

歩行者自立航法測位の誤差を地図情報を元に位置を修正する方法

田中風助^{1,a)} 串田 高幸¹

概要：GPS の届かない屋内の位置測定の方法の一つに方向者自立航法測位（Pedestrian Dead Reckoning:PDR）というものがあるが、相対位置で位置測定をするため誤差が蓄積するという課題がある。これを解決するために、マップマッチングを用いてルート外に位置が検知されてしまっている時が誤差が発生していると仮定し、歩行ルートに修正するようなソフトウェアを作成する。

1. はじめに

背景

スマートフォンや携帯電話などのモバイルデバイスには、GPS 機能が標準搭載されるようになり、屋外での位置の取得（測位）は一般的なものとなった [1]。しかし、GPS 機能による測位は衛星が使用され、衛星からの信号を地上で受信し位置を取得するが、屋内環境ではその信号が届きにくいいため GPS で正確な位置の取得ができないという課題がある。

屋内環境でも測位可能なものとして Wi-Fi や Bluetooth を使う技術 [2][3] がある。しかしこれらはアンカーの設置や、複雑な屋内環境下においてはワイヤレス信号に障害が起こってしまうこともある。これに対し、歩行者が所持したセンサーから相対的に移動距離を取得する歩行者航法自立測位（Pedestrian Dead Reckoning:以下 PDR）と呼ばれる技術がある [4]。この技術は加速度センサー・ジャイロセンサー・磁気センサーを搭載したデバイスを歩行者が持つて移動することで、歩行者の相対的な移動距離を測り、現在地を求めることができる。またインフラストラクチャーに依存しないという点や、移動距離を求めるために必要なセンサーはデバイスにすべて搭載されているため、新たに測定機器を設置する必要がなく比較的低コストで測位を行うことができる [5]。また、PDR に必要なセンサーはスマートフォンにすべて内蔵されている。スマートフォンでの計測が容易なものとなれば、必要なセンサーを用意することなく計測が行える。そのため、スマートフォンを使つての計測も注目されている [6]。

課題

PDR では、加速度センサー・ジャイロセンサー・磁気センサーから取得した情報を元に移動距離を求める。GPS などのように絶対位置で位置情報を測位するのではなく、スタート地点からの相対位置で測位することになるため、移動する距離が長くなるほど誤差が蓄積され正確な位置取得は困難になる。そのため、この蓄積する誤差をなくす提案が必要になる。誤差が発生する原因は値を取得するセンサーによる誤差や歩行者の歩幅によるずれなどがある。

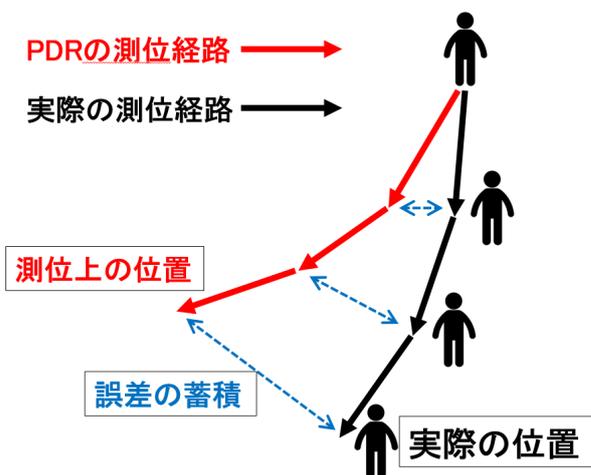


図 1 誤差発生時の図

図 1 は誤差が発生するメカニズムを表したものである。このように、距離が長くなるにつれてずれの幅が大きくなっていくことが分かる。このずれが修正されなければ、正確な位置が取得できず誤った情報が伝えられることになってしまう

¹ 東京工科大学コンピュータサイエンス学部
〒192-0982 東京都八王子市片倉町 1404-1

^{a)} C0118180

各章の概要

第2章では関連研究として誤差補正に利用される技術や歩行者ではなく自動車に利用する方法について述べる。第3章では提案するシステムについて述べる。第4章では提案したシステムの実装や実験環境について述べる。第5章では実験の評価と分析について述べ、第6章では提案したシステムの議論を述べる。第7章で研究のまとめを行う。

2. 関連研究

PDRの誤差補正には様々な方法がある。その一つに音波測位を用いたものがある[7]。これは音波測定器を通路に配置することで、IoTデバイスが測定機に近づいたときに、音波の反射で距離を測り誤差を修正するという方法である。この方法では、ルートに音波測定器の設置が必要になり、コストがかかるという課題がある。

また、グローバルナビゲーション衛星システム(GNSS)を利用した、推測航法(DR)が統合されるカーナビゲーションシステムなども関連研究としてあげられる[8]。これは、歩行者の位置測位に使われるのではなく、自動車の位置修正に利用される。方法としてはエンコーダ、地磁気センサー、電子コンパスなどのセンサーを使用して、移動物体の位置データと経路データを取得、それを衛星を使用して位置を決定するGNSSで修正を行うというものである。この方法は高精度で位置の情報が得ることが可能になる。しかし、衛星を使用することになるため、屋内の測位には利用しづらいという課題がある。

さらに、移動ロボティクスの分野では3軸磁気センサーを利用して環境中の磁場をスキャンし、データベースを構築することにより、位置追跡を行っている。環境中の磁場の強度とその位置との関係を考慮に入れることで、移動ロボットは、データベースに基づいて軌道追跡を実行することによって追跡の精度を高めている[9]。これらの研究はロボットの追跡を目的にしているため、歩行者の追跡には応用しにくい。

統一された方位誤差モデルを使用したPDRのアルゴリズムで誤差を減らす方法[10]もある。これは、非線形モデルから推定されたステップ長と、拡張カルマンフィルター(EKF)によって提案された統合モデルによって、方位エラーモデルを算出する。これを2軸コンパスのキャリブレーションに適用することで誤差をなくすというものである。この方法はほかの測位システムとの組み合わせることで誤差を補正するのではなく、あらかじめモデル化された誤差を位置修正に利用しているため追加でアンカーを設置する必要がないなどの利点がある。また、スマートフォンを利用したPDRの測位方法も存在する[11][12]。今の世の中では、スマートフォンは人々に欠かせないものになっており、その使用率は非常に高い。そのスマートフォンには、Wi-Fi、GPS、マイク、磁力計、慣性センサー(ジャイロ

スコープ、加速度計など)などのさまざまなセンサーが組み込まれている。これらの内臓センサーをそのまま利用して計測を行う。この研究では、センサーデバイスをスマートフォンで想定している。そのためこれまでの技術にはなかった、ポケットに入れている場合のデバイスの姿勢推定や、歩くことを想定した太ももの角度と加速度を融合させた検出方法が必要となる。しかしこれらの研究は、誤差が発生してしまうときの根本的な解決にはなっていない。方位誤差モデルに当てはまらない誤差の発生や、スマートフォンでのPDRの誤差の蓄積を解決することが課題がある。

3. 提案

本研究では、マップマッチングを用いて位置情報を修正する方法を提案する。マップマッチングとは、測位技術(GPSまたはDRと統合されたGPSなど)から生成された入力を使用し、これを高解像度の空間道路ネットワークマップからのデータで補足して、精度の高い測位出力を提供する技術である[13]。図2は本研究で提案するマップマッチングの方法の概要図である。

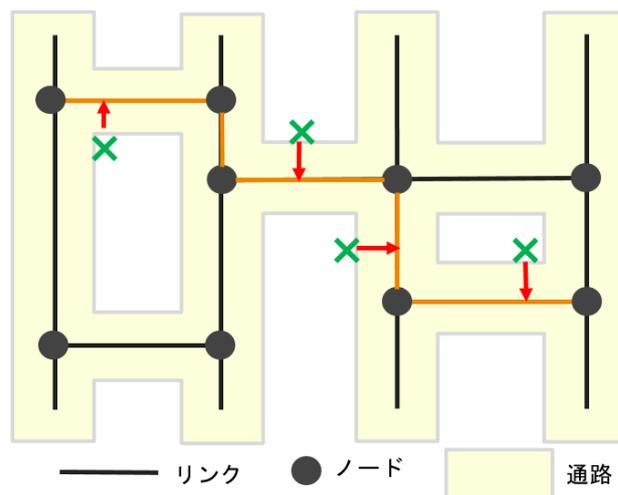


図2 マップマッチングの概要図

マップマッチングのアルゴリズムはいくつか種類がある。測位点を一番近いリンクにマッチングさせる方法や、候補となるノードを探索し候補ノードから候補経路を作成する方法などもある[14]。提案では、測位点とネットワーク上のすべてのノードとの距離を計算し、一定の時間ごとに測位点を最も距離の小さかったリンクにマッチングするというものである。これを実装を行う屋内の地図データと組み合わせ、歩行ルートとノードを一致させる。この歩行ルートと測位地の最短距離を計算し、歩行ルートに位置を強制的におくことで位置を修正できると考える

図3は位置修正の流れを示したものである。9軸センサーから習得した加速度・ジャイロ・磁気の3つのデータをサーバー側に送信し、そこで位置情報計算の処理を行う。

そのあとに独自のソフトウェアであるアップマッチングによる修正のソフトウェアを組み込むことで位置の修正を行う。[15].

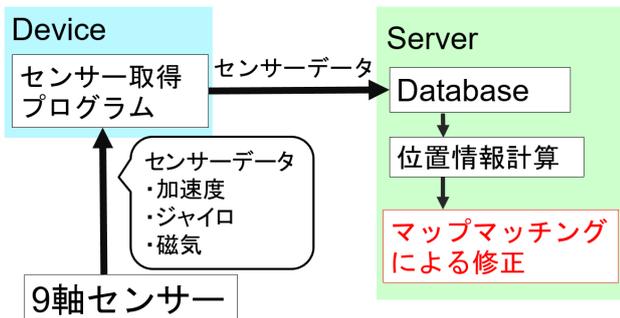


図 3 アーキテクチャ図

4. 実装と実験環境

4.1 実装

図 4 にソフトウェア構成図を示す。

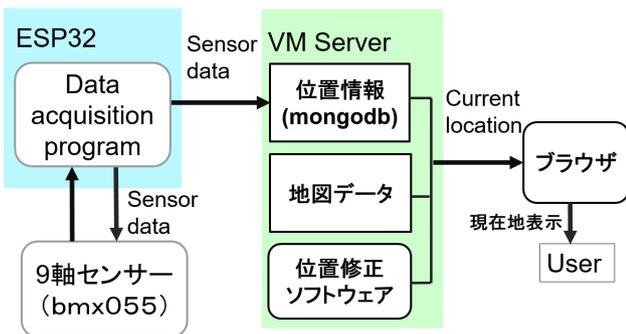


図 4 ソフトウェア構成図

実装は以下の手順で行う。

- (1) 9 軸センサーからセンサーデータを取得
- (2) 取得したセンサーデータをサーバーに送信
- (3) 受信したデータをデータベースに格納
- (4) 位置修正ソフトウェアを組み込み地図データと合わせる
- (5) 位置情報をブラウザに表示

センサーは BMX055 を使用する。ESP32 でセンサーから値を取得し、VM サーバーにその値を送り monngoDB に格納する。位置情報の計算はすべてサーバー内で行い、地図データやマップマッチングを行うソフトウェアと組み合わせ、ブラウザに現在地を表示する。地図データの作成はメジャーで実際に計測して行う。

4.2 実験環境

ハードウェア

- BMX055

ソフトウェア

- ESP32:micropython
- Management Server:Apache
- Database:MongoDB

実験は東京工科大学研 A910 で行う。図 5 は部屋の大きな間取りである。机は障害物として通れない道という判定にし、歩行ルート外にする。スタート地点をある一定の場所に決め、その位置から研究室内にある机の間を歩いて歩行する。地面は平面であり坂などの傾斜や凸凹な道はない。

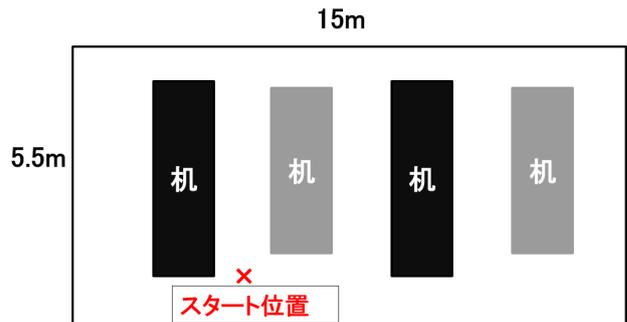


図 5 実験場所の間取り

5. 評価と分析

図 6 に実装図を示す。

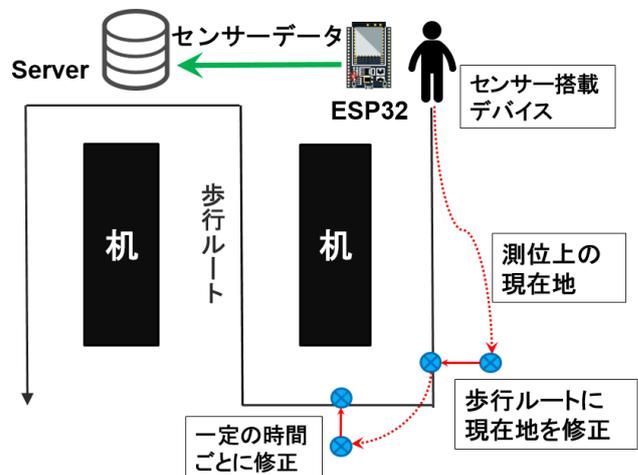


図 6 実装図

本実験では、人がセンサーを搭載したデバイスを所持し、決められた歩行ルートを複数回歩く。マップマッチングによる位置の修正のソフトウェアを組み合わせた場合と組み合わせない場合でそれぞれ平均値をとって比較を行い、どのくらい位置が修正されているかで評価を行う。最終的な位置が正しいかどうかの判定は、スタート地点からの X 軸・Y 軸方向にどのくらい移動しているかを実際にメジャーを使って図ることで決定する。今回は実際に移動す

る距離を X 軸に 4.20 m, Y 軸に 5.60 m と設定した。

図 7 は 10 回の計測を行った結果とその平均値である。これを見ると値が大きい時と小さい時の差が明確になってしまっている。これでは正確な移動距離が計測できていないことが分かる。

	X軸 (m)	Y軸 (m)
1回目	0.05135112	0.02037239
2回目	0.02557533	0.02471905
3回目	0.9663934	0.5028239
4回目	0.9010016	13.02286
5回目	13.73651	15.19266
6回目	1.145587	0.4947798
7回目	1.429175	10.58935
8回目	18.25876	5.240351
9回目	8.028478	15.15375
10回目	7.189219	31.4178
平均値	5.173205045	9.165946614

図 7 結果

6. 議論

本実験では移動距離の表示を X 軸・Y 軸の方向にスタート位置からどのくらい動いたかで位置を特定している。今回は、9 軸センサーを使って移動距離を計算して、相対的な位置情報を取得した。計測結果は大きな誤差が生じ、正確な位置測定ではなかった。その原因としてあげられるのは、位置情報を計算する際の計算式が簡易的なものであったことである。本実験では物理の等加速度運動の 3 式を応用し、一定の時間で求められた移動距離を足していくようにした。その時に、計測した加速度の値のフィルタリングが足りていなかったと考える。加速度センサーで求められる加速度は、物理の計算で使われる加速度とは異なったものとなっており、その値のずれが位置情報計算の際の誤差を大きくしている。さらに、歩行者が IoT デバイス所持して歩行するときに、デバイスの向きが固定されていないため歩行者の特徴によって、向きや角度にずれが発生する。今回のソフトウェアではデバイスの姿勢推定が行われていない。これも誤差が発生する原因となっていると考える。そのため、誤差を減らすにはフィルタリングやデバイスの姿勢推定を定める計算式の追加が必要になる。

またセンサー自体に生じる誤差も要因の一つと考える。センサーが算出する値は必ずしも正確というわけではなく、センサーの種類によっても様々な誤差が発生する。 Jianwei

Zheng らの研究 [16] でおこなった慣性計測装置 (IMU) のパフォーマンス分析によると、本事件で使用した BMX055 は最も低いパフォーマンス結果となっていた。慣性計測装置 (IMU) には多くの種類があり [17], 用途によってプラス面やマイナス面がある。慣性計測装置 (IMU) の性能は、歩行者追跡装置にとってとても重要なものとなっているため、パフォーマンスの良いセンサーを利用することは、誤差を減らす方法の一つとしてあげられる。

ソフトウェアの構成を変更した点がある。当初は ESP32 で取得したセンサーデータをサーバーに送信し、位置情報の計算はサーバー側で処理するというものであった。しかしそれらを変更し、ESP32 側でセンサーデータの取得と同時に位置情報の計算が行われるようにした。理由としては、サーバー側での処理の手順が減らせるからである。サーバー側で位置測位を行う場合、MongoDB に格納したセンサーデータを取り出し、移動距離の計算プログラムが必要になってくるが、ESP32 側で位置測位を行う場合ではセンサーデータ取得・移動距離の計算・サーバーに送信の 3 つをまとめて動作させることができた。これと同様にして、位置情報修正のプログラムも ESP32 側で行い、そこで取得した位置情報をサーバー側に送信しようと考えている。ブラウザに表示するためのソフトウェアは作成済みであるため、位置修正のプログラムを作成すれば比較を行うことが可能となる。

だがこれらの方法は、簡易的に動作させることは可能になるが、互換性が無いという欠点がある。地図情報も同じソフトウェアに組み込んでしまうため、違う場所での位置測位などを行うときに同じ作業が必要になってしまう。そのため、実用性を考慮するときには実装環境で手順が増えたとしても、互換性のあるようにソフトウェアを構成するほうがよいと考える。

7. おわりに

今回の研究では PDR の誤差が発生するという課題を解決するためにマップマッチングによる歩行ルートの修正を提案した。9 軸センサーを使って移動距離を計算し、相対的な位置情報を取得した。しかし計測結果は大きな誤差が生じてしまい、正確な位置測定ではなかったため、正確な計算やセンサーデータのフィルタリングが必要となる。また、位置修正を行うプログラムが完成していないため、優先的に作成する必要がある。

参考文献

- [1] Brown, A. K. and Sturza, M. A.: GPS tracking system (1995).
- [2] Yang, C. and Shao, H.: WiFi-based indoor positioning, *IEEE Communications Magazine*, Vol. 53, pp. 150–157 (2015).
- [3] Heydon, R. and Hunn, N.: Bluetooth low energy (2012).

- [4] Jimenez, A. R., Seco, F., Prieto, C. and Guevara, J.: A comparison of pedestrian dead-reckoning algorithms using a low-cost MEMS IMU, *2009 IEEE International Symposium on Intelligent Signal Processing*, pp. 37–42 (2009).
- [5] Pratama, A. R., Hidayat, R. et al.: Smartphone-based pedestrian dead reckoning as an indoor positioning system, *2012 International Conference on System Engineering and Technology (ICSET)*, pp. 1–6 (2012).
- [6] Wang, B., Liu, X., Yu, B., Jia, R. and Gan, X.: Pedestrian dead reckoning based on motion mode recognition using a smartphone, *Sensors*, Vol. 18, No. 6, p. 1811 (2018).
- [7] Zhang, G. and Wang, B. B.: Music-Based Positioning Aided By Dead Reckoning (2015).
- [8] Koo, B. W., Lee, S. S., Gim, J. S. and Lee, D. K.: Car navigation system and method in which global navigation satellite system (gnss) and dead reckoning (dr) are merged (2015).
- [9] Rahok, S. A., Shikanai, Y. and Ozaki, K.: Navigation using an environmental magnetic field for outdoor autonomous mobile robots, *Advanced Robotics*, Vol. 25, No. 13-14, pp. 1751–1771 (2011).
- [10] W. Chen, R. Chen, J. W.: An effective Pedestrian Dead Reckoning algorithm using a unified heading error model (2010).
- [11] Zhao, H., Zhang, L., Qiu, S., Wang, Z., Yang, N. and Xu, J.: Pedestrian dead reckoning using pocket-worn smartphone, *IEEE Access*, Vol. 7, pp. 91063–91073 (2019).
- [12] Kang, W. and Han, Y.: SmartPDR: Smartphone-based pedestrian dead reckoning for indoor localization, *IEEE Sensors journal*, Vol. 15, No. 5, pp. 2906–2916 (2014).
- [13] Quddus, M. A., Ochieng, W. Y. and Noland, R. B.: Current map-matching algorithms for transport applications: State-of-the art and future research directions, *Transportation research part c: Emerging technologies*, Vol. 15, No. 5, pp. 312–328 (2007).
- [14] Streit, D. A., Gilmore, B. J., Lu, M. and Pusateri, M. A.: Map-matching navigation system (1998).
- [15] 羽藤英二・伊藤創太・伊藤篤志編: マップマッチングのアルゴリズム (2014).
- [16] Zheng, J., Qi, M., Xiang, K. and Pang, M.: IMU performance analysis for a pedestrian tracker, *International Conference on Intelligent Robotics and Applications*, pp. 494–504 (2017).
- [17] Zheng, J., Qi, M., Xiang, K. and Pang, M.: IMU performance analysis for a pedestrian tracker, *International Conference on Intelligent Robotics and Applications*, pp. 494–504 (2017).