

スマートフォンとスマートウォッチを併用した PDR による屋内位置推定

若泉 朋弥^{1,a)} 戸川 望^{1,b)}

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一般的に GPS (Global Positioning System) を利用して, 自らの現在位置を測位しているが, 屋内や地下では GPS の精度が落ちるため, GPS に代わる位置測位が必要である. PDR (Pedestrian Dead Reckoning) は屋内における現在位置推定手法の一つであり, 歩行者がセンサによって得た加速度や角速度などの歩行データから現在位置を予測する手法である. PDR は外部インフラを利用しないため導入コストが低いという利点がある. 一方で, スマートフォンを用いた PDR では位置測位中に端末の向き (モードと呼ばれる) を固定するのが難しいことや蓄積誤差が問題となる. 端末のモードを考慮した研究は多くなされているが, 端末のモードを 1 つに固定した手法が多く, 実用的とは言えない. 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 手法を提案する. 提案手法は, スマートフォンとスマートウォッチの双方から得られるセンサデータを併用することで, 外部インフラを使用せず, 様々なスマートフォンのモードに対応した PDR を実現する. 同時にドリフト誤差を削減し, 精度の高い位置推定を実現する. 実験の結果, 既存手法と比較して, 位置推定誤差を平均約 87%削減する結果となった.

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 [6]. 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 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 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 [18, 20] に近い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 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 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 手法は現在までも複数提案されている. 特にスマートフォンのモードに着目した研究として以下のものがある. Lee らの研究では閾値を用いたスマートフォンのモードの認識を提案している [11]. Lee らの手法では歩行中の加速度や角速度の 1 歩の平均を求め, 閾値を用いてモードを推定することにより, モードに基づいた進行方向推定を適用する. Tian らの手法では, スマートフォンのモードによってステートマシンを構成し, 角速度をベースとして状態を遷移させることで, スマートフォンのモードに依存した位置推定を実現する [18]. これらの閾値を用いる手法では, 該当モードでの推定精度の向

¹ Waseda University

^{a)} tomoya.wakaizumi@togawa.cs.waseda.ac.jp

^{b)} togawa@togawa.cs.waseda.ac.jp

上が見込めるが、モード変更時の閾値を適切に定める必要がある。

スマートフォンのモード識別として機械学習を用いた研究がある [10, 16, 20]。機械学習を用いた手法では加速度の周期や平均値、分散値などを入力とし、予め定められたモードに分類する。機械学習を用いた手法では仮定したモードについて高精度に分類できる。しかし、仮定したモード以外では推定が難しい。また、学習するためのコストがかかり、多くの歩行者に対して同一のモデルが有効であるかという点は示されていない。いずれの手法も外部インフラを用いていないが、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、スマートフォンのモードを予め仮定せず、様々なスマートフォンのモードで位置推定が可能である [2]。また、外部インフラを用いない推定が可能である。しかし、角速度のみでの進行方向推定ではドリフトによる推定誤差が生じやすく、(条件 a) および (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 [1, 14] 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 の実現が期待される。これまで、Loh らはスマートウォッチとスマートグラスを併用した PDR 手法を提案している [13]。Loh らの手法ではスマートウォッチで加速度と地磁気情報を用いて方位推定を実現し、その上で、スマートグラスを用いて、歩行者の視線を利用した歩幅検知を実現している。Loh らの手法は、上述の (条件 b) は満足するが、方位推定にスマートウォッチのセンサ情報のみを用いるため、ドリフトによる蓄積誤差が生じやすい。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 [8]。Correa らの手法ではスマートフォンおよびスマートウォッチを併用して Wi-Fi の received signal strength を測定し、これをもとにスマートフォンのセンサを用いて比較的精度の高い位置測位を実現している。しかしながら、Correa らの手法では 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。

ここで、スマートフォンのセンサ情報とスマートウォッチのセンサ情報とを併用することを考える。上述のようにスマートウォッチは手首に固定されて装備するため、スマートウォッチのセンサ情報は、モードの影響を受けにくく、周期性を持った値となることが予想される。この値と、スマートフォンのセンサ情報とを併用することで、スマートフォンのモードの影響を受けにくく、精度の良い自己位置の推定が実現できると期待できる。その結果、上述の (条件 a)～(条件 c) を同時に満足することが期待できる。

我々の知る限り、これまでスマートフォンのセンサ情報とスマートウォッチのセンサ情報を併用して、(条件 a)～(条件 c) を同時に満足する PDR 手法は提案されていない。

1.3 本稿の提案

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 手法を提案する。提案する PDR 手法は、

- (1) 外部インフラを利用せず、
- (2) スマートフォンのモードに関わらず位置推定を可能とする。
- (3) この際、進行方向誤差を削減し、PDR の誤差を小さくすることができる。

提案手法では、まず、スマートフォンとスマートウォッチの各端末で、各端末のセンサから得られる加速度および角速度を用いて、歩数推定、端末角推定、方位推定を実行する。その後、スマートフォンとスマートウォッチの同期をとり、両者の方位推定の結果を比較することによって、精度よく進行方向を推定する。最終的に歩数推定の結果と進行方向推定の結果を用いて、現在位置を推定する。その結果、上述の (条件 a)～(条件 c) を満足する PDR を実現する。

1.4 本稿の貢献

本稿の貢献は以下の通りである。

- (1) スマートフォンのモードによらない推定を可能にすることで、ユーザビリティの高い屋内位置推定を実現する。
- (2) 外部インフラを使用しない推定によって、コストの低い屋内位置推定が可能である。
- (3) スマートフォンとスマートウォッチを併用することで、ドリフト誤差を軽減し、精度の高い PDR を実現する。
- (4) 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、既存手法 [2] と比較し、平均約 87% 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 による屋内位置推定問題を定義する。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) から自己位置を推定するものとする。

スマートフォン [5, 9] 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 $[\text{m/s}^2]$ 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

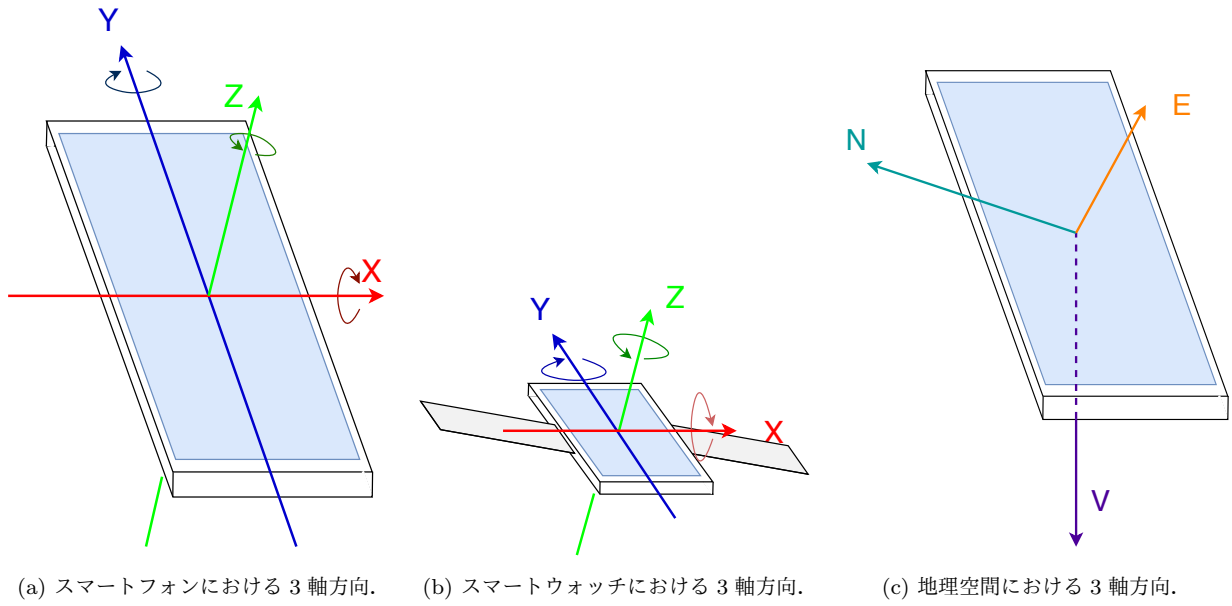


図 1: 3 軸方向の定義.

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. 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 [4, 14] 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 [m/s²] 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 [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.

端末の初期位置 $\vec{p} = (x_0, y_0)$ ならびに初期方位 θ_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, 端末の初期位置 $\vec{p} = (x_0, y_0)$ ならびに初期方位 θ_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.

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

1 章の議論のように, スマートフォンのセンサ情報とスマートウォッチのセンサ情報とを併用することを考える. スマートウォッチは手首に固定されて装備するため, スマートウォッチのセンサ情報は, モードの影響を受けにくく, 周期性を持った値となることが予想される. この値と, スマートフォンのセンサ情報とを併用することで, 外部インフラを用いず, スマートフォンのモードによらない自己位置の推定が実現できると期待できる.

さらにスマートフォンやスマートウォッチをそれぞれ単独で用いる PDR では, 測位時間が増加するにつれ蓄積誤差が大きくなる. 通常, 蓄積誤差を削減するためには Wi-Fi

^{*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 [9, 14].

^{*2} 4 章の評価実験では, $t_i = 50$ ミリ秒とした.

など外部インフラを使用し補正するが [3, 12], 外部インフラを利用すると導入コストが増加する。スマートフォンとスマートウォッチを併用して PDR を実現すれば、蓄積誤差の問題も解決されることが期待できる。

以上の議論のもと、本章では、まず 3.1 節でスマートフォンとスマートウォッチを併用する際の問題点と解決方法についてまとめ、3.2 節で提案手法の流れを示す。3.3 節から 3.8 節で、提案手法の詳細を示す。

3.1 スマートフォンとスマートウォッチによる PDR の問題点とその解決

スマートフォンとスマートウォッチによる PDR では、スマートフォンとスマートウォッチのそれぞれで計測した 3 軸加速度や 3 軸角速度などのセンサ情報を併用し、歩数や進行方向を予測する。スマートフォンとスマートウォッチで計測したセンサ情報を併用するためには、どちらかの端末でセンサ情報を集約する必要があるが、我々の予備実験によれば、スマートフォンとスマートウォッチ相互のデータの送受信には数 $100\mu\text{s}$ 程度の遅延が発生し、そのため、同時刻に得られたデータを一意に結びつけることが困難である。この問題を解決するため、提案手法では、時刻を陽に用いず、歩数をベースとした同期によって解決する (詳細は 3.6 節を参照)。

また、スマートフォンとスマートウォッチのセンサ情報を併用するには、その併用の仕方が問題となる。例えばスマートフォンおよびスマートウォッチのそれぞれで進行方向を予測し、それらの平均値を最終的な進行方向とるように併用する方法が考えられるが、この方法ではスマートフォンおよびスマートウォッチがそれぞれ同じ方向にドリフトしてしまう場合に誤差の低減にならない。提案手法では、スマートフォンとスマートウォッチそれぞれで予測した進行方向に対し共分散をとり、共分散によって角速度をフィルタリングすることでスマートフォンとスマートウォッチのセンサ情報の併用を実現する (詳細は 3.7 節を参照)。これにより直進時と方向転換時を明確に区別することが可能となり、直進時の蓄積誤差の低減が期待される。

さらにスマートフォンのモードによらない推定をするためには、スマートフォンの傾きを検知する必要がある。既存手法としてスマートフォンのセンサ情報から、モードを明確に識別する方法が提案されているが [10, 11], モードの識別では予め用意されたモード以外では、位置推定ができない。提案手法では、スマートフォンの 1 歩の平均加速度を用い、地面の水平面からどの程度スマートフォンが傾いているかを認識することにより、スマートフォンがいかなるモードであっても、進行方向の算出を可能にする (詳細は 3.4 節を参照)。

その結果、提案手法は上述の (条件 a)~(条件 c) を同時にすべて満足することが期待できる。

3.2 提案手法の流れ

以上の議論のもと、提案手法の流れを図 2 に示す。提案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。

以下に提案手法の流れを示す。

(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

提案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。提案手法では、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。提案手法では、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。

以下では Step 1~Step 6 の各ステップを提案する。

3.3 歩数推定 (Step 1)

3.3.1 スマートフォンの場合

スマートフォンで取得した歩行時の加速度を図 3 に示す。図 3 は Google 社のスマートフォン Pixel3 を身体の前で手に持ち直線経路を歩行し、10 秒間のデータを示している。横軸は計測時間、縦軸は加速度の大きさである。図 3 において加速度は式 (1) で求めた時刻 t における合計

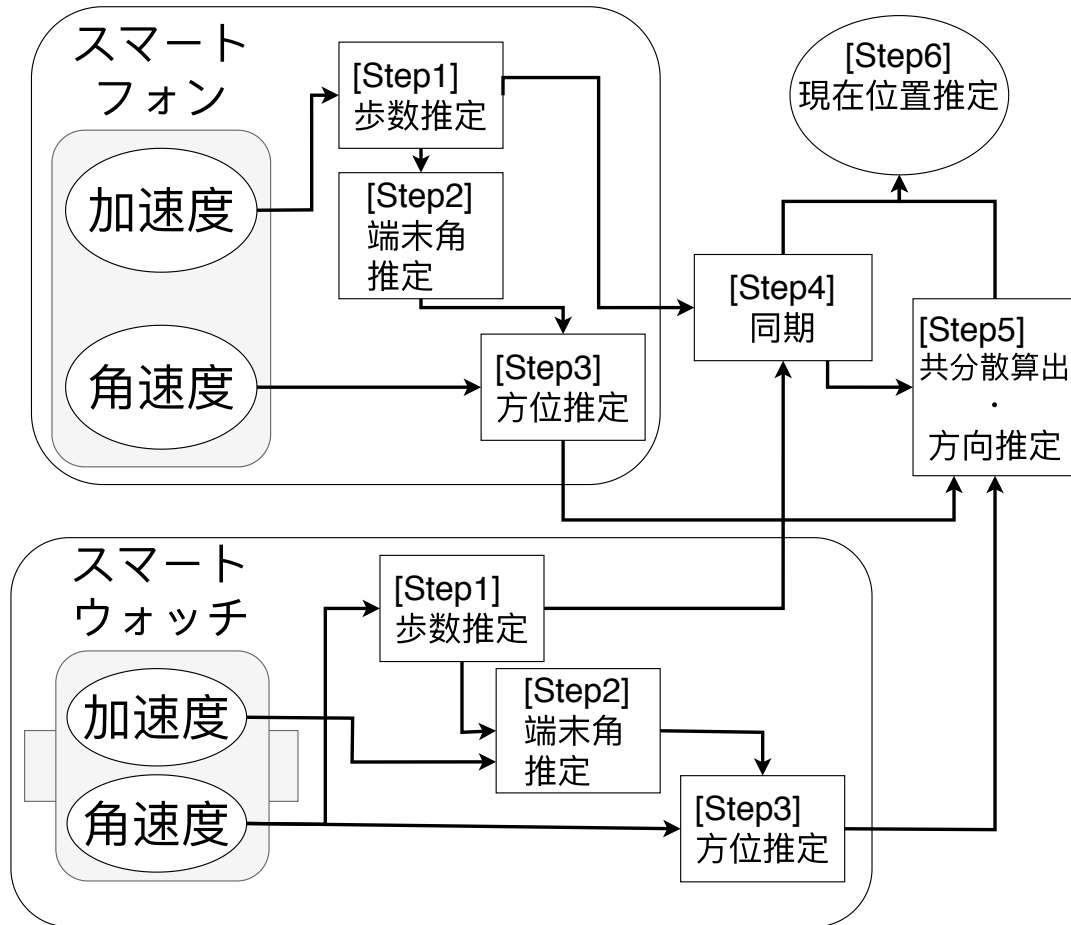


図 2: 提案手法のフローチャート.

加速度 $a_p^{sum}(t)$ である.

$$a_p^{sum}(t) = \sqrt{\bar{a}_p^x(t)^2 + \bar{a}_p^y(t)^2 + \bar{a}_p^z(t)^2} \quad (1)$$

式 (1) において $\bar{a}_p^x(t), \bar{a}_p^y(t), \bar{a}_p^z(t)$ はそれぞれ, スマートフォンで計測した X 軸加速度, Y 軸加速度, Z 軸加速度であり, それぞれの加速度はノイズを除去するため平滑化されている. 平滑化は既存研究 [20] を参考にし, 式 (2) によって実行する.

$$\begin{cases} \bar{a}_p^x(t) = \frac{1}{3} \sum_{k \in \{t^-, t, t^+\}} a_p^x(k) \\ \bar{a}_p^y(t) = \frac{1}{3} \sum_{k \in \{t^-, t, t^+\}} a_p^y(k) \\ \bar{a}_p^z(t) = \frac{1}{3} \sum_{k \in \{t^-, t, t^+\}} a_p^z(k) \end{cases} \quad (2)$$

ここで, t^-, t^+ はそれぞれ, スマートフォンのセンサが時刻 t より 1 回前に加速度を計測した時刻, 1 回後に加速度を計測した時刻を表す.

図 3 に示すように, 歩行中のスマートフォンの加速度は周期的に観測される. 我々の予備実験によれば, 他のスマートフォンのモードであっても, 同様に加速度は周期的に観測された.

そこで, Step 1 では, スマートフォンから取得された X 軸方向, Y 軸方向, Z 軸方向の加速度に対して, これを式 (2) により平滑化し, その後, 式 (1) により合計加速度を算出した上で, 合計加速度のピークを検出することによって, スマートフォンにおける歩数を得ることとする.

3.3.2 スマートウォッチの場合

スマートウォッチで取得した歩行中の加速度および角速度を図 4 および図 5 に示す. 図 4 および図 5 は mobvol 社のスマートウォッチ TicWatch Pro を使用して直線経路を歩行し, 10 秒間のデータを示している. 図 4 の横軸は計測時間, 縦軸は加速度の大きさである. ここで加速度は, 前述の通り, まず X 軸加速度, Y 軸加速度, Z 軸加速度を平滑化し, その後, 合計加速度を算出することで得ている. 図 5 の横軸は計測時間, 縦軸は角速度である. ここで角速度は, 同様に, まず X 軸角速度, Y 軸角速度, Z 軸角速度を平滑化し, その後, 合計角速度を算出することで得ている.

図 4 のように, スマートウォッチの加速度は周期的に変化するものの, 図 4 中の赤点の部分のように, 一部加速度の極大値と極小値の差が, 小さくなっている部分がある. そのためこのような部分では極小値と極大値のどちらかを検出できない可能性がある. 一方で図 5 のようにスマートウォッチの角速度は歩行に対して周期的である. つまり,

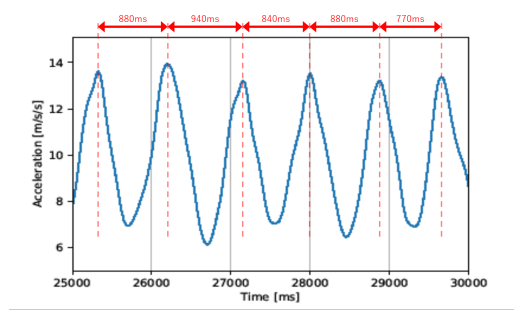


図 3: 歩行時のスマートフォンにおける加速度.

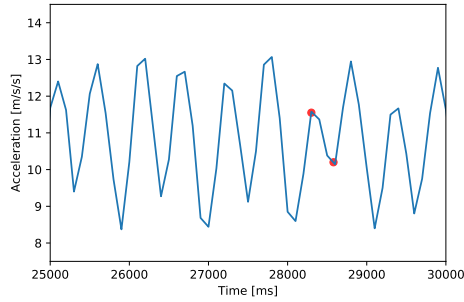


図 4: 歩行時のスマートウォッチにおける加速度.

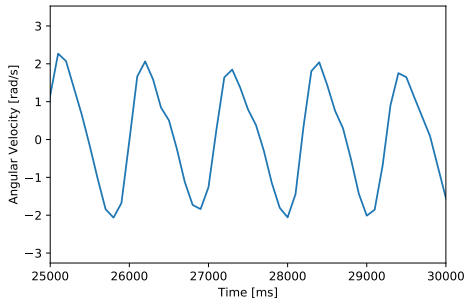


図 5: 歩行時のスマートウォッチにおける角速度.

スマートウォッチでは加速度よりも角速度がより顕著にピークをとらえることができる.

そこで、スマートフォンと同様に、スマートウォッチでは合計角速度のピークを検出することで歩数を得る.

3.4 端末角推定 (Step 2)

スマートフォンやスマートウォッチで取得される 3 軸加速度および 3 軸角速度は図 1(a) のように、端末を基準とした 3 軸で取得される. しかし位置推定をするにあたり、3 軸加速度および 3 軸角速度を図 1(c) のように北方向、東方向、重力方向の 3 軸で構成される地理空間に変換する必要がある.

3.4.1 スマートフォンの場合

提案手法では、スマートフォンの様々なモードに対応するため、合計加速度を利用することで端末角を推定する.

図 6 に歩行者の 1 歩の動きと加速度を示す. 赤色の矢

印で描かれているのが歩行者の加速度の例, 緑色の矢印で描かれているのが歩行者に掛かる重力加速度である. 歩行中は地面を足でけることにより, 図 6 の赤矢印のように上下・前後に加速度が発生する. ここで歩行者が等速で歩行していると仮定すると, 重力を除く加速度の和は 0 であると仮定できる. すなわち歩行中には上下・前後に加速度が発生するが, それらは一定の時間で観測すれば相殺される. 歩行の特性から 1 歩ごとに上下・前後の加速度は周期的であるため, 1 歩の加速度の平均をとれば地面と水平方向の加速度は 0 になる. 一方重力加速度は図 6 の緑矢印のように常に地面に垂直下向きに歩行者にかかる.

以上をまとめると, 等速に歩行していると仮定した場合, 1 歩当たりの水平方向加速度の和は 0 であり, 1 歩あたりの合計加速度の平均値を求めると重力加速度のみが観測される. 1 歩あたりの合計加速度の平均値から重力ベクトルを得ることができれば, 重力ベクトルとスマートフォンの X 軸, Y 軸, Z 軸との角度を求めることができる.

n 歩目におけるスマートフォンの推定重力ベクトルを $\vec{g}_p(n)$ とする. その大きさ $|\vec{g}_p(n)|$ は加速度 $\vec{a}_p(t) = (a_p^x(t), a_p^y(t), a_p^z(t))$ を用いて式 (3) で求められる.

$$|\vec{g}_p(n)| = \frac{1}{|\mathbb{T}_p(n)|} \sum_{t \in \mathbb{T}_p(n)} \sqrt{(a_p^x(t))^2 + (a_p^y(t))^2 + (a_p^z(t))^2} \quad (3)$$

式 (3) において $\mathbb{T}_p(n)$ は n 歩目に含まれるスマートフォンの計測時刻の集合である.

ここで n 歩目におけるスマートフォンの X 軸, Y 軸, Z 軸の平均加速度ベクトルをそれぞれ $\vec{a}_{p,ave}^x(n)$, $\vec{a}_{p,ave}^y(n)$, $\vec{a}_{p,ave}^z(n)$ とすると, スマートフォンの推定重力ベクトル $\vec{g}_p(n)$ は式 (4) で表される.

$$\vec{g}_p(n) = \vec{a}_{p,ave}^x(n) + \vec{a}_{p,ave}^y(n) + \vec{a}_{p,ave}^z(n) \quad (4)$$

n 歩目におけるスマートフォンの重力ベクトルと X 軸, Y 軸, Z 軸との角度を $\vec{\Omega}_p(n) = (\Omega_p^x(n), \Omega_p^y(n), \Omega_p^z(n))$ とすると, $\vec{a}_{p,ave}^x(n)$, $\vec{a}_{p,ave}^y(n)$, $\vec{a}_{p,ave}^z(n)$ と $\vec{g}_p(n)$ の内積は式 (5) で表される.

$$\begin{cases} \vec{a}_{p,ave}^x(n) \cdot \vec{g}_p(n) = |\vec{a}_{p,ave}^x(n)| \cdot |\vec{g}_p(n)| \cdot \cos \Omega_p^x(n) \\ \vec{a}_{p,ave}^y(n) \cdot \vec{g}_p(n) = |\vec{a}_{p,ave}^y(n)| \cdot |\vec{g}_p(n)| \cdot \cos \Omega_p^y(n) \\ \vec{a}_{p,ave}^z(n) \cdot \vec{g}_p(n) = |\vec{a}_{p,ave}^z(n)| \cdot |\vec{g}_p(n)| \cdot \cos \Omega_p^z(n) \end{cases} \quad (5)$$

ここで, 式 (5) に式 (4) を代入し, $\vec{a}_{p,ave}^x(n)$, $\vec{a}_{p,ave}^y(n)$, $\vec{a}_{p,ave}^z(n)$ は互いに直交していることに注意すれば, n 歩目における重力ベクトルと X 軸, Y 軸, Z 軸との角度 $\vec{\Omega}_p(n) = (\Omega_p^x(n), \Omega_p^y(n), \Omega_p^z(n))$ は $\vec{g}_p(n)$ を用いて式 (6) で求められる.

$$\begin{cases} \Omega_p^x(n) = \arccos \frac{|\vec{a}_{p,ave}^x(n)|}{|\vec{g}_p(n)|} \\ \Omega_p^y(n) = \arccos \frac{|\vec{a}_{p,ave}^y(n)|}{|\vec{g}_p(n)|} \\ \Omega_p^z(n) = \arccos \frac{|\vec{a}_{p,ave}^z(n)|}{|\vec{g}_p(n)|} \end{cases} \quad (6)$$

以上の操作により、スマートフォンが任意の向きであっても、スマートフォンの傾きを求めることができる。これにより、スマートフォンがいかなるモードであっても、スマートフォンの傾きから加速度や角速度を地理空間の水平成分と垂直成分に変換可能であり、(条件 b) を満たすことができる。

3.4.2 スマートウォッチの場合

スマートウォッチではモードが限られているため、手首にあると仮定することが可能である。しかし、腕の振り方が個人で異なるほか、腕が地面と垂直な面で振られているとは限らない。そのため、スマートウォッチについてもスマートフォンと同様の操作により、端末角を求める。\$n\$ 歩目におけるスマートウォッチの推定重力ベクトルを \$\vec{g}_w(n)\$ とする。その大きさ \$|\vec{g}_w(n)|\$ は加速度 \$\vec{a}_w(t) = (a_w^x(t), a_w^y(t), a_w^z(t))\$ を用いて式 (7) で求められる。

$$|\vec{g}_w(n)| = \frac{1}{|\mathbb{T}_w(n)|} \sum_{t \in \mathbb{T}_w(n)} \sqrt{(a_w^x(t))^2 + (a_w^y(t))^2 + (a_w^z(t))^2} \quad (7)$$

式 (7) において \$\mathbb{T}_w(n)\$ は \$n\$ 歩目に含まれるスマートウォッチの計測時刻の集合である。

ここで \$n\$ 歩目におけるスマートウォッチの X 軸, Y 軸, Z 軸の平均加速度ベクトルをそれぞれ \$\vec{a}_{w,ave}^x(n)\$, \$\vec{a}_{w,ave}^y(n)\$, \$\vec{a}_{w,ave}^z(n)\$ とすると、スマートウォッチの推定重力ベクトル \$\vec{g}_w(n)\$ は式 (8) で表される。

$$\vec{g}_w(n) = \vec{a}_{w,ave}^x(n) + \vec{a}_{w,ave}^y(n) + \vec{a}_{w,ave}^z(n) \quad (8)$$

\$n\$ 歩目におけるスマートウォッチの重力ベクトルと X 軸, Y 軸, Z 軸との角度を \$\Omega_w(n) = (\Omega_w^x(n), \Omega_w^y(n), \Omega_w^z(n))\$ とすると、\$\vec{a}_{w,ave}^x(n)\$, \$\vec{a}_{w,ave}^y(n)\$, \$\vec{a}_{w,ave}^z(n)\$ と \$\vec{g}_w(n)\$ の内積は式 (9) で表される。

$$\begin{cases} \vec{a}_{w,ave}^x(n) \cdot \vec{g}_w(n) = |\vec{a}_{w,ave}^x(n)| \cdot |\vec{g}_w(n)| \cdot \cos \Omega_w^x(n) \\ \vec{a}_{w,ave}^y(n) \cdot \vec{g}_w(n) = |\vec{a}_{w,ave}^y(n)| \cdot |\vec{g}_w(n)| \cdot \cos \Omega_w^y(n) \\ \vec{a}_{w,ave}^z(n) \cdot \vec{g}_w(n) = |\vec{a}_{w,ave}^z(n)| \cdot |\vec{g}_w(n)| \cdot \cos \Omega_w^z(n) \end{cases} \quad (9)$$

ここで、式 (9) に式 (8) を代入し、\$\vec{a}_{w,ave}^x(n)\$, \$\vec{a}_{w,ave}^y(n)\$, \$\vec{a}_{w,ave}^z(n)\$ は互いに直交していることに注意すれば、\$n\$ 歩目における重力ベクトルと X 軸, Y 軸, Z 軸との角度 \$\Omega_w(n) = (\Omega_w^x(n), \Omega_w^y(n), \Omega_w^z(n))\$ は \$\vec{g}_w(n)\$ を用いて式 (10) で求められる。

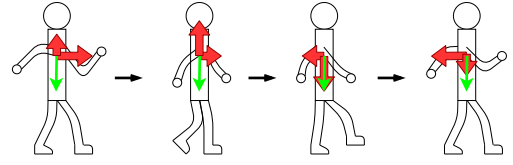


図 6: 歩行者にかかる加速度。

$$\begin{cases} \Omega_w^x(n) = \arccos \frac{|\vec{a}_{w,ave}^x(n)|}{|\vec{g}_w(n)|} \\ \Omega_w^y(n) = \arccos \frac{|\vec{a}_{w,ave}^y(n)|}{|\vec{g}_w(n)|} \\ \Omega_w^z(n) = \arccos \frac{|\vec{a}_{w,ave}^z(n)|}{|\vec{g}_w(n)|} \end{cases} \quad (10)$$

3.5 方位推定 (Step 3)

スマートフォンを用いた PDR での方位推定では、地磁気を使った手法が多く提案されている [19]。しかし、地磁気は建物や電子機器からの影響を受けやすく、不安定である [11]。よって提案手法ではスマートフォンおよびスマートウォッチの双方で角速度を利用した相対的な方位推定を考える。

3.5.1 スマートフォンの場合

スマートフォンの方位推定では、水平方向の角速度を使用する。時刻 \$t\$ におけるスマートフォンの水平角速度を \$\omega_p^{hol}(t)\$ とする。いま、\$i\$ 歩目を考えると、重力ベクトルとスマートフォンの X 軸, Y 軸, Z 軸との角度は、\$\Omega_p^x(i)\$, \$\Omega_p^y(i)\$, \$\Omega_p^z(i)\$ で与えられる。このとき \$\omega_p^{hol}(t)\$ は、式 (11) で記述できる。

$$\begin{aligned} \omega_p^{hol}(t) = & \omega_p^x(t) \times \cos \Omega_p^x(i) \\ & + \omega_p^y(t) \times \cos \Omega_p^y(i) + \omega_p^z(t) \times \cos \Omega_p^z(i) \end{aligned} \quad (11)$$

スマートフォンで推定する \$i\$ 歩目の進行方向変化 \$\Delta \theta_p(i)\$ は、\$i\$ 歩目に相当する時刻の水平角速度データを積分することで求められる。スマートフォンの \$i\$ 歩目の開始時刻を \$t_p^{start}(i)\$, 終了時刻を \$t_p^{end}(i)\$ データの取得間隔を \$\Delta t_p\$ とすると式 (14) で算出される。

$$\Delta \theta_p(i) = \sum_{k=t_p^{start}(i)}^{t_p^{end}(i)} (\omega_p^{hol}(k) \times \Delta t_p) \quad (12)$$

3.5.2 スマートウォッチの場合

スマートウォッチの進行方向変化もスマートフォンと同様に求められる。時刻 \$t\$ におけるスマートウォッチの水平角速度を \$\omega_w^{hol}(t)\$ とする。いま、\$i\$ 歩目を考えると、重力ベクトルとスマートフォンの X 軸, Y 軸, Z 軸との角度は、\$\Omega_w^x(i)\$, \$\Omega_w^y(i)\$, \$\Omega_w^z(i)\$ で与えられる。このとき \$\omega_w^{hol}(t)\$ は、式 (13) で記述できる。

$$\begin{aligned}\omega_w^{hol}(t) = & \omega_w^x(t) \times \cos \Omega_w^x(i) \\ & + \omega_w^y(t) \times \cos \Omega_w^y(i) + \omega_w^z(t) \times \cos \Omega_w^z(i)\end{aligned}\quad (13)$$

スマートウォッチで推定する i 歩目の進行方向変化 $\Delta\theta_w(i)$ は、スマートウォッチの i 歩目の開始時刻を $t_w^{start}(i)$ 、終了時刻を $t_w^{end}(i)$ データの取得間隔を Δt_w とすると式 (14) で算出される。

$$\Delta\theta_w(i) = \sum_{k=t_w^{start}(i)}^{t_w^{end}(i)} (\omega_w^{hol}(k) \times \Delta t_w) \quad (14)$$

3.6 同期 (Step 4)

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。図 7 にスマートフォンとスマートウォッチでの送受信の遅延時間を示す。図 7 は以下のように描画されている。Google 社のスマートフォン Pixel3 および mobvol 社のスマートウォッチ TicWatch Pro を実機として、

- (1) スマートフォンからスマートウォッチとを bluetooth によって接続し、スマートフォンからスマートウォッチに 16 バイトの文字列 A を発信し (時刻 t_A)、
- (2) スマートフォンから発信された文字列 A をスマートウォッチで受け取った後、スマートウォッチからスマートフォンに 16 バイトの文字列 B を発信し、
- (3) スマートウォッチから発信された文字列 B をスマートフォンで受け取る (時刻 t_B)、
- (4) この時、時刻 t_A において、 $\frac{t_B - t_A}{2}$ をスマートウォッチ・スマートフォン間の遅延時間とする。
- (5) (1)~(4) を 500ms ごとに繰り返し、得られた遅延時間をプロットしたものが図 7 である。図 7 は横軸が時刻、縦軸が遅延時間である。

スマートフォンとスマートウォッチは、必ずしも同時刻にセンサデータを取得するとは限らず、また図 7 のようにスマートフォンとスマートウォッチ間の遅延は必ずしも一定でない。スマートフォンとスマートウォッチで完全に同時刻に得られたデータを取得するのは困難である。

一方で歩行周期は図 3 のように、およそ 700ms から 1000ms である。提案手法ではスマートフォンとスマートウォッチ間の遅延の大きさが歩行周期よりも短いことに注目し、スマートフォンおよびスマートウォッチで Step 1 で求めた歩数を用いて同期をとることでスマートフォンとスマートウォッチ間の遅延の問題を解決する。つまり、PDR を開始した時点基準に、スマートフォンで検出した歩数を n 、スマートウォッチで検出した歩数を m としたとき、 $n = m$ となるときを同じ時刻として、そのときに双方で推定した方位を使って Step 5 を実行する。

歩数を利用して同期をとる場合、どちらかの端末で歩行の検知に漏れがあると時刻がずれてしまうことが予想さ

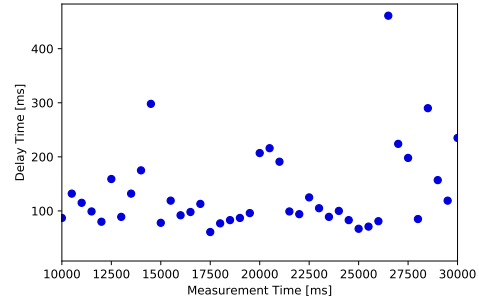


図 7: スマートフォンとスマートウォッチの送受信遅延時間。

れる。そのため、両端末間で歩数のずれが生じた場合は補正する必要がある。前述したように、スマートフォンとスマートウォッチ間の遅延時間は歩行周期に比較して十分に小さく、スマートフォンで検知された歩数とスマートウォッチで検知された歩数が 2 歩以上ずれを生じた場合、どちらかの端末で多く歩数を検出もしくは検出漏れが生じていることがわかる。ここで歩数検出は 4.2 節の実験結果から検出漏れが生じる可能性が高いため、スマートフォンによりスマートウォッチで計測した歩数を受信した際に、スマートフォンとスマートウォッチの歩数を比較し、2 歩以上の差が生じていた場合は、歩数を少なく計測している端末の歩数を多く計測している端末の歩数に合わせることで、以降の計測において同期をとる際にずれが蓄積しないようにする。

3.7 共分散算出・方向推定 (Step 5)

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 [7, 15]。蓄積誤差の問題に対する解決方法として、外部インフラを活用する等、PDR 以外の位置情報を併用し誤差を補正する手法が提案されている。しかし PDR 以外の位置情報を利用する場合、導入コストが大きい。提案手法では、スマートフォンとスマートウォッチの双方で推定された方位を利用することで、外部インフラを用いることなく、誤差を補正する。

3.7.1 共分散算出

スマートフォンの角速度センサとスマートウォッチの角速度センサの 2 つの角速度センサを用いるとき、歩行者が右折あるいは左折すれば、2 つの角速度センサも同じようなデータを取得することが期待される。一方、スマートフォンやスマートウォッチの角速度は歩行者の動きや熱誤差等のノイズによって歩行者の方向転換時以外でも変動しているが、2 つの角速度センサに同じノイズが加わり、両者が同じ波形になるのは稀である。

そこで、2 つの角速度センサのデータがあるとき、両者の共分散 [17] をとることにする。共分散は、2 つのデータが同様な変化をするときに大きな値を示し、さらに元のデー

タの値が大きい場合に、共分散の値もより大きな値を示すという性質を持つ。そのため、2つの角速度センサのデータが同様の変化をし、さらにその変化が大きいことを認識することが可能となり、その結果、角速度センサに加わったノイズを除去し、歩行者の方向転換を正確に認識することができる。つまり、提案手法では、まずスマートフォンとスマートウォッチで計測された角速度の共分散を求め、次に、共分散が正かつ大きい場合のみ歩行者の方向転換を判断することによって、直進時の角速度の不要な変動を防ぎドリフト誤差を小さく抑える。

i 歩目のスマートフォンおよびスマートウォッチの角速度の共分散 s_i を式 (15) で定義する。

$$s_i = \frac{1}{3} \sum_{k=i-1}^{i+1} ((\Delta\theta_p(k) - \Delta\theta_p^{ave}) \times (\Delta\theta_w(k) - \Delta\theta_w^{ave})) \quad (15)$$

式 (15) で $\Delta\theta_p(i)$, $\Delta\theta_w(i)$ はそれぞれ Step 3 で得た i 歩目のスマートフォンの角度変化, i 歩目のスマートウォッチの角度変化である。また, $\Delta\theta_p^{ave}$, $\Delta\theta_w^{ave}$ はそれぞれ $i-1$ 歩目から $i+1$ 歩目までのスマートフォンの角度変化の平均値, $i-1$ 歩目から $i+1$ 歩目までのスマートウォッチの角度変化の平均値である。ここで3歩分の共分散を算出しているのは、リアルタイムな測位のためには長い歩数の共分散を算出することは現実的でないため、平均の算出を短い時間区間に絞る必要があるからである。また、人間の歩行は右足と左足をそれぞれ前に出すため2歩で一つの周期になり、2歩で共分散を算出する場合、この周期によって方向転換時以外であっても共分散が大きくなってしまいうことを防ぐために2歩で共分散を算出するのは好ましくない。よって提案手法では3歩分の共分散を算出している。

3.7.2 共分散を使った方向推定

補正した i 歩目の角度変化 $\Delta\theta_{cor}(i)$ は、 i 歩目のスマートフォンおよびスマートウォッチの角速度の共分散 s_i によって式 (16) で求められる。

$$\Delta\theta_{cor}(i) = \begin{cases} \Delta\theta(i) \times a_{pass} & (\text{either } s_i > s_{th}, s_{i\pm 1} > s_{th}, \\ & \text{or } s_{i\pm 2} > s_{th} \text{ is satisfied}) \\ \Delta\theta(i) \times a_{cut} & (\text{otherwise}) \end{cases} \quad (16)$$

方向転換時には一歩で曲がることは少なく、数歩かかることを考慮し、共分散の大きかった部分から前後2歩も角速度を受け付けることとする。ここで a_{pass} , a_{cut} はそれぞれ方向転換時、直進時に角速度をどの程度受け付けるかを定める定数であり、 s_{th} は方向転換時を判断する共分散の閾値である (閾値の設定については3.7.3項を参照)。 $\Delta\theta(i)$ はスマートフォンの i 歩目の進行方向変化 $\Delta\theta_p(i)$ とスマートウォッチの i 歩目の進行方向変化 $\Delta\theta_w(i)$ の平均

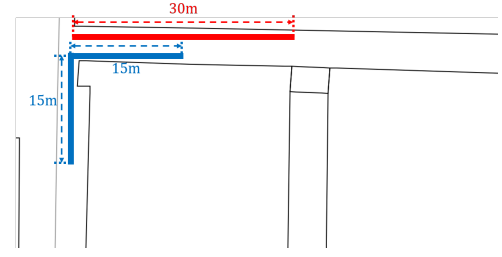


図 8: 共分散測定に用いた歩行ルート。

値で式 (17) で求められる。

$$\Delta\theta(i) = \frac{1}{2}(\Delta\theta_p(i) + \Delta\theta_w(i)) \quad (17)$$

初期方位 θ_0 から n 歩目までの相対的な方位差 θ_n は i 歩目の進行方向変化 $\Delta\theta_{cor}(i)$ を用いて式 (18) で与えられる。

$$\theta_n = \sum_{i=0}^n \Delta\theta_{cor}(i) \quad (18)$$

3.7.3 閾値の設定

式 (16) において、いかに閾値 s_{th} を設定するかが重要となる。Google 社のスマートフォン Pixel3 および mobvol 社のスマートウォッチ TicWatch Pro を用いて図 8 の赤線のように約 30m の直線経路を歩行をしたときの共分散を図 9 に示す。また、図 8 の青線のように約 15m 直進した後、左方向に 90° 方向転換をし、さらに約 15m 直進した時の共分散を図 10 に示す。図 9, 図 10 では、身体の前方で手で持っている状態 (Hand)、身体の側面で手に持ち振っている状態 (Swing)、ズボンの右前ポケットに入れている状態 (FrontPocket)、ズボンの右後ろポケットに入れている状態 (BackPocket) の4つのスマートフォンのモードで計測した。スマートウォッチは左手首に装着している。また、図 9 および図 10 のどちらも、縦軸が共分散、横軸は歩数である。

図 9 から、直線時の共分散は概ね 15 以下であった。一方図 10 から、方向転換時はいずれのモードであっても共分散が 15 を越える値を示した。よって式 (16) において共分散の閾値 s_{th} を 15 に設定する。また、方向転換時には計測した角速度を進行方向の変化とし、直進時には進行方向を変化させないようにするため、 $a_{pass} = 1$, $a_{cut} = 0$ とした。

3.8 現在位置測位 (Step 6)

Step 1 および Step 4 において求めた歩数 n と、Step 5 において求めた i 歩目の進行方位 θ_i から、 i 歩目の現在位置 $\vec{p}_i = (x_i, y_i)$ を推定する。現在位置は式 (19) で求められる。ここで歩幅 l は既知であるとする。

$$\begin{cases} x_n = l \times \cos \theta_n + x_{n-1} \\ y_n = l \times \sin \theta_n + y_{n-1} \end{cases} \quad (19)$$

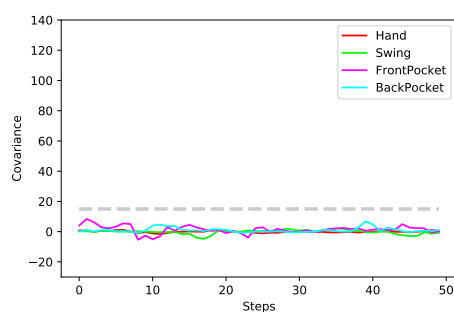


図 9: 直線歩行時における共分散

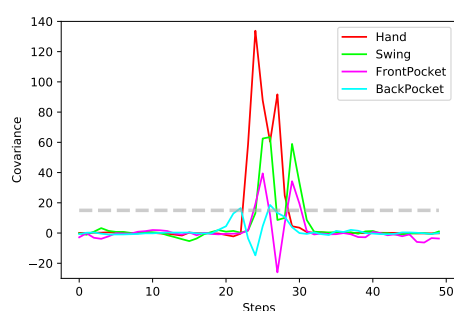


図 10: 方向転換時における共分散

4. 評価実験

4.1 実験条件

提案手法を 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 mobvol 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。図 11 に示す早稲田大学西早稲田キャンパスの約 70m の直線経路および約 100m の L 字型経路を様々なスマートフォンのモードで歩行し、3 種類の実験を行った。いずれの実験においても、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 4.2 節～4.4 節に記述するようにさまざまなモードによって実験結果を得ている。以下に実験の種類と目的を示す。

(1) 歩数推定実験

様々なスマートフォンのモードにおいて、直線経路および L 字型経路を歩行し、実際の歩数と推定歩数を比較することにより、歩数推定 (Step 1) の精度を確かめる。

(2) 方向推定実験

様々なスマートフォンのモードにおいて、直線経路および L 字型経路を歩行し経路を推定する。スマートフォンのみ、スマートウォッチのみを使用して経路を推定した場合と比較することにより、方向転換が認識されているか、ドリフト誤差を軽減できているかを確認する。

(3) 比較実験

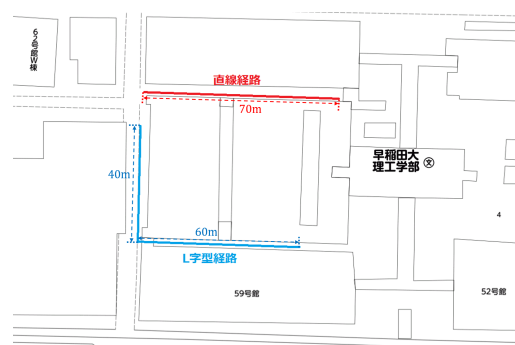


図 11: 実験に用いた歩行ルート。

1 章で記述したように、スマートフォンとスマートウォッチを併用した既存手法はこれまで提案されていない。一方、既存手法 [2] は、他の既存手法が予め定められたスマートフォンのモードを仮定して位置推定を実施するのに対して、スマートフォンのモードを手で持っている状態やズボンのポケット、鞆の中など任意のスマートフォンのモードでの位置推定が可能であり、提案手法と同様に地磁気を用いず角速度のみでの進行方向推定を提案している。

そこで様々なスマートフォンのモードにおいて、直線経路及び L 字型経路を歩行し、既存手法 [2] と比較することにより、提案手法の有効性を評価する。

4.2 歩数推定実験

提案手法による歩数推定の結果を表 1 に示す。提案手法の Step 1 によって、スマートフォンならびにスマートウォッチの歩数推定の結果を表 1 に示す。歩数推定実験では、身体の前方で手で持っている状態 (手持ち)、身体の側面で手に持ち振っている状態 (手振り)、ズボンの右前ポケット (前ポケット)、右後ろポケットに入れている状態 (後ろポケット)、肩掛け鞆の中に入れている状態の 5 つのスマートフォンのモードおよびスマートウォッチを左腕に装着した状態で歩行した。いずれの端末状態においても相対誤差が 1% 前後であり、比較的精度を高く歩数を推定できていると考える。

誤差の原因として、歩き始めや歩き終わりのピークが正しく取得できていないことが考えられる。

4.3 方向推定実験

提案手法による位置推定結果と、スマートフォンまたはスマートウォッチ単独での位置推定結果を図 12 および図 13 に示す。図 12 は図 11 の直線経路、図 13 は図 11 の L 字型経路において、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、ショルダーバッグの中に入れている状態の 7 つのスマートフォンのモードで歩行した推定結果である。図 12 は直

表 1: 歩数推定実験の結果結果.

端末		直進 (100 歩)		L 字型 (150 歩)	
		推定歩数	誤差	推定歩数	誤差
スマートフォンの 端末モード	手持ち	101	1.0%	151	0.7%
	手振り	100	0.0%	149	-0.7%
	前ポケット	98	-2.0%	148	-1.3%
	後ポケット	96	-4.0%	148	-1.3%
	肩掛け鞆	100	0.0%	150	0.0%
スマートウォッチ		100	0.0%	149	-0.7%

線経路の進行方向を X 軸, 垂直方向を Y 軸として, 提案手法による推定経路 (Proposal), スマートフォンのみでの推定経路 (Phone Only), スマートウォッチのみでの推定経路 (Watch Only), 実際の経路 (Actual) を, どちらの軸も cm 単位でプロットしたものである. 図 13 は L 字型経路の歩き始めの一辺を X 軸, 方向転換後の一辺を Y 軸として, 図 12 と同様に提案手法による推定経路 (Proposal), スマートフォンのみでの推定経路 (Phone Only), スマートウォッチのみでの推定経路 (Watch Only), 実際の経路 (Actual) を, どちらの軸も cm 単位でプロットしたものである.

図 12 に示す通り, 提案手法では, スマートフォンとスマートウォッチを併用することにより, どのモードであってもほぼ誤差なく直線経路を認識し, 自己位置を推定している. 一方, 図 13 に示す通り, L 字型経路では, 図 13(a) や図 13(b), 図 13(d) のように, ほぼ誤差なく実経路と同じ経路を推定しているモードや, 図 13(c) や図 13(e)~(g) のように, 実経路に比較して誤差を持っているモードもある. いずれにしても, どのスマートフォンのモードにおいても, スマートフォンまたはスマートウォッチ単独で推定する場合よりも, ドリフト誤差が抑制され, 高精度に経路が推定できている. なお, 図 13(c) や図 13(e)~(g) のように, 誤差を持っているモードであっても, 従来手法と比較してドリフト誤差は小さく, 例えば, マップマッチング等と組み合わせることにより, より正確な自己位置測定が可能となると期待できる.

4.4 比較実験

提案手法と既存手法 [2] の推定位置誤差を表 2 に示す. 表 2 における「直線経路」は, 図 11 に示した直線経路を, 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, ショルダーバッグの中に入れている状態の 7 つのスマートフォンのモードで歩行し, 提案手法と既存手法 [2] の推定位置誤差を求めた結果である. 表 2 における「L 字型経路」は, 図 11 に示した L 字型経路を直線経路同様に 7 つのスマートフォンのモードで歩行し, 提案手法と既存手法の推定位置誤差を求めた結果

である.

ここで, 推定位置誤差 err とは式 (20) で与えられるもので, 推定位置と実際の位置との差を 1 歩ごとに求め, 累積したものを総歩数 n で割った値である.

$$err = \frac{1}{n} \sum_{i=0}^n \sqrt{(x_i^{est} - x_i^{act})^2 + (y_i^{est} - y_i^{act})^2} \quad (20)$$

式 (20) において n は歩数を表し, x_i^{est}, y_i^{est} は i 歩目における推定した x 座標および y 座標であり, x_i^{act}, y_i^{act} は i 歩目における実際の経路の x 座標および y 座標である. 座標の単位は cm である.

表 2 に示す通り, 直線経路において 99.94%, L 字型経路において 75.14% の推定位置誤差を削減できた. いずれのスマートフォンのモードにおいても推定位置誤差を大きく削減できている.

その結果, 提案手法は 1 章で述べた (条件 a)~(条件 c) を全て満足していると言える.

5. 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 手法を提案した. 提案手法では様々なスマートフォンのモードに対応したうえで, ドリフト誤差を削減した. 提案手法では, 既存手法と比較し平均約 87% の位置誤差を削減した.

今後の課題として, 方向転換時の誤差補正手法の考案や, 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] 総務省, “平成 28 年版 情報通信白書 第 1 部,” 2018, <https://www.soumu.go.jp/johotsusintokei/whitepaper/ja/h28/pdf/n2200000.pdf>.
- [2] 星尚志, 藤井雅弘, 羽多野裕之, 伊藤篤, 渡辺裕 *et al.*, “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.
- [3] 住田明優, 小坂隆浩 *et al.*, “Fingerprint と pdr を用いたユーザ自己補正による屋内位置推定手法の提案,” 第 79 回

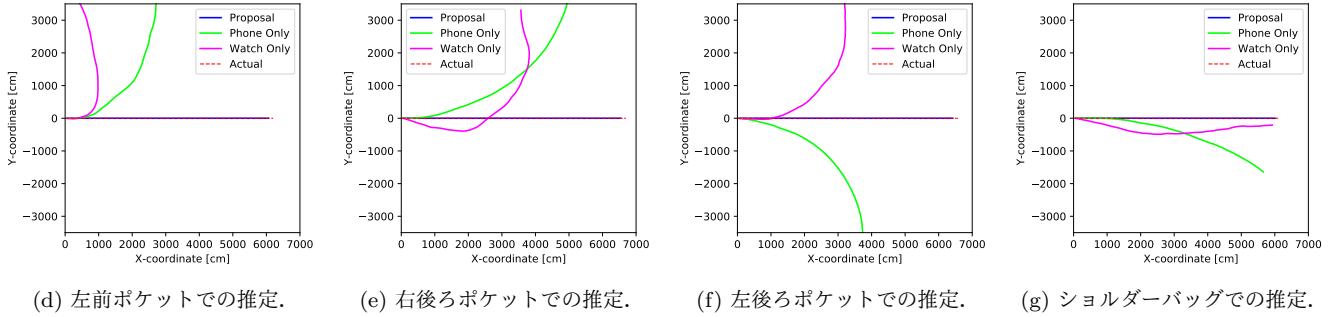
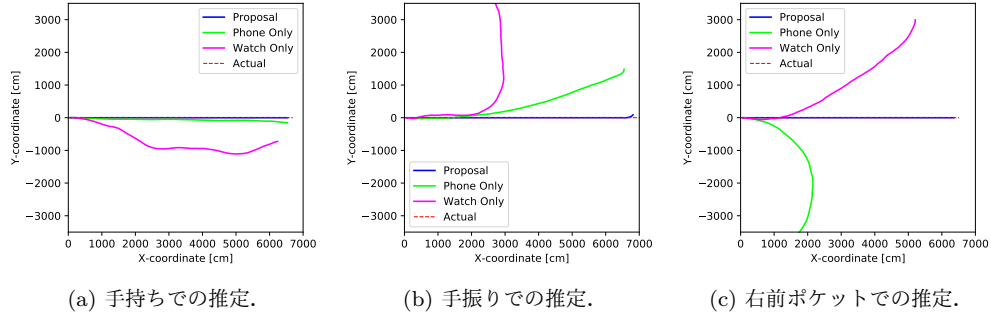


図 12: 直線経路における推定軌跡.

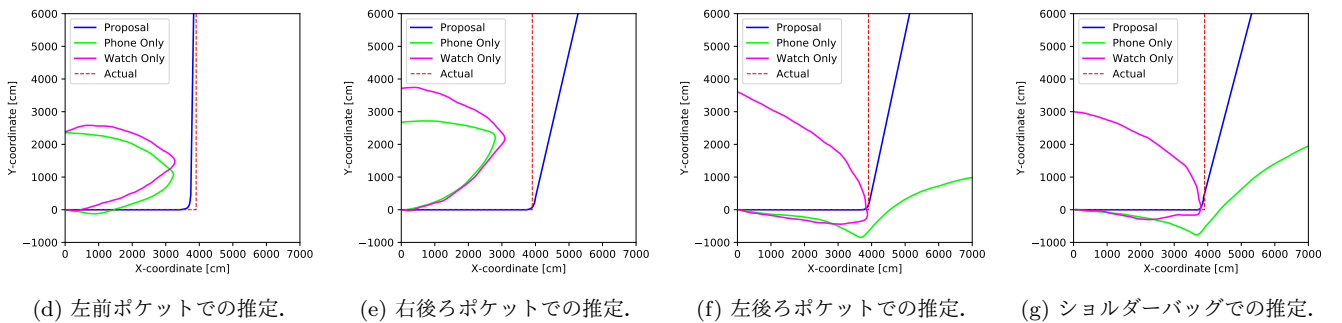
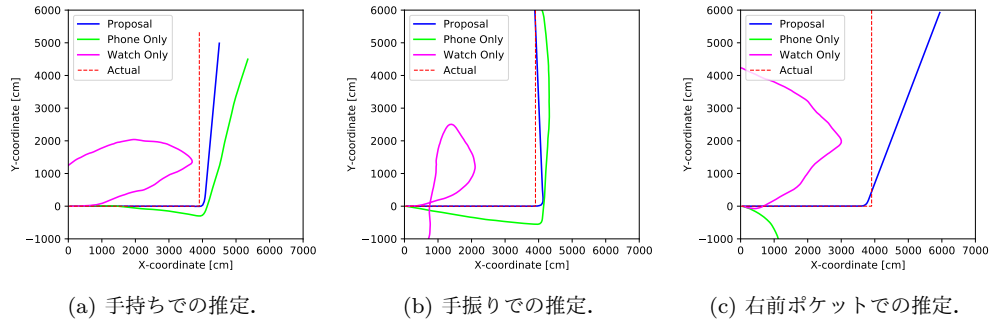


図 13: L 字型経路における推定軌跡.

- 全国大会講演論文集, vol. 2017, no. 1, pp. 801–802, 2017.
- [4] Apple, “Apple watch,” 2020, <https://www.apple.com/jp/watch/>.
- [5] Apple, “iphone xr,” 2020, <https://www.apple.com/jp/iphone-xr/specs/>.
- [6] 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.
- [7] J. Borenstein, L. Ojeda, and S. Kwanmuang, “Heuristic reduction of gyro drift for personnel tracking systems,” *The Journal of Navigation*, vol. 62, no. 1, pp. 41–58, 2009.
- [8] A. Correa, E. Munoz Diaz, D. Bousdar Ahmed,

表 2: 推定位置誤差 err の比較.

推定位置誤差 (cm)		手持ち	手振り	右前 ポケット	左前 ポケット	右後 ポケット	左後 ポケット	ショルダー バッグ	削減率 (%)
直線経路	提案手法	0.0	2.6	0.0	0.0	0.0	0.0	0.0	99.94
	[2]	89.8	271.9	455.4	1452.0	1579.0	559.4	333.5	
L 字型経路	提案手法	125.5	64.0	712.8	80.2	433.3	450.1	478.3	75.14
	[2]	625.1	171.2	1839.1	612.4	1980.3	1899.2	2300.7	

A. Morell, and J. Lopez Vicario, “Advanced pedestrian positioning system to smartphones and smartwatches,” *Sensors*, vol. 16, no. 11, p. 1903, 2016.

- [9] Google, “Google pixel 3a,” 2020, https://store.google.com/jp/product/pixel_3a.
- [10] 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.
- [11] J.-S. Lee and S.-M. Huang, “An experimental heuristic approach to multi-pose pedestrian dead reckoning without using magnetometers for indoor localization,” *IEEE Sensors Journal*, vol. 19, no. 20, pp. 9532–9542, 2019.
- [12] Y. Li, Y. Zhuang, H. Lan, Q. Zhou, X. Niu, and N. El-Sheimy, “A hybrid wifi/magnetic matching/pdr approach for indoor navigation with smartphone sensors,” *IEEE Communications Letters*, vol. 20, no. 1, pp. 169–172, 2015.
- [13] D. Loh, S. Zihajehzadeh, R. Hoskinson, H. Abdollahi, and E. J. Park, “Pedestrian dead reckoning with smart-glasses and smartwatch,” *IEEE Sensors Journal*, vol. 16, no. 22, pp. 8132–8141, Nov 2016.
- [14] mobvio, “Ticwatch pro,” 2020, <https://www.mobvio.com/jp/pages/ticwatchpro>.
- [15] J. Nozaki, K. Hiroi, K. Kaji, and N. Kawaguchi, “Compensation scheme for PDR using sparse location and error model,” in *Proceedings of the 2017 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2017 ACM International Symposium on Wearable Computers*, pp. 587–596. ACM, 2017.
- [16] S. Y. Park, S. J. Heo, and C. G. Park, “Accelerometer-based smartphone step detection using machine learning technique,” in *2017 International Electrical Engineering Congress (iEECON)*, pp. 1–4, March 2017.
- [17] J. A. Rice, *Mathematical statistics and data analysis*. Cengage Learning, 2006.
- [18] 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.
- [19] Z. Xiao, H. Wen, A. Markham, and N. Trigoni, “Robust pedestrian dead reckoning (R-PDR) for arbitrary mobile device placement,” *2014 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, pp. 187–196, 2014.
- [20] 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.