

Pedestrian Motion Tracking by Using Inertial Sensors on the Smartphone

Yingying Wang*, Hu Cheng* and Max Q.-H. Meng[✉], *Fellow, IEEE*

Abstract—Inertial Measurement Unit (IMU) has long been a dream for stable and reliable motion estimation, especially in indoor environments where GPS strength limits. In this paper, we propose a novel method for position and orientation estimation of a moving object only from a sequence of IMU signals collected from the phone. Our main observation is that human motion is monotonous and periodic. We adopt the Extended Kalman Filter and use the learning-based method to dynamically update the measurement noise of the filter. Our pedestrian motion tracking system intends to accurately estimate planar position, velocity, heading direction without restricting the phone’s daily use. The method is not only tested on the self-collected signals, but also provides accurate position and velocity estimations on the public RIDI dataset, i.e., the absolute transmit error is 1.28m for a 59-second sequence.

I. INTRODUCTION

The need for accurate and reliable indoor localization in where the global positioning system (GPS) is not serviced sufficiently has continued to grow, such as customer navigation in supermarkets, augmented reality in public places. The smartphone can be a critical part of human indoor localization solution because of the wide range of sensors and its powerful computational ability [1]. Meanwhile, the cost of using the phone is negligible, because almost every one equips with one phone.

There are kinds of approaches used in indoor localization based on a phone. Traditionally, the signal between a transmitter and a receiver is used. For example, time of arrival (TOA) is used by using the signal’s travel time from transmitter to receiver to calculate the distance [2]. Angle of arrival (AOA) is the method that determines the direction of the incoming signal by exploiting and detecting phase difference among antennas [3], [4]. There is also the distance information reflected by the strength of the received signal [5]. These transmitted signal based methods require at least two types of equipment and the accuracy suffers due to the huge changeability of the environments, such as moving crowds. Fingerprinting method is conducted by the unique signal distribution at a specific place [6]. Liang et al. [7] combined multi-opportunistic signals, including Wifi,

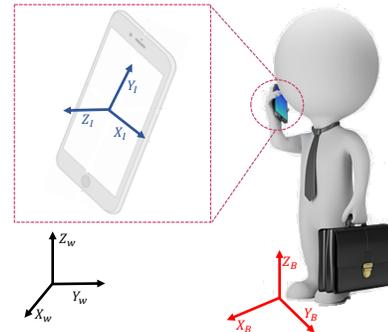


Fig. 1: Three coordinate frames used in this paper: W is the world frame (black), I represents the IMU frame (Blue), which is fixed with the phone, and B stands for the ideal body frame (red).

magnet, and lights, for offline mapping and online localization afterward. Camera-based localization has produced high precision accuracy and visual-inertial odometry combines visual features and IMU signals to produce a more robust result [8], [9], such as Google Tango project [10]. These kinds of methods need the camera to be exposed, which is power consumed and privacy affected, and the accuracy is closely connected with the lights of the environment.

IMU double integration for motion estimation has always been attractive to people. First, IMU is energy efficient. It is equipped with every phone and capable of running a whole day without much battery consumption. Second, IMU works anywhere without any prior regional knowledge needed, and the tracking result would not be affected by the environments. The idea for IMU motion estimation is simple, given the orientation by the Android platform, subtracting the gravity from acceleration, integrating the residual acceleration once to get velocity, integrating again to obtain the position. However, small drift in acceleration would result in huge bias in position, in fact, a bias of σ has an impact of $\sigma t^2/2$ in position after t seconds. High precision navigation systems, like a million-dollar military-grade IMU, achieve very small errors, but are too costly in daily use.

Based on the observations that human motions are repetitive and periodic [11], there are various IMU bias correction methods. For example, the widespread Zero velocity UpdaTe (ZUPT) [12], in which the IMU is attached on the foot, and then the velocity is corrected to zero each time the foot is connected to the ground. In recent years, due to the rapid development of deep learning and the computation ability of

*Yingying Wang and Hu Cheng contributed equally to this work.

[✉]Corresponding author, max.meng@ieee.org

Yingying Wang and Hu Cheng are with the Robotics, Perception and Artificial Intelligence Lab in the Department of Electronic Engineering, The Chinese University of Hong Kong, N.T., Hong Kong SAR, China.

Max Q.-H. Meng is with the Department of Electronic and Electrical Engineering of the Southern University of Science and Technology in Shenzhen, China, on leave from the Department of Electronic Engineering, The Chinese University of Hong Kong, Hong Kong, and also with the Shenzhen Research Institute of the Chinese University of Hong Kong in Shenzhen, China.

hardware, learning-based methods are gaining much more interest in inertial navigation. In [13], Yan et al. proposed a robust IMU double integration (RIDDI) method, in which the velocity is generated by using support vector regression according to different phone placements. In [14] and [15], recurrent neural networks are used for end-to-end inertial navigation.

In this paper, we propose to only utilize the IMU of the phone to realize the pedestrian motion tracking. The Invariant Extended Kalman Filter (IEKF) [16] and deep learning-based measurement noise adapter are used to reduce the noise in linear acceleration. The main contributions of this paper are summarized as follows:

1. We propose the ideal body coordinate frame (Fig. 1) to represent the heading direction, to avoid different estimates of the same motion because of changing the smartphone's holding methods.

2. Our method takes advantage of the IEKF and the learning-based measurement noise adapter. The observation of the IEKF are designed as 3 scalars (velocity in the body frame).

3. Our method is trained by a single object's motion data, and estimates the position and orientation accurately even on the public dataset, in which the data is collected by different object and device.

The remainder of this paper is organized as follows: We first describe the related methods on IMU model and IEKF model in Sec. II. The system framework is declared in Sec. III. The implementation details of the system as well as the results are stated in Sec. IV, and V, respectively. Finally, the conclusion and future work of this paper are stated in Sec. VI.

II. RELATED WORK

An IMU is usually a combination of the gyro, the acceleration and magnetometer. Given a fixed coordinate frame and the platform's initial configuration (\mathbf{R}_0^i , \mathbf{v}_0^i , \mathbf{p}_0^i), the following orientation \mathbf{R}_n^i , velocity \mathbf{v}_n^i and position \mathbf{p}_n^i could be calculated by using the angular velocity and acceleration. Kalman Filter is one of the most widely used methods in motion estimation because it is both extremely simple and general, and we introduce the state-of-the-art IEKF in this section.

A. IMU model

In this paper, we use the linear acceleration provided by the smartphone, whose gravity has already been removed from the acceleration. The output linear acceleration \mathbf{a}_n^i and the angular velocity $\boldsymbol{\omega}_n^i$ are in response to the true linear acceleration $\bar{\mathbf{a}}_n^i$ and the true angular velocity $\bar{\boldsymbol{\omega}}_n^i$ [17]:

$$\mathbf{a}_n^i = \bar{\mathbf{a}}_n^i + \mathbf{b}_n^a + \mathbf{w}_n^a \quad (1)$$

$$\boldsymbol{\omega}_n^i = \bar{\boldsymbol{\omega}}_n^i + \mathbf{b}_n^\omega + \mathbf{w}_n^\omega \quad (2)$$

The \mathbf{b}_n^a and \mathbf{b}_n^ω are constant biases, \mathbf{w}_n^a and \mathbf{w}_n^ω are zero-mean Gaussian noises. The bias \mathbf{b}_n^a and \mathbf{b}_n^ω follow random

walks:

$$\mathbf{b}_{n+1}^a = \mathbf{b}_n^a + \mathbf{w}_n^{w^{b^a}} \quad (3)$$

$$\mathbf{b}_{n+1}^\omega = \mathbf{b}_n^\omega + \mathbf{w}_n^{w^{b^\omega}} \quad (4)$$

Where $\mathbf{w}_n^{w^{b^a}}$ and $\mathbf{w}_n^{w^{b^\omega}}$ are zero-mean Gaussian noise. Using the Lie group [16], the forward kinematic functions are:

$$\mathbf{R}_{n+1}^i = \mathbf{R}_n^i \exp((\bar{\boldsymbol{\omega}}_n dt)_\times) \quad (5)$$

$$\mathbf{v}_{n+1}^i = \mathbf{v}_n^i + \mathbf{R}_n^i \bar{\mathbf{a}}_n dt \quad (6)$$

$$\mathbf{p}_{n+1}^i = \mathbf{p}_n^i + \mathbf{v}_n^i dt \quad (7)$$

The $(\cdot)_\times$ in equation (5) represents the skew-symmetric matrix associated with the cross product of a 3-dimensional vector. Specifically, in our paper, \mathbf{R}_n^i is the IMU frame with respect to the world frame, \mathbf{v}_n^i and \mathbf{p}_n^i are velocity and position of the IMU in the world frame.

B. Invariant Extended Kalman Filter (IEKF)

Extended Kalman Filter (EKF) is designed for bias correction from a series of measurements with statistical noise and other inaccuracies, which is calculated by linearizing a non-linear function around the average of the current state. The algorithm contains two-step processes: prediction and update step. In the first step, the next step variables with their uncertainties are estimated according to the current state and the measured input variable \mathbf{u}_n :

$$\hat{\mathbf{x}}_{n+1} = f(\mathbf{x}_n, \mathbf{u}_n) + \boldsymbol{\omega}_n \quad (8)$$

$$\hat{\mathbf{P}}_{n+1} = \mathbf{F}_n \mathbf{P}_n \mathbf{F}_n^T + \mathbf{G}_n \mathbf{Q}_n \mathbf{G}_n^T \quad (9)$$

The $\boldsymbol{\omega}_n$ is the process noise and is assumed to be zero-mean Gaussian noise with covariance matrix \mathbf{Q}_n . The covariance matrix \mathbf{P}_n is the uncertainty associate with the state \mathbf{x}_n . The \mathbf{F}_n and \mathbf{G}_n are Jacobian matrices of $f(\cdot)$ with respect to \mathbf{x}_n and \mathbf{u}_n .

The second step is to update the next time state \mathbf{x}_{n+1} based on the observation function:

$$\hat{\mathbf{y}}_{n+1} = h(\hat{\mathbf{x}}_{n+1}) + \mathbf{n}_{n+1} \quad (10)$$

Where $h(\cdot)$ is the observed function, and the \mathbf{n}_{n+1} is the observation noise which is assumed to be zero-mean Gaussian white noise with covariance \mathbf{N}_n : $\mathbf{n}_n \sim \mathcal{N}(0, \mathbf{N}_n)$. The covariance matrix \mathbf{N}_n is set by the user, the smaller \mathbf{N}_n respects the higher accuracy of the $\hat{\mathbf{x}}_{n+1}$ in equation (8).

The following steps are computing the Kalman gain \mathbf{K} and update the state and its covariance matrix:

$$\mathbf{K} = \hat{\mathbf{P}}_{n+1} \mathbf{H}^T (\mathbf{H} \hat{\mathbf{P}}_{n+1} \mathbf{H}^T + \mathbf{R})^{-1} \quad (11)$$

$$\mathbf{x}_{n+1} = \hat{\mathbf{x}}_{n+1} + \mathbf{K}(\mathbf{y}_{n+1} - \hat{\mathbf{y}}_{n+1}) \quad (12)$$

$$\mathbf{P}_{n+1} = (\mathbf{I} - \mathbf{K} \mathbf{H}) \hat{\mathbf{P}}_{n+1} \quad (13)$$

Where \mathbf{H} is the Jacobian of $h(\cdot)$ with respect to \mathbf{x}_n , \mathbf{y}_{n+1} is the measured observations.

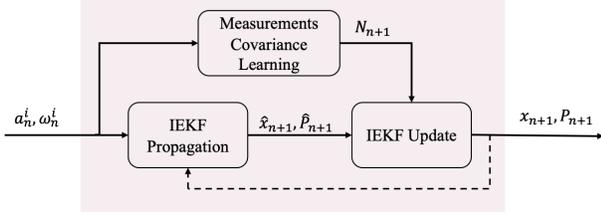


Fig. 2: The structure of the whole system. The dotted black line represents the state of $n + 1$ is the input of $n + 2$. The system yields a real-time estimate of the state x_{n+1} along with covariance P_{n+1} from raw linear acceleration and angular velocity of the smartphone signal only.

III. SYSTEM DESIGN

In this section, we will first give an overview of our system, and then introduce the key issues in the whole system design.

A. System Overview

The main observation of this paper is that human walkings are normally slow and we only focus on planar motions and leave non-planar movements for future work, thus the vertical velocity is null. Based on these assumptions, we use the traditional IMU signal processing model, and IEKF for the bias correction. For the measurement noise matrix in IEKF, a learning-based method is used to adapt with different human walking. The framework of the system is shown as Fig. 2.

Given the initial configuration (R_0^i, v_0^i, p_0^i) , our purpose is to estimate (R_n^i, v_n^i, p_n^i) only from the linear acceleration a_n^i and angular velocity ω_n^i . We observed that people's daily movements are only in a fixed orientation. As shown in Fig. 1, the walking direction is X_B in the body frame, and the velocity along with Y_B and Z_B are zero, thus $v_b = [v_{bx}, 0, 0]$ could be used as observations. Our state variables are:

$$x_n = (R_n^i, v_n^i, p_n^i, b_n^\omega, b_n^a, R_{in}^b) \quad (14)$$

The R_{in}^b is the orientation discrepancy of body frame with respect to the IMU frame.

Our state-space is similar as in [18], except R_{in}^b . The system in [18] is designed for motion estimation of car. The difference between human tracking and [18] is that in the car tracking system, the IMU is fixed to the car, and the orientation misalignment between the car frame and the IMU frame approximately equals to a three-dimensional identity matrix. The forward direction of the car could also roughly be treated as of the IMU moving direction, this is also why the IMU has null lateral and vertical velocities in the car frame. In our indoor tracking problem, the orientation from IMU frame to the body frame is changeable, and the IMU's velocity expressed in the body frame is also dynamic and non-zero in lateral or upward direction.

B. IEKF propagation

The propagations of $(R_n^i, v_n^i, p_n^i, b_n^\omega, b_n^a)$ are already given by equations (3) to (7). The floor of indoor building is

usually parallel to the horizontal plane, thus, the z axes of the body frame and the world frame are in the same direction. The initial velocity in the world frame $v_0^i = [v_{0x}^i, v_{0y}^i, v_{0z}^i]$ is given, and the initial velocity in the body frame v_{b0} could be calculated:

$$v_{b0} = R_0^{bT} v_0^i \quad (15)$$

$$= \left[\sqrt{v_{0x}^i{}^2 + v_{0y}^i{}^2}, 0, v_{0z}^i \right] \quad (16)$$

According to equation (15) and (16), the initial rotation from the body frame to the world frame R_0^b could be obtained. Then R_{i0}^b could also be calculated according to property of continuous rotation [19]:

$$R_{i0}^b = R_0^{iT} R_0^b \quad (17)$$

We observed that the orientation misalignment between the body frame and the smartphone-based IMU frame is an approximate constant in a period of time due to the repeatability and periodic of both human motion and behavior. Thus, we can model the noise of R_{in}^b as a zero-mean Gaussian noise:

$$R_{i(n+1)}^b = R_{in}^b \exp((\omega_n^{R^b})_\times) \quad (18)$$

Now we have decided the $f(\cdot)$ of equation (8), and the process covariance matrix Q_n is set as a constant matrix. The Jacobians F_n and G_n could also be calculated. For the rotation matrix propagation function (5) and (18), we use the first-order Taylor expansion on $\exp(\omega t) = I + \omega t$ to realize the derivate.

$$F_n = I_{18 \times 18} + \begin{bmatrix} 0 & 0 & 0 & -R_n^i & 0 & 0 \\ 0 & 0 & 0 & -(v_n^i)_\times R_n^i & -R_n^i & 0 \\ 0 & I_{3 \times 3} & 0 & -(p_n^i)_\times R_n^i & -0 & 0 \\ 0 & 0 & 0 & -0 & 0 & 0 \end{bmatrix} \quad (19)$$

$$G_n = \begin{bmatrix} R_n^i & 0 & 0_{3 \times 9} \\ (v_n^i)_\times R_n^i & R_n^i & 0_{3 \times 9} \\ (p_n^i)_\times R_n^i & 0 & 0_{3 \times 9} \\ 0 & 0 & 0_{3 \times 9} \end{bmatrix} \quad (20)$$

C. Pseudo-Measurements

The velocity in the body frame could be expressed as:

$$v_{bn} = R_n^i R_{in}^{bT} v_n^i \quad (21)$$

We could further determine the Jacobian of observed function:

$$H = \left[0 \quad R_n^i R_{in}^{bT} \quad 0_{3 \times 12} \right] \quad (22)$$

In [18], the null lateral and vertical velocities are used as observations, while there is no constraint in the forward direction. The method works fine in the car inertial navigation system, but in our smartphone-based tracking, the lateral and vertical constraints make limited effect to the forward direction. As shown in Fig. 3, the forward velocity would drift a lot within the first 5 seconds.

In this paper, except for the null lateral and vertical velocity, we utilize the periodic and slowness of human

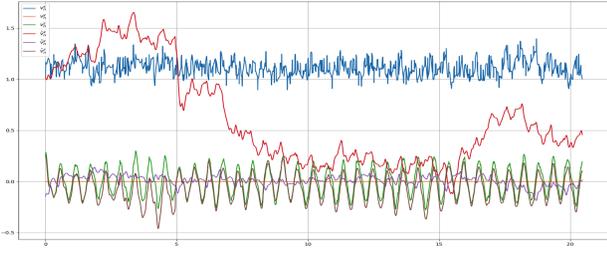


Fig. 3: The velocity in the body frame without observation in v_{bx} . The ground truth of the velocity along with x axis (blue line) is a quasi-constant, while the calculated velocity (red line) deviated a lot in only 20s.

walking, which indicates that human motions are relatively slow and smooth. We treat the average speed of the past 5 seconds as the current speed:

$$\mathbf{v}_{b(n+1)x} = \sqrt{(\mathbf{p}_n^i - \mathbf{p}_{n-win*5}^i)/5} \quad (23)$$

Where win in (23) is the sampling rate. Moreover, we utilize the constraints of human movements. In daily life, the walking velocity is usually slower than $2m/s$. Thus, we have generated 3-scalar pseudo-observations:

$$\mathbf{y}_{n+1} = [\mathbf{v}_{b(n+1)x} \quad \mathbf{0} \quad \mathbf{0}] \quad (24)$$

D. Learning-based Measurement Noise Parameter Adapter

As shown in the equation (10), the covariance \mathbf{N}_n is used at each instant n for the filter update. The measurement noise parameter adapter dynamically computes covariances meant to improve the localization accuracy by just using the linear acceleration \mathbf{a}_n^i and angular velocity $\boldsymbol{\omega}_n^i$. The main component of the adapter is a Convolutional Neural Network (CNN) [20], whose input is a window of M \mathbf{a}_n^i and $\boldsymbol{\omega}_n^i$. A relatively simple network with small number of parameters is used to avoid over-fitting and made its output independent of the state estimates.

The total components of the adapter consist of several full layers followed by a full layer outputting a vector $\mathbf{Z}_n = 3 * [\tanh(z_n^{fw}), \tanh(z_n^{lat}), \tanh(z_n^{up})]$. The output covariance matrix \mathbf{N}_{n+1} is then computed by:

$$\mathbf{N}_{n+1} = \text{diag}(\sigma_{fw}^2 10^{\mathbf{Z}_n[1]}, \sigma_{lat}^2 10^{\mathbf{Z}_n[2]}, \sigma_{up}^2 10^{\mathbf{Z}_n[3]}) \quad (25)$$

Where σ_{fw}^2 , σ_{lat}^2 and σ_{up}^2 are our initial guesses for the measurement noise parameters. The output of the network may expand the covariance to $1000\sigma^2$ and compress it up to $10^{-3}\sigma^2$.

IV. IMPLEMENTATION DETAILS

We provide the settings and implementation details of our method in this section. The inertial signals are collected by smartphone. The latter process is implemented in Python with Pytorch [21] library.



Fig. 4: Different phone holding methods in daily life, while exposing the camera (green box) all the time to collect the ground-truth motions by visual-inertial odometry. The configurations from left to right are: playing while walking; putting in the pocket and holding in the hand.

A. Data collection and preprocessing

We use a Google Tango phone, ASUS Zenfone AR, to record the \mathbf{a}_n^i , $\boldsymbol{\omega}_n^i$, and 3D camera poses. We make sure that the camera has not been blocked all the time (see Fig. 4). The signals we used for training are from the same person with different motions.

The sampling rate of the collected data is 200Hz. Asynchronous signals from various sensors are synchronized into the timestamps of Tango poses by using linear interpolation, which is a preprocessing step of our system. The method of collecting the data is related to RIDI dataset, but the constraints of our phone holding methods are looser, all the three phone carrying manners are daily, instead of strapped tightly to the body. We also use the RIDI dataset to verify our system.

B. System Settings

The initial parameters of IEKF are as follows:

$$\mathbf{P}_0 = \text{diag}(\sigma_0^R \mathbf{I}, \sigma_0^v \mathbf{I}, \mathbf{0}_3, \sigma_0^{b\omega} \mathbf{I}, \sigma_0^{b^a} \mathbf{I}, \sigma_0^{R^b} \mathbf{I})^2 \quad (26)$$

$$= \text{diag}(10^{-6} \mathbf{I}, 10^{-5} \mathbf{I}, \mathbf{0}_3, 10^{-6} \mathbf{I}, 10^{-3} \mathbf{I}, 10^{-5} \mathbf{I}) \quad (27)$$

$$\mathbf{Q}_0 = \text{diag}(\sigma_\omega \mathbf{I}, \sigma_a \mathbf{I}, \sigma_{b\omega} \mathbf{I}, \sigma_{b^a} \mathbf{I}, \sigma_{R^b} \mathbf{I})^2 \quad (28)$$

$$= \text{diag}(2 \cdot 10^{-4} \mathbf{I}, 10^{-3} \mathbf{I}, 10^{-6} \mathbf{I}, 2 \cdot 10^{-5} \mathbf{I}, 10^{-3} \mathbf{I}) \quad (29)$$

$$\mathbf{N}_0 = \text{diag}(\sigma_{fw}^2, \sigma_{lat}^2, \sigma_{up}^2) = \text{diag}(3, 2, 0.2) \quad (30)$$

Where \mathbf{I} stands for $\mathbf{I}_{3 \times 3}$. Except for the 3 scalars output layer, the adapter was designed as a 1D temporal convolutional neural network with 2 layers. To obtain a measurement covariance, we want to it is related to last 20 frames of the \mathbf{a}_n^i , $\boldsymbol{\omega}_n^i$. Thus, for the first layer, $in_channels = 6$, $out_channels = 32$, $kernel_size = 6$, $dilation = 2$, while for the second layer, $in_channels = 32$, $out_channels = 32$, $kernel_size = 5$, $dilation = 3$.

C. Training

We use a batch size of 400, and for each single batch, six 40-second sequences (8000 sampling timestamps) are sampled. The minimum length of the six sequences is 28000. During training, we use an Adam optimizer with an initial learning rate of 0.001 and the minimum learning rate of

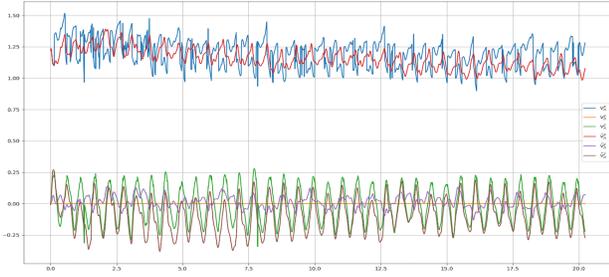


Fig. 5: The velocity in the body frame with observation in v_{bx} . The estimation of the velocity along with x axis (red line) is almost in the same value with the ground truth (blue line), and the discrepancy would be further minimized by the measurement noise adapter.

10^{-5} . For any CNN layer, we apply dropout with the keep probability of 0.5. The loss is determined by:

$$loss = \frac{1}{8000} \sum_i^{8000} \|v_{bi} - v_{b_{gti}}\|^2 \quad (31)$$

Where $v_{b_{gti}}$ is the ground-truth velocity in the body frame. Given the 3D camera pose and the timestamp, the velocity in the world frame could be obtained. The velocity and the orientation of the ideal body frame could then be calculated by using equation (15) and (16). The v_{bi} is the velocity in the body frame calculated from the output of IEKF.

For the test set, we use three different types of data: the whole sequences used in the training data; the signals from the same object but not in the training set; the data from RIDI dataset, which is from different carrier and device.

V. RESULTS AND ANALYSIS

In this section, the results of our system are provided, including the pseudo-observations of IEKF, and the position trajectories of three different test datasets (Fig. 6). We compare four methods which are all IMU-based estimations: NDI denotes the naive double integration of the raw linear accelerations; RIDI is the regressed trajectory proposed in [13], and the training data is also from [13]; IEKF represents the p_n^i and Inter is the integration of the v_n^i in the state variables.

We use two standard metrics proposed in [22]: Absolute Trajectory Error (ATE) defined as the average Root Mean Squared Error (RMSE) between estimated and ground-truth trajectories as a whole; Relative Trajectory Error (RTE) defined as the average RMSE over a fixed time interval, i.e. 1 minute in our evaluation.

A. Pseudo-Observation

By adding observation in the forward direction, the velocity in the body frame along the X_B axis could be highly corrected. Fig. 5 shows the estimated velocity in the body frame at the started batch. Compared with Fig. 3, the walking velocity is corrected and the bias is minimized with dynamic measurement covariance based on deep learning.

B. Seen Motion

We notice that the estimated velocity v_n^i , one of the state variables of the IEKF, could be almost accurately calculated, even with two turns within 50 seconds (see Fig. 7). The estimated p_n^i is not as robust as the velocity, and the error drifts over time. Surprisingly, the integration of v_n^i performs a better position estimation (IEKF and Inter in Fig. 6a), in terms of the start and end point, and the calculated RTE (TABLE I).

C. Unseen Motion

For the sequence not contained in the training set, the estimation of our system is guaranteed, too (Fig. 6b). In this simple straight walk with a left turn, the output position estimate of IEKF outperforms Inter, which may come from the fact that Kalman Filter is more likely to converge within small noise.

TABLE I: Positional accuracy evaluations (in meters)

Metric	Method	Seen	Unseen	RIDI data	Averaged
ATE	NDI	446.80	149.52	17.44	204.59
	RIDI	41.95	1.24	0.77	14.65
	IEKF	3.92	0.91	1.28	2.04
	Inter	3.96	1.68	0.95	2.20
RTE	NDI	34.25	249.11	33.19	105.52
	RIDI	52.27	1.80	0.90	18.32
	IEKF	3.63	0.85	0.28	1.59
	Inter	2.85	1.76	0.98	1.87

D. Data in RIDI Dataset

The test results of RIDI data is shown as Fig. 6c, both the IEKF and Inter methods perform well towards the RIDI data, especially the RTE of IEKF (0.28m) is much lower than RIDI method itself (0.90m), whose training data is from the same device as the test set. Besides, when integrating the raw data twice directly, the ATE of RIDI data is much smaller than our data, which also indicates the extensiveness of our data and the robustness of our method.

From the tests, and the average error of position from IEKF and Inter, we conclude that when the sequence is within 1 min, the IEKF could provide accurate position estimation, but when the time is longer, the Inter method would be more accurate, which may benefit from the velocity in the body frame as the loss function.

VI. CONCLUSION AND FUTURE WORK

We propose a novel approach for inertial pedestrian dead reckoning in this paper. The neural network is exploited to dynamically adapt the covariance of the EKF that performs localization, velocity and sensor bias estimation. The whole system is fed with the IMU signals collected by the smart-phone only. The method leads to surprisingly accurate results not only for the self-collected data but for the data in public dataset. The position state variable has a higher accuracy within 1-minute walking, while the integration of the velocity variable outperforms in longer walkings. In the future, we may take advantage of these two variables to further improve positional accuracy.

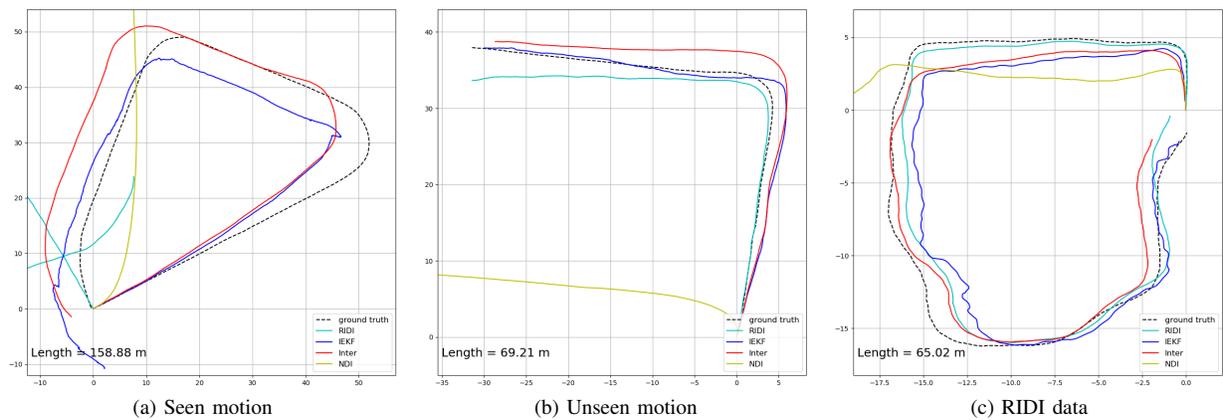


Fig. 6: Estimated positions: ground truth (black dashed), RIDI (cyan), IEKF (blue), Inter (red) and NDI (yellow). The seen motion is a 145s sequence used in the training data and the phone was playing in the hand while walking. Unseen motion is with the same object as the seen one, but is not contained in the training set, and it is a 65s sequence with the phone holding in the hand. The RIDI data is from the public dataset, which is collected by fixed the phone to the body within 59s.

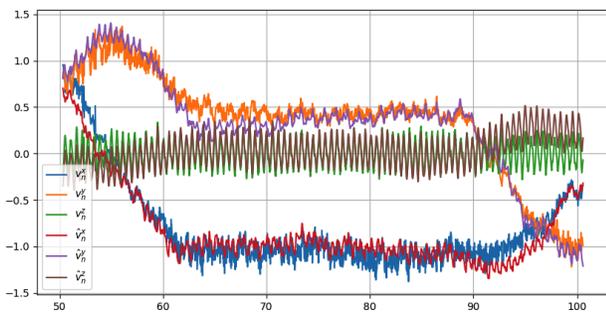


Fig. 7: The ground-truth (v_n^x, v_n^y, v_n^z) and recovered $(\hat{v}_n^x, \hat{v}_n^y, \hat{v}_n^z)$ velocity in the world frame of the sequence whose trajectory is shown in Fig. 6a. The time index is from 50s to 100s, where two turns are contained.

VII. ACKNOWLEDGEMENT

This project is partially supported by Hong Kong ITC ITSP Tier 2 grant #ITS/105/18FP: An intelligent Robotics System for Autonomous Airport Passenger Trolley Deployment awarded to Max Q.-H. Meng.

REFERENCES

- [1] C. Langlois, S. Tiku, and S. Pasricha, "Indoor localization with smartphones: Harnessing the sensor suite in your pocket," *IEEE Consumer Electronics Magazine*, vol. 6, no. 4, pp. 70–80, 2017.
- [2] K. W. Cheung and H.-C. So, "A multidimensional scaling framework for mobile location using time-of-arrival measurements," *IEEE transactions on signal processing*, vol. 53, no. 2, pp. 460–470, 2005.
- [3] Y. Shen and M. Z. Win, "Fundamental limits of wideband localization—part i: A general framework," *IEEE Transactions on Information Theory*, vol. 56, no. 10, pp. 4956–4980, 2010.
- [4] L. Zimmermann, A. Goetz, G. Fischer, and R. Weigel, "Gsm mobile phone localization using time difference of arrival and angle of arrival estimation," in *International Multi-Conference on Systems, Signals & Devices*. IEEE, 2012, pp. 1–7.
- [5] C. Yang and H.-R. Shao, "Wifi-based indoor positioning," *IEEE Communications Magazine*, vol. 53, no. 3, pp. 150–157, 2015.
- [6] B. Wang, Q. Chen, L. T. Yang, and H.-C. Chao, "Indoor smartphone localization via fingerprint crowdsourcing: Challenges and approaches," *IEEE Wireless Communications*, vol. 23, no. 3, pp. 82–89, 2016.
- [7] Q. Liang, L. Wang, Y. Li, and M. Liu, "Indoor mapping and localization for pedestrians using opportunistic sensing with smartphones," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 1649–1656.
- [8] J. A. Hesch, D. G. Kottas, S. L. Bowman, and S. I. Roumeliotis, "Camera-imu-based localization: Observability analysis and consistency improvement," *The International Journal of Robotics Research*, vol. 33, no. 1, pp. 182–201, 2014.
- [9] T. Qin, P. Li, and S. Shen, "Vins-mono: A robust and versatile monocular visual-inertial state estimator," *IEEE Transactions on Robotics*, vol. 34, no. 4, pp. 1004–1020, 2018.
- [10] "Google. project Tango," [https://en.wikipedia.org/wiki/Tango_\(platform\)](https://en.wikipedia.org/wiki/Tango_(platform)).
- [11] D. Ormoneit, H. Sidenbladh, M. J. Black, and T. Hastie, "Learning and tracking cyclic human motion," in *Advances in Neural Information Processing Systems*, 2001, pp. 894–900.
- [12] X. Yun, E. R. Bachmann, H. Moore, and J. Calusdian, "Self-contained position tracking of human movement using small inertial/magnetic sensor modules," in *Proceedings 2007 IEEE International Conference on Robotics and Automation*. IEEE, 2007, pp. 2526–2533.
- [13] H. Yan, Q. Shan, and Y. Furukawa, "Ridi: Robust imu double integration," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 621–636.
- [14] H. Yan, S. Herath, and Y. Furukawa, "Ronin: Robust neural inertial navigation in the wild: Benchmark, evaluations, and new methods," *arXiv preprint arXiv:1905.12853*, 2019.
- [15] C. Chen, X. Lu, A. Markham, and N. Trigoni, "Ionet: Learning to cure the curse of drift in inertial odometry," in *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.
- [16] A. Barrau and S. Bonnabel, "The invariant extended kalman filter as a stable observer," *IEEE Transactions on Automatic Control*, vol. 62, no. 4, pp. 1797–1812, 2016.
- [17] M. Kok, J. D. Hol, and T. B. Schön, "Using inertial sensors for position and orientation estimation," *arXiv preprint arXiv:1704.06053*, 2017.
- [18] M. Brossard, A. Barrau, and S. Bonnabel, "Ai-imu dead-reckoning," *IEEE Transactions on Intelligent Vehicles*, 2020.
- [19] B. Siciliano, L. Sciavicco, L. Villani, and G. Oriolo, *Robotics: modelling, planning and control*. Springer Science & Business Media, 2010.
- [20] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural networks*, vol. 61, pp. 85–117, 2015.
- [21] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, et al., "Pytorch: An imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems*, 2019, pp. 8024–8035.
- [22] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers, "A benchmark for the evaluation of rgb-d slam systems," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2012, pp. 573–580.