

GPSおよびオドメトリ機能を搭載した移動ロボットの 不整地フィールドにおける位置推定実験

大谷 和彦（東北大学） 永谷 圭司（東北大学） 吉田 和哉（東北大学）

Position Estimation Experiment in Uneven Field for a Mobile Robot using GPS and Odometry

K.Otani(Tohoku Univ.), K.Nagatani(Tohoku Univ.), K.Yoshida(Tohoku Univ.)

Abstract :

One of the important technologies for outdoor mobile robot is 3D localization. Generally, GPS is a very common method to estimate its location in an outdoor environment. However, GPS signals might be blocked because of shield objects, such as trees, rocks, or ceilings of tunnels, above the GPS. To counter such a problem, other positioning sensors and a sensor fusion technique are generally required. In this research, we aim to realize a sensor fusion method for localization of field mobile robots using particle filtering technique. Envisioned sensors are GPS, odometry, 3 gyroscopes, and visual odometry. To validate effectiveness of our method, we implemented the sensor fusion method on our field vehicle and conducted some experiments. In this paper, we introduce our sensor fusion method and report the experimental results.

1 緒言

近年、移動ロボットの活躍が望まれる作業フィールドは多岐にわたり、その一つが屋外へと広がっている。屋外移動ロボットが作業を行う上で欠かせない技術が、自己位置・姿勢の推定である。一般的には、GPSの利用により、近年、高精度に自己位置を推定することが可能となりつつあるが、GPS受信機の上部に遮蔽物が存在する場合、観測が不可能になるという欠点を持つ。一方、車輪の回転を計数することで移動速度を取得し、これを積分して自己位置を推定するオドメトリ手法や、環境中の物体の移動速度をカメラ（またはステレオカメラ）で獲得して移動速度を取得し、自己位置を推測するビジュアルオドメトリ手法[1][2]といった方法も提案されているが、いずれの場合も、速度を積分して位置を求めるため、累積誤差が存在するという欠点を持つ。特に、屋外移動ロボットでは、走行距離が長い場合、累積誤差問題の影響は大きい。そこで、このような複数の位置推定手法を融合させることが重要となるが、近年、その一手法としてパーティクルフィルタが提案されている[3]。本研究では、屋外移動ロボットの高精度な位置推定の実現を目指し、上述のパーティクルフィルタを用いたオドメトリ情報、ビジュアルオドメトリ情報、GPS情報、姿勢角情報の融合手法を、屋外移動ロボットに実装した。また、屋外フィールドにおいて、この手法を利用した位置推定実験を行った。本稿では、これらの実装ならびに実験結果について報告する。

2 パーティクルフィルタによる自己位置推定

2.1 パーティクルフィルタ

パーティクルフィルタとは、推定対象（本研究では、移動ロボットの位置・姿勢）の状態遷移確率と観測値（本研究では、GPSによる観測情報を想定）が与えられたとき、対象物の状態推定に用いられるフィルタである。

Fig.1に、パーティクルフィルタを用いた状態推定の基本アルゴリズムを示す。図中の尤度とは、各々のパーティクルの正しさに関する確率変数である。本研究では、移動ロボットの位置・姿勢推定にノイズを加えて状態の予測（遷移モデルに従って遷移）を行い、GPSの観測によって、尤度計算から得た正しいと思われるパーティクルを取捨選択（リサンプリング）していくという手順を、観測値

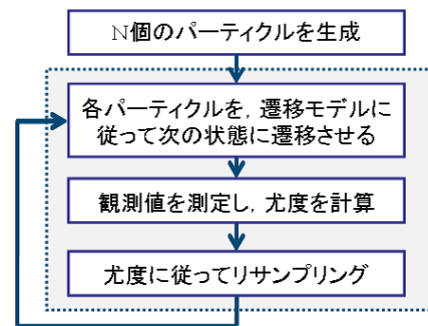


Fig. 1: particle filter

が得られる毎に繰り返す。これにより、GPSの利用の信号が得られない区間においても、ロボットの位置推定を逐次行うことが可能となる。

2.2 オドメトリ手法について

本研究では、パーティクルフィルタにおいて、推定対象である移動ロボットの位置・姿勢推定にノイズを加えて、逐次ロボットの状態予測を行う際に、車輪オドメトリとビジュアルオドメトリの2種類を実装し、その比較を行うこととした。

車輪オドメトリとは、左右の車輪回転速度の平均からロボットの並進量、左右の車輪回転速度の差分からロボットの旋回角速度を推定し、これらを積分してロボットの自己位置・姿勢を求めるものである。しかしながら、角度推定が路面と車輪のスリップに大きく影響される上述の方法を利用することは、望ましくない。そこで、本研究では、並進量については、車輪回転の計数を利用し、ロボットの向きについては、シリコンジャイロセンサを利用するジャイロベース車輪オドメトリを利用することとした。

一方、ロボットの並進速度を獲得する方法として、ビジュアルオドメトリ手法についても、本研究の対象とすることとした。我々が採用した方法は、高速に画像情報取得可能なカメラを、地面に向けて光軸が地面と垂直となるように設置し、取得した画像を処理することで、画像中の特徴点の移動速度を計算し、ロボットの並進速度を得るのである。なお、ロボットの旋回速度についても、理論上

は、ビジュアルオドメトリから計算が可能である。しかし、精度や動作速度を考慮すると、ここでの巡回速度取得については、ジャイロセンサを利用する方が現実的である。そこで、本研究では、この巡回速度の取得についても、シリコンジャイロセンサを利用することとした。

2.3 三次元オドメトリ

前節のオドメトリでは、二次元平面における二種類のオドメトリについて説明したが、屋外走行を行うロボットは、その位置・姿勢を三次元的に変化させる必要がある。そこで、本研究では、前節の手法により並進速度を求め、これをロボットの姿勢方向に積分することで、三次元的に自己位置・姿勢を推定する三次元オドメトリを実装した。

サンプリング時刻 $t = t_n$ における機体座標系を $x_n - y_n - z_n$ 、姿勢クォータニオンを 0q_n とする。次のサンプリング時刻 $t = t_{n+1}$ までに、ジャイロセンサにより、 x_n 軸周り (roll 角) に $\Delta\theta_{x,n}$ 、 y_n 軸周り (pitch 角) に $\Delta\theta_{y,n}$ 、 z_n 軸周り (yaw 角) に $\Delta\theta_{z,n}$ なる微小回転が測定されたとする。このとき、これらの回転を合成した姿勢変動クォータニオン ${}^nq_{n+1}$ は、以下の様に計算できる。

$${}^nq_{n+1} = \begin{bmatrix} \cos \frac{\Delta\theta_{x,n}}{2} \cdot \cos \frac{\Delta\theta_{y,n}}{2} \cdot \cos \frac{\Delta\theta_{z,n}}{2} \\ \sin \frac{\Delta\theta_{x,n}}{2} \cdot \cos \frac{\Delta\theta_{y,n}}{2} \cdot \cos \frac{\Delta\theta_{z,n}}{2} \\ \cos \frac{\Delta\theta_{x,n}}{2} \cdot \sin \frac{\Delta\theta_{y,n}}{2} \cdot \cos \frac{\Delta\theta_{z,n}}{2} \\ \cos \frac{\Delta\theta_{x,n}}{2} \cdot \cos \frac{\Delta\theta_{y,n}}{2} \cdot \sin \frac{\Delta\theta_{z,n}}{2} \end{bmatrix} \quad (1)$$

また、時刻 $t = t_{n+1}$ における姿勢クォータニオンは、上記姿勢変動クォータニオン ${}^nq_{n+1}$ と、前サンプリング時刻 $t = t_n$ における姿勢クォータニオン 0q_n の積により、

$${}^0q_{n+1} = {}^0q_n \times {}^nq_{n+1} \quad (2)$$

と計算できる。

また、時刻 $t = t_n$ から $t = t_{n+1}$ までの間に、ロボットが ΔX_n 進んだと計測されたとする、絶対座標系から見たロボットの位置変動ベクトル ${}^np_{n+1}$ は、

$$\begin{bmatrix} 0 \\ {}^np_{n+1} \end{bmatrix} = {}^0q_{n+1} \times \begin{bmatrix} 0 \\ \Delta X_n \\ 0 \\ 0 \end{bmatrix} \times {}^{n+1}q_0 \quad (3)$$

と与えられる。

以上により、絶対座標系から見たロボットの位置ベクトル ${}^0p_{n+1}$ は、

$${}^0p_{n+1} = {}^0p_n + {}^np_{n+1} \quad (4)$$

と計算できる。

2.3.1 遷移モデル

パーティクルフィルタでは、センサ情報に誤差が含まれているものとして誤差のモデルを立て、次の状態予測に際して、そのモデルを基にした誤差を加えたものを扱う。本研究における遷移モデルは、前項で述べた三次元オドメトリに誤差を付与したものである。そこで、ジャイロセンサ、車輪オドメトリ及びビジュアルオドメトリの誤

差を、それぞれ $\sigma_{\theta_x}, \sigma_{\theta_y}, \sigma_{\theta_z}, \sigma_{X_n}$ と仮定し、式 (1)、式 (3) をそれぞれ以下の様に変更する。

$${}^nq_{n+1} = \begin{bmatrix} \cos \frac{\Delta\theta_x + \sigma_{\theta_x}}{2} \cos \frac{\Delta\theta_y + \sigma_{\theta_y}}{2} \cos \frac{\Delta\theta_z + \sigma_{\theta_z}}{2} \\ \sin \frac{\Delta\theta_x + \sigma_{\theta_x}}{2} \cos \frac{\Delta\theta_y + \sigma_{\theta_y}}{2} \cos \frac{\Delta\theta_z + \sigma_{\theta_z}}{2} \\ \cos \frac{\Delta\theta_x + \sigma_{\theta_x}}{2} \sin \frac{\Delta\theta_y + \sigma_{\theta_y}}{2} \cos \frac{\Delta\theta_z + \sigma_{\theta_z}}{2} \\ \cos \frac{\Delta\theta_x + \sigma_{\theta_x}}{2} \cos \frac{\Delta\theta_y + \sigma_{\theta_y}}{2} \sin \frac{\Delta\theta_z + \sigma_{\theta_z}}{2} \end{bmatrix} \quad (5)$$

$$\begin{bmatrix} 0 \\ {}^np_{n+1} \end{bmatrix} = {}^0q_{n+1} \times \begin{bmatrix} 0 \\ \Delta X_n + \sigma_{X_n} \\ 0 \\ 0 \end{bmatrix} \times {}^{n+1}q_0 \quad (6)$$

2.3.2 尤度関数

本研究では、 xy 平面に投影した各パーティクルとGPS観測地間距離が正規分布に従っていると仮定し、尤度 w_t を計算する際に用いる尤度関数を以下に示す式のように設定した。

$$w_t^{[i]} = \frac{1}{\sqrt{2\pi}\sigma_{GPS}} \exp\left(-\frac{(d_t^{[i]})^2}{2\sigma_{GPS}^2}\right) \quad (7)$$

ここで $d_t^{[i]}$ は時刻 t における i 番目のパーティクルとGPS観測値間距離、 σ_{GPS} は正規分布に則った尤度関数の広がりを決める値である。

2.3.3 リサンプリング

各パーティクルは、遷移モデルに従って次の状態へと遷移される。また、GPSによる観測値との2点間距離から、尤度関数に基づき、各パーティクルの尤度が計算される。ここで、尤度が低いパーティクルは、推定位置として間違えたものである可能性が高いため、そのパーティクルを消去し、消去した数だけ、より尤度の高いパーティクルから選りばいをおこなう。これをリサンプリングという。今回はリサンプリング法として、計算効率が良い等間隔リサンプリング[3]を用いた。

3 フィールド実験

本研究では、前章で紹介したパーティクルフィルタによる自己位置推定手法の有用性、特に、通常のジャイロベースオドメトリとビジュアルオドメトリの性能比較を行うために、フィールド実験を行った。本実験の目的は、前章で述べた三次元オドメトリでの二種類の手法比較を行う予定であったが、現状では、その解析が完了していない。そこで、本稿では、ロール角、ピッチ角ならびに、Z軸方向に関するパラメータを無視した二次元でのオドメトリ手法の比較について、以下に述べる。

3.1 対象とする実験環境

実験フィールドは、東北大学の裏手にある、旧ゴルフ場の不整地走行フィールドで行った (Fig.2)。実験の許可を得たフィールドは、全長が約300[m] × 150[m]で、実験を実施した8月上旬の路面は、丈が1m以上の草木が所々に生い茂った環境である。また、走行経路内には、トータルで数メートルの高低差が存在する。



Fig. 2: The target environment

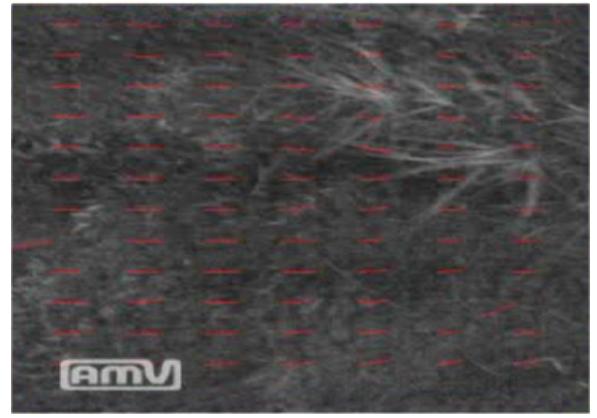


Fig. 4: A snapshot of visual odometry

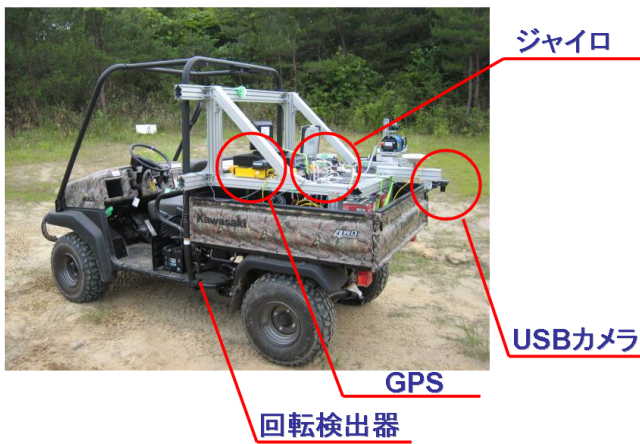


Fig. 3: The target mobile robot

3.2 対象とする屋外移動ロボット

本実験で対象とする屋外移動ロボットは、市販の二人乗り用オフロードバギーの後部に、各種センサ機器やカメラなどを取り付けたものである。このロボットの概要をFig.3に示す。このロボットに搭載する機器は、以下の通りである。

- 小野測器社製の光学回転検出器 LG-916 (車輪オドメトリ用)
- アルゴ社製USBカメラ Lu075 (ビジュアルオドメトリ用)
- シリコンセンシングシステムズ社製 1軸 Sシリコンジャイロ CRS03 3台
- クロスボー 3軸加速度計 CXL04GP3 1台
- Trimble社製 GPS受信機 MS750

ここで、車輪オドメトリ用の回転検出器は、 $9.89[\text{pulse/round}]$ であり、車輪直径が $0.58[\text{m}]$ であるため、車輪オドメトリのレゾリューション $t[\text{m/pulse}]$ は、

$$t = \frac{0.58 \times \pi}{9.89} = 0.184[\text{m/pulse}]$$

となる。なお、ビジュアルオドメトリを行うためのフロー計算を行う計算機は、高速に大きな処理を行う必要があるため、車輪オドメトリならびに、パーティクルフィルタリングを行う計算機とは独立とした。

3.3 ビジュアルオドメトリの実装

ビジュアルオドメトリ用カメラLu075は、 $640 \times 480[\text{pixel}]$ の解像度を $60[\text{フレーム/秒}]$ で獲得できる白黒USBカメラである。このカメラの取り付け位置は、移動ロボット中心軸上の、地面からの高さ約 $120[\text{cm}]$ とした。この時、画像1ピクセルの幅を、路面における長さに変換する係数 t は

$$t = 3.3[\text{mm/pixel}]$$

となる。フロー画像を取得するソフトウェアには、OpenCV[4]の画像処理ライブラリのテンプレートマッチングを使用した。テンプレートの大きさは、 $11 \times 11[\text{pixel}]$ 、探索範囲は、 $40 \times 30[\text{pixel}]$ と設定した。このテンプレートマッチング処理を画像中の84カ所で行い、テンプレートの輝度値の偏差の大きなものに対して、テンプレートの移動平均を計算することで、画像中のフローを求めることとした。Fig.4は、ある路面における取得画像に、フローの処理結果を赤色の線分でスーパーインポーズしたものである。

3.4 パーティクルフィルタの実装

パーティクルフィルタについては、各種パラメータをTable1に示す通り設定した。これらは、各センサの計測精度に加えて、車輪のスリップによる並進量の計測誤差、温度変動に伴うセンサ情報の変動を加味して設定した。GPSの測定については、観測値が真値より $2[\text{m}]$ 以内に $66.7 [\%]$ 含まれるものと仮定し、式(7)の正規分布に則った尤度関数の広がりを決めた。なお、各パーティクルとGPS 観測値間の距離については、 $4[\text{m}]$ を閾値と設定した。

Table 1: error configuration

角速度計測誤差 σ_{θ_z}	$\sigma = 0.7[\text{deg/sec}]$ なる正規乱数
並進量計測誤差 σ_{X_n}	併進量の $10[\%]$ を σ とする正規乱数
尤度関数誤差 σ_{GPS}	$\sigma = 2[\text{m}]$ なる正規分布
サンプリング周期	$100[\text{msec}]$
パーティクル数	1000

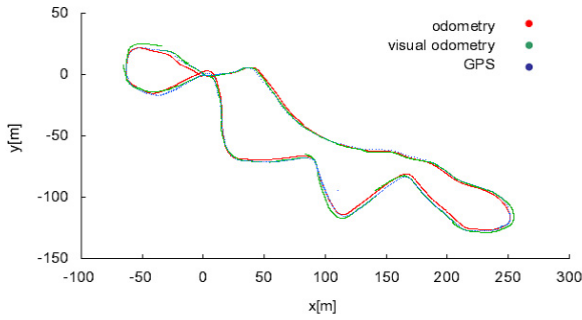


Fig. 5: An experimental result

3.5 実験方法と結果

上節で紹介した屋外移動ロボットを、実験環境において時速約7[km]で走行し、回転検出器により車輪回転数を、USBカメラにより路面フローを、ジャイロセンサによりロボットの方角角を、GPS受信機によりGPSのデータを取得した。取得データを元に前節で説明したアルゴリズムでパーティクルフィルタリングを行い、オドメトリ及びビジュアルオドメトリによる推定結果と、GPSによる推定結果を融合させた。

実験結果をFig.5に示す。この図における結果の表示は、全パーティクルの平均値とした。このグラフより、車輪オドメトリとビジュアルオドメトリによる自己位置推定の結果は、共に、全体的にGPSの結果に比較的良く一致していることがわかる。ただし、これは、共にGPSのデータを融合した結果を表示したものであるため、当然の結果とも言える。このグラフを詳細に検討すると、ビジュアルオドメトリの結果は、車輪オドメトリと比較して、GPSで得た結果から若干離れている様子が見て取れる。これは、Fig.2にあるように、本実験の環境は、草木が生い茂った環境であるため、カメラが取得した路面画像の高さと、想定した路面画像の高さにずれが生じたためであると考えられる。本実験で対象とした草地のような、地面の高さが不均一な環境では、距離センサなどを利用し、その地点における高さ情報を逐次獲得する必要があると考えられる。一方、車輪オドメトリに関しては、搭載したエンコーダの性能が低いにも関わらず、車輪の大きなスリップや横滑りがほとんど無かったためか、高い精度で自己位置推定ができた。

次に、途中でGPSが一部観測不可になったとした場合を想定する。本研究で採用する複数センサの融合による位置推定手法は、GPSが継続して必ず計測できる場合には、それほど有用ではない。しかしながら、何らかの理由により、GPS信号が途絶えた、または大きく変動してしまった場合に、特に有用となる。そこで、本実験において、走行途中からGPSが観測されなかった場合を仮定し、車輪オドメトリのパーティクルの推移の様子を観察した。その結果をFig.6に示す。

この図において、GPSは、ロボットの位置が $(x, y) = (200, -120)$ 付近から観測不可能になったと仮定している。その位置から、パーティクルは発散し始めるが、これは、GPS観測値による尤度計算ならびに、リサンプリングが行われない

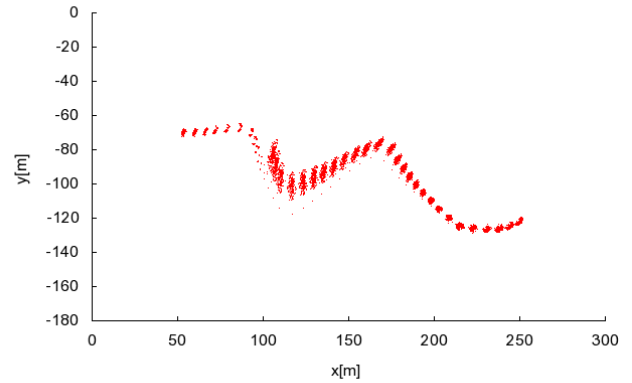


Fig. 6: process of particle

ため、遷移モデルに従って、車輪オドメトリのパーティクルが拡散するためである。なお、 $(x, y) = (100, -70)$ 付近より、再びGPSが観測可能となると仮定しているため、この点においてGPS観測値による尤度計算、リサンプリングが行われることで、拡散したパーティクルが収束していることがわかる。

4 結言

本稿では、屋外移動ロボットが作業する際に重要な要素となる自己位置推定について、車輪オドメトリならびに、ビジュアルオドメトリとGPS観測値をパーティクルフィルタにより融合させる手法について述べ、屋外実験フィールドにおいて、走行実験を行った。この実験から、上記手法は、共に屋外移動ロボットの自己位置推定において有用であることが示された。

今後の課題としては、今回実装が間に合わなかった、3次元オドメトリに関する比較検討を行うことが、直近の課題と言える。また、各手法のさらなる精度向上のため、遷移モデルにおける誤差の与え方の検討や、ビジュアルオドメトリにおけるカメラの地面からの高さの逐次獲得などが、今後の課題として挙げられる。

参考文献

- [1] D.Helmick, Yang Cheng, Stergios, and I.Roumeliotis. Path following using visual odometry for a mars rover in high-slip environments. In *Proc. of the 2004 IEEE Aerospace Conference*, pp. 772-789, 2004.
- [2] David Nister, Oleg Naroditsky, and James Bergen. Visual odometry for ground vehicle applications. *Journal of Field Robotics*, Vol. 23, No. 1, pp. 3-20, 2006.
- [3] Sebastian Thrun, Wolfram Burgard, and Dieter Fox. *Probabilistic Robotics*. The MIT Press, 2005.
- [4] 奈良先端科学技術大学院大学OpenCVプログラミングブック制作チーム. OpenCV プログラミングブック. 毎日コミュニケーションズ, 2007.