

# スマートフォンとスマートウォッチを併用した PDR手法の地図情報を利用した高精度化

若泉 朋弥<sup>1,a)</sup> 戸川 望<sup>1,b)</sup>

## 概要：

スマートフォンの普及により、歩行者向けナビゲーションシステムが多く利用されている。こうしたナビゲーションシステムではGPSを利用し現在位置を測位しているが、屋内や地下ではGPSによる測位が困難である。屋内での位置推定手法として Pedestrian Dead Reckoning (PDR) が提案されている。PDR手法は歩行者の身に付ける端末からセンサ情報を取得し、初期地点からの経路をたどることで現在位置を推定する。これまでに我々は、スマートフォンとスマートウォッチを併用したPDR手法を提案し、外部インフラを用いず、スマートフォンの位置によらず自己位置推定が可能であり、そのうえで推定誤差を削減できることを、評価実験により確認している。しかし、方向転換時に進行方向の誤差が残り、誤差が蓄積するため、右左折の多い環境では推定誤差が大きくなる問題がある。本稿では、スマートフォンとスマートウォッチを併用したPDR手法の精度改善手法を提案する。提案手法では、新たに地図情報を組み合わせ、マップマッチングを導入することにより、従来手法の精度を向上させる。実現評価を通じて、提案手法の有効性を確認した。

## 1. はじめに

### 1.1 PDRによる屋内位置推定

近年、スマートフォンをはじめとする携帯端末の普及により、歩行者向けナビゲーションシステムが多く利用されている。こうしたナビゲーションシステムでは一般的にGPS (Global Positioning System) を利用して、自らの現在位置を推定している。しかしGPSでは屋内や地下では衛星からの電波が届きにくく、精度が大幅に低下する [1]。そのような場所ではGPSに代わる手段により現在位置を推定する必要がある。

PDR (Pedestrian Dead Reckoning) は屋内位置推定手法の一つであり、歩行者に身に付けられたセンサから得られる加速度や角速度などの歩行データをもとに、初期地点からの歩行距離および進行方向を予測することで現在位置を推定する手法である。PDR手法は、一般的にWi-FiアクセスポイントやBluetoothビーコンなどの外部インフラを利用せず、導入コストが低い傾向がある。一方で、スマートフォンを用いたPDR手法では、位置測位中の端末の向きあるいは端末モード (本稿では文献 [2, 3] にならぬ端末の持ち方や向きをモードと呼ぶ) として、端末を身体の前

方に手で持っていたり、鞆の中に端末を入れていたり、端末に対して特定のモードを仮定することが難しい。しかもあるモードに最適化されたPDR手法で、別のモードの端末の位置測位を実行すると、位置測位の誤差が大きくなる恐れがある。

つまり、スマートフォンを用いたPDR手法では、(条件 a) 特定の外部インフラを用いず、(条件 b) スマートフォンのモードによらず自己位置を推定する手法が強く求められる。しかも (条件 c) PDR手法による測位誤差をできるだけ小さくすることが求められる。

### 1.2 先行研究

スマートフォンを用いたPDR手法は現在まで複数提案されている [2-5]。これらの手法は仮定したモードについて、高精度な位置測位が可能である。しかし、仮定したモード以外ではモードの推定が難しい。これらの手法は外部インフラを用いないPDR手法となっているが、スマートフォンのモードに依存した位置測位になっており、(条件 a) は満足するが (条件 b) は満足しない。

星らの手法では、外部インフラを必要とせず、またスマートフォンのモードを予め仮定せずに位置推定が可能である [6]。しかし、角速度のみでの進行方向推定を行っており、ドリフトによる推定誤差が生じやすく、(条件 a) お

<sup>1</sup> Waseda University

<sup>a)</sup> tomoya.wakaizumi@togawa.cs.waseda.ac.jp

<sup>b)</sup> togawa@togawa.cs.waseda.ac.jp

よび (条件 b) は満足するが (条件 c) は満足しない。

スマートフォンが多くのモードを持つのに対し、スマートウォッチ [7,8] は通常、手首に固定されて装備するため、モード変化の影響を受けにくく、頑健な PDR 手法の実現が期待される。Correa らはスマートフォンとスマートウォッチを併用した PDR 手法を提案している [9]。Correa らの手法では比較的精度の高い位置測位を実現しているが、Wi-Fi が必須であり、予め Wi-Fi 基地局の設置と場所の登録が不可欠となり上述の (条件 a) を満足しない。またスマートフォンのモードも固定されており (条件 b) も満足しない。

我々は、スマートフォンとスマートウォッチを併用することで、任意のスマートフォンモードに対応した位置推定を提案した (以降、PDR-PW 手法と呼ぶ)。さらに、提案した手法は既存手法と比較して、高い精度での位置推定を可能にすることを評価実験によって確認している [10]。しかし、角速度によって進行方向を推定しているため、方向転換時に角度誤差が生じる場合がある。上述の (条件 a) および (条件 b) を満たしているが、(条件 c) の満足が十分とは言えない。

### 1.3 マップマッチング

PDR の誤差補正手法として、マップマッチングを用いた PDR 手法が提案されている [11–13]。マップマッチングは地図情報を用いることで、PDR によって推定した位置を補正し、精度の高い位置推定を実現している。安齋らの手法では直進時には通路に沿って進行方向を補正し、方向転換時には曲がり角の地点に推定位置を補正している [11]。Shin らの手法では、経路が分岐したあとの推定経路を考慮して、補正する経路を選択することで、誤補正の少ないマップマッチングを実現している [12]。特に最近では、岡里らが地図情報で得た壁と衝突する際、歩行経路を通路に沿って修正する手法を提案している [13]。

マップマッチングを PDR 以外に組み合わせた屋内位置手法には以下のものがある。石丸らは、PDR 測位と Bluetooth ビーコンでの測位にマップマッチングを適用した手法を提案している [14]。この手法では、屋内地図によって人が侵入できない場所を定義し、Bluetooth から推定される位置候補を絞ることで、推定精度を向上させている。村田らは行動情報と屋内地図を利用した測位手法を提案している [15]。この手法では、階段の上り下りなどの歩行者の行動と、地図上の階段の位置などを照らし合わせることで、歩行者の位置を補正している。

マップマッチングは地図情報を用いることで、PDR によって推定した位置を補正し、精度の高い位置推定を実現している。しかし、マップマッチングによって補正した経路が実際の経路と異なる可能性がある。補正を誤ると大きな誤差 (以降、マップマッチング誤差と呼ぶ) が生じ、その

後の推定に大きな蓄積誤差が生じる。より頑健な PDR 手法を実現するためには、マップマッチング誤差をいかに軽減するかが大きな課題となる。

スマートフォンとスマートウォッチを併用した PDR 手法にマップマッチングを組み合わせることができれば、方向転換時の角度誤差を削減し、より高精度な PDR 手法が実現できると期待できる。

### 1.4 本稿の提案

以上の議論のもと、本稿では、スマートフォンとスマートウォッチを併用した PDR 手法に対し、マップマッチングを適用することで、測位精度を向上する PDR 手法を提案する (以降、改良 PDR-PW 手法と呼ぶ)。提案手法ではマップマッチングを適用する際に、複数のシナリオを用意し、尤度をもとに最適なシナリオを切り替える。これによりマップマッチング誤差の軽減を図る。地図情報を位置推定に取り入れることで、方向転換時の進行方向誤差を抑え、位置推定精度の向上を図る。

### 1.5 本稿の貢献

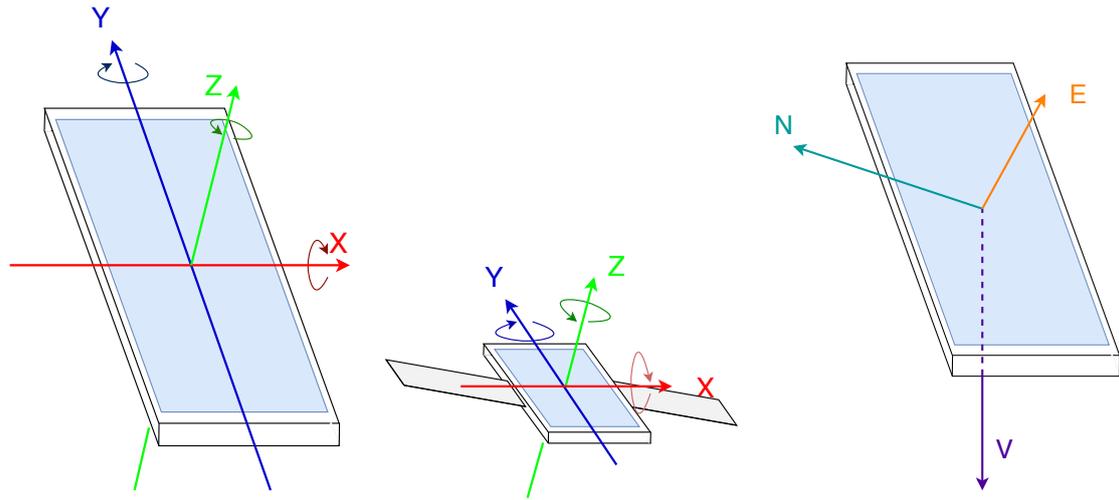
本稿の貢献は以下の通りである。

- (1) 我々が提案した PDR-PW 手法に地図情報を取り入れることで、PDR-PW 手法における進行方向推定の精度向上を図る。
- (2) 推定経路の候補をシナリオと定義する。各シナリオごとに尤度を計算し、最も尤もらしいシナリオを推定経路として採用することで、マップマッチング誤差を軽減する。
- (3) スマートフォンを様々なモードで保持し、歩行実験を実施した。その結果、従来手法 [10] と比較し、平均約 85% の位置推定誤差を削減した。

## 2. PDR による屋内位置推定

本章では PDR 手法による屋内位置推定問題を定義する。本稿における位置推定とは、スマートフォンならびにスマートウォッチに搭載されているセンサ (3 軸加速度と 3 軸角速度) と、屋内地図から自己位置を推定するものとする。

スマートフォン [16,17] に搭載された 3 軸加速度センサは、一般に、図 1(a) のようなスマートフォンの画面に対し水平横方向 (図中 X 軸)、水平縦方向 (図中 Y 軸)、垂直方向 (図中 Z 軸) の各軸の方向に対応した加速度  $[m/s^2]$  を取得する。時刻  $t$  におけるスマートフォンの 3 軸加速度センサの値を  $\vec{a}_p(t) = (a_p^x(t), a_p^y(t), a_p^z(t))$  とする。  $a_p^x(t), a_p^y(t), a_p^z(t)$  はそれぞれ時刻  $t$  の X 軸、Y 軸、Z 軸方向の加速度を表す。スマートフォンに搭載された 3 軸角速度センサは、一般に、図 1(a) のようなスマートフォンの画面に対し水平横方向 (図中 X 軸)、水平縦方向 (図中 Y 軸)、垂直方向 (図中 Z 軸) の各軸を回転軸とした角速度  $[rad/s]$  を取得する。



(a) スマートフォンにおける 3 軸方向. (b) スマートウォッチにおける 3 軸方向. (c) 地理空間における 3 軸方向.

図 1: 3 軸方向の定義.

時刻  $t$  におけるスマートフォンの 3 軸角速度センサの値を  $\vec{\omega}_p(t) = (\omega_p^x(t), \omega_p^y(t), \omega_p^z(t))$  とする.  $\omega_p^x(t), \omega_p^y(t), \omega_p^z(t)$  はそれぞれ時刻  $t$  の X 軸, Y 軸, Z 軸の各軸を回転軸とした角速度を表す.

スマートウォッチ [8, 18] に搭載された 3 軸加速度センサは, 一般に, 図 1(b) のようなスマートウォッチの画面に対し水平横方向 (図中 X 軸), 水平縦方向 (図中 Y 軸), 垂直方向 (図中 Z 軸) の各軸の方向に対応した加速度  $[\text{m/s}^2]$  を取得する. 時刻  $t$  におけるスマートウォッチの 3 軸加速度センサの値を  $\vec{a}_w(t) = (a_w^x(t), a_w^y(t), a_w^z(t))$  とする.  $a_w^x(t), a_w^y(t), a_w^z(t)$  はそれぞれ時刻  $t$  の X 軸, Y 軸, Z 軸方向の加速度を表す. スマートウォッチに搭載された 3 軸角速度センサは, 一般に, 図 1(b) のようなスマートフォンの画面に対し水平横方向 (図中 X 軸), 水平縦方向 (図中 Y 軸), 垂直方向 (図中 Z 軸) の各軸を回転軸とした角速度  $[\text{rad/s}]$  を取得する. 時刻  $t$  におけるスマートウォッチの 3 軸角速度センサの値を  $\vec{\omega}_w(t) = (\omega_w^x(t), \omega_w^y(t), \omega_w^z(t))$  とする.  $\omega_w^x(t), \omega_w^y(t), \omega_w^z(t)$  はそれぞれ時刻  $t$  の X 軸, Y 軸, Z 軸の各軸を回転軸とした角速度を表す.

屋内地図はフロアの障害物を構成する図形の集合  $B$  によって与えられるものとする. 歩行者は障害物以外の場所を歩行可能とする. 図 2 に屋内地図の例を示す. 図 2 において, 障害物は灰色で示す図形である. 各障害物は  $B_1$  から順に ID が付けられており, 隣接している壁などは 1 つの図形として表される.

また, 端末の初期位置  $\vec{p} = (x_0, y_0)$  ならびに初期方位  $\theta_0$ , 歩行者の歩幅  $l$  は与えられるものとする. またスマートフォンの 3 軸加速度センサと 3 軸角速度センサは,  $t_i^p$  秒ごとに取得する\*1. スマートウォッチの 3 軸加速度センサと

3 軸角速度センサは,  $t_i^w$  秒ごとに取得する. なおスマートフォンの 3 軸加速度センサと 3 軸角速度センサと, スマートウォッチの 3 軸加速度センサと 3 軸角速度センサは必ずしも同期しておらず, 同じ時刻のセンサデータが取得できるとは限らないことに注意する.

以上のもと, 自己位置推定問題を次のように定義する.  
**定義 1.** スマートフォンとスマートウォッチを併用した自己位置推定問題とは,  $t_i^p$  秒ごとにスマートフォンから 3 軸加速度  $\vec{a}_p(t)$  と 3 軸角速度  $\vec{\omega}_p(t)$  が与えられ,  $t_i^w$  秒ごとにスマートウォッチの 3 軸加速度  $\vec{a}_w(t)$  と 3 軸角速度  $\vec{\omega}_w(t)$  が与えられたとき,  $t_i$  秒ごとに現在の自己位置  $(x, y)$  を推定することである\*2. ただし, 屋内地図, 端末の初期位置  $\vec{p} = (x_0, y_0)$  ならびに初期方位  $\theta_0$ , 歩幅  $l$ , は与えられ, スマートフォンの 3 軸加速度センサと 3 軸角速度センサと, スマートウォッチの 3 軸加速度センサと 3 軸角速度センサは必ずしも同期しておらず, 同じ時刻のセンサデータが取得できるとは限らない. また歩行者は与えられた屋内地図において歩行可能な場所のみ歩行するものとする.

### 3. スマートフォンとスマートウォッチを併用した PDR 手法

本章では, 我々が文献 [10] によって提案したスマートフォンとスマートウォッチを併用した PDR 手法 (PDR-PW 手法) の概要を説明する.

PDR-PW 手法ではスマートフォンとスマートウォッチ各端末において, 3 軸加速度および 3 軸角速度を取得し, 取得したデータに基づいて歩数や端末角度を算出する. その後, スマートフォンとスマートウォッチの時間を同期させ, 方向推定を実行する.

以下に PDR-PW 手法の流れを示す.

\*1 4 章の評価実験では,  $t_i^p = 25$  ミリ秒,  $t_i^w = 50$  ミリ秒とした [8, 17].

\*2 4 章の評価実験では,  $t_i = 50$  ミリ秒とした.

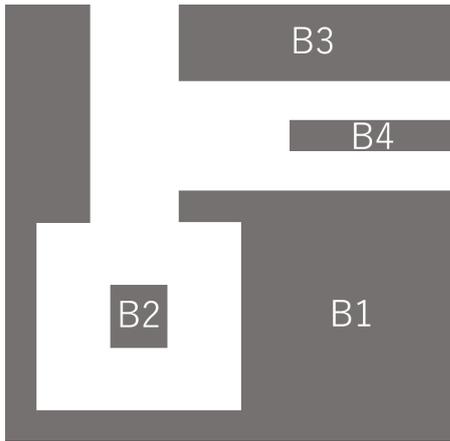


図 2: 屋内地図の例。

#### (Step 1) 歩数推定

歩数推定では、スマートフォンの加速度の周期性を利用し歩数をカウントする。スマートウォッチでは角速度の周期性を利用し歩数をカウントする。

#### (Step 2) 端末角推定

端末角推定では、スマートフォンおよびスマートウォッチを基準とした 3 軸を、加速度をもとに、地理空間上の 3 軸に変換する。1 歩ごとの合計加速度の平均を用いて、重力ベクトルを求めることで実現する。

#### (Step 3) 方位推定

PDR-PW 手法では屋内における地磁気の不安定さを避けるため、方位推定に角速度を利用する。スマートフォンとスマートウォッチの水平角速度を初期地点から積分することで、初期方位との差分を求め、進行方位を推定する。

#### (Step 4) 同期

スマートフォンとスマートウォッチを併用する際にデータ送受信の遅延が問題となる。PDR-PW 手法では、Step 1 で算出した歩数を用い、スマートフォンとスマートウォッチの同期をとることで、遅延の問題を解決する。

#### (Step 5) 共分散算出・方向推定

Step 3 の角速度による方位推定では、ドリフトによる蓄積誤差が問題となる。PDR-PW 手法では、スマートフォンとスマートウォッチの角速度の共分散を取り、正の共分散の大きい場合のみ進行方向を変化させることで、直進時のドリフトを抑制し精度の高い方向推定を実現する。

#### (Step 6) 現在位置推定

Step 4 で同期をとった歩数と、Step 5 で算出した方向を用いて、現在位置を推定する。

### 4. マップマッチング併用による高精度化

PDR-PW 手法は文献 [3] や文献 [6] と比べて、モードを

固定せず、またドリフト誤差を補正できるため、比較的精度良い自己位置推定ができる。しかしながら、移動時の端末の振れや、方向転換での速度の違いによって、方向転換時に進行方向誤差が生じる場合がある。進行方向の誤差はその後の推定にも影響を与えるため、推定時間が長くなるにつれ、大きな誤差を生じる問題がある。

マップマッチングは地図情報を用いて、歩行者の位置や進行方向を補正する手法である。歩行者は歩行可能な場所のみを歩行するため、推定位置が壁と衝突した場合は、推定結果に誤りがあると判定できる。マップマッチングは、地図情報と推定位置を照らし合わせることで、誤りのある推定結果を補正することが期待できる。

一方で、マップマッチングは地図情報で推定結果を補正した際に、誤った通路に位置を補正をするマップマッチング誤差が生じる可能性がある。マップマッチング誤差はその地点での誤差が大きくなるだけでなく、その後の推定に大きな誤差が蓄積し続ける可能性がある [19]。

マップマッチング誤差の削減方法として、マルチシナリオマップマッチングを導入する。マルチシナリオマップマッチングは経路補正をする際に、シナリオと呼ばれる補正候補を複数保持し続ける。各シナリオは尤度を用いて評価し、最も尤もらしいシナリオを現在の補正位置として採用する。マルチシナリオマップマッチングは複数のシナリオを保持し続けるため、一度誤った補正をしても、その後の推定によって、尤度に変化があれば、正しいシナリオを選択しなおす可能性がある。よってマップマッチング誤差を削減が期待される。

以上の議論をもとに、文献 [20] を参考に、マルチシナリオマップマッチングによる PDR-PW 手法の高精度化手法 (改良 PDR-PW 手法) を提案する。改良 PDR-PW 手法は、あらかじめ歩行者が歩行する可能性がある経路を設定し、マップマッチングを用いて、推定結果を補正する。その際にマルチシナリオマップマッチングを用いることで、マップマッチング誤差を削減する。

#### 4.1 改良 PDR-PW 手法の流れ

以上の議論のもと、改良 PDR-PW 手法の流れを図 3 に示す。改良 PDR-PW 手法は 3 章で示した PDR-PW 手法に Step 0 および Step 7, Step 8 を追加し、地図情報によって推定位置を補正している。また、推定にあたり、事前に地図情報から、歩行者が歩行する可能性のあるルートを抽出し、歩行エッジとして管理する。

以下に追加した改良 PDR-PW 手法の流れを示す。Step 1 ~ Step 6 については、3 章で示した PDR-PW 手法と同様である。

#### (Step 0) 歩行エッジの作成

屋内地図をもとに、歩行者が歩く可能性のある経路を歩行エッジ、転換する地点を歩行ノードとして作成

する。

#### (Step 7) マップマッチングによる経路補正

Step 5 から検出した方向転換地点と Step 6 で推定した現在位置をマップと照らし合わせ、補正候補 (シナリオ) を作成する。方向転換地点でいくつか補正候補がある場合は複数のシナリオを作成し、保持し続ける。

#### (Step 8) 二次現在位置推定

Step 7 で作成したシナリオのうち、最も尤もらしいシナリオを用いて、Step 6 で求めた推定位置を補正する。補正された位置を最終的な推定地点とする。

### 4.2 歩行エッジの作成 (Step 0)

PDR-PW 手法は方向転換時の進行方向推定に誤差が生じており、蓄積誤差の増加の要因となっている一方で、方向転換地点は正しく認識されている。つまり PDR-PW 手法において、方向転換時に通路や壁に沿って進行方向を補正することで、より正確な位置推定が期待できる。

以上の議論のもと、改良 PDR-PW 手法はあらかじめ歩行者が歩く経路を歩行エッジ、歩行者が転換する地点を歩行ノードとして管理し、歩行エッジおよび歩行ノードを基準とした位置、進行方向を補正する。

ここでは、屋内地図に含まれる障害物の集合  $B$  をもとに可視グラフ [21] を生成することで、歩行エッジおよび歩行ノードを生成する。

以下に図 5 を用いて、可視グラフによる歩行エッジおよび歩行ノードの生成手順を示す。歩行エッジおよび歩行エッジ作成にあたって、図 5 (a) のような屋内地図が与えられると仮定する。灰色の図形が障害物、黄色線が歩行エッジ、青点が歩行ノードを表している。

- (1) 図 5 (b) に示すように、障害物の頂点から、一定距離離れた地点に、歩行ノードの候補点を作成する。図 5 (b) において、緑点が歩行ノードの候補点を表している。
- (2) 図 5 (c) に示すように、通路に対して歩行ノードの対応付けを行う。図 5 (c) において赤枠で囲われた部分が通路である。薄緑の点是对対応付け前の歩行ノードの候補点であり、青点是对対応付け後の歩行ノードである。通路は歩行可能なエリアのうち、細長いエリアを抽出する。通路の出入口付近の歩行ノード候補に対して対応付けを行うことで、通路出入口の歩行ノードがそれぞれの通路出入口に対し 1 つになるようにする。また、通路の歩行ノードが通路に対して平行に並ぶようにする。
- (3) 図 5 (d) に示すように、歩行ノード同士を接続し、歩行エッジを作成する。歩行エッジは歩行ノード同士の間に障害物がない場合、つまり、歩行ノード同士が可視の場合に接続する。

図 5 (e) に示すように、手順 (3) において、通路外の歩行ノードと通路内に存在する歩行ノードと接続する際には、

通路外の歩行ノードから該当の通路の一番近い出入口に存在する歩行ノードのみと接続し、その他の通路内の歩行ノードとは接続しないようにする。図 5 (e) において、歩行ノード A と歩行ノード B は間に障害物が存在しないが、通路外の歩行ノード A からみて、通路内の歩行ノード B は、一番近い出入口に存在する歩行ノードではないため、接続しないようにする。

また、上記の手順により、図 6 (a) に示すような歩行ノードおよび歩行エッジが作成されたと仮定する。ここで図 6 (a) の緑矢印のように歩行者が移動したと仮定する。このとき、歩行者は 1 回方向転換をしているが、図 6 (a) で作成されている歩行ノードと歩行エッジで表現するためには、歩行ノードを 2 回曲がる必要がある。改良 PDR-PW 手法では、方向転換の回数を実際の動きと一致させるために、新たに歩行ノードを追加する。図 6 (b) に示す青点線のように、通路の歩行エッジを延長する。延長した線上と、他の歩行エッジが交差した場合、図 6 (b) に示す緑点のように、新しく歩行ノードを追加する。この手順により、歩行者の実際の方向転換回数と、作成した歩行ノードおよび歩行エッジにおける方向転換回数が一致する。

### 4.3 マップマッチングによる経路補正 (Step 7)

Step 7 では Step 6 までに求めた推定位置と推定進行方向を用いて、歩行ノードおよび歩行エッジと照らし合わせることで、位置や進行方向を補正する。

マップマッチングは進行方向の補正に有用である一方で、ある歩行ノードからつながる歩行エッジが複数存在する場合に、マップマッチング誤差が生じる。

改良 PDR-PW 手法では、マルチシナリオマップマッチングを用いることで、誤った歩行エッジを選択しても、その後の推定で正しい歩行エッジへの推定を可能にする [22]。マルチシナリオマップマッチングは歩行エッジを選択する際、考えられる歩行経路をシナリオとして複数保持し続ける。各シナリオは尤度を用いて評価され、その地点での尤度が一番高いシナリオを採用することで、最も実際の歩行に近い歩行経路を推定できる。各シナリオは保持され続けるため、一時的に誤った経路に補正されても、その後の推定により尤度が変化した場合、正しい経路に補正なおされることが期待される。以下に図 7 を用いて、マルチシナリオマップマッチングによる補正手順を示す。図 7 において、青点線が歩行エッジ、赤点が歩行ノード、黄色点が現在推定位置、黄色矢印が現在推定進行方向、青矢印が存在するシナリオを表している。

- (1) 図 7 (a) に示すように、現在推定位置から最も近い歩行ノードを一つ選択し、その歩行ノードから接続するそれぞれの歩行エッジに対して、1 つずつシナリオを作成する。図 7 (a) では、現在推定位置から最も近い歩行ノードから、シナリオ 1 とシナリオ 2 が作成され

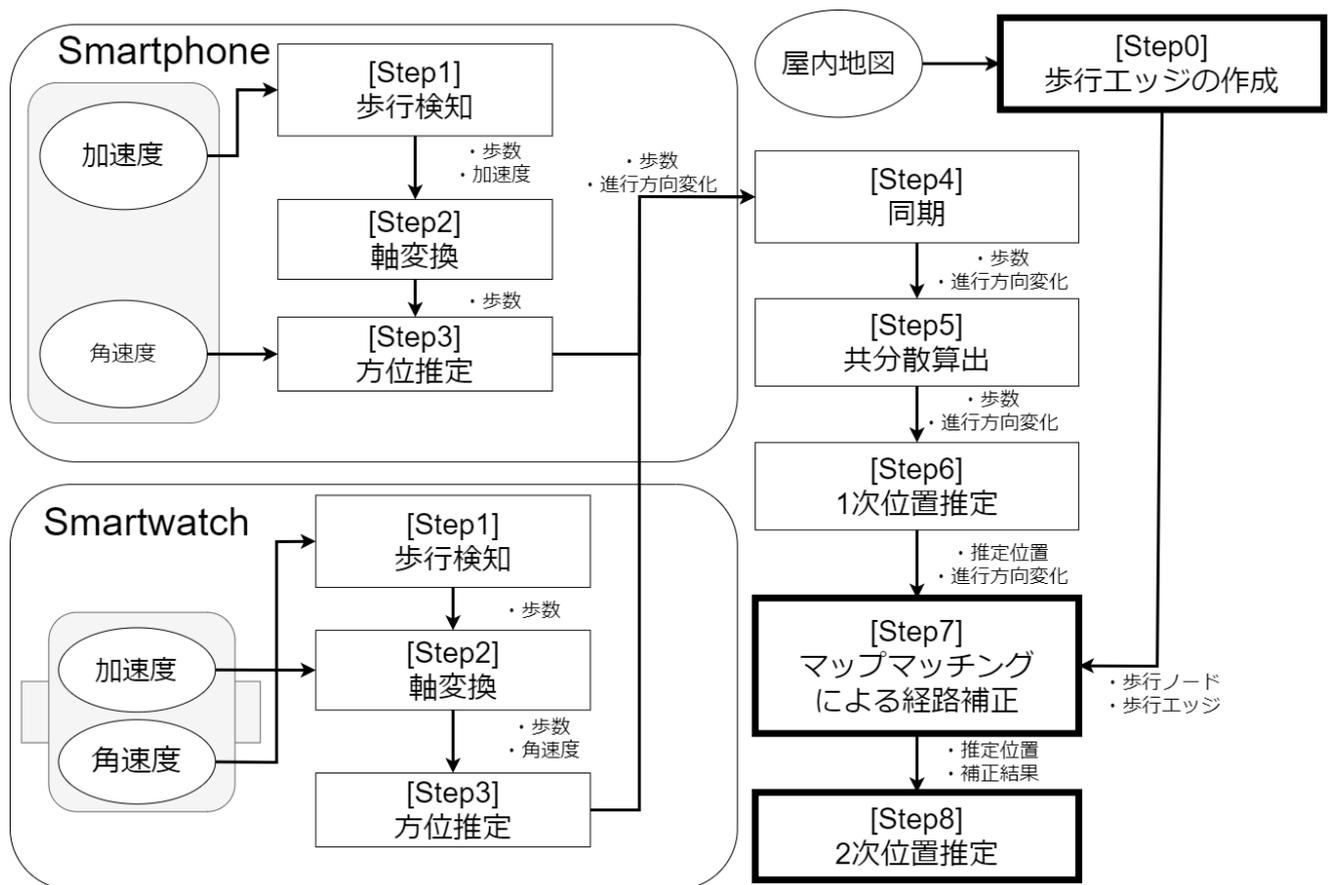


図 3: 改良 PDR-PW 手法のフローチャート。

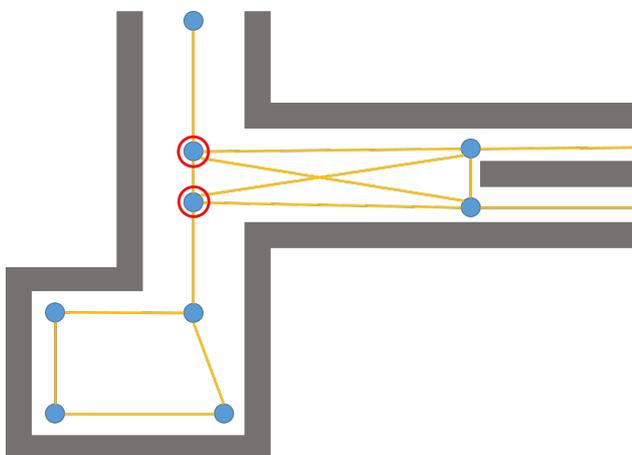


図 4: 歩行エッジと歩行ノードの例。

ている。このとき、現在推定進行方向と歩行エッジとの角度によって各シナリオの尤度を決定する。

- (2) 歩行者が右に向かって直進したと仮定する。最後に方向転換した点から現在までの歩行距離と現在歩行中の歩行エッジの長さを比較する。図 7 (b) に示すように、最後に方向転換した点 (図 7 (b) の黒点) からの歩行距離が、歩行エッジの長さを越えていた場合、そのシナリオの尤度を下げる。図 7 (b) では、シナリオ 2 が歩行エッジの長さを越えているため、尤度を下げる。

- (3) 歩行者が点 X で方向転換したと仮定する。現在歩行中の歩行エッジにおいて閾値よりも近い歩行ノードを転換地点ノードとして選択する。図 7 (c) に示すように、補正した距離によって尤度を変更し、補正した距離が長ければ長いほど尤度が減少するように設定する。
- (4) 図 7 (c) において、シナリオ 2 は尤度が下がり、候補から削除されたとする。そして図 7 (d) に示すように、方向転換の認識から  $x$  歩分歩行したところで (実験では  $x = 5$  としている)、進行方向の変化角度を転換角度とする。方向転換前の進行方向と転換角度をもとに、転換地点ノードから接続するエッジの中から転換後に進むエッジを選択し、その歩行エッジに対応する新たなシナリオを作成する。新たなシナリオの作成の例を図 7 (e) に示す。図 7 (e) では、シナリオ 3 とシナリオ 4 がそれぞれ作成されている。このとき元のシナリオは削除する。新たなシナリオは元のエッジの尤度および、転換角度と歩行エッジの角度の誤差から決定する。
- (5) 歩行中、各シナリオの尤度をチェックし、最も尤度の高いシナリオの現在位置を推定現在位置として採用する。また、尤度の下限を定め、尤度が下限以下になった場合、そのシナリオは削除する。例えば、図 7 (f) では、シナリオ 3 とシナリオ 4 の 2 つのシナリオが存在している。ここで、左に曲がったと仮定すると、シ

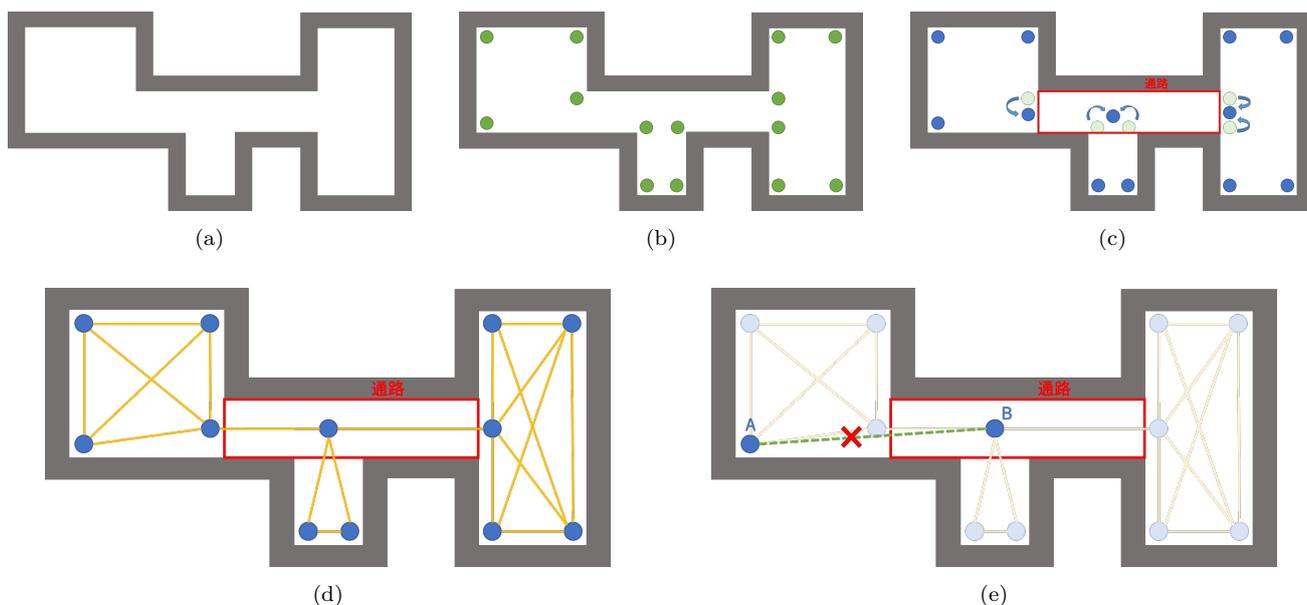


図 5: 可視グラフによる歩行ノードおよび歩行エッジ作成の例.

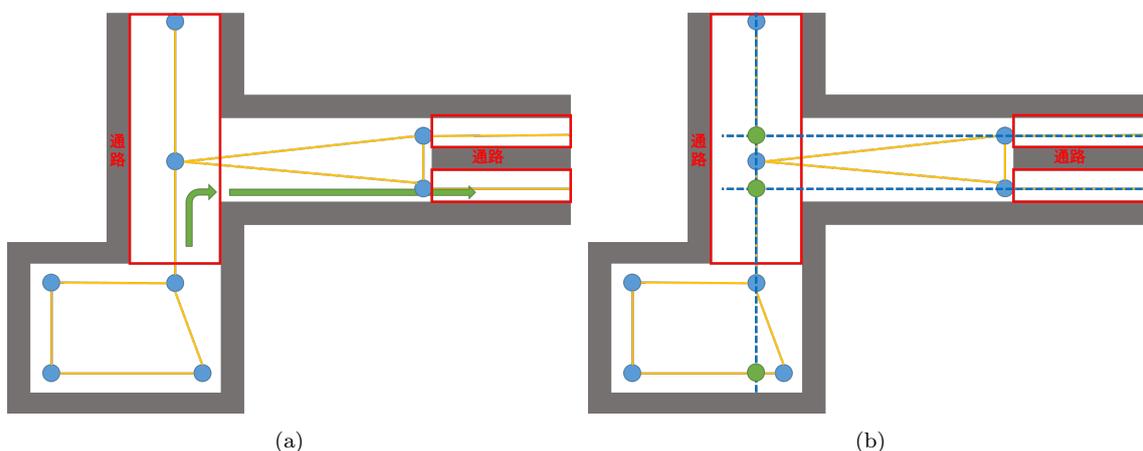


図 6: 可視グラフによる歩行ノード作成時の改良の例.

ナリオ 3 では左に曲がる経路がないため、尤度は低くなる。その結果、図 7 (g) に示すように、シナリオ 4 から生じた、シナリオ 5 のみが保持される。

(6) 手順 (2) から手順 (5) を繰り返す、現在位置を推定する。

尤度  $\phi$  は 0 から 1 までの値をとるものとする。手順 (2) において、歩行距離が歩行エッジの長さを越えていた場合は、1 歩ごとに式 (1) にしたがって尤度を変更する。

$$\phi = \phi - \alpha \quad (1)$$

式 (1) で  $\alpha$  は定数である。手順 (3) において、尤度は式 (2) にしたがって変更する。

$$\phi = \phi \times \frac{L_m - d}{L_m} \quad (2)$$

式 (2) で  $L_m$  は近隣の歩行ノードとして扱う距離の閾値を表し、 $d$  は現在推定位置と選択した歩行ノードの距離で

ある。手順 (4) において、尤度は式 (3) にしたがって変更する。

$$\phi = \phi \times \frac{180 - |\Delta\theta|}{180} \quad (3)$$

式 (3) で  $\Delta\theta$  が転換角度とエッジの角度の誤差である。

#### 4.4 二次現在位置推定 (Step 8)

Step 7 で作成したシナリオのうち、最も尤度の高いシナリオを現在のシナリオとして採用する。採用したシナリオを用いて、Step 6 で求めた推定位置を補正する。補正された位置を最終的な現在推定位置として出力する。

### 5. 評価実験

#### 5.1 実験条件

改良 PDR-PW 手法を Android アプリケーションとして実装し、Google 社のスマートフォン Pixel3 および mobvoi

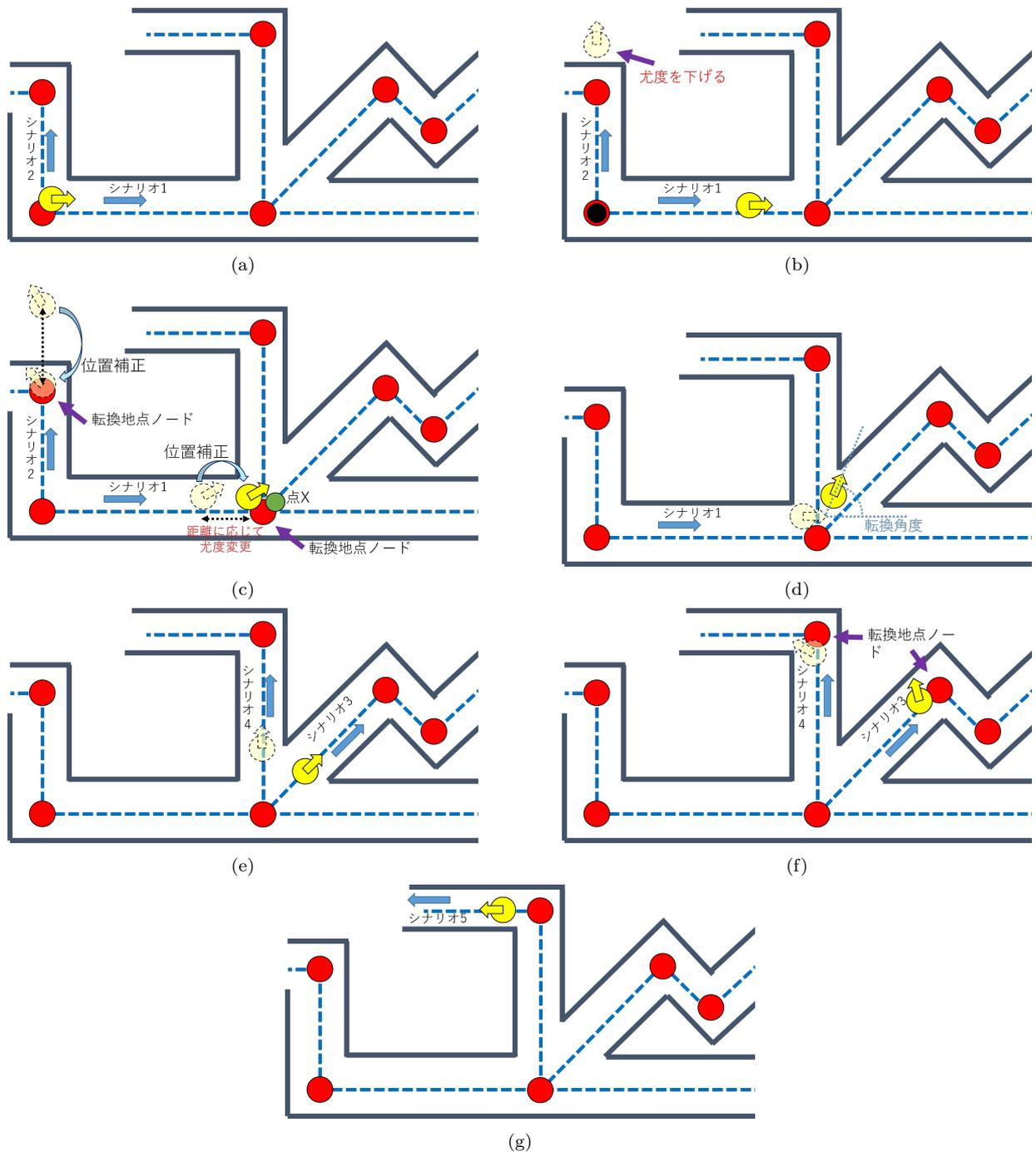


図 7: マルチシナリオマップマッチングの例.

社のスマートウォッチ TicWatch Pro を実機として評価実験を行った。

評価実験では、図 8 の赤線で示す早稲田大学西早稲田キャンパスの約 90m の経路を様々なスマートフォンのモードで歩行し、PDR-PW 手法と改良 PDR-PW 手法それぞれの推定精度を比較した。スマートウォッチは左手首に装着し、左手を振って歩行した。またスマートフォンは身体の前方で手で持っている状態、身体の側面で手に持ち振っている状態、ズボンの右前、左前、右後ろおよび右後ろポケットに入れている状態の 6 つのさまざまなモードによって実

験結果を得ている。屋内地図として同キャンパスの屋内地図を用い、これをもとに改良 PDR-PW 手法では Step 0 で可視グラフを生成している。

図 8 の黄色点で示すようなチェックポイントを用いて、チェックポイント通過時における位置誤差を算出し、それらを平均することで、推定位置誤差を求めた。

## 5.2 実験結果

PDR-PW 手法と改良 PDR-PW 手法それぞれで推定した際の、推定値誤差の結果を表 1 に示す。

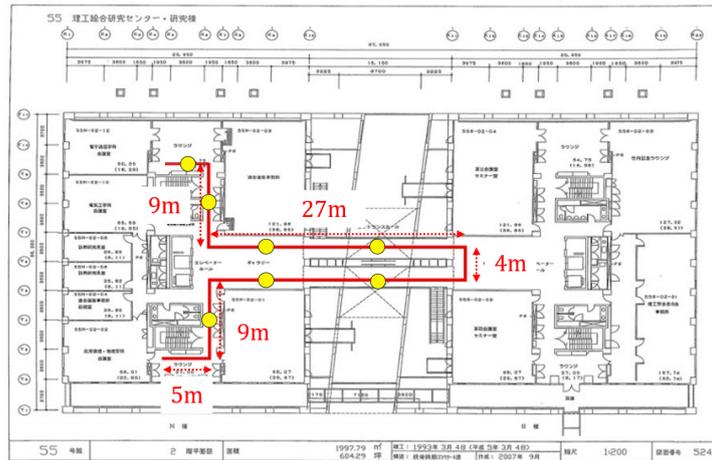


図 8: 実験に用いた歩行ルート。

表 1: 位置推定誤差

	モード	誤差 [cm]							平均位置 誤差 [cm]	誤差 削減率 [%]	
		CP1	CP2	CP3	CP4	CP5	CP6	CP7			全 CP 平均
改良 PDR-PW 手法	手持ち	44.6	309.9	62.1	48.5	208.1	77.3	493.4	177.7	178.6	84.9
	手振り	44.6	309.9	62.1	48.5	208.1	77.3	444.3	170.7		
	右前ポケット	22.1	260.0	4.2	48.5	208.1	347.5	271.9	166.1		
	左前ポケット	22.1	260.0	4.2	18.4	143.2	13.9	361.9	117.7		
	右後ポケット	44.6	309.9	62.1	18.4	143.2	77.3	430.4	155.1		
	左後ポケット	44.6	364.4	127.0	362.1	522.0	77.3	493.4	284.4		
PDR-PW 手法 [10]	手持ち	220.1	531.3	643.4	1108.0	1592.3	1801.7	2072.2	1138.4	1183.1	84.9
	手振り	220.1	531.3	643.4	1108.0	1592.3	1801.7	2112.2	1144.1		
	右前ポケット	196.6	510.4	520.3	889.5	1519.8	1579.5	1844.1	1008.6		
	左前ポケット	176.8	562.7	838.8	1310.4	1981.0	2352.7	2602.4	1403.5		
	右後ポケット	201.9	503.2	779.4	1306.9	1967.3	2284.1	2753.5	1399.5		
	左後ポケット	318.1	851.5	932.0	988.4	1133.6	1196.3	1611.1	1004.4		

表 1 に示す通り、84.9%の位置推定誤差を削減できた。いずれのスマートフォンのモードにおいても位置推定誤差を大きく削減できている。また、改良 PDR-PW 手法では、中間のチェックポイントで誤差が大きくなっても、その後の推定で誤差を補正でき、蓄積誤差が軽減されていることが確認できた。

## 6. おわりに

本稿では、スマートフォンとスマートウォッチを併用した PDR 手法に対し、マップマッチングを適用した、改良 PDR-PW 手法を提案した。改良 PDR-PW 手法では、PDR-PW 手法と比較し平均約 85%の位置誤差を削減した。今後の課題として、高低差を含めた位置推定の考案が考えられる。

## 謝辞

本研究は、一部、JST CREST (Grant No. JPMJCR19K4) の支援を受けたものである。

## 参考文献

- [1] T. Bhattacharya, L. Kulik, and J. Bailey, "Automatically recognizing places of interest from unreliable gps data using spatio-temporal density estimation and line intersections," *Pervasive and Mobile Computing*, vol.19, pp.86–107, 2015.
- [2] Q. Tian, Z. Salcic, K.I. Wang, and Y. Pan, "A multi-mode dead reckoning system for pedestrian tracking using smartphones," *IEEE Sensors Journal*, vol.16, no.7, pp.2079–2093, April 2016.
- [3] L. Xu, Z. Xiong, J. Liu, Z. Wang, and Y. Ding, "A novel pedestrian dead reckoning algorithm for multi-mode recognition based on smartphones," *Remote Sensing*, vol.11, no.3, p.294, 2019.
- [4] S.Y. Park, S.J. Heo, and C.G. Park, "Accelerometer-based smartphone step detection using machine learning technique," *2017 International Electrical Engineering Congress (iEECON)*, pp.1–4, March 2017.
- [5] I. Klein, Y. Solaz, and G. Ohayon, "Pedestrian dead reckoning with smartphone mode recognition," *IEEE Sensors Journal*, vol.18, no.18, pp.7577–7584, 2018.
- [6] 星尚志, 藤井雅弘, 羽多野裕之, 伊藤篤, 渡辺裕他, "スマートフォンを用いた歩行者デッドレコニングのための進行方向推定に関する研究," *情報処理学会論文誌*, vol.57, no.1, pp.25–33, 2016.
- [7] 総務省, "平成 28 年版 情報通信白書 第 1 部,"

- <https://www.soumu.go.jp/johotsusintokei/whitepaper/ja/h28/pdf/n2200000.pdf>.
- [8] mobvio, “Ticwatch pro,” <https://www.mobvoi.com/jp/pages/ticwatchpro>.
  - [9] A. Correa, E. Munoz Diaz, D. Bousdar Ahmed, A. Morell, and J. Lopez Vicario, “Advanced pedestrian positioning system to smartphones and smartwatches,” *Sensors*, vol.16, no.11, p.1903, 2016.
  - [10] W. Tomoya and T. Nozomu, “An indoor positioning method using smartphone and smartwatch independent of carrying modes,” *Proceedings of IEEE 39th International Conference on Consumer Electronics (ICCE)*, 2021.
  - [11] 安齋恵一, 岡島匠吾, 坪川宏他, “スマートフォンを用いた屋内位置の推定と歩行ナビゲーションシステム,” *マルチメディア, 分散協調とモバイルシンポジウム 2011 論文集*, vol.2011, pp.921–927, 2011.
  - [12] S.H. Shin, C.G. Park, and S. Choi, “New map-matching algorithm using virtual track for pedestrian dead reckoning,” *ETRI journal*, vol.32, no.6, pp.891–900, 2010.
  - [13] 岡里篤, 西尾信彦他, “Pdr のオフィスビル向け改良方式の提案,” *研究報告ユビキタスコンピューティングシステム (UBI)*, vol.2021, no.2, pp.1–6, 2021.
  - [14] 石丸智也, 宮村紅葉, 富樫宏謙, 古川浩他, “屋内環境における高精度な測位手法の検討,” *研究報告高度交通システムとスマートコミュニティ (ITS)*, vol.2017, no.13, pp.1–5, 2017.
  - [15] 村田雄哉, 梶克彦, 廣井慧, 河口信夫他, “歩行者自律測位における行動センシング知識の利用,” *マルチメディア, 分散協調とモバイルシンポジウム 2014 論文集*, vol.2014, pp.1614–1619, 2014.
  - [16] Apple, “iphone xr,” <https://www.apple.com/jp/iphone-xr/specs/>.
  - [17] Google, “Google pixel 3a,” [https://store.google.com/jp/product/pixel\\_3a](https://store.google.com/jp/product/pixel_3a).
  - [18] Apple, “Apple watch,” <https://www.apple.com/jp/watch/>.
  - [19] S. Yoshimi, K. Kanagu, M. Mochizuki, K. Muraio, and N. Nishio, “Pdr trajectory estimation using pedestrian-space constraints: Real world evaluations,” *Adjunct Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2015 ACM International Symposium on Wearable Computers*, pp.1499–1508, 2015.
  - [20] 西村天晴, 高山敏典, 柳澤政生, 戸川望他, “動的な歩幅更新をベースとするマップマッチングによる pdr 手法,” *マルチメディア, 分散協調とモバイルシンポジウム 2019 論文集*, vol.2019, pp.1663–1669, 2019.
  - [21] 町田理, 柳澤政生, 戸川望, 新田知之, 進藤大介, 田中清貴他, “歩行通路形状を考慮した可視グラフに基づく屋内環境ナビゲーションシステム,” *組込みシステムシンポジウム 2014 論文集*, vol.2014, pp.96–101, 2014.
  - [22] 岩名地良太, 新田知之, 柳澤政生, 戸川望, “Pdr の測位誤差補正のためのマルチシナリオ化マップマッチング手法 (画像工学),” *電子情報通信学会技術研究報告 = IEICE technical report: 信学技報*, vol.116, no.464, pp.387–392, 2017.