

端末姿勢の安定性により蓄積誤差を 低減するスマートグラスPDR手法

佐藤 大生^{1,a)} 戸川 望^{1,b)}

概要: スマートグラスによる歩行者ナビゲーションは、視界に直接様々な情報が表示されるため、画面を確認するために視線を落とす必要がなく、より直観的で分かりやすいナビゲーションを実現できる。歩行者ナビゲーションは自己位置の推定が不可欠であり、広く Global Positioning System (GPS) が用いられるが、屋内、地下、高層ビル街などでは測位精度が著しく低下する。屋内空間で位置測位する手法として、Pedestrian Dead Reckoning (PDR) があり、端末のセンサを用いて現在位置を推定する。PDR は外部インフラを必要とせず低コストで導入することができるが、初期位置を基準とした相対位置を推定するため、歩行距離が増大するにつれて誤差が蓄積する。蓄積誤差を低減することが PDR の実現において重要である。本稿では、スマートグラスによる歩行者ナビゲーションを対象に、蓄積誤差を低減する PDR 手法を提案する。提案手法では、歩行者が正面を向いて静止したときスマートグラス端末の鉛直下向きの加速度が重力加速度に近づくことを利用し、端末の水平面の角度の蓄積誤差の低減を実現し、その結果、PDR による蓄積誤差の低減を可能とする。評価実験の結果、提案手法により、推定位置の誤差が最大で約 13% 低減されることが分かった。

1. はじめに

1.1 スマートグラス PDR 手法

近年、スマートフォンやタブレット、ウェアラブル端末の普及に伴い、歩行者向けのナビゲーションが広く利用されている。その中でもスマートフォンによる歩行者ナビゲーションシステムは広く普及し、多くのユーザに利用されている [1-3]。一方、スマートフォンによる歩行者ナビゲーションは、経路案内を確認するために歩きながら画面を注視する「歩きスマホ」の危険性 [4] や、ディスプレイが小さいこと、さらに地図上に経路誘導情報を表示するため、地図の読めないユーザにとっては分かりにくいという問題点が指摘されている [5]。

一方で、スマートグラス上で動作するアプリケーションでは、視界に直接様々な情報を表示することができる [6]。スマートグラスによる歩行者ナビゲーションが実現できれば、地図ではなく視界に直接案内表示が出ることで、画面を確認するために視線を落とす必要がなく、より直観的で分かりやすいナビゲーションを実現できる。

ナビゲーションシステムには位置推定、経路探索、経路

誘導の 3 つの機能がある。その中でも、位置推定は最も基本的な機能である。

主流な位置推定手法として、Global Positioning System (GPS) がある。GPS は屋外では高精度な測位が可能だが、屋内や地下、高層ビル街 (以下、このような環境を屋内環境あるいは屋内と呼ぶことにする) などでは、著しく精度が低下する。特に都心部では、GPS が十分な精度を得られない場所が多くあるが、歩行者は屋内と屋外の両方でナビゲーションを必要とするため、屋内では GPS に代わる位置推定手法が必要である [7]。

屋内位置推定に有力な手法として、Pedestrian Dead Reckoning (PDR) がある。PDR 手法では、主に端末の加速度センサ・角速度センサ・磁気センサを用いて、初期位置を基準とした相対的な現在位置を推定する。衛星や基地局との通信を必要としないため、オフラインで利用できる。また、端末本体のセンサを使用するため新たな外部インフラの整備が必要なく、導入コストが低いことも利点である。しかし、PDR 手法では、移動距離が長くなるにつれて誤差が蓄積し、最終的に大きな蓄積誤差が生じるという欠点がある。PDR 手法を実現するためには、蓄積誤差の低減が重要である。

本稿では、スマートグラスを用いた歩行者ナビゲーションに注目し、中でも PDR 手法を用いた屋内位置推定に焦

¹ 早稲田大学
Waseda University

a) dai.sato@togawa.cs.waseda.ac.jp

b) togawa@togawa.cs.waseda.ac.jp

点を当てる。前述のように、スマートグラスを用いた PDR 手法では、以下の 2 点を満足する必要がある

A 外部インフラを利用せず、スマートグラスのみを利用した PDR を実現する。

B PDR による蓄積誤差を低減する。

本稿では、要件 A, B を共に満たすスマートグラス PDR 手法を提案するものとする。

1.2 先行研究

スマートグラスを利用した PDR 手法として以下のものがある。

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 [8]。Loh らはスマートウォッチ単体の PDR、スマートグラス単体の PDR、2 つの端末を併用した PDR の 3 パターンを実験している。スマートウォッチ単体での PDR 手法では、歩数検知は概ね正しく動作するものの、腕を振る方向によって進行方向が推定されるため、進行方向および推定位置のドリフトが大きいという欠点がある。スマートグラス単体の PDR 手法では、同様に歩数推定は正しく動作するが、進行方向を頭の向きで推定するため、歩行に伴う進行方向の変更とは関係のない端末姿勢の傾きが進行方向推定を阻害するという欠点がある。

Loh らが提案するスマートウォッチとスマートグラスの併用手法では、2 つのデバイスを組み合わせることによって、これらの欠点を補うことができる。しかし、2 種類の端末が必要になる上、2 つの端末を併用するために歩行者の歩行動作がより制限される。例えば、歩行実験中は視線と進行方向は一致させる必要があり、腕を常に一様に振る必要がある。また、この研究では 2 つの端末を適切に組み合わせ精度を向上させることを主題としており、それぞれ端末単体での PDR では必ずしも精度が良いとは言えない。すなわち、前述の要件 A 並びに B の両方が満足されない。

Hou らはスマートグラスとセンサ搭載のマウスピースを用いた PDR 手法を提案している [9]。Hou らの手法では、加速度センサデータと角速度センサデータをフィルタを通して融合することで、進行方向の誤差を低減している。センサ搭載のマウスピースは一般的に普及しておらず、ユーザが手軽に利用できる端末とは言えない。また、60m 程度の実験結果しかなく、より長距離になった場合により多くの蓄積誤差が発生する可能性がある。そのため、この手法も要件 A と要件 B を同時に満足しているとは言えない。

Hasan らはスマートグラスのみを利用した PDR 手法を提案している [10]。Hasan らのスマートグラス PDR 手法では、Extended Kalman Filter (EKF) アルゴリズムを加速度データと角速度データに適用し、データを融合することでセンサデータの誤差を低減している。しかし、入力と

なるセンサデータの誤差を低減しても、歩行距離に応じて微小な誤差が蓄積し、最終的に大きな誤差になる課題を解決できていない。そのため、要件 B を満足しているとは言えない。

1.3 スマートグラスの端末姿勢安定性

ここでスマートグラスを装着した歩行者の動作に注目する。一般に、歩行者は歩行途中に交差点や施設出入口、エレベーターなどで一時停止をしたり、ランドマークで自己位置を確認するために一時停止をしたりする傾向がある [11, 12]。スマートグラスを着用した歩行者が、地平面に対して垂直な状態で正面を向いて停止している場合 (以下、このような動作を静止と呼ぶこととする)、スマートグラス端末は前方を向いていると見なすことができ、スマートグラス端末の姿勢を補正することができる。端末姿勢の蓄積誤差の低減に大きく寄与する可能性がある。

ここで問題となるのが歩行者の動作の推定である。本稿では、歩行者の静止を推定するために、スマートグラス端末の加速度データに注目する。重力加速度は、歩行者の動作にかかわらず、同じ大きさかつ同じ方向である。一方、歩行者の歩行中は、歩行動作や歩行とは関係のない頭部の揺れ等で加速度データは重力加速度と有意に異なる数値となるが、歩行者が静止中は、加速度データは重力加速度と等しくなる傾向にある (詳細は 3.6 節参照)。このような性質を利用して、歩行者の静止を推定し、これによりスマートグラス端末の姿勢を補正する。その結果、上述した要件 A ならびに要件 B を同時に満足することを目指す。

1.4 本稿の提案

本稿では、スマートグラスの特徴である端末姿勢の安定性に着目し、蓄積誤差を低減するスマートグラスを用いた PDR 手法を提案する。特に、歩行者がしばしば静止することを利用し、端末角度を補正する。ユーザが正面を向き静止しているとき、端末にかかる鉛直下向きの加速度は重力加速度と一致するとみなすことができる。つまり、端末の鉛直下向きの加速度が重力加速度の大きさに近いとき、重力加速度ベクトルを特定することができる。このとき、端末姿勢の水平面の角度を 0° と補正することで、蓄積誤差を低減する。端末姿勢の蓄積誤差を低減することで、推定位置の蓄積誤差を低減し、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) スマートグラス端末にかかる加速度を利用して、蓄積誤差を低減することができるスマートグラス PDR 手法を提案した。
- (2) 評価実験を実施し、蓄積誤差の低減アルゴリズムを適用しない場合に比べて、推定位置の誤差が最大で約

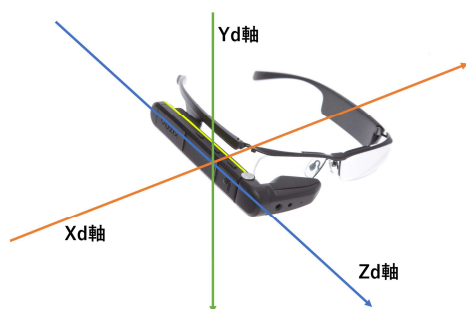


図 1: スマートグラスのデバイス座標系.

13%, 右折時の角度の誤差が約 9.5% 低減された.

1.6 本稿の構成

本稿は以下のように構成される. 2 章ではスマートグラス PDR 手法を用いた屋内位置推定問題を定義する. 3 章では蓄積誤差を低減するスマートグラス PDR 手法を提案する. 4 章では評価実験とその結果について述べる. 5 章では本稿をまとめる.

2. スマートグラス PDR 手法を用いた屋内位置推定問題

2.1 地理空間座標系とスマートグラスのデバイス座標系

本節では, 地理空間座標系とスマートグラスのデバイス座標系 (以下, デバイス座標系と呼ぶ) を定義する. 図 1 にデバイス座標系を示す [10]. また, 図 2 に地理空間における座標系を示す. 図 1 に示す通り, スマートグラス端末の視線方向を Zd 軸, 静止時に鉛直下向きになる方向を Yd 軸, Zd 軸と Yd 軸と直交する軸を Xd 軸とする. また, 図 2 に示す通り, 測位開始時のユーザの視線方向を Zg 軸, 地平面に垂直な方向を Yg 軸, Zg 軸と Yg 軸と直交する軸を Xg 軸とする.

さらに, 図 3 にスマートグラスの端末姿勢を決定するための 3 軸周りの角度を示す. 図 3(a) に示す通り, 地理空間の座標系における Xg-Zg 平面とデバイス座標系の Zd 軸の角度をピッチ角とする. これはスマートグラスの着用者が地平面に対してどれだけ見上げる・見下げるかを表す角度である. 図 3(b) に示す通り, Xg-Zg 平面と Xd 軸の角度をロール角とする. これは着用者が地平面に対してどれだけ首を傾けているかを表す角度である. 図 3(c) に示す通り, Yg-Zg 平面と Zd 軸の角度をアジマス角とする. 視線と進行方向が一致している場合, この角度が進行方向を表す.

2.2 センサデータ

一般に, スマートグラスは 3 軸加速度センサならびに 3

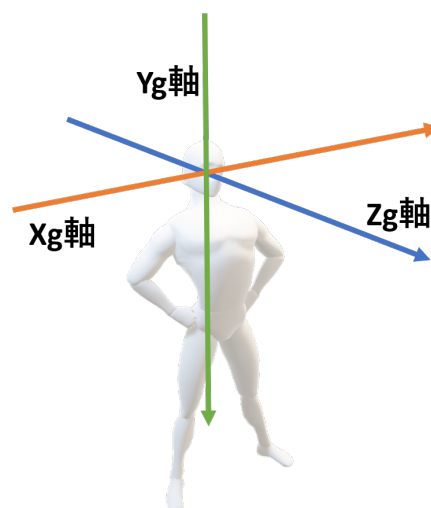


図 2: 地理空間の座標系.

軸角速度センサを持つ [8–10]. スマートグラスに搭載された 3 軸加速度センサは, 図 1 に示す, Xd 軸, Yd 軸, Zd 軸の各軸の方向にかかる加速度 $[\text{m/s}^2]$ を取得する. 時刻 t における加速度センサの値を $\vec{a}(t) = (a_x(t), a_y(t), a_z(t))$ とする.

スマートグラスに搭載された 3 軸角速度センサは, 図 1 に示す, Xd 軸, Yd 軸, Zd 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 における角速度センサの値を $\vec{\omega}(t) = (\omega_x(t), \omega_y(t), \omega_z(t))$ とする.

以上のセンサデータが一定時間 $c[\text{ms}]$ ごとに取得される. なお, 本稿の評価実験では, 加速度データ, 角速度データともに $c = 50\text{ms}$ とした.

2.3 PDR による屋内位置推定問題

本稿で取り扱う PDR による屋内位置推定問題を以下に定義する.

定義 1. 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 は既知とする. このとき, スマートグラスを利用した PDR による屋内位置推定問題とは, $c[\text{ms}]$ ごとにスマートグラスのセンサから加速度データと角速度データが与えられたとき, $c[\text{ms}]$ ごとに現在の自己位置を推定することである.

3. 蓄積誤差を低減するスマートグラス PDR 手法

3.1 アルゴリズムの流れ

本節では, 提案手法のアルゴリズムについて説明する. 図 4 に, 提案手法のフローチャートを示す. また, 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, 歩行中の加速度の周期性を利用し歩数を観測する.

Step 2 (端末姿勢推定) 端末姿勢推定では, 現在の端末

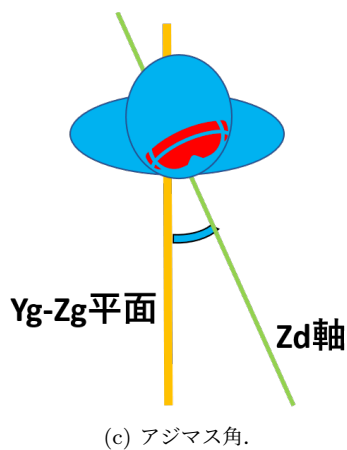
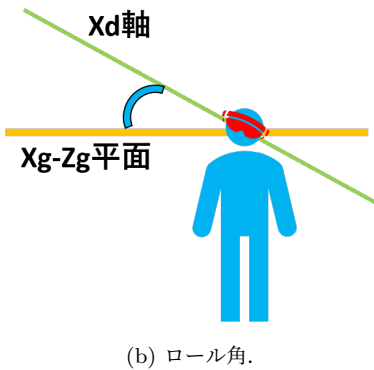
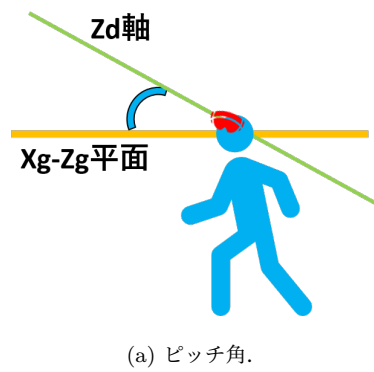


図 3: 端末姿勢を決定する角度.

姿勢をもとに、角速度のセンサデータを補正する。補正された角速度データを用いて端末姿勢を推定する。

Step 3 (進行方向推定) Step 2 の端末姿勢推定の結果得られたアジマス角を進行方向の角度とする。

Step 4 (位置推定) Step 1 で推定した歩数と、Step 3 で推定した進行方向の角度を用いて現在の自己位置を推定する。

Step 5 (端末姿勢推定の蓄積誤差低減) 端末の加速度が静止認識条件 (3.6 節参照) を満たす場合に、ピッチ角とロール角を 0° に補正し、蓄積誤差を低減する。以降、本章では上記の Step 1 から Step 5 を提案する。

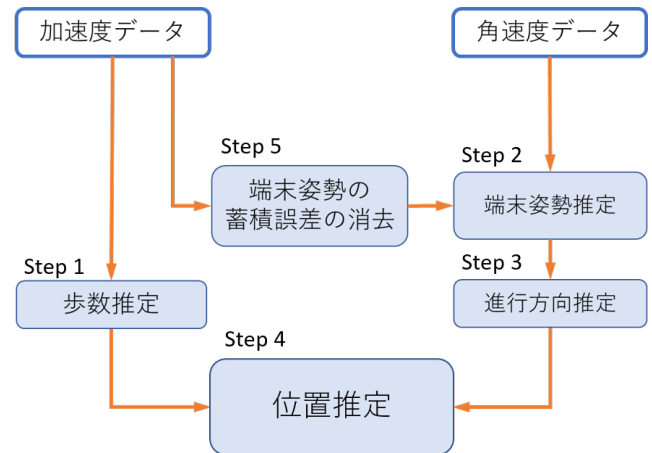


図 4: 提案手法のフローチャート.

3.2 歩数推定 (Step 1)

Step 1 として、スマートグラス端末のセンサによって取得した合計加速度を用いて歩数を推定する。時刻 t における合計加速度 $a_{sum}(t)$ は式 (1) で求められる。

$$a_{sum}(t) = \sqrt{a_x(t)^2 + a_y(t)^2 + a_z(t)^2} \quad (1)$$

ただし、 $a_x(t)$ 、 $a_y(t)$ 、 $a_z(t)$ はそれぞれ図 1 における Xd 軸、Yd 軸、Zd 軸方向の時刻 t における加速度の大きさである。式 (1) によって求めた歩行時の合計加速度波形を図 5 に示す。図 5 に示す通り、歩行者の歩行時の加速度波形は極大値と極小値が周期的に順番に表れる。この特性を利用し、極大値と極小値の 1 組を 1 歩と見なし歩数を測定する。

ここでは、Windowed Peak Detection (WPD) [8] を適用する。WPD では、一定の時間枠を設定し、時間枠内の加速度の極大値と極小値を見つける。この時間枠をウィンドウと呼ぶ。人はランニング時に毎分約 180 歩で移動することから [8]、少なくとも 1 歩当たり約 333ms かかると仮定できる。そこで、本手法ではウィンドウを 300ms に設定する。

続いて、合計加速度の極大値と極小値を得るために閾値を設定する。時刻 t の合計加速度 $a_{sum}(t)$ が条件式 (2) を満足するとき $a_{sum}(t)$ を極大値と呼ぶ。

$$th_2 < a_{sum}(t) < th_1 \quad (2)$$

th_2 は、極大値を与える最小の合計加速度であり、 th_1 は何らかの影響で異常な合計加速度をとった場合に、それを回避するための極大値の上限値である。同様に、時刻 t の合計加速度 $a_{sum}(t)$ が条件式 (3) を満足するとき $a_{sum}(t)$ を極小値と呼ぶ。

$$th_4 < a_{sum}(t) < th_3 \quad (3)$$

th_3 は、極小値を与える最大の合計加速度であり、 th_4 は何らかの影響で異常な合計加速度をとった場合に、それを回避

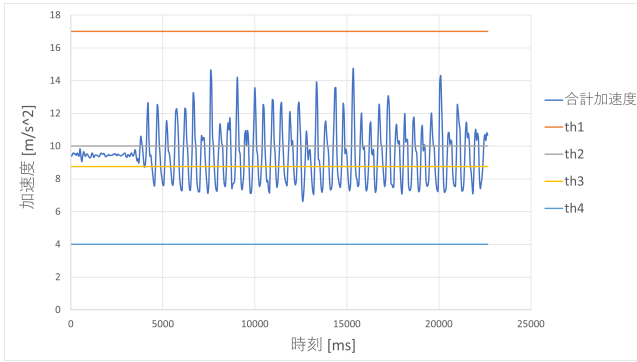


図 5: 歩行時の加速度波形.

表 1: 歩数検知の閾値.

閾値	値 [m/s ²]	用途
th1	17.0	過度に大きい加速度を除外
th2	10	極大値の検知
th3	8.75	極小値の検知
th4	4	過度に小さい加速度の除外

避するための極小値の最小値である.

今, ある時刻 t_0 で 1 歩を観測したと仮定する. その後の時刻において, 合計加速度が極小値, 極大値をこの順で取ったとする. このとき極大値が $t_0 + 300\text{ms}$ 以上で観測された場合, その時刻で次の 1 歩を検知する. つまり, 時刻 t_0 以降に, 1 回以上の極小値をとり, 極大値が $t_0 + 300\text{ms}$ 以上で観測された場合, その時刻で次の 1 歩を検知するものとする.

ここで, 閾値 th_1, th_2, th_3, th_4 の値は予備実験を通して次のように設定した. 予備実験では, スマートグラスを装着して 2 回の左折を含む 40m 程度のコースを歩行した. その後, 図 5 のような合計加速度波形をいくつか取得した. その上で, まず通常の歩行では超えないと考えられる閾値 th_1, th_4 を決定し, 続いて, 通常の歩行では少なくとも超えろと考えられる閾値 th_2, th_3 を決定した. 決定した 4 つの閾値を表 1 に示す.

3.3 端末姿勢推定 (Step 2)

Step 2 として, スマートグラス端末のセンサによって取得した角速度を用いて端末姿勢を推定する. スマートグラスで取得される 3 軸加速度および 3 軸角速度は, 図 1 のように, 端末の 3 軸を基準とする値である. しかし, 進行方向を推定するにあたり, 3 軸角速度を図 2 のような地理空間座標系の 3 軸を基準とする値に変換する必要がある. ここで, ピッチ角とロール角は, 図 3 に示すように, スマートグラス端末の水平面の傾きを表す角度である. 水平面の傾きを表す 2 つの角度から回転行列 [13] を作成し, デバイス座標系の 3 軸角速度に乗算することで, 地理空間座標系の 3 軸角速度を求めることができる. 図 6 に端末姿勢推定

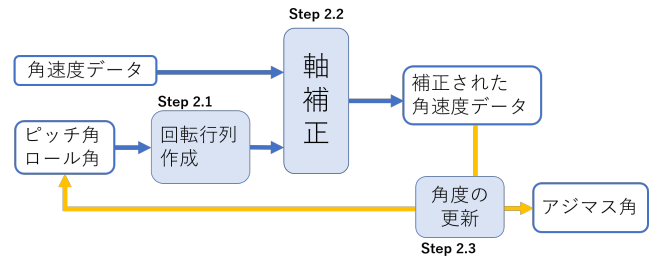


図 6: 端末姿勢推定 (Step 2) の流れ.

の流れを示す.

今, 時刻 t_N をセンサから角速度データを受け取るある時刻とする. 時刻 t_{N-1} を前回センサから角速度データを受け取った時刻とする. 2.2 節で述べた通り, 本手法ではセンサデータの取得間隔は $c = 50\text{ms}$ であるので, t_N と t_{N-1} は 50ms の間隔を持つ. 端末姿勢推定では, はじめに Step 2.1 としてピッチ角とロール角を用いて回転行列を作成する. 回転行列 R_{t_N} を時刻 t_N の回転行列とすれば, R_{t_N} は式 (4) で表される.

$$R_{t_N} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos \theta_r(t_{N-1}) & \sin \theta_r(t_{N-1}) \\ 0 & -\sin \theta_r(t_{N-1}) & \cos \theta_r(t_{N-1}) \end{pmatrix} \times \begin{pmatrix} \cos \theta_p(t_{N-1}) & 0 & \sin \theta_p(t_{N-1}) \\ 0 & 1 & 0 \\ -\sin \theta_p(t_{N-1}) & 0 & \cos \theta_p(t_{N-1}) \end{pmatrix} \quad (4)$$

ただし, $\theta_p(t_{N-1}), \theta_a(t_{N-1}), \theta_r(t_{N-1})$ は, 時刻 t_{N-1} におけるピッチ角, アジマス角, ロール角とする. これらの値は, 時刻 t_{N-1} にて Step 2 を実行した際に算出されるものである. 式 (4) の一番目の行列はピッチ角を用いたスマートグラス座標系の Xd 軸の回転を表し, 二番目の行列はロール角を用いたデバイス座標系の Zd 軸の回転を表す. これらの行列の積をとると, デバイス座標系の Xd-Zd 平面を地理空間座標系の Xg-Zg 平面に写像する回転を表すことになり, その結果, R_{t_N} が回転行列となる.

次に Step 2.2 として, センサから取得した角速度データに回転行列を乗ずることで軸を補正する. 時刻 t_N にスマートグラス端末で取得された, Xd 軸, Yd 軸, Zd 軸周りの角速度データを $g_x(t_N), g_y(t_N), g_z(t_N)$ とすると, 補正後の角速度データ $g_X(t_N), g_Y(t_N), g_Z(t_N)$ は回転行列 $R(t_N)$ を用いて, 式 (5) で表される.

$$\begin{pmatrix} -g_Z(t_N) \\ g_X(t_N) \\ -g_Y(t_N) \end{pmatrix} = R_{t_N} \begin{pmatrix} -g_z(t_N) \\ g_x(t_N) \\ -g_y(t_N) \end{pmatrix} \quad (5)$$

式 (5) を計算すると, 補正後の角速度データ $g_X(t_N), g_Y(t_N), g_Z(t_N)$ は, 式 (6)–式 (8) で表される.

$$g_X(t_N) = g_x(t_N) \cos \theta_r(t_{N-1}) - g_y(t_N) \sin \theta_r(t_{N-1}) \quad (6)$$

$$g_Y(t_N) = -g_z(t_N) \sin \theta_p(t_{N-1}) + \cos \theta_p(t_{N-1})(g_x(t_N) \sin \theta_r(t_{N-1}) + g_y(t_N) \cos \theta_r(t_{N-1})) \quad (7)$$

$$g_Z(t_N) = g_z(t_N) \cos \theta_p(t_{N-1}) + \sin \theta_p(t_{N-1})(g_x(t_N) \sin \theta_r(t_{N-1}) + g_y(t_N) \cos \theta_r(t_{N-1})) \quad (8)$$

Step 2.3 として、補正後の角速度データを用いてピッチ角、アジマス角、ロール角の3つの角度を求める。時刻 t_N におけるピッチ角、アジマス角、ロール角はそれぞれ式 (9)、式 (10)、式 (11) で求める。

$$\theta_p(t_N) = \sum_{k=1}^N (g_X(t_k) + g_X(t_{k-1})) \times \frac{c}{1000} \times \frac{1}{2} \quad (9)$$

$$\theta_a(t_N) = \sum_{k=1}^N (g_Y(t_k) + g_Y(t_{k-1})) \times \frac{c}{1000} \times \frac{1}{2} \quad (10)$$

$$\theta_r(t_N) = \sum_{k=1}^N (g_Z(t_k) + g_Z(t_{k-1})) \times \frac{c}{1000} \times \frac{1}{2} \quad (11)$$

$g_X(t_k), g_Y(t_k), g_Z(t_k)$ は時刻 t_k における、式 (5) で求めた時刻 t_k の補正後の角速度データを表す。また、 $\theta_p(t_N), \theta_a(t_N), \theta_r(t_N)$ は時刻 t_N における端末姿勢のピッチ角、アジマス角、ロール角を示す。式 (9)、式 (11) で求めたピッチ角およびロール角は、時刻 t_{N+1} における回転行列の作成に使用される。

3.4 進行方向推定 (Step 3)

Step 3 では Step 2 の結果を用いて進行方向を推定する。ユーザの実際の進行方向と視線方向が同じであれば、Step 2.3 で得られたアジマス角 $\theta_a(t_N)$ を進行方向とみなすことができる。アジマス角 $\theta_a(t_N)$ は式 (10) で求めることができる。

3.5 位置推定 (Step 4)

Step 4 として、主に Step 1 と Step 3 の結果から、自己位置を推定する。時刻 t_N のとき、Step 1 で現在の歩数が s と推測され、Step 3 で進行方向を表すアジマス角 $\theta_a(t_N)$ が推定される。歩数 s の時の相対位置 $(pos_{x,s}, pos_{y,s})$ は、進行方向を表すアジマス角 $\theta_a(t_N)$ を用いて式 (12)、式 (13) のように表される。

$$pos_{x,s} = pos_{x,s-1} + l \times \cos \theta_a(t_N) \quad (12)$$

$$pos_{y,s} = pos_{y,s-1} + l \times \sin \theta_a(t_N) \quad (13)$$

l は1歩当たりの歩幅を表す。1歩ごとに式 (12)、式 (13) を計算し、初期位置を基準とする相対的な現在の自己位置を推定する。

3.6 角度の初期化による蓄積誤差の低減 (Step 5)

Step 5 では、端末の加速度が特定の条件を満足するとき、ピッチ角とロール角の蓄積誤差を低減する。3.4 節で述べたように、進行方向の角度はアジマス角に等しい。アジマス角は、式 (7) で表される補正後の Yd 軸周りの角速度 $g_Y(t_N)$ を用いて、式 (10) で表される。つまり、進行方向角度であるアジマス角を求める式の中に、ピッチ角とロール角が含まれているため、ピッチ角とロール角の誤差が進行方向角度の誤差の要因となることが分かる。したがって、より正確なアジマス角を得るためには、ピッチ角とロール角の蓄積誤差を低減する必要がある。

一般的に、歩行者は歩行と停止を繰り返しながら目的地へ向かう。3分に1回程度停止をすることもあり、歩行中の停止行動は珍しくない [11,12]。スマートグラスの着用者が静止している場合、スマートグラス端末の Yd 軸が地平面に対して垂直な状態に近づくため、Yd 軸の加速度は重力加速度 9.8m/s^2 に近づく。このとき、ピッチ角、ロール角はともに 0° に近くなる。つまり、歩行者の静止を検知し、そのときにピッチ角とロール角を 0° にすれば、式 (9) や式 (11) によって算出されたピッチ角とロール角の蓄積誤差を低減することが期待できる。

歩行者の静止を検知するため、予備実験を実施し、歩行中と静止状態の加速度波形を取得した。予備実験の結果、静止時でも頭部の小さな傾きや動き、その他のノイズがあるため、Yd 軸の加速度が重力加速度 9.8m/s^2 に等しくなることは少なく、多くの場合、やや重力加速度より小さい値を示すことが分かった。ただし、Yd 軸の加速度 $a_y(t)$ が重力加速度に近い状態は歩行中にも瞬間的には表れる。そこで、 $a_y(t)$ が式 (14) を一定時間満たし続けることを静止認識条件*1と呼ぶこととする。

$$9.2 < a_y(t) \leq 9.8 \quad (14)$$

端末の Yd 軸の加速度が静止認識条件を満足する場合、歩行者が静止状態であると仮定する。

屋内における歩行者ナビゲーションにおいて、ドアの開閉やエレベーターを待ったり、案内板や看板、商品の確認のために短い時間だけ立ち止まったりするケースは屋外に比べて多いと予想される。したがって、0.5 秒間の静止であれば、屋内の歩行者にとっては自然な歩行を阻害しないものと考えられる。

以上より、提案手法では、歩行者が歩行中に 0.5 秒間静止することによって、ピッチ角とロール角を 0° に補正し、その地点まで発生している蓄積誤差を低減する。

*1 予備実験の結果、500ms の間、 $a_y(t)$ が式 (14) を満足したときとした。



図 7: VUZIX M300.

名称	VUZIX M300
OS	Android6.0.1
CPU	Dual Core Intel x86
搭載センサ	3 軸加速度センサ
	3 軸ジャイロセンサ
	3 軸地磁気センサ

4. 評価実験

4.1 実験環境

スマートグラスとして VUZIX M300 (図 7) を取り上げ、提案手法を VUZIX M300 に実装した。OS は Android6.0.1 で、実装に使用した言語は Java である。表 2 に VUZIX M300 の性能を示す。表 2 に示す通り、VUZIX M300 には加速度、角速度、磁気のセンサが搭載されており、本実験では加速度と角速度データを用いる。図 8 に歩行コースを示す。スタート位置からゴール位置まで、合計で約 101 m である。スタート地点では、地理空間座標とデバイス座標系は一致しているとする。また、歩幅は 0.70m で固定値とする。

本実験では比較のため、提案手法と比較手法によって評価を行った。ここで比較手法は、Loh らの手法 [8] にならない、図 4 のフローにおいて Step 5 を削除したものをを用いた。提案手法では、図 8 に示す点 P で 0.5 秒間立ち止まり、Step 5 を実施するものとする。一方、比較手法では点 P で立ち止まらず、Step 5 を実施しないものとする。いずれの場合も 8 回ずつ計測を行った。50ms ごとに加速度データと角速度データを記録し、歩数と進行方向を推定・記録する。歩行実験後、得られたデータから、推定位置を図示する。

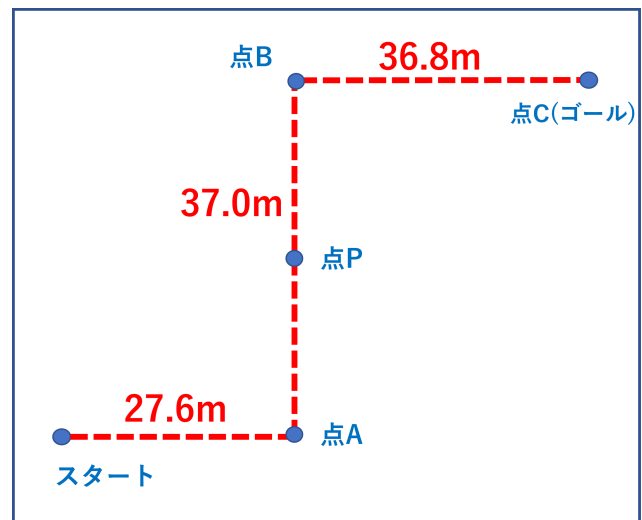


図 8: 歩行コース.

表 3: 歩数推定の結果.

手法	回数	実際の歩数	推定歩数	相対誤差 [%]
比較手法	1 回目	141	141	0
	2 回目	137	137	0
	3 回目	143	142	-0.699
	4 回目	139	135	-2.88
	5 回目	141	141	0
	6 回目	139	140	0.719
	7 回目	140	139	-0.714
	8 回目	137	137	0
提案手法	1 回目	139	138	-0.719
	2 回目	140	137	-2.14
	3 回目	140	138	-1.43
	4 回目	139	136	-2.16
	5 回目	140	138	-1.43
	6 回目	137	138	0.730
	7 回目	137	137	0
	8 回目	140	140	0

4.2 実験結果

4.2.1 歩行推定の評価

提案手法と比較手法で 8 回ずつ図 8 に示すコースを歩き、実際の歩数と推定歩数を比較した。表 3 に歩数推定実験の結果を示す。表 3 に示す通り、提案手法および比較手法において絶対誤差は最悪の場合で 4 歩、相対誤差は -2.88% であった。比較手法における Step 5 の実施に伴う停止や、コースの歩行を終える際の停止では加速度が小さくなるため 1 歩として正しく検知できない場合があるが、表 3 の通り、概ね正しく歩数を推定できると言える。

4.2.2 端末姿勢の蓄積誤差低減アルゴリズムの評価

提案手法と比較手法で 8 回ずつ図 8 に示すコースを歩き、実際の到達位置と推定位置を比較した。図 9 に、比較

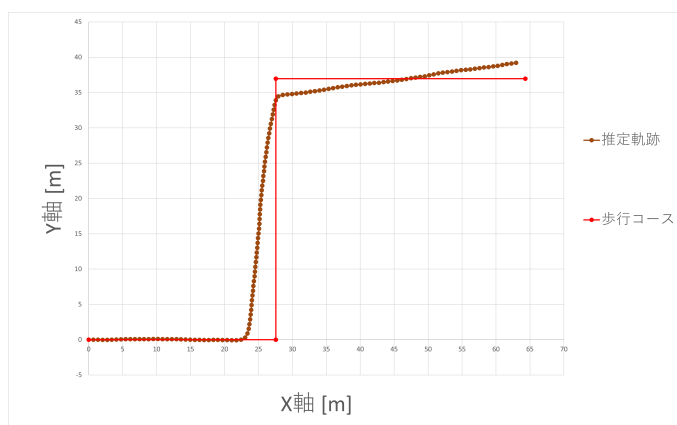


図 9: 比較手法の推定軌跡.

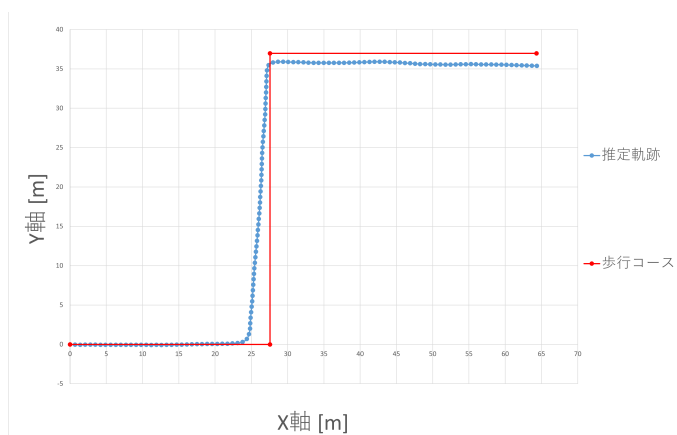


図 10: 提案手法の推定軌跡.

表 4: 実際の各地点と推定位置の平均誤差.

評価地点	比較手法	提案手法
	平均誤差 [m]	平均誤差 [m]
A	3.98(1.00)	4.09(1.03)
B	3.79(1.00)	3.31(0.87)
C	9.26(1.00)	8.37(0.90)
平均	5.68(1.00)	5.26(0.93)

手法の推定軌跡を示す. 同様に, 図 10 に提案手法の推定軌跡を示す. なお, いずれの図も赤色の実線の経路は実際の歩行コースを示し, 点線の経路が推定軌跡を示す. 図 9, 図 10 に示す通り, 比較手法に対して提案手法では, 点 P 通過後の軌跡が実際の歩行コースに近くなっている. 表 4 に実際の各地点と推定位置の平均誤差を示す. 平均誤差とは, 8 回歩行実験を実施し, 図 8 に示す点 A, 点 B, 点 C における実際の位置と推定位置の誤差を評価し, それらを地点ごとに平均したものを表す. なお, 括弧の中の値は比較手法の平均誤差を 1.00 としたときの提案手法の平均誤差の割合を表す. 表 4 に示す通り, 点 A における平均誤差は概ね同じ値であるが, 点 P 通過後の点 B および点 C では, 提案手法の方が平均誤差を低減できている. 提案手法

表 5: 推定右折角度.

回数	比較手法	提案手法
	右折角度 [°]	右折角度 [°]
1 回目	100	97.7
2 回目	99.4	95.7
3 回目	122	97.1
4 回目	97.6	98.7
5 回目	124	91.9
6 回目	106	98.2
7 回目	98.3	98.0
8 回目	108	96.6
平均	107	96.7

により, 推定位置の平均誤差が点 B では約 13%, 点 C では約 9.6%, 各地点の平均では約 7.4% 低減される.

一方で, 最初の左折時点で既に実施の歩行コースから大きくずれている場合, 実際の到達地点と推定位置の誤差による評価が有効でない可能性がある. そのため, 提案手法と比較手法で, 点 P 通過後の右方向への方向転換の精度を評価する. 表 5 に歩行コースにおける点 P 通過後の右折時の推定右折角度を示す. 実験では, 一定の歩行コースに従って歩行しているため, 実際の右折角度はおよそ 90° である. 表 5 に示す通り, 提案手法の方が実際の右折角度 90° に近く, アジマス角の誤差が低減されている. 提案手法により, 推定右折角度の誤差は約 9.5% 低減される.

以上より, 蓄積誤差低減の有効性が確認でき, 提案手法は 2.3 節で議論した要件 A ならびに要件 B を満足していると言える.

5. おわりに

本稿では, 加速度データを用いて蓄積誤差を低減するスマートグラス PDR 手法を提案した. 提案手法では, 端末にかかる鉛直下向きの加速度が一定時間重力加速度に近いことを検知する. その際, 歩行者が正面を向いて水平面の傾きが無い状態で静止していると判断できるため, ピッチ角とロール角を 0° に補正する. ピッチ角とロール角は進行方向の角度であるアジマス角の算出に用いるため, ピッチ角とロール角の蓄積誤差を低減することで, 推定位置の誤差を低減することができる. 評価実験を実施した結果, 比較手法に対し, 提案手法の推定位置の誤差とアルゴリズム適用後の右折角度の誤差が小さくなった.

今後の課題としては, 提案手法による位置測位をベースとし, スマートグラスによる歩行者ナビゲーションにおける経路誘導の実装が考えられる.

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] 安齋恵一, 岡島匠吾, 坪川宏, “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.
- [2] 星尚志, 羽多野裕之, 藤井雅弘, 渡辺裕, “スマートフォンのセンサを用いた保持状態に基づく歩行者デッドレコニングに関する一検討,” **第 76 回全国大会講演論文集**, vol. 2014, no. 1, pp. 211–212, 2014.
- [3] M. S. Kiran, P. Rajalakshmi, M. K. Giluka, and B. R. Tamma, “A novel system architecture for real-time, robust and accurate step detection for pdr based indoor localization,” in *2018 IEEE 4th World Forum on Internet of Things (WF-IoT)*, pp. 754–759. IEEE, 2018.
- [4] 中村葵, 村田伸, 飯田康平, 井内敏揮, 鈴木景太, 中島彩, 中嶋大喜, 白岩加代子, 安彦鉄平, 阿波邦彦, “歩きスマホが歩行に及ぼす影響について,” **ヘルスプロモーション理学療法研究**, vol. 6, no. 1, pp. 35–39, 2016.
- [5] 吉野孝 奥村賢悟, “迷いやすい人の特徴を考慮した屋内ナビゲーションシステム docoka の開発と評価,” **研究報告 グループウェアとネットワークサービス (GN)**, vol. 2013, no. 18, pp. 1–8, 2013.
- [6] 亀津敦, “ウェアラブルコンピューティングの動向とアプリケーションの展望,” **成形加工**, vol. 28, no. 1, pp. 2–7, 2015.
- [7] 河口信夫, “測位技術の変遷,” **電気学会誌**, vol. 134, no. 12, pp. 832–835, 2014.
- [8] 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, 2016.
- [9] X. Hou and J. Bergmann, “A pedestrian dead reckoning method for head-mounted sensors,” *Sensors*, vol. 20, no. 21, p. 6349, 2020.
- [10] M. A. Hasan and M. N. Mishuk, “Mems imu based pedestrian indoor navigation for smart glass,” *Wireless Personal Communications*, vol. 101, no. 1, pp. 287–303, 2018.
- [11] 宮下浩一 寺田努, “歩行・停止情報を用いたユーザの移動経路推定手法,” **情報処理学会論文誌**, vol. 57, no. 7, pp. 1601–1610, 2016.
- [12] 村田雄哉, 梶克彦, 廣井慧, 河口信夫, “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.
- [13] 木山昇, 高橋利光, 祖父江恒夫, 相川哲盛, “傾斜したスマートフォンによる自動車の 3 軸加速度算出手法,” **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. 16–23, 2014.