

自車両位置認識結果の信頼度推定

赤井 直紀^{*}, Luis Yoichi MORALES^{*}, 平山 高嗣^{*}, 村瀬 洋[†]

Reliability Estimation for Self-Vehicle Pose Recognition Result

Naoki AKAI^{*}, Luis Yoichi MORALES^{*}, Takatsugu HIRAYAMA^{*}, and Hiroshi MURASE[†]

Abstract : This paper presents a reliability estimation method of localization results. In the method, an ego-vehicle pose and reliability are treated as hidden variables and are estimated simultaneously via Rao-Blackwellized particle filter (RBPF). The ego-vehicle pose is estimated by a sampling-based method, i.e., particle filter, and the reliability is estimated by an analytical method using prediction results of convolutional neural network (CNN). The CNN learns whether localization has failed or not and its output is used as an observable variable to estimate the reliability in the RBPF. Through experiments, it is shown that the estimated reliability could be used as an exact criterion for describing successful and fault localization results.

KEY WORDS: Localization, Reliability, Convolutional Neural Network, Rao-Blackwellized Particle Filter

1. はじめに

位置推定とは、与えられた地図上でセンサ（もしくはセンサを搭載した車両）の相対位置を求める問題である。多くの場合、地図とセンサ観測の間で対応を求め、その対応から作成される誤差関数を最小とする状態を求めることで位置を推定する [1]。もしくは、センサ観測をモデル化し、その観測が得られる最大の確率となる状態を求める方法をとる [2]。ここで重要なことは、最小の誤差、もしくは最大の確率となる状態が、数值的に尤もらしいという状態を意味するのみであり、正しい位置であるという保証がないことである。すなわち、位置推定の成功や失敗を自動運転車両自身が判断する術はなく、位置推定結果の保証が行えないということの意味する。

本稿では、位置推定結果の保証をすることを目的とし、その信頼度を求める問題を考える。このために、位置推定結果の正誤を判断するための機械学習アルゴリズムを導入する。しかしながら、機械学習による識別の結果が完全になるとは言い難い。この考えに基づき、機械学習による推定の正誤判断の結果を新たな可観測変数として利用し、位置推定結果の信頼度を推定するモデルを提案する。提案法では、車両の位置と上記信頼度を同時に求める問題を定式化し、Rao-Blackwellized particle Filter (RBPF) を用いてこの問題を解く。実機車両およびシミュレーションを用いて得られた実験結果より、推定された信頼度が位置推定結果の正誤を正しく説明するために利用できることを示す。

なお、本稿で述べられる手法の基礎部分は、文献 [3] にて提案されている。しかし従来法には、計算時間の課題があり、自動車のような高速な移動体へ実装することが困難であった。本稿の貢献は、計算時間の問題を解決し、実際の自動車で利用可能とした点にある。

2. 関連研究

前述の通り位置推定の問題を解くことで得られる結果は、数值的に尤もらしいという状態のみであり、1つ

位置推定の結果が手に入った場合に、その結果の正誤までは判断できない。これに対して有効な方法は、複数の位置推定法を用いた冗長なシステムを構築することである（例えば [4, 5]）。これにより、多数決のような方法を採用し、誤った位置推定結果を検知することができる。しかしながら、あくまで各々の位置推定結果に対する信頼度は定義されておらず、多数決の結果が正しい結果となるとは言えない。

位置推定の問題を解くために、よく Monte Carlo localization (MCL) が利用される [2]。MCL による位置推定の失敗状態を検知するために、Gutmannらは augmented MCL (AMCL¹) を提案した [6]。AMCL では、尤度の履歴を監視することで、位置推定状態の失敗を検知する。しかしながら、尤度はセンサ観測をモデル化することで得られるものであり、かつ動的な環境で確実なセンサ観測のモデル化を行うことは困難であるため [7, 8]、動的環境における AMCL による失敗検知が、必ずしも妥当になるとは言い難い。

近年では、機械学習を用いて位置推定の失敗を検知する取組も報告されている [9, 10, 11]。この様な例は、GNSS を用いた方法に対しても適用例がある [12]。しかしながら、機械学習による識別の性能が完全になるとは言い難いが、これらの方法では機械学習の結果を直接利用している。一方で提案法では、機械学習による識別の結果が不完全であることを前提とし、位置推定結果の信頼度を隠れ変数と見なして推定する方法を用いる。そのため、機械学習の識別に誤りが含まれる場合においても、安定して信頼度を推定できる。

本稿で述べる方法の基礎的な部分は、文献 [3] において提案されている。文献 [3] の手法では、画像ベースのデータを畳み込みニューラルネットワーク (convolutional neural network: CNN) に入力して、位置推定の成功・失敗の判断を行っていた。この画像データの作成、および CNN による計算時間のコストが大きいため、自動車のような高速な移動体へ実装が困難であった。本稿における提案は、CNN への入力データの

^{*}名古屋大学未来社会創造機構 (464-8601 名古屋市千種区不老町)

[†]名古屋大学大学院情報学研究所

¹多くの場合 AMCL と言うと adaptive MCL を表すが、本稿では augmented MCL を表すこととする。

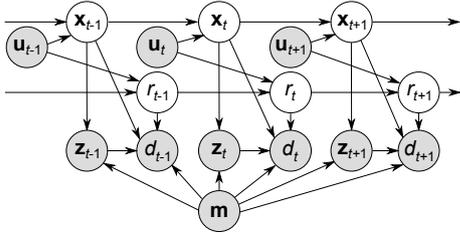


Fig.1 Graphical model for the proposed model [3].

改善を行い、識別性能の低下がない状態での計算速度の向上を実現する。結果として、約 50 km/h で移動する自動車に搭載して利用可能となることを示す。

3. 位置と信頼度の同時推定

3.1 位置推定における信頼度の定義

工学分野において信頼度は、「対象が与えられた条件で、規定の期間中、要求された機能を果たすことが出来る性質を確率で表現したもの」と定義される。通常自動運転においては、位置推定結果、すなわち推定された位置情報は、他のモジュールに利用される（例えば物体認識や走行制御など）。そのため、使用する他のモジュールに従い、位置推定結果に対する許容誤差が定義されるといえる。そこで本稿では、この許容誤差内に位置推定結果が含まれる場合を、位置推定が要求された機能を果たしている（成功）状態であると定義する。つまり位置推定結果に対する信頼度は、この許容誤差内に位置推定結果が含まれる確率を意味する。

位置推定に成功・失敗している状態は r を用いて表す。 $r = 1$ が成功、 $r = 0$ が失敗状態を表すものとする。すなわち信頼度は $p(r = 1)$ と表記される²。

3.2 グラフィカルモデル

Fig. 1 には、提案法で利用されるグラフィカルモデルを示す。本モデルでは、自車両位置 \mathbf{x} と信頼度 r が隠れ変数として扱われている。これらの隠れ変数を、センサ観測 \mathbf{z} 、制御入力（もしくは移動量） \mathbf{u} 、地図 \mathbf{m} 、および学習器による位置推定結果の正誤判断 d の可観測変数を用いて同時に推定する。なお正誤判断 d は、位置推定に成功している確率を表す変数である³。

我々の目的は、以下に示す時刻 t の自己位置 \mathbf{x}_t と信頼度 r_t の同時確率分布を求めることである。

$$p(\mathbf{x}_t, r_t | \mathbf{z}_{1:t}, \mathbf{u}_{1:t}, \mathbf{m}, d_{1:t}) \quad (1)$$

ここで $1:t$ は、時刻 1 から t の時系列を表す。まず上式に乗法定理を適用して次式を得る。

$$p(\mathbf{x}_t | \mathbf{z}_{1:t}, \mathbf{u}_{1:t}, \mathbf{m}, d_{1:t}) p(r_t | \mathbf{x}_t, \mathbf{z}_{1:t}, \mathbf{u}_{1:t}, \mathbf{m}, d_{1:t}) \quad (2)$$

本研究では、Rao-Blackwellized particle filter (RBPF) を用いて上述の同時確率分布を求めることとし、上式第 1 項の位置に関する分布をサンプリング (PF) により、第 2 項の信頼度に関する分布を解析により求める。

²文献 [3] では、 r を 0 から 1 の連続値として定義し、 $\int r p(r) dr$ を信頼度として定義している。

³正誤判断 d は信頼度 $p(r = 1)$ と同様の意味を持つが、 d は可観測変数であり、 r は d を基に推定される変数である。

まず第 1 項の分布に着目する。本モデルでは、位置に依存する可観測変数としてセンサ観測 \mathbf{z}_t と正誤判断 d_t が利用可能であるため、ベイズの定理を 2 度適用することができる。

$$p(\mathbf{x}_t | \mathbf{z}_{1:t}, \mathbf{u}_{1:t}, \mathbf{m}, d_{1:t}) = \eta p(\mathbf{z}_t | \mathbf{x}_t, \mathbf{m}) p(d_t | \mathbf{x}_t, \mathbf{z}_t, \mathbf{m}) p(\mathbf{x}_t | \mathbf{z}_{1:t-1}, \mathbf{u}_{1:t}, \mathbf{m}, d_{1:t-1}) \quad (3)$$

ここで η は正規化係数を表す。また上述の式変形の際には、マルコフ性を適用している。次に、上式に全確率の定理を適用する。

$$\eta p(\mathbf{z}_t | \mathbf{x}_t, \mathbf{m}) \int p(d_t | r_t, \mathbf{x}_t, \mathbf{z}_t, \mathbf{m}) p(r_t) dr_t \int p(\mathbf{x}_t | \mathbf{x}_{t-1}, \mathbf{u}_t) p(\mathbf{x}_{t-1} | \mathbf{z}_{1:t-1}, \mathbf{u}_{1:t-1}, \mathbf{m}, d_{1:t-1}) d\mathbf{x}_{t-1} \quad (4)$$

上述の確率分布を PF によって推定する場合、 $p(\mathbf{z}_t | \mathbf{x}_t, \mathbf{m})$ と $\int p(d_t | r_t, \mathbf{x}_t, \mathbf{z}_t, \mathbf{m}) p(r_t) dr_t$ の 2 つの分布を用いてパーティクルの尤度を計算することになる。1 つ目の分布は観測モデルとして知られている [2]。2 つ目の分布に関しては、次章に詳細を述べる。

次に式 (2) の第 2 項に着目する。上述の式展開と同様に、ベイズの定理とマルコフ性を適用する。

$$p(r_t | \mathbf{x}_t, \mathbf{z}_{1:t}, \mathbf{u}_{1:t}, \mathbf{m}, d_{1:t}) = \eta p(d_t | r_t, \mathbf{x}_t, \mathbf{z}_t, \mathbf{m}) p(r_t | \mathbf{x}_t, \mathbf{z}_{1:t-1}, \mathbf{u}_t, d_{1:t-1}) \quad (5)$$

次に、上式に全確率の定理を適用する。

$$\eta p(d_t | r_t, \mathbf{x}_t, \mathbf{z}_t, \mathbf{m}) \int p(r_t | r_{t-1}, \mathbf{u}_t) p(r_{t-1} | \mathbf{x}_{t-1}, \mathbf{z}_{1:t-1}, \mathbf{u}_{1:t-1}, d_{1:t-1}) dr_{t-1} \quad (6)$$

ここで $p(r_t | r_{t-1}, \mathbf{u}_t)$ は、車両の移動に伴う信頼度の変化を表す分布であり、 $p(d_t | r_t, \mathbf{x}_t, \mathbf{z}_t, \mathbf{m})$ は、信頼度を推定するための尤度分布である。一般的に、車両の移動に伴い位置推定の誤差は増大するため、信頼度は減衰すると考えられる。そのため $p(r_t | r_{t-1}, \mathbf{u}_t)$ を信頼度減衰モデル (reliability decay model) と呼ぶ。また $p(d_t | r_t, \mathbf{x}_t, \mathbf{z}_t, \mathbf{m})$ は、位置推定結果に対する正誤判断に対する尤度分布であるため、判断モデル (decision model) と呼ぶ。これらの詳細は次章にて述べる。

3.3 同時推定の利点

式 (4) に示す様に、本手法ではパーティクルの尤度計算を 2 種類の尤度分布を用いて行う。 $\int p(d_t | r_t, \mathbf{x}_t, \mathbf{z}_t, \mathbf{m}) p(r_t) dr_t$ を用いて尤度計算を行うことで、同時推定による利点を得ることになる。この分布は、信頼度が高い場合に位置推定結果が正しいと判断された場合、または、信頼度が低い場合に位置推定結果が誤っていると判断された場合に高い尤度をとる。すなわち本手法では、位置推定結果に対する正誤判断が突発的に誤った場合に、パーティクルの尤度が低下することとなる。そのため、正誤判断結果 d がノイズを含む場合のロバスト性を向上させることになる。

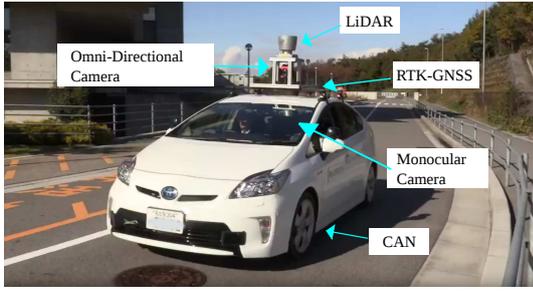


Fig.2 Experimental platform.

4. 実装

4.1 実験車両

Fig. 2には、本研究で使用した車両を示す。前章で述べた同時推定の問題を解くために、車両上部のLiDAR (HDL-64E)とCANによる移動量の情報を用いた。なお使用したLiDARは3次元の計測が可能であるが、実験の評価の容易さなどを考慮し、2次元のスキャンを仮想的に作りだして利用することとした。スキャンデータの仕様は、最大計測距離80 m、計測角度190 degree、角度分解能0.125 degreeとした⁴。

4.2 位置推定結果の正誤判断の学習

文献[3]で提案された手法では、画像データをCNNに入力し、位置推定の正誤判断の回帰を行っている。入力画像の例をFig. 4に示す。画像データの作成、および処理の計算コストが高く、従来法を自動車に適用することが困難であった。提案法では、軽量の計算プロセスで使用できるデータを用いることで、性能の低下なく計算速度を向上させる。

CNNに入力されるデータを作成するにあたり、文献[2]で述べられている観測モデルを参考とする。文献[2]では、高速にパーティクルの尤度を計算するために、尤度場モデル (likelihood field model: LFM) が提案されている。LFMによる尤度計算では、センサの距離データ r_z とその計測が当たった地点から最も近い地図上に存在する障害物までの距離 d_z が用いられる。CNNに入力されるデータ D は以下の様になる。

$$D = (\mathbf{z}_0, \mathbf{z}_1, \dots, \mathbf{z}_K) \quad (7)$$

$$\mathbf{z}_i = (r_{z_i}, d_{z_i})^T \quad (8)$$

ここで K はセンサ観測の数を表す。このデータは、横 K 、高さ1、チャンネル数2の画像データと見なされる。なおCNNの構造を作成するにあたっては、文献[13]で提案されている2チャンネルモデルを参考とした。

学習データの作成にあたり、まず3次元地図を作成し、その地図に基づいて3次元点群を用いた位置推定 (3D NDT [14, 15, 16]) を行った。その結果のマッチングにずれがないことを目視で確認し、その際の2次元の位置・姿勢の状態を位置推定成功状態とし、ラベル $y_i = 1$ を付与した。この状態に対して故意的にノイズを加えた状態を作成し、元の推定位置とのずれ量が一定以

⁴スキャンの作成にあたり以下のセンサの仕様を参考とした (<https://en.manu-systems.com/HOK-UXM-30LAH-EWA.shtml>)。

上になった状態を失敗状態とし、ラベル $y_i = 0$ を付与した。ずれ量に関する閾値は、位置誤差 $\sqrt{\Delta x^2 + \Delta y^2}$ が50 cm、角度誤差 $\Delta \theta$ が3 degreeとした。CNNは以下に示すロス関数を最小にするように学習を行う。

$$\frac{\lambda}{2} \|\omega^{\text{net}}\|_2 + \sum_{i=1}^N (y_i - d_i)^2 \quad (9)$$

ここで λ は正則化項に対する係数、 ω^{net} と d_i はネットワークの重みと出力、 N は学習データ数、 $y_i \in \{0, 1\}$ はラベルである。出力層にはシグモイド関数を活性化関数として利用し、 d の値域を0から1とした。

4.3 信頼度減衰モデル

信頼度減衰モデル $p(r_t | r_{t-1}, \mathbf{u}_t)$ では、時刻 t の信頼度は、時刻 $t-1$ の信頼度と時刻 t の制御入力 \mathbf{u}_t に依存する。しかしながら、これらの関係を厳密にモデル化することは困難である。本研究では、以下に示すヒューリスティックな関係式を利用することとした。

$$p(r_t = 1) = (1 - (\alpha_1 \Delta d_t^2 + \alpha_2 \Delta \theta_t^2)) p(r_{t-1} = 1) \quad (10)$$

ここで α_1 と α_2 は任意係数、 $\mathbf{u}_t = [\Delta d_t, \Delta \theta_t]^T$ はCANより得られる進行方向と角度方向に関する移動量である。

4.4 判断モデル

判断モデル $p(d_t | r_t, \mathbf{x}_t, \mathbf{z}_t, \mathbf{m})$ は、信頼度に対する位置推定の正誤判断の尤度分布として定義される。本研究では、この分布を以下の様にモデル化する。

$$p(d_t | r_t, \mathbf{x}_t, \mathbf{z}_t, \mathbf{m}) = \begin{pmatrix} d_{\text{pos}} \\ d_{\text{neg}} \end{pmatrix}^T \cdot \begin{pmatrix} p_{\text{pos}}(d_t | r_t, \mathbf{x}_t, \mathbf{z}_t, \mathbf{m}) \\ p_{\text{neg}}(d_t | r_t, \mathbf{x}_t, \mathbf{z}_t, \mathbf{m}) \end{pmatrix} \quad (11)$$

ここで d_{pos} と d_{neg} は任意の係数であり、 $d_{\text{pos}} + d_{\text{neg}} = 1$ の条件を満たす。また p_{pos} と p_{neg} は正誤判断が成功、および失敗する際の確率分布を表す。本研究では、 p_{pos} と p_{neg} を以下の確率分布により定義した。

$$p_{\text{pos}}(d_t | r_t, \mathbf{x}_t, \mathbf{z}_t, \mathbf{m}) = \frac{d_t^{a-1} (1 - d_t)^{b-1}}{B(a, b)} \quad (12)$$

$$p_{\text{neg}}(d_t | r_t, \mathbf{x}_t, \mathbf{z}_t, \mathbf{m}) = \text{unif}(0, 1) \quad (13)$$

ここで $B(\cdot)$ と $\text{unif}(\cdot)$ はベータ関数と一様分布を表す。 a と b の値は実験的に決定することとし、 $a = 5, b = 1$ (if $r_t = 1$)、および $a = 1, b = 5$ (if $r_t = 0$) をそれぞれ用いることとした。 d_{pos} と d_{neg} に関しては、次章で述べる事前実験の結果 (5.2) を参考にし、それぞれ $d_{\text{pos}} = 0.88, d_{\text{neg}} = 0.12$ とした。

4.5 RBPF

パーティクルの状態 \mathbf{s} は2次元空間の車両位置と信頼度とし、 $\mathbf{s} = [x, y, \theta, p(r = 1)]^T$ とする。提案するRBPFでは、車両位置をPFにより求め、信頼度を解析計算により求める。時刻 t における i 番目のパーティクルの重み ω_t^i は、次式により計算する。

$$\omega_t^i = p(\mathbf{z}_t | \mathbf{x}_t, \mathbf{m}) \int p(d_t | r_t, \mathbf{x}_t, \mathbf{z}_t, \mathbf{m}) p(r_t) dr_t \omega_{t-1}^i \quad (14)$$

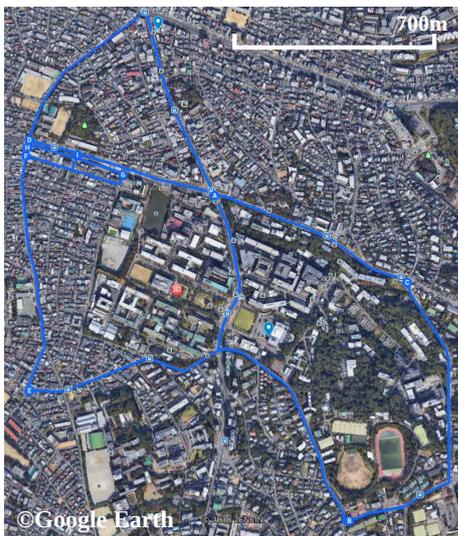


Fig.3 Experimental environment.

なお実装では、パーティクル数 M は 200(固定)とした。また信頼度が 95 % を低下した際に、位置推定結果が失敗していると判断し、パーティクルの分布を現在の推定位置を中心にして膨張させることとした [17].

5. 実験

5.1 実験環境

Fig. 3に、本研究で使用した実験環境を示す。本実験で複数回のデータ収集を行い、それぞれ別のデータを用いて位置推定の正誤判断の学習と実験を行った。本経路の1周の全長はおよそ 17 km であり、実験時の車両の最大速度は約 50 km/h である。

5.2 位置推定正誤判断における入力データの比較

位置推定正誤判断を行うにあたり、入力データによる判断性能の比較をまず行った。文献 [2] では、PF の尤度計算のためにビームモデル (beam model: BM) も提案されている。BM では、LFM で使用する情報に加え、センサ位置からレーザービーム方向の障害物までの距離 m_z も利用される。今回の比較では、LFM, BM, および文献 [3] で利用される画像ベースのデータを用いて比較を行った。なお、画像データは解像度 0.1 m で 800×500 の大きさで作成し、CNN に入力する前に 160×100 に縮小した。入力データの例を Fig. 4 に示す。

学習用のデータとして 27362 個のデータを作成した (13681 個の位置推定成功・失敗データをそれぞれ含む)。まずこれらのデータの平均の作成時間を計測した。データ作成の処理は CPU (Intel(R) Xeon(R) CPU E5-1650 v3 @ 3.50 GHz) 上で行った。Table 1 にそれぞれのデータ作成に要した平均時間を示す。BM ではセンサ位置から障害物までの距離を計算するためにレイキャスティングを行う必要があるため、LFM に比べて計算時間が長くなっている。

次にテスト用のデータを 22638 個作成した (11319 個の位置推定成功・失敗データをそれぞれ含む)。Table 1 には、それぞれ 1 つのデータを CNN により処理するのに要した平均時間も示されている。CNN の処理は

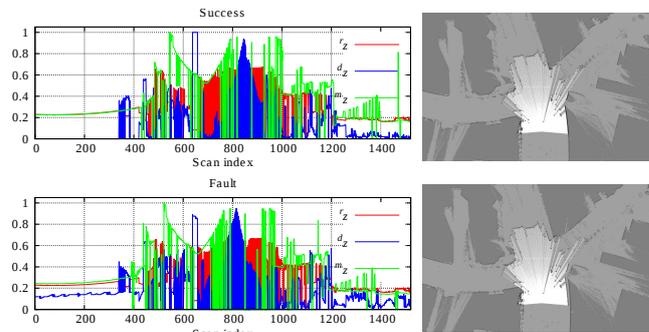


Fig.4 Input data examples of LFM-, BM-, and image-based data. Top and bottom data show successful and fault cases of data, respectively.

Table 1 Average data creation and prediction times per one data.

	Data creation time (msec)	Prediction time (msec)
LFM	0.006	0.390
BM	3.953	0.397
Image	13.545	0.861

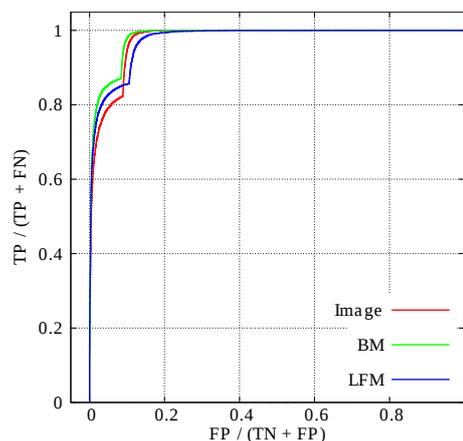


Fig.5 ROC curves. TP, FP, FN, and TN denote true positive, false positive, false negative, and true negative, respectively.

GPU (GeForce GTX TITAN X) 上で行った。Table 1 から、LFM に基づくデータの処理が他のデータに比べて高速に行えていることが確認できる。

また Fig. 5 には、それぞれのデータによる正誤判断の性能を評価した際の receiver operating characteristic (ROC) 曲線を示す。CNN は 0 から 1 の連続値を出力するため、0.5 を閾値とし、出力値が 0.5 以上となった場合を成功と判断している、それ以外の場合を失敗と判断しているものとした。Fig. 5 からわかるように、どのデータでも同様の性能が得られていることが確認できる。なお分類精度はそれぞれ、LFM 87.58 %, BM 89.33 %, Image 86.68 % となった。以上の結果から、提案法で用いられる LFM に基づくデータを利用することで、著しい性能の低下なく、計算を高速化できることが確認できた。

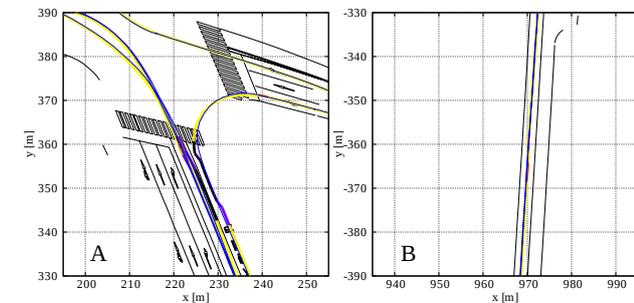
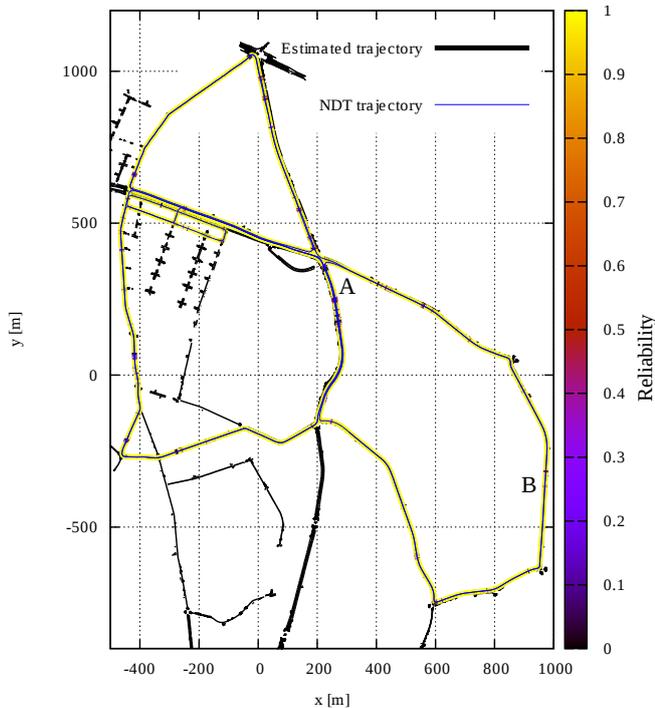


Fig.6 Estimated trajectories by the proposed method and 3D NDT scan matching. Color level denotes the estimated reliability.

5.3 走行実験

Fig. 6に、提案法によって推定された走行軌跡と信頼度、および3次元点群を用いた位置推定法により推定された走行軌跡をそれぞれ示す⁵。提案法では2次元のセンサ観測のみを用いて位置推定を行っているため、位置推定の精度は3D NDTに比べて精度が劣る状況がある。特にFig. 6Aでは、提案法により推定された軌跡が3D NDTによる推定軌跡から離れていることが確認できる。提案法により推定された信頼度は、この状況で95%を下回ったため、パーティクルの分布が膨張された。結果として、位置推定の失敗状態から復帰し、正しい位置の推定が可能となった。

信頼度の低下した状況の例をFig. 7に示す。Fig. 7左は信頼度が低いと判断された状況である。この地点には傾斜があるため、2次元のセンサ観測のみを用いて、時系列の位置推定問題を解くと、推定の失敗が頻発する。特に黄色枠で囲んだ領域を見てわかるように、ランドマーク(黒)とセンサ観測(赤)が適切に照合され

⁵本稿では3D NDTによって推定された軌跡を位置推定結果の真値と仮定している

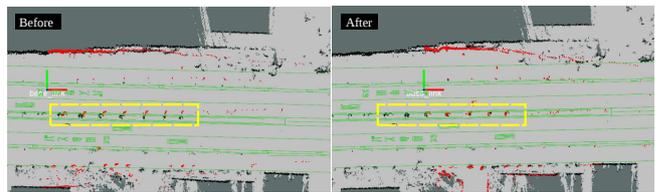


Fig.7 Example of unreliable scene detection and recovery performance.

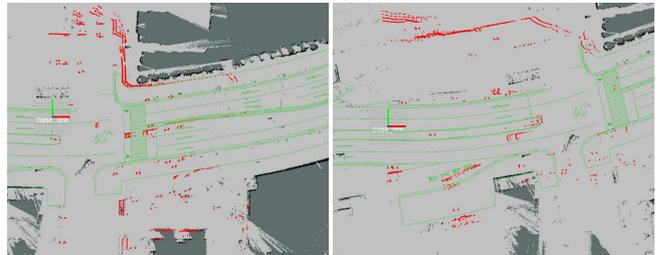


Fig.8 Miss-recognition cases where the localization result is unreliable.

ていないことが確認できる。提案法は、この状況を信頼度が低い状態であると判断し、パーティクルの分布の膨張処理を行った。結果として、Fig. 7右に示す様に、位置推定の成功状態へ復帰することができた。

しかし一方で、Fig. 6Bに示す様に、位置推定に成功している(3D NDTによる推定軌跡に近い)状態においても信頼度が低下するような状況があった。Fig. 8には、その様な状況の例を示す。これらの状況はランドマークとなりえる物体が少ない環境を走行しており、かつ路面のようなランドマークとして地図化されていない障害物を多くセンサが観測している状況である。CNNによる位置推定の成功・失敗の学習は、おおよそセンサ観測とランドマークの照合の失敗を学習しているといえる[3]。そのためこのような状況では、CNNによる正誤判断自体がうまく機能せず、誤った信頼度を推定してしまうこととなった。

5.4 シミュレーション実験

シミュレーションを用いて推定された信頼度 r_t 、最尤パーティクルの状態に対するCNNの正誤判断結果 d_t をFig. 9上段に、位置推定結果の位置と角度の誤差をFig. 9中段、および推定された軌跡と真値をFig. 9下段にそれぞれ示す。この実験では、途中で位置推定に失敗するパラメータをわざと利用しているため、移動に伴い誤差は増大する。しかし提案法を用いることで、誤差が増大した状態で信頼度が低下し、位置推定結果の失敗を早期に検知することができた。この失敗が検知された時点でパーティクルの分布を膨張させたため、最後まで位置を追跡することができた。

最尤パーティクルに対するCNNの正誤判断結果は、推定誤差が小さい場合でも振動していることが確認できる。これはCNNによる識別が完全に正しい結果となっていないためである。一方で提案法により推定された信頼度は、誤差が上昇した状態を信頼度が低い状

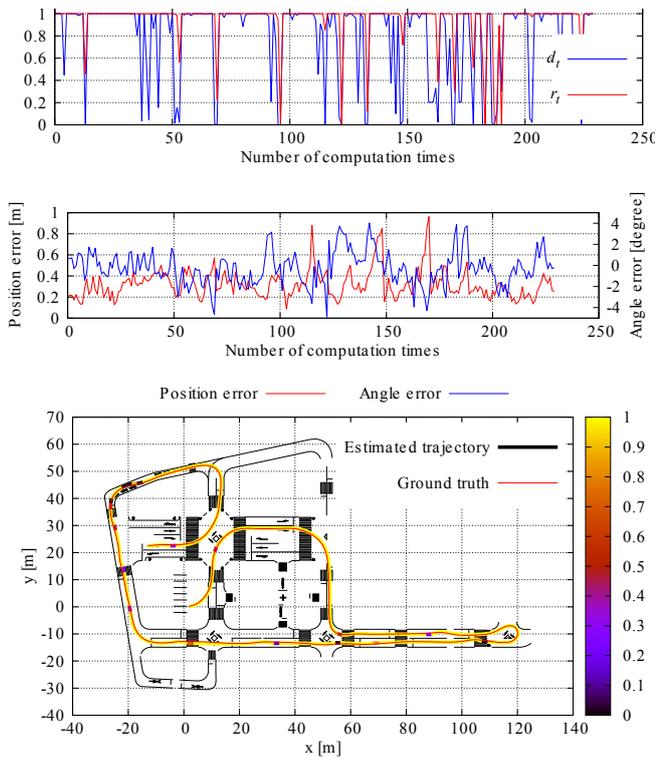


Fig.9 Simulation result with incorrect localization parameters.

態であると正確に認識している。CNNの結果を直接利用するよりも、提案法を用いることで安定した推定が可能となることが確認できた。

6. おわりに

本稿では、位置推定結果に対する信頼度を求める問題に関して議論した。基礎部分に関しては、すでに文献 [3] で提案しているが、計算時間の問題があった。これに対して本稿では、データの改善を行うことで、高速化する手法を提案した。これにより、時速 50 km/h で移動する自動車の位置と信頼度を推定することを実現し、その有用性を実機、およびシミュレーション実験により確認した。実験結果から、提案法により推定された信頼度が、位置推定結果の正誤を正しく説明できることを示した。

今後は、本稿で述べた高精度かつ高信頼度な位置推定機能を活用することで、文献 [18] で述べられているような、死角のある交差点における安全な行動選択に関する研究を行う予定である。

謝辞

本研究は国立研究開発法人科学技術振興機構の研究成果展開事業「センター・オブ・イノベーションプログラム (名古屋 COI: 高齢者が元気になるモビリティ社会)」および人工知能研究振興財団の支援により行われた。

参考文献

[1] P. J. Besl et al. A method for registration of 3-d

shapes. *IEEE Trans. Pattern Anal. Mach. Intell.*, 14(2):239–256, February 1992.

[2] S. Thrun et al. *Probabilistic Robotics*. The MIT Press, 2005.

[3] N. Akai et al. Simultaneous pose and reliability estimation using convolutional neural network and rao-blackwellized particle filter. *Advanced Robotics*, (accepted).

[4] P. Sundvall et al. Fault detection for mobile robots using redundant positioning systems. In *Proc. IEEE ICRA*, pages 3781–3786, 2006.

[5] J. P. Mendoza et al. Mobile robot fault detection based on redundant information statistics. In *Proc. IEEE/RSJ IROS*, 2012.

[6] J. Gutmann et al. An experimental comparison of localization methods continued. In *Proc. IEEE/RSJ IROS*, pages 454–459, 2002.

[7] N. Akai et al. Mobile robot localization considering class of sensor observations. In *Proc. IEEE/RSJ IROS*, 2018 (accepted).

[8] N. Akai et al. Toward localization-based automated driving in highly dynamic environments: Comparison and discussion of observation models. In *Proc. IEEE ITSC*, 2018 (accepted).

[9] Z. Alsayed et al. Failure detection for laser-based slam in urban and peri-urban environments. In *Proc. IEEE ITSC*, pages 126–132, 2017.

[10] Z. Alsayed et al. 2d slam correction prediction in large scale urban environments. In *Proc. IEEE ICRA*, pages 5167–5174, 2018.

[11] S. Nobili et al. Predicting alignment risk to prevent localization failure. In *Proc. IEEE ICRA*, pages 1003–1010, 2018.

[12] L. T. Hsu. Gnss multipath detection using a machine learning approach. In *Proc. IEEE ITSC*, pages 1414–1419, 2017.

[13] S. Zagoruyko et al. Learning to compare image patches via convolutional neural networks. In *Proc. IEEE CVPR*, pages 4353–4361, 2015.

[14] E. Takeuchi et al. A 3-d scan matching using improved 3-d normal distributions transform for mobile robotic mapping. In *Proc. IEEE/RSJ IROS*, pages 3068–3073, 2006.

[15] N. Akai et al. Robust localization using 3d ndt scan matching with experimentally determined uncertainty and road marker matching. In *Proc. IEEE IV*, pages 1357–1364, 2017.

[16] N. Akai et al. Autonomous driving based on accurate localization using multilayer lidar and dead reckoning. In *Proc. IEEE ITSC*, pages 1147–1152, 2017.

[17] R. Ueda et al. Expansion resetting for recovery from fatal error in monte carlo localization-comparison with sensor resetting methods-. In *Proc. IEEE/RSJ IROS*, pages 2481–2486, 2004.

[18] L. Y. Morales et al. Proactive driving modeling in blind intersections based on expert driver data. In *Proc. IEEE IV*, pages 901–907, 2017.