

LiDARを用いた 高度自己位置推定システム

— 移動ロボットのための
自己位置推定の高性能化とその実装例 —

博士(工学) 赤井直紀 著

コロナ社

まえがき

自己位置推定 (localization) とは、その名のとおりに「自己」の「位置」を「推定」する技術です。どういうことかといえば、あらかじめ地図が構築されているという前提のもとで、移動体が有するセンサの観測値を地図と照合し、いま現在、移動体が地図上のどの地点にいるかを特定する技術です。ここでいう移動体とは、ロボットや自動車、スマートフォンなど、多様なものが挙げられます。これだけ聞くと、「位置を知ることができて何が嬉しいのか」と感じるかもしれません、自己位置推定はロボットや自動車の自動走行を実現するための根幹技術となっています。自己位置推定はロボティクス分野では古くから研究されており、現在においても非常にポピュラーなトピックです。特に近年では、深層学習を初めとした機械学習の発展が目覚ましく、自己位置推定問題にも深層学習を応用する例が数多く報告されています。

上述のとおり自己位置推定は、自動走行実現のために重要な役割を担います。自動走行を構成する技術は多数ありますが、自己位置推定技術の発展は、自動運転を初めとした自動走行技術が社会実装されるために、不可欠なものになると著者は考えています。しかし、上述のような、深層学習を単に自己位置推定に応用するというアプローチは、自動走行技術の社会実装に直接的にはつながらないと著者は考えています。これはいわゆる、深層学習の「ブラックボックス性^{†1}」が問題になるというのではなく、根本的に解決困難な課題が存在すると考えているためです。

自己位置推定の「推定」という言葉は、「ある事実を手がかりにして、推し量って決めること」を意味しています。つまり、「センサの観測値を基に、おそらくこの地点に存在するだろう」という認識を行っています。何を指摘したいかということ、「この地点に存在しています」と「断定」しているわけではなく、あくまで「存在するだろう」と「推定」しているのです。自動運転のようなものは、失敗すれば大事故を引き起こしかねないものです。しかも厄介なことに、自己位置推定に基づく自動走行システムを実装すると、自己位置推定の失敗とともに自動走行も失敗するということが頻発します。自動走行を実施するにあたり、このような曖昧な自己位置推定結果に基づいて走行することは、安全保障の観点から適切ではないと著者は考えています。

上記の問題に対して著者は、「自己位置推定の結果を信頼するために何をすべきか」という問題を定め、これに関する研究を行ってきました。この問題に取り組み、実感したことは、「Probabilistic Robotics」^{†2,3}に記述されている自己位置推定法には限界があるということでした。つまりこの自己位置推定法だけでは、自己位置推定結果を信頼するに足る情報までを得ることができず、

^{†1} 学習器がどのような入力に対して、どのような出力を行うかを予測することは困難であるという性質です。

^{†2} Thrun, Burgard, Fox によって書かれた、この分野のバイブル的な書籍です。

^{†3} 肩付き数字は巻末の引用・参考文献を示します。

上記の問題は解決できないと考えました。なお、ここでいう「解決できる」とは、「数式として定めてモデルを定義し、そのモデルを解くことで欲しい情報が得られる」という状態を意味していることに留意してください。このことから、「これまでの自己位置推定法ではできないことをできるようにする」というモチベーションを持ち、研究を行ってきました。これこそが、本書が主張している「自己位置推定の高度化」になります。つまり、単に使う情報を工夫して自己位置推定の精度や頑健性を向上させるのではなく、自己位置推定問題単体とは異なる定式化を行い、自己位置と同時に他の重要な情報を知ることができるような拡張を行っています。その代表的な取組みが7章で述べる「信頼度付き自己位置推定」であり、自己位置推定結果の信頼度を推定するというものです。このように、これまでの自己位置推定でできないことをできるようにする「高度化に基づく高性能化」こそ、真に自動走行技術の社会実装につながると著者は考えています。

このような理由から、本書は自己位置推定を実装、または利用した経験があり、その問題の存在を認識し、解決したいというような考えを持った読者にとって、有益であると考えられます。入門的な部分については、1~5章までで自己位置推定の基礎とそれを理解するために必要な数学的知識、またその実装方法も解説します。自己位置推定以外の内容についてさらに詳しく知りたい方は、他の書籍（例えば、「Probabilistic Robotics」¹⁾、「詳解 確率ロボティクス」²⁾、「SLAM 入門」³⁾）を参考にしてください。

本書では、論文のように単に数式を説明するだけでなく、C++による実装例も掲載し、より実現の方法がわかりやすくなるように配慮しました。また、近年のオープンソースソフトウェアの流れに乗り、利用した実際のソースコードも公開しています[†]。このソフトウェアを通して、著者がこれまでに培ってきた技術が、少しでも社会に還元されることを願っています。

著者自身は、2016年3月に博士号を取得した身分であり、いわゆる大御所と呼ばれるような研究者ではありません。それにもかかわらず、私のような若手の研究者に、本書の執筆の機会を与えてくださったコロナ社には、大変感謝しています。微力とは思いますが、本書がロボティクス分野、ひいてはわが国の科学技術力向上につながれば幸いです。

最後になりますが、本書は、2016年からこれまでの間に、著者が名古屋大学で行った研究の成果を基にまとめたものです。この間、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）」から多大な支援をいただきました。この支援なしには、本書は存在し得ませんでした。またこの研究期間を通して、名古屋大学内外からの多くの研究者・技術者から助言をいただきました。本来ならお一人ずつ名前を挙げて感謝を申し上げたいところですが、紙面の都合で割愛させていただきます。これらの支援や助言に対してここに感謝し、お礼申し上げます。

2022年4月

赤井直紀

[†] <https://www.coronasha.co.jp/np/isbn/9784339032406/>

目 次

1. 自己位置推定およびその高度化について

1.1 自己位置推定およびその高度化	1
1.2 本書で扱う手法	4
1.3 本書の構成と内容	4
1.4 ま と め	7

2. 開発環境構築とシミュレータ

2.1 開発環境構築	8
2.1.1 開発環境の構築	8
2.1.2 ALSEdu のインストール	9
2.2 シミュレータの概要	10
2.2.1 ロボットの構成	10
2.2.2 シミュレータの起動の確認	11
2.2.3 シミュレータの中身の確認	12
2.3 ま と め	14

3. 数学的基礎

3.1 確率の基礎	15
3.1.1 確率分布	15
3.1.2 期待値と分散	17
3.1.3 同時確率と条件付き確率	17
3.1.4 加法定理, 乗法定理, 全確率の定理, ベイズの定理	18
3.2 グラフィカルモデル	20
3.3 ベイジアンネットワーク	20
3.3.1 条件付き独立性	21
3.3.2 有 向 分 離	23
3.4 マルコフ確率場	26
3.4.1 マルコフ確率場の因数分解	26
3.4.2 ポテンシャル関数の具体例	27

3.4.3	一直線のマルコフ確率場	29
3.5	ま と め	33

4. 自己位置推定の定式化と動作モデル，観測モデル

4.1	自己位置推定の定式化	34
4.1.1	グラフィカルモデル	34
4.1.2	定 式 化	35
4.2	動 作 モ デ ル	37
4.3	観 測 モ デ ル	38
4.3.1	観測の独立性の仮定	38
4.3.2	ビームモデル	38
4.3.3	尤度場モデル	41
4.3.4	その他の観測モデル	43
4.3.5	自己位置推定はなぜ環境変化に対する頑健性を保てないのか	43
4.4	ま と め	44

5. モンテカルロ位置推定の実装

5.1	モンテカルロ位置推定	46
5.1.1	パーティクルフィルタ	46
5.1.2	動作モデルによる更新	49
5.1.3	観測モデルによる尤度の計算	50
5.1.4	自己位置の推定	54
5.1.5	リサンプリング	56
5.2	モンテカルロ位置推定の実行	58
5.2.1	実 装 例	58
5.2.2	実行と位置推定の結果	59
5.3	モンテカルロ位置推定の拡張	61
5.3.1	適応的なパーティクル数の決定	61
5.3.2	誤収束の認識と復帰	61
5.3.3	動的障害物を観測している観測値の棄却	62
5.4	ま と め	63

6. 自己位置と観測物体のクラスの同時推定

6.1	グラフィカルモデルと定式化	64
6.1.1	グラフィカルモデル	64

6.1.2	定 式 化	65
6.2	自己位置と観測物体のクラスの同時推定の実装	66
6.2.1	ラオ・ブラックウェル化パーティクルフィルタ	66
6.2.2	処 理 手 順	67
6.2.3	尤 度 計 算	67
6.2.4	known の場合の観測モデル	68
6.2.5	unknown の場合の観測モデル	69
6.2.6	c_t の事前確率	69
6.2.7	クラス条件付き観測モデルによる尤度計算の実装例	69
6.2.8	センサ観測値のクラスに関する確率分布の推定	71
6.2.9	センサ観測値のクラス推定の実装例	71
6.3	自己位置と観測物体のクラスの同時推定の実行	72
6.3.1	実 行	72
6.3.2	尤度分布の比較	73
6.3.3	棄却性能の比較	74
6.3.4	欠 点	76
6.4	汎用的クラスを用いた自己位置推定法への拡張	77
6.4.1	汎用的クラスを導入することの難しさ	77
6.4.2	クラス情報を活用した尤度計算モデル	78
6.5	関 連 研 究	80
6.6	ま と め	81

7. 信頼度付き自己位置推定

7.1	グラフィカルモデルと定式化および信頼度の解釈	83
7.1.1	グラフィカルモデル	83
7.1.2	定 式 化	84
7.1.3	信頼度付き自己位置推定における信頼度の解釈	87
7.1.4	信頼度と確信度	87
7.2	信頼度付き自己位置推定の実装	88
7.2.1	処 理 手 順	88
7.2.2	信頼度遷移モデル	89
7.2.3	判 断 モ デ ル	90
7.3	信頼度付き自己位置推定の実行	97
7.3.1	正誤判断分類器の学習の実行	97
7.3.2	実 行	99
7.3.3	信頼度推定の結果	100
7.3.4	性 能 限 界	102
7.4	関 連 研 究	103

7.5	ま と め	104
-----	-------	-----

8. センサ観測値と地図間の誤対応認識

8.1	観測の独立性	106
8.1.1	観測の独立性を仮定することの必要性	107
8.1.2	観測の独立性を仮定することの問題	108
8.1.3	観測の独立性の仮定の正しさ	109
8.1.4	自己位置推定の性能保証の観点から見た SLAM の重要性	110
8.2	センサ観測値全体の関係性を考慮した誤対応認識	111
8.2.1	未知変数全結合型のマルコフ確率場のグラフィカルモデル	111
8.2.2	定 式 化	112
8.3	誤対応認識の実装	114
8.3.1	グラフィカルモデルにおける変数の物理的意味	114
8.3.2	尤度ベクトル	114
8.3.3	二つの未知変数がなすクリーク上のポテンシャル関数	116
8.3.4	事後分布の推定	117
8.3.5	事後分布推定のプログラムによる実装	118
8.3.6	自己位置推定失敗の検知	119
8.3.7	自己位置推定失敗の検知のプログラムによる実装	120
8.3.8	全結合の意味と注意点	121
8.4	誤対応認識の実行	122
8.4.1	実 装 例	122
8.4.2	プログラムの実行	123
8.4.3	性 能 限 界	125
8.5	関 連 研 究	125
8.6	ま と め	126

9. One-shot 自己位置推定と MCL の融合

9.1	One-shot 自己位置推定	128
9.2	誘拐状態の検知と復帰およびそれらの課題	129
9.2.1	誘拐状態の検知	129
9.2.2	センサリセット	131
9.2.3	膨張りリセット	131
9.2.4	誘拐状態の検知・リセットの難しさ	131
9.3	深層学習を用いた One-shot 自己位置推定と MCL の融合	133
9.3.1	重点サンプリング	133

9.3.2	深層学習の予測の不確かさ	134
9.3.3	重点サンプリングを介した One-shot 自己位置推定の融合	135
9.4	One-shot 自己位置推定と MCL の融合の実装	135
9.4.1	One-shot 自己位置推定からサンプリングされたパーティクルの尤度計算	136
9.4.2	尤度に従った自己位置の推定	138
9.4.3	リサンプリング	139
9.5	One-shot 自己位置推定と MCL の融合の実行	141
9.5.1	実装例	141
9.5.2	実行結果	142
9.5.3	性能限界	144
9.6	関連研究	145
9.7	まとめ	146

10. 自己位置推定の高性能化に向けて

10.1	環境変化に対する自己位置推定の頑健性	147
10.1.1	頑健性保証の難しさ	147
10.1.2	頑健性向上のためのアプローチ	148
10.1.3	観測モデルに関するさらなる発展	149
10.2	自己位置推定結果の信頼度	149
10.2.1	自己位置推定結果の確信度と信頼度	149
10.2.2	信頼度推定のためのアプローチ	150
10.2.3	より正確な信頼度を推定するために	151
10.3	自己位置推定結果の正誤判断	151
10.3.1	センサ観測値と地図間の誤対応認識	151
10.3.2	誤対応認識のアプローチ	152
10.3.3	より正確な誤対応認識	152
10.4	誘拐状態からの復帰	153
10.4.1	誘拐状態の検知と復帰の難しさ	153
10.4.2	誘拐状態からの復帰アプローチ	154
10.4.3	より正確な誘拐状態からの復帰	154
10.5	まとめ	155
	引用・参考文献	156
	索引	162

1

自己位置推定および その高度化について



本章では、本書で扱う自己位置推定がどのようなものか、またなぜそれが重要なのかといった基礎的なことを、自動走行の例を挙げながら解説します。その後、本書の目的でもある自己位置推定の高性能化を行うことのモチベーションに関して解説していきます。そして、本書で扱う手法について述べた後に、本書の構成を整理します。

1.1 自己位置推定およびその高度化

自己位置推定 (localization) 問題とは、与えられた地図上において、対象とする移動体の相対位置を求める問題です。これはロボットや自動車が自動走行を行うためにきわめて重要な要素技術であり、これまでに多くの研究が行われてきました。「自己位置推定を行う」ということは、基本的には「センサから得られた情報を地図と照合する」ということです。例えば図 1.1(a) に示すように、本書で扱う 2 次元のレーザセンサ (2D LiDAR) を想定すると、2 次元のスキャンデータ (点群 (point cloud)) を得ることができます。この点群を、環境形状を表現した地図と照合、すなわち合わせ込む・重ねることで、地図に対してセンサがどの位置にあるか (相対位置) を知ることができます (図 (b))。これだけ聞くと簡単な問題に聞こえますが、やはり多くの研究が行われてきたという背景もあり、難しい問題が多数含まれます。

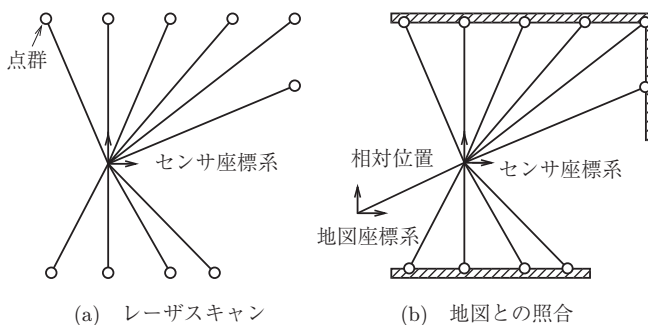


図 1.1 2D LiDAR による計測と自己位置推定

自己位置推定の議論をしていてしばしば問題視されることは、実際の環境と地図が変化してしまった際の推定の頑健性 (ロバスト性) です。環境が地図を構築したときと比べて変化してしまうと、当然ながら得られるセンサ観測値の形状と、地図の形状が異なることとなります。セ

2 1. 自己位置推定およびその高度化について

ンサの観測値と地図を照合させることが自己位置推定の基本なので、これは大きな影響となります。なお、地図と環境が変わってしまった場合に自己位置推定が難しくなるというのは、非常に直観的な説明でしかありませんが、数式として表現した場合になぜ難しくなるのかという解説は、4章にて行っています。

また、自己位置推定を応用することを考えると、安全性の確保のためにも、「自己位置推定した結果が本当に正しいか・信頼できるか」を知りたいという要望が出てきます。しかし、推定した結果が本当に正しいかを知ることはきわめて困難です。自己位置推定の失敗を数式で定義することは可能なのですが、その式を計算することが不可能なのです[†]。ここでは詳細は省きますが、上述のとおり、自動運転にも自己位置推定の機能が使われています。一例ですが、図 1.2 に、自己位置推定をベースとする一般的な自動走行システムのブロック図を示します。著者はこれまでに、屋外自律移動ロボットの開発^{4),5)} や、一般公道での自動運転の実証実験^{6),7)} といった取組みに関わってきましたが(図 1.3)、それらのすべては図 1.2 に示すブロック図を基本として実装されています。

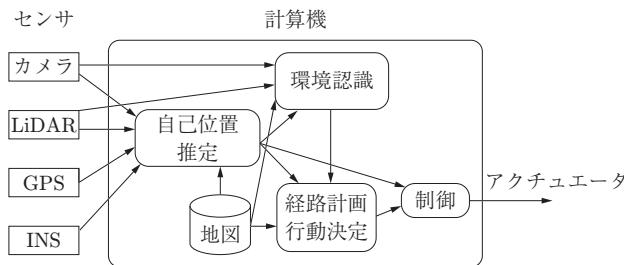


図 1.2 一般的な自動走行システムのブロック図



これらの自動走行でも、図 1.2 に示すブロック図に基づいたシステムが実装されています。

図 1.3 著者がこれまでにやってきた屋外自律移動^{4),5)} や、一般公道での自動運転の実証実験^{6),7)} の例

図 1.2 からわかるとおり、自動走行を実施するにあたり、まず自己位置推定のモジュールがセンサからの観測値を受け取ります。センサ観測値には、2D LiDAR の他にも、3D LiDAR やカメラ、GPS、慣性航法システム (inertial navigation system : INS) などからの観測値があ

[†] この式を計算するためには、自己位置の真値を知る必要があります。自己位置推定の目的はこの真値を推定することにありますので、真値がわかっているなら、そもそも自己位置推定を行う必要がなくなります。この詳細については 9 章で述べています。

る場合もあります。慣性航法システムとは、車速計などの内界センサを用いた移動量を計算するシステムです。そして、外界センサの観測値を地図と照合して自己位置推定を行います。この自己位置推定の結果が、環境認識・経路計画モジュールなどに受け渡されます。

ここで例えば、なぜ環境認識のために自己位置推定の結果を用いるかですが、もちろん単にセンサの観測値から環境を認識するだけであれば、環境認識モジュールは自己位置推定の結果を用いません。しかし、ロボットが侵入してはいけない領域など、一般的にセンサから認識することが困難な情報は、地図にその情報を記述しておくことで、自己位置推定の結果から認識することができます。また経路計画を行う場合にも、地図に混雑しているなどの情報を付加できれば、その情報を活用した計画を実施することができます。このような認識や計画は、高度な自動走行を実現するためにきわめて有効に働きます（むしろ、ないと自動走行の実施が難しくなる場合もあります）。現状の自動走行システムでは、地図を活用してさまざまな情報の取得を行うため、正確に自己位置を知ることがきわめて重要になってきます。

図 1.2 に示したとおり、自己位置推定は自動走行において最初に実行されるモジュールとなっています。そして、自己位置推定の後段に続く処理は、自己位置推定の結果が正しいという仮定のもとで実装されていることが多いです。つまり、自己位置推定の失敗は、自動走行の失敗に直結するようなものです。そのため、自己位置推定結果が正しいかどうかについて知ることは、自動走行の安全性を保障するうえで、かなり重大な問題となります。

また、もし上述の自己位置推定の結果が正しいかどうかを知ることができたとしても、推定が正しくなくなった場合には、当然その状態からの復帰処理が必要になります。自己位置推定は基本的に、時系列情報を用いて実行されます。つまり、自己位置推定に失敗している状況とは、この時系列情報に基づいて推定を行った結果であるため、失敗状態から復帰するためには、これまで推定に利用してきた時系列情報とは無関係な、新しい情報を用いる必要が出てきます。すなわち、まったく事前情報がない状態から自己位置推定を行う問題とも解釈できます。このような問題は、**大域的自己位置推定** (global localization) や**初期位置推定** (initial pose estimation) と呼ばれます。なお大域的自己位置推定に対して、初期位置が与えられた状態で移動体の位置を逐次的に推定していく自己位置推定問題は**位置追跡** (position tracking) と呼ばれます。近年では、大規模な地図の利用も可能となってきていますが、大規模環境下で大域的自己位置推定問題を解くことは容易ではありません。

上述のとおり、自己位置推定問題はいまだに多くの問題を抱えています。そしてこれらの問題は、「従来の自己位置推定法の枠組みでは、数学的に扱うことが困難な事象が存在する」ということに起因しています。本書はこのような、従来の自己位置推定法では対処することができない問題に対して、その枠組みを拡張、もしくは新しい手法を導入することで、その解決を目

指した著者の研究をまとめたものです[†]。つまり、単に自己位置推定の精度やロバスト性を向上させるためではなく、「従来の自己位置推定システムでは実現できなかったことを実現できるようにする」という取組みをまとめています^{8)~14)}。この取組みを著者は、「自己位置推定の高度化」と呼んでいます。そして、この高度化に基づく、自己位置推定の高性能化の方法を本書は解説しています。

本書では、論文には記載できないようなプログラムによる具体的な実装例から、必要となる数学的知識をすべて解説しました。さらには、自己位置推定が抱える根本的な課題の解説も行っています。読者はこれら、特に根本的な課題を理解することで、自己位置推定問題とは単に地図上における相対位置を求める問題というだけでなく、異なった視点での課題を有する問題ということが理解できるようになると思います。そして、これらをどのように解決していくかといった具体的な手順が理解できるようになります。「自己位置推定とは、単に位置を求めるものという理解だけでなく、関連して困難な問題が多数存在する」ということを伝えたいというのも、本書を執筆した目的でもあります。

1.2 本書で扱う手法

自己位置推定問題を解く方法は大きく、最適化アプローチ (optimization approach) と確率論的アプローチ (probabilistic approach) の二つに分類されます。最適化アプローチで有名な方法としては、ICP (iterative closest points) スキャンマッチング¹⁵⁾ や、NDT (normal distributions transform) スキャンマッチング¹⁶⁾ がありますが、本書では後者の確率論的アプローチにのみ焦点を当てます。これは、確率論的アプローチのほうが柔軟なモデル化が可能であり、実際に高度化を行うにあたり、理論の展開が容易なためです。また、確率論的アプローチで有名な方法としてはカルマンフィルタ (Kalman filter) やパーティクルフィルタ (particle filter) を用いた方法がありますが、本書ではパーティクルフィルタを用いた実装方法のみを扱います。これは、パーティクルフィルタの実装の容易さと、推定性能の高さの利点のためです。特に6、7章で解説する手法では、ラオ・ブラックウェル化パーティクルフィルタ (Rao-Blackwellized particle filter) を用います。また本書では、グラフィカルモデル (graphical model) を用いた確率モデリングを多用します。

1.3 本書の構成と内容

本書は、本章も含めて10章で構成されています。以下、簡単に各章で解説する内容をまとめ

[†] 上述の大域的自己位置推定に関する方法は9章で解説しますが、厳密には、これは従来の自己位置推定法の枠組みから外れません。しかし、実際にその実装が可能となったのは、深層学習の発展による恩恵が大きいです。その意味で、新しい技術の登場により現実的に実装ができた方法であるため、本書で解説しています。

ます。

2章では、本書が利用するプログラムの開発環境構築方法と、シミュレータについて解説します。もし本書から論理的な部分のみを学びたいと考えている場合は、読み飛ばしていただいて問題ない章です。

3章では、確率的自己位置推定を理解するために必要な数学的知識を解説します。特に本書では、グラフィカルモデルの解説まで行っています。これは、確率変数間の関係をノードとエッジのグラフで表現したものです。文献1)~3)にもグラフィカルモデルに関する記述はありますが、本書では少し複雑なグラフを扱うため、グラフィカルモデルの解説まで行っています。このグラフィカルモデルからの式展開の理解は、新たに自分でモデルを構築する場合に、きわめて有効な手助けになります。この点は、他書と比べた本書の利点であると考えています。

4章では、通常確率的自己位置推定問題の定式化について解説します。またこれに付随して導出される、動作・観測モデルについても解説します。この章は、後で述べる自己位置推定の高性能化を行うためのモチベーションになる箇所を解説している部分にも相当します。

5章では、4章で述べた確率的自己位置推定法をパーティクルフィルタを用いて実装する方法を解説します。もし、パーティクルフィルタを用いた自己位置推定を理解しているなら、この章までは読み飛ばしてください。

6章からが、本書が扱うメインテーマとなります。6章では、環境の変化に対して頑健に自己位置推定を行うことを目的とした、「観測物体のクラスを考慮した自己位置推定法」について解説します。ここでいう観測物体のクラスとは、「地図に存在する・しない」を意味しています。もし「地図に存在しない」という情報が自己位置推定を行う前に理解できていれば、それを無視した自己位置推定を行うことが可能になり、環境変化に対する頑健性が向上します。「自己位置推定の頑健性が向上する」とだけ聞くと、従来の自己位置推定法を改善しているだけのように聞こえますが、この方法は、通常自己位置推定法をモデル化するグラフィカルモデルと異なるグラフによりモデル化されます。すなわち、従来の自己位置推定法を拡張したモデルとなっており、これにより自己位置推定の頑健性を向上させることを実現しています。6章でも詳しく解説しますが、この手法は、観測モデルが持つ問題点解決に寄与する手法にもなっています。

7章では、自己位置推定結果の正しさを知ることを目的とした、「信頼度付き自己位置推定」について解説します。信頼度付き自己位置推定法は、自己位置推定結果の信頼度までを同時推定できるモデルに拡張しています。従来の自己位置推定法を解いても、明示的にその推定結果が正しいと判断できるパラメータは得られません。本手法はそのパラメータを得ることを可能にします。なお、ここでいう信頼度とは、「自己位置推定に成功している確率」を表した値です。つまり信頼度が100%に近ければ、自己位置推定に成功していると判断できます。ただし注意していただきたいのは、本当に正しく「自己位置推定に成功している確率」を答えられるわけではありません（自己位置推定の真値は神のみぞ知る値であり、実際に正しいかどうかを知るすべは存在し得ません）。本章で解説する手法では、自己位置推定の正誤を分類する正誤判

索 引

	【い】		【く】		【せ】
位相的データ解析	146	クラス条件付き観測モデル	65	信頼度	83, 149
位置追跡	3	クラス予測モデル	79	信頼度減衰モデル	89
陰 性	91	グラフィカルモデル	4, 20	信頼度遷移モデル	86
【う】		グラフカット	29		
運動学	37	クリーク	26		
【え】		【こ】			
エネルギー関数	27	子ノード	20	正規化	19
【お】		混合ガウスモデル	136	正規化係数	35
オドメトリ	10	【さ】		正規化相互相関	43
オドメトリ座標系	11	再帰的ベイズフィルタ	36	正規分布	16
親ノード	20	最大事後確率	54	世界座標系	11
【か】		最適化アプローチ	4	遷移行列	116
可観測変数	20	最 尤	55	全確率の定理	19
確信度	88, 149	左右独立 2 輪駆動	10	センサ座標系	11
確率分布	15	残 差	95	センサリセット	62, 131
確率変数	15	【し】		占有格子地図	11
確率密度関数	16	自己位置推定	1		
確率論的アプローチ	4	事後確率	19	【た】	
隠れ変数	20	事後分布	19	大域的自己位置推定	3
隠れマルコフモデル	116	指数分布	39	多層パーセプトロン	93
加法定理	18	事前確率	19		
カルバック・ライブラー・		事前分布	19	【ち】	
ダイバージェンス	61	重点サンプリング	132, 133	長・短期記憶	145
カルマンフィルタ	4	周辺化	18		
慣性航法システム	2	周辺確率	18	【て】	
観測の独立性	38	周辺確率分布	18	提案分布	36, 133
観測モデル	35	重要度係数	133	ディリクレ分布	79
【き】		条件付き確率	18	点 群	1
偽陰性	91	条件付き確率場	126	伝承サンプリング	50
期待値	17	条件付き独立性	21		
教師あり学習	23	状 態	37	【と】	
偽陽性	91	状態方程式	37	動作モデル	36
共分散	17	乗法定理	18	同時確率	17
極大クリーク	26	初期位置推定	3	独 立	17
距離場	41	真陰性	91	ドロップアウト	134
		深層学習	128		
		真 値	10	【は】	
		真陽性	91	パーシステントホモロジー	146
				パーティクルフィルタ	4, 46
				バイナリ変数	15
				判断モデル	85
				【ひ】	
				ビームモデル	40

<p>【ふ】</p> <p>不確かさ 149 分散 16 分配関数 27</p> <p>【へ】</p> <p>平均 16 平均絶対誤差 95 ベイジアンネットワーク 20 ベイズの定理 19 ベータ分布 94</p> <p>【ほ】</p> <p>膨張リセット 62, 131 母数 16 ポテンシャル関数 26 ボルツマン分布 27</p> <p>【ま】</p> <p>マップマッチング 43 マルコフ確率場 20 マルコフ性 36</p> <p>【み】</p> <p>未知変数 20</p>	<p>【む】</p> <p>無向グラフ 20</p> <p>【め】</p> <p>メッセージ 31</p> <p>【も】</p> <p>目標分布 133 モラル化 20 モンテカルロ位置推定 46 モンテカルロドロップアウト 134</p> <p>【ゆ】</p> <p>誘拐状態 129 誘拐ロボット問題 62, 129 有向グラフ 20 有効サンプル数 57 有向非循環グラフ 20 有向分離性 23 尤度 19 尤度場 42 尤度場モデル 41 尤度分布 19</p>	<p>【よ】</p> <p>陽性 91 予測分布 36</p> <p>【ら】</p> <p>ラオ・ブラックウェル化 パーティクルフィルタ 4, 66</p> <p>【り】</p> <p>離散時間状態方程式 37 離散変数 16 履歴センサリセット 130</p> <p>【る】</p> <p>ループあり確率伝播 113, 118</p> <p>【れ】</p> <p>レイキャスティング 41 連鎖グラフ 20 連続時間状態方程式 37 連続変数 16</p> <p>【ろ】</p> <p>ロバスト推定 76 ロボット座標系 11</p>
---	--	--

<p>【A】</p> <p>AdaBoost 91 adaptive MCL 61 augmented MCL 62, 131</p> <p>【E】</p> <p>EM アルゴリズム 116 End-to-End 128</p>	<p>【H】</p> <p>head-to-head 21 head-to-tail 21</p> <p>【M】</p> <p>M 推定 77 mixture MCL 62, 134</p>	<p>【O】</p> <p>One-shot 自己位置推定 128</p> <p>【T】</p> <p>tail-to-tail 21</p> <p>【数字】</p> <p>1-of-K 符号化法 112</p>
--	---	---

— 著者略歴 —

2012年 宇都宮大学工学部機械システム工学科卒業
2013年 宇都宮大学大学院工学研究科博士前期課程修了（機械知能工学専攻）
2016年 宇都宮大学大学院工学研究科博士後期課程修了（システム創成工学専攻）、
博士（工学）
2016年 名古屋大学特任助教
2020年 名古屋大学助教
現在に至る

LiDAR を用いた高度自己位置推定システム

— 移動ロボットのための自己位置推定の高性能化とその実装例 —

An Advanced Localization System Using LiDAR

— Performance Improvement of Localization for Mobile Robots
and Its Implementation —

© Naoki Akai 2022

2022年6月16日 初版第1刷発行

★

検印省略

著者 赤井直紀
発行者 株式会社 コロナ社
代表者 牛来真也
印刷所 三美印刷株式会社
製本所 有限会社 愛千製本所

112-0011 東京都文京区千石 4-46-10

発行所 株式会社 コロナ社
CORONA PUBLISHING CO., LTD.

Tokyo Japan

振替 00140-8-14844・電話(03)3941-3131(代)

ホームページ <https://www.coronasha.co.jp>

ISBN 978-4-339-03240-6 C3053 Printed in Japan

(森)



JCOPY <出版者著作権管理機構 委託出版物>

本書の無断複製は著作権法上での例外を除き禁じられています。複製される場合は、そのつど事前に、出版者著作権管理機構（電話 03-5244-5088、FAX 03-5244-5089、e-mail: info@jcopy.or.jp）の許諾を得てください。

本書のコピー、スキャン、デジタル化等の無断複製・転載は著作権法上での例外を除き禁じられています。購入者以外の第三者による本書の電子データ化及び電子書籍化は、いかなる場合も認めていません。落丁・乱丁はお取替えいたします。