

完全自動運転実現のための信頼度付き自己位置推定の提案*

○赤井直紀 平山高嗣 村瀬洋 (名古屋大学)

概要 経路計画などの完全自動運転に必要な機能の多くは、自己位置推定結果が正しいものと仮定して実行されている。そのため、位置推定の失敗は自動運転の失敗にも直結しうるが、現存する位置推定機能は、自身の推定失敗を検知する機能を有さない。すなわち、自動運転の性能保証を行うことが極めて困難な課題となっている。本稿では、従来の位置推定法とは異なる信頼度付き位置推定法を提案し、明示的に推定結果の正しさが得られるようになることを示す。そして、得られた信頼度が自動運転車両の位置推定結果に対する正誤を適切に説明できることを示すことで、提案法が自動運転の性能保証に貢献できる可能性を示す。

キーワード: 自己位置推定, 信頼度, 完全自動運転, パーティクルフィルタ, ラオ・ブラックウェル化

1 はじめに

完全自動運転(以下, 自動運転)を実現するにあたり, 車両自身が高精度地図上のどこに位置するかを知る技術, すなわち自己位置推定(以下, 位置推定)は極めて重要な技術となる。特に, 障害物検知や経路計画などの技術が, 位置推定結果が正しいものと仮定して実行されていることが多い(例えば [3]), 位置推定の失敗は自動運転の失敗にも直結しうる。すなわち, 位置推定の正誤を正しく予測することが自動運転の性能を保証することにも繋がるため, 自動運転実現のためにも, 位置推定の正誤予測は必須技術であるといえる。しかしながら, 現存する位置推定機能の多くは, 自身の推定失敗を検知する機能を有さない。これは, 位置推定を実行するにあたり, 「観測の独立性」という仮定を適用していることに起因する [4]。

本稿では, 従来の位置推定とは異なる枠組みである「信頼度付き位置推定」を提案する。本手法の詳細, および従来法との違いに関しては3章で詳細を述べる。本手法は, 文献 [5] にて初めて提案した手法であるが, 計算時間に課題があった。特に, 畳み込みニューラルネットワーク(以下, CNN)が利用されており, これに関する処理が計算時間のボトルネックとなっていた。本稿では, CNNに入力するデータを改善することで, 性能の低下なしに計算時間の高速化を実現し, 自動車に搭載可能とした信頼度付き位置推定について報告する。

2 関連研究

位置推定はロボティクスにおいて活発に研究されてきたトピックであり, 特に環境変化に対する頑健性を向上させるための研究が多く行われてきた [6]。最も簡単に頑健性を向上させる方法としては, 歩行者や車両などの動的特性の高い物体を外れ値として検知し, 位置推定時に利用しないことである [7, 8]。しかしこのような方法で, 駐車車両の様な準静的な物体に対処することは難しい。これに対して, オンラインで地図を更新しながら, 逐次地図を環境に適合させつつ位置推定を行う方法が提案されている [9-14]。また, 異なる時間尺度, すなわちオンラインとオフラインで構築された地図を用いて位置推定を行う方法も提案されている [15, 16]。これらの手法は, 環境変化に適応した位置推定の実行を可能とさせるが, オンラインの地図構築に失敗すると, 位置推定にも失敗してしまう。すなわち, システム信頼性の観点から見ると, 位置推定単体

を適用する場合よりも信頼性が低くなってしまう。

Gutmannらは, モンテカルロ位置推定 [6] における位置推定の失敗検知方法を実験的に提案した [17]。この手法は, 観測モデル(センサ観測値と地図を照合させる際に用いるモデル)により計算される尤度の履歴を監視することで, 位置推定に失敗したかどうかを判定する。しかし, 一般に観測モデルが環境の変化に完全に適応できる保証はなく, 尤度の履歴監視による失敗検知が正しく機能する保証はない。これに対して, 観測モデルを環境変化に対してロバストに機能させることを目的とした研究も報告されている [18, 19]。また Yangらは, Feasibility grids と呼ばれるものを導入し, 地図に障害物の属性を持たせることで, 観測モデルによる尤度計算のロバスト性を向上させる方法を提案している [20]。また著者らも, センサ観測のクラス, 例えば観測物体の地図上での有無などを位置と同時推定することで, 位置推定がよりロバストに行えるようになる方法を提案している [21, 22]。しかし, ロバスト性がどれだけ向上したとしても, 明示的な推定結果の正しさを表すような指標を得ることはできない。また繰り返しになるが, これらの高度化された観測モデルを Gutmannらの手法と併用したとしても, 観測モデルが環境変化に完全に適応できる保証はないため, 失敗検知の正しさの有無も保証できない。

複数の位置推定法を同時に利用することで, 位置推定の失敗を多数決的に検知する方法も提案されている [23, 24]。しかしこの様な冗長システムの多くは, 多数決的な枠組みのみで推定の正誤を判断するため, 推定に失敗しているものを正しいと判断することも有り得る。

近年の機械学習の発展に伴い, 位置推定の失敗検知のために機械学習を適用する例も報告されている [25-28]。これらは, 従来の失敗検知方法と比較して, 高い精度での検知を実現している。しかし当然ながら, 推定結果の正誤の判断が完全になることはない。本稿で提案する信頼度付き位置推定法は, 同様に機械学習を用いて推定結果の正誤判断を行うが, その判断結果の不確かさまで考慮することを可能とするモデルを提案している。そして, 従来の位置推定問題とは異なる問題の定式化を行っている。これにより, 機械学習による識別が失敗した場合にも, その影響を低減させる効果を持たせることができる。

3 信頼度付き位置推定

図1に, 通常的位置推定(上)と信頼度付き位置推定(下)問題におけるグラフィカルモデルをそれぞれ

*本研究は自動車技術会 2018 年秋季大会で発表し [1], かつ自動車技術会論文集へ同時投稿したもの [2] である。

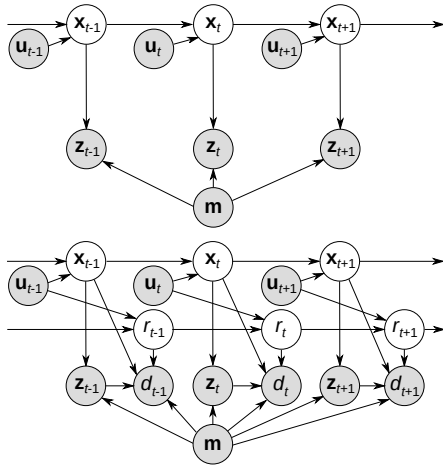


Fig. 1: グラフィカルモデル。(上) 通常の位置推定モデル, (下) 提案する信頼度付き位置推定モデル [2].

示す。グラフィカルモデルでは、未知変数が白、可観測変数が灰色で表記される。また、これらの変数の依存関係が矢印で表され、「矢印の先の変数が矢印の根本の変数に依存している」ことを意味する。本章では、通常の位置推定・信頼度付き位置推定モデルの違い、および後者のモデルの利点をそれぞれ説明する。

3.1 通常の位置推定モデル

通常の位置推定のモデルでは、位置 \mathbf{x} が未知変数として扱われ、制御入力 \mathbf{u} 、センサ観測 \mathbf{z} 、および地図 \mathbf{m} が可観測変数として扱われる。そして、次式に示す様に、時刻 t における位置に関する事後分布を求めることを目標とする。

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

ここで、 η は正規化係数、 $p(\mathbf{z}_t | \mathbf{x}_t, \mathbf{m})$ は観測モデル、 $p(\mathbf{x}_t | \mathbf{x}_{t-1}, \mathbf{u}_t)$ は動作モデルである。また $1:t$ は、時刻 1 から t までの時系列データを意味している。この式を端的に述べると、「時刻 $t-1$ における位置に関する事前分布を動作モデルにより更新し、観測モデルによる尤度計算を行うことで、時刻 t における位置に関する事後分布を得る」ということを意味する。

式 (1) に示す様に、通常の位置推定モデルでは、位置に関する事後分布のみを求めている。すなわち、位置推定実行後には、確率分布の不確かさに基づく、推定結果の確信度を得ることができるが、この確信度は、位置推定の正誤を明示的に説明するものではない。なぜなら、位置推定に失敗、すなわち誤収束した場合には、推定に失敗しているにも関わらず確信度が高い状態になるためである。すなわち、通常の位置推定モデルに基づいて位置推定を実行したとしても、推定結果に対する正誤を明示的に説明する情報、すなわち信頼度を得ることはできない。

3.2 信頼度付き位置推定モデル

信頼度付き位置推定モデルでは、通常の位置推定モデルにて使用した変数に加えて、信頼度 $r \in \{0, 1\}$ と、推定結果に対する正誤判断 d ($0 \leq d \leq 1$) がそれぞれ未知・可観測変数として追加される。ここで可観測変

数として利用される正誤判断は、何かしらの機能（例えば CNN）により実現されるものである。そして、次式に示す様に、時刻 t における位置と信頼度に関する同時事後分布を求めることを目標とする。

$$p(\mathbf{x}_t, r_t | \mathbf{u}_{1:t}, \mathbf{z}_{1:t}, d_{1:t}, \mathbf{m}) = \eta p(\mathbf{z}_t | \mathbf{x}_t, \mathbf{m}) \sum_{r_t} p(d_t | \mathbf{x}_t, r_t, \mathbf{z}_t, \mathbf{m}) p(r_t) \underbrace{\int p(\mathbf{x}_t | \mathbf{x}_{t-1}, \mathbf{u}_t) p(\mathbf{x}_{t-1} | \mathbf{u}_{1:t-1}, \mathbf{z}_{1:t-1}, d_{1:t-1}, \mathbf{m}) d\mathbf{x}_{t-1}}_{\text{位置に関する分布}} \underbrace{p(d_t | \mathbf{x}_t, r_t, \mathbf{z}_t, \mathbf{m}) \int p(r_t | r_{t-1}, \mathbf{u}_t) p(r_{t-1} | \mathbf{x}_{1:t-1}, \mathbf{u}_{1:t-1}, \mathbf{z}_{1:t-1}, d_{1:t-1}, \mathbf{m}) dr_{t-1}}_{\text{信頼度に関する分布}} \quad (2)$$

上記の同時分布は、位置と信頼度に関する分布に分解されているが、この分解のためにラオブラックウェル化を適用している。これは、位置が与えられた場合に、信頼度が解析的に計算できると仮定したためである。信頼度付き位置推定では、明示的に信頼度という変数を追加している。そのため、上述の同時分布の推定を終えた時点で、明示的に位置推定結果の信頼度を得ることができる。

式 (2) に示す位置に関する分布と、式 (1) に示す分布は、ともに時刻 t に関する事後分布を示しているが、信頼度付き位置推定では、尤度計算のために新たな項 $\sum_{r_t} p(d_t | \mathbf{x}_t, r_t, \mathbf{z}_t, \mathbf{m}) p(r_t)$ が追加される。この項に含まれる $p(d_t | \mathbf{x}_t, r_t, \mathbf{z}_t, \mathbf{m})$ は、信頼度が条件として与えられた下で、正誤判断が得られる確率を表したモデルとなる。そのためこの項を「正誤判断モデル」と呼ぶ。この正誤判断モデルを利用することで、信頼度付き位置推定モデルは利点を得ることができる。また $p(r_t | r_{t-1}, \mathbf{u}_t)$ は、移動に伴う信頼度の変化を表すモデルである。一般的に、移動に伴い位置推定結果の信頼度は減衰していくため、このモデルを「信頼度減衰モデル」と呼ぶ。

3.3 信頼度付き位置推定モデルの利点

信頼度付き位置推定モデルでは、位置推定結果に対する正誤判断を行う機能、すなわち推定結果の正しさを確率値として算出する機能を使用する必要がある。この実装方法は様々なものが考えられる。例えば文献 [5] では、CNN を用いてこれを実装している。しかしどのような方法を用いて実装したとしても、このような正誤判断が常に正しくなる保証はなく、正誤判断の結果は不確かさを含むものとして考慮すべきである。

信頼度付き位置推定モデルでは、正誤判断モデルを用いることで、この不確かさを扱うことができる。例えば文献 [5] では、CNN の予測結果を事前に調べることで、どの程度の割合で正誤判断を誤るかなどを調査しておく。そして、正誤判断モデルを、そのような誤りを含むような分布としてモデル化することで、安直に CNN の判断を信用させない効果を与えることができる。すなわち信頼度付き位置推定モデルを利用することで、正誤判断に使用した機能の出力の不確かさを扱うことができるようになり、誤った出力に起因する影響を小さくする利点が得られる。

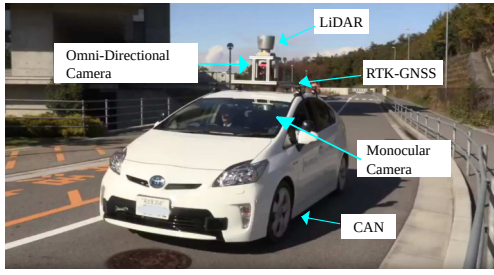


Fig. 2: 実験車両 [2]

4 実装

4.1 変数定義

本研究では、LiDAR を用いた 2 次元の位置推定問題に焦点を当てる。つまり、ある時刻 t において、観測値 $\mathbf{z}_t = (\mathbf{z}_t^1, \mathbf{z}_t^2, \dots, \mathbf{z}_t^K)$ が得られる下で、2 次元の位置 x_t, y_t と姿勢 θ_t を求める。ここで K は LiDAR の計測値の数であり、各観測値 \mathbf{z}_t^k は、距離 l_t^k と角度 ψ_t^k を有する。また、地図は事前に構築されており、占有格子地図により $\mathbf{m} = (m_1, m_2, \dots, m_M)$ とモデル化されているものとする。ここで M は地図の格子数であり、 $m_m \in \{0, 1\}$ は、 m 番目の格子が占有か非占有かを表す変数である。さらに実験で使用する車両は内界センサを有しているものとし、これにより計測された並進・回転方向の移動量 Δd_t と $\Delta \theta_t$ を制御入力 \mathbf{u}_t として使用する。

4.2 実験車両

図 2 に実験で使った車両を示す。実験車両は 3D LiDAR (Velodyne Lidar 社製 HDL-64e) を有するが、この任意部分の計測値をクリッピングし、仮想的に 2 次元 LiDAR の計測値を得る。また内界センサとして、CAN の値を使用する。

4.3 同時分布の推定方法

式 (2) に示す様に、同時分布を分解するにあたりラオブラックウェル化を適用した。このラオブラックウェル化の適用にあたり、位置が与えられた時に、信頼度が解析的に計算できると仮定した。そこで式 (2) に示す同時分布を、ラオブラックウェルライズドパーティクルフィルタ (以下、RBPF) を用いて推定する。すなわち、位置に関する分布をパーティクルフィルタで求め、信頼度に関する分布を解析的に計算する。

4.4 動作モデル

車両の後輪の中央を位置推定により求める位置とし、内界センサは後輪の回転量を計測できるものとする。そのため、車両の運動モデルは左右独立 2 輪駆動モデルとしてモデル化する [6]。

4.5 観測モデル

各 LiDAR の計測値は独立していると仮定し、観測モデルは以下の様に分解できると仮定する。

$$p(\mathbf{z}_t | \mathbf{x}_t, \mathbf{m}) = \prod_{i=1}^K p(\mathbf{z}_t^i | \mathbf{x}_t, \mathbf{m}) \quad (3)$$

ここで、各 LiDAR の観測値に対するモデル $p(\mathbf{z}_t^i | \mathbf{x}_t, \mathbf{m})$ を、尤度場モデルによりモデル化する [6]。

4.6 信頼度減衰モデル

信頼度減衰モデル $p(r_t | r_{t-1}, \mathbf{u}_t)$ は、移動に伴う位置推定結果の信頼度の変化を表すモデルである。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。そこで本研究では、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。

$$p(r_t = 1) = (1 - (\alpha_1 \Delta d_t^2 + \alpha_2 \Delta \theta_t^2)) p(r_{t-1} = 1) \quad (4)$$

ここで α_1 と α_2 は任意係数である。

4.7 CNN による正誤判断

位置推定の正誤判断を行うにあたり CNN を使用する。CNN には、LiDAR により計測した距離ベクトル $\mathbf{l}_t = (l_t^1, l_t^2, \dots, l_t^K)$ と、各計測値に対する残差ベクトル $\mathbf{e}_t = (e_t^1, e_t^2, \dots, e_t^K)$ を入力する。ここで e_t^k は、 k 番目の計測値を地図上に座標変換した際の点と、その点から最も近い障害物までの距離であり、尤度場モデルによる尤度計算を行うために必要な値である。そして、位置推定成功時には 1、失敗時には 0 とそれぞれラベルを与え、以下のロス関数を最小化するように学習を行う。

$$\mathcal{L}(\mathbf{l}_t, \mathbf{e}_t) = \frac{\lambda}{2} \|\omega^{\text{net}}\| + \sum_{i=1}^N (y_i - d_i)^2 \quad (5)$$

ここで λ は正則化項に対する係数、 ω^{net} はネットワークの重み、 N は学習するデータの数、 $0 \leq d_i \leq 1$ と $y_i \in \{0, 1\}$ は i 番目の入力値に対する CNN の予測値とラベルである。なお、本研究で使用した CNN を実装するにあたり文献 [29] を参考にした。また、CNN の実装には Keras を用いた [30]。

上述の通り、CNN には尤度場モデルの計算に必要な値を入力する。この値は高速に計算できるため、入力データの作成に対する計算時間を大幅に短縮できる。また入力するのは 2 つのベクトルのみであるため、CNN による予測演算自体も高速に行うことができる。

4.8 正誤判断モデル

正誤判断モデルは以下の様にモデル化する。

$$p(d_t | \mathbf{x}_t, r_t, \mathbf{z}_t, \mathbf{m}) = \begin{pmatrix} d_{\text{posi}} \\ d_{\text{nega}} \end{pmatrix}^T \cdot \begin{pmatrix} p_{\text{posi}}(d_t | \mathbf{x}_t, r_t, \mathbf{z}_t, \mathbf{m}) \\ p_{\text{nega}}(d_t | \mathbf{x}_t, r_t, \mathbf{z}_t, \mathbf{m}) \end{pmatrix} \quad (6)$$

ここで d_{posi} と d_{nega} は任意の係数であり、 $d_{\text{posi}} + d_{\text{nega}} = 1$ の条件を満たす。また $p_{\text{posi}}(\cdot)$ と $p_{\text{nega}}(\cdot)$ は正誤判断が成功、および失敗する際の確率分布をモデル化したものである。本研究では、 $p_{\text{posi}}(\cdot)$ と $p_{\text{nega}}(\cdot)$ を以下の様にモデル化した。

$$p_{\text{posi}}(d_t | \mathbf{x}_t, r_t, \mathbf{z}_t, \mathbf{m}) = \frac{d_t^{a-1} (1 - d_t)^{b-1}}{B(a, b)} \quad (7)$$

$$p_{\text{nega}}(d_t | \mathbf{x}_t, r_t, \mathbf{z}_t, \mathbf{m}) = \text{unif}(0, 1) \quad (8)$$

ここで $B(\cdot)$ と $\text{unif}(\cdot)$ はベータ関数と一様分布を表す。また 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。

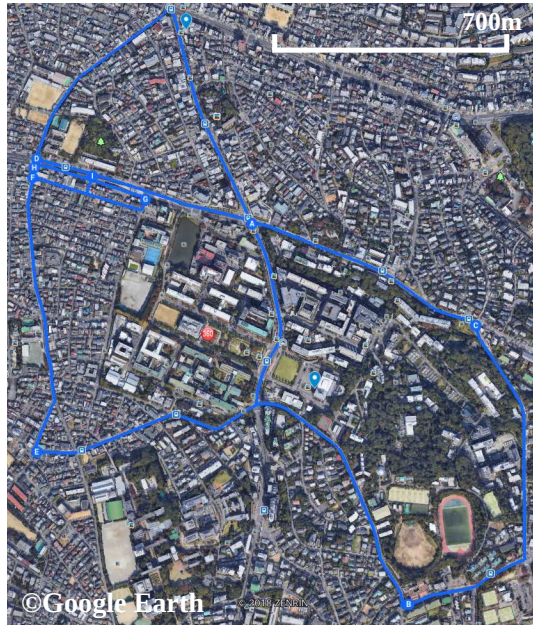


Fig. 3: 実験環境 [2]

4.9 RBPF による信頼度付き位置推定の計算手順

以下の処理を繰り返し実行することで、RBPF を用いた信頼度付き位置推定を実行する。

1. 動作モデルにより各パーティクルの状態を更新する。また、信頼度減衰モデルにより各パーティクルの信頼度を更新する。
2. 観測モデルにより各パーティクルの尤度計算を行う。同時に、各パーティクルの状態に基づき CNN への入力データを作成する。
3. CNN による予測値を用いて、正誤判断モデルによる尤度計算を行う。
4. 正誤判断モデルによる尤度を用いて、信頼度の更新、および各パーティクルの尤度計算を行う。
5. 各パーティクルの重み付き平均の状態を推定位置とする。また、最尤パーティクルが有する信頼度を現在の信頼度とする。
6. 各パーティクルの尤度に基づき、適宜リサンプリングを行う。

本研究における実装では、信頼度が 95 % を下回った際に、位置推定に失敗したと見なし、現在の推定位置を中心としてパーティクルの分布をリセットする機能を実装した。これにより、車両の位置追跡の失敗から復帰できる効果が期待できる。

5 実験

5.1 実験環境

図 3 に実験環境を示す。本環境で複数回のデータを取得し、それぞれ正誤判断を行う CNN の学習、および位置推定実験を行った。なお本経路の全長は 1 周およそ 17 km であり、実験時の車両の最大速度は約 50 km/h である。

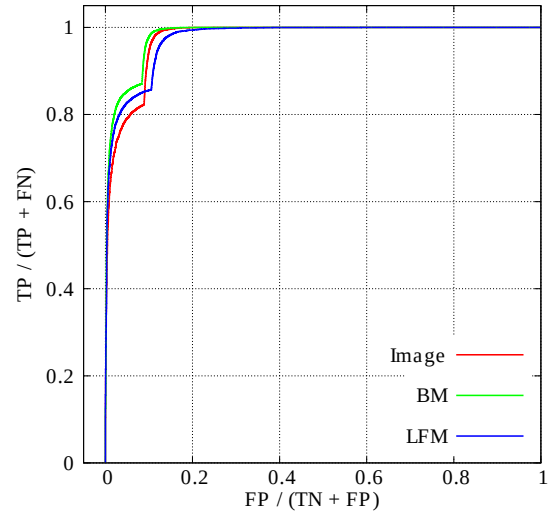


Fig. 4: ROC 曲線 [2]

5.2 CNN による正誤判断

はじめに、CNN による正誤判断の精度を検証した。なお今回の検証では、4.7 章で述べた CNN 以外に、2 種類の異なるデータを入力する CNN の比較も行った。比較する 2 種類のうち、1 つは文献 [6] で述べられているビームモデルを計算するために必要なデータを入力する CNN、もう 1 つは文献 [5] で提案している画像データを入力する CNN である。表記の簡略化のため、本節ではそれぞれ LFM (提案法)、BM、Image と表記する。なお学習データを作成するにあたり、位置推定の成功・失敗時を定める必要がある。本研究では、3D NDT スキャンマッチング [31,32] (以下、3D NDT) を使用し、これによる推定位置が真値であると仮定し、これを位置推定成功状態として利用した。そして、この推定位置にランダムノイズを加え、位置誤差 $\sqrt{\Delta x_t^2 + \Delta y_t^2}$ が 50 cm、または角度誤差 $|\Delta \theta_t|$ が 3 度を越えた状態を、位置推定失敗状態とした。

図 4 に、それぞれの CNN 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、それ以外の場合を失敗と判断しているものとした。図 4 からわかるように、どの CNN でも同等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。Image の性能がわずかに低下しているのは、画像の解像度の影響が大きいといえる。

さらに表 1 には、それぞれ 1 つのデータを作成、および CNN により処理するのに要した平均時間を示す。入力データの作成は CPU (Intel(R) Xeon(R) CPU E5-1650 v3 @ 3.50 GHz) 上でを行い、CNN による予測の処理は GPU (GeForce GTX TITAN X) 上で行った。表 1 から、LFM の処理が、他の処理に比べて高速に行えていることが確認できる。これらの結果から、4.7 章で述べた CNN を用いることで、著しい性能の低下なく、正誤判断の計算を高速化できることが確認できた。また後に示す位置推定実験では、パーティクル数 200 に対して 0.1 秒以下の周期で信頼度付き位置推定が実行可能なことを確認した。

Table 1: 正誤判断に関する入力データ作成と予測の平均計算時間 [2]. 単位はミリ秒.

	入力データ作成時間	予測時間
LFM	0.006	0.390
BM	3.953	0.397
Image	13.545	0.861

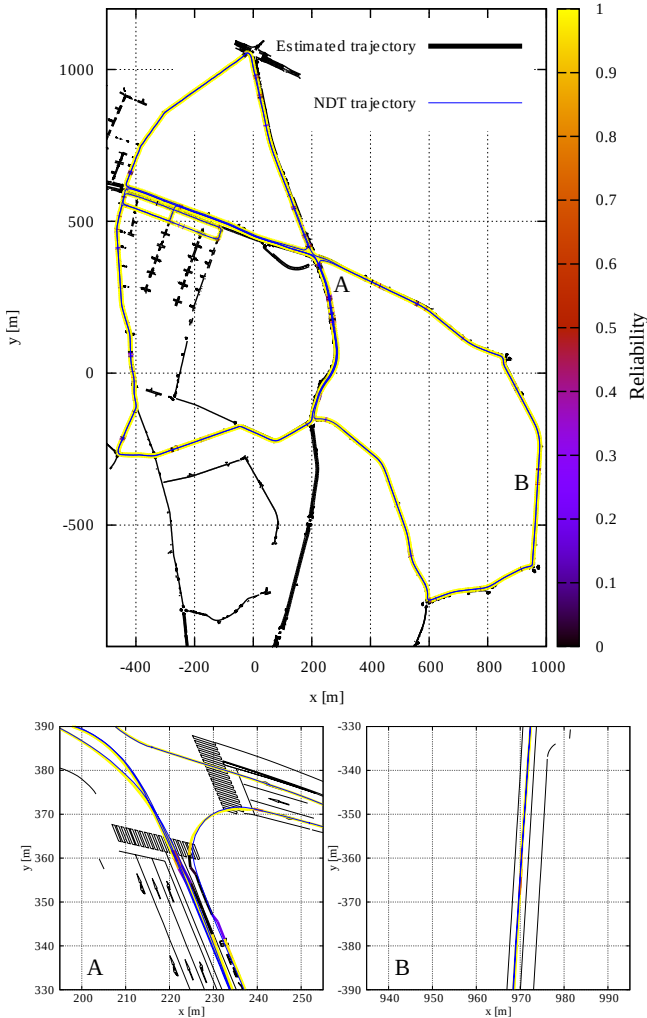


Fig. 5: 走行軌跡と信頼度 [2]

5.3 走行実験

CNN の学習のために使用した以外のデータを用いて、走行実験を行った結果を図 5 に示す。なお参考として、3D NDT による推定軌跡を青線で示す。提案法により、始点から終点までの位置推定が成功できたことを確認した。

図 5 下図には、上図に示す A・B 地点の拡大図を示す。A の拡大図からわかる様に、提案法と 3D NDT による位置推定結果がずれていることが確認できる。なお、明確な位置の真値はわからないものの、3D NDT による推定結果の方が真値に近いことを目視で確認している。この地点において、信頼度が低下して 95% を下回ったため、パーティクルの分布がリセットされ、位置推定に再度成功することができた。このような状況の例を図 6 に示す。図 6 左は、信頼度が低下した瞬間を示している。黄色の枠内に示す通り、LiDAR の計測値

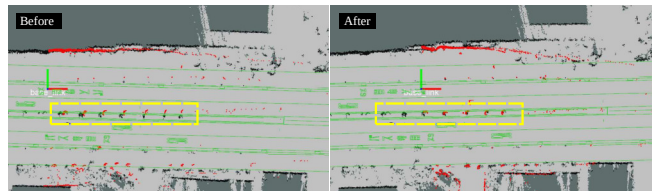


Fig. 6: 信頼度低下状態からの失敗復帰の例 [2]

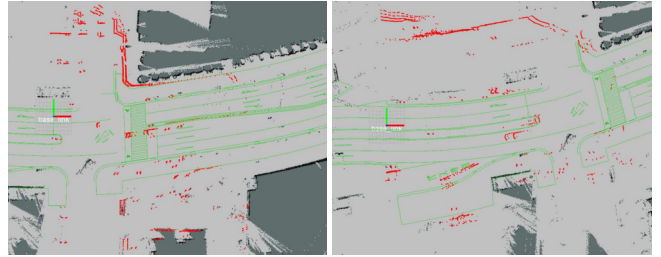


Fig. 7: 誤って低信頼度状態と判断された例 [2]

(赤)と地図(黒)が正しく照合されていないことが確認できる。これは、3D LiDAR の計測値を 2 次元に縮退して利用していることにより、時系列的な位置推定の結果が誤収束してしまった状況である。提案法を用いることで、このような状況を低信頼度状態として認識することができた。そしてパーティクルの分布のリセット処理が行われ、図 6 右に示す状態に復帰し、この状態を高信頼度状態として識別できた。提案法により推定された信頼度を用いることで、位置推定の正誤を説明できることが可能となった。

しかしながら、図 5 下図の B の拡大図に示す様に、信頼度の推定が正しく機能しない状況も見られた。この状況では、提案法による位置推定結果が 3D NDT による推定結果と同様の場合でも、信頼度が低い状態であると判断された。このような状況の例を図 7 に示す。この状況では、位置推定には成功しているものの、LiDAR の観測値と地図が照合している点が少ないことが確認できる。これは、3D LiDAR の観測値を 2 次元に縮退させたことで、地図に記録された目印となる障害物が少なくなってしまうことに起因する。この状況では、目視でも推定の正誤を判断することは困難であり、提案法も正しく機能することができなかった。

5.4 性能限界

位置推定結果の正誤を明示的に説明することが可能な指標を得ることを目的とし、信頼度付き位置推定を提案してその性能を検証した。実験を通して、提案法により推定された信頼度が位置推定の正誤を説明することが可能なことを確認した。しかしながら、図 7 に示す様に、位置推定が成功しているにもかかわらず、誤って低信頼度と判断してしまう瞬間も見られた。提案法では、位置推定の正誤判断に CNN を用いているため、正確に信頼度推定が行えるかどうかは CNN の性能に依存するが、図 7 示す状況は、目視でも判断が難しい状況であった。より安定して信頼度を推定するためにも、この正誤判断の性能を上げることは重要となる。

さらに提案法は、計算時間の課題も有する。CNN に入力するデータを改善することで、文献 [5] で提案した手法より高速化は実現できたものの、3 次元への位

位置推定問題への拡張は未だに困難である。これは特に、提案法がRBPFで実装されていることにも起因しており、次元の増加に伴う必要なパーティクル数の増加に対処できないためである。3次元に拡張した場合においても、正誤判断を高速に行うための方法や、カルマンフィルタの様な計算時間の少ない方法に基づく実装方法の構築などが今後の課題となる。

6 おわりに

本稿では、完全自動運転実現のために、位置推定機能の保証を行うことを目的とし、位置推定結果の正誤を説明可能な信頼度を明示的に推定する「信頼度付き位置推定」を提案した。信頼度付き位置推定のモデルは、明示的に信頼度を未知変数として導入している部分が、従来の位置推定のモデルとは異なる。また、位置推定の正誤判断を行う機能を要求する部分も、通常の位置推定のモデルと異なる。本稿では、この位置推定の正誤判断を行う機能をCNNを用いて実装した、2次元の位置推定問題を対象とした信頼度付き位置推定に関して述べた。なお、CNNを用いた信頼度付き位置推定に関しては、文献 [5] で初めて提案したが、計算時間の問題があった。本稿で述べた実装例は、CNNへの入力データの工夫により、この計算時間の問題を解決し、自動運転車両への実装を可能とした方法である。実機実験を通して、推定した信頼度が位置推定の正誤を適切に説明することが可能なことを確認した。

今後は、提案法の3次元位置推定問題への拡張を行うことを計画しており、それに伴うさらなる計算時間の高速化に関する問題に取り組む予定である。

謝辞

本研究は、JST, COI, JPMJCE1317, 科研費40786092の支援により行われた。

参考文献

- 1) 赤井直紀, 他. “自転車位置認識結果の信頼度推定”, 自動車技術会 2018 年秋季大会, 2018.
- 2) 赤井直紀, 他. “LiDAR を用いた自転車位置認識結果の信頼度推定”, 自動車技術会論文集, vol. 50, no. 2, pp. 609–615, 2019.
- 3) N. Akai *et al.* “Autonomous driving based on accurate localization using multilayer LiDAR and dead reckoning,” In *Proc. of the IEEE ITSC*, pp. 1147–1152, 2017.
- 4) N. Akai *et al.* “Misalignment recognition using Markov random fields with fully connected latent variables for detecting localization failures,” *IEEE Robotics and Automation Letters*, vol. 4, no. 4, pp. 3955–3962, 2019.
- 5) N. Akai *et al.* “Simultaneous pose and reliability estimation using convolutional neural network and Rao-Blackwellized particle filter,” *Advanced Robotics*, vol. 32, no. 17, pp. 930–944, 2018.
- 6) S. Thrun *et al.* “Probabilistic robotics,” *The MIT Press*, 2005.
- 7) D. Fox *et al.* “Markov localization for mobile robots in dynamic environments,” *Journal of Artificial Intelligence Research*, vol. 11, pp. 391–427, 1999.
- 8) W. Burgard *et al.* “Experiences with an interactive museum tour-guide robot,” *Artificial Intelligence*, vol. 114, no. 1–2, pp. 3–55, 1999.
- 9) D. F. Wolf *et al.* “Mobile robot simultaneous localization and mapping in dynamic environments,” *Autonomous Robots*, vol. 19, no. 1, pp. 53–65, 2005.

- 10) L. Montesano *et al.* “Modeling the static and the dynamic parts of the environment to improve sensor-based navigation,” In *Proc. of the IEEE ICRA*, 2005.
- 11) P. Biber *et al.* “Dynamic maps for long-term operation of mobile service robots,” In *Robotics: Science and Systems*, 2005.
- 12) C.-C. Wang *et al.* “Simultaneous localization, mapping and moving object tracking,” *The International Journal of Robotics Research*, vol. 26, no. 9, pp. 889–916, 2007.
- 13) D. Meyer-Delius *et al.* “Occupancy grid models for robot mapping in changing environments,” In *Proc. of the AAAI Conference on Artificial Intelligence*, pp. 2024–2030, 2012.
- 14) G.D. Tipaldi *et al.* “Lifelong localization in changing environments,” *The International Journal of Robotics Research*, vol. 32, no. 14, pp. 1662–1678, 2013.
- 15) D. Meyer-Delius *et al.* “Temporary maps for robust localization in semi-static environments,” In *Proc. of the IEEE/RSJ IROS*, 2010.
- 16) R. Valencia *et al.* “Localization in highly dynamic environments using dual-timescale NDT-MCL,” In *Proc. of the IEEE ICRA*, 2014.
- 17) J. Gutmann *et al.* “An experimental comparison of localization methods continued,” In *Proc. of the IEEE/RSJ IROS*, pp. 454–459, 2002.
- 18) S. Olufs *et al.* “An efficient area-based observation model for Monte-Carlo robot localization,” In *Proc. of the IEEE/RSJ IROS*, pp. 13–20, 2009.
- 19) E. Takeuchi *et al.* “Robust localization method based on free-space observation model using 3D-map,” In *Proc. of the IEEE ROBIO*, pp. 973–979, 2010.
- 20) S.-W. Yang *et al.* “Feasibility grids for localization and mapping in crowded urban scenes,” In *Proc. of the IEEE ICRA*, 2011.
- 21) N. Akai *et al.* “Mobile robot localization considering class of sensor observations,” In *Proc. of the IEEE IROS*, pp. 3159–3166, 2018.
- 22) N. Akai *et al.* “Toward localization-based automated driving in highly dynamic environments: Comparison and discussion of observation models,” In *Proc. of the IEEE ITSC*, pp. 2215–2222, 2018.
- 23) P. Sundvall *et al.* “Fault detection for mobile robots using redundant positioning systems,” In *Proc. of the IEEE ICRA*, pp. 3781–3786, 2006.
- 24) J.P. Mendoza *et al.* “Mobile robot fault detection based on redundant information statistics,” In *Proc. of the IEEE/RSJ IROS*, 2012.
- 25) Z. Alsayed *et al.* “2D SLAM correction prediction in large scale urban environments,” In *Proc. of the IEEE ICRA*, pp. 5167–5714, 2018.
- 26) S. Nobili *et al.* “Predicting alignment risk to prevent localization failure,” In *Proc. of the IEEE ICRA*, pp. 1003–1010, 2018.
- 27) L.T. Hsu. “GNSS multipath detection using a machine learning approach,” In *Proc. of the IEEE ITSC*, pp. 1414–1419, 2017.
- 28) H. Almqvist *et al.* “Learning to detect misaligned point clouds,” *Journal of Field Robotics*, vol. 35, pp. 662–677, 2018.
- 29) S. Zagoruyko *et al.* “Learning to compare image patches via convolutional neural networks,” In *Proc. of the IEEE/CVF CVPR*, pp. 4353–4361, 2015.
- 30) F. Chollet. “Keras,” <https://keras.io>, 2015.
- 31) E. Takeuchi *et al.* “A 3-D scan matching using improved 3-D normal distributions transform for mobile robotic mapping,” In *Proc. of the IEEE/RSJ IROS*, pp. 3068–3073, 2006.
- 32) N. Akai *et al.* “Robust localization using 3D NDT scan matching with experimentally determined uncertainty and road marker matching,” In *Proc. of the IEEE IV*, pp. 1357–1364, 2017.