

# LSTM を用いた高精度歩行者測位方法に関する検討

井上真樹 湯素華 小花貞夫

**概要:** 自動運転の実現に向けて交通事故防止が大きな課題となっている。見通し外の歩行者との事故を回避するために、歩行者が自身の所有する端末から位置情報を含むパケットを送信し周囲の車両に知らせる歩車間通信が提案されている。歩行者位置の測位には一般的に GPS が用いられるが、都市部では建物の遮蔽などにより測位精度が大幅に劣化する。この問題を解決するために、GPS に加えて車両を測位の基準点とし、チャネル状態情報 (CSI) から歩車間距離を推測して歩行者位置を算出する測位方式が提案されている。本稿では、歩行者と車両の位置関係が連続的に変化していくことに着目し、従来の SVR (Support Vector Regression) を利用して瞬間 CSI から歩車間距離を算出する代わりに、LSTM (Long short-term memory) ネットワークを用いた深層学習を介して CSI の時系列変化から歩車間距離を高精度に推測する手法を検討し、測距・測位精度を評価した。3D レイトレーシングを用いたシミュレーションにより、LSTM を用いた検討方式は SVR を利用した先行方式に比べ、平均歩車間距離誤差を 44.9%、平均水平測位誤差を 46.9%削減可能であることを確認した。

## LSTM-based High Precision Pedestrian Positioning

MASAKI INOUE SUHUA TANG SADAO OBANA

### 1. はじめに

2021 年現在、政府は官民 ITS 構想・ロードマップのひとつとして自動運転の実現を掲げており、2030 年に一部機能の実現を目標としている[1]。また、交通事故防止のために、ドライバーの運転を補助する先進運転支援システムはすでに実用化がなされている。ところが、内閣府の交通安全白書によると、状態別交通事故死者数の内、歩行者の死者数は 36.6%と最も高く、歩行者との衝突事故をどのように減らすのが課題となっている[2]。歩車間事故防止策は複数あるが、歩行者が携帯端末で測位した自位置を周囲の車両へ通知する歩車間通信方式が有力視されている。この方式の利点としては車載センサ (カメラ、LiDAR など) では不可能とされる死角にいる歩行者の検知が容易になり、出会い頭の事故などの防止につなげることができる。

歩車間通信を実現するには高精度な歩行者測位が重要であり、一般的に GPS をはじめとした GNSS (global navigation satellite system) が使用されるが、都市部では、建物遮蔽・反射による衛星数不足・測位誤差増大問題が頻発するため、測位精度の安定性に欠ける。この問題に対し、GNSS による測位に加えて車両を測位の基準点とし、車両同士が車車間通信で自身の位置情報等を周辺車両へ配信するパケットを歩行者の携帯端末が傍受し、瞬間のチャネル状態情報 (CSI) から歩車間距離を推測することで、歩行者の測位精度を高めるという方式が提案されている[3]。CSI から歩車間距離の推測には、線形回帰モデルや[3]、SVR

(Support Vector Regression) [4]が使用されている。これらの方式では、歩車間距離の精度が歩行者位置の測位精度に大きく関わっており、高精度な歩車間距離の測距モデルが必要とされている。

本稿では、歩行者と車両の位置関係が連続的に変化していくことに着目し、従来の SVR を利用して瞬間 CSI から歩車間距離を算出する[4]代わりに、CSI の時系列変化から LSTM (Long short-term memory) ネットワークを用いた歩車間距離の推測手法を検討し、さらに Kalman filter を用いた歩行者測位手法を検討・評価したので、その結果を報告する。

### 2. 先行研究

#### 2.1 衛星による歩行者測位

歩行者の測位では、GPS 衛星を用いた手法が一般的である。歩行者の 3 次元座標と信号受信機の時刻誤差の 4 つを未知変数とした際、位置  $(x_k, y_k, z_k)$  にある GPS 衛星  $k$  と位置  $(x, y, z)$  にいる歩行者について衛星-歩行者間の距離  $d_k$  は、

$$\sqrt{(x-x_k)^2+(y-y_k)^2+(z-z_k)^2}+c\Delta t=d_k \quad (1)$$

と表すことができる。ただし、 $c$  は光速、 $\Delta t$  は歩行者受信機の時刻誤差である。4 機以上の衛星の信号から得られた衛星の位置情報と、それぞれの衛星-歩行者間の疑似距離から解を求めることにより、歩行者の位置を決定する。しかし都市部の場合、高層ビル等の影響により信号が遮断され、使用できる衛星数が大きく減少する。GPS 以外の衛星を用

いた方式[5]や、準天頂衛星（日本の上空に常に存在する衛星）を用いた方式[6]も存在するが、信号を受信できる衛星は建物の影響を受けづらい天頂付近に集中することから、多角的な測位を行えないため測位精度の改善は限定的なものになる。また、信号を受信できた場合にも、ビル等で反射した信号（マルチパス）を受信する影響で、歩衛間の距離推測に誤差が発生し測位精度が低下する問題[7]がある。

## 2.2 車両による歩行者測位

### 2.2.1 概要

歩行者の測位精度向上のため、自動運転及び安全運転支援システム下で車車間通信により車両が互いの位置情報を交換している環境において、車車間通信の信号を歩行者が所持する携帯端末が傍受し、これを測位に利用する手法が提案されている[3]。この手法では、GPS 信号に加えて、車両を測位の基準点として扱い、車両から受信信号の時間・周波数ごとの電波強度や位相を表す CSI と呼ばれる情報を受け取り、それをもとに歩車間距離を推測して歩行者位置を算出する（図 1）。歩行者付近の車両を用いて測位を行うため、建物による電波の遮蔽などの影響を受けにくいことから、特に GPS のみを使用した測位と比較して都市部における測位精度を大きく向上させることに成功している。

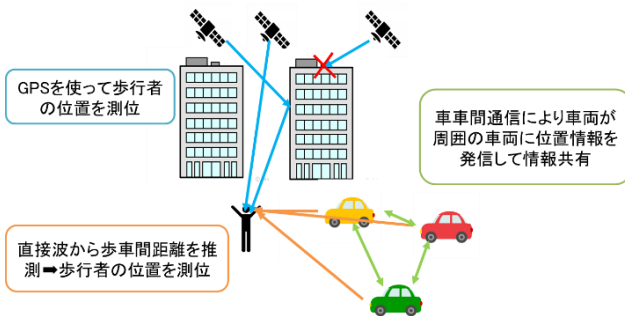


図 1 先行方式[3]の概要

### 2.2.2 歩車間距離の導出

歩行者が車両から受信した CSI には、直接波と反射波・回折波が混在する。反射波や回折波は余分な減衰が発生しているため、正確な距離推定を行うためには直接波のみを利用することが必要である。先行方式[3]では、CSI から取得した時間軸の信号のうち、最初に到達した信号（第一波）を直接波として用いることで歩車間距離推定を行っている。

また、CSI から取得した直接波の電波強度から、伝搬距離（歩車間距離）を算出する。このとき、直接波の電波強度  $L$  と距離  $d$  には以下の式 (2) が成り立つ。

$$L = a \log_{10} d + b. \quad (2)$$

先行方式[3]において、定数  $a, b$  は既知の直接波の電波強度と伝搬距離の組のデータから線形回帰を行って導出する。また、式 (2) の両辺を微分すると以下の式を得られる。

$$\Delta d = \ln \frac{d}{a} \cdot \Delta L. \quad (3)$$

式 (3) より、歩車間距離  $d$  が大きくなると歩車間距離の誤差である  $\Delta d$  も大きくなることがわかる。そのため、測距モデルの作成に使用するデータの歩車間距離には一定の上限を設けることが好ましいと考えられる。

また、歩車間距離  $d$  と歩車間距離の誤差  $\Delta d$  との間の相関関係は線形回帰以外に SVR で測定を行う方法存在する[4]。

### 2.2.3 歩行者の測位演算

周囲の車両から受信した電波のうち、直接波があると判別されたものと、GPS 衛星からの電波を併用して、歩行者の測位を行う。歩行者の位置は以下の式 (4) ~ (8) で算出される。ここで  $x$  は歩行者の 3 次元の位置と時刻誤差とする。初期の歩行者の位置の ENU 座標が  $(e, n, u)$ 、時刻誤差を  $\Delta t$  としたとき、 $\hat{x}_0$  は、

$$\hat{x}_0 = (e, n, u, \Delta t)^T \quad (4)$$

と表せる。

次に、歩行者の移動を表す方向ベクトル  $\vec{H}$  は、

$$\vec{H} = \partial d(x) / \partial x |_{x=x_0} \quad (5)$$

となる。ここで、 $d(x)$  は歩行者の位置と衛星・車両位置との距離のベクトルである。さらに、 $y$  を歩行者と衛星・車両との距離とすると、歩衛間の疑似距離  $y_{\text{衛星}}$ 、歩車間の推測距離  $y_{\text{車両}}$  はそれぞれ以下のような式で表すことができる。

$$y_{\text{衛星}} = d(x) + \delta + \varepsilon \approx d(x_0) + \vec{H} \cdot (x - x_0) + \delta + \varepsilon, \quad (6)$$

$$y_{\text{車両}} = d(x) + \varepsilon \approx d(x_0) + \vec{H} \cdot (x - x_0) + \varepsilon. \quad (7)$$

ただし、 $\delta$  は歩行者の時刻誤差による距離誤差、 $\varepsilon$  は測距時に発生する誤差である。これらの情報を基に、時刻  $i+1$  における歩行者の位置  $\hat{x}_{i+1}$  は式 (8) のように求めることができる。

$$\hat{x}_{i+1} = \hat{x}_i + (\vec{H}^T \cdot Q_y^{-1} \cdot \vec{H})^{-1} \cdot \vec{H}^T \cdot Q_y^{-1} \cdot (y - d(\hat{x}_i) - \delta_i). \quad (8)$$

$Q_y$  は事前に設定された重みである。先行方式[3]において、重みは距離誤差の分散値から求められ、GPS 衛星の仰角や歩車間距離によって変化する。

## 2.3 LSTM ネットワーク

LSTM は Recurrent Neural Network (RNN) の一種である。RNN とは、時系列データを扱うためのモデルとして考案された再帰的構造をもったニューラルネットワークである。特徴としては、前の計算時での中間の状態を次の入力として扱うことで、時系列データの履歴を保持しながら新たな学習を行うことができる。しかし、通常の RNN の場合は、時系列を長期的にさかのぼった学習をさせようとした際に、

多くの状況で勾配が収束しなくなってしまうという問題が発生する。

時系列データにおける長期間の情報保持を可能にするため、Sepp Hochreiter らは LSTM ネットワークを RNN の拡張方式として考案した[8]。従来型の RNN では入力  $x_t$  から  $y_t$  を予測する際、前ステップで予測した  $y_{t-1}$  の情報をそのまま入力として扱っていたが、LSTM では前の出力を LSTM ユニットの場所を経由させることで情報の長期依存性を保つことができる(図 2)。また、勾配消失をさらに抑制するため、Felix らにより改良がなされた**エラー! 参照元が見つかりません。**

LSTM ユニットの仕組みは図 3 のようになっている。モデル内部には忘却ゲート、入力ゲート、出力ゲートが存在する。それぞれのゲートの持つ機能として、忘却ゲートでは状態  $s$  の持つ情報をどの程度保持するかを調整を行い、入力ゲートでは  $s$  が新しい値によりどの程度影響を受けるかの調整を行い、出力ゲートでは  $s$  の値が LSTM ユニットの出力活性化にどの程度影響を与えるのかを調整する。これら 3 つのゲートにはいずれもシグモイド関数を使用し、入力時には、入力ゲートの他に  $y$  の情報を使用する際にもシグモイド関数を使用する。

現在 LSTM には、株式市場の価格予測プログラム[10]や、音声認識システム[11]などの主に連続したデータの分析、予測などの幅広い分野で使用されている。中でも、BLE を利用した屋内測位方法[12]では、LSTM を使用した End-to-end のニューラルネットワークによる測位方法が高い効果を示している。本稿と[12]ではどちらも LSTM を用いた測位方式であるが、[12]は歩行者位置算出の段階で LSTM を使用し、本稿では歩車間距離算出に LSTM を使用する。

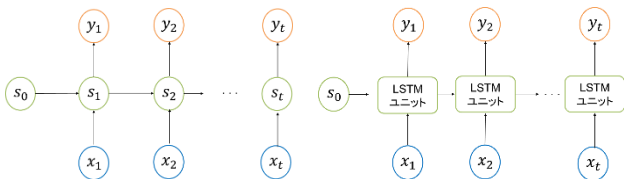


図 2 RNN 及び LSTM 概略図 (左: RNN, 右: LSTM)

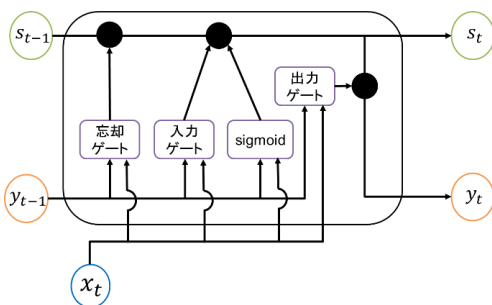


図 3 LSTM ユニットの構造

### 3. 検討方式

#### 3.1 概要

検討方式では車両を基準点とした歩行者測位方式に着目する。時間経過によって歩行者及び車両は位置を変化させていくが、その動きは連続的なものである。そのため、ある時刻の歩行者と車両の距離は現在の CSI のみならず、少し前の時刻の歩行者・車両間の CSI にも依存することが考えられる。

検討方式では、LSTM ネットワークにより、歩行者と車両の時系列データを用いて歩車間測距モデルを作成し、それにより得られた歩車間距離情報から歩行者位置を推測する。検討方式における距離推測の大まかな流れは図 4 のようになっている。まず、測位を始める前にオフラインステップとして、事前に測定した CSI と歩車間距離の真値を用いて LSTM ネットワークを作成し、CSI から適切な歩車間距離を推測するモデルを作る。次にオンラインステップとして作成したモデルを使用して測位地点の CSI から未知の歩車間距離を推測する。図に示していないが、最後に得られた歩車間距離に対して測位演算を行い歩行者の位置を決定する。次節以降より詳しい手順を記述していく。

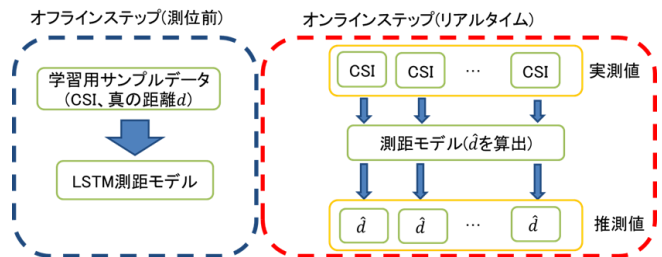


図 4 検討方式における距離推測

#### 3.2 前提条件

前提条件として以下を設ける。

- (1) 車両は無線機を搭載し位置情報を周期的に配信する。
- (2) 歩行者は車両の電波を受信可能な端末を所持する。
- (3) 車両の位置は正確である。
- (4) 測位には衛星の測位情報は使用しない。

(1) については、車車間通信が一般に普及した際、お互いが位置情報を含むパケットを交換し合うという状況を想定している。(2) については、歩行者の持つ端末が、車車間通信に使用される信号を傍受し、信号から算出した CSI から距離を推測し、測位演算を行うということである。このとき、歩行者自身の体は車両から受け取る電波を遮蔽しないものとする。(3) については、車両位置の測位では、歩行者測位で使われる方式以外にも、自律航法 (Dead Reckoning)、車速パルス、マップマッチング、カメラによる車線検知などの方式が挙げられ、歩行者の測位に対して非常に高い精度で測位が可能であると考えられるため、車両の位置は正確であると仮定している。(4) については歩

車間距離の改善による歩行者測位精度を検証するために、GPS 衛星を使用せず車両単独での測位を行う。理由としては、GPS 衛星での測位を組み合わせた際に、車両に比べ GPS 衛星での誤差が大きくなり、検討方式による測位誤差の改善効果がわかりにくくなってしまうのを防ぐためである。

### 3.3 LSTM ネットワークを用いた測距モデルの構築

この節では事前に測定した CSI と歩車間距離を用いて LSTM による測距モデルを作成する方法を述べる。

#### 3.3.1 入力用シーケンスの作成

検討方式において、歩行者は周辺車両から車車間通信信号を傍受することにより各車両との間の CSI を取得する。先行方式では各 CSI と歩車間距離のペアのデータを時系列データでなく、独立した学習用データとして利用する。しかし、それにより前後のデータの相関性が崩れてしまうという問題が存在する。検討方式では、車両ごとに、その CSI を時系列順に格納したものをシーケンスとして扱うことで、前後のデータに関連性が現れるようにした。以下図 5 に検討方式での入力データのシーケンスを示す。

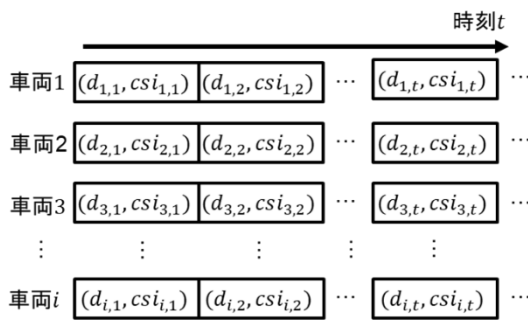


図 5 入力データの作成

また、LSTM における学習では入力データの標準化を行っている。検討方式で入力として読み込むのは直接波とみなす成分の信号強度と位相の情報であり、この 2 種のデータについて偏りがでないよう、データ  $X$  の平均値を  $a$ 、標準偏差を  $b$  として式 (9) を適用する。

$$X' = \frac{X - a}{b}. \quad (9)$$

#### 3.3.2 LSTM を含めた距離推定モデル

LSTM を含めた距離推定モデルは次のようになっている。まず、CSI の信号強度と位相の 2 種類の入力を受け取るためのサイズ 2 のシーケンス入力層、次に隠れユニット数が  $n$  ( $n$  はシミュレーションにより決定) の LSTM 層、そして出力サイズが 1 の全結合層、最後に回帰出力層を設定した。

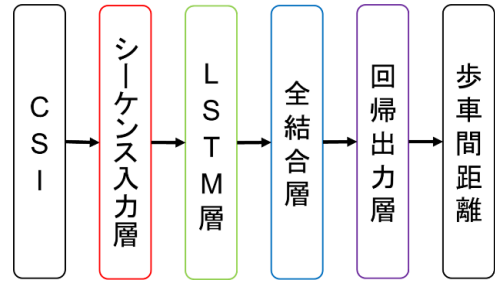
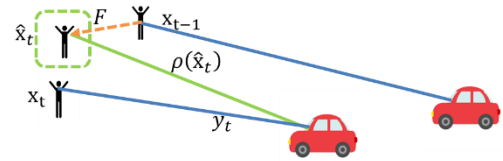


図 6 LSTM を含めた距離推定モデル

#### 3.4 カルマンフィルタを利用した測位演算

測位に使用する重みづけの方法として、検討方式ではカルマンフィルタを採用した。カルマンフィルタとは、誤差のある測定値を用いて動的システムの状態を推定するものである。状態（ここでは歩行者の位置）を  $\mathbf{x} = (x, y, z)^T$ 、測定値（車両が発信する信号の直接波の強度から算出した歩車間距離）を  $\mathbf{y} = (d_1, d_2, \dots, d_n)^T$  とする。このときカルマンフィルタでは予測ステップと更新ステップの二つのステップから、それぞれの工程を経て時刻  $t-1$  での  $\mathbf{x}$  の状態  $\mathbf{x}_{t-1}$  から時刻  $t$  での  $\mathbf{x}$  の状態  $\mathbf{x}_t$  を予測する。



まず、予測ステップにおいて、状態遷移モデルを  $F$ 、状態から算出した歩車間距離を  $\rho(\mathbf{x})$ 、状態ノイズ分散を  $Q$  としたとき、推測歩行者位置  $\hat{\mathbf{x}}_t$  と状態推定誤差分散  $\hat{P}_t$  は、

$$\hat{\mathbf{x}}_t = F\mathbf{x}_{t-1}, \quad (10)$$

$$\hat{P}_t = F\hat{P}_{t-1}F^T + Q, \quad (11)$$

となる。

さらに、状態変換方向ベクトル  $H_t$  を次のように表し、

$$H_t = \left. \frac{\partial \rho(\mathbf{x})}{\partial \mathbf{x}} \right|_{\mathbf{x}=\hat{\mathbf{x}}_t}, \quad (12)$$

測定ノイズ分散を  $R$  とすると、カルマンゲイン  $K_t$  は、

$$K_t = \hat{P}_t H^T (H \hat{P}_t H^T + R)^{-1} \quad (13)$$

となり、更新後の歩行者位置  $\mathbf{x}_t$ 、更新後の状態推定誤差分散  $P_t$  はそれぞれ、

$$\mathbf{x}_t = \hat{\mathbf{x}}_t + K_t(\mathbf{y}_t - \rho(\hat{\mathbf{x}}_t)), \quad (14)$$

$$P_t = (I - K_t H_t) \hat{P}_t. \quad (15)$$

となる。



## 4. シミュレーション評価

### 4.1 シミュレーション環境・条件

シミュレーションにおいて、車両と歩行者の通信の模擬には衛星測位分析ツール“GPS-Studio”[13]の一部として提供される電波伝搬解析ツール“Raplab v10.0”[14]を、測距・測位計算には数値解析ソフトウェア“Matlab”[15]を用いた。また、LSTM については“Matlab”[15]の Deep Learning Toolbox を使用した。

シミュレーションでは歩行者と車両の移動を想定して、0.1s 毎に連続した 200 シーンの世界環境を 2 地点作成し、各シーンで電波伝播シミュレーション・測位計算を行った。

#### 4.1.1 車両・歩行者の配置

シミュレーションには、都市部の環境を想定して銀座付近の 3D 建物データ (NTT データ社製) を用いた。RapLab で 3D 建物データを読み込み、車両・歩行者を、マップ上の道路に沿って配置した。歩行者は各測位地点で、秒速 1m で移動する。また、車両は時速 60km で移動するものとした。シミュレーション地点は 3D マップ内の 2 地点であり、それぞれ地点 A、B とした。図 8 において、緑色が歩行者の位置、赤色が車両の位置を示す。歩車間距離誤差、歩行者の水平測位誤差の計測は地点 B で行い、測距モデルの学習に地点 A のデータを用いてシミュレーションを行う。

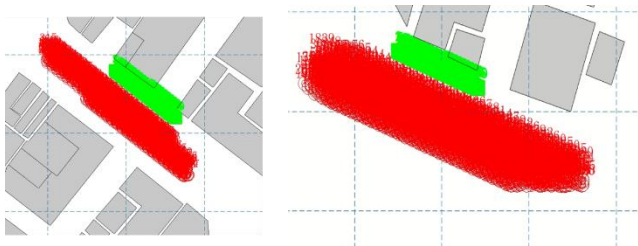


図 8 地点 A (右), 地点 B (左)

表 1 電波伝搬シミュレーションのパラメータ

反射回数	1 回
透過回数	0 回
回折回数	1 回
大地反射	あり
車両の送信周波数	700MHz
車両の送信電力	20dBm

#### 4.1.2 電波伝搬シミュレーション

車両から歩行者への通信の模擬には“Raplab v10.0”を用いた[14]。このソフトウェアは「3D レイトレース法」を使用した電波伝搬の解析ツールである。計算手法は基本的な電磁波理論に従い、レイトレース法の基本要素である反射・回折・透過による電波損失計算を行う。ここでは、4.1.1 節の GPS-Studio で配置した車両・歩行者のそれぞれの位置と NTT 製 3D 地図データを読み込ませ、それを利用して電

波伝搬シミュレーションを行ない、各パスの振幅・位相・到着時間などの情報 (CSI) を出力した。この時のパラメータは表 1 の通りである。

#### 4.1.3 時間分解能の模擬

RapLab では受信機の時間分解能の制限がないため、非常に細かい時間間隔で CSI が得られる。しかし、実際の受信機では時間分解能の制限が存在し、それほど細かい CSI を取得することができない。現実に近いシミュレーションを行うため、RapLab で得られた CSI に一般的な無線 LAN と同じ 50ns の時間分解能を設定し、図 9 で示すように 50ns の範囲内の信号強度値を合成して一つの値として扱った。 $n$  個の信号強度値の合成  $\alpha$  は、以下の式 (16) で表される。

$$\alpha = |(\alpha_1 + \alpha_2 + \dots + \alpha_n) + i(\beta_1 + \beta_2 + \dots + \beta_n)|. \quad (16)$$

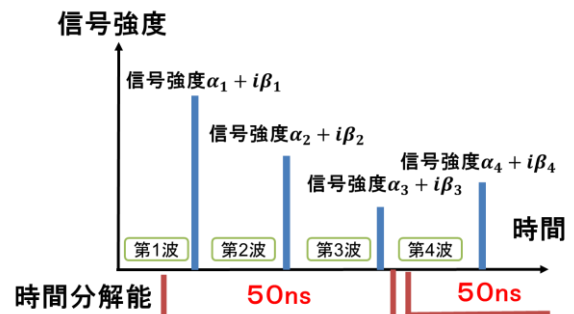


図 9 時間分解能の模擬

#### 4.1.4 距離推定モデルのパラメータ設定

測距モデル作成に使用した入力を CSI の信号強度と位相のシーケンス、出力を歩車間距離に設定した LSTM ニューラルネットワークについて、学習に使用したパラメータを表 2 に示す。また、一部パラメータはシミュレーションにより決定する。

表 2 距離測定のための学習パラメータ設定

バッチサイズ	64
初期学習率	0.01
学習率低下処理	5 エポック毎に 20%低下
シーケンス長	ミニバッチ内の最大のシーケンス長
使用ソルバー	adam
エポック数	シミュレーションで決定
LSTM 隠れユニットの総数	シミュレーションで決定

## 4.2 シミュレーション結果

### 4.2.1 ハイパーパラメータの決定

4.1.4 節で述べたエポック数と隠れユニットの数について、シミュレーションにて決定する。最初に隠れユニット

の数の決定を行う。エポック数は 25 で指定し、隠れユニットの数を 10~150 まで 10 刻みで追加していき、歩車間距離誤差を調べた。以下図 10 と表 3 に結果を示す。

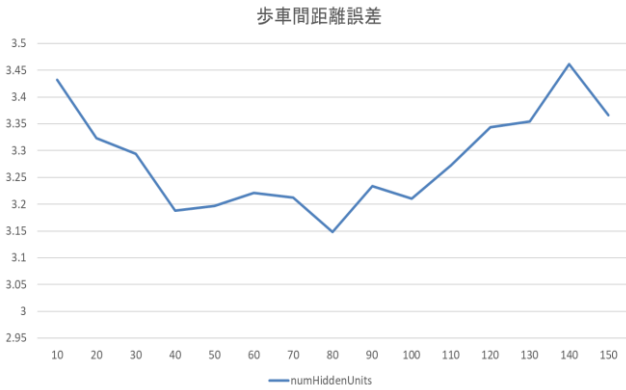


図 10 隠れユニットの総数

表 3 隠れユニットの総数

隠れユニット (個)	10	20	30	40	50
平均距離誤差 (m)	3.43	3.32	3.29	3.19	3.20
隠れユニット (個)	60	70	80	90	100
平均距離誤差 (m)	3.22	3.21	3.17	3.23	3.21
隠れユニット (個)	110	120	130	140	150
平均距離誤差 (m)	3.27	3.34	3.35	3.46	3.66

結果より隠れユニットの数は 80 の時が最も小さくなるのがわかる。

次に、学習時のエポック数を決定する。エポック数は 5~40 エポックにおいて 5 刻みでシミュレーションを行い、比較を行った。隠れユニットの数については、表 3 のシミュレーションで決定した通り 80 個に設定した。以下図 11 と表 4 に結果を示す。

結果より、エポック数が 20 の時最も距離誤差が小さくなるのがわかる。よって学習のエポック数は 20 と定める。

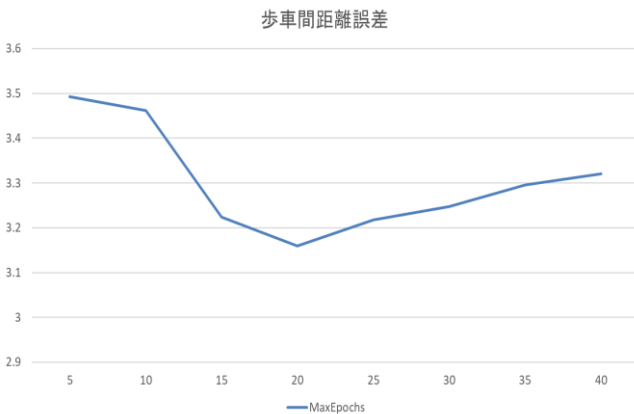


図 11 エポック数

表 4 エポック数

エポック数	5	10	15	20
平均距離誤差 (m)	3.49	3.46	3.22	3.16
エポック数	25	30	35	40
平均距離誤差 (m)	3.22	3.25	3.30	3.32

#### 4.2.2 SVR モデルと LSTM モデルの性能比較

この節では先行方式 (SVR)、検討方式 (LSTM) の 2 種の方式において測距モデルの性能を比較する。検証用の地点 B のデータを用いて SVR, LSTM の交差検証を行って算出した累積度数分布グラフと平均歩車間距離誤差をそれぞれ図 12、表 5 に示す。また、地点 A のデータを用いて学習を行い、地点 B のデータで検証を行った場合についての結果を図 13、表 6 に示す。

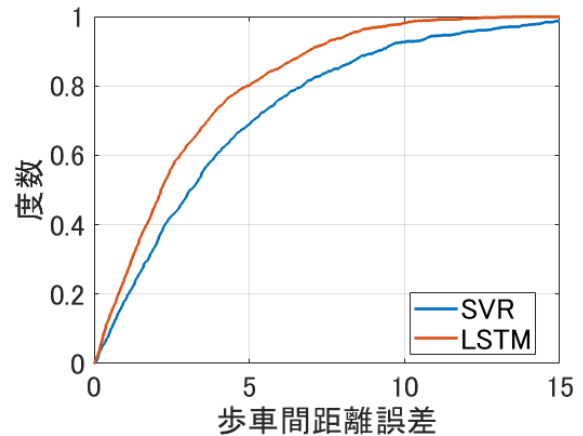


図 12 交差検証による歩車間距離誤差

表 5 平均歩車間距離誤差

方式	平均歩車間距離誤差 (m)
先行方式 (SVR)	4.09
検討方式 (LSTM)	2.96

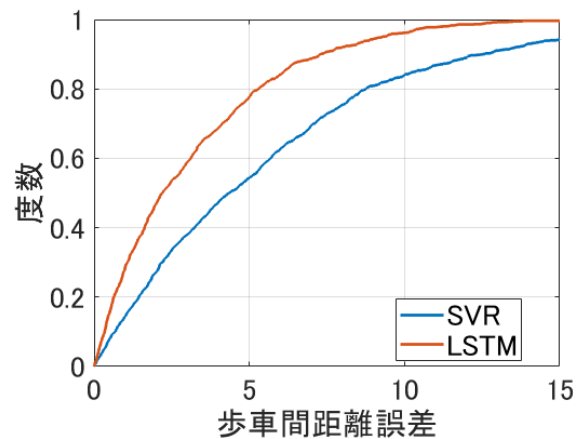


図 13 歩車間距離誤差

表 6 平均歩車間距離誤差

方式	平均歩車間距離誤差 (m)
先行方式 (SVR)	5.73
検討方式 (LSTM)	3.16

結果より、学習データが同シナリオ時、異なるシナリオ時の双方で検討方式での LSTM の測距精度が高くなる結果となった。改善幅は同シナリオ時では 27.6%、異なるシナリオ時では 44.9%となり、異なる環境で特に大きな改善幅を示した。

また、図 13 について、SVR の場合と LSTM の場合について歩車間距離別の測距誤差を比較するとそれぞれ図 14、図 15 のようになる。

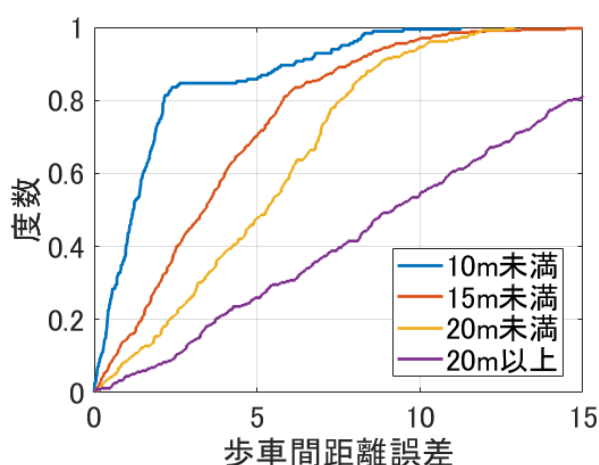


図 14 歩車間距離と測距誤差の関係 (SVR 時)

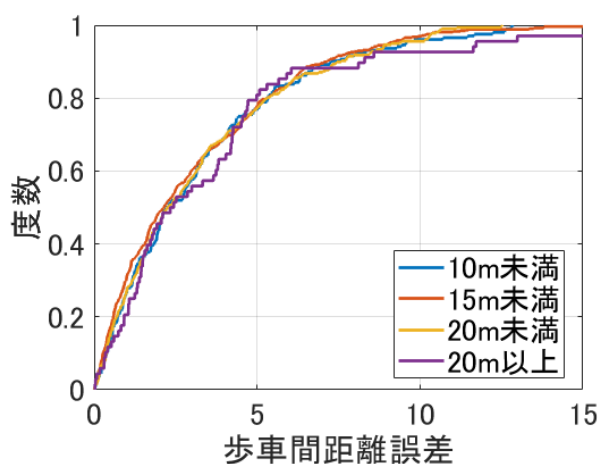


図 15 歩車間距離と測距誤差の関係 (LSTM 時)

図 14、図 15 では、推測した歩車間距離がそれぞれ 10m 未満、10m 以上 15m 未満、15m 以上 20m 未満、20m 以上の 4 種類のデータに分かれており、SVR では歩車間距離が小さいほど歩車間距離誤差も小さくなるのに対して、LSTM では歩車間距離の大小によらず測距誤差がほぼ一定で変化しないということがわかる。従って、LSTM を用い

た距離推定モデルが SVR モデルと比べ高精度だったのは歩車間距離が大きな場合でも特に精度を落とすことなく測距を行うことができたからではないかと考えられる。

#### 4.2.3 検討方式と先行方式の測位誤差の比較

最後に、検討方式 (LSTM+KF) と以下 (1) (2) (3) の方式について歩行者の水平測位誤差の比較を行う。検討方式は LSTM で測距を行った後、カルマンフィルタによる補正をしながら測位演算を行ったものであり、比較する方式は (1) SVR で測距をして単独測位 (Single Point) を行ったもの (SVR+SP)、(2) SVR で測距をしてカルマンフィルタを使った測位を行ったもの (SVR+KF)、(3) LSTM で測距をした後単独測位で測位を行ったもの (LSTM+SP) である。以下図 16、表 7 に平均水平測位誤差の累積度数分布グラフを示す。

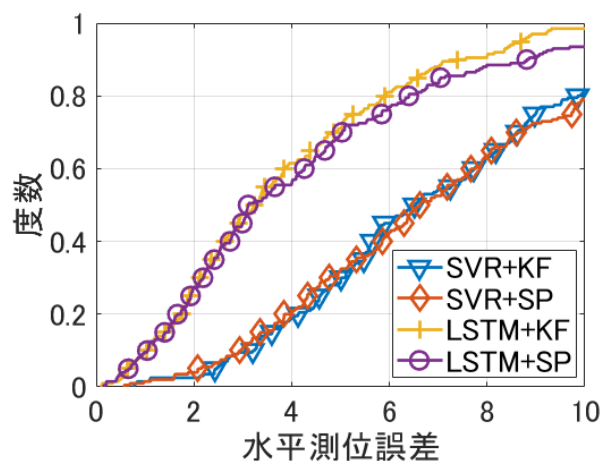


図 16 歩行者の水平測位誤差

表 7 カルマンフィルタの有無による測位誤差の違い

方式	平均水平測位誤差 (m)
(1) SVR+SP	7.88
(2) SVR+KF	7.16
(3) LSTM+SP	4.63
(検討方式) LSTM+KF	3.80

結果より、カルマンフィルタ、LSTM を用いた検討方式が最も良い効果を示した。各方式からの改善幅は、(1) からは 51.8%、(2) からは 46.9%、(3) からは 17.9%の改善がみられた。(1)、(2) と (3) と検討方式との比較をすると、カルマンフィルタでの精度改善より LSTM による精度改善の方が大きいことがわかる。これは状態についての平滑化は測位演算時よりも測距時に行う方が有用であるということが考えられる。また、SVR、LSTM において、カルマンフィルタでの改善幅はそれぞれ 9.5%、17.9%であり、LSTM でのモデル作成時の方が有効に働くと示している。

## 5. おわりに

本稿では、高精度歩行者測位のために、入力データとして用いた歩車間距離と CSI のペアを車両ごとのシーケンスにまとめることにより、歩車間距離の測距モデルの作成に LSTM ニューラルネットワークを用いた高精度歩車間測距モデルを検討し、測距・測位の性能評価を行った。検討方式は従来の SVR を使用した先行方式と比較して歩車間距離誤差を 44.9%改善し、それに伴い歩行者の水平測位誤差も 46.9%改善することが確認できた。今後は、環境変化の影響を考慮し、観測したデータを用いてより動的に測距モデルも更新できるような方式を考案していきたいと考えている。

## 参考文献

- [1] 官民 ITS 構想・ロードマップ 2020, [https://www.kantei.go.jp/jp/singi/it2/kettei/pdf/20200715/2020\\_roadmap.pdf](https://www.kantei.go.jp/jp/singi/it2/kettei/pdf/20200715/2020_roadmap.pdf)
- [2] 内閣府, 令和 2 年交通安全白書, 2020
- [3] 山下遼, 湯 素華, 小花 貞夫, “歩行者事故削減のための GPS と車両からの電波を用いた歩行者位置の高精度測位方式の提案と評価,” 情報処理学会論文誌, vol.59, no.1, pp.113–123, 2018.
- [4] Suhua Tang and Sadao Obana, “Improving performance of pedestrian positioning by using vehicular communication signals,” *IET Intelligent Transport Systems*, vol. 12, no. 5, pp. 366-374, 2018.
- [5] B. Hou and X. Zhang, “A dual-satellite GNSS positioning algorithm of high accuracy in incomplete condition,” *China Communications*, vol.13, no.10, pp.58-68, 2016.
- [6] 則島景太, 高木直, 坪内和夫, “準天頂衛星による位置捕捉精度の検討,” 第 75 回全国大会講演論文集, pp.203-204, 2013
- [7] Yearlor Patou, Sadao Obana, and Suhua Tang, “Improvement of pedestrian positioning precision by using spatial correlation of mutipath error,” in *Proc. IEEE ICVES'18*, Sept. 2018.
- [8] Sepp Hochreiter and Jürgen Schmidhuber, “Long Short-Term Memory,” *Neural Computation*, vol.9, no.8, <https://doi.org/10.1162/neco.1997.9.8.1735>, 1997.
- [9] Felix A. Gers, Jiürgen Schmidhuber, and Fred Cummins, “Learning to Forget: Continual Prediction with LSTM,” *Neural Computation*, vol.12, no.10, <https://doi.org/10.1162/089976600300015015>, 1999.
- [10] David M. Q. Nelson, Adriano C. M. Pereira, and Renato A. de Oliveira, “Stock Market’s Price Movement Prediction with LSTM Neural Networks,” in *Proc. International Joint Conference on Neural Networks (IJCNN)*, pp. 1419-1426, 2017.
- [11] Jinkyu Lee and Ivan Tashev, “High-level Feature Representation using Recurrent Neural Network for Speech Emotion Recognition,” *Interspeech'15*, 2015.
- [12] 浦野 健太, 廣井 慧, 米澤 拓郎, 河口 信夫, “ニューラルネットワークを用いた大規模イベント向け BLE 屋内位置推定の比較評価,” 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, pp.29-35, 2019.
- [13] 構造計画研究所, 衛星測位分析ツール GPS-Studio <https://network.kke.co.jp/products/gps-studio/>
- [14] 構造計画研究所, 伝搬解析ツール RapLab <https://network.kke.co.jp/products/raplab/>
- [15] MathWorks, MATLAB - 技術計算言語 <https://jp.mathworks.com/products/matlab.html>