計算したずれ推定楕円式は以下となる .

$$1 = (3.3129\Delta x^2 \times 10^6 - 3.0732\Delta x\Delta\varphi \times 10^5 + 7.9603\Delta\varphi^2 \times 10^7)/9 \quad (2.36)$$

このずれ推定楕円内に含まれる Δr は , 94.4% であった (Fig. 2.23) .

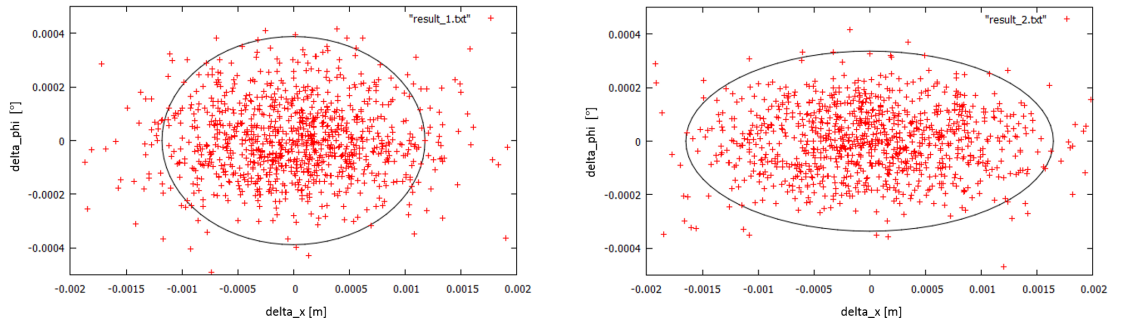


Fig. 2.22 Ellipse obtained at 1.0[m] after thinning out the data **Fig. 2.23** Ellipse obtained at 2.0[m] after thinning out the data

$x=3.0$ [m] でのモデルから計算したずれ推定楕円式は以下となる .

$$1 = (1.8432\Delta x^2 \times 10^6 - 3.3642\Delta x\Delta\varphi \times 10^5 + 6.3265\Delta\varphi^2 \times 10^7)/9 \quad (2.37)$$

このずれ推定楕円内に含まれる Δr は , 94.6% であった (Fig. 2.24) .

$x=4.0$ [m] でのモデルから計算したずれ推定楕円式は以下となる .

$$1 = (1.0963\Delta x^2 \times 10^6 - 3.3532\Delta x\Delta\varphi \times 10^5 + 4.8739\Delta\varphi^2 \times 10^7)/9 \quad (2.38)$$

このずれ推定楕円内に含まれる Δr は , 96.6% であった (Fig. 2.25) .

$x=5.0$ [m] でのモデルから計算したずれ推定楕円式は以下となる .

$$1 = (6.9659\Delta x^2 \times 10^5 - 3.1680\Delta x\Delta\varphi \times 10^5 + 4.0256\Delta\varphi^2 \times 10^7)/9 \quad (2.39)$$

このずれ推定楕円内に含まれる Δr は , 96.9% であった (Fig. 2.26) .

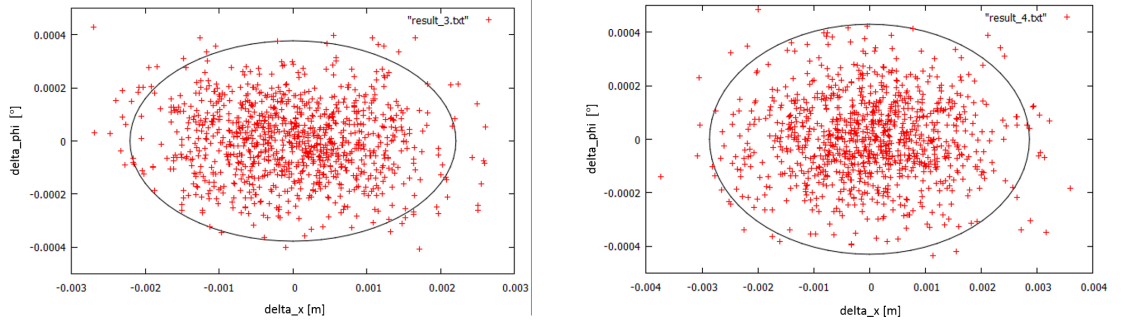


Fig. 2.24 Ellipse obtained at 3.0[m] after thinning out the data **Fig. 2.25** Ellipse obtained at 4.0[m] after thinning out the data

$x=6.0$ [m] でのモデルから計算したずれ推定楕円式は以下となる .

$$1 = (4.4658\Delta x^2 \times 10^5 - 2.8836\Delta x\Delta\varphi \times 10^5 + 3.1960\Delta\varphi^2 \times 10^7)/9 \quad (2.40)$$

このずれ推定楕円内に含まれる Δr は , 96.6% であった (Fig. 2.27) .

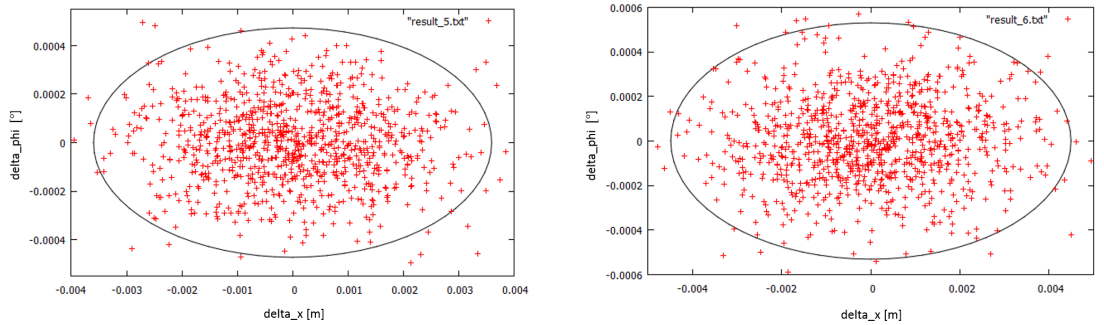


Fig. 2.26 Ellipse obtained at 5.0[m] after thinning out the data **Fig. 2.27** Ellipse obtained at 6.0[m] after thinning out the data

$x=7.0$ [m] でのモデルから計算したずれ推定楕円式は以下となる .

$$1 = (2.8487\Delta x^2 \times 10^5 - 2.5502\Delta x\Delta\varphi \times 10^5 + 2.4374\Delta\varphi^2 \times 10^7)/9 \quad (2.41)$$

このずれ推定楕円内に含まれる Δr は , 98.3% であった (Fig. 2.28) .

$x=8.0$ [m] でのモデルから計算したずれ推定楕円式は以下となる .

$$1 = (1.7818\Delta x^2 \times 10^5 - 2.2044\Delta x\Delta\varphi \times 10^5 + 1.7727\Delta\varphi^2 \times 10^7)/9 \quad (2.42)$$

このずれ推定楕円内に含まれる Δr は , 99.1% であった (Fig. 2.29) .

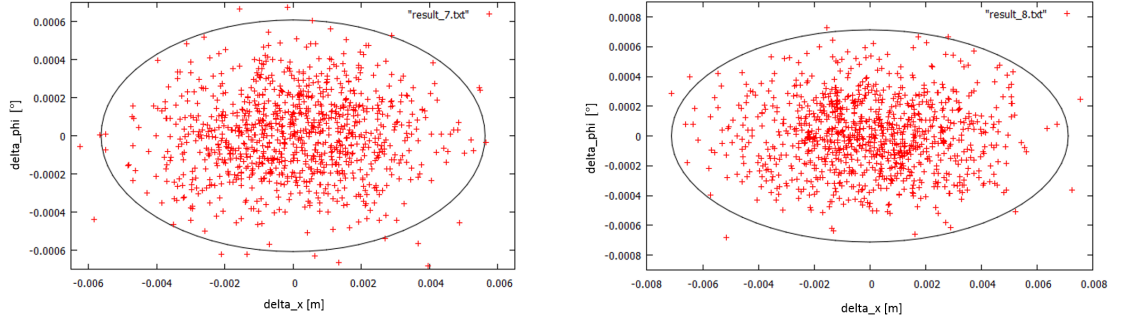


Fig. 2.28 Ellipse obtained at 7.0[m] after thinning out the data **Fig. 2.29** Ellipse obtained at 8.0[m] after thinning out the data

$x=9.0$ [m] でのモデルから計算したずれ推定楕円式は以下となる .

$$1 = (1.0695\Delta x^2 \times 10^5 - 1.8712\Delta x\Delta\varphi \times 10^4 + 1.2083\Delta\varphi^2 \times 10^7)/9 \quad (2.43)$$

このずれ推定楕円内に含まれる Δr は , 98.6% であった (Fig. 2.30) .

$x=10.0$ [m] でのモデルから計算したずれ推定楕円式は以下となる .

$$1 = (5.9011\Delta x^2 \times 10^4 - 1.5660\Delta x\Delta\varphi \times 10^4 + 7.4140\Delta\varphi^2 \times 10^6)/9 \quad (2.44)$$

このずれ推定楕円内に含まれる Δr は , 98.4% であった (Fig. 2.31) .

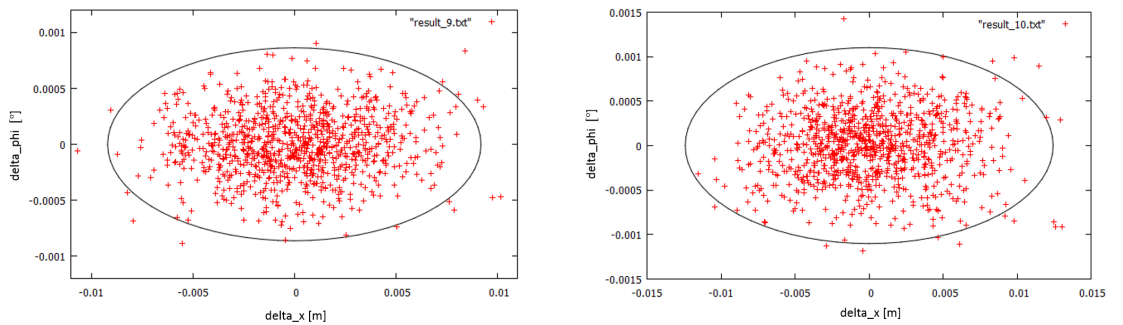


Fig. 2.30 Ellipse obtained at 9.0[m] after thinning out the data **Fig. 2.31** Ellipse obtained at 10.0[m] after thinning out the data

間引いた結果，ずれ推定楕円内部に実験から求めた Δr の 98.9% 近くが含まれていることを，1.0[m] ~ 10.0[m] の 1.0[m] ごとにおいて確認できた．このことから，このずれ推定楕円を使用して計測していない地点でのずれも推定可能であると言える．このずれ推定楕円はロボットのずれの分布を示している．今回使用した LRF のレーザー 1 本 1 本の持つ誤差は節 3.5 で求めた標準偏差であり，その値は 0.035[m] が上限となっていた．このような値の偶然誤差を持っているデータでも，この LRF のように高密度なデータ取得によってロボットの位置姿勢ずれを非常に小さいものに抑えることができるということが分かった．

2.5 ずれ推定楕円による自己位置推定精度評価

各地点 Δx におけるずれ推定楕円の面積を 1 つの自己位置推定評価値として使用することを考える． Δx が 0.1[m] ~ 12.7[m] までのずれ推定楕円の面積を以下に示す．

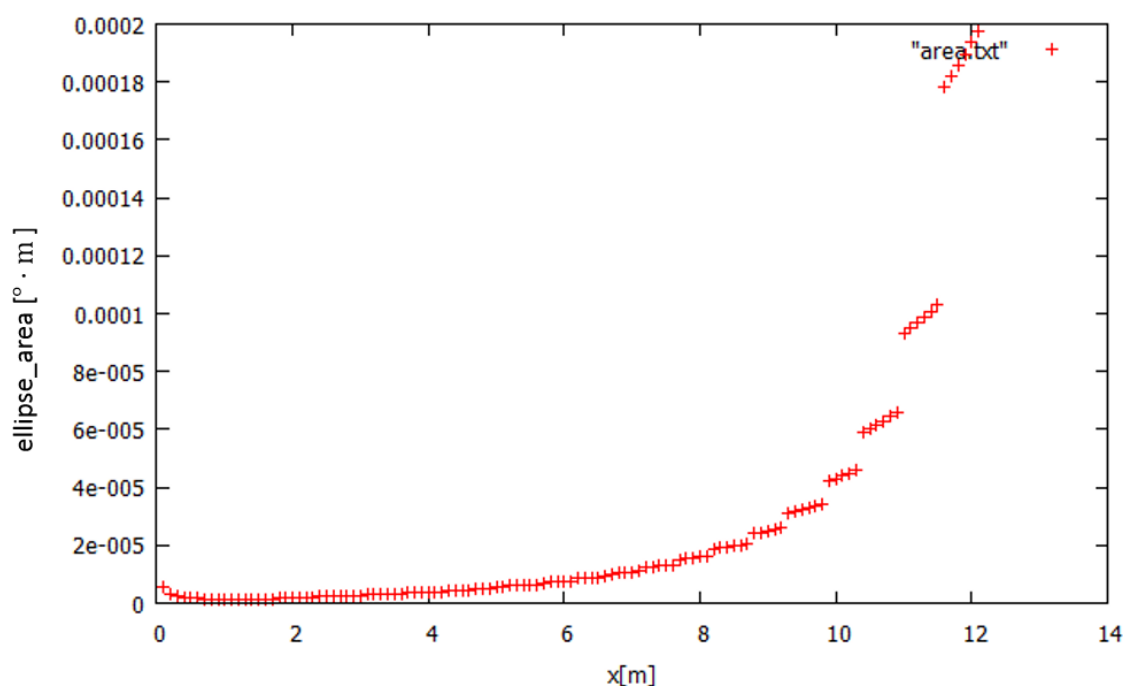


Fig. 2.32 This figure shows the area of slip estimating ellipse.

この面積が小さいほどずれが少なく，データを取得する位置として適切であるということができる．また，今回扱う誤差は偶然誤差であり，計測回数を増加させることで計測精度を上げることが可能である．例えば同じ場所で n 回計測を行った場合，標準偏差は $1/\sqrt{n}$ となり，計測精度は向上する．これはこの推定楕円面積にも適用できるため，例えば次のように定量的に位置推定精度を比較することができる．

例) $\Delta x=3.0[\text{m}]$ における 1 回の自己位置推定精度の評価値は 2.6186×10^{-6} である．一方， $\Delta x=6.0[\text{m}]$ における 2 回の自己位置推定精度の評価値は $7.4896/\sqrt{2} \times 10^{-6} = 5.294 \times 10^{-6}$ となり，L 字角から離れたところから 2 回計測をするよりも L 字角に近い位置で 1 回計測する方がずれが小さく高精度な計測であると言える．

2.6 自己位置推定精度解析のまとめと今後の展望

本提案手法では計測モデルから計測精度を定量的に評価することを目的とし，最終的にずれ推定楕円を導出し，各位置での計測精度を定量的に示すことができた．今回は屋外環境において自律移動する際，頻繁に計測される L 字角に焦点を置いて研究を進めてきた．将来的には，L 字角のみではなく，モデルを変え，幾何学的に変換行列 J を算出すれば同じように精度を解析が可能である．さらに，自律移動ロボットが経路設定をする際精度の良い計測箇所を自ら選択し，より安定した自律移動を可能とすることが期待される．

第3章

マッチング取捨選択

3.1 概要

本章ではLRF から取得した 2 値画像マッチングを用いる自己位置推定において、マッチング結果の正誤を評価する手法について述べる。画像マッチングは OpenCV の `cvMatchTemplate` をベースにしたものを使用しており、第 2 章でも使用した 2-D LRF で取得した距離、角度データから得られる 2 値画像を使用する。この際算出されるマッチング率を評価値とし、環境毎にマッチング正誤の判断のためのマッチング閾値を設定する。本研究では、1 枚の 2 値画像はマッチングの正誤を決定するあるマッチング閾値を 1 つ有していると考えている。LRF で取得した 2 値画像マッチングにおいて、建物の壁などデータ点列が直線に並び、同じパターンの特徴が出現しやすい環境ではマッチング率は高く、点列がばらばらに並び、同じパターンの特徴が出現しにくい生垣や草むらなどでは低く出やすい傾向がある。前者の環境ではマッチングが成功するときのマッチング率は高く、後者では低くなる。1 枚の画像上の環境データ点列を分類し、閾値を与え、その平均値を算出することで画像のマッチング閾値を設定する。この手法において、どのようなデータ点列にどのマッチング閾値を付与するべきかを判断する必要があり、この判断には SVM を使用する。SVM モデルを作成するために

与える訓練サンプルは環境データ点列のばらつき具合とし、画像生成前の距離データを加工したものとする。閾値設定が適切であるかどうかを環境を変えて実験し、検証した。また、自律移動に本提案手法を搭載し、閾値が適切に設定され、安定した自律移動が可能であることを確認した。

3.2 LRF を用いたの自己位置推定

3.2.1 マッチングの基礎技術について

一般的に用いられるマッチング手法には、ICP マッチング法 [20] と、画像マッチング法があり、第 1 章で挙げた自己位置推定の従来手法もこのどちらかの手法をマッチングの基本技術としている。ICP は元データ点と、比較対象となるデータ点の距離が最小となるような点を探索する手法であり、自己位置推定に適用すれば、元データと走行時取得データを比較し、ずれを算出することが可能である。この手法では精度の高いマッチングが可能であるが、データ点 1 つ 1 つの距離を計算することになるため計算量が多い。自律移動においてミリ単位の精度は必要が無い場合、本研究では画像マッチング法を採用した。次小節で画像マッチング手法について述べる。

3.2.2 画像マッチング、マッチング率について

本研究で使用する自律移動ロボット MieC が採用している自己位置推定手法は、LRF から取得された 2 値画像による局所マップマッチングである。このマッチング手法では、アフィン変換による画像の回転、平行移動と画像、動画処理に必要な様々な機能が実装されているオープンソースライブラリである OpenCV の `cvMatchTemplate` を組み合わせて使用している。自律移動の流れとしては、まず自律移動をする前に手動で走行コースの LRF データを取得し、オリジナル画像としてストアしておく。その後、自律移動時にデッドレコニングを頼りにしながら決められた位置で LRF 画像を取得し、走行時取得画像として保存する。こ

のオリジナル画像と走行時取得画像を使ってマッチングする．まずオリジナル画像をアフィン行列を使ってあらかじめ指定しておいた角度分回転させ，中心から指定したピクセル分を抜き出しテンプレートとする．二次元アフィン変換はアフィン変換行列 A を用いて次のように表すことができる．

$$A = \begin{pmatrix} \cos \theta & \sin \theta & t_x \\ -\sin \theta & \cos \theta & t_y \\ 0 & 0 & 1 \end{pmatrix} \quad (3.1)$$

ここで θ は回転角を表し， t_x, t_y は x, y 方向それぞれの平行移動量を表す．点 $(x, y)^t$ を θ 回転させたときの座標を $(x', y')^t$ とすると

$$(x', y')^t = A(x, y)^t \quad (3.2)$$

と計算される．経験としてデッドレコニングが正常であれば機体が大きく回転することはなく，オリジナル画像を回転させる角度は $[-32, -24, -20, -16, -8, -4, 0, 4, 8, 16, 20, 24, 32]$ としている．このテンプレートを使用して画像上でピクセル毎にマッチングする．その中で最もマッチング率の高かった回転角度，距離を補正し，次のマッチング点に向かって走行する手法をとっている．Fig. 3.2 は Fig. 3.1 で取得したオリジナルと走行時取得した 2 値画像の例である．この 2 値画像上では，ロボットの進行方向は画像左方向から右方向となっている．距離，角度データを画像データに変換してストアし，走行中にマッチングすることで位置姿勢を補正している．この手法は ICP と比較して精度は落ちるが，局所的な地図を取得しマッチングするためマッチング回数が少なく処理するデータ量が少ないことが利点である．

各ピクセル毎に計算されるマッチング率は以下の正規化相互相関を採用している．

$$R(x, y) = \frac{\sum \sum T'(x', y') \cdot I'(x + x', y + y')}{\sqrt{\sum \sum T'(x', y')^2 \cdot \sum \sum I'(x + x', y + y')^2}} \quad (3.3)$$

この式において， $R(x, y)$ は (x, y) でのマッチング率， $I'(x, y)$ はオリジナル画像の座標 (x, y) の輝度値， $T'(x', y')$ はテンプレート画像の座標 (x', y') の輝度値を表す．この式の分子部分ではテンプレート画像の各ピクセルの輝度値ベクトルと走行時取得画像の輝度値ベクトルの内



Fig. 3.1 Picture of the landscape LRF data are taken



Fig. 3.2 left:Original bynary image, rigth:Data bynary image

積が計算され、それを正規化したものをマッチング率としている．正規化相互相関はテンプレートや画像の明るさの変動があっても安定的に類似度計算ができるものであり、今までの自律移動ではこのマッチング率を使用することでマッチングに異常が出るなどしたことが無かったため、本研究においてもこのマッチング率を使用する． $R(x,y)$ が 1 に近ければ近いほどオリジナル画像から作られたテンプレート画像が走行時取得画像に合致していることになる．1 ピクセル毎にこの $R(x,y)$ が計算され、この値が最大となる場所の回転、 x,y 方向のずれを基にして位置姿勢の補正している．マッチング率 $R(x,y)$ は環境ごとに異なる値を取り、取得した特徴量が直線として取得されることが多い建物の壁などであれば、輝度値も一致しやすくマッチング率は高い値を取ることが想定される．しかし、人間の足や自転車など、短時間で移動してしまう不安定な物体に対して画像マッチングを行うと、輝度値が一致せず誤

マッチングが発生しやすくなり，それに伴いマッチング率も低くなることが考えられる．不安定物体の他にも，屋外環境では比較的に良く使用される生垣や茂みなどのデータに関しても高いマッチング率は期待できない．

3.3 自己位置推定破綻防止のためのマッチング取捨選択

私が所属する本研究室の環境認識班，人工知能班は 2012-2015 のつくばチャレンジに MieC LM と参加してきた．2014, 2015 では誤マッチングによって自律移動を断念することになってしまった．この失敗から，本提案手法の「マッチング結果の取捨選択」を取得したデータから判断基準を作成して実行するという考えに至った．自己位置推定の際，マッチングの疎，密に関わらず誤マッチングが連続して発生してしまうと自己位置推定の破綻につながる．そのため，マッチングが正確であるかどうかを評価したいという課題が発生することはこれまでに述べた．本研究では評価値としてマッチング率 (Eq. 3.3) を使用する．取得したデータからそのデータを取得した環境で必要となるマッチング率の閾値を SVM を使用して判別するのが本提案手法である．

3.3.1 提案手法: 距離データ点列からマッチング閾値設定

LRF2 値画像において，建物の壁，階段などデータ点列が直線に並び，同じパターンの特徴が出現しやすい環境ではマッチング率は高く，点列がばらばらに並び，同じパターンの特徴が出現しにくい生垣，草むらなどでは低く出やすい傾向がある．そのため前者の環境ではマッチングが成功するときのマッチング率は高く，後者では低くなる．このことから，全環境の画像マッチング率とマッチングの成功率の関係性は Fig. 3.3 のようになると考えられる．

Fig. 3.4 は異なる 3 種類の環境で取得された 3 枚の画像のマッチング正誤を判断するグラフであり，データを取得する場所によってマッチング閾値は変化すると考えられる．あらゆる環境において Fig. 3.4 のグラフが存在すると考えると，全てのグラフを足し合わせて平均

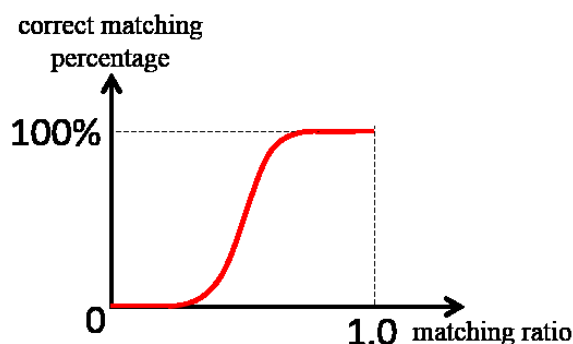


Fig. 3.3 The relationship between matching ratio and correct matching percentage

を取ると Fig. 3.3 が表現できる。

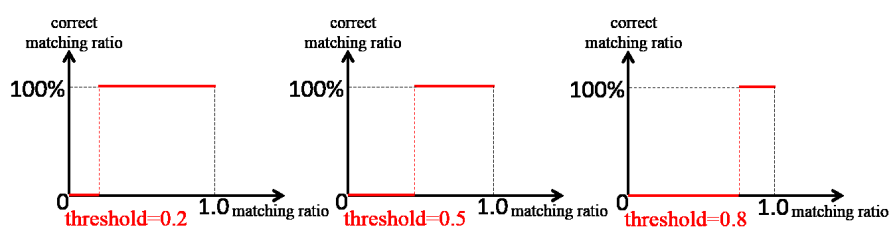


Fig. 3.4 The threshold of matching in each environment

閾値の高い画像が集まる環境がある一方で、低い画像が集まりやすい環境も存在すると考えられ、閾値に偏りが生じることが想定される。例として、屋外は低閾値の画像が集まりやすく、Fig. 3.5(a) のように、屋内は高閾値の画像が集まりやすく、Fig. 3.5(b) のようになると考えられる。この偏りを基に環境に適したマッチング閾値を設定する必要があると考えられる。画像のマッチング閾値は各点列から起因するものであると考え、取得した距離データ点列の連続 5 点を 1 組にまとめる (Fig. 3.6)。この際、環境ごとに点の凹凸具合が異なり、この凹凸具合によってマッチング率変動する考えたため、この 5 点に最小二乗法を適用して生成した直線からの符号付距離を特徴量とし、凹凸具合に合った閾値を与える。先頭点を次の点にずらし、特徴量を作成、閾値の付与を繰り返す。画像のマッチング閾値はこの点

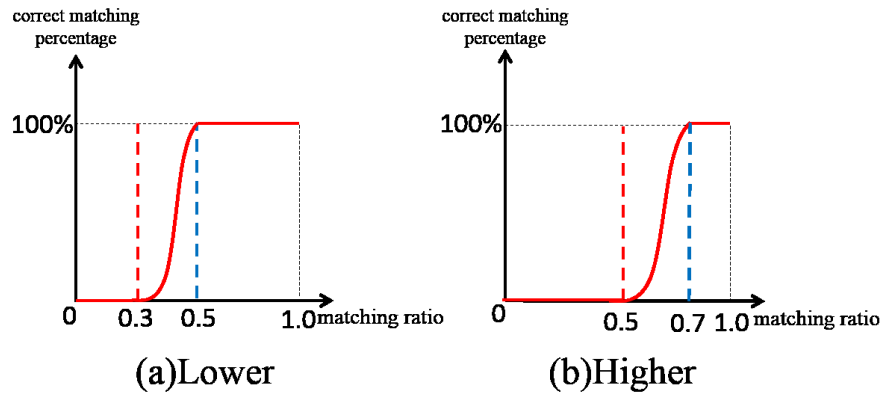


Fig. 3.5 The relationship between matching ratio and correct matching percentage in indoor and outdoor environment

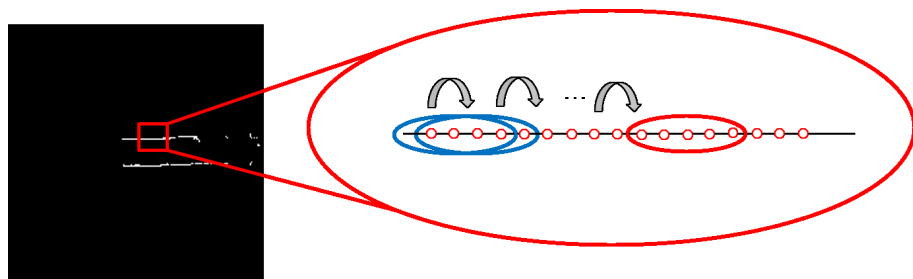


Fig. 3.6 Make a group from 5 data points

列の閾値の平均として考えることとする。

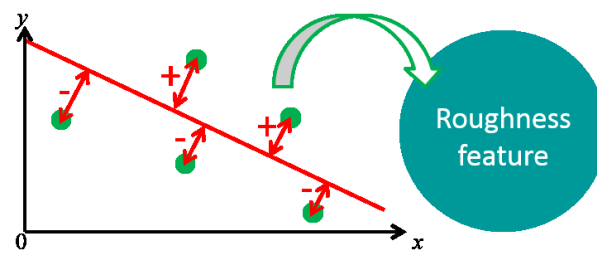


Fig. 3.7 How to make roughness feature

3.3.2 サポートベクターマシン (SVM) を使用した点列の分類

点列の凹凸具合をあらわす特徴量を作成したが、その特徴量の持つ閾値を決定したい。そのためにはどの環境にその凹凸具合が分類されるか判定する必要がある。本研究ではこの凹凸具合をサンプルとし、サンプルがどの環境に分類されるか偏呈する分類器として SVM を使用する [22][23][24]。SVM は教師あり学習を用いる分類器であり、現在知られている多くの手法の中でも最も認識性能の優れる学習モデルの 1 つとして考えられている。SVM は主に 2 クラスを最適に分類する手法であり、教示データをもとに判断面を作成する。SVM は高次元空間上で計算量的に効率よく、分離する超平面を探し出すことを得意とする。

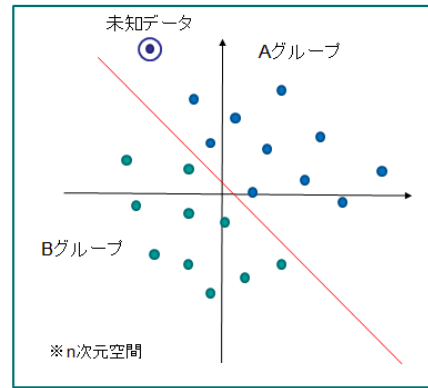


Fig. 3.8 Support Vector Machine

与えられた教示データを非線形写像することにより線形マシンの使用できる特徴空間へと変化させる。非線形に写像した要素を $\phi(\mathbf{x})$ とすると、

$$f(\mathbf{x}) = \sum_{i=1}^N \omega_i \phi_i(\mathbf{x}) + \mathbf{b} \quad (3.4)$$

線形マシンは双対表現で表現できるため [23]、決定規則は、テスト点とトレーニング点と単なる内積を用いて評価できる。

$$f(\mathbf{x}) = \sum_{i=1}^l \alpha_i y_i \langle \phi(\mathbf{x}_i) \cdot \phi(\mathbf{x}) \rangle + \mathbf{b} \quad (3.5)$$

α は双対変数, y は出力である．特にカーネルトリックと呼ばれる手法を用いることにより写像後の特徴空間での内積を計算することなく導き出すことが可能である．

$$K(\mathbf{x}, \mathbf{z}) = \langle \phi(\mathbf{x}) \cdot \phi(\mathbf{z}) \rangle \quad (3.6)$$

本手法では汎用性がたかく, 事前に分離知識を必要とせず, 広い分野にわたって使用されているラジアル基底関数カーネル (RBF カーネル) を用いることとする．

$$K(\mathbf{x}, \mathbf{z}) = \exp(-\gamma \|\mathbf{x} - \mathbf{x}'\|^2) \quad (3.7)$$

今回使用する SVM は, libsvm という SVM である．この SVM は, SVM の専門家でなくとも SVM を使用できるようにしたい, 使用してほしいという考えから作成された SVM のオープンソースライブラリである [25]．この libsvm は svm モデルの作成, 分類プログラムはもちろんのこと, 最適な分類をするためのパラメータ調整をしてくれるグリッドサーチプログラムも備えており, 非常に扱いやすいものとなっている．訓練サンプルとしては小節 3.3.1 で LRF データから作成した, 凹凸具合を表現する 5 次元のベクトルを与える．グリッドサーチプログラムによって最適な 2 パラメータ (コスト C (誤分類の許容値), γ (非線形性)) を設定した上でモデルを作成する．

3.4 事前実験: マッチング率の差異の検証

本提案手法は環境毎にマッチングの閾値が異なるという前提で進めてきたが, 実際に環境によってマッチング率に差異があるのかどうか実験して確認した．実験環境は屋内環境として三重大学機械創成棟 2F の廊下, 屋外環境として三重大学三翠ホール前のモニュメント付近遊歩道, 安定した特徴ではない駐輪場に駐車してある自転車の 3 環境である．3 環境とも自律移動ロボットが走行する際に頻繁に現れる特徴点が取得される環境であり, 屋内では壁が, 屋外では生垣が, 駐輪場では多くの自転車が周囲に存在する環境となっている．各環境下でオリジナル画像を最初に取得し, マッチング用画像は Fig. 3.13 の順で Fig. 3.12 の実

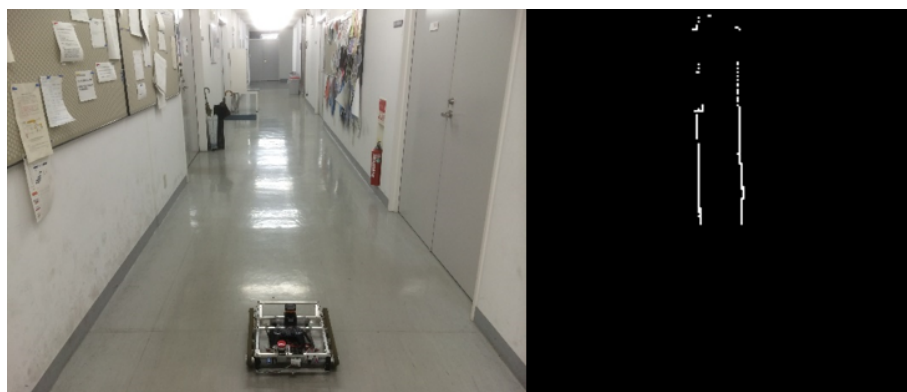


Fig. 3.9 Picture and binary image of indoor environment



Fig. 3.10 Picture and binary image of outdoor environment

験風景のように，位置を MieC LM 1 台分，各地点で角度を ± 30 度回転させて設置して取得する．オリジナル画像とマッチング用画像をマッチングし，正しく補正値が算出できているかどうか実験者が判断し，可能であった場合のマッチング率，失敗した時のマッチング率から，最低限必要であると考えられるマッチング閾値を設定した．マッチングが正しいかどうかの判断については，自己位置推定の際最も大きく影響を及ぼす角度補正が正しいか確認した後，位置補正が MieC 1 台分までは許容するものとした．屋内環境，屋外環境，駐輪場付近で取得された LRF によるオリジナル 2 値画像はそれぞれ Fig. 3.9，Fig. 3.10，Fig. 3.11 である．この実験から環境によってマッチングの閾値に差異があることが分かつとともに，

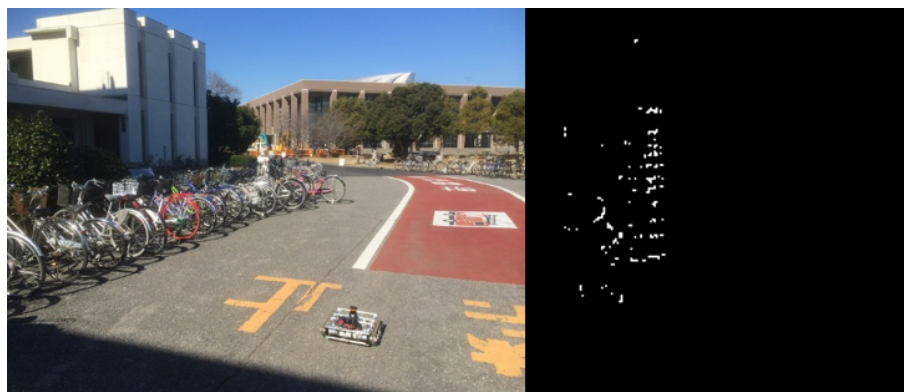


Fig. 3.11 Picture and binary image of bicycle

各環境における点列の凹凸具合に与えるマッチング閾値は屋内環境 0.56，屋外環境 0.40，駐輪場 0.22 と設定することができた．



Fig. 3.12 Example of experiment

3.5 SVM による判別閾値設定実験

事前実験で取得した 3 環境それぞれのオリジナル LRF データから，各環境における点列の凹凸具合の訓練サンプルを作成し，屋内，屋外，自転車の 3 クラス分類器を作成する．本研究では 3 クラスのみの分類であるため，1 対他分類を 3 回行い，屋内のみ，屋外のみ，自



Fig. 3.13 The order and position of experiment

転車のみに分類する．未知データが取得された際，未知データの点列の凹凸具合ベクトルを作成し，屋内，屋外，自転車のいずれかに分類された場合のみ閾値を与え，それ以外のベクトルは無視するものとする．最終的に未知データから生成される画像の閾値は各点列の凹凸具合ベクトルの持つ閾値の平均で設定される．

実際に適切な閾値設定が可能か否か検証した．屋内 (Fig. 3.9 の創成棟 2F 廊下 (Passage1))，廊下 2 (Fig. 3.14 の階段付近 (Passage2))，屋外 (Fig. 3.10 の三翠ホール前広場の遊歩道 (Outdoor walk))，Fig. 3.11 の学務前駐輪場 (Bicycle parking) の 4 地点においてデータを取得し，適切な閾値設定が可能であるか確認する．類似環境も含まれるが，環境の形状が異なる環境も含まれており，適切に閾値設定できるかどうかの検証には十分であると考える．各環境で Fig. 3.13 の順にデータを取得する．各地点において角度ずれが無い場合， $\pm 30^\circ$ 回転させた場合の 26 回データを取得し，設定された閾値の平均，適切に閾値が設定された確率を Tab. 3.1 に示す．

廊下 1 のみ適切なマッチング閾値が設定できているかどうか低い値が出ている．この原因としては屋内環境においてテンプレートマッチングをした際に，Fig. 3.15 に示すテンプレートとなり，テンプレートマッチングにおいてテンプレートを走行時取得データに合致するように角度を合わせた後は，両側の壁が写ったデータのみとなるので，平行移動距離が



Fig. 3.14 The landscape of experiment near the stairs

Table 3.1 The result of matching experiment

	Average threshold	Percentage of correct threshold
Passage1	0.532	81.8
Passage2	0.527	92.3
Outdoor walk	0.420	97.7
Bicycle parking	0.302	92.3

誤ってしまってもマッチング率が高く出てしまうためであると考えられる．この問題については本手法で閾値を設定することは困難となるが，廊下等の直線特徴量が出続ける環境を走行する場合，角度補正のみ実行し，直線距離を補正できる箇所までデッドレコニングのみで走行をすることが十分に可能である．各点列の凹凸具合に与えられる閾値は屋内 0.56，屋外 0.40，自転車 0.22 であった．この実験から屋内，屋外，自転車共に各環境の適切な閾値が設定できているのが確認できた．

次に，自律移動に本手法を適用した．走行アルゴリズムを Fig. 3.16 の NS チャートに示す．本提案手法を搭載した自律移動において，適切な閾値設定によるマッチング結果の取捨

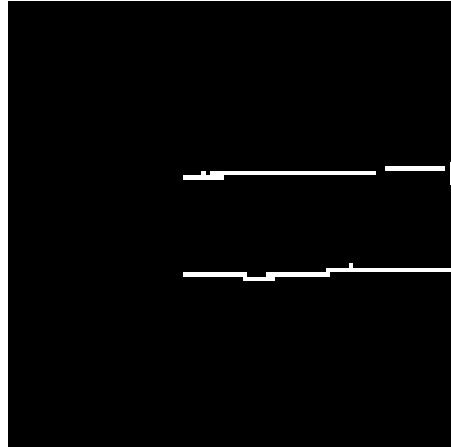


Fig. 3.15 The template of indoor passage

選択をしながら安定した自律移動が可能であることを確認することができた．走行コースは三重大学工学部学務前から構内のミニストップまでである．

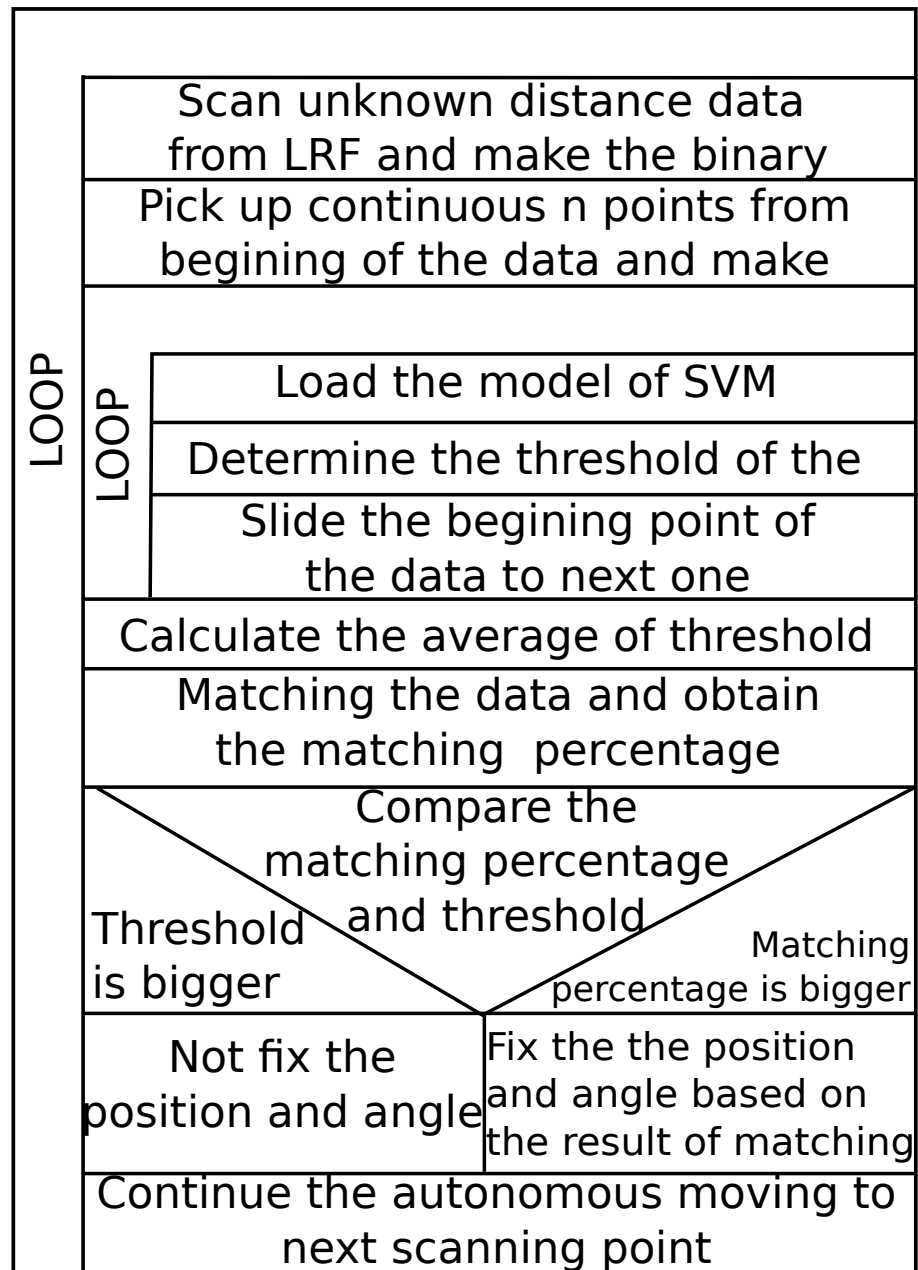


Fig. 3.16 NS chart of autonomous moving applied proposed method

3.6 マッチング取捨選択のまとめと今後の展望

本提案手法は、環境毎にマッチング閾値を設定するものであり、ある画像のマッチング閾値は LRF で取得した環境の点列の凹凸具合がもつ閾値の平均で表す。この考え方が正しいかどうか検証した上で、ある環境のマッチング閾値を設定できたことを示し、さらに自律移動プログラムに搭載することで安定した自律移動を実現可能であることを示した。

今後の展望として、提案手法を自律移動に適用して実験した回数が多くないため、より多様な環境に対応できるようにデータ収集と実際に走行実験することが望まれる。自律移動時にデータを取得し、自律走行時とは別の機会に SVM に学習させることが可能であるため、もし未知のデータが収集された場合でも新しく学習させてより多種多様な環境での閾値設定が期待できる。また、本提案手法のマッチング閾値を可変とすることについては画像マッチングだけでなく ICP による自己位置推定にも適用できるものであるため、自己位置推定の評価に使用できると考える。さらに現在自律移動手法の主流であるパーティクルフィルタのリサンプリング時にこの閾値設定を適用すれば、より正確に候補パーティクルを残すことが可能であると考えられる。

第4章

結びに

最終成果と今後の展望

本研究では、マッチング前とマッチング後にデータを評価する手法を提案し、実際の有用性を実験において示した。自己位置推定精度解析においては対象物体のどの位置でデータ取得すべきか定量的に示すことができ、マッチング取捨選択においてはマッチング閾値を特徴量の持つ閾値の平均とすることで、未知データの特徴量を分類し、閾値設定をすることを可能とした。

自己位置推定精度の解析は、ずれの少ない計測点を計算できるようになったため、現在は人の手によって経験則で作成されているグローバルマップをロボット自身が自動で生成する第一歩となりうると考えている。本研究ではL字角に限定された精度解析となっているが、モデルを変えることで計算手法を変えることなくL字角以外の物体に対する精度解析も可能である。人間がまず走りたいコースを手動で走らせることは変わらないが、どこでマッチングをするかをロボットが判断できるようになることが期待できるということである。

マッチング結果の取捨選択においては、最も期待される適用先としてパーティクルフィルタへの適用が考えられる。パーティクルフィルタのリサンプリング時のマッチングに本提案

手法の閾値を設定することで、マッチング結果が正しくないのに生き残っていたパーティクルの削除が期待できる。この閾値設定という考え方については画像マッチングのみではなく ICP 法にも応用できるため、マッチングの精度向上に非常に有用であると考えている。

両提案手法ともに自律移動に搭載可能な手法であり、今後様々な環境での自律移動に使用されることが期待される。

付録

1. SVM による閾値設定法と訓練サンプルの設定法

訓練サンプルを最終的に凹凸具合決定した流れを示す。

1.1 2 値画像の輝度値データを使用

まず初めに 2 値画像データの輝度値を訓練サンプルとして使用する手法を考える。MieC で使用する 1 枚の画像は 200×200 の 40000 ピクセルからなり、一枚の画像からこの輝度値ベクトルの訓練サンプルが入手できる。自律移動の際、マッチング閾値と輝度値データをストアしておき、そのデータに基づいてオフラインで SVM に学習させ、判別器を作る。別の自律移動の際、その SVM を用いて未知環境で取得した輝度値データがどれほどの閾値を必要とするのか判別する手法である。しかし、この手法ではベクトルは 40000 次元をもつことになるため、判断面を引くために相当数のサンプルが必要となり、モデル作成が困難である。

$$L = \begin{pmatrix} L_1 \\ \vdots \\ n_i \\ \vdots \\ L_{40000} \end{pmatrix} \quad (4.1)$$

1.2 距離データを無加工のまま使用

1枚の画像から多くの訓練サンプルを取得できればそれだけモデル作成が容易になり、また正確な判断面を作成できる。そこで、輝度値データではなく LRF で取得した距離データをそのまま使用することを第2に考える。LRF によって取得可能な距離データは後方 90° を除く 270° でステップ角が 0.25° であるので、取得データ数は一回のスキャンで 1081 点である。このデータ点を 5 点 1 組としてグループ化し、5 次元のベクトルとして訓練サンプルを作成する。先頭をを 1 点ずつずらして 5 次元ベクトルを生成することで画像一枚からベクトルを多数取得することが可能となる (Fig. 3.6)。

この訓練サンプルを使用して SVM に学習させてモデルを作成、未知データも同様に距離の 5 次元ベクトル特徴量として、判別器にかけることで特徴量がどのマッチング閾値を形成するものか判別する。この手法を用いて屋外、屋内環境の分類と実験した結果、分類できていることが確認できた。しかし、この訓練サンプルは適用範囲が狭いため、次の手法を考える。

1.3 取得距離データから算出した凹凸データを使用

距離データをそのまま使用する手法では距離が同程度のデータでなければ判別ができないことが考えられる。例として、同じ屋内環境であっても廊下で取得したデータから作成した SVM は居室では使用できない。そのため、特徴量として取得距離データから算出した凹凸データを訓練サンプル、判別するデータとして使用する手法を考案

する．具体的には，1.2 で考案した手法と同様に 5 点を 1 組とする．その 5 点から最小二乗法により近似直線を求め，点と直線の距離から符号付の距離データを算出，これを訓練サンプルとして使用する．これにより距離が及ぼす影響は小さくなり，適用範囲を狭くしてしまっていた原因を取り除くことができ，点列の持つ凹凸具合で特徴量を分類できる．

2. 2 値分類実験 3 値分類実験

3 値分類に辿り着くまでの流れを示す．まずは屋外特徴と屋内特徴の分類が可能かどうかを確認した．この実験は試験的に実施したものであるため，自転車については考えていない．Fig. 3.9, Fig. 3.10 を取得した際の距離データから，1.2 で提案した訓練サンプルを作成する手法と，1.3 で提案した訓練サンプルを作成する手法を使用し，SVM に学習させ，モデルを作成した．その後 Fig. 4.1, Fig. 4.2 に示す元画像の類似環境の判別閾値設定を行った．屋内の訓練サンプルの持つ閾値は事前実験で決定された 0.56，屋外のものは 0.40 とし，それぞれ 1, -1 とラベル付をして判別した．閾値は特徴量の持つ閾値の平均とする．距離データを直接使用した場合，Fig. 4.1 の特徴量は 1061 点中ラベル 1 の点が 774 個，-1 の点が 287 個となり，閾値は 0.52 と算出された．また Fig. 4.2 に関しては 839 点中ラベル 1 の点が 108 個，-1 の点が 731 個となり，閾値は 0.42 となった．いずれも類似の未知データではあるが判別できていることが確認できる．取得距離データから算出した凹凸データの場合，特徴量は 1061 点中ラベル 1 の点が 918 個，-1 の点が 143 個となり，閾値は 0.54 と算出された．また Fig. 4.2 に関しては 839 点中ラベル 1 の点が 190 個，-1 の点が 649 個となり，閾値は 0.44 となった．こちらも環境に合った閾値を設定できていると言える．

走行実験をした結果，閾値よりもはるかに低いマッチング率が計算されており，全く閾値が機能しなかった．そのため，自転車をクラスに追加し，3 クラス分類とした．多クラス分類に関しては，`libsvm` に標準搭載されているが，C のプログラムソースが無かった (python ではあった) ため，提案手法の方法で 3 クラス分類をした．

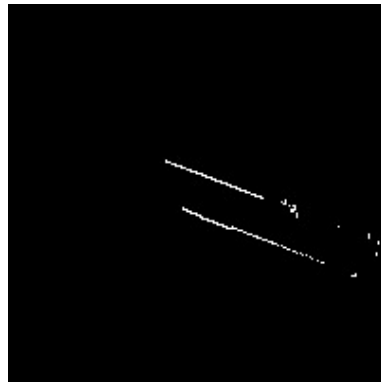


Fig. 4.1 Experiment data of indoor environment



Fig. 4.2 Experiment data of outdoor environment

参考文献

- [1] 酒井 龍雄, 村井 亮介, 上松 弘幸, 木下 慎太郎, 北野 斉, 自律移動ロボットを用いた搬送システム, パナソニック技報 9月号 2009 Vol.57 No.3,2009
- [2] 内閣府 政策統括官 (科学技術イノベーション担当) SIP(戦略的イノベーション創造プログラム) 自動走行システム研究開発計画 2015/5/21
- [3] National Highway Traffic Safety Administration Preliminary Statement of Policy Concerning Automated Vehicles
- [4] 油田 信一, 岩田 光弘, つくばチャレンジ 2015:市街地における移動ロボットの自律走行の公開実験 -2015 の課題と結果・成果-, 第 16 回システムインテグレーション部門講演会 (SI2015) 1K1_1
- [5] 総務省 平成 27 年度版情報通信白書
- [6] 総務省情報通信局国際戦略情報通信政策課情報通信経済室 社会課題解決のための新たな ICT サービス・技術への人々の意識に関する調査研究 -報告書- 平成 27 年
- [7] John J. Leonard, Hugh F. Durrant-White, Simultaneous Map Building and Localization for an Autonomous Mobile Robot : Intelligent Robots and Systems '91. 'In-

- telligence for Mechanical Systems, Proceedings IROS '91. IEEE/RSJ International Workshop p.1442-1447
- [8] Michael Montemerlo, Sebastian Thrun, Daphne Koller, Ben Wegbreit, FastSLAM: A Factored Solution to the Simultaneous Localization and Mapping Problem, Proceedings of the AAAI National Conference on Artificial Intelligence, 2002
- [9] 小川 陽子, 白井 良明, 島田 信敬, カメラ位置姿勢の高速な推定のためのランドマーク選択法と単眼視によるロボットのナビゲーション, 日本ロボット学会誌 Vol.29 No.9, pp811 ~ 820, 2011
- [10] 薄 充孝, 中川 知香, 佐藤 智和, 横矢 直和, ランドマークデータベースに基づく静止画像からのカメラ位置・姿勢推定, 日本バーチャルリアリティ学会論文誌 Vol.13, No.2, 2008
- [11] 小暮 和重, 塩谷 敏昭, 太田 直哉, 最小認識ハードウェアの自律移動ロボット-つくばチャレンジ参加ロボット MG-11-, 計測自動制御学会論文集 Vol.49, No.7, 703/712 (2013)
- [12] 鹿貫 悠多, 中村 弘樹, 清水 美咲, 竹野 翔太, 太田 直哉, つくばチャレンジ 2014 参加報告書, TSUKUBA CHALLENGE 2014 参加レポート集
- [13] Frank Dellaert, Dieter Fox, Wolfram Burgard, Sebastian Thrun, Monte Carlo Localization for Mobile Robot : Robotics and Automation, 1999. Proceedings. 1999 IEEE International Conference p.1322-1328
- [14] 江口 純司, 尾崎 功一, DGPS を用いて設定したウェイポイント経路追従による実環境自律移動法の開発, 日本ロボット学会誌 Vol.30 No.3, pp.314-323, 2012

- [15] Naoki Akai, Satoshi Hoshino, Kazumichi Inoue and Koichi Ozaki, Monte Carlo Localization using Magnetic Sensor and LIDAR for Real World Navigation, Proceedings of the 2013 IEEE/SICE International Symposium on System Integration p682-687
- [16] 畑尾 直孝, 花井 亮, 斉藤 学, 山崎 公俊, 稲葉 雅幸, 3次元 LRF による道路構造の理解に基づく屋外マッピングシステムの構築, Proceedings of the 2010 JSME Conference on Robotics and Mechatronics, 1A1-D18
- [17] 岩科進也, 山下淳, 金子透, LRF 搭載移動ロボットを用いた動的環境における 3-D SLAM 第 26 回日本ロボット学会学術講演会予稿集, 2008
- [18] 吉田 智章, 入江 清, 小柳 栄次, 友納 正裕, 3D スキャナとジャイロを用いた屋外ナビゲーションプラットフォーム, 計測自動制御学会論文集 Vol.47, No.10, 493/500 (2011)
- [19] 倉島 一輝, 大川 一也, 単一固定の測域センサに基づく位置推定用三次元地図構築および進路上の障害物検出, 第 16 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 (SI2015) 1K1.4
- [20] P.J. Besel, N.D. McKay "A method for registration of 3-D shapes", *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol.14, No.2 pp.239-256, 1992
- [21] Shinji Nakabayashi, Hirokazu Matsui, and Norihiko Kato, Instable Objects Recognition to Estimate Localization for Autonomous Mobile Robots, proc. of Artificial Life and Robotics 20th
- [22] Nello Cristianini, John Shawe-Taylor, "An Introduction to Support Vector Machines and other kernel-based learning methods", Cambridge University Press, 2003

- [23] 栗田 多喜夫, ”サポートベクターマシン入門”, 産業技術総合研究所 脳神経情報研究部門, 2002
- [24] 小野田 崇, ”知の科学 サポートベクターマシン”, 人工知能学会編, 2007
- [25] <http://www.csie.ntu.edu.tw/~cjlin/libsvm/>, libsvm, Chin-Chung Chang and Chih-Jen Lin

謝辞

本研究を遂行するにあたり、御指導と御鞭撻を賜りました三重大学 松井 博和 助教に対して、深く感謝いたします。

本論文をまとめるにあたり、三重大学 加藤典彦 准教授から懇切なる御指導と御助言を頂き、謹んで感謝の意を表します。

本研究に際して、様々なご指導を頂きました三重大学 矢野賢一 教授に対して深謝いたします。

また、本研究を進めるにあたり、メカトロニクス研究室の諸氏、その中でも同じ班で研究し、実験等多岐に渡って研究を手伝っていただきました、池本先輩、倉田先輩、井畑先輩、中林先輩、山本君、永井君、益川君に心より感謝いたします。