IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received October 6, 2019, accepted October 28, 2019, date of publication October 31, 2019, date of current version November 18, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2950728

Pedestrian Heading Estimation Based on Spatial Transformer Networks and Hierarchical LSTM

QU WANG[1], HAIYONG LUO[2], (Member, IEEE), LANGLANG YE[2], AIDONG MEN[1], (Member, IEEE), FANG ZHAO[3], YAN HUANG[4], AND CHANGHAI OU[5]

[1]School of Information and Communication Engineering, Beijing University of Posts and Telecommunications, Beijing 100876, China
[2]Beijing Key Laboratory of Mobile Computing and Pervasive Device, Institute of Computing Technology, University of Chinese Academy of Sciences, Beijing 100190, China
[3]School of Software Engineering, Beijing University of Posts and Telecommunications, Beijing 100876, China
[4]School of Electronics Engineering and Computer Science, Peking University, Beijing 100871, China
[5]School of Computer Science and Engineering, Nanyang Technological University, Singapore 639798

Corresponding authors: Haiyong Luo (yhluo@ict.ac.cn) and Aidong Men (menad@bupt.edu.cn)

This work was supported in part by the National Key Research and Development Program under Grant 2016YFB0502000, in part by the National Natural Science Foundation of China under Grant 61872046, Grant 61671264, and Grant 61671077, in part by the Action Plan Project of the Beijing University of Posts and Telecommunications supported by the Fundamental Research Funds for the Central Universities under Grant 2019XD-A06, in part by the Special Project for Youth Research and Innovation, Beijing University of Posts and Telecommunications, supported by the Fundamental Research Funds for the Central Universities under Grant 2019PTB-011, in part by the Key Research and Development Project from Hebei Province under Grant 19210404D, in part by the Open Project of the Beijing Key Laboratory of Mobile Computing and Pervasive Device, in part by the Blue Fire Plan (Huizhou) Industry-University Joint Innovation Project of Ministry of Education under Grant CXZJHZ201729, and in part by the BUPT Excellent Ph.D. Students Foundation under Grant CX2018102.

ABSTRACT Accurate heading estimation is the foundation of numerous applications, including augmented reality, pedestrian dead reckoning, and human-computer interactions. While magnetometer is a key source of heading information, the poor accuracy of consumer-grade hardware coupled with the pervasive magnetic disturbances makes accurate heading estimation a challenging issue. Heading error is one of the main error sources of pedestrian dead reckoning. To reduce the heading error and enhance robustness, we proposed a novel heading estimation method based on Spatial Transformer Networks (STNs) and Long Short-Term Memory (LSTM), termed DeepHeading, which uses sensors embedded in a smartphone without any historical training data or dedicated infrastructure. We automatically annotate heading data based on map matching, and augment heading data based on device attitude. We leverage the STNs to align the device coordinate system and the navigation coordinate system, allow an unconstrained use of smartphones. Based on the characteristics of pedestrian heading continuity, we designed a hierarchical LSTM-basedSeq2Seq model to estimate the walking heading of the pedestrian. We conducted well-designed experiments to evaluate the performance of deepheading and compared it with the state-of-the-art heading estimation algorithms. The experimental results on real-world demonstrated that deepheading outperformed the compared heading estimation algorithms and achieved promising estimation accuracy with a median heading error of 4.52°, mean heading error of 6.07° and heading error of 9.18° at the confidence of 80% when a pedestrian is walking in indoor environments with magnetic field disturbances. The proposed method is high-efficiency and easy to integrate with various mobile applications.

INDEX TERMS Indoor positioning, heading estimation, pedestrian dead reckoning, deep learning.

I. INTRODUCTION

In various spatiotemporal network demands, location-based service (LBS) is one of the essential network service applications [1]. Accurate and pervasive indoor positioning has been applied in a variety of domains such as asset and personnel tracking, search and rescue, health monitoring, location-based social networks, and location-specific push notifications [2]. To meet the explosive demand, various indoor positioning and indoor navigation approaches have recently been developed, including RFID [3], UWB [4], BLE [5], Wi-Fi [6]–[8], magnetic [9], [10], visible light

The associate editor coordinating the review of this manuscript and approving it for publication was Mohammad Anwar Hossain.

[11], [12] and visual methods [13]. Most existing technologies depend on historical training data or dedicated infrastructure, which is always not continuously available during pedestrian walking. Moreover, the hardware cost of largescale deployment is very expensive

Among various indoor positioning schemes, pedestrian dead reckoning (PDR) become mainstream technology because PDR does not rely on any dedicated infrastructures [14]–[16]. Furthermore, PDR based on smartphone mainly benefits from the proliferation of smartphones, which have integrated inertial measurement unit (IMU) sensor sensors (gyroscope and accelerometer) and magnetometer. Smartphone is always carried almost everywhere we go, thus making that smartphone plays an indispensable role in our daily lives. Moreover, unlike radio-frequency signal-based methods, PDR has no coverage limitation, thus providing seamless positioning services connecting indoor and outdoor environments. Given an initial location, PDR estimates the relative position of a pedestrian in real-time by joining the heading estimation and displacement of each step. An ocean of studies about step detection and step counting have been done, including context-based, peak detection, frequency domain features, auto-correlation, zero-crossing and neural networks [17]. Considerable studies about step length estimation have been developed to enhance the accuracy of step length estimation, and they are summarized as double integration of the acceleration, empirical relationships, linear models, nonlinear models, biomechanical models, context-based and deep learning [18], [19].

The heading (walking direction of pedestrian at each step) estimation of pedestrian is a dominant factor in PDR algorithms [20]. For instance, 5 degrees heading estimation error will result in 5.2 meters positioning error after traveling 100 meters. Nowadays, to develop PDR systems with smartphoneembedded low-cost sensors, accurate heading estimation over a long period is a crucial issue [21]. Unlike step counting and step length estimation, heading estimation is more challenging. Many heading estimation methods employ wearable devices worn in a specific position to estimate the heading of pedestrian [22]. Foxlin [23] leveraged foot-mounted sensors estimate reliable heading of pedestrian by introducing the zero velocity updates into extended Kalman Filter. Jirawimut *et al.* [24] used stereo-cameras fixed on the pedestrian's chest to achieve the walking heading of pedestrian. However, carrying a dedicated device is unpractical and inconvenient for users in their daily lives. For the mass users, using smartphone as a navigation carrier for heading estimation is more practical.

Most smartphones-based heading estimation methods are based on gyroscopes, accelerometers, magnetometers. Magnetometers estimate the heading angle relative to the magnetic north by sensing the earth magnetic field. Attitude and Heading Reference Systems (AHRS) combines inertial sensors and magnetometer observations to estimate heading. Madgwick *et al.* [25] presented a real-time heading estimating algorithm for the x-IMU sensor board [26] based on the gradient descent method, which is widely known as the most accurate open source heading estimating algorithm. The Madgwick *et al.* [25] and Mahony *et al.* [27] only correctly estimated the heading of pedestrian under ideal conditions (i.e. no magnetic disturbances). However, the ferromagnetic materials in buildings severely affected magnetometer readings, which resulted in heading errors [28]. Gyroscopes only give a short-term accurate heading estimation, while magnetometers give a long-term heading estimation. Kang *et al.* [29] proposed a reliable heading estimation method based on the correlation between magnetometer and gyroscope using a selection and weighting algorithm. Poulose *et al.* [14] estimated heading by fusing the magnetometer and gyroscope sensor values. To enhance heading estimation accuracy, Xu *et al.* [30] designed a heading correction method based on zero angular velocity and lateral velocity limitation to reduce the error of heading estimation. Zhao *et al.* [15] improved gradient descent algorithm to reduce the heading drift. To obtain a better heading estimation, extended Kalman filtering (EKF) [31]–[33] and complementary filters (CF) [34]–[36] are used to fuse inertial sensors and magnetometer readings, and reduce sensor noise. In References [37], [38], smartphone-based heading estimation algorithms required users to hold a smartphone in a fixed mode as long as possible. Most existing smartphone-based heading estimation methods assume that the misalignment angle between the device and pedestrian remains constant, and pedestrian's heading is determined by removing the heading offset [22]. However, the misalignment angle varies with body locomotion. The pedestrians may put their smartphone in bag, make a phone call, or even swing in hand during walking. To solve this problem, references [39]–[43] proposed smartphone mode or posture recognition-based PDR method that calculated pedestrian heading by selecting heading model with corresponding to smartphone mode. However, the limited mode classification cannot reflect the diversity of smartphone carrying modes. The unconstrained posture and position of a smartphone brings great challenges for heading estimation.

To estimate the heading of pedestrian without the constraints of smartphone attitude, some researchers utilize the principal component analysis (PCA) of acceleration to determine the offset between the misalignment angle between smartphone and pedestrian. Wang *et al.* [40] combined PCA with global accelerations to estimate pedestrian headings. Kunze *et al.* [44] utilized PCA over the horizontal acceleration components for estimating walking heading of a pedestrian with a smartphone placed in pocket. However, the performance of PCA-based approaches relies on the estimation of the horizontal acceleration plane that is prone to the device coordinates change. To enhance the estimation accuracy of the horizontal acceleration plane, RMPCA [22] leveraged a rotation matrix that tracks the device coordinates accurately and continuously, and combined the rotation matrix with PCA to estimate pedestrian heading. However, the approach is specifically designed for the smartphone

placed in a trouser pocket, and it may be not suitable for other positions. Moreover, the PCA-based heading estimation methods have the problem of 180° ambiguity [45] that is difficult to solve. For estimating the heading of pedestrian without attitude constraints, uDirect [46] directly estimated the heading of pedestrian within a specific area of acceleration, where the forward acceleration (the first principal component of principal component analysis) dominated the horizontal acceleration components. However, the specific area was always corrupted by the lateral acceleration during walking. More importantly, both PCA-based nor uDirect approaches utilize compass to convert the local heading into the global one. Compass is prone to magnetic perturbations [47]. Therefore, neither PCA nor uDirect approach can obtain an accurate and robustness heading estimation in the indoor environment. In addition, studies [48]–[54] leveraged map information or map matching technology to reduce heading estimation error and improve heading estimation accuracy. However, map limits the practicability of PDR.

Deep learning automatically learns features with high-level abstractions by using multiple nonlinear transformations [55]. Deep learning has been deemed to be powerful for natural language processing and speech-recognition, without additional processing. Neural Networks show excellent potential to exploit and analyze the collected data without handcraft feature extraction. Motivated by the fact that speech recognition based on deep learning outperforms other existing traditional speech recognition methods, studies [56]–[62] leveraged deep learning technologies to reckon trajectory of pedestrian. All of them achieve superior localization accuracy. However, they only provide position information, no heading information. This paper proposed a novel heading estimation method based on Spatial Transformer Networks (STNs) [63] and Long Short-Term Memory (LSTM), termed DeepHeading, to overcome the aforementioned shortcomings. We automatically annotate the heading data based on map matching, and augment the heading data based on device attitude. The STNs is used to align the coordinate system. In addition, we design a hierarchical LSTM-basedSeq2Seq model to estimate the heading of the pedestrian according to the characteristics of pedestrian heading continuity. DeepHeading estimates pedestrian's heading with inertial sensors and magnetometer embedded in commercial smartphones. The main contributions of paper are as follows:

- We propose a method for automatic data labeling in pedestrian heading estimation problem. The ground-truth of pedestrian heading is hard to collect, which is rarely mentioned and mostly jumped over in most existing related studies. We leverage a foot-mounted IMU module to collect the relative trajectory, then match the trajectory with the real indoor path to obtain the accurately heading of pedestrian in each step.
- We propose a heading data augmentation method based on device attitude. To reduce the amount of training data collection and solve the terminal attitude problem, data augmentation method is used to expand the collected data, thus promoting the fit of the model to the pedestrian's heading and enhancing the robustness of the heading.
- We propose a pedestrian heading estimation method based on STNs and hierarchical LSTM. We leverage the STNs to align the device coordinate system and the navigation coordinate system. Based on the characteristics of pedestrian heading continuity, we design a hierarchical LSTM-based Seq2Seq model to estimate the walking heading of the pedestrian.
- We conduct elaborate experiments to compare the heading estimation accuracy of single-step heading model with Seq2Seq heading model, and analyze the impact of data augmentation. The experimental results on real-world data set show that the heading error is within 9.18° at the confidence of 80%.

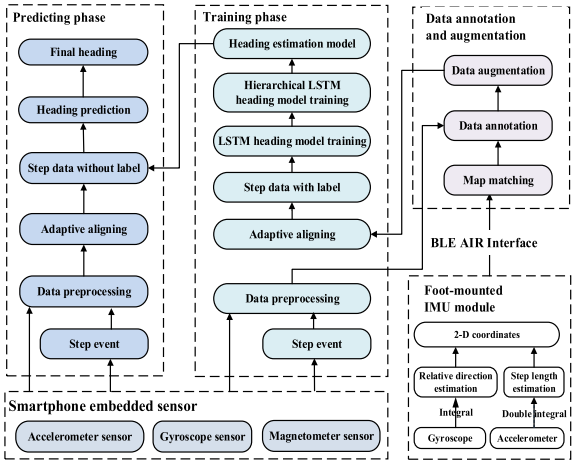


FIGURE 1. System architecture of deepheading.

The rest of the paper is organized as follows: we detail the solution of the proposed deepheading in Section II. After that, we evaluate deepheading in Section III. At last, we draw our conclusions and future work in Section IV.

II. MATERIALS AND METHODS

In this section, overview of proposed method is depicted. Then, the essential modules are described, which include heading data annotation based on map matching, heading data augmentation method based on device attitude, and heading estimation method based on STNs and hierarchical LSTM.

A. SYSTEM ARCHITECTURE

Figure 1 illustrates the overall structure of Seq2Seq pedestrian heading estimation model based on Spatial Transformer Networks (STNs) and hierarchical LSTM. In the offline phase we utilized the inertial-sensors and magnetometer measurements from smartphone as training data, and the corresponding PDR trajectory from the foot-mounted IMU module and map matching as labels to train a heading model. In the online predicting phase, we leveraged the trained heading estimation model and real-time sensor data to accurately estimate the heading of pedestrian in each step.

B. HEADING DATA ANNOTATION BASED ON MAP MATCHING

For heading estimation, a key problem is how to obtain the ground-truth of the pedestrian's heading. The ground-truth is the key to verify and tune the heading estimation algorithm. For the heading estimation model based on deep learning, the ground-truth of the user's heading is the necessary data label for model training. At present, many papers avoid the question of getting the user's heading ground-truth. Some papers directly use the known heading of the straight path as the ground-truth of the user's heading. However, users do not strictly follow a straight line most of the time. Moreover, even if user walks in a straight line, there will be a slight swing in heading during the walking process. Therefore, it is challenging to accurately capture the heading of pedestrians at every step.

This section introduces a map matching-based heading data annotation algorithm, which utilizes the user's trajectory information and indoor map to obtain the true heading of the user's current trajectory, and further calculates the user's one-step heading and completes the data labeling. We use Hidden Markov Model (HMM) to match the PDR trajectory to the key points of the map (called node). Pedestrian trajectory matching is modeled as a decoding problem. In other words, given a model $\lambda = \{M, N, \pi, A, B\}$ and observation sequence O, pedestrian trajectory matching aims to find an optimal node sequence Q that best explains the observation sequence O. In our model, λ, M and N represent the number of hidden states and observation states, respectively. π denotes the initial state distribution. A and B represent the state transition matrix and the emission matrix. We present our HMM as follows:

- Hidden States: in our HMM, we define the key nodes of the map as hidden states. each state represents a node. $M = \{S_1, S_2, \cdots, S_i, \cdots\}$ is the set of hidden states, where $S_i = [kp_{i,0}, kp_{i,1}]$ represents the position of the i^{th} state. Pedestrian only moves between adjacent nodes.
- Observations: we use the walking distance and average heading during two consecutive turns as observations. $N_i = \{Dis_i, Dir_i\}$ is the set of observations. Dis_i and Dir_i are the walking distance and the mean of heading angle observations during two consecutive turns.
- Transition Probabilities: a transition between hidden states is triggered by a turn event. Since the building topology constrains the accessible areas of pedestrians, we leverage the length information of indoor path to construct a transition matrix. Therefore, the transition probability of each node in our HMM model is defined as follow,

$$A_{i,j} = \frac{1}{1 + Dis_{i,j}} \tag{1}$$

$$TP_{i,j} = \frac{A_{i,j}}{\sum_{j=1}^{N} A_{i,j}} \tag{2}$$

where Dis_i is the walking distance observations during two consecutive turns. TP is transition probabilities

- Emission Probabilities: the emission probability consists of the distance and heading information. Since walking distance and heading observations are independent, we define the emission probability distribution EP as follows,

$$\begin{aligned} EP_k &= P(Dis_{PDR}, Dir_{PDR} | S_{k-1}, S_k) \\ &= \frac{1}{\sqrt{2\pi} DisStd_{s_{k-1},s_k}} e^{-\frac{1}{2DisStd^2_{s_{k-1},s_k}}(Dis_{PDR} - Dis_{s_{k-1},s_k})^2} \\ &\quad \bullet \frac{1}{\sqrt{2\pi} DirStd_{s_{k-1},s_k}} e^{-\frac{1}{2DirStd^2_{s_{k-1},s_k}}(Dir_{PDR} - Dir_{s_{k-1},s_k})^2} \end{aligned} \tag{3}$$

where

$$Dis_{i,j} = \begin{cases} \sqrt{(kp_{i,0} - kp_{j,0})^2 + (kp_{i,1} - kp_{j,1})^2}, \\ \quad if\ \mathrm{i} \neq \mathrm{j}\ \text{and}\ \mathrm{sc}_{i,j} = 1 \\ \infty, \\ \quad if\ \mathrm{i} = \mathrm{j}\ \text{or}\ \mathrm{sc}_{i,j} = 0 \end{cases} \tag{4}$$

$$Dir_{i,j} = \begin{cases} 0, \\ \quad if\ \mathrm{i} = \mathrm{j} \\ \cos^{-1}(\frac{-(kp_{j,1} - kp_{i,1})}{dis_{i,j}}) * 180, \\ \quad if\ \mathrm{i} \neq \mathrm{j}\ \text{and}\ \mathrm{sc}_{i,j} = 1\ \text{and}\ kp_{j,1} - kp_{i,1} \geq 0 \\ 360 - \cos^{-1}(\frac{-(kp_{j,1} - kp_{i,1})}{dis_{i,j}}) * 180, \\ \quad if\ \mathrm{i} \neq \mathrm{j}\ \text{and}\ \mathrm{sc}_{i,j} = 1\ \text{and}\ kp_{j,1} - kp_{i,1} < 0 \\ NaN, \\ \quad if\ \mathrm{i} \neq \mathrm{j}\ \text{and}\ \mathrm{sc}_{i,j} = 0 \end{cases} \tag{5}$$

where S_k is the kth hidden state. $\mathrm{sc}_{i,j}$ represents connectivity between state i and state j. $\mathrm{sc}_{i,j} = 1$ represents reachable path, and $\mathrm{sc}_{i,j} = 0$ represents unreachable path. $[kp_{i,0}, kp_{i,1}]$ represents the position of the i^{th} state. $DirStd_{s_{k-1},s_k}$ and $DisStd_{s_{k-1},s_k}$ represent the standard deviation of the heading and distance observations, respectively. Dis_{PDR} is the calculative walking distance calculated by PDR, and Dis_{s_{k-1},s_k} is the distance between s_{k-1} and s_k. Dir_{PDR} is the average heading estimated by PDR, and Dir_{s_{k-1},s_k} is the angle between vector s_{k-1},s_k and magnetic North.

- Initial State Distribution: the initial state distribution is uniform when the first turn event is detected.

$$\Pi_i = \frac{1}{N} \tag{6}$$

where N represents the number of observation states.

- Pedestrian Trajectory Matching: after obtaining the transition and emission probabilities, we solve the HMM-based map-matching problem utilizing the

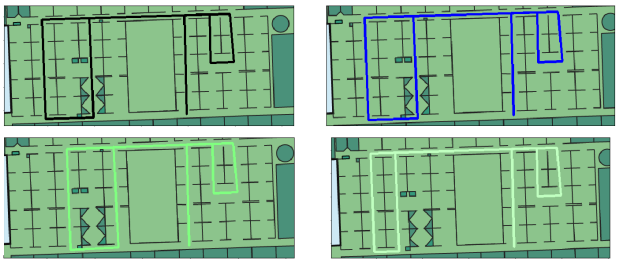


FIGURE 2. Map matching based on HMM.

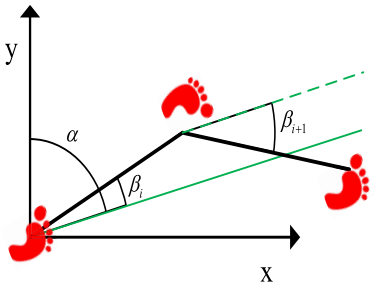


FIGURE 3. Data annotation of pedestrian step heading.

Viterbi decoder [64]. To decrease the computational overhead of Viterbi decoding procedure, we leverage the user's heading and walking distance to remove the invalid states. In addition, PDR usually works with other positioning algorithms that can be used to decrease the computational overhead of map matching and improve matching accuracy.

$$p(s_t = x) = \max_i p(s_{t-1} = i) \bullet p(s_t = x | s_{t-1} = i) \bullet p(Dis_{PDR}, Dir_{PDR} | s_{t-1} = i, s_t = x) \quad (7)$$

As shown in Figure 2, the upper left is the true walking path, the top right is the most likely path in the matching result, and the bottom left and bottom right are the candidate paths for the second and third probability in the matching results.

After getting the most likely path, it is easy to get the true heading value of each atomic path (a straight from map information). As shown in Figure 3, we calculate the real heading of user one-step by combining the deviation between the atomic path and trajectories from foot-mounted IMU model. The green line is atomic path, and the black line is the walking trajectory from foot-mounted IMU module.

$$h_i = \alpha_k + \beta_i, \quad step_i \in seg_k \quad (8)$$

where seg_k is kth atomic path, α_k is the heading of the kth atomic path. β_i is the angle between the heading of the $step_i$ in the kth atomic path and the heading of the corresponding atomic path. h_i is the ground-truth of walking heading in navigation frame.

C. HEADING DATA AUGMENTATION METHOD BASED ON DEVICE ATTITUDE

The key problem with deep learning in practice is the amount of data. The training process of the deep learning model is a process of model parameter adjustment. The number of parameters in the deep learning model is often in the millions or even hundreds of millions of magnitudes Therefore, a massive amount of data is needed to learn these parameters. However, the process of collecting massive amounts of data is often a time-consuming and laborious task. It makes sense and is necessary to expand the amount of data by using data-expanding techniques on the original data set. In addition, increasing the data in the dataset effectively can prevent the network from learning irrelevant characteristics and focus more on the characteristics related to the goal thus significantly enhancing the model performance and generalization capabilities. However, the data augmentation method for time-series data (sensor data) has not been well investigated so far.

We expand the inertial sensor data by attitude conversion, so that the model extracts more useful features directly related to the user's heading from the sensor data, thus eliminating the influence of the device attitude. We first randomly generate a set of Euler angles (yaw angle α, roll angle β, pitch angle γ),

$$(\alpha, \beta, \gamma) = Random(3) \quad (9)$$

The corresponding attitude conversion matrix is calculated for the resulting Euler angle. (10)–(13), as shown at the bottom of the next page.

We convert the sensor data to generate new samples.

$$D = \begin{bmatrix} x_1 & \cdots & x_T \\ y_1 & \cdots & y_T \\ z_1 & \cdots & z_T \end{bmatrix} \quad (14)$$

$$D_{new} = RD \quad (15)$$

where (x,y,z) represent three-axis sensor data, T is the number of sensor readings in the time interval of each step. D and D_{new} denote the original and generated three-axis sensor data-series, respectively.

It is important to emphasize that the proposed heading data augmentation method is performed for the accelerometer, the gyroscope and the magnetometer data, respectively.

D. PEDESTRIAN HEADING ESTIMATION BASED ON STNS AND HIERARCHICAL LSTM

1) ADAPTIVE ALIGNING ALGORITHM BASED ON STNs

The alignment of device coordinate system and navigation coordinate system is a key problem in heading estimation. The traditional way is to maintain a real-time attitude estimation and convert the observations from device coordinate system to the navigation coordinate system. In response to this problem, we present an adaptive alignment algorithm based on Spatial Transformer Networks (STNs) [63]. STNs is proposed by Google DeepMind to solve the spatial invariance of computer vision models on input images. STNs mainly transforms the input image or feature map, outputs the new image or feature map, and in theory, the STNs can correct the image or feature map as the ideal image or feature

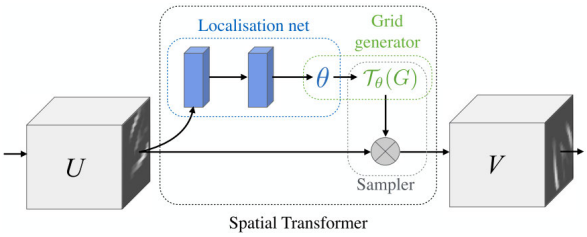


FIGURE 4. Structure of spatial transformer networks [63].

map after the transformation of the image or feature map. As shown in Figure 4, the input of STNs is a matrix or feature map U, and the output is a corrected image or feature map V, which mainly consists of a localization network, a grid generator, and a sampler. The input feature map U is passed to a localization network which regresses the transformation parameters θ. The regular spatial grid G over V is transformed to the sampling grid $T_\theta(G)$, which is applied to U, producing the warped output feature map V. The combination of the localization network and sampling mechanism defines a spatial transformer.

First, the localization network is a customizable network whose primary function is to calculate the transformation parameters θ through the input U. This parameter is used to map the coordinates (x,y) in U to the coordinates (x', y') in V. For instance, zoom in on the input image k times, or rotate the input by θ degrees.

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} k & 0 \\ 0 & k \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \end{bmatrix} \tag{16}$$

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} cos\theta & -sin\theta \\ sin\theta & cos\theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \end{bmatrix} \tag{17}$$

The prediction parameters are determined by the performed transformation. For the two two-dimensional affine transformations described above, the prediction is six parameters that represent the coordinate mapping relationship between the input and output.

Secondly, after the transformation parameters are obtained by localization network, the corresponding coordinate points can be found in the output map V by making matrix operations for each point in the input feature map U according to the parameter obtained by the localization network.

$$\begin{bmatrix} x_i^s \\ y_i^s \end{bmatrix} = \Theta \begin{bmatrix} x_i^t \\ y_i^t \\ 1 \end{bmatrix} = \begin{bmatrix} \theta_{11} & \theta_{12} & \theta_{13} \\ \theta_{21} & \theta_{22} & \theta_{23} \end{bmatrix} \begin{bmatrix} x_i^t \\ y_i^t \\ 1 \end{bmatrix} \tag{18}$$

where (x_i^t, y_i^t) is the target coordinates of the regular grid in the output feature map, Θ is the affine transformation matrix, and (x_i^s, y_i^s) is the source coordinates in the input feature map that define the sample points.

Finally, the grid generator generates samples. Each pixel in output map V is filled with the corresponding coordinates in input map U to obtain a pixel value. Because the calculated coordinates can be decimal, the coordinates need to be processed, and the surrounding pixels need to be considered when filling, the filling formula is as follows.

$$V_i = \sum_n \sum_m U_{nm} * k\left(x_i^s - m; \emptyset_x\right) * k\left(y_i^s - n; \emptyset_y\right) \tag{19}$$

where m and n will traverse all coordinate points in the input map U. U_{nm} refers to the pixel value of a point in the input map U. k is the sampling core, indicating that different methods are used to fill. $\emptyset_x$ and $\emptyset_y$ are parameters. $\left(x_i^s, y_i^s\right)$ represents the corresponding point in the ith point in V to U. Filling is usually done by bilinear interpolation. After filling, a converted image or feature map is obtained.

$$V_i = \sum_n \sum_m U_{nm} * max\left(0, 1 - \left|x_i^s - m\right|\right) * max\left(0, 1 - \left|y_i^s - n\right|\right) \tag{20}$$

STNs is a plug-and-play module that can be inserted into an existing model, thus allowing the model to achieve the spatial transformation of an image or feature map without additional supervision and without changing the original optimization process. STNs is mainly used in such tasks like image classification and image entity co-positioning.

$$R(\alpha) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\alpha) & -\sin(\alpha) \\ 0 & \sin(\alpha) & \cos(\alpha) \end{bmatrix} \tag{10}$$

$$R(\beta) = \begin{bmatrix} \cos(\beta) & 0 & \sin(\beta) \\ 0 & 1 & 0 \\ -\sin(\beta) & 0 & \cos(\beta) \end{bmatrix} \tag{11}$$

$$R(\gamma) = \begin{bmatrix} cos(\gamma) & -\sin(\gamma) & 0 \\ \sin(\gamma) & \cos(\gamma) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{12}$$

$$R = R(\gamma)$$

$$R(\beta)R(\alpha) = \begin{bmatrix} \cos(\beta)\cos(\gamma) & \sin(\alpha)\sin(\beta)\cos(\gamma) - \cos(\alpha)\sin(\gamma) & \cos(\alpha)\sin(\beta)\cos(\gamma) + \sin(\alpha)\sin(\gamma) \\ \cos(\beta)\sin(\gamma) & \sin(\alpha)\sin(\beta)\sin(\gamma) + \cos(\alpha)\cos(\gamma) & \cos(\alpha)\sin(\beta)\sin(\gamma) - \sin(\alpha)\cos(\gamma) \\ -\sin(\beta) & \sin(\alpha)\cos(\beta) & \cos(\alpha)\cos(\beta) \end{bmatrix} \tag{13}$$

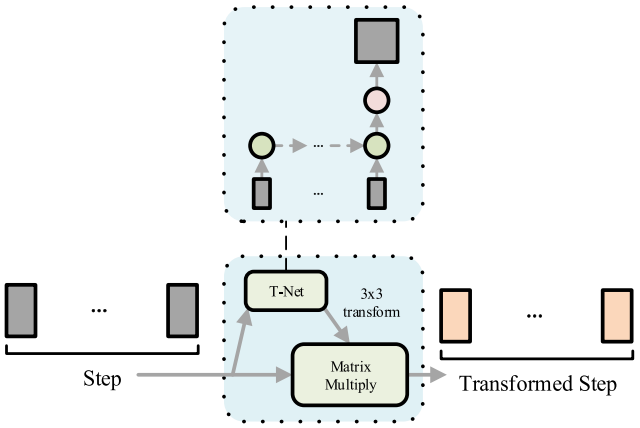


FIGURE 5. A daptive aligning module in the proposed heading estimation model.

For smartphone sensor data, a set of sensor data is similar to a set of three-dimensional point clouds. Therefore, we are inspired by the approach PointNet [65] and implement a simpler approach that is similar to STNs. As shown in Figure 5, we design a transformation network (T-Net) module that predicts an affine transformation matrix based on input data, transforms the input data with the predicted transformation matrix and then passes it into the original model for adaptive alignment.

Before the Step LSTM layer, sensor data for a gait cycle is passed into a small LSTM network. This LSTM network integrates sensor data in the gait cycle and eventually predicts a conversion matrix by regression, which uses the conversion matrix to convert the input sensor data. We use the converted data as input to the Step LSTM layer. The conversion module is inserted between the input layer and the Step LSTM layer. Before extracting the time-series features, the current sensor data input is predicted and converted, so that the model has some affine invariant.

2) SEQ2SEQ PEDESTRIAN HEADING ESTIMATION MODEL BASED ON HIERARCHICAL LSTM

There is a strong correlation between the heading of the adjacent steps within the same user track, and simply defining the sample by step will lose this continuous information, and the noise of the single-step data causes large-scale heading fluctuations, seriously affecting the performance of the heading estimation. Therefore, we propose a Seq2Seq pedestrian heading estimation model based on hierarchical LSTM to accurately estimate the heading of pedestrian travel, as shown in Figure 6.

The method takes a user trajectory as a sample. The sensor data collected by the trajectory is the input sequence, and the heading corresponding to each step of the trajectory is the output sequence. Sensor data in a gait cycle is processed by the Step LSTM to extract the heading information of the current step. The output of the Step LSTM at each step of the throughout trajectory forms a sequence that acts as an input to the Track LSTM layer. The time step in the Step LSTM layer is a set of sensor data, a multi-to-one LSTM structure, while the time step in the Track LSTM layer is a gait cycle, a many-to-many LSTM structure. The Heading Regression layer calculates the output of each time step of the Track LSTM layer. Finally, the Track LSTM layer output the predicted heading of pedestrian for each step.

3) LOSS FUNCTION DESIGNING FOR HEADING ESTIMATION

For general regression problems, Mean Squared Error (MSE) is usually used as a loss function to adjust the model weight according to the MSE between the model prediction value and the ground-truth. For the heading estimation problem, MSE cannot reflect the true deviation between the predicted value and the ground-truth, due to the circular period problem of the heading as shown in Figure 7. For instance, when the true heading is 5°, the MSE of the predicted 355° is 122500 while the MSE of the predicted 25° is 400. The loss value of the former is much higher than the loss value of the latter. However, the deviation between the former and the real heading is 10°, and the deviation between the latter and the real heading is 20°. The inappropriate loss function directly causes the error in the heading of model optimization, thus preventing the model from converging.

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(\theta_i - \hat{\theta}_i)^2 \tag{21}$$

where θ is the real heading obtained by combining map information and walking trajectory from foot-mounted IMU model, $\hat{\theta}$ is the prediction of heading estimation algorithm.

As shown in Figure 8, we consider the problem of the heading circumference and define new loss function for heading estimation, termed Heading Loss, which reflects the deviation between the predicted heading and the true heading by calculating the European distance between the intersection of the true and predicted headings on the unit circle according to equation (22).

$$Heading\ Loss = \frac{1}{2N}\sum_{i=1}^{N}\left(\left(sin\left(\theta_i\right) - sin\left(\hat{\theta}_i\right)\right)^2 + \left(cos\left(\theta_i\right) - cos\left(\hat{\theta}_i\right)\right)^2\right) \tag{22}$$

where θ is the real heading, $\hat{\theta}$ is the prediction heading.

Figure 9 compares the loss value change of Heading Loss with that of MSE. If the real heading of the pedestrian is 0° (positive north). Heading Loss has a minimum value when the model prediction is 0° or 360°, and the value of the Heading Loss increases gradually as the predicted heading deviates from 0° or 360°. Heading Loss reaches the maximum value when the model prediction is 180° (positive south), which is the opposite heading of the real heading. This result indicates that the Heading Loss is more reasonable than MSE for heading estimation problem.

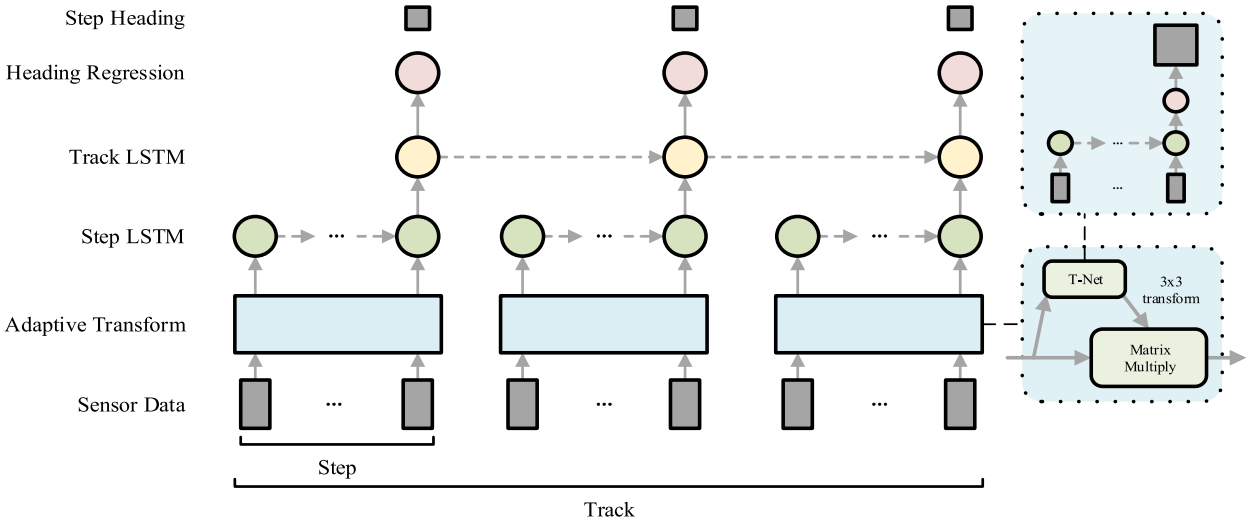


FIGURE 6. The architecture of the proposed Seq2Seq pedestrian heading estimation model based on hierarchical LSTM.

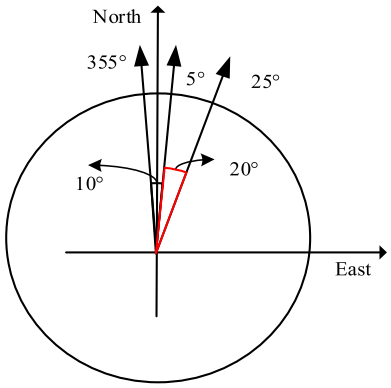


FIGURE 7. The circular period problem of the heading.

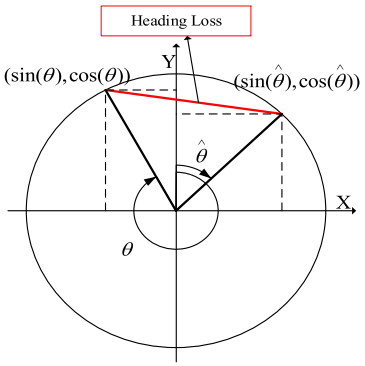
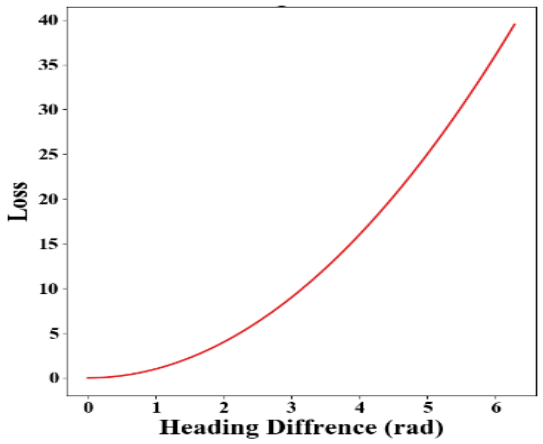
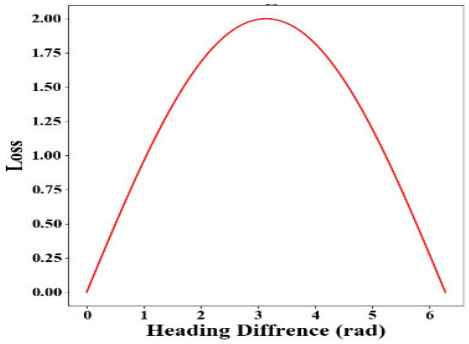


FIGURE 8. Diagram of heading loss.



(a) MSE



(b) Heading Loss

FIGURE 9. Comparison of MSE and heading loss.

E. COMPLETE PROCEDURES OF PROPOSED HEADING ESTIMATION METHOD

Algorithm 1 describes the complete procedures of the proposed heading estimation method. The proposed algorithm takes a set of training samples with precise heading label as input to train the network. The inertial sensors and magnetometer readings are divided into segments by step event. The step data is fed to heading model. The actual heading is used to train the regression layer on the top of the network. Once the training is done, the heading model will be used to predict the heading of pedestrian at each step.

Algorithm 1 Pedestrian Heading Estimation Based on STNs and Hierarchical LSTM

Input: sensors data with PDR trajectory, indoor map, test data without PDR trajectory
Output: heading estimation of pedestrian
// Data annotation and augmentation
Segment the inertial sensor data according to the step event.
For each trajectory **do**
    Perform HMM-based map matching
    Split atomic path according to the most likely path
    Calculate heading of each step according to the deviation between the atomic path and trajectory
    Annotate heading with sensors data for each step
End for
Augment heading data according to device attitude
// Model training
Coordinate system aligning using STNs
Build and train step LSTM model
Build and train track LSTM (Seq2Seq) model
// Testing
Step detection and counting
Coordinate system aligning using STNs
Leverage trained model to predict heading of pedestrian at each step

F. FEVALUATION METRICS

The paper uses a one-step heading error to measure the performance of the model. The one-step heading error is calculated as follow.

$$E_h = \frac{1}{N}\sum_{i=1}^{N}\left(\left(H_e^i - H_t^i\right) mod\ 360\right) \tag{23}$$

where H_e^i and H_t^i denote the estimated heading and the real heading of the i-th step, respectively. E_h is heading error. N is the number of step.

III. EXPERIMENTATION AND EVALUATION

A. AEXPERIMENTAL SETUP

To evaluate the effectiveness and explore the limitations of DeepHeading, we implemented and evaluated DeepHeading in indoor environments with magnetic field disturbances. Obtaining ground-truth of pedestrian heading in indoor environments is not trivial and usually requires dedicated infrastructure such as an optical motion capture system (Vicon [66]) during experimental setup. Here, we have designed a new methodology for obtaining ground-truth of relative heading according map information and walking trajectory. As shown in Figure 10, we obtained data with precise heading labels using a foot-mounted IMU module (NGIMU [67] with a three-axis accelerometer (range ±16 g), and gyroscope (range ±2000 deg/sec) come from x-io technologies), which controls localization error in 0.3% of the entire travel distance and an

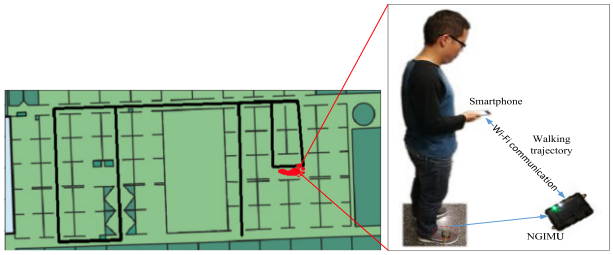


FIGURE 10. The system of training data collection and performance evaluation.

TABLE 1. Description of subjects.

Subject	Gender	Age	Height(cm)	Weight(kg)
S1	M	27	171	68
S2	M	26	183	82
S3	F	23	155	45
S4	M	35	192	85
S5	F	25	161	54
S6	F	27	158	57
S7	M	31	175	70

TABLE 2. Description of devices.

Abbreviation	Model	OS Version	Inertial sensor
D1	Huawei Mate 9	8.0	Icm20690
D2	Huawei P9	7.0	LSM330
D3	Huawei Mate10	9.0	Lsm6dsm
D4	Samsung Galaxy S6	6.0	MPU6500
D5	Vivo X9	7.1	Lsm6ds3
D6	Mi 9	9.0	Lsm6ds3c

Android smartphone, which equipped with inertial sensors (consist of a three-axis accelerometer (range ±8 g), and gyroscope (range ±2000 deg/sec)) from InvenSense (ICM-20690) and a three-axis magnetometer (amk09911, range 2000 uT). The precise PDR trajectory of pedestrian from foot-mounted IMU module was sent to smartphone through a Wi-Fi module. We matched the trajectory with the real indoor path to accurately obtain the heading of pedestrians at every step and synchronize the obtained heading with the MEMS observations of the smartphone. We utilized the obtained heading as ground-truth to train heading model and evaluate the performance of model. The data collected from 7 test subjects and 6 devices were randomly divided into three parts: 70% of the data for heading model training and 30% of the data for heading model validation (15%) and testing (15%). The subjects and devices information were detailed in Table 1 and Table 2.

B. COMPARISON OF MODEL WITH AND WITHOUT DATA AUGMENTATION

To verify the influence of the data augmentation method on model performance, we trained two single-step heading models (one model with data augmentation, another model without data augmentation) and compared their performance using the same data set. We first directly used the original

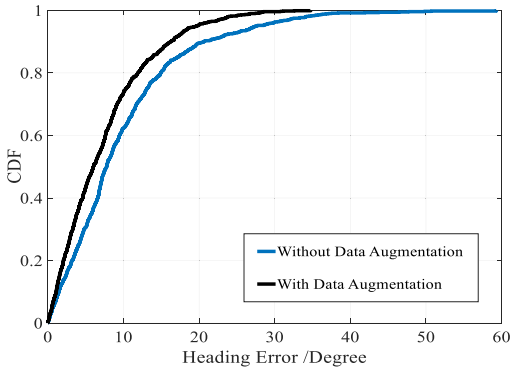
TABLE 3. Comparison of model with and without data augmentation (statistics).

Attributes	Without data augmentation (Degree)		With data augmentation (Degree)	
	Office	Shopping Mal	Office	Shopping Mal
Mean	9.94	10.31	7.44	8.26
Std	8.52	8.17	6.12	6.23
25%	4.06	4.13	2.71	3.48
50%	7.78	8.93	6.04	7.13
75%	13.11	14.22	10.34	11.56
90%	20.60	21.03	16.22	16.73

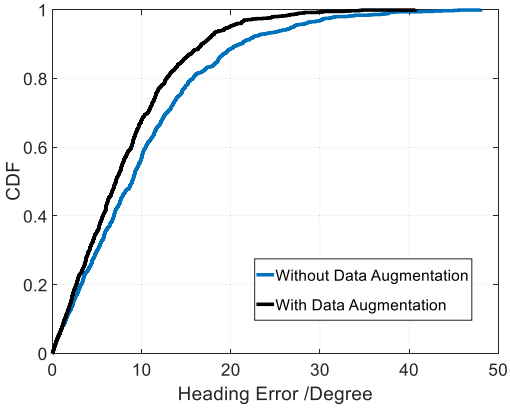
training set to train a single-step heading estimation model. Then, we extended the original data set using the proposed heading data augmentation method based on device attitude, and used the expanded training set to train a heading model. The results of the comparison are as shown in Table 3 and Figure 11. Under 80%, the error of the single-step heading model with data augmentation in office and shopping mall are 11.91° and 12.81°, respectively. Compared to the single-step heading model without data augmentation, the mean heading estimation error of the model with data augmentation decreased from 9.94° to 7.44° in office scenario and 10.31° to 8.26° in shopping mall scenario. In other word, the mean error of the model with data augmentation in office and shopping mall are reduced by 25.15% ((9.94°–7.44°)/ 9.94°)∗100% and 19.88% ((10.31°–8.26°)/ 8.26°)∗100%, respectively. The experimental results demonstrated that data augmentation help improves heading estimation accuracy. The most important thing was that the data augmentation greatly reduced the collection and storage overhead of training data.

C. COMPARISON OF SINGLE-STEP HEADING MODEL AND SEQ2SEQ HEADING MODEL

Considering the continuous characteristics of heading estimation, we designed a Seq2Seq model based on hierarchical LSTM using data from a walking trajectory as a sequence sample. To verify the impact of the Seq2Seq model architecture on heading estimation performance, we used the same augmented data to test and compare performance between the single-step model and the Seq2Seq model. As shown in Figure 12 and Table 4, we used the statistical values and cumulative distribution function (CDF) to demonstrate heading estimation performance comparison between the LSTM-based single-step heading model and the hierarchical LSTM-based Seq2Seq heading model. The heading estimation error of the Seq2Seq heading model was within 9.18° at the confidence of 80%. Compared with the single-step heading model, the mean heading estimation error of the Seq2Seq model decreased from 7.44° to 6.07°. In other words, the performance of the Seq2Seq heading estimation model was enhanced by 18.41%. The experimental results demonstrated that the Seq2Seq heading estimation model for the continuity of the hierarchical LSTM-based heading estimation problem dramatically improved the accuracy of the heading estimation.



(a) **Office**



(b) **Shopping Mall**

FIGURE 11. Comparison of model with and without data augmentation (CDF).

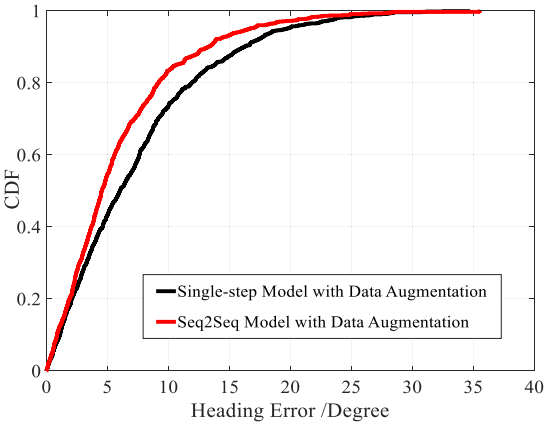


FIGURE 12. Comparison of the single-step LSTM model and the Seq2Seq model (CDF).

D. COMPARISON WITH OTHER METHODS

To justify the superiority of our pedestrian heading estimation method, we compared the proposed pedestrian heading estimation method with the following state-of-the-art pedestrian heading estimation methods:

- uDirect [46], directly estimates the heading of pedestrian within a specific area of acceleration, where the forward acceleration (the first principal component of principal

TABLE 4. Comparison of the single-step model and the Seq2Seq model (statistics).

Attributes	Single-step Model (Degree)	Seq2Seq Model (Degree)
Mean	7.44	6.07
Std	6.12	5.32
25%	2.71	2.31
50%	6.04	4.54
75%	10.34	8.26
90%	16.22	13.24

component analysis) dominated the horizontal acceleration components. However, the specific area was always corrupted by the lateral acceleration during walking.

- RMPCA [22], achieves accurate user heading estimation based on Rotation Matrix and Principal Component Analysis, regardless of the smartphone orientation and placement within the pocket.
- Deng [43], detects the three motion states (random hand movements, carrying position transitions, and user turns,) timely and discriminate them accurately by a position classifier, estimates walking heading with respect to motion state.
- SmartPDR [38]: a PDR solution based on a smartphone kept on hands all the time, which includes an heading estimation method that logically selects how to blend previous heading estimation, gyroscope measurements with magnetic data based on the variation of magnetometer as well as the correlation between magnetometer and gyroscope.

We studied the overall heading estimation accuracy of the proposed method and compared methods with different holding smartphone styles. To make a fair comparison, we utilized the same training set to train the heading estimation models and utilized the same testing set to validate the performance of models. Figure 13 shows the cumulative error distribution of the proposed method and compared heading estimation method. As seen in Figure 13 the 50th percentile heading estimation error of the proposed method is 4.52 degrees, while those of uDirect, RMPCA, Deng and smartPDR are 10.96, 9.76, 6.84 and 5.75 degrees, respectively. the 75th percentile heading estimation error of the proposed method is 7.93 degrees, while those of uDirect, RMPCA, Deng and smartPDR are 19.03, 17.60, 11.17 and 9.94 degrees, respectively. The 90th percentile heading estimation error of the proposed approach is 13.41 degrees, while those of uDirect, RMPCA, Deng and smartPDR are 28.11, 25.01, 16.25 and 16.18 degrees, respectively. As seen in Figure 14, compared with uDirect, RMPCA, Deng and smartPDR, the proposed method reduces the mean heading estimation error by 53.98 percent (7.12 degrees), 48.69 percent (5.76 degrees), 23.65 percent (1.88 degrees), and 17.97 percent (1.33 degrees), respectively. For standard deviation of heading estimation error, the proposed approach reduces it by 44.67 percent (4.57 degrees), 36.33 percent (3.23 degrees), 7.36 percent (0.45 degrees), and 17.49 percent

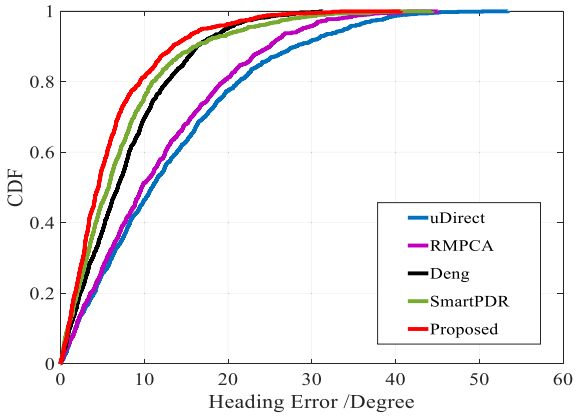


FIGURE 13. Comparison of the proposed method and other methods (CDF).

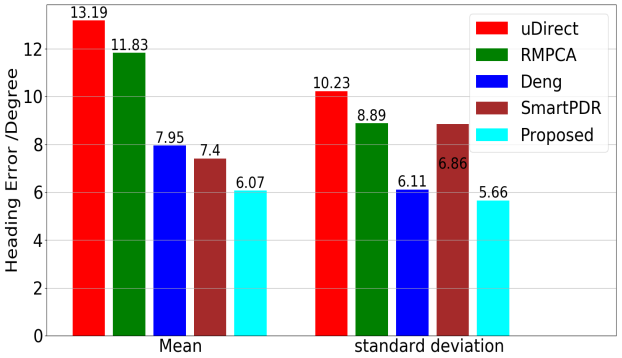


FIGURE 14. Comparison of the proposed method and other methods (Mean and standard deviation).

(1.2 degrees), respectively. The proposed method achieved more significant heading estimation accuracy than those of compared methods.

E. TIME COMPLEXITY ANALYSIS

The top 2 most time-consuming procedures of the proposed heading estimation method are the training data collection and neural-network training. Fortunately, both two procedures are performed in the offline phase In other words, the two procedures do not consume time during the online prediction phase. The time consumption of collecting training data equals walking time. The heading model training is performed on a personal computer equipped with an Intel Core i5-4460 CPU and 16 GB DDR4 RAM.

Table 5 details the training and testing time of the proposed heading estimation method on the whole training dataset and test data. The sample unit of the single-step heading model is the step, while the sample unit of the Seq2Seq heading model is the trajectory. The training set size is 25 trajectories (2790 steps), and the test set size is 7 trajectories (872 steps). Compared to the single-step heading model, the Seq2Seq model adds an LSTM layer, and increases the number of parameters, but the total training time only increases 16 minutes. In the testing phase, the Seq2Seq heading model and the

TABLE 5. Time complexity analysis of heading estimation models in time complexity.

Model	Training Dataset Size	Testing Dataset Size	Training time	All testing time	Testing time of each step
Single step	2790 steps	872 steps	57min	1.97s	2.26ms
Seq2Seq	25 trajectories	7 trajectories	1hour13min	2.02s	2.32ms

single-step model have no obvious difference in time, and the average prediction time per step is about 2.32 milliseconds, with good real-time.

IV. CONCLUSION

Walking heading of pedestrian provides indispensable information for pedestrian dead reckoning, augmented reality/virtual reality, sports training, and human-computer interactions. Notably, the heading is one of the most critical factors in PDR-based indoor positioning. The accurate heading estimation of pedestrians is a challenging research topic. This paper proposed a novel heading estimation by exploiting the inertial sensors built-in smartphones. We automatically annotated heading data based on map matching, and augment the heading data based on device attitude. We leveraged the STNs to align the device coordinate system and the navigation coordinate system. Based on the characteristics of pedestrian heading continuity, we designed a hierarchical LSTM-based Seq2Seq model to estimate the heading of the pedestrian. The proposed method achieved a median heading error of 4.52°, mean heading error of 6.07° and heading error of 9.18° at the confidence of 80%.

Deepheading requires a significant amount data for training, validation and testing. In addition, highly precise labels (ground-truth of walking heading) are extremely important in model training. The acquisition of the ground-truth of the walking direction is a very difficult problem. This paper obtains the walking heading based on map match and user walking trajectories from the foot-mounted inertial module. This method provides a new way for obtaining the ground-truth of walking heading. However, mismatch and non-normal walking will cause significant error of ground-truth. Fortunately, the map matching-based heading data annotation algorithm only used in the offline phase, the training data collector is required to strictly walk along the pre-planned trajectory, and the error of ground-truth be controlled within an acceptable range.

REFERENCES

[1] Z. Peng, S. Gao, B. Xiao, G. Wei, S. Guo, and Y. Yang, "Indoor floor plan construction through sensing data collected from smartphones," *IEEE Internet Things J.*, vol. 5, no. 6, pp. 4351–4364, Aug. 2018.
[2] Y. Shu, C. Bo, G. Shen, C. Zhao, L. Li, and F. Zhao, "Magicol: Indoor localization using pervasive magnetic field and opportunistic WiFi sensing," *IEEE J. Sel. Areas Commun.*, vol. 33, no. 7, pp. 1443–1457, Jul. 2015.
[3] F. Seco and A. R. Jiménez, "Smartphone-based cooperative indoor localization with RFID technology," *Sensors*, vol. 18, no. 1, p. 266, 2018.
[4] J. Sidorenko, V. Schatz, N. Scherer-Negenborn, M. Arens, and U. Hugentobler, "Decawave UWB clock drift correction and power self-calibration," *Sensors*, vol. 19, no. 13, p. 2942, Jul. 2019.
[5] C. Xiao, D. Yang, Z. Chen, and G. Tan, "3-D BLE indoor localization based on denoising autoencoder," *IEEE Access*, vol. 5, pp. 12751–12760, 2017.
[6] R. Wang, Z. Li, H. Luo, F. Zhao, W. Shao, and Q. Wang, "A robust Wi-Fi fingerprint positioning algorithm using stacked denoising autoencoder and multi-layer perceptron," *Remote Sens.*, vol. 11, no. 11, p. 1293, May 2019.
[7] W. Shao, H. Luo, F. Zhao, Y. Ma, Z. Zhao, and A. Crivello, "Indoor positioning based on fingerprint-image and deep learning," *IEEE Access*, vol. 6, pp. 74699–74712, 2018.
[8] S. Xu, R. Chen, Y. Yu, G. Guo, and L. Huang, "Locating smartphones indoors using built-in sensors and Wi-Fi ranging with an enhanced particle filter," *IEEE Access*, vol. 7, pp. 95140–95153, 2019.
[9] D. Liu, S. Guo, Y. Yang, Y. Shi, and M. Chen, "Geomagnetism-based indoor navigation by offloading strategy in NB-IoT," *IEEE Internet Things J.*, vol. 6, no. 3, pp. 4074–4084, Jun. 2019.
[10] B. Bhattarai, R. K. Yadav, H.-S. Gang, and J.-Y. Pyun, "Geomagnetic field based indoor landmark classification using deep learning," *IEEE Access*, vol. 7, pp. 33943–33956, 2019.
[11] Q. Wang, H. Luo, A. Men, F. Zhao, and Y. Huang, "An infrastructure-free indoor localization algorithm for smartphones," *Sensors*, vol. 18, no. 10, p. 3317, Oct. 2018.
[12] Q. Wang, "Light positioning: A high-accuracy visible light indoor positioning system based on attitude identification and propagation model," *Int. J. Distrib. Sens. Netw.*, vol. 14, no. 2, pp. 1–14, Feb. 2018.
[13] T. Liu, X. Zhang, Q. Li, and Z. Fang, "A visual-based approach for indoor radio map construction using smartphones," *Sensors*, vol. 17, no. 8, p. 1790, 2017.
[14] A. Poulose, O. S. Eyobu, and D. S. Han, "An indoor position-estimation algorithm using smartphone IMU sensor data," *IEEE Access*, vol. 7, pp. 11165–11177, 2019.
[15] H. Zhao, L. Zhang, S. Qiu, Z. Wang, N. Yang, and J. Xu, "Pedestrian dead reckoning using pocket-worn smartphone," *IEEE Access*, vol. 7, pp. 91063–91073, 2019.
[16] M. Zhang, Y. Wen, J. Chen, X. Yang, R. Gao, and H. Zhao, "Pedestrian dead-reckoning indoor localization based on OS-ELM," *IEEE Access*, vol. 6, pp. 6116–6129, 2018.
[17] X. Kang, B. Huang, and G. Qi, "A novel walking detection and step counting algorithm using unconstrained smartphones," *Sensors*, vol. 18, no. 1, p. 297, 2018.
[18] Q. Wang, L. Ye, H. Luo, A. Men, F. Zhao, and C. Ou, "Pedestrian Walking Distance Estimation Based on Smartphone Mode Recognition," *Remote Sens.*, vol. 11, no. 9, p. 1140, May 2019.
[19] Q. Wang, L. Ye, H. Luo, A. Men, F. Zhao, and Y. Huang, "Pedestrian stride-length estimation based on LSTM and denoising autoencoders," *Sensors*, vol. 19, no. 4, p. 840, Feb. 2019.
[20] A. Manos, I. Klein, and T. Hazan, "Gravity-based methods for heading computation in pedestrian dead reckoning," *Sensors*, vol. 19, no. 5, p. 1170, 2019.
[21] M. Edel and E. Köppe, "An advanced method for pedestrian dead reckoning using BLSTM-RNNs," in *Proc. Int. Conf. Indoor Positioning Indoor Navigat. (IPIN)*, Oct. 2015, pp. 1–6.
[22] Z.-A. Deng, G. Wang, Y. Hu, and D. Wu, "Heading estimation for indoor pedestrian navigation using a smartphone in the pocket," *Sensors*, vol. 15, no. 9, pp. 21518–21536, 2015.
[23] E. Foxlin, "Pedestrian tracking with shoe-mounted inertial sensors," *IEEE Comput. Graph. Appl.*, vol. 25, no. 6, pp. 38–46, Nov. 2005.
[24] R. Jirawimut, S. Prakoonwit, F. Cecelja, and W. Balachandran, "Visual odometer for pedestrian navigation," *IEEE Trans. Instrum. Meas.*, vol. 52, no. 4, pp. 1166–1173, Aug. 2003.
[25] S. O. H. Madgwick, A. J. L. Harrison, and R. Vaidyanathan, "Estimation of IMU and MARG orientation using a gradient descent algorithm," in *Proc. IEEE Int. Conf. Rehabil. Robot.*, Jun. 2011, pp. 1–7.
[26] *X-IMU Sensor Board*. Accessed: Apr. 15, 2019. [Online]. Available: http://x-io.co.uk/x-imu/
[27] R. Mahony, T. Hamel, and J. M. Pflimlin, "Nonlinear complementary filters on the special orthogonal group," *IEEE Trans. Autom. Control*, vol. 53, no. 5, pp. 1203–1217, Jun. 2008.

[28] B. Fan, Q. Li, and T. Liu, "How magnetic disturbance influences the attitude and heading in magnetic and inertial sensor-based orientation estimation," *Sensors*, vol. 18, no. 1, p. 76, 2018.

[29] W. Kang, S. Nam, Y. Han, and S. Lee, "Improved heading estimation for smartphone-based indoor positioning systems," in *Proc. IEEE 23rd Int. Symp. Pers., Indoor Mobile Radio Commun. (PIMRC)*, Apr. 2012, pp. 2449–2453.

[30] 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 Sens.*, vol. 11, p. 294, Jan. 2019.

[31] J. L. Marins, X. Yun, E. R. Bachmann, R. B. McGhee, and M. J. Zyda, "An extended Kalman filter for quaternion-based orientation estimation using MARG sensors," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. Expanding Societal Role Robot. Next Millennium*, vol. 4, Oct. 2002, pp. 2003–2011.

[32] A. M. Sabatini, "Quaternion-based extended Kalman filter for determining orientation by inertial and magnetic sensing," *IEEE Trans. Biomed. Eng.*, vol. 53, no. 7, pp. 1346–1356, Jun. 2006.

[33] D. Wu, L. Xia, and J. Geng, "Heading estimation for pedestrian dead reckoning based on robust adaptive Kalman filtering," *Sensors*, vol. 18, no. 6, p. 1970, Jun. 2018.

[34] H. Fourati, N. Manamanni, L. Afilal, and Y. Handrich, "Complementary observer for body segments motion capturing by inertial and magnetic sensors," *IEEE/ASME Trans. Mechatronics*, vol. 19, no. 1, pp. 149–157, Nov. 2014.

[35] T. S. Yoo, S. K. Hong, H. M. Yoon, and S. Park, "Gain-scheduled complementary filter design for a MEMS based attitude and heading reference system," *Sensors*, vol. 11, no. 4, pp. 3816–3830, 2011.

[36] Q. Fan, H. Zhang, P. Pan, X. Zhuang, J. Jia, P. Zhang, Z. Zhao, G. Zhu, and Y. Tang, "Improved pedestrian dead reckoning based on a robust adaptive Kalman filter for indoor inertial location system," *Sensors*, vol. 19, no. 2, p. 294, Jan. 2019.

[37] A. Bilke and J. Sieck, "Using the magnetic field for indoor localisation on a mobile phone," in *Progress in Location-Based Services* (Lecture Notes in Geoinformation and Cartography). Berlin, Germany: Springer, 2013, pp. 195–208.

[38] W. Kang and Y. Han, "SmartPDR: Smartphone-based pedestrian dead reckoning for indoor localization," *IEEE Sensors J.*, vol. 15, no. 5, pp. 2906–2916, May 2015.

[39] R. Gao, "Sextant: Towards ubiquitous indoor localization service by photo-taking of the environment," *IEEE Trans. Mobile Comput.*, vol. 15, no. 2, pp. 460–474, Mar. 2016.

[40] B. Wang, X. Liu, B. Yu, R. Jia, and X. Gan, "Pedestrian dead reckoning based on motion mode recognition using a smartphone," *Sensors*, vol. 18, no. 6, p. 1811, Jun. 2018.

[41] I. Klein, Y. Solaz, and G. Ohayon, "Pedestrian dead reckoning with smartphone mode recognition," *IEEE Sensors J.*, vol. 18, no. 18, pp. 7577–7584, Sep. 2018.

[42] M. Elhoushi, J. Georgy, A. Noureldin, and M. J. Korenberg, "Motion mode recognition for indoor pedestrian navigation using portable devices," *IEEE Trans. Instrum. Meas.*, vol. 65, no. 1, pp. 208–221, Jan. 2015.

[43] Z. Deng, X. Liu, Z. Qu, C. Hou, and W. Si, "Robust heading estimation for indoor pedestrian navigation using unconstrained smartphones," *Wireless Commun. Mobile Comput.*, vol. 2018, pp. 1–11, Jul. 2018.

[44] K. Kunze, P. Lukowicz, K. Partridge, and B. Begole, "Which way am I facing: Inferring horizontal device orientation from an accelerometer signal," in *Proc. Int. Symp. Wearable Comput.*, Sep. 2009, pp. 149–150.

[45] U. Steinhoff and B. Schiele, "Dead reckoning from the pocket—An experimental study," in *Proc. IEEE Int. Conf. Pervasive Comput. Commun. (PerCom)*, Mar. 2010, pp. 162–170.

[46] S. A. Hoseinitabatabaei, A. Gluhak, R. Tafazolli, and W. Headley, "Design, realization, and evaluation of uDirect—An approach for pervasive observation of user facing direction on mobile phones," *IEEE Trans. Mobile Comput.*, vol. 13, no. 9, pp. 1981–1994, Apr. 2014.

[47] M. H. Afzal, V. Renaudin, and G. Lachapelle, "Use of earth's magnetic field for mitigating gyroscope errors regardless of magnetic perturbation," *Sensors*, vol. 11, no. 12, pp. 11390–11414, 2011.

[48] F. Zampella, A. R. J. Ruiz, and F. S. Granja, "Indoor positioning using efficient map matching, RSS measurements, and an improved motion model," *IEEE Trans. Veh. Technol.*, vol. 64, no. 4, pp. 1304–1317, Apr. 2015.

[49] H. Bao and W. C. Wong, "A novel map-based dead-reckoning algorithm for indoor localization," *J. Sens. Actuator Netw.*, vol. 3, no. 1, pp. 44–63, 2014.

[50] Q. Tian, Z. Salcic, K. I.-K. Wang, and Y. Pan, "A hybrid indoor localization and navigation system with map matching for pedestrians using smartphones," *Sensors*, vol. 15, no. 12, pp. 30759–30783, Dec. 2015.

[51] L. Pei, D. Liu, D. Zou, R. L. F. Choy, Y. Chen, and Z. He, "Optimal heading estimation based multidimensional particle filter for pedestrian indoor positioning," *IEEE Access*, vol. 6, pp. 49705–49720, 2018.

[52] F. Holzke, J.-P. Wolff, and C. Haubelt, "Improving pedestrian dead reckoning using likely paths and backtracking for mobile devices," in *Proc. IEEE Int. Conf. Pervasive Comput. Commun. Workshops (PerCom Workshops)*, Mar. 2019, pp. 273–278.

[53] J. Qian, L. Pei, J. Ma, R. Ying, and P. Liu, "Vector graph assisted pedestrian dead reckoning using an unconstrained smartphone," *Sensors*, vol. 15, no. 3, pp. 5032–5057, 2015.

[54] L. Ma, Y. Fan, Y. Xu, and Y. Cui, "Pedestrian dead reckoning trajectory matching method for radio map crowdsourcing building in WiFi indoor positioning system," in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2017, pp. 1–6.

[55] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, May 2015.

[56] S. Cortes, A. Solin, and J. Kannala, "Deep learning based speed estimation for constraining strapdown inertial navigation on smartphones," in *Proc. IEEE 28th Int. Workshop Mach. Learn. Signal Process. (MLSP)*, Sep. 2018, pp. 1–6.

[57] C. Chen, X. Lu, A. Markham, and N. Trigoni, "IONet: Learning to cure the curse of drift in inertial odometry," in *Proc. Conf. Artif. Intell. (AAAI)*, 2018, pp. 6468–6476.

[58] C. Chen, P. Zhao, C. X. Lu, W. Wang, A. Markham, and N. Trigoni, "OxIOD: The dataset for deep inertial odometry," Sep. 2018, *arXiv:1809.07491*. [Online]. Available: https://arxiv.org/abs/1809.07491

[59] H. Yan, Q. Shan, and Y. Furukawa, "RIDI: Robust IMU double integration," in *Proc. Eur. Conf. Comput. Vis.*, in Lecture Notes in Computer Science, vol. 11217, 2018, pp. 641–656.

[60] H. Yan, S. Herath, and Y. Furukawa, "RoNIN: Robust neural inertial navigation in the wild: Benchmark, evaluations, and new methods," May 2019, *arXiv:1905.12853*. [Online]. Available: https://arxiv.org/abs/1905.12853

[61] C. Chen, Y. Miao, C. X. Lu, P. Blunsom, A. Markham, and N. Trigoni, "Transferring physical motion between domains for neural inertial tracking," in *Proc. NIPS*, Oct. 2018, pp. 1–4.

[62] S. Cortés, A. Solin, E. Rahtu, and J. Kannala, "ADVIO: An authentic dataset for visual-inertial odometry," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, in Lecture Notes in Computer Science, vol. 11214, 2018, pp. 425–440.

[63] R. Pelossof, I. Singh, J. L. Yang, M. T. Weirauch, T. R. Hughes, and C. S. Leslie, "Affinity regression predicts the recognition code of nucleic acid–binding proteins," *Nature Biotechnol.*, vol. 33, no. 12, pp. 1242–1249, Dec. 2015.

[64] A. J. Viterbi, "Error bounds for convolutional codes and an asymptotically optimum decoding algorithm," *IEEE Trans. Inf. Theory*, vol. 13, no. 2, pp. 260–269, Apr. 1967.

[65] R. Q. Charles, H. Su, M. Kaichun, and L. J. Guibas, "PointNet: Deep learning on point sets for 3D classification and segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jan. 2017, pp. 77–85.

[66] Vicon. Accessed: Aug. 15, 2019. [Online]. Available: http://www.vicon.com

[67] *NGIMU Sensor Board*. Accessed: Apr. 15, 2019. [Online]. Available: http://x-io.co.uk/ngimu/

QU WANG received the B.S. degree from the School of Software Engineering, Beijing University of Posts and Telecommunication, China, in 2013, and the M.S. degree from the University of Chinese Academy of Sciences, Beijing, China, in 2016. He is currently pursuing the Ph.D. degree with the School of Information and Communication Engineering, Beijing University of Posts and Telecommunications. His current main interests include location-based services, pervasive computing, computer vision, and machine learning.

HAIYONG LUO received the B.S. degree from the Department of Electronics and Information Engineering, Huazhong University of Science and Technology, Wuhan, China, in 1989, the M.S. degree from the School of Information and Communication Engineering, Beijing University of Posts and Telecommunication, China, in 2002, and the Ph.D. degree in computer science from the University of Chines Academy of Sciences, Beijing, China, in 2008. He is currently an Associate Professor with the Institute of Computer Technology Chinese Academy of Science, China. His main research interests include location-based services, pervasive computing, mobile computing, and the Internet of Things.

LANGLANG YE received the B.S. degree from the School of Basic Medical Science, Zhejiang University, in 2016. He is currently pursuing the M.S. degree in computer technology with the University of Chinese Academy of Sciences. His main research interests include location-based services, data mining, and deep learning.

AIDONG MEN received the B.S., M.S., and Ph.D. degrees from the Department of Radio Engineering, Beijing University of Posts and Telecommunications, Beijing, in 1994. From 1994 to 2000, he was an Associate Professor with the Department of Radio Engineering, Beijing University of Posts and Telecommunications. Since 2000, he has been a Professor with Telecom Engineering College, Beijing University of Posts and Telecommunications. He has published over 100 articles in journals and international conference. His research interests include multimedia communication, digital TV, images and speech signal processing and transmission, and so on. He is a Fellow of the Chinese Institute of Electronics and China Institute of Communications. He is also an Invited Fellow of the Science and Technology Committee of State Administration of Radio, Film, and Television.

FANG ZHAO received the B.S. degree from the School of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan, China, in 1990, and the M.S. and Ph.D. degrees in computer science and technology from the Beijing University of Posts and Telecommunications, Beijing, China, in 2004 and 2009, respectively. She is currently a Professor with the School of Software Engineering, Beijing University of Posts and Telecommunication. Her research interests include mobile computing, location-based services, and computer networks.

YAN HUANG received the B.S. degree from Guangxi Normal University, in 2013, and the M.S. degree from the Institute of Software, Chinese Academy of Sciences, Beijing, China, in 2017. She is currently pursuing the Ph.D. degree with Peking University, China. Her current main interests include location-based services, optical-based location and communication, computer vision, and machine learning.

CHANGHAI OU was born in Tongren, Guizhou, China, in 1989. He received the bachelor's degree in computer science and technology from the School of Computer and Information Technology, Beijing Jiaotong University, China, in 2013, and the Ph.D. degree in cyber security from the Institute of Information Engineering, Chinese Academy of Sciences (i.e., School of Cyber Security, University of Chinese Academy of Sciences), in July 2018. He is currently a Research Fellow with the Hardware & Embedded Systems Laboratory (HESL), School of Computer Science and Engineering, Nanyang Technological University, Singapore. His current research interests include cryptographic hardware and embedded system security, side channel attacks, machine learning, and privacy preserving.