

環境知能化による移動ロボットの モンテカルロ位置推定法の性能向上

○池勇勲（東京大学） 山下淳（東京大学） 浅間一（東京大学）

1. 序論

移動ロボットのナビゲーションにおいて地図上での現在位置を正確に認識する問題は非常に重要である。地図上での位置推定問題において、車輪のオドメトリ情報とロボットに装着された距離センサ（以下、オンボードセンサ）情報をカルマンフィルタおよびパーティクルフィルタを用いて融合し、位置推定を行う様々な手法が提案されている[1]。これらの手法の中で、パーティクルフィルタに基づくモンテカルロ位置推定法（MCL: Monte Carlo localization）は、オンボードセンサベースの位置推定問題に対して優れた性能が保証されているため、よく使用されている。しかし、人とロボットが共存する環境においてこれらの従来手法では解決が困難である場合が多く存在する。

図 1(a)に建物の内部に多く存在する単調で対称的な環境の特定の位置で測定したオンボードセンサの距離情報を、図 1(b)に、この距離情報に基づいて地図上のすべての位置での尤度を計算した結果を示す。オンボードセンサだけでは単調な環境内の区別が明確にできないため、ロボットの存在する確率（黒の明るさ）が高い場所が1つに決まらない。すなわち、このような環境において、オンボードセンサ情報だけでは位置推定に明確に役立つ情報を得られないため、特に検出範囲が狭いオンボードセンサを用いる場合、従来手法の適用は困難である。

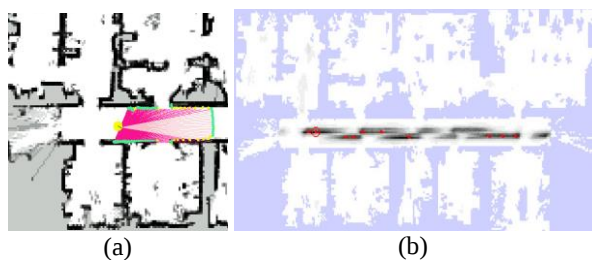


図 1 単調な環境：(a)距離センサ情報，(b)各位置での尤度[1]。

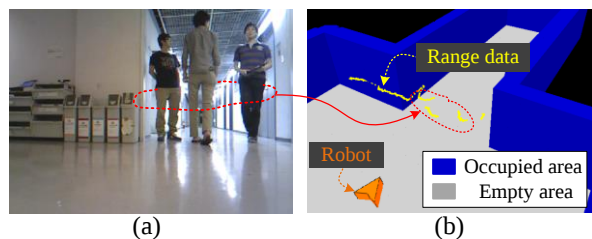


図 2 混雑環境：(a)実環境と動的物体，(b)距離センサ情報と地図の一部。

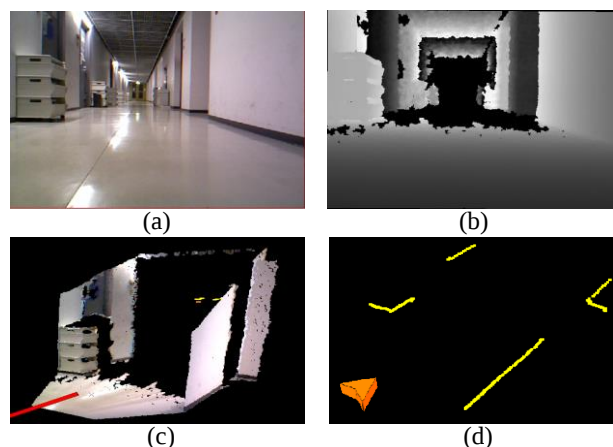


図 3 RGB-D センサからの情報：(a)RGB 画像，(b)距離画像，(c)3 次元点群データ，(d)2 次元距離情報。

また、近年、図 2(a)のように、人と共存する環境でのロボットの運用が増加している。このような混雑環境でロボットを運用する場合、図 2(b)に示すように、地図上に存在しない動的物体が距離センサによって検出されてしまう。この現象も位置推定問題を非常に難しくする原因である。

一方、人間を取り巻く環境内に分散センサインフラ（以下、環境センサ）を構築する知能化空間（iSpace: intelligent space）概念が提案されている[2]。このような知能化空間で、オンボードセンサだけでなく、環境センサも同時に活用し、ロボットの位置推定問題を解決する手法が提案された[3,4]。しかし、これらの研究では、環境センサ情報から、動的物体とロボットを区別するため、ロボットに付加的な装置を設置したり、ロボットの形状に対するモデルをあらかじめ定義するなどの事前知識が必要である。

以上のような問題をふまえ、本論文では、従来のオンボードセンサベースのモンテカルロ位置推定法に環境センサからの情報を追加した新しいモンテカルロ位置推定法を提案する。オンボードセンサ情報からの更新によるパーティクルの確率分布と、環境センサ情報から生成される動的物体の確率分布を融合し、パーティクルの重み付けに用いることで、図 1, 2 のようなオンボードセンサのみでは困難である環境においても、ロバストな位置推定を可能にする。

本論文においては、オンボードセンサと環境センサとして RGB-D センサを使用する。RGB-D センサからは、図 3(a), (b)に示すように、環境に対する RGB 画像と距離画像をリアルタイムで取得可能である。また、これら进行处理することで、図 3(c), (d)のよう

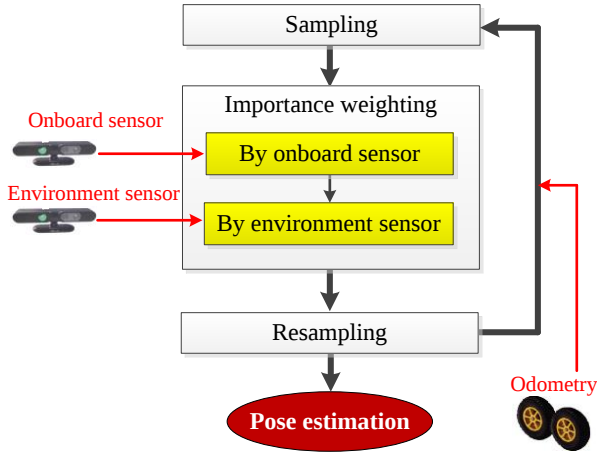


図4 提案するモンテカルロ位置推定法.

な3次元点群データと2次元距離情報の生成が可能である. オンボードセンサとしてのRGB-Dセンサからはロボットの周辺の距離情報を取得可能であるものの, RGB-Dセンサの特徴の上, 検出範囲が非常に狭い. 環境センサとしてのRGB-Dセンサからは背景差分手法を用いることで, 環境内の動的物体を検出可能であるものの, ロボットと人間などの動的物体の区別は一般に不可能である. すなわち, 提案手法はこれらの2つのセンサ情報に対する長短所を相互補完する.

2. 知能化空間におけるモンテカルロ位置推定

提案手法の全体的なプロセスを図4に示す. モンテカルロ位置推定法は基本的にSIR (sampling importance resampling) アルゴリズムに基づくパーティクルフィルタを用いる[5]. パーティクルフィルタはロボットの位置と姿勢に対する確率分布を次のような重み付けられたパーティクルの集合 $\mathbf{X}_t$ として表現する.

$$\mathbf{X}_t = \{ \langle \mathbf{x}_t^{(i)}, w_t^{(i)} \rangle | i=1 \dots N_t \} \quad (1)$$

ここで, $\mathbf{x}_t^{(i)} = [x_t^{(i)} \ y_t^{(i)} \ \theta_t^{(i)}]^T$ は i 番目のパーティクルの位置と姿勢であり, N_t はパーティクルの数である.

SIR アルゴリズムは以下の3ステップに分けられ, 提案手法では, 重み付けステップに環境センサから検出された動的物体情報を追加することで, オンボードセンサのみでは対応しきれない環境においてもロボットの位置推定を可能にする.

2.1 サンプリング (Sampling)

サンプリングステップでは次式のように移動ロボットの運動モデル $\mathbf{f}(\cdot)$ に, ランダムノイズを追加して, 各パーティクルを移動させる.

$$\mathbf{x}_t^{(i)} = \mathbf{f}(\mathbf{x}_{t-1}^{(i)}, \mathbf{u}_t) + \mathbf{e}_t \quad (2)$$

ここで, $\mathbf{u}_t$ はオドメトリ情報を意味し, $\mathbf{e}_t$ はランダムノイズである.

2.2 重み付け (Importance Weighting)

重み付けステップでは, オンボードセンサ情報と環境センサ情報に基づき, 各パーティクルの重みを付与する.

2.2.1 オンボードセンサによる重み付け

オンボードセンサによる重み付けは, 次式のように, あるロボットの状態に対し, 現時点の観測値 $\mathbf{z}_t$ が得られうる確率を意味する尤度を計算して付与する.

$$w_t^{(i)} \leftarrow \eta P(\mathbf{z}_t | \mathbf{x}_t^{(i)}) \quad (3)$$

ここで $P(\mathbf{z}_t | \mathbf{x}_t^{(i)})$ に対する尤度関数は, 様々に定義可能であり, 本論文では, 次式のようにガウス関数を使用して定義する.

$$P(\mathbf{z}_t | \mathbf{x}_t^{(i)}) = \prod_{k=1}^L \frac{1}{\sqrt{2\pi\sigma_z^2}} \exp\left(-\frac{(z_t^k - \hat{z}_t^k)^2}{2\sigma_z^2}\right) \quad (4)$$

ここで z_t^k はオンボードセンサからの k 番目の距離情報, $\hat{z}_t^k$ は $\mathbf{x}_t^{(i)}$ の状態での予測される k 番目の距離情報, L は1回環境を観測したとき得られる距離情報の数である.

2.2.2 環境センサによる重み付け

環境センサによる重み付けは, 図6(a), (b)に示すような混合正規分布に基づく背景差分法[6]を用いて検出した環境内の動的物体情報に対応する点群データと, パーティクルの位置分布に対して, それぞれの確率分布を推定し, 次式のようにパーティクルの最終的な重みを決定する.

$$w_t^{(i)} \leftarrow \eta w_t^{(i)} \frac{P_{\text{ISpace}}(\mathbf{x}_t^{(i)})}{P_{\text{onboard}}(\mathbf{x}_t^{(i)})} \quad (5)$$

ここで η はすべてのパーティクルの重みの和を1にするための正規化定数である. また, $P_{\text{onboard}}(\cdot)$ と $P_{\text{ISpace}}(\cdot)$ は, それぞれ動的物体とパーティクルの位置情報に対する確率密度関数を意味し, それぞれの確率密度関数 $P(\cdot)$ を求めるため, 次式のようなカーネル密度推定法を使用する.

$$P(\mathbf{x}) = \frac{1}{Nh^D} \sum_{i=1}^N K\left(\frac{\mathbf{x} - \mathbf{x}^{(i)}}{h}\right) \quad (6)$$

$$K(\mathbf{x}) = \frac{1}{(2\pi)^{D/2}} \exp\left(-\frac{1}{2} \mathbf{x}^T \mathbf{x}\right) \quad (7)$$

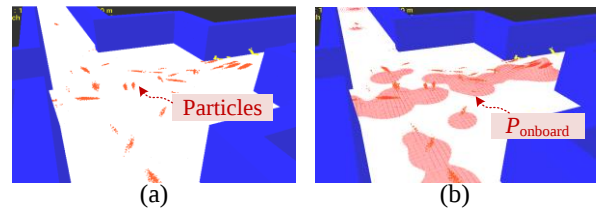


図5 オンボードセンサからの率密度関数推定: (a)パーティクルの分布, (b)推定された確率密度関数 P_{onboard} .

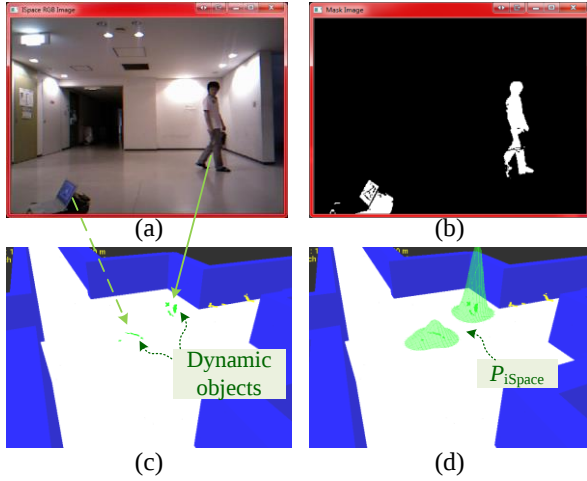


図 6 環境センサからの率密度関数推定：(a)入力画像，(b)背景差分による動的物体検出結果，(c)動的物体を表す点群の分布，(d)推定された確率密度関数 P_{iSpace} 。

推定された確率密度関数 $P_{onboard}(\cdot)$ と $P_{iSpace}(\cdot)$ をそれぞれ図 5(b) と図 6(d) に示す。ここで、カーネル関数 $K(\cdot)$ は標準的なガウス関数（平均が $\mathbf{O}$ で分散が $\mathbf{I}$ ）を採用しており、 h 平滑化パラメータ、 D はデータの次元数（本論文では 2）、 $\mathbf{x}$ は確率推定対象の位置を意味する。また、 $\mathbf{x}^{(i)}$ は $P_{onboard}(\cdot)$ を推定する場合、パーティクルの位置を意味し、 $P_{iSpace}(\cdot)$ を推定する場合、動的物体を表す点群の位置座標を意味する。すなわち、パーティクルと動的物体を表す点群の分布からそれぞれの確率密度関数を推定し、図 7(a) と式 (5) のようにパーティクルの重みを決定する。

2.3 リサンプリング (Resampling)

リサンプリングステップでは、各パーティクルの重みに従い、パーティクルを再配置する。すなわち、高い重みを持つパーティクルからは、次の段階のパーティクルを多く生成し、低い重みを持つパーティクルからは、少ない数のパーティクルを生成、あるいはパーティクルを生成しない。パーティクルの重みに従い、再配置を行った後、各パーティクルの重みを $w_t^{(i)} = 1/N_t$ にリセットする。リサンプリングを行った結果を図 7(b) に示す。

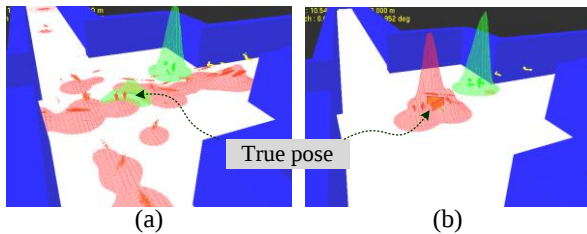


図 7 パーティクルの重み付けとリサンプリング：(a)各確率密度関数により重み付けられたパーティクル，(b)リサンプリングにより再配置されたパーティクルと各確率密度関数。

ここで、ロボットの位置と姿勢に対する推定値は次式のようにパーティクルの加重平均により求められる。

$$\hat{\mathbf{x}}_t = \frac{1}{N_t} \sum_{i=1}^{N_t} w_t^{(i)} \mathbf{x}_t^{(i)} \quad (8)$$

これにより、実際のロボットの位置周辺にパーティクルが多く残り、安定した自己位置推定が可能となる。

3. 実験結果

提案した知能化環境でのモンテカルロ位置推定法の検証のため、図 8(a) のような単調であり、人間などの動的物体が頻繁に登場する建物の廊下環境で実験を行った。実験装置としては、図 8(b), (c) に示すように MobileRobot 社の Pioneer 3-DX ロボット、オンボードセンサと環境センサとしての RGB-D センサは ASUS 社の Xtion live pro を使用した。

図 9 はオンボードセンサだけを用いた従来のモンテカルロ位置推定法と、オンボードセンサと環境センサ情報の融合による提案手法との比較実験を行った結果である。ここで、使用したパーティクルの最大個数は 10,000 個、最小個数は 1,000 個である。序論で前述したように、単調かつ動的物体が存在する環境では、オンボードセンサからの距離情報を信頼できないため、従来手法による移動ロボットの大域的推定の成功率はおよそ 40% であった。一方、提案手法では、環境センサ情報を組み合わせ、移動ロボットの位置に対する確率情報の適切な更新を追加的に行うため、ほぼ 90% 以上の成功率で大域的推定が可能であった。

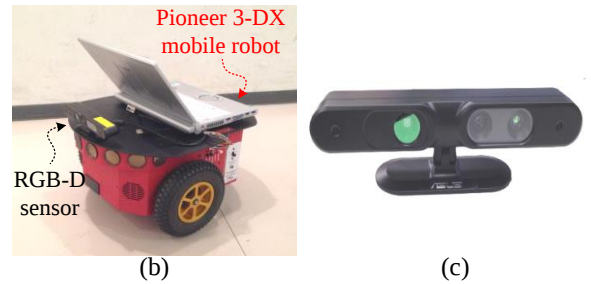
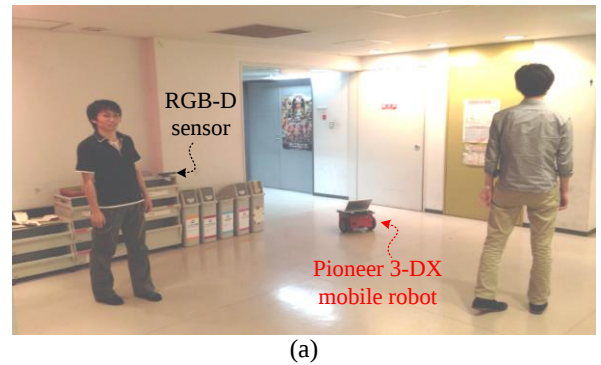


図 8 実験の様子：(a)実験環境，(b)移動ロボットシステム，(c)RGB-D センサ（ASUS Xtion live pro）。

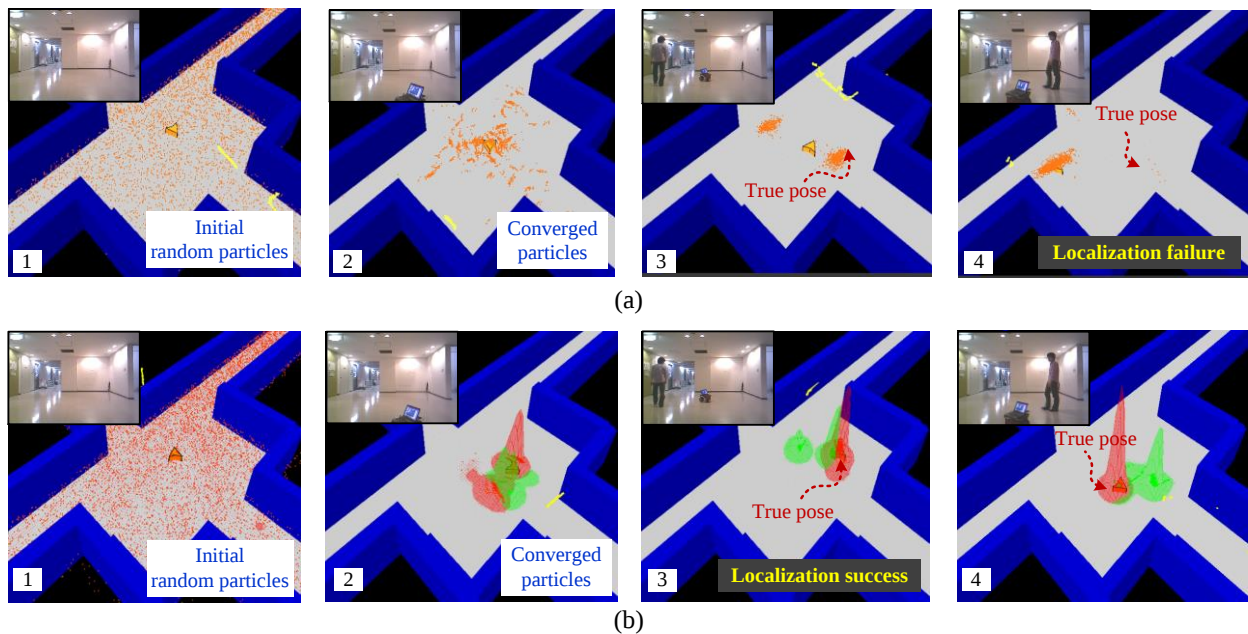


図 9 実験結果：(a)従来手法（オンボードセンサ情報によるモンテカルロ位置推定法）による大域的位置推定結果, (b)提案手法（オンボードセンサ情報と環境センサ情報の融合によるモンテカルロ位置推定法）による大域的位置推定結果。

図 10 は時間に対してパーティクルの位置分布に対する標準偏差を比較したグラフである。提案手法のほうのパーティクルの分布が、より速い速度で収束した。すなわち、大域的位置推定の完了時間も、従来手法より短く可能であった。

4. 結論

本論文では、オンボードセンサから取得したロボット周辺の距離情報と、環境センサから取得した動的物体情報を確率的に融合し、単調かつ混雑環境においても、安定的な位置推定を行うモンテカルロ位置推定法を提案した。このため、オンボードセンサから推定した確率分布と、環境センサから推定した確率分布をパーティクルフィルタの重み付けステップに応用する手法を提案した。また、より高い成功率で大域的位置推定が可能であることが実験結果より証明された。

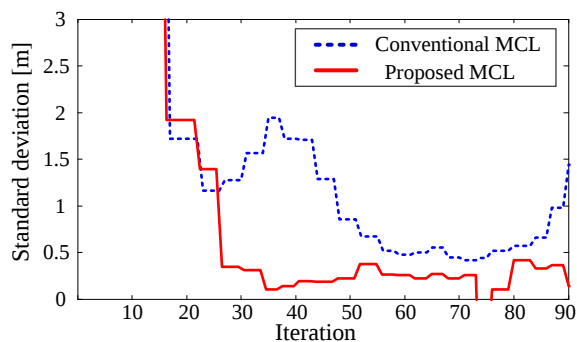


図 10 従来手法と、提案手法とのパーティクルの分布に対する標準偏差比較結果。

今後の展望としては、提案手法はロボットが環境センサから検出できない範囲に存在するときには対応できないため、より多様な場合に対応可能な手法を考慮する計画である。

謝辞

本研究の一部は、栢森情報科学振興財団の援助を受けた。

参考文献

- [1] S. Thrun, W. Burgard, and D. Fox, Probabilistic Robotics, The MIT Press, 2005.
- [2] J.-H. Lee and H. Hashimoto, "Intelligent Space - Concept and Contents", *Advanced Robotics*, Vol. 16, No. 3, pp. 265-280, 2002.
- [3] P. Jayasekara, T. Sasaki, H. Hashimoto, and T. Kubota, "Simultaneous Localization Assistance for Mobile Robot Navigation in Real, Populated Environments", *Journal of Control, Measurement, and System Integration*, Vol. 5, No. 6, pp. 349-358, 2012.
- [4] D. Brscic and H. Hashimoto, "Robot Localization using Distributed and Onboard Sensors, Robotics 2010 Current and Future Challenges", pp. 401-418, InTech, 2010.
- [5] A. Doucet, N. Freitas, and N. Gordon, Sequential Monte Carlo Methods in Practice, Springer, 2001.
- [6] C. Stauffer and W. Grimson, "Adaptive Background Mixture Models for Real-time Tracking", *Proceedings of the 1999 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Vol. 2, pp. 246-252, 1999.