

# 歩容に頑健なPDRのための 深層学習を用いた進行方向推定手法

吉田 拓人<sup>1,a)</sup> 廣井 慧<sup>1</sup> 米澤 拓郎<sup>1</sup> 河口 信夫<sup>1</sup>

## 概要：

慣性センサを搭載したスマートフォンの普及に伴い、慣性データから歩行者の相対位置を推定する手法である Pedestrian Dead Reckoning (PDR) の研究が盛んに行われている。前の位置を元に次の位置を推定する PDR では、誤差の累積が課題とされてきた。PDR は、速度推定と進行方向推定の 2 つの要素からなり、両方が高精度でないと誤差が大きくなる。我々は以前に深層学習を用いた PDR 手法を検討し、深層学習による高精度な速度推定を実現した。進行方向推定にも深層学習を取り入れるにあたり、歩行速度と進行方向には密接な関係があることを考えると、進行方向と速度を同時に推定することでさらなる高精度化が実現できる。本稿では、加速度と角速度の時系列入力から、速度と進行方向を End-to-end で推定する LSTM ベースの深層学習モデルを提案する。評価は有限オートマトンベースの既存手法と深層学習ベースの提案手法を速度推定、進行方向推定、経路推定の 3 つの項目で比較する。結果、速度推定、進行方向推定、推定経路全ての項目において提案手法の方が高精度であることが確認できた。

## 1. はじめに

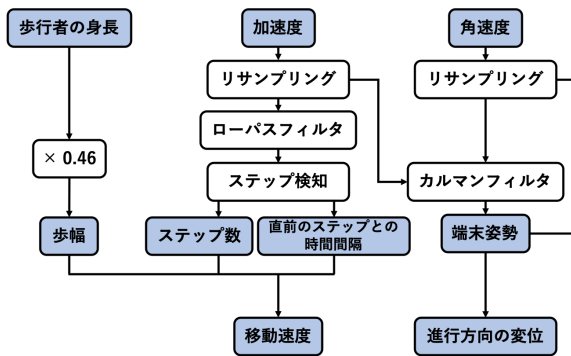
屋内位置推定は地下街や駅におけるナビゲーション、展示会における人流解析、工場における作業員の安全管理など様々な場面で必要とされている。さらに近年のスマートフォンの普及に伴い、スマートフォンの IMU を使用した PDR の研究が盛んに行われている。PDR は屋内に Wi-Fi や BLE ビーコンのようなインフラの設置をする必要のない点に特徴がある。現在研究されている多くの PDR は Strapdown algorithm と Step-and-heading algorithm に分類される [1], [2], [3], [4]。Strapdown algorithm は正しい位置情報を推定するために高精度のデータを観測できる IMU を必要とする。しかし、現在スマートフォンに搭載されている低コストな IMU では正確な位置を推定するだけの十分な精度のデータを観測できず、推定位置の誤差が蓄積されてしまう。Step-and-heading algorithm は位置推定のために速度と進行方向の推定を行う。速度推定は歩行者の歩幅推定とステップ検知から行い、進行方向推定は端末姿勢推定と水平角速度の積分によって行う。しかし、歩容の変化やセンサのノイズに対するロバスト性は低く、実際の工場現場などで用いると位置の推定精度が大きく低下する。

我々は深層学習を用いて PDR の精度、ロバスト性を向

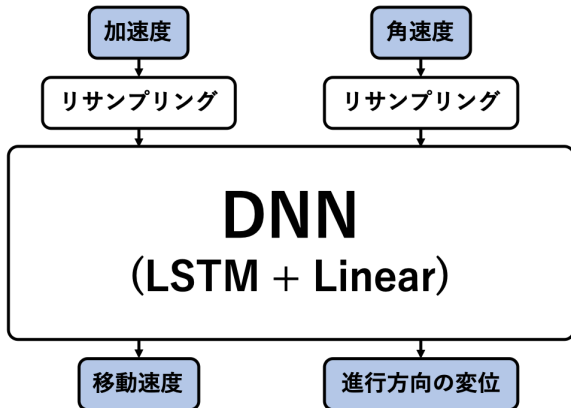
上させる手法を検討してきた。これまでに、深層学習による歩行者の速度推定手法を提案し、推定精度と歩容に対するロバスト性の向上を実現した [5], [6]。さらに、進行方向推定にも深層学習を取り入れるにあたり、歩行速度と進行方向には密接な関係があることから、進行方向と速度を同時に推定することでさらなる高精度化が実現できると考えた。そこで本研究では、加速度と角速度の時系列入力から、速度と進行方向を End-to-end で推定する LSTM ベースの深層学習モデルを提案する。図 1 に既存手法である Step-and-heading algorithm と提案手法である深層学習ベースの推定の流れの比較を示す。図 1a の既存手法では結果を出力するまでにステップ検知やカルマンフィルタによる姿勢推定の処理が挟まるのに比べ、図 1b の提案手法では加速度と角速度から End-to-end で速度と進行方向の変位を推定できる点が大きな違いである。既存手法のステップ検知ノードのようにマニュアル設計されたコンポーネントは入力データの潜在的な情報を取り除いてしまう場合があるのに対して、提案手法のような End-to-end の学習モデルは入力データの潜在的な情報まで学習できるため、推定精度とノイズ、歩容、端末保持位置に対するロバスト性の向上が期待できる。

<sup>1</sup> 名古屋大学  
Nagoya University

a) takuto@ucl.nuee.nagoya-u.ac.jp



(a) 既存手法 (Step-and-heading algorithm)



(b) 提案手法

図 1: 予測モデルの比較

Fig. 1 Comparison of prediction models

## 2. 関連研究

### 2.1 屋内位置推定手法

Pedestrian localization systems を扱った研究は多数存在する [1], [2], [3]. 一定の成功を収めた研究の一つとして足につけたセンサで位置誤差を補正する ZUPT (Zero Velocity Updates) [7] をベースとした研究がある. しかし, この手法は足にセンサをつけなくてはならずスマートフォンを使用できない. 現在, スマートフォンを用いた PDR のほとんどは Step-and-heading algorithm である. ステップ検出のために, Alzantot ら [8] はピーク検出を有限オートマトンで行う手法を提案した. ステップ幅推定では Stacked autoencoders [9] を使用した手法がある. このように PDR の精度向上のため, 様々な手法が提案されている. しかし, どの研究も歩容, 端末姿勢, ノイズなどに対するロバスト性がまだ十分ではなく, 実用化されているものは少ない.

### 2.2 End-to-end Machine Learning System

近年の深層学習の技術は様々なドメインにおいて End-to-end の機械学習を可能にしている [10], [11], [12]. 我々はこれらの研究から様々なヒントを得た. 例えば, Speech-Recognition を End-to-end で可能にした Deep Speech [13]

である. Deep Speech は全結合層と Bidirectional Recurrent Neural Network を結合したモデルを用いて音声データセットからの学習を可能にした. また, 行動認識分野においても深層学習を利用した手法が増加している. Ordóñez らの研究 [14] では, 複数の畳み込み層とリカレント層を用いた構造の深層学習モデルを作成し行動認識を行なっている. この研究では, センサデータを短いセグメントに分割しモデルに入力し, セグメント毎に認識を行うことを繰り返して予測を行う. Peng らの研究 [15] では, 複数の畳み込み層を用いた構造の深層学習モデルを用いて歩く, 止まるなどの単純な行動の認識を行い, 畳み込み層の出力をリカレント層に入力し, 複雑な行動の認識を行う. これらのリカレント層と時系列データを一定時間のセグメントごと扱う手法を用いれば, 歩行者の速度と進行方向の推定も可能ではないかと我々は考えた.

## 3. 提案手法

本章では深層学習を用いて慣性データから歩行者の位置を推定する手法を提案する. また, 慣性データと位置データの収集方法と学習データの生成方法についても説明する.

### 3.1 深層学習による歩行者の位置推定

提案手法では深層学習を用いて慣性データから歩行者の位置を推定する. 以降では, センサデータの  $i$  サンプル目を観測した時刻  $t_i$  における 3 軸加速度, 3 軸角速度のそれぞれを  $\mathbf{a}^{t_i}$ ,  $\mathbf{w}^{t_i}$  で示す. また, 加速度と角速度を時刻  $t_m$  から  $t_n$  ( $t_m < t_n$ ) まで並べた  $\mathbf{I}^{(t_m, t_n)}$  を式 (1) で示す. また, 時刻  $t$  における歩行者の正解位置は  $\mathbf{P}^t$  で示し, 時刻  $t_m$  と  $t_n$  の変位は式 (2) の  $\Delta \mathbf{P}^{(t_m, t_n)}$  で示す.  $\Delta l^{(t_m, t_n)}$  と  $\Delta \phi^{(t_m, t_n)}$  はそれぞれ歩行者の移動距離と進行方向の変位を示す. また, 時刻  $t_m$  から  $t_n$  までのデータのサンプル数は  $N (= m - n + 1)$  とする.

$$\mathbf{I}^{(t_m, t_n)} = \begin{pmatrix} \mathbf{a}^{t_m} & \mathbf{a}^{t_{m+1}} & \dots & \mathbf{a}^{t_n} \\ \mathbf{w}^{t_m} & \mathbf{w}^{t_{m+1}} & \dots & \mathbf{w}^{t_n} \end{pmatrix} \quad (1)$$

$$\Delta \mathbf{P}^{(t_m, t_n)} = \left( \Delta l^{(t_m, t_n)}, \Delta \phi^{(t_m, t_n)} \right) \quad (2)$$

提案手法では入力を  $\mathbf{I}^{(t_m, t_n)}$  として,  $\Delta \mathbf{P}^{(t_m, t_n)}$  を出力とする深層学習モデルを考える. 入力にセンサデータを一定の時間幅の系列に分割しそれを時系列方向にスライドさせ, それぞれにおいて特徴量を抽出するスライディングウィンドウ方式を採用する. そこで, 一定時間幅におけるデータのサンプル数を等しくするために取得した時系列データは全て 100Hz にリサンプリングしてから使用する. 時間幅は調整する必要があるが, 一般的には数秒程度に設定されることが多い. 数秒程度の時間幅の変位を正解データに用いると図 2a のように, 実際の経路とのギャップが大きくなり深層学習モデルがうまく学習できない可能性が高

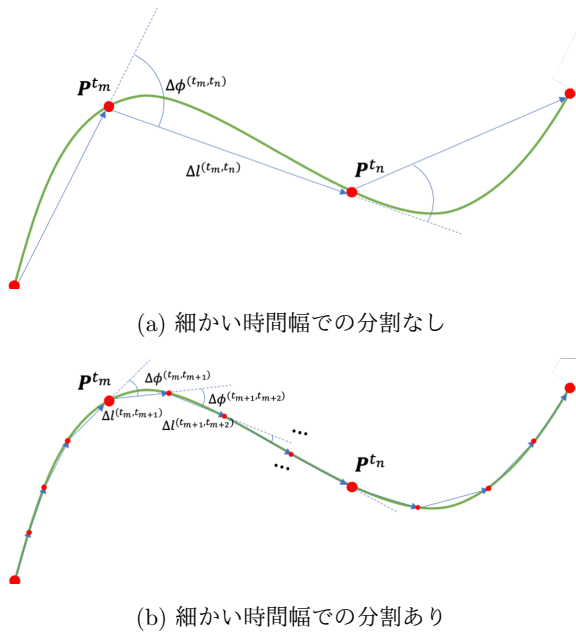


図 2: 時刻  $t_m$  から  $t_n$  の間の変位

Fig. 2 Displacement between time  $t_m$  and  $t_n$

い。一方で、図 2b のように時間幅を小さく設定すると実際の経路に近づくことがわかる。そこでは我々は時刻  $t_m$  から  $t_n$  の間でさらに細かく 1 サンプルごとの変位を求め、式 (3), (4) に従い平均をとり、それを正解データとした。しかし、平均をとることによって変位の時系列方向に移動平均フィルタをかけることになる。変位の時系列データのサンプリングレートを  $f_s$  とした時、振幅特性は式 (5) で表される。カットオフ周波数は振幅特性が  $1/\sqrt{2}$  となる周波数と定義されているため、式 (6) で表せる。式 (6) からわかるように  $f_s$  が一定である場合、 $N$  が大きくなるとともにカットオフ周波数が低くなり、正解データから高周波の情報が失われるため時間幅は慎重に決める必要がある。

$$\Delta l^{(t_m, t_n)} = \frac{1}{N} \sum_{i=m}^{n-1} \Delta l^{(t_i, t_{i+1})} \quad (3)$$

$$\Delta \phi^{(t_m, t_n)} = \frac{1}{N} \sum_{i=m}^{n-1} \Delta \phi^{(t_i, t_{i+1})} \quad (4)$$

$$G = \frac{f_s}{2\pi N f_c} \sqrt{2(1 - \cos \frac{2\pi N f_c}{f_s})} \quad (5)$$

$$f_c = \frac{0.443}{N} f_s \quad (6)$$

図 3 は深層学習モデルが時刻  $t_m$  から  $t_n$  間の歩行者の変位を推定する際のデータフローである。  $I^{(t_m, t_n)}$  の時系列の特徴量を LSTM (Long short-term memory) を用いて学習し、  $\Delta P^{(t_m, t_n)}$  の出力には全結合層を用いる。経路の推定は図 4 のように時間幅を 1 サンプルずつずらし推定した変位を積分して求める。

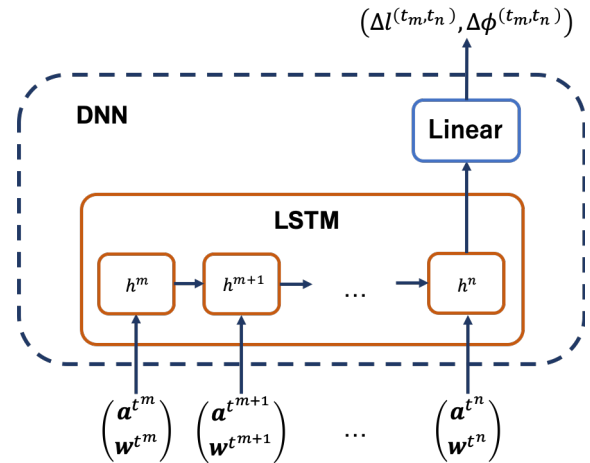


図 3: LSTM を用いた深層学習モデルによる時刻  $t_m$  から  $t_n$  の歩行者の変位の推定

Fig. 3 Estimation displacement from  $t_m$  to  $t_n$  by using LSTM based model

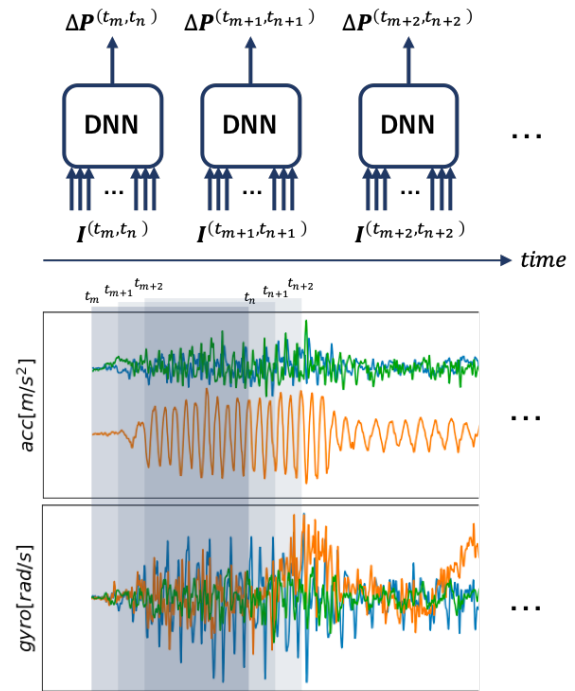


図 4: 時間幅を 1 サンプルずつずらして推定する流れ

Fig. 4 Estimation flow for each time sequence

### 3.2 データ収集と正解データの生成

歩行者の移動速度と進行方向の変位の推定を行うモデルの学習のためには、端末の正確な位置情報と入力データとなる加速度、角速度が必要となる。そこで本稿では正解位置情報を Google [16], 入力データは HASC Logger[17] を用いて取得する。図 5 は実際にデータを収集している様子を示す。Google Tango は加速度や角速度に加えてカメラや深度センサを用いることで、端末の向きや移動量を高精度に推定でき、計測開始時を原点として相対位置を取得できる。本研究では 2 次元空間での位置推定を行うため、3 次元位置情報のままではモデルの学習に使えない。そこ



図 5: データ計測の様子

Fig. 5 Data collection

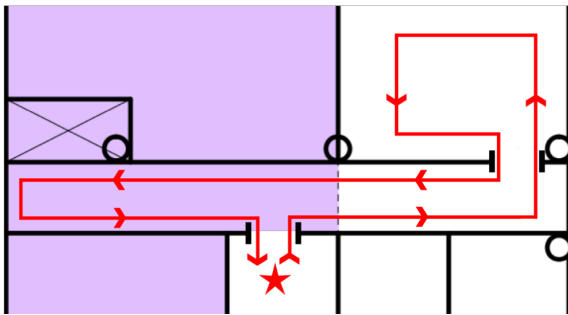


図 6: 経路の例

Fig. 6 Route example

で 3 次元位置ベクトルから水平位置ベクトルを計算する必要がある。まず、3 次元正解位置情報と同時に取得した加速度と角速度データからカルマンフィルタを用いて重力方向を推定する。そして求めた重力方向  $\mathbf{g}$  と 3 次元位置ベクトル  $\mathbf{r}$  を用いて以下の式より水平位置ベクトル  $\mathbf{r}_h$  を算出する:

$$\mathbf{r}_h = \mathbf{r} - \frac{\mathbf{g} \cdot \mathbf{r}}{|\mathbf{g}|^2} \mathbf{r} \quad (7)$$

実際に移動を行った経路の例を図 6 に示す。経路は直線と右回り、左回りの方向転換がバランスよく含まれているものを複数用意した。被験者の人数は 11 名で、そのうち 8 名は訓練用データ、1 名は検証用データ、1 名は深層学習モデルの評価用データ、1 名は提案手法と既存手法の比較評価用データとなるように分けた。データの詳細は表 1 に示す。

#### 4. 評価

本章では提案手法と既存手法の精度評価を行う。提案手法で用いる深層学習モデルは LSTM をベースとした構造で、実装は Pytorch 1.3.1 で行った。評価は速度、進行方向、経路における推定値と正解値の誤差を求め、誤差の平均と標準偏差を比較し行う。評価用のデータは 3 章で説明したデータを用いた。深層学習モデルの学習には訓練用データ、検証用データ、深層学習モデルの評価用データを使用し、提案手法と既存手法の比較には提案手法と既存手

表 1: データセットの詳細

Table 1 Dataset details

被験者数	11 名 (男性:6 名, 女性:5 名)
端末保持位置	手持ち, パンツの左右ポケット
行動種別	歩行, 遅歩行, 速歩行, ジョギング
延べ経路数	260
歩行時間 [s]	9132.33
経路の長さ [m]	10494
端末	PHAB2 Pro Xperia XZ1 Moto Z3 Play OPPO R17 Pro HUAWEI nova3
ソフトウェア	Google Tango HASC Logger for Android
センサデータ	加速度 角速度 GPS 磁気 気圧 WiFi

法の比較評価用データを用いた。

#### 4.1 比較手法

提案手法の比較対象として、既存の Step-and-heading algorithm を用いる。入力するセンサデータは、提案手法と条件を揃えるため、100Hz にリサンプリングした加速度、角速度を用いる。これに加え、FFT を用いたローパスフィルタにより一定以上の周波数成分をカットする処理を行う。カットオフ周波数は今回収集したデータセットの端末保持位置ごとに設定し、手持ちでは 10Hz、パンツの左右のポケットでは 8Hz とした。ポケットに入れた端末では手持ちに比べ細かい振動が多く含まれるため、カットオフ周波数を低く設定した。次に、ローパスフィルタをかけた加速度信号を用いてステップ検知を行う。ステップ検知には図 7 に示す有限オートマトンを利用する。図中の  $x$  は加速度ノルムの値であり、状態遷移の条件となる各閾値はそれぞれ Move: 動作しているか否か, Pos Peek: 極大, Neg Peek: 極小, Step End: 歩行の終了を意味する。各閾値は、手持ち、パンツの左右のポケットの各端末保持位置ごとに調整し、表 2 に示す通りに設定した。最後に、検出されたステップから移動速度を推定し位置を算出する。一歩あたりの歩幅は身長  $\times 0.46\text{m}$  で推定し、直前のステップからの経過時間で割り移動速度を算出する。算出された移動速度は最大サンプル数 4 のキューに入れ、その時刻におけるキュー内の平均を移動速度として出力する。進行方向の変位は加速度と角速度からカルマンフィルタを用いて端末姿勢を推定し、推定した端末姿勢から角速度の水平成分を導くことによって算出する。

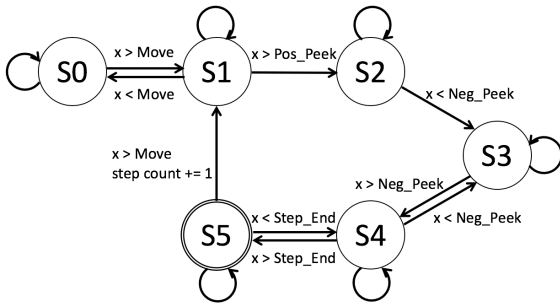


図 7: ステップ検知の有限オートマトン  
Fig. 7 Finite automaton for step detection

表 2: オートマトンの各閾値  
Table 2 Automaton threshold

	手持ち	左右ポケット
Move [ $m/s^2$ ]	1.05	1.05
Pos Peek [ $m/s^2$ ]	1.09	1.13
Neg Peek [ $m/s^2$ ]	0.97	0.89
Step End [ $m/s^2$ ]	0.98	0.96

表 3: 深層学習モデルのパラメータ  
Table 3 Deep learning model parameters

パラメータ	値
入力時系列の長さ [s]	0.5, 1, 2, 3
入力特徴数	6 (3 軸の加速度と角速度)
LSTM の中間層のユニット数	32, 64, 128
LSTM の層数	1, 2
Bidirectional	True, False
バッチサイズ	2048
損失関数	MSELoss
初期学習率	0.001
最適化法	Adam

## 4.2 深層学習モデルの学習

学習にはデータセットの作成時に分けた訓練用データと検証用データを用いる, 評価には深層学習モデルの評価用データを用いる. 学習, 推定を行う深層学習モデルは LSTM と全結合層を組み合わせたモデルを用いる. 使用するパラメータは表 3 に示す. 入力時系列の長さは長くなるほど, 正解データの高周波成分を除去する特性 (式 (6)) があるため, 慎重に調整する必要がある. そこで 0.5, 1, 2, 3 秒の 4 種類で比較を行う. また, LSTM の中間層のユニット数, 層数, Bidirectional LSTM であるかはモデルの学習に大きな影響を与えるため, 最適なパラメータの組み合わせを見つける必要がある. 最適なパラメータは推定経路の誤差平均が最小のものとする. 評価の結果得られた最適なパラメータは表 4 に示す. また, 学習時の訓練用データと検証用データの損失の推移は図 8 に示す.

## 4.3 推定精度の比較

推定精度の評価は速度推定, 進行方向推定, 経路推定の

表 4: 経路推定が最も高精度だったパラメータ

Table 4 Best parameters

パラメータ	値
入力時系列の長さ [s]	2
LSTM の中間層のユニット数	32
LSTM の層数	1
Bidirectional	True

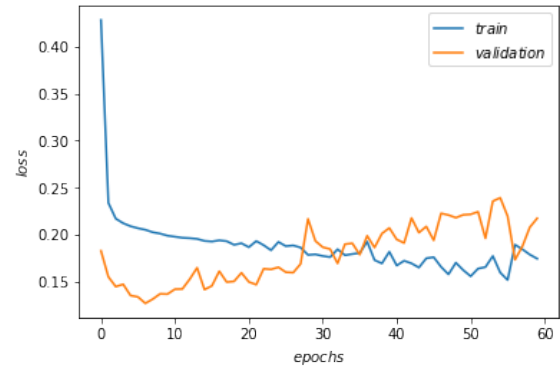


図 8: 訓練用と検証用データの損失の推移

Fig. 8 Loss of training and validation data

3 項目を提案手法, 既存手法について行う. 推定するデータは事前に用意した比較評価用データを用いる. 速度推定の評価結果は表 5 に, 速度推定の評価結果は表 6 に, 経路推定の評価結果は表 7 にそれぞれ示す. 表 5, 6, 7 より, 全ての項目において提案手法の方が誤差の平均と標準偏差が小さく, 推定精度が高いことがわかった.

## 4.4 考察

提案手法が既存手法より速度を高精度に推定できた要因を考察する. 歩行者の速度推定は深層学習を用いることにより歩容に対するロバスト性が向上することはすでに実証されており [5], [6], この歩容に対するロバスト性向上が速度推定の高精度化に繋がったと考えられる.

提案手法が既存手法より進行方向を高精度に推定できた要因を考察する. 表 8 は端末保持位置ごとの進行方向推定の評価を示す. 表より, 全ての端末保持位置において提案手法の推定精度が高いことが確認できる. つまり, 提案手法は既存手法より端末保持位置に対するロバスト性が高い. このことが推定精度向上につながったと考えられる.

最後に提案手法が既存手法より経路を高精度に推定できた要因を考察する. 大きな要因は速度と進行方向の推定精度が向上したことである. 表 9 より, 速度推定の歩容に対するロバスト性の向上が経路推定において反映されていることがわかる. また, 表 10 より, 進行方向推定の端末保持位置に対するロバスト性の向上も経路推定において反映されていることがわかる. もう一つの要因は提案手法では速度と進行方向の変位を一つの深層学習モデルで推定でき

表 5: 速度推定の評価  
Table 5 Evaluation of speed estimation

手法	誤差 [m/s]	
	平均	標準偏差
提案手法	<b>0.22</b>	<b>0.25</b>
既存手法	0.44	0.44

表 6: 進行方向推定の評価  
Table 6 Evaluation of heading estimation

手法	誤差 [rad]	
	平均	標準偏差
提案手法	<b>0.49</b>	<b>0.41</b>
既存手法	0.69	0.59

表 7: 経路推定の評価  
Table 7 Evaluation of route estimation

手法	誤差 [m]	
	平均	標準偏差
提案手法	<b>3.24</b>	<b>2.75</b>
既存手法	3.93	2.91

表 8: 端末保持位置ごとの進行方向推定の評価  
Table 8 Evaluation of heading estimation for each device position

端末保持位置	手法	誤差 [rad]	
		平均	標準偏差
手持ち	提案手法	<b>0.36</b>	<b>0.27</b>
	既存手法	0.56	0.50
右ポケット	提案手法	<b>0.55</b>	<b>0.43</b>
	既存手法	0.81	0.67
左ポケット	提案手法	<b>0.58</b>	<b>0.49</b>
	既存手法	0.72	0.57

る点である。既存手法では速度と進行方向の変位を別々のプロセスで推定していたため、速度と進行方向の変位の関係を考慮していなかったが、提案手法では2つの関係を考慮して推定を行える。図9はある経路における速度と進行方向の推定結果を示したものである。提案手法の推定した速度と進行方向の結果を見ると進行方向が変化しているとき、推定する速度が対応して減速していることがわかる。このように提案手法では速度が減速した時は進行方向が変位し、進行方向が変位している時は速度が減速するといった情報を用いて推定結果を補完し合える。図10より、経路推定の結果からも進行方向が変位するタイミングで生じる誤差が提案手法の方が小さいことがわかる。

## 5. まとめ

本稿では、加速度と角速度の時系列入力から、速度と進行方向をEnd-to-endで推定するLSTMベースの手法を提案した。また、提案手法と既存手法であるStep-and-heading

表 9: 歩容ごとの推定経路の評価  
Table 9 Evaluation of speed estimation for each gait

歩容	手法	誤差 [m]	
		平均	標準偏差
歩行	提案手法	<b>2.74</b>	<b>2.30</b>
	既存手法	2.79	2.58
遅歩行	提案手法	<b>3.55</b>	<b>2.64</b>
	既存手法	4.91	3.12
速歩行	提案手法	<b>2.96</b>	<b>2.67</b>
	既存手法	3.22	2.70
ジョギング	提案手法	<b>3.50</b>	3.31
	既存手法	4.18	<b>2.31</b>

表 10: 端末保持位置ごとの推定経路の評価  
Table 10 Evaluation of route estimation for each device position

端末保持位置	手法	誤差 [m]	
		平均	標準偏差
手持ち	提案手法	<b>2.16</b>	<b>1.36</b>
	既存手法	3.47	2.61
右ポケット	提案手法	<b>3.98</b>	<b>3.20</b>
	既存手法	4.63	3.32
左ポケット	提案手法	<b>3.79</b>	3.08
	既存手法	3.81	<b>2.73</b>

algorithmの推定精度を比較することによって、提案手法の方が速度、進行方向、経路において高精度な、さらに歩容と端末保持位置に対してロバストな推定を行えるとわかった。今後は学習データのバリエーションと量を増やすことにより、推定精度と歩容と端末保持位置に対するロバスト性のさらなる向上を目指すと共に、既存のPDRでは推定が難しかった動きの推定も行えるようにしていきたい。

謝辞 本研究は、JSPS 科研費 (JP17H01762)、総務省SCOPE、NICT 委託研究により支援していただいております。

## 参考文献

- [1] Harle, R.: A Survey of Indoor Inertial Positioning Systems for Pedestrians, *IEEE Communications Surveys Tutorials*, Vol. 15, No. 3, pp. 1281–1293 (2013).
- [2] Correa Vila, A., Barceló, M., Morell, A. and Lopez Vicario, J.: A Review of Pedestrian Indoor Positioning Systems for Mass Market Applications, *Sensors (Switzerland)*, Vol. 17 (2017).
- [3] Sakpere, W., Oshin, M. A. and Mlitwa, N.: A State-of-the-Art Survey of Indoor Positioning and Navigation Systems and Technologies, *South African Computer Journal*, Vol. 29, No. 3 (online), available from <http://sacj.cs.uct.ac.za/index.php/sacj/article/view/452> (2017).
- [4] Ban, R., Kaji, K., Hiroi, K. and Kawaguchi, N.: Indoor positioning method integrating pedestrian Dead Reckoning with magnetic field and WiFi fingerprints, *2015 Eighth International Conference on Mobile Computing*

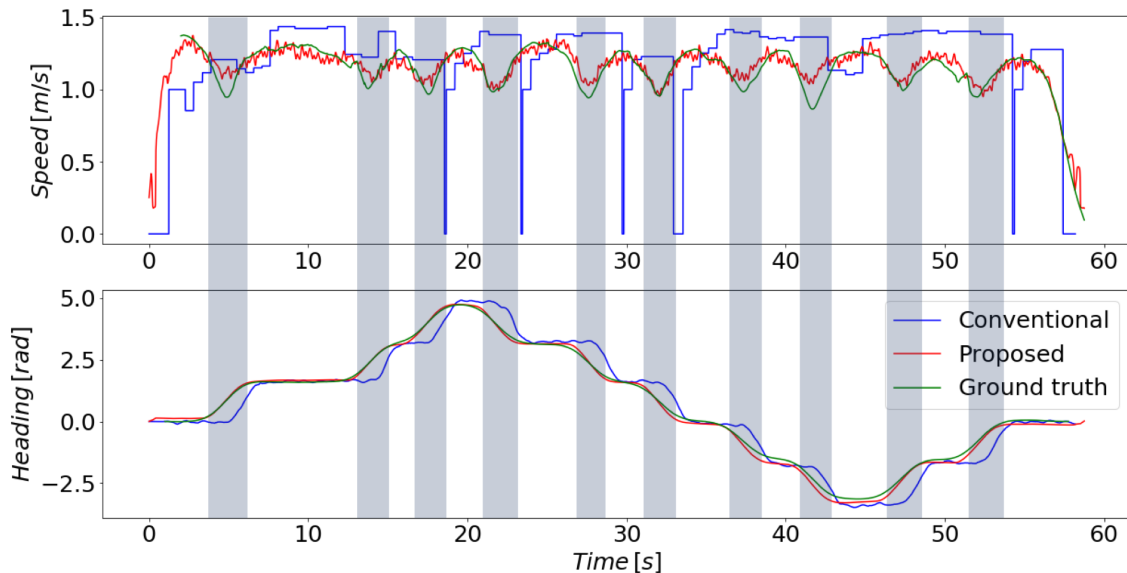


図 9: 進行方向の変化するときに速度が低下する関係を提案手法が捉えている例

**Fig. 9** Example in which the proposed method captures the relationship that the speed decreases when the heading changes

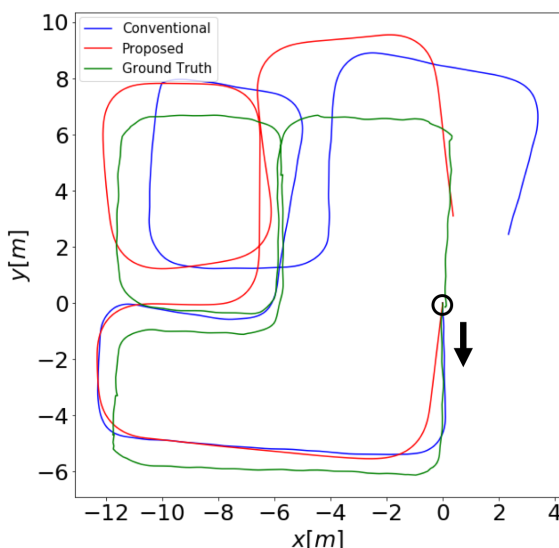


図 10: 図 9 の推定経路

**Fig. 10** Path of Fig. 9

and Ubiquitous Networking (ICMU), pp. 167–172 (online), DOI: 10.1109/ICMU.2015.7061061 (2015).

- [5] Yoshida, T., Nozaki, J., Urano, K., Hiroi, K., Yonezawa, T. and Kawaguchi, N.: Gait Dependency of Smartphone Walking Speed Estimation Using Deep Learning, *Proceedings of the 17th Annual International Conference on Mobile Systems, Applications, and Services, MobiSys '19*, New York, NY, USA, ACM, pp. 641–642 (2019).
- [6] Kawaguchi, N., Nozaki, J., Yoshida, T., Hiroi, K., Yonezawa, T. and Kawaguchi, N.: End-to-End Walking Speed Estimation Method for Smartphone PDR using DualCNN-LSTM (2019).
- [7] Zhang, W., Li, X., Wei, D., Ji, X. and Yuan, H.: A foot-mounted PDR system based on IMU/EKF+HMM+ZUPT+ZARU+HDR+compass algorithm, *2017 International Conference on Indoor*

*Positioning and Indoor Navigation (IPIN)*, pp. 1–5 (2017).

- [8] Alzantot, M. and Youssef, M.: UPTIME: Ubiquitous pedestrian tracking using mobile phones, *2012 IEEE Wireless Communications and Networking Conference (WCNC)*, pp. 3204–3209 (2012).
- [9] Gu, F., Khoshelham, K., Yu, C. and Shang, J.: Accurate Step Length Estimation for Pedestrian Dead Reckoning Localization Using Stacked Autoencoders, *IEEE Transactions on Instrumentation and Measurement*, pp. 1–9 (2018).
- [10] Xu, H., Gao, Y., Yu, F. and Darrell, T.: End-to-End Learning of Driving Models from Large-Scale Video Datasets, *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3530–3538 (2017).
- [11] Kai Wang, Babenko, B. and Belongie, S.: End-to-end scene text recognition, *2011 International Conference on Computer Vision*, pp. 1457–1464 (2011).
- [12] Levine, S., Finn, C., Darrell, T. and Abbeel, P.: End-to-end Training of Deep Visuomotor Policies, *J. Mach. Learn. Res.*, Vol. 17, No. 1, pp. 1334–1373 (online), available from <http://dl.acm.org/citation.cfm?id=2946645.2946684> (2016).
- [13] Hannun, A., Case, C., Casper, J., Catanzaro, B., Diamos, G., Elsen, E., Prenger, R., Satheesh, S., Sengupta, S., Coates, A. and Y. Ng, A.: DeepSpeech: Scaling up end-to-end speech recognition (2014).
- [14] Ordóñez, F. J. and Roggen, D.: Deep Convolutional and LSTM Recurrent Neural Networks for Multimodal Wearable Activity Recognition, *Sensors*, Vol. 16, No. 1 (online), available from <http://www.mdpi.com/1424-8220/16/1/115> (2016).
- [15] Peng, L., Chen, L., Ye, Z. and Zhang, Y.: AROMA: A Deep Multi-Task Learning based Simple and Complex Human Activity Recognition Method Using Wearable Sensors, *ACM Interactive, Mobile, Wearable and Ubiquitous Technologies*, Vol. 2, No. 2, pp. 74:1–74:16 (online), DOI: 10.1145/3214277 (2018).

- [16] Roberto, R., Lima, J. P., Araújo, T. and Teichrieb, V.: Evaluation of Motion Tracking and Depth Sensing Accuracy of the Tango Tablet, *2016 IEEE International Symposium on Mixed and Augmented Reality (ISMAR-Adjunct)*, pp. 231–234 (2016).
- [17] Kaji, K., Watanabe, H., Ban, R. and Kawaguchi, N.: HASC-IPSC: indoor pedestrian sensing corpus with a balance of gender and age for indoor positioning and floor-plan generation researches (2013).