

不確かさを考慮した複数のスキャンマッチング結果の統合による 移動ロボットの移動量推定

子安大士* 三浦 純* 白井良明*

Integrating Multiple Scan Matching Results for Ego-Motion Estimation with Uncertainty

Hiroshi Koyasu*, Jun Miura* and Yoshiaki Shirai*

This paper describes an ego-motion estimation method by integrating multiple scan matching results. The method considers both the uncertainty of scan matching results and that of estimated ego-motions, and not only estimates the latest robot ego-motion but also updates previous ego-motions. The estimation process is formulated as an iterative one using Kalman filter. We implement the ego-motion estimation method using an omnidirectional stereo-based scan matching method which considers the uncertainty of the range data, and estimates the uncertainty of the result of the scan matching. Experimental results show the effectiveness of the proposed method.

Key Words: Ego-motion Estimation, Scan Matching, Mobile Robot, Omnidirectional Stereo

1. はじめに

自己移動量の推定は、移動ロボットが動きながら観測をした情報を統合して、周囲の環境を認識するためには欠かせない機能である。デッドレコニングによる移動量の推定では、タイヤのスリップなどにより誤差が蓄積されていくため、カメラなどの外界センサを用いた移動量推定手法が必要となってくる。そのような外界センサを用いたロボットの自己位置、あるいは自己移動量の推定手法に関してはこれまで多くの研究がなされている。移動の前後の観測における特徴の対応に基づく手法（例えば、文献 [2] [3] [13]）は、環境中に特定の特徴が存在することを前提としているうえ、観測から特徴を取り出す処理や、特徴同士の対応関係の決定などは一般に計算量が多くなる可能性がある。一方で、そのような特徴を用いないスキャンマッチングと呼ばれる手法は、観測した距離情報から得られる二次元の障害物の輪郭同士を対応づけることで移動量を推定しており、そのような問題は生じない。

Lu ら [9] はレーザレンジファインダから得た、前回と今回の二つの観測を用いてスキャンマッチングを行う方法を提案している。Pfister ら [11] は、センサノイズや観測同士の対応づけにおける個々のデータの対応関係の不確かさを考慮するように彼らの手法を拡張し、スキャンマッチングの推定結果の不確かさも求めて、その結果とデッドレコニングによる推定結果とを不確かさを考慮して統合できるようにした。しかしながら、これ

らの手法は二つの観測しか用いていないので、一方の観測におけるノイズによって障害物の輪郭の対応が誤ったものになってしまったときに、その誤りを修正できない。そのため、距離情報の不確かさが大きく、しばしば誤った輪郭の対応づけをしてしまうようなセンサに対して用いるには不適である。

複数の観測を統合した地図を作成し、そこから観測として得られるスキャンを予測し、それと実際に観測したスキャンとを比較する手法（例えば文献 [5] [7]）も提案されている。しかし、これらの手法は、一度推定した移動量を後から推定し直すことをしていない。

一方、過去の移動量（あるいは位置）を再推定する手法としては、環境中を一回りして過去にいた位置の近くに再び来たときに、位置の履歴の整合性をとるという問題（ループ解決）に対処するための手法が提案されている（例えば文献 [1] [6]）。しかし、これらの手法はループを検出したときに初めて再推定しており、継続的な再推定は行っていない。

本論文では、複数のスキャンマッチングの結果を統合して、ロボットの自己移動量を推定する手法について述べる。Fig. 1 に、時刻 $t-2$, $t-1$, および t において観測された三つの距離情報を用いた場合の例を示す。 $\mathbf{X}_t^{(t-1)}$ を $[t-1, t]$ 間の移動量とすると、これは時刻 $t-1$ における観測と時刻 t の観測とを比較することで得られる。 $\mathbf{X}_{t-1}^{(t-2)}$ についても同様の計算で得ることができる。従来手法では、このように二つの観測のみを用いて、それぞれの観測の間におけるロボットの移動量を推定していた。しかし、これらとは別の距離情報の組み合わせである、時刻 $t-2$ の観測と時刻 t の観測による推定結果 $\mathbf{X}_t^{(t-2)}$ を使用することで、時刻 t における自己移動量の推定結果を改善す

原稿受付 2004年4月5日

*大阪大学大学院工学研究科電子制御機械工学専攻

*Computer-Controlled Mechanical Engineering, Osaka University

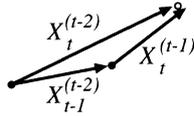


Fig. 1 Integrate three scan matching results

ることができる．また， $X_{t-1}^{(t-2)}$ については，現在の観測を用いずに推定できるため，時刻 t 以前に推定されていることになるが，一方で $X_t^{(t-1)}$ と $X_{t-1}^{(t-2)}$ の差から推定することもできる．よって，この差と以前のスキャンマッチング結果を統合すれば， $X_{t-1}^{(t-2)}$ をより確からしくすることができる．

複数のスキャンマッチング結果を統合する場合， $X_t^{(t-1)}$ ， $X_{t-1}^{(t-2)}$ ，および $X_t^{(t-2)}$ のそれぞれが不確かさを持つ．それぞれのスキャンマッチング結果が持つ不確かさは，センサデータのノイズや環境によって変化する．そのため，統合の際にそれぞれのスキャンマッチング結果の不確かさの大きさを考慮してやる必要がある．

本論文では，時刻 $t-k$ から時刻 t までの観測を用いて得られるスキャンマッチングの結果を，不確かさを考慮して統合することで，ロボットの現在の移動量を推定すると同時に，過去に推定したロボットの移動量を修正する手法を提案する．このとき，毎回 $k+1$ 個のロボットの位置関係を，それらの位置で得られたすべての観測の組み合わせから推定するとすると， $k+1 C_2$ 個の観測の組み合わせから得られるスキャンマッチングについて計算することになり，計算のコストが大きくなる．そこで，提案手法ではカルマンフィルタを用いて逐次的に推定をし，時刻 $t-k$ からみた時刻 t までの k 個のそれぞれのロボットの位置を，現在の観測と直前の k 個の観測の組み合わせから推定する．

本手法を実装するためには，ロボットの移動量とともにその不確かさを推定することができるスキャンマッチングの手法が必要である．そこで，我々は全方位ステレオ視を用いた手法を開発した．全方位ステレオ視は 360° のほぼ全周を観測でき，三次元的な距離情報を取得できるため，複雑な環境における移動ロボットのナビゲーションに適したセンサである．本手法では全方位ステレオ視により得られる三次元の距離情報を二次元の距離情報に変換してからスキャンマッチングを行う．この方法を用いて提案した移動量推定手法を実装し，実環境における移動量推定実験を行って，提案手法の有効性を確かめた．

2. 複数のスキャンマッチングの結果の統合

本章ではカルマンフィルタを用いて複数のスキャンマッチング結果を統合し，ロボットの移動量を推定する手法について述べる．統合するスキャンマッチング結果の個数は最大で k 個とする．本手法では， k 回前の観測地点からみた，現在の位置を含む k 個のロボットの相対位置を， $k+1$ 個の観測を用いて推定する．

まず，時刻 $t-k$ における位置からみた時刻 $t-i$ における相対位置・姿勢を $\mathbf{X}_{t-i}^{(t-k)} = (x_{t-i}^{(t-k)}, y_{t-i}^{(t-k)}, \theta_{t-i}^{(t-k)})^T$ として表す．ここで T は転置を表す．そして，これを用いて状態ベクトル S_t を次のように定義する：

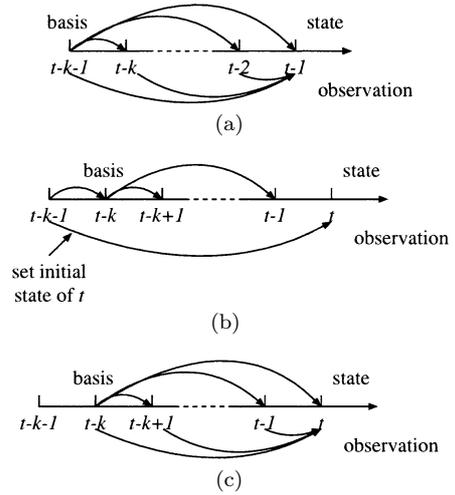


Fig. 2 State transition model

$$S_t = (\mathbf{X}_{t-k+1}^{(t-k)T}, \mathbf{X}_{t-k+2}^{(t-k)T}, \dots, \mathbf{X}_t^{(t-k)T})^T \quad (1)$$

観測として得られるものは，現在の観測とそれ以前の k 個の観測とをスキャンマッチングで比較した結果である．時刻 $t-i$ の観測と時刻 t の観測の組み合わせから，時刻 $t-i$ における位置からみた時刻 t における位置が得られるので，これを $p_{t,t-i}$ とおくと観測ベクトル P_t は：

$$P_t = (p_{t,t-k}^T, p_{t,t-k+1}^T, \dots, p_{t,t-1}^T)^T \quad (2)$$

となる．

Fig. 2 に状態が逐次的に更新される様子を模式的に示す．Fig. 2 (a) は時刻 $t-k-1$ におけるロボットの位置からみた，時刻 $t-k$ から時刻 $t-1$ までの各時刻におけるロボットの相対位置が，時刻 $t-1$ の推定によって得られている状況を表す．Fig. 2 (b) は S_{t-1} から S_t への状態遷移を示しており， $\mathbf{X}_{t-k}^{(t-k-1)}$ を用いて相対位置の基準となる位置を時刻 $t-k$ に変換し，同時に新しい変数である時刻 t におけるロボットの位置の初期値を，時刻 t における観測と時刻 $t-k-1$ における観測とのスキャンマッチングの結果を用いて計算する．そして，Fig. 2 (c) に示すように， k 個の新たに得たスキャンマッチングの結果を統合し，時刻 $t-k+1$ から時刻 t までのロボットの位置を推定する．

Fig. 2 (b) によって表される状態遷移から，状態方程式は以下のように定義される：

$$S_{t/t-1} = \begin{pmatrix} -I & I & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ -I & 0 & \dots & I \\ 0 & 0 & \dots & 0 \end{pmatrix} S_{t-1/t-1} \quad (3)$$

$$+ \begin{pmatrix} 0 \\ \vdots \\ 0 \\ p_{t,t-k-1} \\ -\mathbf{X}_{t-k}^{(t-k-1)} \end{pmatrix} + \mathbf{v}_t$$

$$v_t = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ \delta p_{t,t-k-1} - \delta X_{t-k}^{(t-k-1)} \end{pmatrix}$$

ただし $\mathbf{0}$ は 3×3 のゼロ行列, I は三次元の単位行列, $\delta p_{t,t-k-1}$ と $\delta X_{t-k}^{(t-k-1)}$ は観測 $p_{t,t-k-1}$ と相対位置 $X_{t-k}^{(t-k-1)}$ の誤差である. $\delta p_{t,t-k-1}$ の共分散はスキャンマッチングの結果から得られ, $\delta X_{t-k}^{(t-k-1)}$ の共分散は以前の状態ベクトルの推定誤差共分散から該当する要素を取り出すことで得られる. 上式では, $X_t^{(t-k)}$ の初期値として, $p_{t,t-k-1}$ と $X_{t-k}^{(t-k-1)}$ を用いており, 本論文で提案しているカルマンフィルタでは観測の統合のみを行っている. この式をオドメトリの結果と統合するように変更するのは, 簡単な拡張により可能である.

同様に Fig. 2 (c) によって表される観測の統合から, 観測方程式は以下のように定義される:

$$P_t = \left(\begin{array}{ccc|c} \mathbf{0} & \dots & \mathbf{0} & I \\ -I & \dots & \mathbf{0} & I \\ \vdots & \ddots & \vdots & \vdots \\ \mathbf{0} & \dots & -I & I \end{array} \right) S_t + w_t \quad (4)$$

ただし w_t は観測値の誤差を表す白色ノイズベクトルであり, この共分散行列は観測ベクトル P_t の誤差共分散 Σ_{P_t} と等しい.

スキャンマッチング結果 $p_{t,t-i}$ の誤差要因としては, 「観測した距離情報の誤差」と「個々の観測中の各点の対応付けの誤り」の二つが挙げられる. 「距離情報の誤差」は個々の観測におけるノイズに依存し, また, 「対応付けの誤り」は周囲環境における空き領域の形状に依存している. 観測ベクトル中の任意の二つの要素 $p_{t,t-i}$ と $p_{t,t-j}$ ($i \neq j$) には, それぞれに時刻 t において得た距離情報が使われているため, それらのスキャンマッチング結果は厳密には独立ではない. しかし, 時刻 t における「距離情報の誤差」がスキャンマッチングの結果に与える影響は「対応付けの誤り」に比べると十分に小さいと考えられる. そこで本論文では, 観測ベクトル中の個々のスキャンマッチング結果は互いに独立であると見なす. このとき, Σ_{P_t} は以下ようになる:

$$\Sigma_{P_t} = \begin{pmatrix} \Sigma_{p_{t,t-k}} & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & \Sigma_{p_{t,t-k+1}} & \dots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \dots & \mathbf{0} & \Sigma_{p_{t,t-1}} \end{pmatrix} \quad (5)$$

ただし, $\Sigma_{p_{t,t-i}}$ はスキャンマッチングによって得られる相対位置の誤差共分散行列であり, スキャンマッチング処理の過程で計算される (3.4 節参照).

以上の状態方程式 (式 (3)) と観測方程式 (式 (4)) によりカルマンフィルタを適用することで, 状態ベクトル S_t とその不確かさを逐次的に更新することができる. なお, 初期位置においてはロボットの位置と方向には誤差が存在しないとす. また, 現状では $k=5$ としている (4.1 節参照) が, 観測の回数が $k+1$ よりも少ない場合, 状態ベクトルの次元を観測回数に

応じた数にして推定を行う. そして, 現在のロボットの移動量は $X_t^{(t-k)} - X_{t-1}^{(t-k)}$ により求められる.

3. 全方位ステレオ視によるスキャンマッチング

本章では, 全方位ステレオ視を用いたスキャンマッチング手法について述べる. 本論文で提案している移動量推定には, ロボットの移動量とその誤差の推定ができるスキャンマッチング手法が必要である. そこで, 距離情報における不確かさとそれを観測した位置の不確かさを考慮して, スキャンマッチングにより移動量とその不確かさを推定する手法を提案する.

提案するスキャンマッチング手法では, 以下の手順に従ってロボットの移動量とその誤差を推定する.

- (1) 全方位ステレオ視からロボットの周囲の二次元の距離情報を抽出する.
- (2) 直前のロボットの位置とその不確かさに, オドメトリを用いたデッドレコニングにより得られたロボットの移動量とその不確かさを足すことで, ロボットが現在存在する可能性のある領域を計算し, その領域内からロボットの位置と方向の候補を作成する.
- (3) 各候補において現在の観測で得られた距離情報と, 以前の観測から予測される距離情報とを比較する. その差が小さいほどその候補の位置と方向は正しいロボットの現在の位置と方向である可能性が高い.
- (4) それぞれの位置と方向の候補における距離情報の差の値が小さいほど, その候補位置と方向の重みが大きくなるように重みを設定し, 重み付き最小二乗推定を適用してロボットの位置およびその誤差を推定する.

3.1 全方位ステレオ視

移動ロボットが外界を認識するためのセンサとしては, カメラや超音波センサ, レーザレンジファインダ (LRF) などが一般的に用いられる. 中でも LRF は広い視野を持ち, 角度方向の解像度も高く, 高精度の距離情報を得ることができるが, 特定の高さの二次元平面をスキャンしているため, 特定の高さにある物体しか検出できないという欠点があり, そのため複雑な環境で用いるには適さない. 我々が使用する全方位ステレオ視覚は, 360° のほぼ全周を観測でき, LRF と違って三次元的な距離情報を取得できるため, 複雑な環境における移動ロボットのナビゲーションに適したセンサである.

全方位ステレオ視覚は双曲面ミラーを用いた全方位画像センサ HyperOmni Vision [14] を, Gluckman ら [4] と同様に鉛直に二つ並べて使用する. Fig. 3 に下側カメラの入力画像の例を示す. それぞれのカメラからの入力画像はカメラの光軸に沿った円筒面上に投影することで Fig. 4 に示すようなパノラマ画像に変換する. これにより鉛直方向に平行なエピポーラ線を持つステレオ画像対を得ることができ, 通常の透視投影カメラを用いた平行ステレオ視に用いられる既存の高速な対応付け手法を適用できる. 本手法では SAD を用いた対応付け手法を使用している.

計算速度向上のため, パノラマ画像を半分に分割し, Dual CPU (AMD Athlon MP 2200+) の PC を用いて, そのそれぞれに対し並列にステレオ対応付けを行う. パノラマ画像サイ

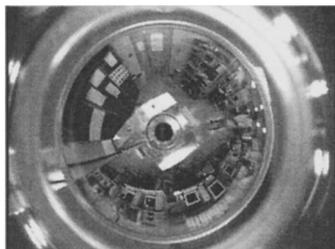


Fig. 3 An example of the original input image



Fig. 4 Panoramic image obtained from the input image shown in Fig. 3



Fig. 5 Panoramic disparity image obtained from the images in Fig. 4. Brighter pixels are nearer

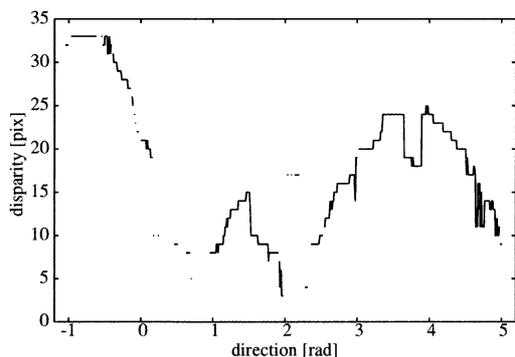


Fig. 6 Example range profile

ズ 720×100 で視差の探索範囲を 80 としたときのパノラマ変換とステレオの対応付けに要する時間は約 0.1 秒である。カメラ焦点間の距離は 30 [cm] であり、パノラマ画像の大きさと視差の探索範囲を上記の設定にした場合、最小約 50 [cm] から最大約 40 [m] までの距離を測定できる。また角度方向はカメラを支える柱の分が死角になるが、ロボット前方を正面として約 -60° から 288° までの視野を約 0.5° 刻みで観測することができる。対応付けの結果得られる視差画像を Fig. 5 に示す。図中で明るいところほど視差が大きい(ロボットに近い)物体を表す。

移動ロボットが障害物のない(移動可能な)空き領域を認識するためには、ロボットから一番近くにある障害物までの距離が認識できれば十分である。そこで、パノラマ画像における横軸はロボットからの方向と一致するので、視差画像中の各列に対して最も近くにある物体を表す視差を抽出する。具体的には、隣り合う画素の視差の差が 1 未満の場合はそれらが連続しているとして、各列をスキャンして連続した視差が 5 画素以上続いている領域のみを取り出し、その中で最も大きな視差を抽出する。このようにして得られた方向と視差の組み合わせを並べたものを本論文ではレンジプロファイル(RP)と呼ぶ。Fig. 5 から得られる RP を Fig. 6 に示す。

3.2 候補位置・方向の作成

Fig. 7 に実験用移動ロボットを示す。この移動ロボットは後ろの二つの車輪を駆動させて移動する。ロボットの状態 $X = (x, y, \theta)^T$ はロボットの位置の中心を全方位ステレオカメラの

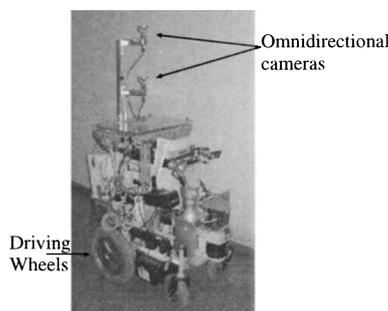


Fig. 7 Our mobile robot

位置としたときの二次元位置 (x, y) と方向 θ からなる。オドメトリによってロボットの位置 X とその不確かさ Σ_X が更新される(付録 A 参照)ので、更新された位置 X を中心としてその分散 Σ_X からいわゆる 3σ の範囲をとった領域をロボットの存在する領域とする。この領域は $x-y-\theta$ の三次元において楕円体をなすが、ロボットの位置の存在する領域はこの楕円体を $X-Y$ 平面上に投影することで得られる楕円で表すこととし、方向の存在する領域はその不確かさが θ の周辺分布であるとして計算する。

このようにして得られた領域内においてロボットの位置と方向の候補を設定する。まず、位置の候補については、推定された楕円内において、二つの主軸に平行な線により格子を設け、その格子点に設定する。格子の原点は楕円の中心をとるようにし、各方向の格子の数はそれぞれの方向の主軸の長さをそれで割ったときに 50 [mm] 以下となる最小の奇数(ただし 3 以上)とする。例えば、楕円の主軸の長さがそれぞれ 200 [mm] と 140 [mm] であった場合、得られる格子の数は 5×3 となる。また、方向の候補は推定した不確かさの範囲を RP の角度解像度 $\delta\phi$ で離散化する。現在のところ、 $\delta\phi$ の値はパノラマ画像の解像度から約 0.5° となっている。

3.3 レンジプロファイルの差の計算

ある候補位置 (x, y) において、現在の観測で得られた RP と過去の観測から得られた RP とを比較するために、まず、その候補位置から観測を行えばどのような RP が得られるかの予測を、過去に観測した RP とそれを観測した位置から計算する。Fig. 8 は時刻 $t-i$ に $X_{t-i} = (x_{t-i}, y_{t-i}, \theta_{t-i})^T$ から観測し

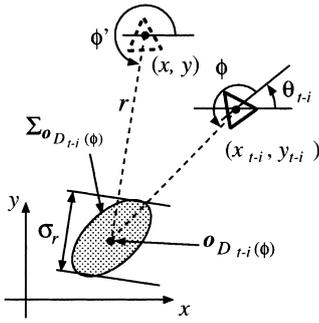


Fig. 8 View of previous observed obstacle from candidate position (x, y)

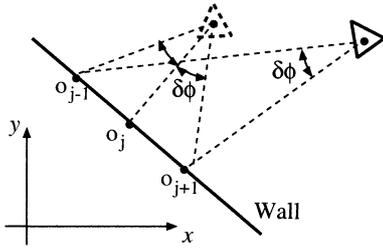


Fig. 9 Problem caused by fixed angular resolution

た物体が候補位置 (x, y) からどのように見えるかを表す．時刻 $t-i$ において観測した RP 中の ϕ 方向の視差 $D_{t-i}(\phi)$ によって表される障害物の位置 $o_{D_{t-i}(\phi)}$ は、時刻 $t-i$ におけるロボットの位置 X_{t-i} を用いて以下のように表される：

$$o_{D_{t-i}(\phi)} = \begin{pmatrix} \frac{bf}{D_{t-i}(\phi)} \cos(\theta_{t-i} + \phi) + x_{t-i} \\ \frac{bf}{D_{t-i}(\phi)} \sin(\theta_{t-i} + \phi) + y_{t-i} \end{pmatrix} \quad (6)$$

ただし、 b は全方位ステレオにおけるベース距離を、 f はパノラマ画像の仮想焦点距離をそれぞれ表す．この $o_{D_{t-i}(\phi)}$ の視差 $\Sigma_{o_{D(\phi)_{t-i}}}$ は位置 X_{t-i} の誤差 $\Sigma_{X_{t-i}}$ と視差 $D(\phi)$ の誤差 $\sigma_{D(\phi)}^2$ からの誤差伝播によって計算でき、後者はパノラマ画像の量子化誤差から $\sigma_{D(\phi)}^2 = \sigma_{D^*}^2 = 1$ とする．

$o_{D_{t-i}(\phi)}$ が計算できると観測した物体と候補位置 (x, y) との位置関係が決定するので、その候補位置から見た物体の方向 ϕ' と距離 r を計算し、さらに距離 r を $D = \frac{bf}{r}$ により視差 $D_{t-i}^{(x,y)}(\phi')$ に変換する．また、その誤差 $\sigma_{D_{t-i}^{(x,y)}(\phi')}$ は、まず、 $\Sigma_{o_{D(\phi)_{t-i}}}$ を候補位置から $o_{D_{t-i}(\phi)}$ へ引いた直線へ投影することにより σ_r^2 を計算し、さらに、その σ_r^2 からの誤差伝播により計算する．

以上の計算を時刻 $t-i$ において得られた RP のすべての方向の視差に対して行うことで、時刻 $t-i$ に観測した物体が候補位置 (x, y) においてどのような視差となるかが得られる．しかし、RP の角度解像度が固定であるために、時刻 $t-i$ において一つの物体としてみなせるような、隣り合う方向の視差の連続している部分が、候補位置から見ると隙間が生じ、別々の物体であるように見える場合がある．Fig. 9 にそのような状況の例を示す．図中では候補位置から見た j 番目のデータに対応する以前の観測が存在しない．このような場合、その両側の視差（例えば図中の o_{j-1} と o_{j+1} ）が同じ物体であるともみなすの

に十分なだけ距離が近ければ、この j 番目の視差をその両側の o_{j-1} と o_{j+1} から線形に補間して求める．

このようにして得られた、時刻 $t-i$ における観測から予測される候補位置 (x, y) での RP と、時刻 t において観測した RP を比較することで RP の差を求める．ある候補となる位置と方向 (x, y, θ) において、時刻 t での観測における ϕ 方向における視差 $D_t(\phi)$ と上で求められる時刻 $t-i$ での観測における視差との差を

$$d(x, y, \theta, i, \phi) = \frac{(D_t(\phi) - D_{t-i}^{(x,y)}(\phi - \theta))^2}{\sigma_{D_{t-i}^{(x,y)}(\phi - \theta)}^2 + \sigma_{D^*}^2} \quad (7)$$

とする． $d(x, y, \phi, i, \theta)$ はマハラノビス距離を表しており、式 (7) 中の二つの視差の誤差が正規分布であるとする、同じ物体から観測された視差同士で d を計算すれば χ^2 分布に従う．そこで、 3σ の範囲が入るようにしきい値を決め、 d がそれよりも大きい場合はそのしきい値の値をとるものとする．このようにすることにより、ステレオの対応付けの誤りや移動障害物によって、誤った物体の対応付けが行われたときの影響を軽減できる．また、式 (7) 中の二つの視差のどちらか一方がステレオの対応付けの失敗により得られていない場合は、 d が得られないものとする．

RP の差は次のように計算する：

$$Diff(x, y, \theta, i) = \frac{1}{N(x, y, \theta, i)} \sum_{\phi=\phi_{min}}^{\phi_{max}} d(x, y, \theta, i, \phi) \quad (8)$$

ここで、 $[\phi_{min}, \phi_{max}]$ は視野の範囲（全方位画像の右と左の端に相当する）を、 $N(x, y, \theta, i)$ は視差の差 d を計算することができたデータの個数をそれぞれ表している．

3.4 二つの RP の比較による移動量とその誤差の推定

Fig. 10 (b) は Fig. 10 (a) に示す廊下環境における、予測された位置のまわりにおける RP の差の分布を表す．Fig. 10 (b) において、正しいロボットの位置は差の分布の谷になっているところのどこかにあると考えられ、この谷の形状はロボットの位置の確率分布と相関があるとみなせる．そこで、この差の分布から、ロボットの位置と方向の確率分布を計算することを考える．

RP の差の分布から位置と方向の確率分布を推定するために、Nickels ら [10] の手法を適用する．彼らは動画像中の特徴の追跡をするという課題において、画像中での追跡対象を差の二乗和 (SSD) を評価値とするテンプレートマッチングによって探索する際に、SSD の差の分布から追跡対象の画像上の位置の確率分布を求めている．確率分布を求めるために、彼らはまず予測された点の周りで SSD を計算し、それを Response Distribution (RD) [12] という各点における信頼性を表す値に変換している．

我々の手法において、RD は次のように計算される：

$$r(x, y, \theta, i) = \exp(-\kappa Diff(x, y, \theta, i)) \quad (9)$$

κ は信頼性に対する重み係数であり、現在のところ実験的に 1 を用いている．Fig. 10 (c) に Fig. 10 (b) を RD に変換した結果

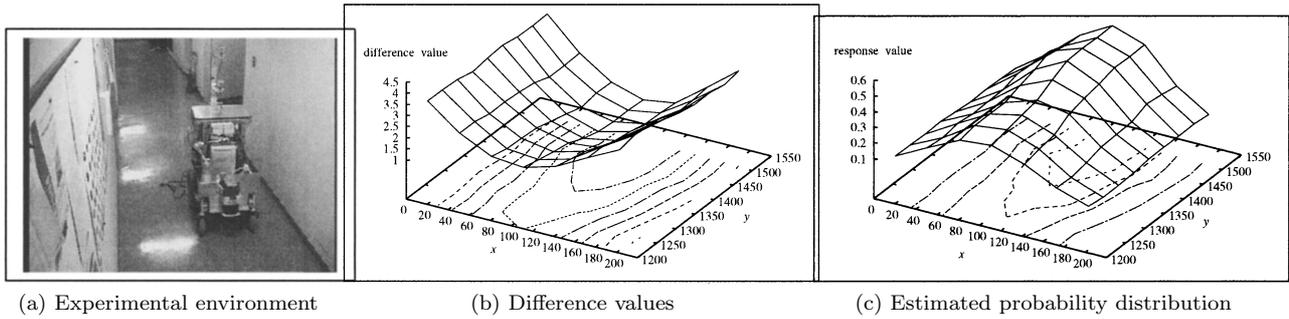


Fig. 10 Estimation of the positional distribution

を示す.

RD は位置と方向の各候補がどれだけ確からしいかを表しており, 正しい位置と方向の確率分布として解釈できる. よって, この形状を (x, y, θ) の三次元ガウス分布に近似すると, その平均と分散がロボットの現在の位置と方向, およびその不確かさとなる. 平均は以下の重み付き最小二乗により計算できる:

$$\begin{aligned} \hat{x}_i &= \frac{\sum_{x,y,\theta} r(x,y,\theta,i)x}{\sum_{x,y,\theta} r(x,y,\theta,i)} \\ \hat{y}_i &= \frac{\sum_{x,y,\theta} r(x,y,\theta,i)y}{\sum_{x,y,\theta} r(x,y,\theta,i)} \\ \hat{\theta}_i &= \frac{\sum_{x,y,\theta} r(x,y,\theta,i)\theta}{\sum_{x,y,\theta} r(x,y,\theta,i)} \end{aligned} \quad (10)$$

分散についても同様に重み付き最小二乗により計算できる. 例えば, x_i の誤差の分散 Σ_{x_i, x_i} は:

$$\Sigma_{x_i, x_i} = \frac{\sum_{x,y,\theta} r(x,y,\theta,i)(x - \hat{x})(x - \hat{x})}{\sum_{x,y,\theta} r(x,y,\theta,i)}$$

となり, 同様に x_i と y_i の誤差の共分散 Σ_{x_i, y_i} は:

$$\Sigma_{x_i, y_i} = \frac{\sum_{x,y,\theta} r(x,y,\theta,i)(x - \hat{x})(y - \hat{y})}{\sum_{x,y,\theta} r(x,y,\theta,i)}$$

のようになる.

このようにして得られた $(\hat{x}_i, \hat{y}_i, \hat{\theta}_i)$ の組み合わせが, 2章で述べた $p_{t,t-i}$ となり, また, $(\hat{x}_i, \hat{y}_i, \hat{\theta}_i)$ の分散が $\Sigma_{p_{t,t-i}}$ となる.

4. 実験

4.1 観測ベクトルの次元 k の選択

観測ベクトル(式(2)参照)の次元 k の値を大きくすれば, より多くの情報を統合することになり, 移動量推定の精度が上がるのが期待できる. しかし, k を大きくすると, 観測地点間の距離が離れるので, スキャン同士の対応点が少なくなって, 結果としてスキャンマッチングの精度は下がる可能性がある. したがって, 適切な k の値を選択する必要がある.

最適な k の値はロボットの移動速度と観測のサイクルタイムに依存する. そこで, 実験に用いるロボットの移動速度(30[cm/s])とサイクルタイム(1[sec])について, 研究室環境で移動量推定実験を行い, 適切な k の値を調べた. Fig. 11 に, k の値を変化させたときの移動量の推定結果について, 外部から測定した移動量との平行移動成分の誤差を調べ, その誤差の平均を計算したものを図示する. この結果では, $k = 6$ が最も良いが,

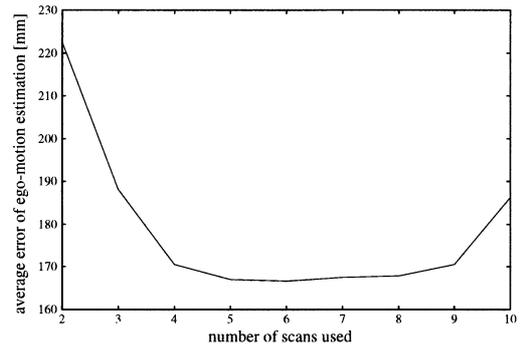


Fig. 11 Average error vs. number of scan used for ego-motion estimation

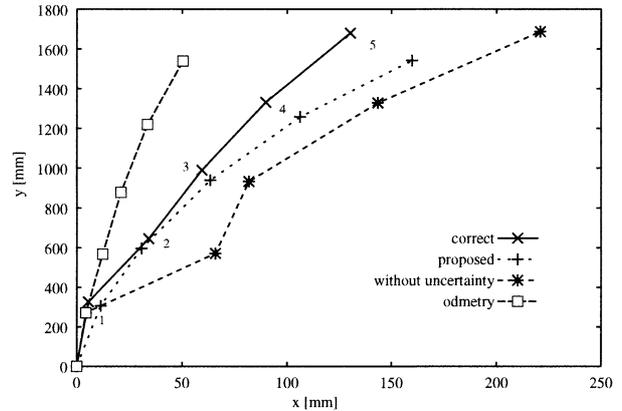


Fig. 12 Results of the experiment in section 4.2

$k = 5$ でもほとんど同じ結果が得られると判断し, より計算量の少ない $k = 5$ を採用した.

4.2 移動量推定における不確かさを考慮することの効果の検証

提案手法の有効性を確かめるために, 静的環境下でロボットを移動させて移動量を推定する実験を行った. 実験を行った環境は Fig. 10(a) に示した廊下環境で, ロボットはこの廊下を前方にほぼまっすぐ移動させた. また, オドメトリに明示的に誤差を与えるためにロボットの車輪がコードを踏むようにした. その影響によりロボットの最終的な位置はおよそ(140, 1600)[mm]となった. 観測は初期位置も含め全部で6個所で行い, それぞれの観測点の間の移動距離はおよそ300[mm]であった.

Fig. 12 に実験結果として

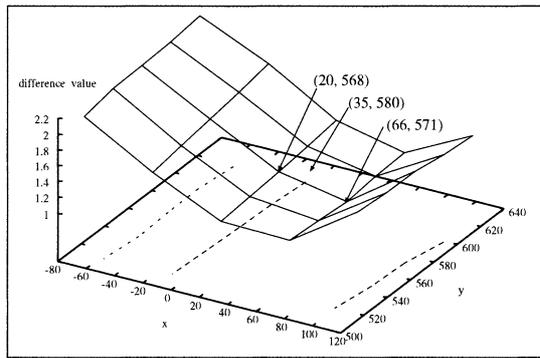


Fig. 13 An example distribution of the sum of differences between scans

- メジャーにより計測したロボットの位置 (図中 “correct”).
 - 提案手法による結果 (同 “proposed”).
 - オドメトリにより予測した範囲内に候補位置, 方向を設定し, そのそれぞれで過去 $k (= 5)$ 回の観測結果と時刻 t に得られた観測とで式 (8) で示した RP の差の和を求め, その和が最小となる候補位置, 方向をロボットの位置, 方向であるとする手法. スキャンマッチング結果における不確かさを考慮しておらず, また現在のロボットの移動量しか推定していない (同 “without uncertainty”).
 - オドメトリによって推定した移動量 (同 “odometry”).
- をそれぞれ示す.

移動量の不確かさを考慮しない手法 (“without uncertainty”) では図中の 1 から 2 への移動における移動量推定において, RP の差の和が最小となる位置は (66, 571) で, ここでの差分の和は 1.02596 であった. 前回の位置からの移動量から計算される正しいロボットの位置はおよそ (35, 580) [mm] であるが, 最もこれに近い候補位置 (20, 568) では差の和は 1.14596 であった. Fig. 13 に示すようにこのときの RP の差の分布の谷は広くっており, わずかなノイズの影響で位置推定結果がずれてしまったものと考えられる. 不確かさを考慮しない手法は, 過去の位置推定結果の誤差を考慮していないので, そのずれは以降の推定にそのまま影響を与えることになる. そのため, 不確かさを考慮しない手法による位置推定の誤差は, 提案手法の推定結果よりも大きくなる.

一方で, 提案手法では不確かさを考慮しない手法で起きたような問題は生じず, x 方向に対してほぼ正しい推定結果が得られた. 一方で y 方向には十分な情報がないうえ, 予測した候補領域の中心の値がそのまま用いられることになり, 結果的にオドメトリの推定結果とほぼ同じになった. 以上の結果から, 提案手法はオドメトリの信頼性が低い状態でも十分に正しく移動量を推定できていることが確認できる.

4.3 移動量推定実験

別の実験として, Fig. 14 に示されるような多数の障害物が存在する複雑な環境における, 移動量の推定実験を行った. 比較のために, 以下の手法で推定した結果も示す.

- 外部の固定されたカメラから撮影したロボットの位置を, 床面に投影することで推定した, ロボットの絶対位置 (図中



Fig. 14 Experimental environment of section 4.3

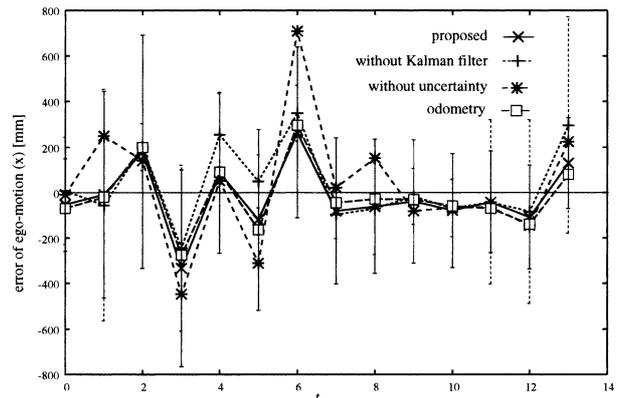


Fig. 15 Error of ego-motion estimation of robot (x)

“correct”).

- 提案手法による結果 (同 “proposed”).
- 前節で述べた不確かさを考慮しない手法 (同 “without uncertainty”).
- オドメトリにより予測した範囲内で候補となる位置と方向を求め, そのそれぞれで RP の差の和を, 不確かさを考慮しない手法 (“without uncertainty”) と同様に計算し, その和の分布について RD (式 (9)) を求めて, 重み付き最小二乗によりロボットの移動量とその誤差を推定する手法. スキャンマッチングの不確かさの推定は行わぬが, 現在の移動量のみを推定し, 過去に推定した移動量のカルマンフィルタによる再推定は行わない (同 “without Kalman filter”).
- オドメトリによって推定した移動量 (同 “odometry”).

各手法で推定されたロボットの移動量をロボット中心座標系で表現し, 外部カメラから得られたロボットの位置の差から計算されるロボットの移動量と推定した移動量の誤差を, x , y , および θ のそれぞれについて計算した結果を Fig. 15 ~ 17 に示す. 図中では, ロボットの移動量の不確かさの推定結果から 3σ の範囲を計算し, エラーバーとして表示している. Fig. 18 は計測した正しいロボットの軌跡と, それぞれの手法で推定された移動量の累積として計算されるロボットの位置の軌跡を示す. また, Table 1 は Fig. 15 ~ 17 に示される移動量の誤差について計算した標準偏差を示す.

方向に関してはすべての手法において比較的大きくずれるということではなく, 良い精度で推定できている. これは全方位ステ

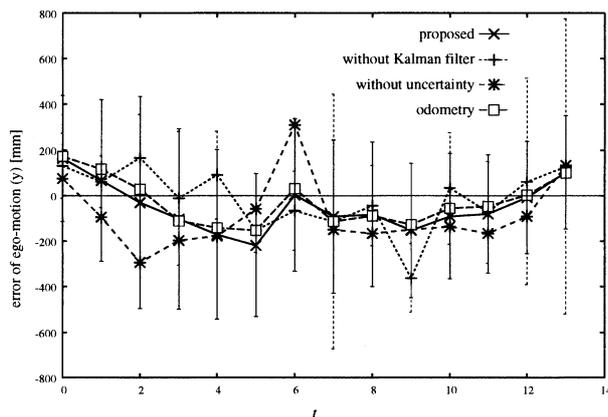


Fig. 16 Error of ego-motion estimation of robot (y)

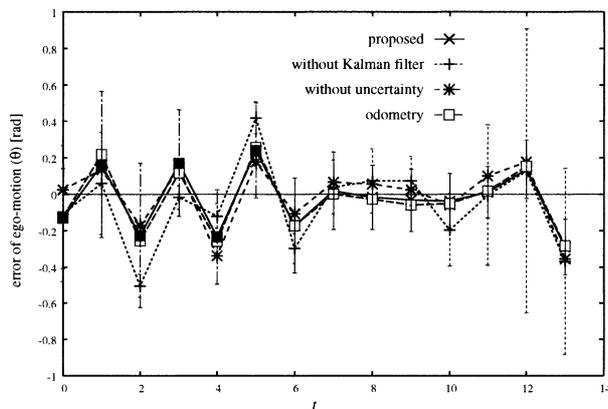


Fig. 17 Error of ego-motion estimation of robot (θ)

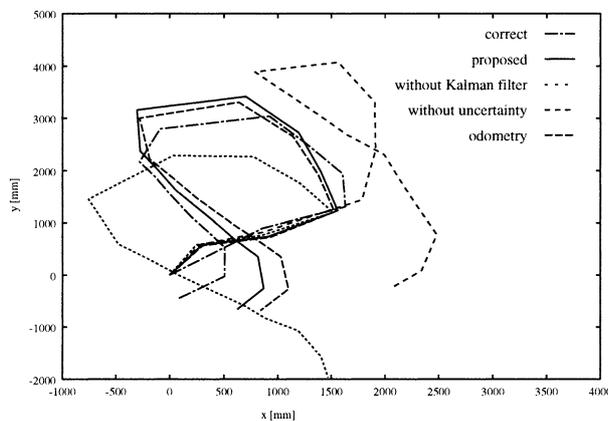


Fig. 18 Estimated position sequences

レオ視により得られる広い視野の情報を用いていることによるものと考えられる。一方で移動量に関しては、Table 1 と Fig. 18 をみれば分かるように、オドメトリに次いで提案手法が最も良い精度で推定できている。また、誤差の推定に関しても、提案手法ではほぼすべてのデータにおいて移動量、方向とも正しい値が推定誤差の範囲内に入っており、提案手法が有効に働いていることが分かる。本実験においては、前節のようにオドメトリに明示的に誤差を与えることはしていないため、距離誤差の大きい全方位ステレオ視を用いた移動量推定よりもオドメトリ

Table 1 Error of the ego-motion estimation

	x [mm]	y [mm]	θ [rad]
proposed	147.03	107.07	0.17
without Kalman filter	173.44	139.88	0.23
without uncertainty	274.42	156.5	0.18
odometry	145.62	104.1	0.18

による推定結果の方が全体的に誤差が少ないが、方向に関しては提案手法の方が誤差が少なくなった。前節の結果から、オドメトリの誤差がある程度大きい場合でも提案手法が有効であるので、提案手法は十分に有効であると考えられる。Fig. 14 におけるロボットの位置は実験での初期位置であり、初期位置の近辺では周囲の障害物との距離が大きいため、ステレオにより得られる距離の誤差も大きく、それにより推定されるロボットの移動量もまた誤差が大きくなると期待される。実験結果では初期位置近辺で不確かさが大きく出ており、推定された不確かさの大きさについても妥当なものと言える。唯一、Fig. 17 の最後のデータのみ正しい値が推定誤差範囲外であるが、このときは正しい値がオドメトリにより推定された不確かさの範囲外であった。そのため、正しい値近傍の情報が得られず推定に失敗した。これはオドメトリの不確かさのモデルを改善すれば修正できると考えられる。

5. おわりに

本論文では複数のスキャンマッチングの結果を不確かさを考慮して統合し、移動ロボットの自己移動量とその不確かさを推定する手法を提案した。本手法は現在のロボットの移動量を推定すると同時に、過去に推定されたロボットの移動量を再度推定しなおしている。本手法はカルマンフィルタを用いた逐次的な手法であるため、ロボットの現在の移動量と過去の $k-1$ 個の移動量を推定するのに、 $k+1$ 個の新たに計算されたスキャンマッチングの結果のみを必要とする。また、我々はこの手法を、 360° のほぼ全周の三次元的な距離情報が取得可能な全方位ステレオ視を用いた、不確かさを考慮したスキャンマッチング手法を用いて実装した。実際の環境において取得したデータを基に行った実験の結果から、提案手法が有効に働くことを示した。

提案手法は複数のスキャンマッチングの結果を用いていることと、スキャンマッチングにおいて明らかに対応しない観測をマハラノビス距離を用いて識別し、その情報の重みを減らすようにしているため、ある程度の移動物体の存在する環境においてもそのまま使用できると思われる。しかし、多くの移動物体が存在する環境では、ある移動物体と別の移動物体が対応するようなことが起きる可能性があり、それによって推定の精度が低下する可能性がある。そこで、移動物体をあらかじめ識別するなどして [8]、そのような動的な環境においてもロバストに推定できるようにすることが今後の課題である。

また、曲がり角のように、みえている物体が急激に変化する場所では、スキャンマッチングにおいて対応するデータの数が少なくなり、結果的に移動量推定の精度が低下する。そのような場所では、移動速度を落として対応点の数を減らさないようにする、といった対策が考えられる。

参考文献

- [1] H. Baltzakis and P. Trahanias: "Closing Multiple Loops while Mapping Features in Cyclic Environments," In Proc. of 2003 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, pp.717-722, 2003.
- [2] G. Dissanayake, H. Durrant-Whyte and T. Bailey: "A Computationally Efficient Solution to the Simultaneous Localization and Map Building (SLAM) Problem," In Proc. of IEEE Int. Conf. on Robotics and Automation, pp.1009-1014, 2000.
- [3] M. Etoh, T. Aoki and K. Hata: "Estimation of Structure and Motion Parameters for a Roaming Robot that Scans the Space," In Proc. of 7th Int. Conf. on Computer Vision, vol.1, pp.579-584, 1999.
- [4] J. Gluckman, S.K. Nayar and K.J. Thoresz: "Real-Time Omnidirectional and Panoramic Stereo," In Proc. of Image Understanding Workshop, vol.1, pp.299-303, 1998.
- [5] D. Hähnel, D. Schulz and W. Burgard: "Map Building with Mobile Robots in Populated Environments," In Proc. of 2002 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, pp.496-501, 2002.
- [6] D. Hähnel, D. Schulz and W. Burgard: "An Efficient Fast-SLAM Algorithm for Generating Maps of Large-Scale Cyclic Environments from Raw Laser Range Measurements," In Proc. of 2003 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, pp.206-211, 2003.
- [7] 城殿, 三浦, 白井: "誘導による移動経験に基づく視覚移動ロボットの自律走行", 日本ロボット学会誌, vol.19, no.8, pp.1003-1009, 2001.
- [8] H. Koyasu, J. Miura and Y. Shirai: "Realtime Omnidirectional Stereo for Obstacle Detection and Tracking in Dynamic Environments," In Proc. of IEEE Int. Conf. on Intelligent Robots and Systems, pp.31-36, 2001.
- [9] F. Lu and 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.
- [10] K. Nickels and S. Hutchinson: "Estimating uncertainty in SSD-based feature tracking," Image and Vision Computing, vol.20, pp.47-58, 2002.
- [11] S.T. Pfister, K.L. Kriechbaum, S.I. Roumeliotis and J.W. Burdick: "Weighted Range Sensor Matching Algorithms for Mobile Robot Displacement Estimation," In Proc. of IEEE Int. Conf. on Robotics and Automation, pp.1667-1674, 2002.
- [12] A. Singh and P. Allen: "Image-Flow Computation: An Estimation-Theoretic Framework and a Unified Perspective," Computer Vision Graphics and Image Processing: Image Understanding, vol.56, no.2, pp.152-177, 1992.
- [13] Y. Yagi, K. Shouya and M. Yachida: "Environmental Map Gen-

eration and Egomotion Estimation in a Dynamic Environment for an Omnidirectional Image Sensor," In Proc. of IEEE Int. Conf. on Robotics and Automation, pp.3493-3498, 2000.

- [14] 山澤, 八木, 谷内田: "移動ロボットのナビゲーションのための全方位視覚センサ HyperOmni Vision の提案", 電子情報通信学会論文誌 D-II, vol.J79-D-II, no.5, pp.698-707, 1996.

付録 A. ロボットの移動の不確かさ

ロボットの移動は二つの車輪の回転量 $U = (l, r)^T$ により制御される。ロボットの状態量の変化は次のように $F(\mathbf{X}_t, U_t)$ を用いて表す:

$$\mathbf{X}_{t+1} = \begin{pmatrix} x_t + \frac{W}{2} \frac{l_t+r_t}{l_t-r_t} (\cos \theta_t - \cos \theta_{t+1}) + L (\sin \theta_t - \sin \theta_{t+1}) \\ y_t + \frac{W}{2} \frac{l_t+r_t}{l_t-r_t} (\sin \theta_t - \sin \theta_{t+1}) - L (\cos \theta_t - \cos \theta_{t+1}) \\ \theta_t - \frac{l_t-r_t}{W} \end{pmatrix} = F(\mathbf{X}_t, U_t) \quad (\text{A.1})$$

ただし, W はロボットの後輪の幅を, L は二つの後輪の midpoint とロボット中心の距離をそれぞれ表す。

式 (A.1) を平均値 $\hat{\mathbf{X}}_t$ と \hat{U}_t のまわりで一次のテイラー展開をして線形化することにより, 状態量の推定誤差共分散行列 $\Sigma_{\mathbf{X}_{t+1}}$ が以下のように得られる:

$$\begin{aligned} \Sigma_{\mathbf{X}_{t+1}} &= E[(\mathbf{X}_{t+1} - \hat{\mathbf{X}}_{t+1})(\mathbf{X}_{t+1} - \hat{\mathbf{X}}_{t+1})^T] \\ &= \frac{\partial F(\mathbf{X}_t, U_t)}{\partial \mathbf{X}_t} \Sigma_{\mathbf{X}_t} \frac{\partial F(\mathbf{X}_t, U_t)}{\partial \mathbf{X}_t}^T \\ &\quad + \frac{\partial F(\mathbf{X}_t, U_t)}{\partial U_t} \Sigma_{U_t} \frac{\partial F(\mathbf{X}_t, U_t)}{\partial U_t}^T \end{aligned} \quad (\text{A.2})$$

ただし, Σ_{U_t} は入力 U_t の共分散行列である。ここで, 誤差 Σ_{U_t} が車輪の滑りのみによって生じると仮定し, 更に左右の車輪の誤差 $\sigma_{l_t}^2$ と $\sigma_{r_t}^2$ は正規分布であり互いに独立であるとすると, Σ_{U_t} は次のように書ける:

$$\Sigma_{U_t} = \begin{pmatrix} \sigma_{l_t}^2 & 0 \\ 0 & \sigma_{r_t}^2 \end{pmatrix} \quad (\text{A.3})$$

$\sigma_{l_t}^2$ と $\sigma_{r_t}^2$ は車輪の移動距離 l_t と r_t に比例すると考えられる。その比例定数は現在のところ不確かさの範囲を大きめにとることとして 0.3 を使用している。



子安大士 (Hiroshi Koyasu)

2000 年大阪大学工学部電子制御機械工学科卒業。2002 年同大学大学院工学研究科電子制御機械工学専攻修士課程修了。現在同大学大学院博士課程在学中。自律移動ロボットの研究に従事。2003 年日本ロボット学会研究奨励賞受賞。

(日本ロボット学会学生会員)



白井良明 (Yoshiaki Shirai)

1964 年名古屋大学工学部機械工学科卒業。1969 年東京大学大学院工学系博士課程修了。工学博士。同年, 電子技術総合研究所入所, コンピュータビジョン, ロボティクスの研究に従事。1971~1972 年, MIT AI ラボ客員研究員。1988 年大阪大学工学部電子制御機械工学科教授。人工知能学会, 電子情報通信学会, 情報処理学会, 日本機械学会各会員。

(日本ロボット学会正会員)



三浦 純 (Jun Miura)

1984 年東京大学工学部機械工学科卒業。1989 年同大学大学院工学系研究科情報工学専攻博士課程修了, 工学博士。同年大阪大学助手。現在同大学大学院工学研究科電子制御機械工学専攻助教授。知能ロボット, 人工知能, コンピュータビジョンの研究に従事。1994~1995 年 CMU 客員研究員。1997 年ロボット学会論文賞受賞。人工知能学会, 電子情報通信学会, 情報処理学会, 日本機械学会, IEEE, AAAI 各会員。(日本ロボット学会正会員)