様々な状況と環境に対応できる PDRベースの屋内位置推定ライブラリの基礎検討

外山 瑠起[1]　梶 克彦[†1]

概要：現代の社会において，屋内位置推定技術は重要な技術である．屋内の人の動きを把握してビル内のナビゲーションなどに活用するなど様々な活用方法が考えられる．しかし，屋内での位置推定は外部環境に大きく影響され，使用できるデータが異なるため，特定の屋内位置推定手法であらゆる場面に対応するのは困難である．多様な状況や環境で屋内位置推定をするには，個々の条件に適した位置推定手法の組み合わせや選択が重要である．例として歩行者の初期位置とスマートフォンから加速度，角速度, 設置されたBLE ビーコンからの電波強度が使用できる場合．PDR に加えて BLE の電波強度をして PDR による誤差を補正できる．このように状況に応じて適切な処理を行えば，より精度のよい屋内位置推定が可能になる．本研究の目的は様々な状況と環境に柔軟に対応できる，包括的な屋内位置推定ライブラリの開発である．

Consideration of a PDR-based indoor location estimation library for various situations and environments

TOYAMA RYUKI[1]　KAZI KATSUHIKO[†1]

1. はじめに

屋内位置置推定技術は，現代社会において重要な役割を果たしており様々な活用が期待できる．屋内位置推定技術が使用される一例として，ショッピングモール施設でのナビゲーションシステムが挙げられる [1]. このシステムでは顧客が店内で商品を探している際，その位置情報を元にしたナビゲーションシステムを通じて目的の商品が置かれている売り場まで案内するシステムである．

屋外における位置推定技術として GPS が広く利用されているが，屋内環境では建物の壁や天井が GPS 衛星からの電波を遮断してしまい，位置推定精度が大きく低下してしまう問題があり，別のアプローチが必要とされている．屋内位置推定手法として，PDR(Pedestrian Dead Reckoning) が挙げられる．PDR は主に，加速度計，ジャイロスコープ，磁気センサなどのセンサを利用して歩行者のステップ数，歩行速度，歩行方向を推定する．その情報を元に歩行者がどのくらいの距離をどの方向に移動したかを累積的に計算して基準となる位置からの相対的な位置を推定する手法である．他の例として，Wi-Fi の電波を使用した屋内位置推定手法がある．Wi-Fi を利用した屋内位置推定は，Wi-Fi アクセスポイントからの信号強度を利用して位置推定を行う．特定の地点でのフィンガープリントを予め取得しておきそれと比較して推定を行う手法や，3 つのアクセスポイントからの電波強度を利用して三角測量を行う手法がなどがある．

しかしこれらの手法は特定の状況や環境に特化したものであり，すべての屋内環境で同様の効果を発揮するわけではない．例として先ほどあげた Wi-Fi を利用した屋内位置推定の場合，地下施設や Wi-Fi アクセスポイントが設置が難しい場所では，信号が弱いため正確な位置推定が難しくなる．PDR の場合各種センサによる誤差が蓄積されるため，時間経過とともに誤差が大きくなってしまう問題がある．そのため長時間の屋内移動場合は精度が低下してしまう．

これらの問題を解決するためには状況や環境に応じて適切な位置推定手法の選択または組みわせる必要がある．本

[1] （社）情報処理学会
IPSJ
[†1] 現在，マルチメディア, 分散, 協調とモバイルシンポジウム
Presently with DICOMO2023

研究の目的は様々な環境に対応できるオフライン屋内位置推定ライブラリの検討である．ここでいうオフラインとは，あらかじめ取得したデータを元に位置推定を行うという意味である．先ほどあげた長時間の屋内移動の場合，PDR で推定した位置情報を Wi-Fi の電波強度を利用した位置推定でセンサの誤差を修正すれば，より正解な位置推定を行える．これらの補正を自由に組み合わせて使えるライブラリの検討を行い，様々な状況に対応できるオフライン屋内位置推定ライブラリを実現する．

2. 関連研究

屋内位置推定の手法において環境内に設置された機器を利用し推定する絶対位置測位手法がある．絶対位置測位は，屋内に設置された機器からの情報を元に位置を推定する手法である．例えば Bluetooth や Wi-Fi などの電波を利用した手法がある．屋内に設置した近接特化型の BLE ビーコン 3 つからの電波強度を利用して三角測量を行い位置推定を行う研究 [1] や GMM を使用して Wi-Fi の電波強度分布をモデル化し，それを元に位置推定を行う研究 [2] がある．これらの手法は設置された機器がない場合や設置が難しい場合は使用できない問題がある．

PDR と絶対位置測位を組み合わせて屋内位置推定を行う手法がある．PDR と Wi-Fi の受信強度を用いたプロキシミティベースの位置推定を行う研究 [3] や BLE ビーコンの受信信号強度の変異を利用した移動変異推定と PDR との併用による累積測位誤差の補正を行う研究 [4] がある．また PDR とマップマッチングを組み合わせて位置推定を行う研究 [5] や歩行時の磁気データのみを用いて位置推定を行う研究 [6] がある．これらの研究で示されているように PDR と絶対位置測位を組み合わせた手法は，お互いのデメリットを補えるため屋内位置推定をする上で有用な手法である．

本研究でも PDR による位置推定を行いその結果に対して，BLE ビーコンの電波強度を使用した補正やマップマッチングによる補正をなどが行えるライブラリを検討する．さらに初期位置や終了位置などの様々な状況の情報で補正できるようにより包括的かつ柔軟な屋内位置推定ライブラリの検討を行う．

3. PDR ベースの屋内位置推定ライブラリの検討

3.1 要求仕様

屋内位置推定は状況や環境によって推定に使用できる情報が異なるため，ライブラリを作る上でその具体的な例を考える必要がある．例えば，大学内や病院などの Wi-Fi のアクセスポイントが多く設置されている場所では Wi-Fi の電波強度を利用した位置推定が有効である．他の例として展示会場や大きなアトリウムなどの広い開放空間が考えられる．このような場所では Wi-Fi のアクセスポイントの配置が難しく，信号のカバレッジが不均一になりやすく Wi-Fi を利用した位置推定は難しい．このような場所の場合 BLE ビーコンを配置してその電波強度を利用した位置推定ができると考えられる．正しこのような空間で正確な位置推定を行うには,BLE ビーコンの多数配置する必要があり，労力や配置したとしてもメンテナンスコストがかかる可能性がある．このような場合は PDR を利用した位置推定が有効である．PDR は歩行者の歩行速度や歩行方向を推定して位置推定を行う手法である．この手法では歩行者がスマートフォンを持っているだけで良いため，先ほどあげたような場所でもコストをかけずに位置推定を行えると考えられる．しかし PDR はセンサのわずかな誤差が，時間経過とともに蓄積され位置推定の誤差が大きくなってしまう問題がある．そのため長時間の歩行の場合は位置推定の精度が低下してしまう．このような問題を解決する手法として [3] や [4] や方法が提案されている．

これらの手法のように基本的なベースを PDR で位置推定を行い，その結果を補正できるようなライブラリの検討を行う．ライブラリを作る上での想定環境として XDR Challenge Track5 の環境を元に考える．XDR Challenge は屋内位置推定の精度を競うコンテストである．このコンテストの Track5 では歩行者が高速道路のサービスエリアを歩行する．歩行者は腰にスマートフォンをつけた状態で LiDAR と呼ばれる光を使った距離測定技術を搭載したハンドヘルド LiDAR(以下,LiDAR) を持ち施設内を歩行する．参加者にはスマートフォンから得られた，加速度，角速度，磁気センサのデータと LiDAR から得られた正解の歩行軌跡の座標が提供される．施設内には BLE ビーコンが配置されており，歩行時の各ビーコンからの電波受信強度が提供される．他に与えられるデータとして，フロアマップ情報，フロアマップにおける各 BLE ビーコンの配置情報，歩行者の初期位置，終了位置が与えられる．本来であれば病院や学校などのより多くの施設で利用できるライブラリにする必要がある．しかしそのようなライブラリを構築するのは難しいためまず基本的な屋内位置推定環境が整っている XDR Challnge の環境でのライブラリの基礎検討と検証を行う．

関数に必要な引数の情報とその対応表を表 1 に示す．詳しい関数の説明や内部実装については次節で述べる．引数の情報は大きく分けてセンサ情報，環境情報，その他の 3 つに分けられる．センサ情報はスマートフォンから得られる加速度，角速度，磁気センサ,BLE ビーコンの電波情報が含まれる．環境情報はフロアマップ情報，フロアマップにおける各 BLE ビーコンの配置情報が含まれる．これらの環境情報は全てセンサデータが与えられる前に得られる情報である．その他はセンシング中，またはセンシング前に得られる情報であり，初期位置，終了位置などの情報が

該当する．

3.2 ライブラリ

以下節で説明する関数の引数に必要な情報は全て表 1 と紐づいている．まず基本的な PDR の処理を行う関数をリスト??に示す．この関数では加速度 (acc)，角速度 (gyro) を使用して位置推定を行う．加速度，角速度のデータフレームのカラム名とデータ型を表 2，表 3 に示す．初期座標は辞書型で表 4 に示す．戻り値は時間経過に伴う 2 次元座標のデータフレームと角度のデータフレームである．この 2 つの戻り値を返す理由として，次の関数に引き継ぐためである．

歩幅の推定を行っている研究は多くある．機会学習を用いた研究 [7], 多変量解析を用いた研究 [8]，超音波センサーガジェットを用いた研究 [9] などがある．本関数の内部処理では歩幅の値は固定値として扱っている．本来であれば歩幅は身長，性別，年齢などの複数の要素によって動的に変化するため固定値なのはありえず, 先ほど挙げた研究のように歩幅を推定する必要がある．しかし本ライブラリの目的は正確な歩幅を用いた PDR による位置推定ではない．PDR で推定した歩行軌跡を環境情報などを用いて補正を行い軌跡全体の最適化を行えるライブラリの検討である．そのため歩幅の推定は行わず固定値として扱う．また同様の理由で歩行タイミングの検出も正確には行わず，加速度の値が特定の閾値を超えた時に歩行タイミングとして扱っている．図 2 にリスト??を用いて PDR による位置推定を行った結果を示す．LiDAR で取得した座標をもとに出力された軌跡を図 3 に示す．これを本論では正解軌跡とする．図 2 と図 3 を比較すると PDR による軌跡は正解軌跡と比べて大きくずれているのがわかる．PDR 特有の解決すべきものとして軌跡そのものの形状を正解奇跡に近づける問題と絶対位置との関連付けの問題がある．これらの問題を環境情報などを用いて補正を行い正解軌跡に近づけていく．

```
1 Axis2D = Literal["x", "y"]
2 def estimate_trajectory(
3     acc_df: pd.DataFrame,
4     gyro_df: pd.DataFrame,
5     *,
6     ground_truth_first_point: dict[Axis2D,
          float] | None = None,
7 ) -> tuple[pd.DataFrame, pd.DataFrame]:
```

図 1: 基本 PDR

正解初期座標が予め判明している場合にリスト??に示される関数にその座標を与えたのが図 4 である．予め正解座標が判明している場合は PDR による軌跡の初期位置を補正できる．

図 4 の軌跡には PDR 特有のドリフト現象が見られる．

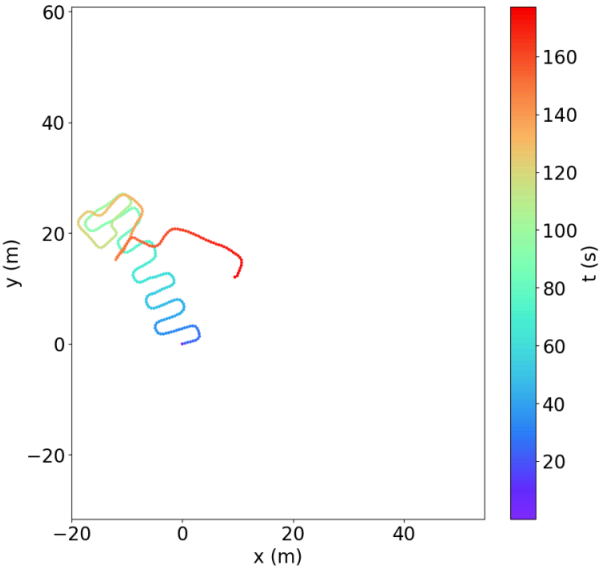


図 2: PDR による軌跡

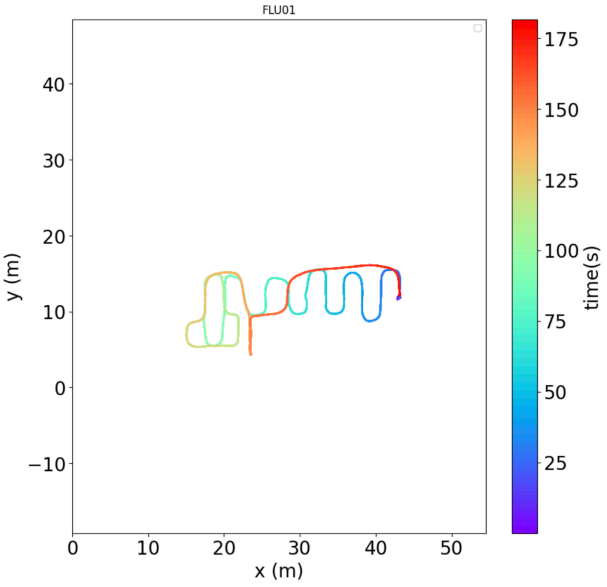


図 3: 正解軌跡

PDR では角速度から進行方向を求めてその方向を元に歩行軌跡を描く．そのため角速度センサーにわずかなでも誤差が含まれると時間経過とともにその誤差が大きくなり軌跡の形状が本来の軌跡から外れる．この問題を解決するために，角速度データに含まれる累積誤差を取り除く必要がある．

ドリフト補正のプロセスは，ドリフトを動的に計算し，それを角度データから差し引くことで補正を行う．具体的には、‘apply_remove_drift_to_angle‘関数を用いて，各時点での角度データからドリフトを除去する。この関数は，角度データフレーム，ドリフトの値，引数として取り、補正された角度データフレームを返す．このドリフト補正プロセスは，式 (1) で表される．$\theta'(t)$ は時間 t における補正後の角度，$\theta(t)$ は補正前の角度，d はドリフト率を意味する．

	関数名	センサ情報				環境情報				その他			
					BLE ビーコン		磁気	BLE ビーコン		正解初期		補正	
		加速度	角速度	角度	電波強度	フロアマップ	FP	基地局位置	FP	座標	方向	座標	方向
基本 PDR	estimate_trajectory	○	○							△			
角速度から角度推定	convert_to_angle_from_gyro		○										
ドリフト補正	remove_drift_in_angle	○		○						○		○	
初期方向補正 フロアマップ	rotate_trajectory_to_optimal_alignment	○		○		○				△			
初期方向補正 BLE		○	○		○			○					
マップマッチング補正		○	○			○							
安定歩行区間補正		○	○										
初期方向補正 BLE FP		○	○						○				

表 1: 関数に必要な情報とその対応表

注: ○ は必須引数,△ はオプショナル引数を示す

カラム名	単位	データ型
ts	s (seconds)	float
x	m/s^2	float
y	m/s^2	float
z	m/s^2	float

表 2: 加速度 DF

カラム名	単位	データ型
ts	s (秒)	float
x	rad/s (ラジアン/秒)	float
y	rad/s (ラジアン/秒)	float
z	rad/s (ラジアン/秒)	float

表 3: 角速度 DF

	データ型	説明
key	`str`	x または y
value	`float`	初期座標

表 4: 初期座標 DICT

カラム名	単位	データ型
ts	s (秒)	float
x	m(メートル)	float
y	m(メートル)	float

表 5: 座標 DF

カラム名	単位	データ型
ts	s (秒)	float
x	rad (ラジアン)	float
y	rad (ラジアン)	float
z	rad (ラジアン)	float

表 6: 角度 DF

この式は，時間経過に伴うドリフトの累積効果を補正するために使用される．

$$\theta'(t) = \theta(t) - (\mathrm{d} \times (t)) \tag{1}$$

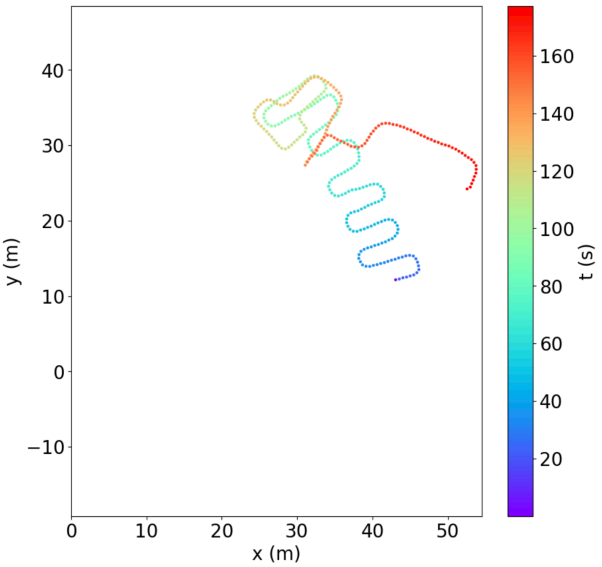


図 4: 正解初期座標が存在

```
def remove_drift_in_angle_df(
    acc_df: pd.DataFrame,
    angle_df: pd.DataFrame,
    ground_tooth_point_df: pd.DataFrame,
) -> tuple[pd.DataFrame, pd.DataFrame]:
```

図 5: ドリフト除去

```
def rotate_trajectory_to_optimal_alignment(
    acc_df: pd.DataFrame,
    angle_df: pd.DataFrame,
    map_dict: dict[str, np.ndarray],
    floor_name: str,
    dx: float,
    dy: float,
    *,
    ground_truth_first_point: dict[Axis2D,
        float] | None = None,
) -> tuple[pd.DataFrame, pd.DataFrame]:
```

図 6: 初期方向補正

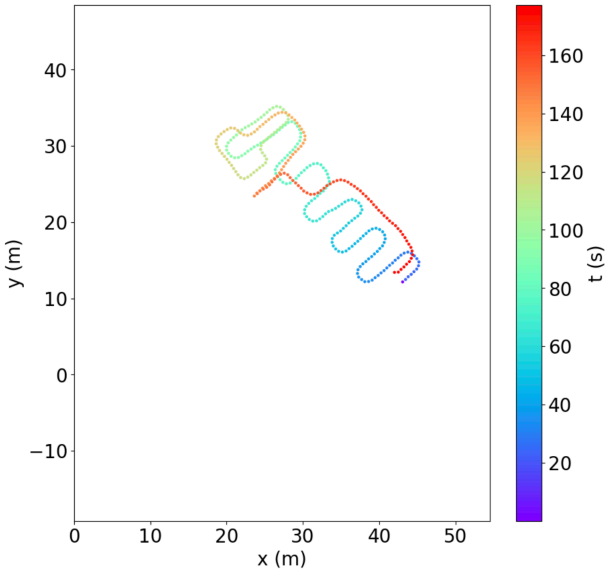
補正の効果を評価するために，ユークリッド距離を用いて，補正後の軌跡と基準軌跡との差異を計算する．式 (2) は，補正後の軌跡の座標 $(x_{\mathrm{g}}, y_{\mathrm{g}})$ と，基準軌跡の座標

	データ型	説明
key	`str`	floor の名前
value	`np.ndarray`	フロアマップの画像データ. 各フロアのブール値の NumPy 配列

表 7: フロマップ DICT

$(x_{\mathrm{ref}}, y_{\mathrm{ref}})$ との間のユークリッド距離 E を示している．この距離を用いてドリフト補正の効果を定量的に評価する．この式に基づきドリフト値に対してグリッドサーチを行い距離が最小になるドリフト値を探す．最小のドリフト値を角速度データから引きそれに基づいた歩行軌跡を返す．図**??**に示すように，ドリフト補正後の軌跡は，元の軌跡と比較して正解軌跡に近づいている．このアルゴリズムは開始地点と終了地点の距離が近い時に特に有効である．

$$\mathrm{E} = \sqrt{(x_{\mathrm{g}} - x_{\mathrm{ref}})^2 + (y_{\mathrm{g}} - y_{\mathrm{ref}})^2} \tag{2}$$



図**??**の軌跡の問題点として初期方向の誤差がある．初期方向が誤っていると，歩行者の実際の移動経路と大きく異なる軌跡が生成されてしまう．この問題を解決するために．角度データと環境マップ情報を組み合わせた初期方向の補正アルゴリズムを導入する．このプロセスは，_find_best_alignment_angle 関数によって実装される．この関数は，角度データフレーム，地上の真実データ，編集可能なマップ情報，フロア名，およびマップのグリッドサイズを引数として受け取る．まず，異なる回転角で軌跡の水平垂直方向のカウントを計算し，その後，各回転角度での軌跡が環境マップ上で通行可能かどうかを判断する．軌跡の水平垂直方向のカウントは，_calculate_horizontal_and_vertical_counts 関数によって計算される．この関数は，角度データフレームと回転角を引数として受け取り，回転後の角度に基づいて水平および垂直方向の軌跡カウントを計算する．軌跡における垂直成分と水平成分を可視化したものが図 7 である．これにより，回転角度によって軌跡の水平成分と垂直成分の割合の大きさがわかる．次に，_calculate_exist_counts 関数は，回転後の軌跡が環境マップ上で通行可能なポイントの数を計算します．この関数は，角度データフレーム，回転角の結果データフレーム，地上の真実データ，環境マップ情報，フロア名，およびマップのグリッドサイズを引数として受け取る．最適な方向補正角度は，最も多くの通行可能なポイントを持つ回転角として決定される．最終的に，_get_optimal_angle 関数は，回転後の軌跡が最も多くの通行可能なポイントを持つ回転角度を選択し，これが最適な方向補正角度として選ばれます．この角度を用いて，process_find_best_alignment_angle 関数は，補正された軌跡データを生成する．この方向補正アルゴリズムを適用することで，PDR に基づく屋内位置推定の精度はさらに向上します．図**??**は，方向補正後の軌跡を示しており，補正前の軌跡と比較して，実際の移動経路により密接に一致していることがわかる．

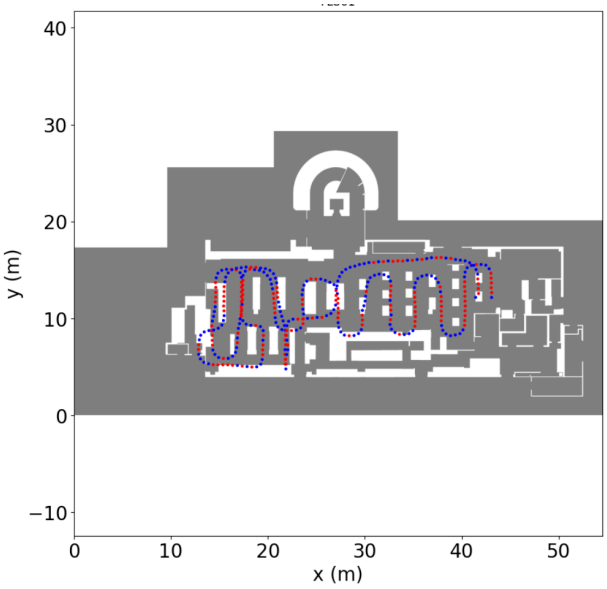


図 7: 垂直成分と水平成分の可視化

参考文献

[1] 酒井瑞樹, 森田裕之. Bluetooth を用いた屋内位置推定手法の提案. 経営情報学会 全国研究発表大会要旨集 2016 年秋季全国研究発表大会, pp. 53–56. 一般社団法人 経営情報学会, 2016.

[2] 浅見宗広, 福井隆司, 葛城友香, 大杉苑子ほか. Gmm を用いた wifi 位置推定手法の実用化に向けた課題解決. 研究報告高度交通システム (ITS), Vol. 2013, No. 2, pp. 1–8, 2013.

[3] 田巻櫻子, 田中敏幸. Wi-fi および端末センサ情報を用いた 3 次元屋内位置測位手法の検討. 国際 ICT 利用研究学会論文誌, Vol. 2, No. 1, pp. 24–30, 2018.

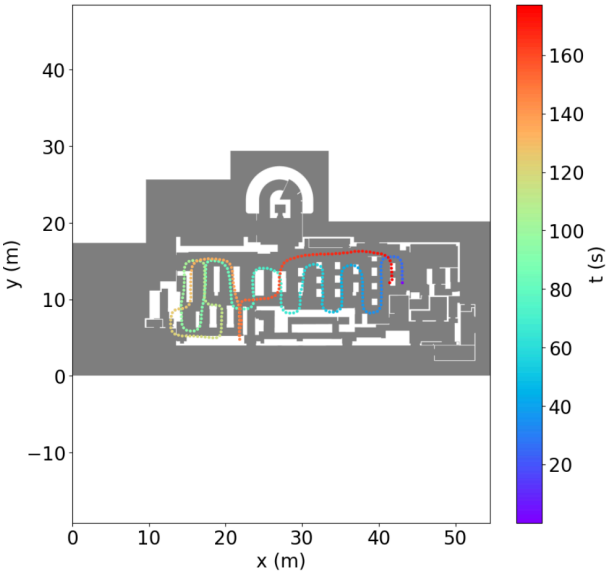


図 8: 初期方向の補正後の軌跡

[4] 工藤大希, 堀川三好, 古舘達也, 岡本東ほか. Ble 測位および pdr を用いたハイブリッド型屋内測位手法の提案. 第 79 回全国大会講演論文集, Vol. 2017, No. 1, pp. 325–326, 2017.

[5] 吉見駿, 村尾和哉, 望月祐洋, 西尾信彦ほか. マップマッチングを用いた pdr 軌跡補正. 研究報告ユビキタスコンピューティングシステム (UBI), Vol. 2014, No. 20, pp. 1–8, 2014.

[6] 村田雄哉, 梶克彦, 廣井慧, 河口信夫, 神山剛, 太田賢, 稲村浩ほか. 歩行時の磁気センシングデータを利用した屋内位置推定手法. 情報処理学会論文誌, Vol. 58, No. 1, pp. 57–67, 2017.

[7] 三宅孝幸ほか. デッドレコニング開始前のセンサデータを自動学習に用いた屋内歩幅推定手法の検討. 第 74 回全国大会講演論文集, Vol. 2012, No. 1, pp. 443–444, 2012.

[8] 根岸拓郎, 藤田悟ほか. 携帯端末のセンサ値を用いた多変量解析による歩幅推定. 第 77 回全国大会講演論文集, Vol. 2015, No. 1, pp. 299–300, 2015.

[9] 柏本幸俊, 荒川豊, 安本慶一ほか. デッドレコニングの高精度化に向けた超音波による歩幅推定法. 研究報告モバイルコンピューティングとユビキタス通信 (MBL), Vol. 2015, No. 56, pp. 1–6, 2015.