

指示を求めるオフィスロボットの開発

Development of Office Robot Asking for Directions

田下 裕一† 中村 恭之†
Yuuichi Tashita Takayuki Nakamura

1. はじめに

近年、ロボットを一般社会へ普及させるため、様々な環境で動作可能なロボットを開発するための研究が盛んに行われてきている。そのような研究の 1 つとして、移動ロボットの自律ナビゲーションシステムに関する研究がある。この研究では、移動ロボットにとって基本的な機能である、自己位置推定と地図構築に関して様々な手法が提案されている[1][2][3][4]。地図構築は、自己位置推定や経路計画に必要であり、移動ロボットにとって最も必須な機能である。自己位置推定に関しても、経路計画された経路に従って、自分の位置を確認しながら正しく目的地に向かうために必要な機能である。本研究ではこの自己位置推定機能に注目する。

自己位置推定は、一般的にセンサデータに基づいて処理が行われ、実世界を観測したセンサデータには不確実性が伴うために、自己位置推定を行う方法は、状態遷移のマルコフ性を仮定した確率に基づく方法がよく用いられる。そのような方法の中でも、パーティクルフィルタに基づく自己位置推定法が、センサデータのノイズに対して頑健な方法として有名である[1][2]。しかしながら、この手法では、ベイズの公式だけで状態遷移が推定できないような状況に陥ると、誤った自己位置推定を行ってしまう。このような問題は、Kidnapped Robot Problem と呼ばれ[4]、最適性の保証されたこの問題を解決する解法は、未だに提案されていない。

そこで、本研究では、最適性は保証しないが、この Kidnapped Robot Problem を解決する 1 つの方法として、この問題が発生するような状況に陥りそうになると、ロボット自身が周りにいる人間に指示を求めることで、この問題を回避するシステムを搭載したオフィスロボットの開発を目指している。そこでまず、本報告では、Kidnapped Robot Problem が発生しそうな状況を検出する方法と、人間の指示を認識する方法を提案する。以下、第 2 節では、提案する方法についての説明を述べ、第 3 節では、提案する方法に関する実験結果について述べる。最後、第 4 節では、本報告のまとめと今後の予定について述べる。

2. 指示を求めるオフィスロボット

2.1 問題設定

本研究では、車輪型移動ロボットが床面が平らなオフィス環境で動作する状況を想定している。車輪型移動ロボットには、センサとして、レーザーレンジファインダが搭載されており、これによって地図構築、自己位置同定を行う。オフィス環境の地図は、あらかじめ構築されているものとする。また、ロボットには人間の指示を観測するためのセンサとして、Microsoft Kinect センサー

を搭載している。Kinect センサーには、RGB カメラと 3 次元深度カメラを内蔵している。また、このセンサーから情報を取得するためのライブラリとして OpenNI[5]を利用する。このライブラリを用いることで、3 次元深度カメラの情報をもとにして、人間の骨格認識・実時間追跡が可能になる。これを利用して人間の指示をロボットが認識するシステムを構築する。

本研究で想定しているシナリオは次のようなものである。あらかじめ構築された地図上で、人間が出発点、ゴール地点を指定し、ロボットが経路計画を行い、計画された経路上を移動し始める。移動中は常に自己位置同定を行い、計画された経路に追従するようにロボットは動く。移動中、地図構築を行ったときから環境の状態が変化したため、自己位置同定の結果の不確実性が増し、Kidnapped Robot Problem が起こる。このような場面において、本論文で提案する方法によって、自己位置同定の結果の不確実性が増すことを検出し、その時点で、人間の指示を認識して、この問題を回避し、ロボットがゴール地点に到達する。

2.2 パーティクルフィルタを用いた自己位置推定

自己位置推定とは、地図を与えられているとしてロボットが、ロボットに搭載されたセンサから取得されるデータをもとにして地図上のどこにいるのかを推定する問題である。自己位置推定でのロボットの位置姿勢は、状態ベクトル $\mathbf{r}_t = (x_t, y_t, \theta_t)^T$ で表す。 (x_t, y_t) は、地図上でのロボットの座標値を、 θ_t は、ロボットの向きを表す。以下では、文献[2]に従って、パーティクルフィルタを用いた自己位置同定法について説明する。

パーティクルフィルタに基づく方法では、多数のパーティクルでロボットの位置姿勢の確率分布を表す。 i 番目のパーティクル $p_t^{(i)}$ は、ロボットの位置姿勢 $\mathbf{r}_t^{(i)} = (x_t^{(i)}, y_t^{(i)}, \theta_t^{(i)})^T$ と重み $w_t^{(i)}$ を持つ。パーティクルフィルタに基づく方法の手順を以下に示す。

① 動作モデルによる $\mathbf{r}_t^{(i)}$ の予測

各パーティクル $\mathbf{r}_{t-1}^{(i)}$ を、ロボットの動作コマンド \mathbf{a}_t と、動作の誤差分布（普通は、正規分布）に従って乱数 $\boldsymbol{\varepsilon}_t^{(i)}$ を発生させて、この乱数値をに加えて時刻 t におけるパーティクルを生成する。

② 計測モデルによる重み計算

各パーティクルから見えるランドマークの位置とセンサデータとの一致度から尤度を計算して重みとする。重みは合計が 1 になるように正規化する。

③ リサンプリング

各 $\mathbf{r}_t^{(i)}$ に対して $w_t^{(i)}$ に比例した個数だけ複製

† 和歌山大学 システム工学部 情報通信システム学科,
Wakayama University, Faculty of Systems Engineering, Dept. of Computer and
Communication Sciences

を作り時刻 $t+1$ のパーティクルとする。

パーティクルフィルタによる自己位置推定では、各パーティクルがロボットの位置姿勢を模擬している。動作モデルに従ってパーティクルが移動して、その位置から見えるべきランドマークが予測と実測でよく一致すれば重みが大きくなる。重みの大きなパーティクルはリサンプリングによって多くの複製が作られ、その周辺の密度が高くなる。このパーティクルの密度でロボットの位置姿勢の分布を表す。最終的に、ロボットの位置姿勢は、

$$\tilde{\mathbf{r}}_t = \sum_{i=1}^N w_t^{(i)} \mathbf{r}_t^{(i)}$$

によって推定する。

2.3 自己位置同定失敗の検出

次に、自己位置同定の失敗を検出する方法について述べる。あらかじめ用意した地図で表現されている環境と、ロボット走行時の環境とに差異が生じると、自己位置同定する際に、環境の変化した部分のデータの影響で、自己位置同定結果の信頼性が低くなる。パーティクルフィルタを用いた自己位置同定では、全てのパーティクルに関する分散共分散行列 \mathbf{S}_t は

$$\mathbf{S}_t = \frac{1}{N} \sum_{i=1}^N (\mathbf{r}_t^{(i)} - \tilde{\mathbf{r}}_t)(\mathbf{r}_t^{(i)} - \tilde{\mathbf{r}}_t)^T$$

と求めることができ、この \mathbf{S}_t の固有値の大きさを評価することで自己位置同定結果の信頼性を評価することができる。本研究では、複数ある固有値の比を計算し、固有値比の変化量がしきい値以上になったときに自己位置同定が失敗したと判定する。

2.4 ジェスチャ認識の概要

ロボットに搭載されたジェスチャ認識用のセンサ Microsoft Kinect を用いて、ロボットの周囲にいる人間を観測して、人間の指示を認識する。Kinect の 3 次元深度カメラの情報をもとにして、OpenNI ライブラリを使用することで人間の骨格認識・実時間追跡が可能になる。これにより、人間の、頭、首、両肩、両肘、両手、胴体、両尻、両膝、両足首の 3 次元座標値を取得できる。本研究では、人間の指示を認識するために、両肩、両肘、両手の 3 次元座標値を利用して、両肩を結ぶ線 \mathbf{A} と、肘と手を結ぶ線 \mathbf{B} の間の角度 θ を以下のように求め、この角度によって人間の指示を認識する。

$$\cos \theta = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\| \|\mathbf{B}\|}$$

ただし、これらの 3 次元座標値は、人間の指示の認識を安定化するために、手が静止しているときに求められたものを使用する。手が静止しているかどうかの判定は、手の 3 次元座標値を時々刻々と取得し、時刻 t における手の速度を、時刻 t と $t+1$ の座標値から求める。そして、手の速度がしきい値以下になった時に、手が静止したと判定する。

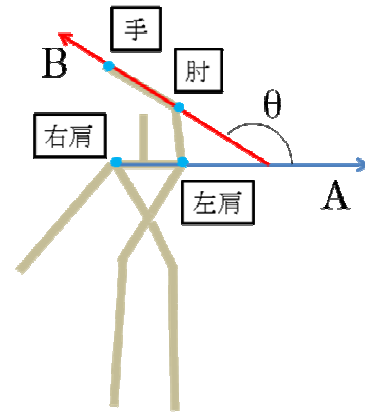


図 1. 指示方向の算出方法

図 2 に、Kinect を用いた人の指示方向の認識手法の手順を示す。

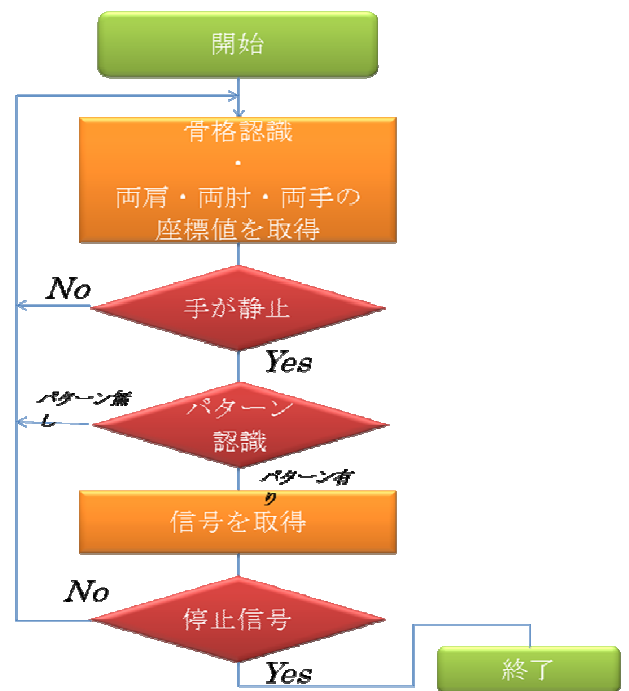


図 2. 指示方向の認識方法

3. 検証実験

3.1 実験環境

実験に使用した機材は、移動ロボット(Pioneer 3-DX), レーザレンジファインダ (Sick 製 LMS-100), ノート PC(OS: Ubuntu10.04LTS, CPU: Corei7 620M)である。



図 3. 実験に使用した機材

3.2 実験結果（自己位置同定失敗の検出）

図 4 にロボット走行中の自己位置同定結果の一例を示す。

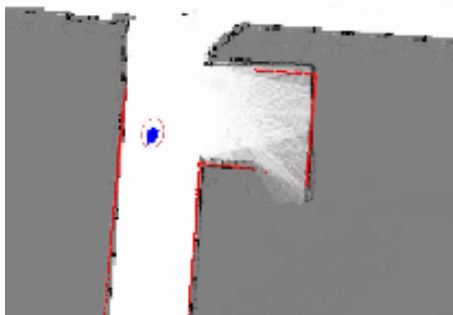


図 4. 自己位置同定結果の例

実験では、車輪型移動ロボット用の MRPT ライブラリ [6] を用いて、地図構築、自己位置同定を行っている。この図の中で示されている楕円が、自己位置同定結果の信頼性を表す分散共分散行列で示される楕円である。青色の点群が、自己位置同定に使用している全てのパーティクルを示している。

次に、先述した自己位置同定失敗検出方法の検証を行った。ある部屋の地図を予め生成し、ロボットが自己位置同定を行いながら移動している最中に、分散共分散行列の固有値の比を求めた。また、ロボットが動き始めた後、地図には載っていない物体を部屋の中に設置した。図 5 に、横軸にロボットが移動し始めたときを 1 としたステップ数、縦軸に固有値の比の値をとったグラフを示す。

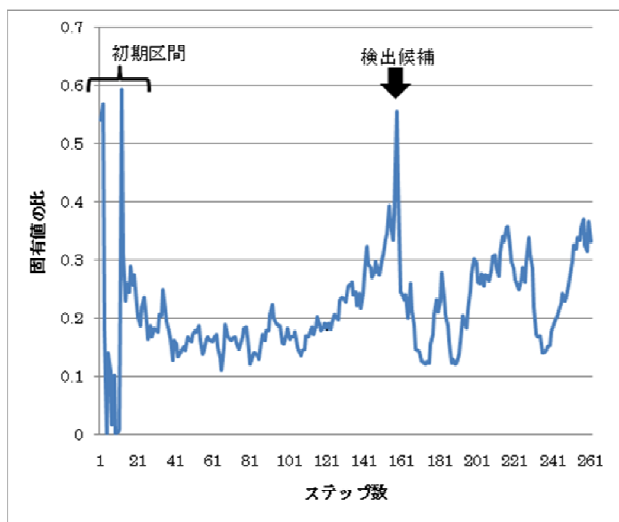


図 5. 固有値の比の変化の様子

図 6 (a) に、図 5 内の「初期区間」における自己位置同定結果、図 6 (b) に、図 5 内の「検出候補」における自己位置同定結果を示す。図 6 (a) では、パーティクルフィルタによる自己位置同定結果がパーティクルの初期配置から収束していない状態にあるので、この時点における観測データ（図内の赤色の点群）と地図とが大きくずれている。これが原因で、固有値の比の値も多くなっている。図 6 (b) では、地図には描かれていない物体が、この時点で初めて観測されたために、パーティクルフィルタによる自己位置同定結果が不安定になっている。これに伴い、

図 5 内の「検出候補」における固有値の比の値が他の時刻に比べて大きな値となっている。従って、この実験においては、提案手法によって、しきい値を 0.4~0.5 に設定しておけば、自己位置同定の失敗が検出できることが分かった。

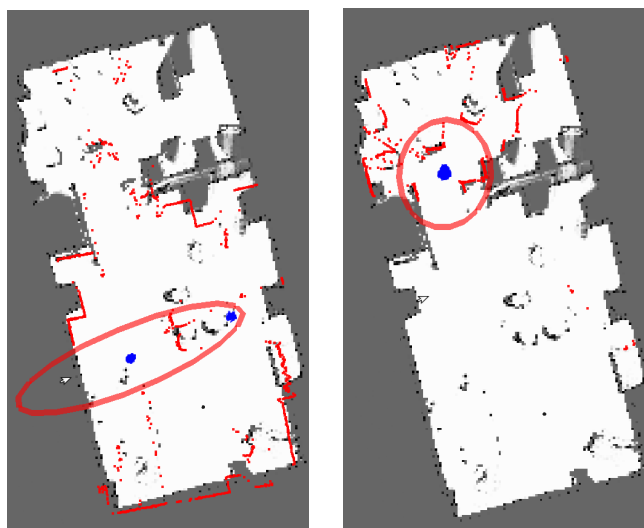


図 6. 信頼性の低い自己位置同定結果の例

3.3 実験結果（ジェスチャ認識の結果）

本報告では、人間の指示の認識の例として、人間がセマフォ手旗信号で指示を与えることを想定した場合の実験結果について報告する。セマフォ手旗信号とは、元は 19 世紀に海上で、遠隔通信する手段として用いられていた。現在では、口頭通信や電子通信を行うことが困難な山岳でのコミュニケーションの手段として用いられている。セマフォ手旗信号は、一般的に両手に持った旗などを用いて視覚的なパターンによってある文字を伝えるための信号システムである。旗が特定の位置で静止することで、その位置に対してあるアルファベットの文字を割り当てて視覚的なパターンを符号化する。片手（1 つの旗）で表現できるパターンとしては、図 7 のような 7 パターンある。このパターンを両手で使い、組み合わせることで、より多くの種類の文字に符号化することができる。

実験では、本来ならば手に旗を持った状態で、セマフォ手旗信号を送らなければならないが、今回は旗を持たずに手の位置情報で、手旗信号を認識した。Kinect からのデータを OpenNI ライブラリを使用して、両手、両肘、両肩の 3 次元位置情報を得て、指示認識のための角度 θ を計算して、図 7 に示したセマフォ手旗信号の 1 つのパターンを認識する。算出した角度 θ を、図 8 左に示す 7 つの領域に分類して、人間の手が領域 1~7 のどの領域に属するかを識別する。各領域は、人間の肩を中心とした中心角 30° の扇形をしている。右肩と左肩を結び直線を基準にして、反時計回り 45° ごとに 135° まで、同様に時計回りに 45° ごとに 135° まで、各扇形の基準線を設定し、その基準線を中心にして $\pm 15^\circ$ の領域を扇形としている。このようにして合計 7 個の扇形の領域を設定する。片手に対して 7 個の領域のどの領域に属するかを求め、左右両手で 49 個のパターンを識別できるこ

とになる．この 49 個のパターンとセマフォ手旗信号の
 パターンの対応表をあらかじめ作成しておくことで，セ
 マフォ手旗信号を認識する．

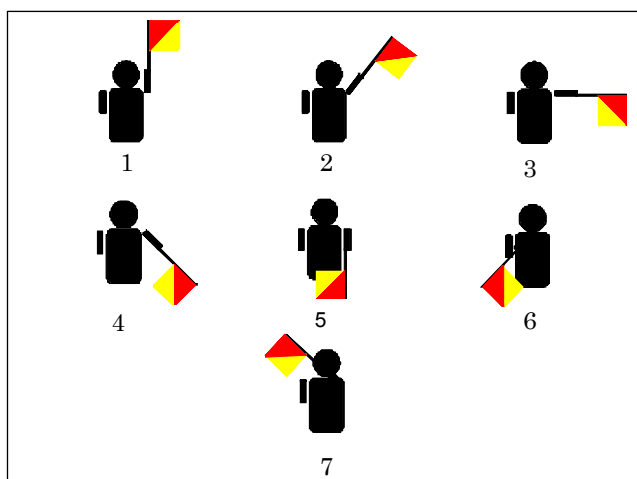


図 7. セマフォ手旗信号の片手のパターン

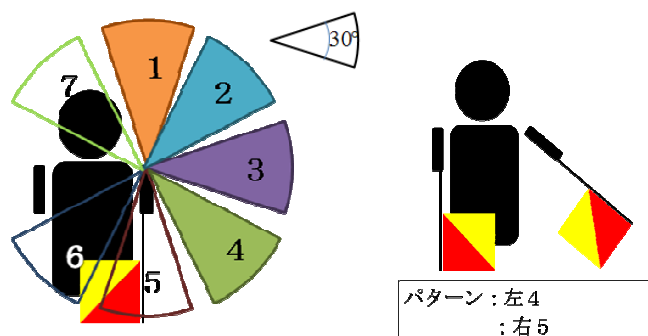


図 8. パターン認識の方法

実験では，セマフォ手旗信号認識の誤認識を防ぐために，
 アルファベット以外に，発信信号や終了信号といった付
 加的なパターンを設定することによって，ジェスチャー
 認識の開始や，終了の判定を行う．

以上をまとめると，ジェスチャー認識実験の手順は以下
 のようになる．人間はまず発信信号を送り，その後，ア
 ルファベットに相当する信号を発信する．手が静止して
 いると判断できるときに，両手の指示認識のための角度
 θ を算出し，両手の角度パターンとセマフォ手旗信号の
 対応表に基づいて，手旗信号で送られた文字を識別する．
 人間が終了信号を送ったときに，文字の識別を終了する．
 実験では，両手，両肩，両肘の 3 次元座標値は，25ms 毎
 に取得している．同じ文字を連続して認識するために，
 手旗信号パターンが 50ms 継続したものを 1 文字として認
 識した．図 9 に，人間がセマフォ手旗信号でアルファベ
 ットの「A」を送っているときの，提案手法による認識
 結果を示す．この図を見ればわかるように，正しく認識
 できている．紙面の都合上，全てのアルファベットに関
 する認識結果を載せることはできないが，提案手法によ
 り全てのアルファベットの認識は成功している．



図 9. セマフォ手旗信号の認識結果

4. まとめと今後の課題

本研究では，最適性は保証しないが，Kidnapped Robot
 Problem が発生するような状況に陥りそうになると，ロ
 ボット自身が周りにいる人間に指示を求めることで，こ
 の問題を回避するシステムを搭載したオフィスロボットの
 開発を目指して，必要不可欠な 2 つ機能について提案
 した．

1 つは，Kidnapped Robot Problem の発生を事前に検出
 する方法と，もう 1 つは人間の指示を認識する方法である．
 実験結果より，提案した方法が有効に働くことが確認で
 きた．

今後は，これらの方法を搭載した自律ナビゲーション
 システムを構築し，自己位置同定の失敗から復帰して，
 最終的にゴールまで到達することを，実際のロボットを
 用いた実験を通じて，検証していく予定である．

謝辞

本研究の一部は，科研費（基盤研究(C) No. 23500243）の
 助成を受けたものである．

参考文献

- [1] S.Thrun, W.Burgard and D.Fox:Probablistic Robotics. MIT Press, 2005.
- [2] 友納正裕:移動ロボットのための確率的な自己位置同定と地図構築, 日本ロボット学会誌 Vol.29, No.5, pp.423-426, 2011.
- [3] Hans Moravec, Sensor fusion in certainty grids for mobile robots, AI Magazine, v.9 n.2, p.61-74, July/August 1988
- [4] S.P. Engelson and D.V. McDermott : "Error correction in mobile robot map-learning". *Proc. Of International Conference on Robotics and Automation.* pp. 2555-2560, 1992.
- [5] OpenNI: <http://www.openni.org/>
- [6] The Mobile Robot Programming Toolkit <http://www.mrpt.org/>