

# 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<sup>1</sup> 梶 克彦<sup>2</sup>

**概要：**現代社会において屋内位置推定技術は重要であり、屋内の人の動きを把握してビル内ナビゲーションに活用するなど様々な用途に利用されている。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。屋内位置推定の手法として PDR がある。PDR はスマートフォンなどから得られるセンサデータを元にある地点からの相対的な位置を推定する手法である。PDR はスマートフォンなどの機器さえあれば環境に左右されず一定の推定ができる。一方で PDR は時間の経過に応じて特有の誤差が蓄積する問題がある。そのため PDR の誤差を環境情報などを使用して補正するハイブリット手法が用いられる場合が多い。しかしハイブリット手法は特定の環境を想定したものが多く、複数の環境を想定したものは多くない。そこで本研究では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を行った。本ライブラリを用いて参加した PDR ベンチマーク委員会が主催する屋内位置推定の精度を競うコンテスト xDR Challenge 2023 において一定の結果を残した。また他の環境での適用を検討し複数の環境での適用可能性を示した。

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

TOYAMA RYUKI<sup>1</sup> KAZI KATSUHIKO<sup>2</sup>

### 1. はじめに

屋内位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されている。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内でのナビゲーションシステムが挙げられる。<sup>\*1</sup> このシステムでは顧客の位置情報を元にして、現在地周辺にある店舗のおすすめ情報や目的地までのナビゲーションを提供する。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する問題がある。そのため別のアプローチが必要とされている。

屋内位置推定の手法には絶対位置推定手法、相対位置推定手法がある。絶対位置推定手法は経度や緯度などの特定の

基準点を元に位置を推定する手法である。その代表例としては Wi-Fi、BLE ビーコン、地磁気などの情報を利用したものがある。Wi-Fi や BLE ビーコンなどの電波を利用した屋内位置推定は、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。予めアクセスポイントの基地局情報が判明している場合、3 つのアクセスポイントからの電波強度を利用して位置推定を行う手法がある。相対位置推定手法はある特定の基準点からの相対的な位置を推定する手法である。その代表例として PDR (Pedestrian Dead Reckoning) がある。PDR は歩行者が身につけたスマートフォンなどに搭載される加速度計、ジャイロ스코プ、などのセンサを利用して歩行者の歩幅、進行方向、ステップタイミングを推定する。その情報を元に歩行者の移動を累積的に計算し、基準位置からの相対的な位置を推定する手法である。

ハイブリット位置推定手法は PDR と絶対位置推定を組み合わせた手法である。絶対位置推定は特定の環境に依存しており、その環境がない場所では推定できない問題点がある。PDR による推定には初期位置と初期進行方向の情報が

<sup>1</sup> 愛知工業大学大学院 経営情報科学研究科

<sup>2</sup> 愛知工業大学 情報科学部

<sup>\*1</sup> <https://www.kkc.co.jp/service/blog/indoor-outdoor-positioning/achievement/article/9725/>

必ず必要な問題がある。また PDR ではセンサーのわずかな誤差が累積し続けるため、長時間に及ぶ歩行では軌跡の形状が大きく変化してしまう問題がある。ハイブリット手法では両方の手法を組み合わせることでこれらの問題の解決を行う。この手法は屋内位置推定の手法の中で有用な手法として様々な情報を用いた推定手法が研究がされている。

しかしこれらの研究は特定の条件下での PDR と組み合わせたものが多く、他の条件下では推定が難しい。例として Wi-Fi を利用した方法の場合、基地局の位置が事前に把握できているケースとできないケースが考えられる。また追加で補正に利用できる条件がある場合も考えられる。環境によって補正に使用できる情報は異なる。多くの環境を想定したような屋内位置推定手法は少ない。

本研究では様々な環境と状況に対応できる PDR ベースの屋内位置推定ライブラリの基礎検討を行う。本研究の概要を図 1 に示す。推定に使用できる情報をセンサ情報と環境情報に分け、これらの情報を元に PDR とその補正を行うライブラリを構築する。これらの関数はそれぞれの環境や条件に応じて適用、組み合わせる使えるような形を目指す。



図 1: 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概要

## 2. 関連研究

絶対位置推定に関する研究がある。これは特定の基準点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推定手法がある。これらの電波を利用した推定手法は Triangulation 方式、Finger Print 方式、Proximity 方式の 3 つに分類される。[1] Triangulation 方式を使用した研究として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がある。[2][3][4] Finger Print 方式は特定の地点での AP からの電波強度モデルを作成して、実際の測定値をこのモデルの情報と照合して位置を推定する方式である。この方式はデータ収集コストが大きい点やモデルを作成しても環境の変化によってモデル

の信頼性が低下してしまう問題があり、それらの問題への対策を行った様々な研究がある。[5][6][7][8] Proximity 方式は特定の AP からの強い電波を受信した際、その AP 付近にいと見なし推定する手法である。これらの手法は状況に合わせて使いわけを行い組み合わせることでより精度の高い位置推定ができる。Proximity 方式と Finger Print 方式を併用した推定手法に関する研究がある。[9] この研究では Wi-Fi のフィンガープリント方式で位置を推定する前に有用に推定できる AP の絞りこみを行っている。電波以外推定手法として磁気データを使用して位置推定を行う研究 [10] や、赤外線を使用して位置推定を行う研究 [11]、カメラを利用した研究 [12] などがある。

PDR と絶対位置推定を組み合わせる屋内位置推定を行う研究がある。PDR は手法の性質上ある地点からの相対的な位置を推定する手法であるため、位置推定をするには絶対位置推定と組み合わせる必要がある。また PDR には 1 章で述べた誤差が蓄積する問題も存在する。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 [13] がある。この研究では加速度センサ、地磁気センサ、気圧センサの値を用いて既知地点からの位置推定を行う。Wi-Fi AP からの電波受信強度が特定の閾値を越えている場合はプロキシミティベースの位置推定に切り替える。これによって PDR で生じる誤差の補正を行っている。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 [14] がある。この手法では Wi-Fi の AP からの情報を使うのではなく BLE ビーコンを配置してその電波と位置情報をもとに PDR の誤差の補正を行っている。また PDR とマッピングを組み合わせて位置推定を行う研究 [15] がある。この手法ではマッピングを用いて歩幅を動的に更新を行い誤差を低減した PDR 手法を提案している。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る手法は、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ピ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なライブラリの検討を目指す。

## 3. PDR ベースの屋内位置推定ライブラリの検討

本章では、屋内位置推定ライブラリの要求仕様の検討およびその実装について述べる。節 3.1 では要求仕様について詳述する。節 3.2 では、軌跡の画像と関数を用いて、どのように補正を行っているかを示す。

### 3.1 要求仕様

PDR と他の情報を使ってライブラリを作成する上で、どのような状況や環境が存在し補正に利用できるのか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有効である。また 2 章で示したように [13][14] などの PDR と電波を利用した推定に関する研究は盛んに行われている。このように電波を使った手法は多くの場所でも有効であり、補正に利用可能な情報として重要度が高い。そのため本ライブラリにおいても採用を行う。他に補正に利用可能な情報としてフロアマップ情報がある。フロアマップ情報は多くの場所で比較的入手が容易だと思われる。そのため本ライブラリにおいても採用を行う。磁気やカメラなどの情報は、磁気データはデータが繊細であり電波と比べると補正に利用する難易度が高い、カメラはプライバシーの問題などの問題があり本ライブラリの基礎段階において採用しない。また気圧センサは基礎段階として 3 次元空間を推定対象としないため採用しない。

本ライブラリの検討および検証において xDR Challenge 2023 の環境を利用した。xDR Challenge 2023 は PDR ベンチマーク委員会が主催する屋内位置推定の精度を競うコンテストである。このコンテストの特定トラックでは人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)を持ち施設内を歩く。コンテストの参加者には、歩行時のデータが複数のトレーニングデータとして提供される。スマートフォンから加速度、角速度、磁気センサのデータ、および施設内に配置された各 BLE ビーコンの AP 情報と受信電波強度が提供される。LiDAR からは、正解歩行軌跡のデータが提供される。またフロアマップ情報、フロアマップ上での各 BLE ビーコンの基地局情報、歩行者の初期位置、終了位置が与えられる。コンテスト本番では LiDAR からの正解の移動軌跡が提供されず、それ以外の情報を使って歩行者の移動経路の推定を行う。xDR Challenge 2023 の環境は、本ライブラリの要求仕様に適合しており、ライブラリの有効性を検証するには適している。よってこの環境を基に検討および検証を行う。

### 3.2 ライブラリ

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後述する。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、BLE ビーコンの電波情報などが含まれる。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などが含まれる。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。

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。言語には Python を使用した。Python はデータ解析や機械学習などの分野で広く使われており、ライブラリを使用するユーザーにとっても比較的扱いやすい利点がある。まず基本的な PDR の処理を行う関数を Listing1 に示す。この関数では加速度データフレーム(以下、DF)、角速度 DF を使用して位置推定を行う。加速度 DF、角速度 DF 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。オプション引数として正解初期座標 (ground.truth.first\_point) を与えられる。正解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 DF と角度の DF であり、それぞれのカラム名とデータ型を表 5、表 6 に示す。戻り値で角度 DF を返す理由として、次の補正処理をする際に角速度よりも扱いやすいためである。

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 [16]、多変量解析を用いた研究 [17]、超音波センサーガジェットを用いた研究 [18] 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 1 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軌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。本ライブラリを用いてこれらの問題を解消し正解軌跡に近づけていく。

Listing 1: 基本 PDR

```
1 Axis2D = Literal["x", "y"]
2 def estimate_trajectory(
3     acc_df: pd.DataFrame,
```

	関数名	センサ情報				環境情報				その他			
		加速度	角速度	角度	BLE ビーコン	フロアマップ	磁気 FP	BLE ビーコン 基地局位置	FP	正解初期		正解補正	
					電波強度・AP 情報					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	○		○						○		○	
初期進行方向補正 フロアマップ	rotate_trajectory_to_optimal_alignment_using_map	○		○		○				△			
初期進行方向補正 BLE	rotate_trajectory_to_optimal_alignment_using_ble	○	○		○			○		△			
マップマッチング補正	move_unwalkable_points_to_walkable	○	○			○				△			
初期進行方向補正 BLE FP	rotate_trajectory_to_optimal_alignment_using_ble_fingerprint	○	○						○	△			

表 1: 関数に必要な情報とその対応表

注: ○ は必須引数, △ はオプション引数を示す

カラム名	単位	データ型
ts	s (秒)	float
x	m/s <sup>2</sup>	float
y	m/s <sup>2</sup>	float
z	m/s <sup>2</sup>	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	座標

表 4: 正解初期座標 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

```

4 gyro_df: pd.DataFrame,
5 *,
6 ground_truth_first_point: dict[Axis2D,
    float] | None = None,
7 ) -> tuple[pd.DataFrame, pd.DataFrame]:

```

Listing1 に示される関数に正解初期座標をを与えたのが

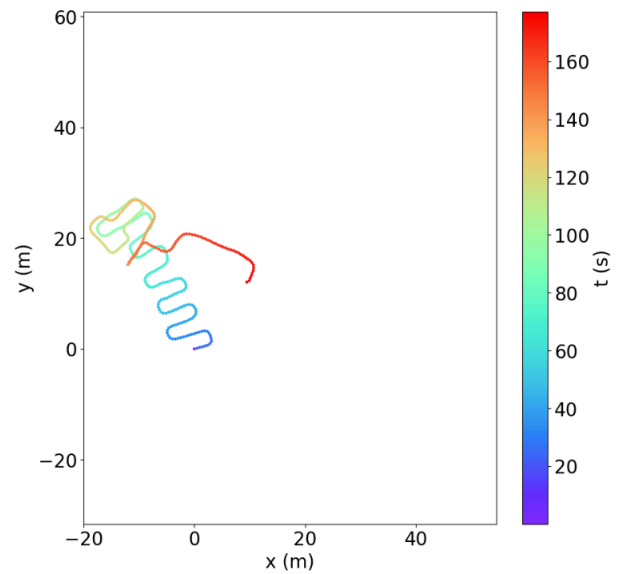


図 2: 基本 PDR の軌跡

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.

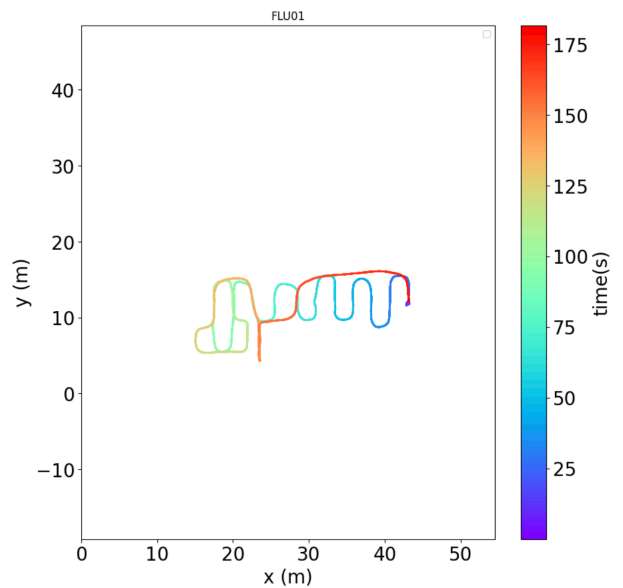


図 3: 正解軌跡

図 4 の軌跡には PDR 特有のドリフト現象が見られる. PDR では角速度から進行方向を求めてその方向を元に歩

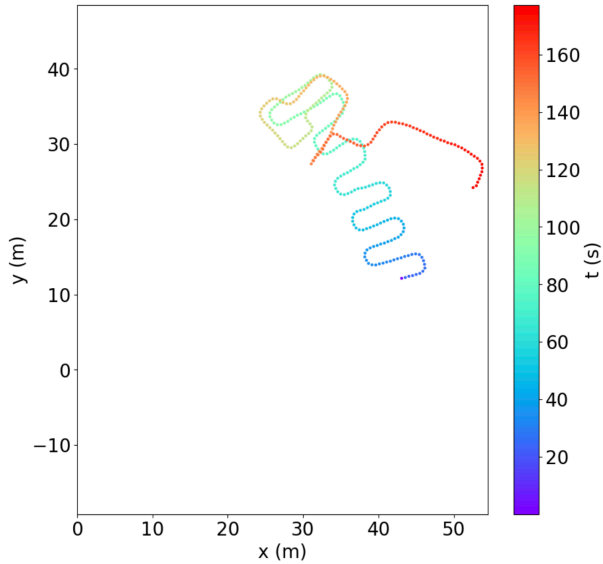


図 4: 正解初期座標が存在

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には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。

Listing 2: ドリフト除去

```
1 def remove_drift_in_angle_df(
2     acc_df: pd.DataFrame,
3     angle_df: pd.DataFrame,
4     ground_truth_point_df: pd.DataFrame,
5 ) -> tuple[pd.DataFrame, pd.DataFrame]:
```

ドリフトを取り除く関数を Listing2 に示す。引数として加速度 DF, 角速度 DF, 正解座標 DF を受け取る。戻り値は時間経過に伴う 2 次元座標の DF と角度の DF を返す。ドリフト補正のプロセスは、ドリフトの値を動的に計算し、それを各時刻の角度データから差し引く。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、 $d$  はドリフトの大きさを意味する。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。

$$\theta'(t) = \theta(t) - (d \times t) \quad (1)$$

補正の効果を評価し適切なドリフトを見つけるために、ユークリッド距離を用いて、2 つの正解座標の差異を計算する。式 (2) は、正解座標  $(x_n, y_n)$  と正解座標  $(x_{n+1}, y_{n+1})$  との間のユークリッド距離  $E$  を示している。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度 DF から引きそれに基づいた座標 DF と角度 DF を返す。図 5 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

解軌跡の形状に近づいている。このアルゴリズムでは正解座標  $(x_n, y_n)$  と正解座標  $(x_{n+1}, y_{n+1})$  の距離が近い時に特に有効である。この処理は  $(x_{n+2}, y_{n+2})$  など 2 つ以上の座標が存在する場合も同様に適用できる。

$$E = \sqrt{(x_{n+1} - x_n)^2 + (y_{n+1} - y_n)^2} \quad (2)$$

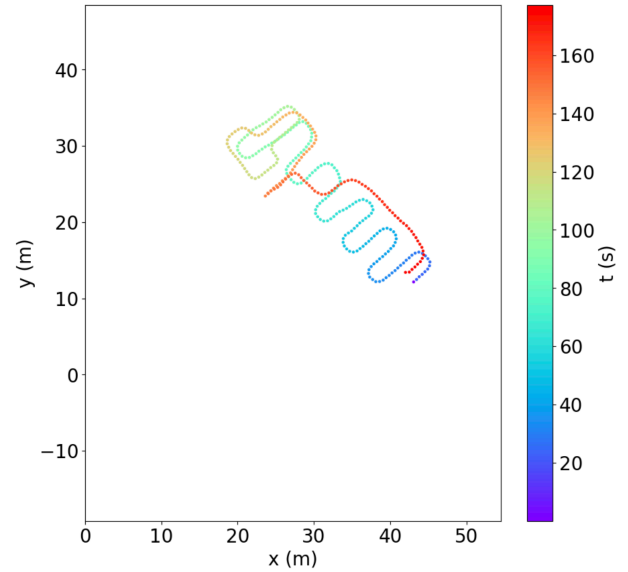


図 5: ドリフト補正後の軌跡

図 5 の軌跡の問題点として初期進行方向の誤差がある。初期進行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になる。この問題を解決するためには、適切な初期進行方向を見つけて軌跡全体を回転させる必要がある。フロアマップ情報を元に軌跡を回転させる関数を Listing3 に示す。引数として加速度 DF, 角度 DF, 表 7 に示すフロアマップ情報 DICT, フロア名, 及びマップの 1px あたりの距離を受け取る。内部の処理としては軌跡を回転させその時の水平垂直方向の割合を計算する。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 6 である。この割合が最も大きい回転角度をグリッドサーチを用いて探し最適な角度を見つける。しかしこの処理だけでは適切な初期進行方向は絞り込めない。割合が大きいものがあったとしても 90 度回転させるごとに水平垂直方向の割合が同一になるため 4 つの角度から適切な初期進行方向を見つける必要がある。この絞り込みの処理としてマップ上の通行可能、不可能な座標の情報を利用する。各回転角度での軌跡座標がマップ上で通行可能なポイントの数を計算し最も多いポイントを持つ回転角度を選択する。この処理を適用した結果が図 7 である。補正前と比べて軌跡の初期進行方向が正解軌跡に近づいている。



Listing 3: 初期進行方向補正

```

1 def rotate_trajectory_to
2     _optimal_alignment_using_map(
3     acc_df: pd.DataFrame,
4     angle_df: pd.DataFrame,
5     map_dict: dict[str, np.ndarray],
6     floor_name: str,
7     dx: float,
8     dy: float,
9     *,
10    ground_truth_first_point: dict[Axis2D,
11    float] | None = None,
12 ) -> tuple[pd.DataFrame, pd.DataFrame]:

```

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

表 7: フロマップ DICT

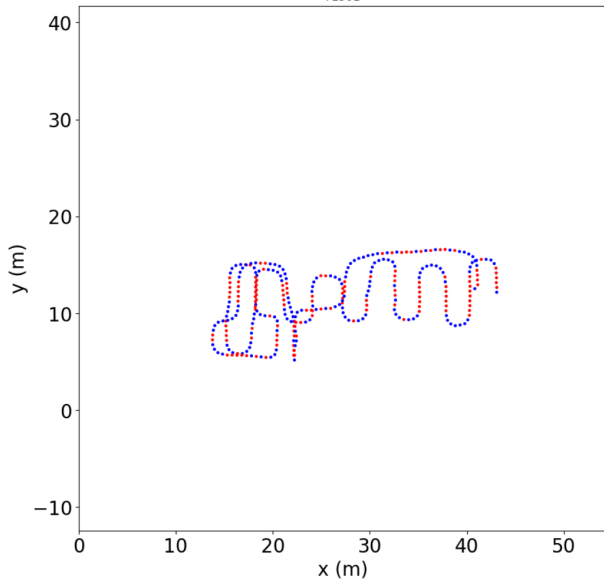


図 6: 垂直成分と水平成分の可視化

フロアマップ情報を用いた初期進行方向補正ではマップの存在可能な点の分布によっては正しく機能しない場合があります, 別の方法として BLE ビーコンの基地局の位置情報を用いた初期進行方向補正を行う関数を提供する. 関数を Listing4 に示す. この関数では加速度 DF, 角度 DF, BLE ビーコンの受信電波 DF, BLE ビーコンの基地局 DF を受け取る. BLE ビーコンの受信電波 DF と BLE ビーコンの基地局 DF のカラム名とデータ型を表 6, 表 7 に示す. 戻り値は時間経過に伴う 2 次元座標の DF と角度の DF を返す. 一定の強い RSSI の電波を受信した際の時間情報を元

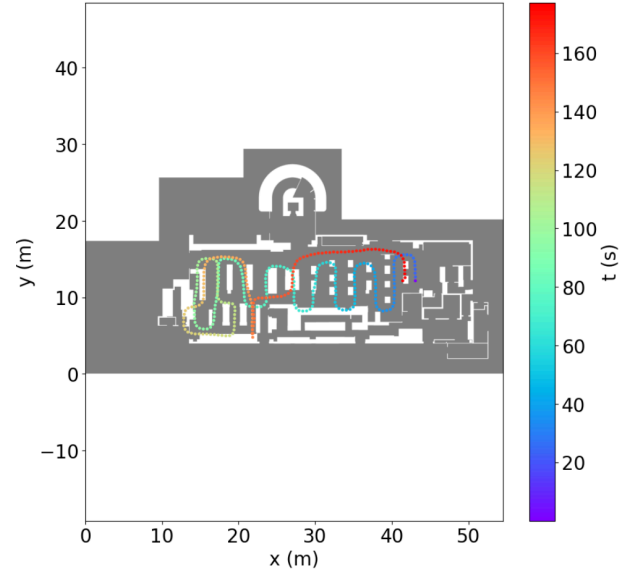


図 7: 初期進行方向の補正後の軌跡

に時間的に近い推定軌跡の座標を取得する. 図 8 に示した図は時間的に近い推定軌跡の座標を時間経過に応じた色で表しており青色の座標が配置された BLE ビーコンの座標を表している. 推定した軌跡の受信した BLE ビーコンの基地局の座標との距離を計算する. この総和が最小となるような回転角度をグリッドサーチで探し最適な角度に補正を行う. BLE ビーコンの基地局の座標との距離を計算する.

Listing 4: BLE ビーコンの基地局の位置情報を使用した初期進行方向補正

```

1 def rotate_trajectory_to_optimal
2     _alignment_using_ble(
3     acc_df: pd.DataFrame,
4     angle_df: pd.DataFrame,
5     ble_scans_df: pd.DataFrame,
6     ble_position_df: pd.DataFrame,
7     *,
8     ground_truth_first_point: dict[Axis2D,
9     float] | None = None,
10 ) -> tuple[pd.DataFrame, pd.DataFrame]:

```

カラム名	単位	データ型
ts	s (秒)	float
bdaddress	なし	str
rsssi	dBm	int

表 8: BLE ビーコン受信電波 DF

BLE ビーコンの基地局情報を元に初期進行方向の補正を行ったが常に基地局の位置情報が利用可能であるとは限らない. 電波を使った手法として Wi-Fi を使った手法もあるがこの場合も同様であり基地局の位置情報の把握にはコストがかかる場合がある. 基地局の位置情報を用いない代替

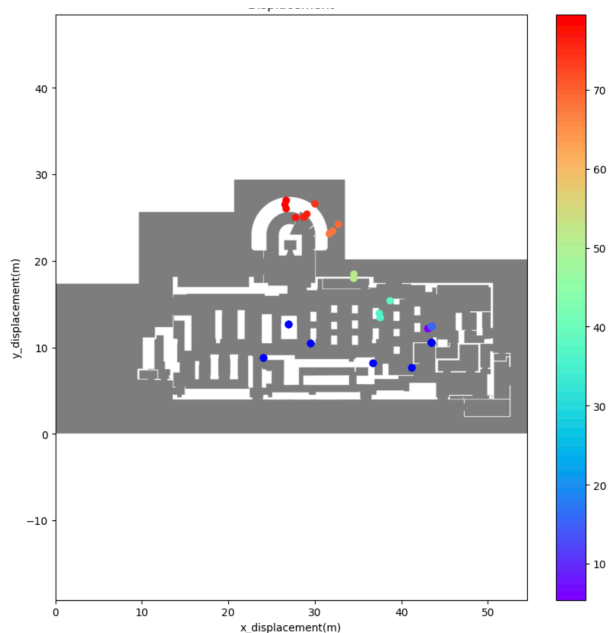


図 8: 強いビーコン電波を受信した際の  
時間的に近い軌跡の座標

カラム名	単位	データ型
bdaddress	なし	str
x	m	float
y	m	float
floor_name	なし	str

表 9: BLE ビーコン基地局 DF

手法として Fingerprinting(FP) を用いた手法がある。この手法は、事前に特定の場所で受信した BLE ビーコンの ID と電波強度のデータを蓄積しておく必要があり、そのデータを基に受信した ID と RSSI の値から位置を推定する。この手法を用いて初期進行方向を補正する関数を 5 に示す。この関数は引数に加速度 DF, 角度データ DF, BLE ビーコンの受信電波 DF, BLE ビーコンの FPDF, フロア名を引数に取る。戻り値として角度 DF と時間経過に 2 次元座標の DF を返す。受信電波情報と FP を基に推定した座標を示したのが図 9 である。図の青色の点が受信した BLE ビーコンの基地局座標であり、赤色の点がこの基地局から受信した電波と FP を元に位置を推定した座標である。理解しやすいように図中では受信した ID が 1 つのみを表示しているが、実際は複数の強い電波を受信した点が存在する。また説明のために基地局情報を示しているが今回の使用ケースではこの座標は判明していないのが前提である。この関数の内部処理では上記で示した受信電波情報と FP を基に推定した座標と推定軌跡の座標との距離の総和を用いて、その和が最小となる角度を探す。BLE ビーコンの基地局情報を元に初期進行方向を回転させた際と、ほぼ同様の結果が得られた。

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
z	m(メートル)	float
bdaddress	なし	str
rssi	dBm	int
floor_name	なし	str

表 10: BLE ビーコン FP の DF

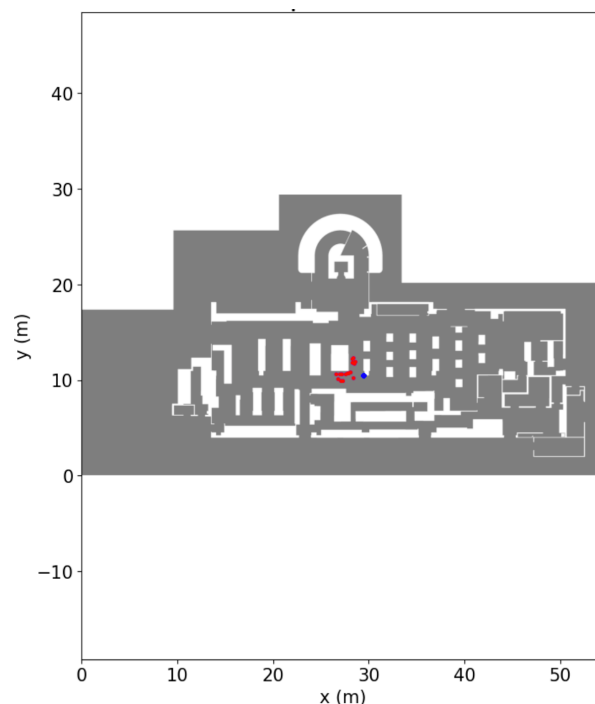


図 9: FP に基づく位置の推定

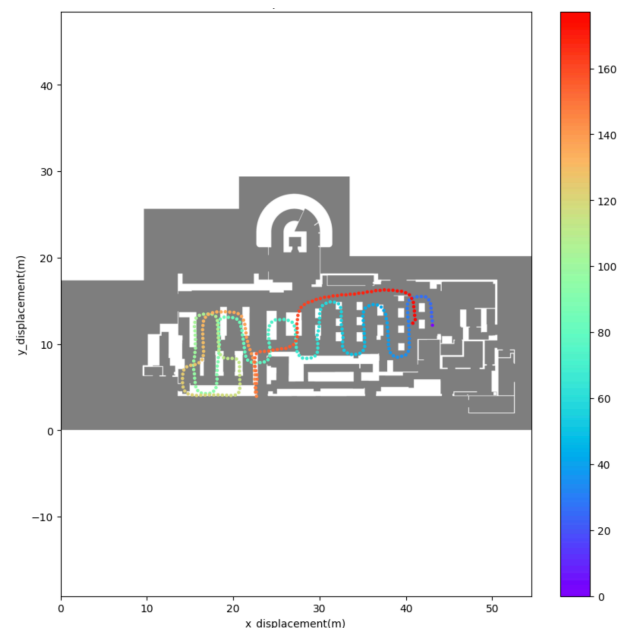


図 10: BLE の FP を補正後の軌跡

Listing 5: BLE ビーコンの FP を使用した初期進行方向補正

```

1 def rotate_trajectory_to_optimal
2     _alignment_using_ble_fingerprint(
3         acc_df: pd.DataFrame,
4         angle_df: pd.DataFrame,
5         ble_scans_df: pd.DataFrame,
6         ble_fingerprint_df: pd.DataFrame,
7         floor_name: str,
8         *,
9         ground_truth_first_point: dict[Axis2D,
10             float] | None = None,
11     ) -> tuple[pd.DataFrame, pd.DataFrame]:

```

図 7 に示す軌跡の問題点として人間が歩行不可能領域を通過している点がある。現実の人間がこのような場所を通過しないため、このような軌跡は不適切である。そのため軌跡が歩行不可領域に存在する場合は、歩行可能な領域に移動させる処理が必要である。この問題を解決する処理として Listing6 に示すマップマッチング補正関数を提供する。マップマッチング補正関数は加速度 DF, 角度 DF, フロアマップ情報 DICT, フロア名, 及びマップの 1px あたりの距離を受け取る必要がある。戻り値は時間経過に伴う 2 次元座標の DF のみを返す。内部の処理の関係上補正後の角度 DF は返すのが難しいためである。関数内部ではまず加速度と角度のデータを基にして軌跡を推定する。この軌跡に対して、各地点での座標が与えられたフロアマップ上の歩行可能な領域に存在するかどうかを検証する。検証の結果、各地点での座標が歩行不可能な領域に存在する場合、当該座標から最も近い歩行可能な座標を幅優先探索アルゴリズムを用いて探す。該当する座標が見つかった場合、該当座標と該当座標以降の軌跡の座標を歩行可能な座標に平行移動して補正を行う。当該座標の補正が終了後、次の座標に対して同様の処理を行う処理を繰り返す。これによって軌跡の各地点が歩行可能な領域に存在するようになり、軌跡全体が最適化される。図 11 に示すように、マップマッチング補正後の軌跡では歩行不可能な領域に存在していた地点が歩行可能な地点に移動されている。

Listing 6: マップマッチング補正

```

1 def move_unwalkable_points_to_walkable(
2     acc_df: pd.DataFrame,
3     angle_df: pd.DataFrame,
4     map_dict: dict[str, np.ndarray],
5     floor_name: str,
6     dx: float,
7     dy: float,
8     *,
9     ground_truth_first_point: dict[Axis2D,
10         float] | None = None,
11 ) -> pd.DataFrame:

```

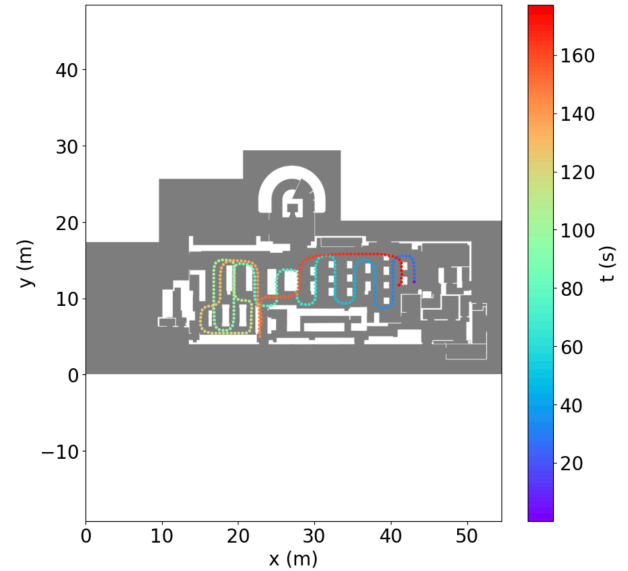


図 11: マップマッチング補正後の軌跡

## 4. ライブラリの検証と他環境における検討

この章では提案したライブラリの有効性を検証する。検証は xDR Challenge での評価を基に行い、また他環境でこれらのライブラリが適用可能かを検討する。

### 4.1 xDR Challenge 環境での評価

xDR Challenge では PDR ベンチマーク標準化委員会によって提供された評価フレームワークを使用して評価が行われた。このフレームワークでは  $I_{ce}$  (CE: 円形誤差),  $I_{ca}$  (CA: 局所空間における円形精度),  $I_{eag}$  (誤差蓄積勾配),  $I_{ve}$  (VE: 速度誤差),  $I_{obstacle}$  (障害物回避要件) の 5 つの評価指標が用いられた。総合評価指標は式 3 に示す式で計算される。このライブラリを使って得られた評価と各指標の重みを表 11 に示す。  $I_{ce}$ ,  $I_{eag}$ ,  $I_{ve}$ ,  $I_{obstacle}$  では一定の精度を得られた。しかし  $I_{ca}$  では精度が低かった。  $I_{ca}$  の値が低いと場合局所空間における位置推定誤差の分布が広い。これは環境条件の変化やセンサデータの微妙な違いが、位置推定結果に大きな影響を与える可能性が高い。この問題を解決するためには、PDR アルゴリズムを改善し、より精度の高い位置推定を行う必要がある。

$$\begin{aligned}
 I_i = & W_{ce} \times I_{ce} + W_{ca} \times I_{ca} \\
 & + W_{eag} \times I_{eag} + W_{ve} \times I_{ve} + W_{obstacle} \times I_{obstacle}
 \end{aligned} \quad (3)$$

### 4.2 駅の構内での検証

駅構内での位置推定をする場合を考える。駅の改札は地上から続いているものもあれば地下にあるものもある。地下の場合は特に衛星からの電波が届きにくい場所であるため GPS が有効ではない。このような環境では PDR が有効な



指標	値 (%)	重み
L <sub>ce</sub> (CE:円形誤差)	88.55	0.25
L <sub>ca</sub> (CAI:局所空間における円形精度)	62.51	0.20
L <sub>ea</sub> (EAG:誤差蓄積勾配)	93.02	0.25
L <sub>ve</sub> (VE:速度誤差)	95.55	0.15
L <sub>obstacle</sub> (障害物回避要件)	93.48	0.15
I (総合評価指数)	86.25	

表 11: 評価指数の概要

手法である。駅の改札の位置は工事などが無い限り、基本的に固定で変化することはない。改札を通った時の位置をPDRの開始地点を正解の初期座標として使用できそうである。改札を通して出た後、その後乗り換えを行う場合がある。このような場合は次の改札口を正解補正座標として利用できる。ICなどを使って駅改札を通った場合、ユーザを一意に識別できる。そのため特定のユーザが乗り換えをしたという情報を収集するのは比較的容易である。正解初期地点と正解補正座標を利用すればListing2に示したドリフト補正を適用できる。また全ての駅ではないがある程度以上の駅構内の場合フロアマップ情報が入手できる可能性が高い。その場合フロアマップ情報を用いたListing6のマップマッチング補正が適用できる。

#### 4.3 大学のキャンパスでの検証

大学のキャンパスで位置推定をする場合を考える。大学には屋外環境と屋内環境がある。建物間の移動経路を把握する場合はGPSが有効である。しかし大学の建物内での移動経路を把握する場合GPSでは困難である。このような場合にPDRを軸とした移動経路の把握を検討する。大学というのは研究室やサークルなど異なるコミュニティが混在している。それらは1つの組織が大本で管理しているのではなく個々が独立運営している。このような場所でBLEビーコンを配置する場合、各コミュニティへの申請のコストや、場合によっては配置を拒否される可能性がある。BLEビーコン以外の電波の利用を考えた場合、Wi-Fiの電波の利用が検討できる。Wi-Fiの基地局なら基本的にどのコミュニティにも配置がしてあり、設置コストの面でBLEビーコンと比べると低い。しかし既知のWi-Fiの基地局位置情報の把握はコストが大きい。そのためこのような場所ではWi-Fiの電波を使ったFP補正が有効だと考えられる。FPを使った手法なら基地局の位置情報を把握していない場合でも利用できる。3章ではBLEビーコンの元にFP処理を行う関数を実装した。Wi-FiとBLEは通信範囲や消費電力などで異なる点はあるが、内部の処理や閾値を変化させる必要はあるが、基本的に与える引数やそのデータ形式を揃えれば同様に適用できる。また部屋に出入りする際には固定の位置の出入り口がある。個人がそこを出入りしたという情報を取得できれば、正解初期座標や正解補正座標と

してドリフト除去を適用できる。

## 5. まとめ

本論文では、環境情報などを利用したPDRベースの位置推定ライブラリの基礎検討を行った。補正に利用できる情報をセンサ情報、環境情報、その他の3つに分類し、それぞれの情報を用いた補正処理を提案した。その結果として、xDR Challenge 環境下では一定の精度を獲得した。また他環境においても本ライブラリが適用可能であるか検討を行った。課題としては、PDR アルゴリズムの改善や挙げられる。また本論文は2次元の屋内位置推定のみを想定したライブラリ構成となっている。現実の屋内では3次元に構成されるものが多いため、本ライブラリを3次元空間に適用できるような拡張を検討したい。具体的にはスマートフォンの気圧センサの値を使用すれば相対的な階層間の移動の検知が可能である。これとフロアマップ情報を組み合わせることで3次元空間での位置推定が実現できると考えられる。

## 参考文献

- [1] 伊藤誠悟, 吉田廣志, 河口信夫ほか. 無線 lan を用いた広域位置情報システム構築に関する検討. 情報処理学会論文誌, Vol. 47, No. 12, pp. 3124–3136, 2006.
- [2] 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.
- [3] 浦野健太, 廣井慧, 梶克彦, 河口信夫ほか. 配布型 ble タグとタンデムスキャナを用いた屋内位置推定手法. 情報処理学会論文誌, Vol. 60, No. 1, pp. 58–75, 2019.
- [4] Hena Kausar and Suvendu Chattaraj. A novel kalman filter based trilateration approach for indoor localization problem. In *2022 International Conference for Advancement in Technology (ICONAT)*, pp. 1–5, 2022.
- [5] 藤田迪, 梶克彦, 河口信夫ほか. Gaussian mixture model を用いた無線 lan 位置推定手法. 情報処理学会論文誌, Vol. 52, No. 3, pp. 1069–1081, 2011.
- [6] 久保田僚介, 田頭茂明, 荒川豊, 北須賀輝明, 福田晃ほか. 無線 lan を用いた屋内位置推定における学習コスト削減のための高精度データ補間手法. 情報処理学会論文誌, Vol. 54, No. 4, pp. 1609–1618, 2013.
- [7] 谷内大祐, 前川卓也ほか. 位置フィンガープリントの自動更新を用いた電波環境変化に頑健な屋内位置推定手法. 情報処理学会論文誌, Vol. 55, No. 1, pp. 280–288, 2014.
- [8] Y. Tian, J. Wang, and Z. Zhao. Wi-fi fingerprint update for indoor localization via domain adaptation. In *2021 IEEE 27th International Conference on Parallel and Distributed Systems (ICPADS)*, pp. 835–842, Los Alamitos, CA, USA, dec 2021. IEEE Computer Society.
- [9] 伊藤誠悟, 河口信夫. アクセスポイントの選択を考慮したベイズ推定による無線 lan ハイブリット位置推定手法とその応用. 電気学会論文誌 C (電子・情報・システム部門誌), Vol. 126, No. 10, pp. 1212–1220, 2006.
- [10] 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.
- [11] 岡田遼太郎, 矢入郁子. 赤外線グリッドセンサを用いた室

内位置・行動推定の粒度及び精度の検討. 人工知能学会全国大会論文集, Vol. JSAI2014, pp. 2J14-2J14, 2014.

- [12] 亀田能成, 大田友一. 街中での歩行者カメラによるオンライン位置推定のための検討. 画像の認識・理解シンポジウム MIRU, pp. 364-369, 2010.
- [13] 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.
- [14] 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.
- [15] 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.
- [16] 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.
- [17] 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.
- [18] 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.