Deep Neural Network Based Inertial Odometry Using Low-cost Inertial Measurement Units

Changhao Chen, Chris Xiaoxuan Lu, Johan Wahlström, Andrew Markham and Niki Trigoni
Department of Computer Science, University of Oxford
firstname.lastname@cs.ox.ac.uk

Abstract—Inertial measurement units (IMUs) have emerged as an essential component in many of today's indoor navigation solutions due to their low cost and ease of use. However, despite many attempts for reducing the error growth of navigation systems based on commercial-grade inertial sensors, there is still no satisfactory solution that produces navigation estimates with long-time stability in widely differing conditions. This paper proposes to break the cycle of continuous integration used in traditional inertial algorithms, formulate it as an optimization problem, and explore the use of deep recurrent neural networks for estimating the displacement of a user over a specified time window. By training the deep neural network using inertial measurements and ground truth displacement data, it is possible to learn both motion characteristics and systematic error drift. As opposed to established context-aided inertial solutions, the proposed method is not dependent on either fixed sensor positions or periodic motion patterns. It can reconstruct accurate trajectories directly from raw inertial measurements, and predict the corresponding uncertainty to show model confidence. Extensive experimental evaluations demonstrate that the neural network produces position estimates with high accuracy for several different attachments, users, sensors, and motion types. As a particular demonstration of its flexibility, our deep inertial solutions can estimate trajectories for non-periodic motion, such as the shopping trolley tracking. Further more, it works in highly dynamic conditions, such as running, remaining extremely challenging for current techniques.

Index Terms—Pedestrian Navigation, Inertial Indoor Localization, Deep Neural Network, Learning from Mobile Sensor Data, Inertial Measurement Units.

1 Introduction

Fast and accurate robust localization is a fundamental requirement for human and mobile agents. Although global navigation satellite system (GNSS) is an adequate solution to most outdoor positioning problems, satellite signals are blocked or suffer from serious attenuation and multi-path effects in and around buildings, and therefore cannot be used for indoor positioning [1]. There is an emerging need to provide ubiquitous location information indoors for applications such as health/activity monitoring, smart retail, public places navigation, human-robot interaction, and augmented/virtual reality. One of the most promising approaches is to perform dead reckoning using measurements from inertial sensors. These methods have attracted great attention from both academia and industry [2], thanks to their mobility and flexibility. Unlike other commonly used sensor modalities, such as GNSS, radio or vision, inertial measurements are entirely egocentric and independent of both pre-deployed infrastructure as well as factors such as line-of-sight and visibility.

Recent advances within micro-electro-mechanical systems (MEMS) sensors have enabled inertial measurement units (IMUs) small and cheap enough to be deployed on smartphones, robots and drones. The physical models of inertial navigation system are based on Newtonian mechanisms, which require the navigation solution to triply integrate the inertial measurements with initial states into the orientation, velocity, and location. However, low-cost inertial sensors are plagued by high sensor noise that propagate to the navigation solution, leading to rapid error growth during stand-alone dead-reckoning.

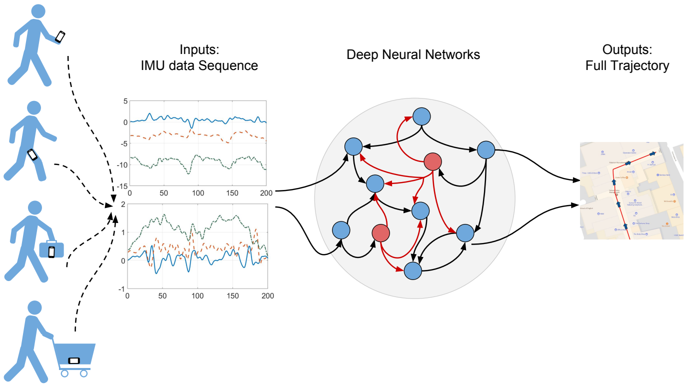


Fig. 1: Overview of our proposed learning-based method: the inertial measurements from mobile devices are directly feed into deep neural networks to predict trajectory.

To address the unbounded error drift problem, domain-specific knowledge is incorporated to enhance the accuracy of the inertial navigation system [3] in the context of pedestrian navigation. One solution is to perform a so-called zero-velocity update (ZUPT) whenever the foot is detected to be at standstill [4]. Essentially, this means that the navigation system uses the fact that the foot will be stationary at many sampling instances to mitigate the error growth. Unfortunately, this approach relies on the assumption that the IMU is attached to the foot and the pedestrian user adheres to a standard periodic walking motion. Another solution is step-based pedestrian dead reckoning (PDR), which infers positional and rotational displacement quantities by detecting steps, estimating step length and direction, and then updating the position estimates

- *Corresponding Author: Chris Xiaoxuan Lu, xiaoxuan.lu@cs.ox.ac.uk* accordingly [5]. Generally, the models used for estimating step length and step direction can be said to contain implicit motion models. Consequently, a PDR system sometimes has slower error growth than a standard inertial navigation system based on threefold integration of inertial measurements. However, the performance of the dynamic step estimation may be degraded by sensor noise, variations in the user's walking habits, and changes in the phone attachment [6]. Moreover, in many navigation situations of interest, no steps can be detected. For example, if a phone is placed on a baby stroller or a shopping trolley, the assumption of periodicity, exploited by step-based PDR, breaks down. The architecture of the two existing methods is illustrated in Figure 2. In summary, both ZUPT-aided inertial navigation and step-based PDR are limited by assumptions on motion dynamics and sensor attachment that prevent widespread adoption in daily life [1].

The emerging deep neural networks have proved their impressive performance in solving machine learning tasks, mainly in the fields of images, audio and speech processing [7]. When applied in tracking and localization, recent deep approaches [8], [9] demonstrate that deep neural networks (DNNs) are capable of extracting high-level motion representations from raw data, while providing the state-of-art results compared with traditional model based techniques in terms of accuracy and robustness. But little prior research has exploited the raw sequential measurements from low-cost noisy inertial sensors to learn deep tracking.

To constrain the unavoidable inertial system drift, we present a general framework - **I**nertial **O**dometry Neural **Net**work (IONet) that reconstructs accurate and robust trajectories from raw inertial measurements. Instead of directly integrating inertial data into system states, we propose to break the cycle of continuous error propagation, and reformulate inertial tracking as a sequential learning problem. This work primarily considers the problem of indoor localization, i.e. tracking objects and people in a planar environment using low-cost inertial sensors only. It relies on a common observation that there is normally no long-term change in height for indoor users. This assumption can be relaxed through the use of additional sensors such as a barometer for floor-change detection. Our proposed model is able to predict motion transformation and provide a 2D trajectory for indoor users from raw data without the need of any hand-engineering, as shown in Figure 1. Our contributions are as follows

- We cast the inertial tracking problem as a sequential learning problem.
- We propose the first deep neural network (DNN) framework that learns location transforms in polar coordinates from raw IMU data, and constructs 2D inertial tracking regardless of IMU attachment.
- Our framework is capable of estimating uncertainties along with the pose prediction, providing a metric for the model prediction confidence.
- We collected a large dataset for training and testing, and conducted extensive experiments across different attachments, motion modes, users/devices and new environment, whose results outperform traditional SINS and PDR mechanisms.
- We demonstrate that our model can generalize to a more general motion without regular periodicity, e.g. trolley or other wheeled configurations, and work in highly dynamic motion patterns, e.g. running and mixed velocity motion.

This paper extends the work presented in [10], with a new contribution on estimating uncertainties for deep inertial navigation framework, more details on proposed neural network framework, a deeper analysis of model performance and significantly more evaluations of new motion modes (running and walking slowly).

The paper is organized as follows: Section 2 provides an overview of related work; Section 3.1 formulates the principals and problems of classic inertial navigation systems; Section 3.2 presents a sequence-based physical model to reformulate the inertial tracking as a learning approach; Section 4 and 5 proposes the deep neural networks framework that reconstructs trajectory from raw inertial data and estimates uncertainties; Section 6 assesses the performance of our proposed model via extensive experiments; Section 7 draws conclusions and discusses future work.

2 Related Work

In this section, we provide a brief overview of some related work in inertial navigation systems, step-based pedestrian dead-reckoning, data-driven inertial motion tracking and deep learning for localization.

2.1 Inertial Navigation Systems

Strapdown inertial navigation systems (SINS) have been studied for decades [11]. Early inertial navigation systems relied on expensive, heavy, high-precision inertial measurement units. Hence, their application was constrained to moving vehicles, such as automobiles, ships, aircraft, submarines, and spacecraft. However, recent advances within MEMS technology has enabled the production of IMUs with significantly lower cost, size, and energy consumption. As a result, MEMS IMUs are today deployed within robotics, unmanned aerial vehicle navigation [12], and mobile positioning [2]. However, the accuracy of a MEMS IMU is very limited, and the sensor measurements typically have to be integrated with other information sources when used over an extended period, for example, as visual inertial odometry (VIO). The effective fusion of inertial sensors and cameras can be achieved via extended kalman filtering (EKF) [13], or graph optimization [14] [15] [16], where the raw visual and inertial measurements are tightly and jointly optimized. For pedestrian tracking, attaching an IMU on the user's foot can take advantage of Zero-Velocity Updates (ZUPTs) to compensate for error drifts of inertial systems [17]. ZUPTs make it possible to break the cubic error growth of stand-alone SINS. Since the foot is stationary at regular intervals during normal gait, the use of ZUPTs will typically lead to a substantial reduction in the position error growth [18]. However, the assumption of zero-velocity detection requires the IMUs to be attached on the user's foot, which makes this approach unsuitable for everyday usage.

2.2 Step-based Pedestrian Dead-reckoning

Unlike the open-loop integration of inertial sensors within SINS, PDR uses inertial measurements to detect steps, as well as to estimate stride length and heading via empirical formulas [19]. However, system errors still quickly accumulate, because of incorrect step segmentation and inaccurate stride estimation. In addition, a large number of parameters have to be carefully tuned according to the user's gait characteristics. There are three primary tasks of step-based pedestrian dead-reckoning: step count or step segmentation, step length estimation, and step direction estimation [20]. Step segmentation is normally based on accelerometer measurements and can be performed using thresholding or peak detection, correlation-based algorithms, or spectral analysis. A performance evaluation of algorithms for step segmentation is presented in [6]. The best performance was achieved with windowed peak detection, a hidden Markov model, and a continuous wavelet transform, which had median error rates in the order of 1.3 %. Given its relative simplicity, the authors in [6] recommend the windowed peak detection algorithm.

Recent research has mainly focused on fusing SINS and PDR with external references, such as a map information [21], WiFi fingerprinting [22], photodiode sensors [23], geomagnetic field distortion [24], and power network electromagnetic field [25]. However, the fundamental problem associated with the rapid error growth of pedestrian navigation systems only based on inertial measurements still remains unsolved. In this paper, we abandon previous approaches and present a new general framework for inertial odometry. This allows us to handle more general tracking problems, including trolley/wheeled configurations, which step-based PDR cannot address.

2.3 Data-driven Inertial Motion Tracking

Data-driven methods have been explored in a variety of inertial tracking tasks. Ahuja et al. adopted the supervised support vector regression to enhance the knee angle estimation during human walking with body-attached IMUs [26]. Inertial-based gesture recognition has been achieved by extracting handcrafted features via probabilistic parameter learning [27]. [28] proposed to estimate human walking speed by a Hidden Markov Model (HMM). Machine learning techniques were also explored in gait and pose analysis with inertial sensors [29], [30], [31]. [32] used multi-layer perceptrons (MLPs) to model sensor-displacement for human motion reconstruction. These approaches mostly learn to analyse human motion rather than human localization. They either rely on hand-designed features or enhance existing models with learnt parameter. Compared with previous work, our proposed model is able to automatically learn motion representation from raw data in an end-to-end manner and reconstruct an accurate trajectory for large-scale long-term indoor localization using deep neural networks.

2.4 Deep Neural Networks for Localization

Deep learning approaches have recently shown excellent performance in handling sequential data, such as speech recognition [33], machine translation [34], visual tracking and video description [35]. Previous learning-based work has tackled localization problems, by predicting ego-motion from visual sensor measurements using deep neural networks instead of applying geometric theory. Deep learning methods are capable of extracting high-level feature representation from large datasets, and provide an alternative to solve the visual odometry (VO) problem. [36] formulated VO as a classification problem, and proposed a Convolutional Neural Network (CNN) architecture to predict the discrete changes of direction and velocity. PoseNet [8] tackled the 6-DOF camera relocalization from a single RGB image with a CNN in an end-to-end manner. Further, [37] presented a CNN-based unsupervised framework to learn depth and camera ego-motion from video sequences by adopting the synthesis as the supervisory signal, showing competitive performance over the classic state-of-art VO. Here, the CNNs serve as a mapping from the raw scene image to the pose or the pose transformation. Only relying on CNNs, their models are easily overfit in scene geometry, limiting their generalization ability in new environments [38]. In contrast, DeepVO [9] and VidLoc [39] leveraged the combination of CNNs and deep recurrent neural networks (RNNs) for sequential learning to capture the temporary dependencies and motion dynamics of image sequences, rather than processing a single image. In these frameworks, the features extracted by CNNs are passed through RNNs for sequential learning, and the CNN and RNN modules are optimized jointly. They achieve excellent performance in pose estimation even in new scenarios. Similarly, VINet [40] processes both the image sequences and inertial measurements to realize visual inertial odometry based on RCNNs. Inspired by their work, we also exploit the ability of RNNs to model motion dynamics and temporal dependencies of inertial readings.

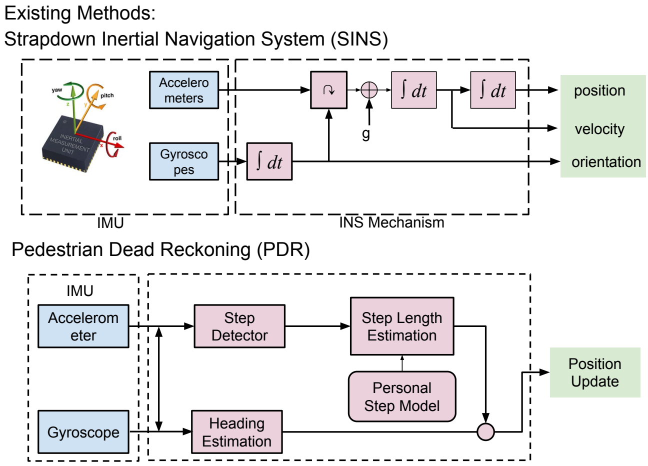


Fig. 2: Architecture of existing methods: SINS and PDR

To the best of our knowledge, IONet is the first neural network framework to achieve inertial odometry using inertial data only.

3 Background

This section introduces the background of inertial navigation mechanisms, and a derivation of sequence based physical model. We discuss the limitations of this model-based method, and show how the sequence based formulation paves the way to our proposed learning based approach.

3.1 The Principles Of Inertial Navigation

The principles of inertial navigation are based on Newtonian mechanics. They allow tracking the position and orientation of an object in a navigation frame given an initial pose and measurements from accelerometers and gyroscopes.

Fig. 2 illustrates the basic mechanism of inertial navigation algorithms. The three-axis gyroscope measures angular velocities of the body frame with respect to the navigation frame, which are integrated into pose attitudes in Equations (1-3). To represent the orientation, the direction cosine $\mathbf{C}_b^n$ matrix is used to represent the transformation from the body (b) frame to the navigation (n) frame, and is updated with a relative rotation matrix $\mathbf{\Omega}(t)$. The 3-axis accelerometer measures proper acceleration vectors in the body frame, which are first transformed to the navigation frame and then integrated into velocity, discarding the contribution of the gravity force $\mathbf{g}$ in Equation (4). The locations are updated by integrating velocity in Equation (5). Equations (1-5) describe the attitude, velocity and location updates at any time stamp t. In our application scenarios, the effects of earth rotation and the Coriolis accelerations are ignored.

The Attitude Update is given by

$$\mathbf{C}_b^n(t) = \mathbf{C}_b^n(t-1) * \mathbf{\Omega}(t) \tag{1}$$

$$\boldsymbol{r} = \boldsymbol{\omega}(t)dt \tag{2}$$

$$\mathbf{\Omega}(t) = \mathbf{C}_{b_t}^{b_{t-1}} = \boldsymbol{I} + \frac{\sin(r)}{r}[\boldsymbol{r}\times] + \frac{1-\cos(r)}{r^2}[\boldsymbol{r}\times]^2, \tag{3}$$

the Velocity Update is given by

$$\mathbf{v}(t) = \mathbf{v}(t-1) + ((\mathbf{C}_b^n(t-1)) * \boldsymbol{a}(t) - \mathbf{g}_n)dt, \tag{4}$$

and the Location Update is given by

$$\mathbf{L}(t) = \mathbf{L}(t-1) + \mathbf{v}(t-1)dt, \tag{5}$$

where $\boldsymbol{a}$ and $\boldsymbol{\omega}$ are accelerations and angular velocities in body frame measured by IMU, $\mathbf{v}$ and $\mathbf{L}$ are velocities and locations in navigation frame, r is the norm of $\boldsymbol{r}$, and $\mathbf{g}$ is gravity.

Under ideal condition, SINS sensors and algorithms can estimate system states for all future times. High-precision INS in military applications (aviation and marine/submarine) uses highly accurate and costly sensors to keep measurement errors very small. They also require a time-consuming system initialization including sensor calibration and orientation initialization. However, these requirements are inappropriate for pedestrian tracking. Realizing a SINS mechanism on low-cost MEMS IMU platform suffer from the following two problems

- The measurements from IMUs embedded in consumer phones are corrupted with various error sources, such as scale factor, axis misalignment, thermo-mechanical white noise and random walking noise [41]. From attitude update to location update, the INS algorithm sees a triple integration from raw data to locations. Even a tiny noise will be highly exaggerated through this open-loop integration, causing the whole system to collapse within seconds.
- A time-consuming initialization process is not suitable for everyday usage, especially for orientation initialization. Even small orientation errors lead to the incorrect projection of the gravity vector. For example, a 1 degree attitude error will cause an additional 0.1712 m/s^2 acceleration on the horizontal plane, leading to 1.7 m/s velocity error and 8.56 m location error within 10 seconds.

3.2 Sequence-based Physical Model

To address the problems of error propagation, our insight is to break the cycle of continuous integration, and segment inertial data into independent windows. This is analogous to resetting an integrator to prevent windup in classical control theory [42].

However, windowed inertial data is not independent, as Equations (1-5) clearly demonstrate. This is because key states (namely attitude, velocity and location) are *hidden* - they have to be derived from previous system states and inertial measurements, and propagated across time. Unfortunately, errors are also propagated across time, cursing inertial odometry. It is clearly impossible for windows to be truly independent. However, we can aim for pseudo-independence, where we estimate the *change* in navigation state over each window. Our problem then becomes how to constrain or estimate these latent system states over a window. Following this idea, we derive a sequence-based physical model from basic Newtonian Laws of Motion, and reformulate it into a learning model.

The unobservable or latent system states of an inertial system consist of orientation $\mathbf{C}_b^n$, velocity $\mathbf{v}$ and position $\mathbf{L}$. In a traditional model, the transformation of system states could be expressed as a transfer function/state space model between two time frames in Equation (6), and the system states are directly coupled with each other

$$[\mathbf{C}_b^n \quad \mathbf{v} \quad \mathbf{L}]_t = f([\mathbf{C}_b^n \quad \mathbf{v} \quad \mathbf{L}]_{t-1}, [\boldsymbol{a} \quad \boldsymbol{\omega}]_t). \tag{6}$$

We first consider displacement. To separate the displacement of a window from the prior window, we compute the change in displacement $\Delta\mathbf{L}$ over an independent window of n time samples, which is simply

$$\Delta\mathbf{L} = \sum_{t=0}^{n} \mathbf{v}(t) \cdot dt. \tag{7}$$

We can separate this out into a contribution from the initial velocity state, and the accelerations in the navigation frame

$$\Delta\mathbf{L} = n\mathbf{v}(0)dt + [(n-1)\mathbf{s}_1 + (n-2)\mathbf{s}_2 + \cdots + \mathbf{s}_{n-1}]dt^2 \tag{8}$$

where

$$\mathbf{s}(t) = \mathbf{C}_b^n(t-1)\boldsymbol{a}(t) - \mathbf{g} \tag{9}$$

is the acceleration in the navigation frame, comprising a dynamic part and a constant part due to gravity.

Then, Equation (8) is further formulated as

$$\begin{aligned}\Delta\mathbf{L} = n\mathbf{v}(0)dt + [(n-1)\mathbf{C}_b^n(0) * \boldsymbol{a}_1 + (n-2)\mathbf{C}_b^n(0)\mathbf{\Omega}(1) \\ * \boldsymbol{a}_2 + \cdots + \mathbf{C}_b^n(0)\prod_{i=1}^{n-2}\mathbf{\Omega}(i) * \boldsymbol{a}_{n-1}]dt^2 \\ - \frac{n(n-1)}{2}\mathbf{g}dt^2\end{aligned} \tag{10}$$

and simplified to become

$$\Delta\mathbf{L} = n\mathbf{v}(0)dt + \mathbf{C}_b^n(0)\mathbf{T}dt^2 - \frac{n(n-1)}{2}\mathbf{g}dt^2 \tag{11}$$

where

$$\mathbf{T} = (n-1)\boldsymbol{a}_1 + (n-2)\mathbf{\Omega}(1)\boldsymbol{a}_2 + \cdots + \prod_{i=1}^{n-2}\mathbf{\Omega}(i)\boldsymbol{a}_{n-1}. \tag{12}$$

In our work, we consider the problem of indoor positioning i.e. tracking objects and people on a horizontal plane. This introduces a key observation: in the navigation frame, there is no long-term change in height[1]. The mean displacement in the z axis over a window is assumed to be zero and thus can be removed from the formulation. We can compute the absolute change in distance over a window as the L-2 norm i.e. $\Delta l = \|\Delta\mathbf{L}\|_2$, effectively decoupling the distance traveled from the orientation (e.g. heading angle) traveled, leading to

$$\begin{aligned}\Delta l &= \|n\mathbf{v}(0)dt + \mathbf{C}_b^n(0)\mathbf{T}dt^2 - \frac{n(n-1)}{2}\mathbf{g}dt^2\|_2 \\ &= \|\mathbf{C}_b^n(0)(n\mathbf{v}^b(0)dt + \mathbf{T}dt^2 - \frac{n(n-1)}{2}\mathbf{g}_0^b dt^2)\|_2.\end{aligned} \tag{13}$$

[1] This assumption can be relaxed through the use of additional sensor modalities such as a barometer to detect changes in floor level due to stairs or elevator.

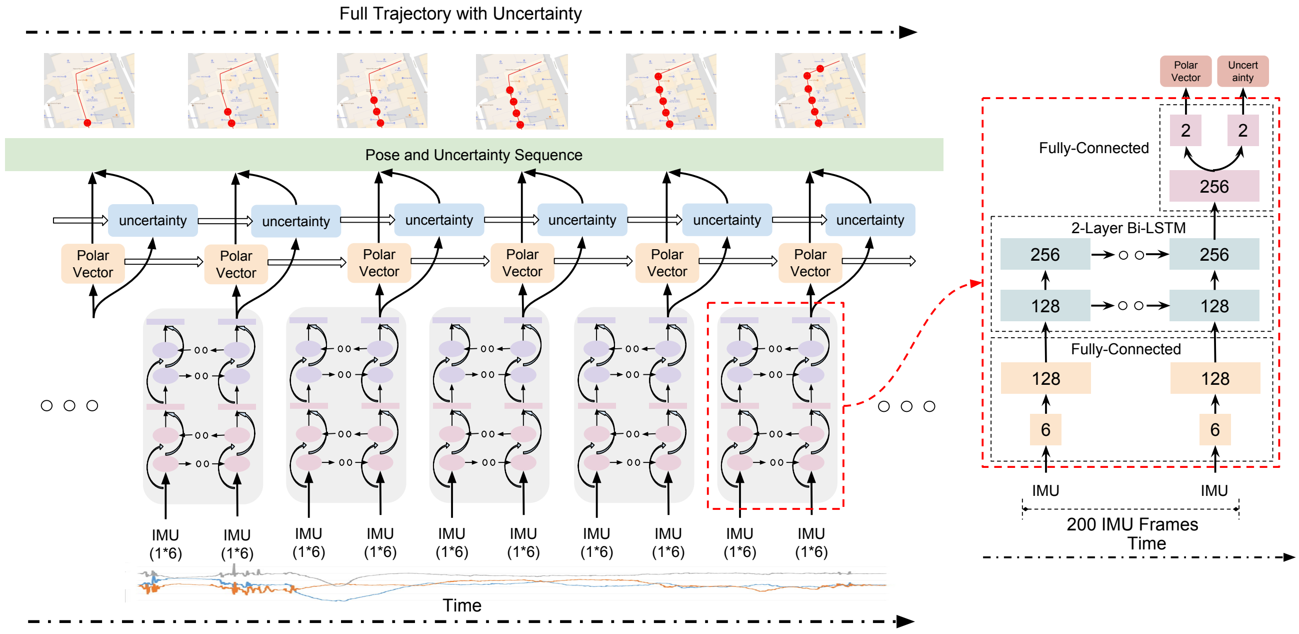


Fig. 3: Overview of IONet framework. Inertial measurements are segmented into independent windows. A 2-layer bi-LSTM is used to estimate the change in heading and displacement (polar vector) as well as the estimation uncertainties.

Because the rotation matrix $\mathbf{C}_b^n(0)$ is an orthogonal matrix i.e. $\mathbf{C}_b^n(0)^T\mathbf{C}_b^n(0) = \mathbf{I}$, the initial unknown orientation has been successfully removed from, giving us

$$\begin{aligned}\Delta l &= \|\Delta\mathbf{L}\|_2 \\ &= \|n\mathbf{v}^b(0)dt + \mathbf{T}dt^2 - \frac{n(n-1)}{2}\mathbf{g}_0^b dt^2\|_2.\end{aligned} \tag{14}$$

Hence, over a window, the horizontal distance traveled can be expressed as a function of the initial velocity, the gravity, and the linear and angular acceleration, all in the body frame

$$\Delta l = f(\mathbf{v}^b(0), \mathbf{g}_0^b, \boldsymbol{a}_{1:n}, \boldsymbol{\omega}_{1:n}). \tag{15}$$

To determine the change in the user's heading, we consider that a user's proper accelerations and angular rates $(\boldsymbol{a}_{1:n}, \boldsymbol{\omega}_{1:n})$ are also latent variables of IMU raw measurements $(\hat{\boldsymbol{a}}_{1:n}, \hat{\boldsymbol{\omega}}_{1:n})$, and on the horizontal plane, only the heading attitude is essential in our system. From Equations (2-3) the change in the heading $\Delta\psi$ is expressed as a function of the raw data sequence. Therefore, we succeed in reformulating traditional model as a polar vector $(\Delta l, \Delta\psi)$ based model, which is only dependent on inertial sensor data, the initial velocity and gravity in the body frame

$$(\Delta l, \Delta\psi) = f_\theta(\mathbf{v}^b(0), \mathbf{g}_0^b, \hat{\boldsymbol{a}}_{1:n}, \hat{\boldsymbol{\omega}}_{1:n}). \tag{16}$$

To derive a global location, the starting location $\mathbf{L} = (L_0^x, L_0^y)$ and heading ψ_0 and the Cartesian projection of a number of windows can be written as

$$\begin{cases} L_n^x = L_0^x + \Delta l\cos(\psi_0 + \Delta\psi) \\ L_n^y = L_0^y + \Delta l\sin(\psi_0 + \Delta\psi). \end{cases} \tag{17}$$

Our task now becomes how to implicitly estimate this initial velocity and the gravity in body frame, by casting each window as an estimation problem.

This section serves as an overview showing the transition from the traditional model-based method to the proposed neural-network-based method. It takes the traditional state-space-model described in Equations (1-5), which converts raw data to poses in a step-by-step manner, to a formulation where a window of raw inertial data is processed in a batch to estimate a displacement and an angle change. Note that in both formulations, the final output depends on the initial attitude and velocity. As a result, in both cases, the curse of error accumulation will not be avoided if using the model-based integration approach. However, our sequence based formulation paves the way to our proposed neural network approach.

4 Inertial Odometry Neural Network

Estimating the initial velocity and orientation in the body frame explicitly using traditional techniques is an extremely challenging problem. Rather than trying to determine the two terms, we instead treat Equation (16) as an estimation problem, where the inputs are the observed sensor data and the output is the polar vector. The unobservable terms simply become latent states of the estimation. Intuitively, the motivation for this relies on the regular and constrained nature of pedestrian motion. Over a window, which could be a few seconds long, a person walking at a certain rate induces a roughly sinusoidal acceleration pattern. The frequency of this sinusoid relates to the walking speed. In addition, biomechanical measurements of human motion show that as people walk faster, their strides lengthen [43]. Similarly, vehicle speed can also be estimated using only raw IMU measurements. This was demonstrated in [44], which used an accelerometer to track the vibrations of the vehicle chassis. The idea relies on the fact that the vibrations have a fundamental frequency that is proportional to the vehicle speed. Moreover, the gravity in body frame is related to the initial yaw and roll angle, determined by the attachment/placement of the device, which can be estimated from the raw data [45]. In this paper, we propose the use of deep neural networks to learn the relationship between raw acceleration data and the polar delta vector, as illustrated in Fig. 3.

Input data are independent windows of consecutive IMU measurements, strongly temporal dependent, representing body motion. To recover latent connections between motion characteristics and data features, a deep recurrent neural network (RNN) is capable of exploiting these temporal dependencies by maintaining hidden states over the duration of a window. Note however that latent states are not propagated between windows. Effectively, the neural network acts as a function f_θ that maps sensor measurements to polar displacement over a window

$$(\boldsymbol{a}, \boldsymbol{\omega})_{200\times 6} \xrightarrow{f_\theta} (\Delta l, \Delta\psi)_{1\times 2}, \tag{18}$$

where a window length of 200 frames (2 seconds) is used here[2].

In the physical model, orientation transformations impact all subsequent outputs. We adopt a Long Short-Term Memory (LSTM) to handle the exploding and vanishing problems of vanilla RNN, as it has a much better ability to exploit the long-term dependencies [46]. In addition, as both previous and future frames are crucial in updating the current frame a bidirectional architecture is adopted to exploit dynamic context.

Equation (16) shows that modeling the final polar vector requires modeling some intermediate latent variables, e.g. initial velocity and gravity. Therefore, to build up higher representation of IMU data, it is reasonable to stack 2-layer LSTMs on top of each other, with the output sequences of the first layer supplying the input sequences of the second layer. The second LSTM outputs one polar vector to represent the transformation relation in the processed sequence. Each layer has 128 hidden nodes. To increase the output data rate of polar vectors and locations, IMU measurements are divided into independent windows of 200 frames (2s) with a stride of 10 frames (0.1s).

The optimal parameter θ^* inside the proposed deep RNN architecture can be recovered by minimizing a loss function on the training dataset $\mathbf{D} = (\mathbf{X}, \mathbf{Y})$ as

$$\theta^* = \arg\min_\theta \mathcal{L}(f_\theta(\mathbf{X}), \mathbf{Y}). \tag{19}$$

The loss function is defined as the sum of Euclidean distances between the ground truth $(\Delta l, \Delta\psi)$ and estimated value $(\tilde{\Delta l}, \tilde{\Delta\psi})$

$$\mathcal{L} = \sum \|\Delta l - \tilde{\Delta l}\|_2^2 + \kappa\|\Delta\psi - \tilde{\Delta\psi}\|_2^2 \tag{20}$$

where κ is a factor to regulate the weights of location displacement Δl and heading change $\Delta\psi$.

5 The Uncertainty Estimation of IONet

In this section, we aim to determine the uncertainty of deep inertial navigation, which represents the confidence in IONet model output. Since deep neural networks are hard to interpret [47], uncertainty estimation allows us to understand to what extent to trust the model prediction [48]. Moreover, quantifying uncertainty is essential in enhancing inertial navigation with sensor fusion and graph SLAM [49]. For example, the inertial sensor can be better integrated with GPS to form a GPS/IMU systems [50], or with a camera to form visual inertial odometry [14], with the aid of uncertainty.

In our work, the uncertainty is first estimated on motion transformation, e.g. polar vector transformation $(\Delta l, \Delta\psi)$ defined in Equation 16, and then we will show how to infer the uncertainty of absolute heading attitude and the location. Unlike the polar vector, which is trained using supervisory labels provided by high precision optical motion tracking system, the hand-crafted labels for uncertainty are impossible to obtain. This is because there is no direct measurement method for the real uncertainty. Therefore, we propose to estimate the uncertainty of inertial tracking in an unsupervised manner by extending the framework in Section 4 to a Bayesian model.

The uncertainty for the polar vector is assumed to be normally distribution. Our IONet architecture proposed in Section 4 is trained by minimizing the mean square loss between the predicted polar vector $f_\theta(\boldsymbol{x})$ and its corresponding labels $\boldsymbol{y}$. Their outputs can be regarded as the mean of conditional distribution: $\mathcal{N}(f_\theta(\boldsymbol{x}), \boldsymbol{\sigma}^2)$. The likelihood of predicting the real motion transformation is defined as a Gaussian distribution with the model prediction and its variance $\boldsymbol{\sigma}^2$

$$\begin{aligned} p(\boldsymbol{y}|f_\theta(\boldsymbol{x})) &= \mathcal{N}(f_\theta(\boldsymbol{x}), \boldsymbol{\sigma}^2) \\ &= \frac{1}{\sqrt{2\pi\boldsymbol{\sigma}^2}} \exp\Big(-\frac{(\boldsymbol{y} - f_\theta(\boldsymbol{x}))^2}{2\boldsymbol{\sigma}^2}\Big). \end{aligned} \tag{21}$$

We alter the deep neural network framework to predict both the motion transformation and its variance. The variance $\boldsymbol{\sigma}$ represents the probabilistic distribution over the model output, termed Aleatoric uncertainty [51]. The aim is to optimize the neural network weights θ by performing a MAP (maximum a posteriori probability) inference. This is equivalent to finding an optimal value θ^* for the model parameters

$$\begin{aligned} \theta^* &= \arg\max_\theta p(\boldsymbol{y}|f_\theta(\boldsymbol{x})) \\ &= \arg\min_\theta - \log p(\boldsymbol{y}|f_\theta(\boldsymbol{x})) \\ &= \arg\min_\theta \frac{1}{2\boldsymbol{\sigma}^2}\|\boldsymbol{y} - f_\theta(\boldsymbol{x})\|^2 + \frac{1}{2}\log\boldsymbol{\sigma}^2. \end{aligned} \tag{22}$$

The minimizing objective of loss is defined as

$$\mathcal{L} = \frac{1}{2}\boldsymbol{\sigma}^{-2}\|\boldsymbol{y} - \tilde{\boldsymbol{y}}\|^2 + \frac{1}{2}\log\boldsymbol{\sigma}^2 \tag{23}$$

where $\tilde{\boldsymbol{y}}$ and $\boldsymbol{\sigma}$ are the model predicted mean and variance. In practice, our framework is designed to predict the log variance $\boldsymbol{s}_i = \log\boldsymbol{\sigma}^2$, considering that regressing $\boldsymbol{s}_i$ is more stable than directly predicting $\boldsymbol{\sigma}^2$ [51]

$$\mathcal{L} = \frac{1}{2}\exp(-\boldsymbol{s}_i)\|\boldsymbol{y} - \tilde{\boldsymbol{y}}\|^2 + \frac{1}{2}\boldsymbol{s}_i. \tag{24}$$

Based on the propagation rules of uncertainty, the uncertainty of absolute heading attitude and the location can be inferred. As the absolute heading is the accumulation of the heading displacement, its variance $\boldsymbol{\sigma}^2_{\psi_t}$ at t time step is the sum of the previous variance $\boldsymbol{\sigma}^2_{\psi_{t-1}}$ and the variance of delta heading $\boldsymbol{\sigma}^2_{\Delta\psi_t}$, estimated by our deep neural network i.e.

$$\boldsymbol{\sigma}^2_{\psi_t} = \boldsymbol{\sigma}^2_{\psi_{t-1}} + \boldsymbol{\sigma}^2_{\Delta\psi_t}. \tag{25}$$

From Equation (17), both variances of the delta location and the absolute heading are considered in order to infer the variance of delta location in the navigation frame - East (x) and North (y) axes:

$$\begin{cases} \boldsymbol{\sigma}^2_{\Delta L^x} = \cos(\psi)^2 * \sigma^2_{\Delta l} + \Delta l^2[\sin(\psi) * \sigma_\psi]^2 \\ \boldsymbol{\sigma}^2_{\Delta L^y} = \sin(\psi)^2 * \sigma^2_{\Delta l} + \Delta l^2[\cos(\psi) * \sigma_\psi]^2 \end{cases}. \tag{26}$$

[2] We experimented with a window size of 50, 100, 200 and 400 frames, and selected 200 as an optimal parameter regarding the trade-off between accumulative location error and predicted loss.

Similar to the uncertainty of the absolute heading, the uncertainty of the absolute location is the accumulation of the variance of the location change

$$\boldsymbol{\sigma}^2_{L_t} = \boldsymbol{\sigma}^2_{L_{t-1}} + \boldsymbol{\sigma}^2_{\Delta L_t}. \tag{27}$$

6 Experiments

In this section, we evaluate our proposed model in terms of accuracy and robustness. A large dataset was collected to train and test our proposed model. We will first describe the data collection and the training of the neural network. We then evaluate the accuracy of IONet and test its ability to generalize across different users, devices, motion modes, and environments, followed by uncertainty estimation. Last, we further apply our model to trolley tracking.

6.1 Training Details

Dataset: There are no public datasets for indoor localization using phone-based IMU. We therefore developed our own dataset[3] with a pedestrian walking inside a room, where an optical motion capture system (Vicon) is installed. Vicon is known for providing high-precision full pose ground truth (0.01m for location, 0.1 degree for orientation) for object localization and tracking [52]. The training dataset is collected from the IMU sensor on an iPhone 7 Plus. The smartphone is attached at different positions with the same pedestrian, including hand-held, in pocket and in handbag and on trolley. We collect 2-hours of IMU data for each attachment scenario. Note that, this phase of training data collection only needs one pedestrian (User 1) to participate. The raw inertial readings are segmented into sequences with a window size of 200 frames (2 seconds). The detailed description of the dataset is given in Table 1.

In order to test our model's ability to generalize across different users, we invited 3 new participants and made further evaluations on two additional phones: an iPhone 6 and an iPhone 5.

Training: We implemented our model on the Pytorch platform, and train the model on a NVIDIA TITAN X GPU. During training, we used Adam, a first-order gradient-based optimizer [53] with a learning rate of 0.0001. On average, the training converges after 100 iterations. To avoid overfitting, we gathered data with abundant moving characteristics inside, and adopted Dropout [54] in each LSTM layer, randomly dropping 25% units from neural networks during training. This method significantly reduces overfitting, and proves to perform well on new users, devices and environments.

Testing: We also found that a separate training on every attachment shows better performance in prediction than training jointly, hence we implemented the prediction model of 2-layer Bi-LSTM trained on separate attachments in our following experiments. In a practical deployment, existing techniques can be adopted to recognize different attachments from pure IMU measurements [6], providing the ability to dynamically switch between trained models.

Baselines: Two traditional methods are selected as baselines, pedestrian dead reckoning (PDR) and strapdown inertial navigation system (SINS) mechanism [11], to compare with our prediction results. PDR algorithms are seldom made open-source,

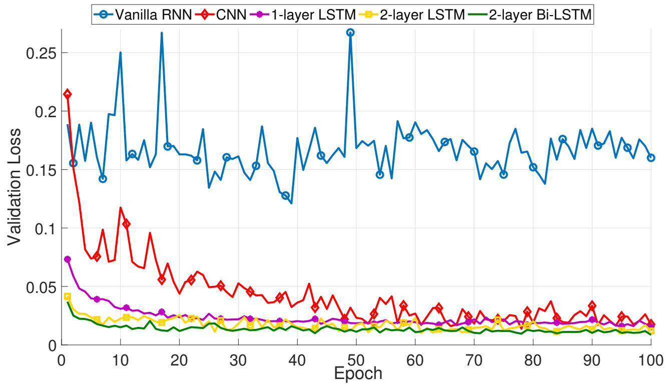
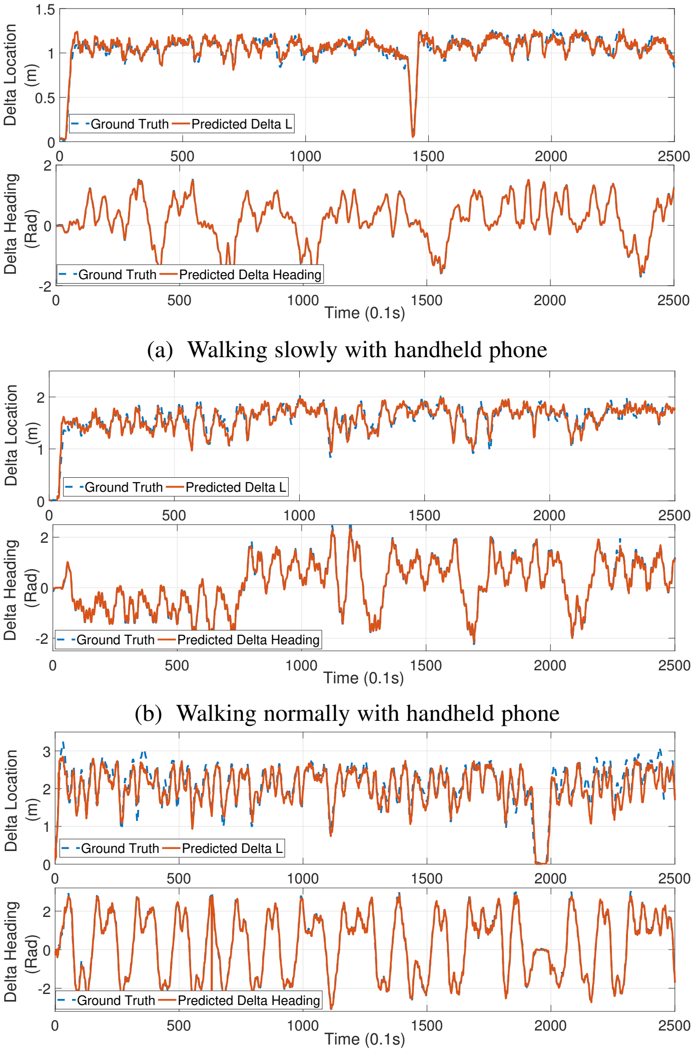


Fig. 4: The losses of adopting various frameworks demonstrate that our proposed IONet framework with 2-layer Bi-LSTM descends more steeply, and stays lower and more smoothly during the training than all other neural networks.



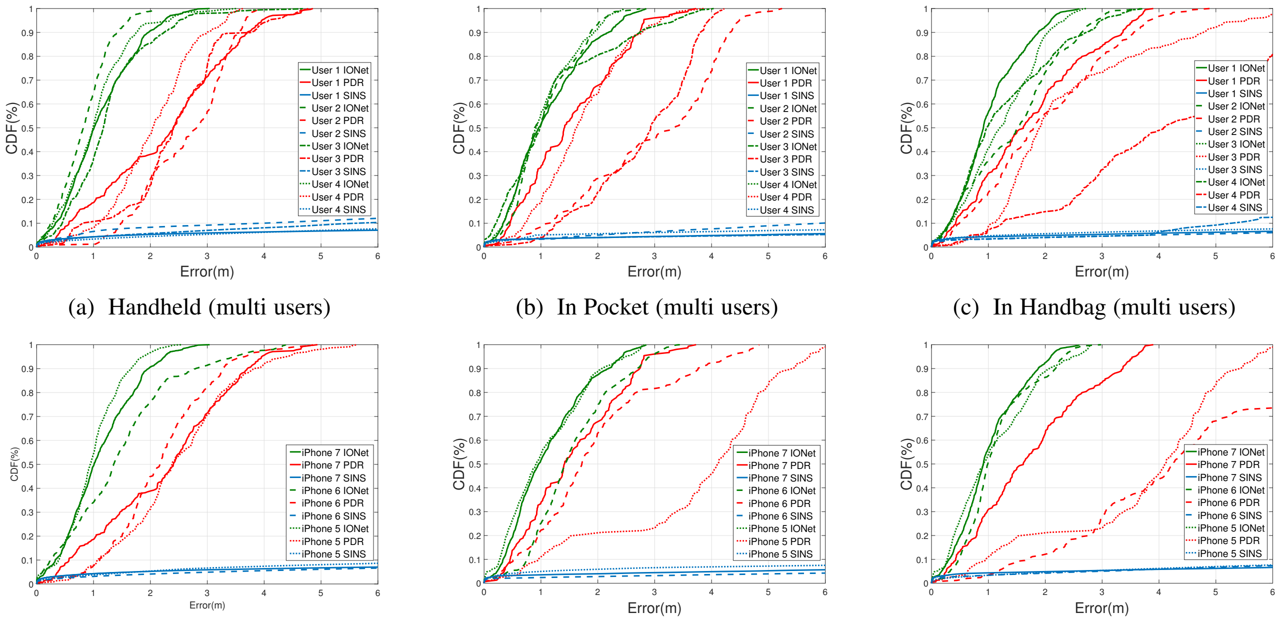
(a) Walking slowly with handheld phone

(b) Walking normally with handheld phone

(c) Running with handheld phone

Fig. 5: The predicted polar vector are very close to the ground truth with handheld phone no matter in (a) slowly walking (b) normally walking and (c) running, showing the effectiveness of our proposed framework on polar vector regression from raw inertial data.

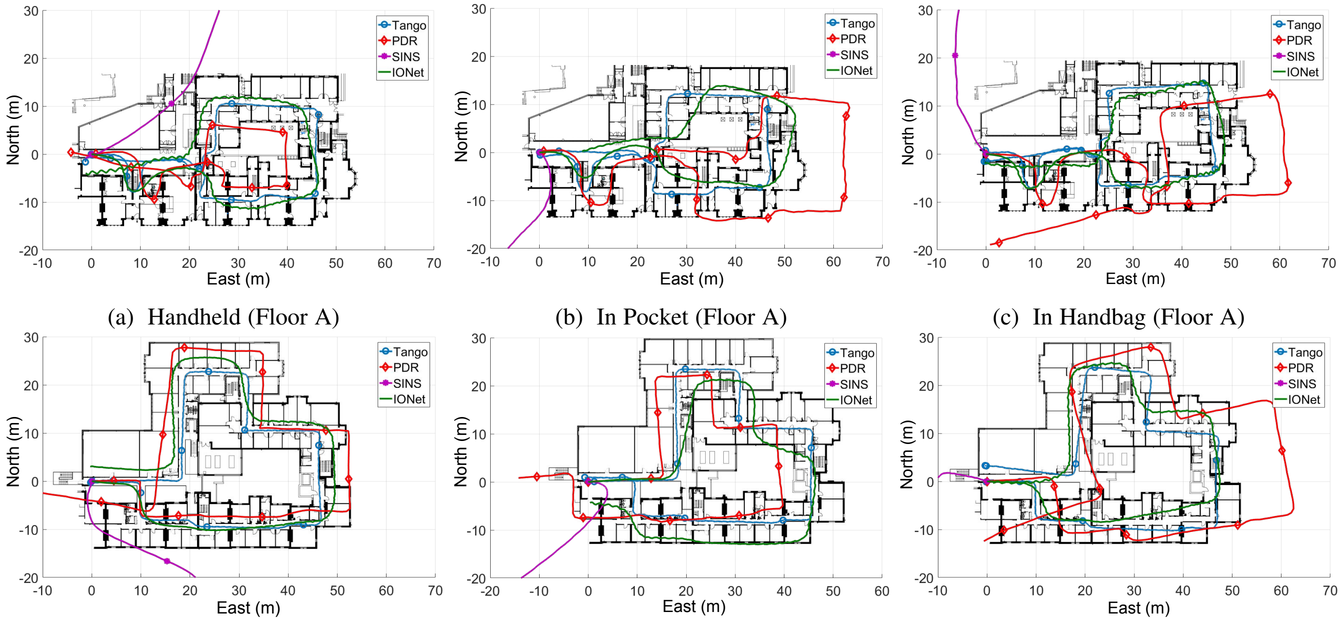
[3] Our Dataset can be found at http://deepio.cs.ox.ac.uk



(a) Handheld (multi users) (b) In Pocket (multi users) (c) In Handbag (multi users)

(d) Handheld (multi devices) (e) In Pocket (multi devices) (f) Handbag (multi devices)

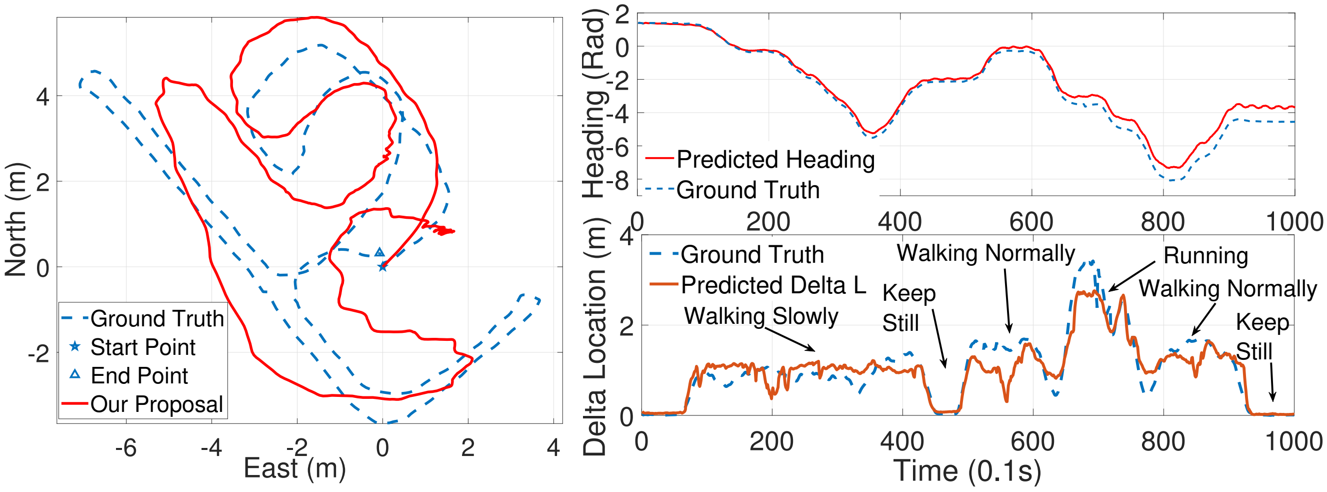
Fig. 6: The performance of our proposed IONet is compared with SINS and PDR. In the experiments involving multiple users, our learning model trained on User 1 is tested directly on other users without further fine-tuning in three attachments: handheld (a), pocket (b) and bag (c), to show the generalization ability across different devices. In the experiments involving multiple devices, our model trained on iPhone 7 is tested directly on other devices without further fine-tuning in three attachments: handheld (d), pocket (e) and bag (f) to show its generalization ability across different devices.



(a) Handheld (Floor A) (b) In Pocket (Floor A) (c) In Handbag (Floor A)

(d) Handheld (Floor B) (e) In Pocket (Floor B) (f) In Handbag (Floor B)

Fig. 7: Our proposed IONet can generate more accurate trajectories in large-scale localization experiments on two office floors - Floor A (1650 m^2) and Floor B (2475 m^2) in three attachments - Handheld, In Pocket, and In Handbag, compared with SINS and PDR. Note that our learning model is trained inside one room (Vicon Room), but generalize well to outside places without further training.



(a) Trajectories

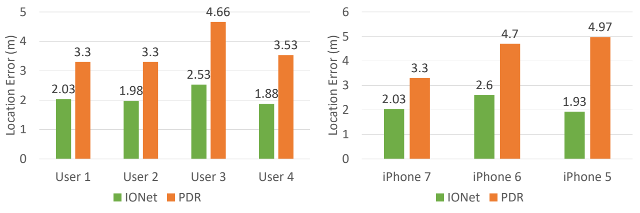
(b) Heading and Delta Location

Fig. 8: The performance of IONet is evaluated on an challenging experiment with varying motion modes including waking slowly, halting, walking normally and running for estimating (a) trajectory and (b) heading and location displacement. IONet can learn an extensive amount of information about the idiosyncratic motion behavior associated with different motion modes.

TABLE 1: Number of Sequences in Dataset

Dataset Domain	Training Sequences	Testing Sequences
Handheld (Normal)	45544	3812
Handheld (Slow)	36242	3095
Handheld (Run)	31161	3007
Pocket	53631	2385
Handbag	36410	4430
Trolley	29001	2718

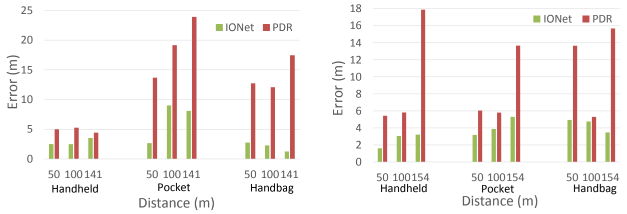
especially a robust PDR used in different attachments, so we implement code ourselves according to [6] for step detection and [55] for heading and step length estimation.



(a) Multiple Users

(b) Multiple Phones

Fig. 9: The maximum position error of IONet stayed around 2 meters within 90% of the testing time, seeing 30%- 40% improvement compared with traditional PDR in multiple users (a) and multiple phones (b).



(a) Floor A

(b) Floor B

Fig. 10: In large-scale indoor localization, absolute position errors of IONet is calculated at a distance of 50m, 100m and the end point on Floor A (a) and Floor B (b), showing competitive performance over traditional PDR.

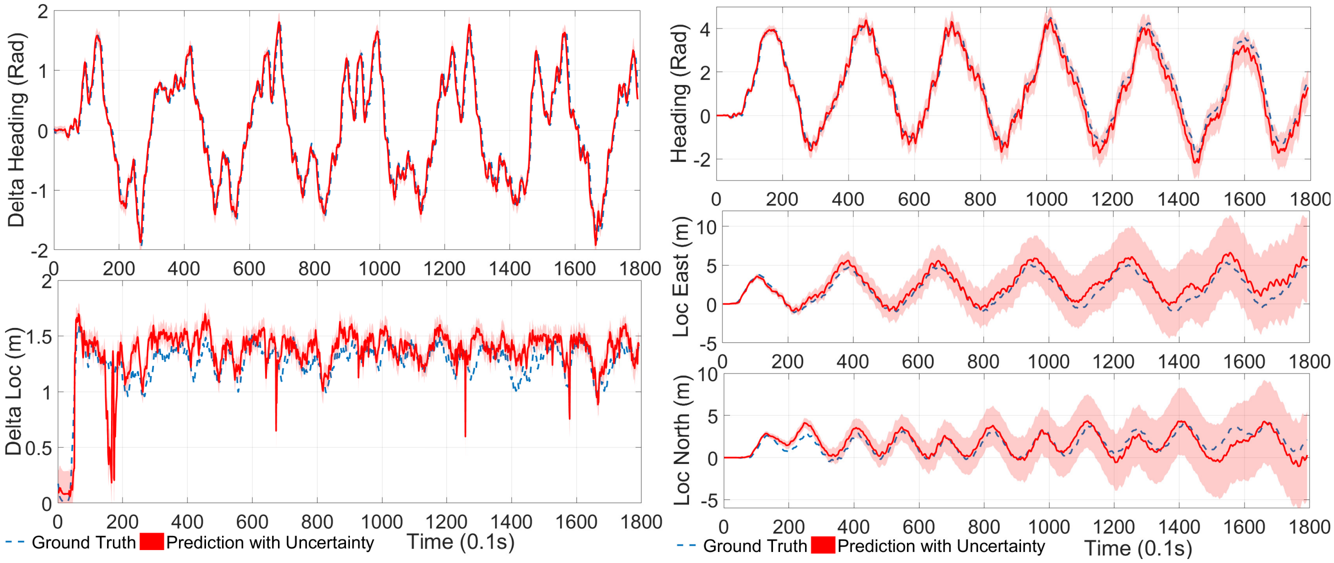
6.2 Comparison with Other DNN Frameworks

To evaluate our assumption of adopting a 2-layer Bidirectional LSTM for polar vector regression, we compare its validation results with various other DNN frameworks, including frameworks using vanilla RNN, vanilla Convolution Neural Network, 1-layer LSTM and 2-layer LSTM without Bi-direction. The training data are from all attachments. Fig. 4 shows their validation loss lines. The dimension of hidden states is chosen the same for all recurrent neural networks architectures (vanilla RNN, 1-layer LSTM, 2-layer LSTM, and 2-layer Bi-directional LSTM). Our proposed framework with 2-layer Bi-LSTM descends more steeply, and stays lower and more smoothly during the training than all other neural networks, supporting our assumption, while vanilla RNN suffers from vanishing gradient problems, and CNN doesn't seem to capture the temporal dependencies well.

6.3 Tests Involving Multiple Users and Devices

A series of experiments were conducted inside a large room with new users and phones to show our neural network's ability to generalize. The Vicon system provides a highly accurate reference to measure the location errors.

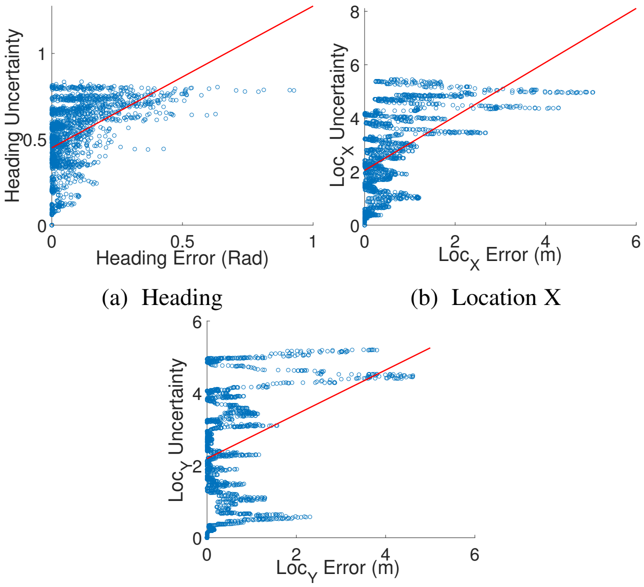
The first group of tests include four participants, walking randomly for two minutes with the phone in different attachments, e.g. in hand, pocket and handbag respectively, covering everyday behaviors. Note that the training dataset is taken from only one of these participants. The performance of our model is measured as cumulative error distribution function (CDF) against Vicon ground truth and compared with conventional PDR and SINS. Fig. 6a, Fig. 6b and Fig. 6c illustrate that our proposed approach outperforms the competing methods in every attachment. If raw data is directly triply integrated by SINS, this results in cubic error growth. The



(a) The uncertainty of the polar vector.

(b) The uncertainty of heading and location

Fig. 11: The predicted mean values of IONet are shown together with their corresponding uncertainty and the ground truth for (a) polar vector and (b) absolute heading and locations.



(a) Heading

(b) Location X

(c) Location Y

Fig. 12: The correlation between uncertainties and errors for global heading (a) and locations (b, c) demonstrates that the uncertainties are in positive correlation with the errors. Note that the small error does not always mean small uncertainty, because the mean errors can cancel with each other.

maximum position error of IONet stayed around 2 meters within 90% of the testing time, seeing 30%- 40% improvement compared with traditional PDR in Fig. 9a.

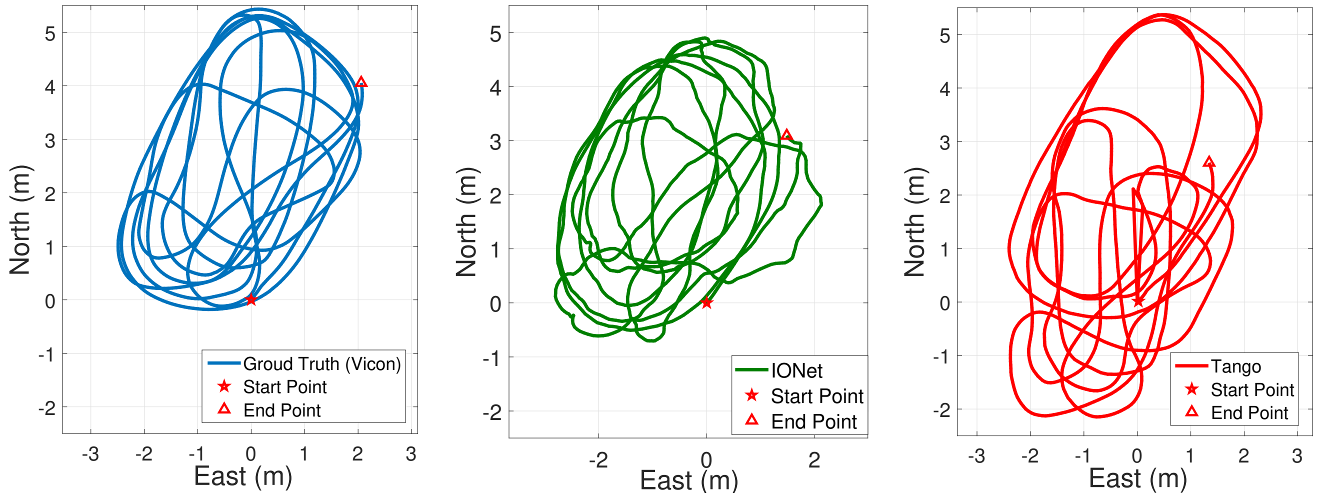
Another group of experiments is to test the performance across different devices, shown in Fig. Fig. 6d, Fig. 6e and Fig. 6f. We choose another two very common consumer phones, iPhone 6 and iPhone 5, whose IMU sensors, InvenSense MP67B and ST L3G4200DH, are quite distinct from our training device, iPhone 7 (IMU: InvenSense ICM-20600). Although intrinsic properties of different sensors influence the quality of inertial measurements, our neural network shows good robustness.

6.4 Tests Involving Multiple Motion Modes

To evaluate the ability of IONet to generalize across different motion modes, several experiments were conducted. Fig. 5 displays estimated and ground truth polar vectors when training and testing individually with three different motion types: slow walk, normal walk, and running. As can be seen from the figures, both relative distance and heading can be estimated with great accuracy over an extended period for all motion modes. This demonstrates that the neural network can achieve excellent performance over a broad range of motion types. By contrast, Fig. 8 illustrates the performance on a more varied data set where all of the three previously studied motion modes are used, and where the training data included all of the three motion modes. As seen in Fig. 8 (b), the heading and distance estimates are of good quality through most of the data set. However, the overall error is higher than when considering the different motion modes separately as in Fig. 5. Hence, taken together, Fig. 8 and Fig. 5 make it possible to conclude that the neural network can learn an extensive amount of information about the idiosyncratic motion behavior associated with different motion modes. For practitioners, this means that motion type classification and individual calibration for different motion types potentially can lead to significant improvements in performance. Despite the fact that quickly varying movement characteristics may be more challenging for IONet, Fig. 8 (a) still demonstrates good ability to reconstruct the overall motion pattern of a trajectory with mixed motion modes.

6.5 Large-scale Indoor Localization

Here, we apply our model on a more challenging indoor localization experiment to present its performance in a new environment. Our model *without training outside* Vicon room, is directly applied to six large-scale experiments conducted on two floors of an office building. The new scenarios contained long straight lines and



(a) Ground Truth (b) IONet (c) Tango

Fig. 13: The trolley tracking trajectories of our proposed (b) IONet are compared with (a) Ground Truth, and (c) Tango. Our proposed IONet shows almost the same accuracy as Tango, and even better robustness, because our pure inertial solution suffers less from environmental factors.

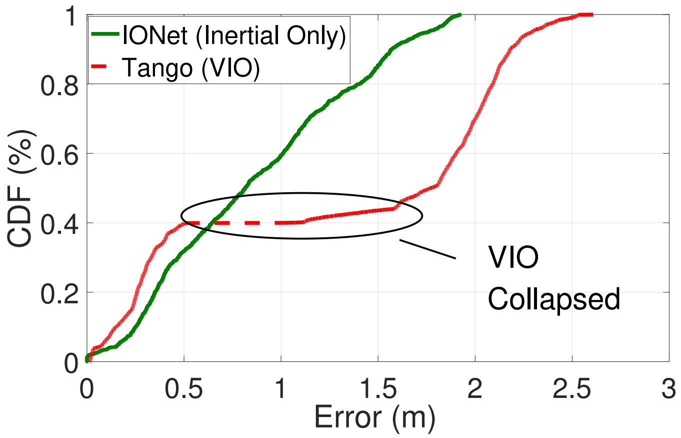


Fig. 14: The error Cumulative Distribution Function (CDF) of absolute locations predicted by IONet is compared with Tango (visual inertial odometry) in trolley tracking.

slopes, which were not contained in the training dataset. Lacking the high precision reference from Vicon, we take Google Tango Tablet [56], a well-known visual-inertial device, as pseudo ground truth.

The floor maps are illustrated in Fig. 7a (about 1650 m^2 size) and Fig. 7d (about 2475 m^2). Participants walked normally along corridors with the phone in three attachments respectively. The predicted trajectories from our proposal are closer to Tango trajectories, compared with the two other approaches in Fig. 7. The continuously propagating error of SINS mechanism caused trajectories with cubic error growth. Impacted by wrong step detection or inaccurate step stride and heading estimation, PDR accuracy is limited. We calculate absolute position error against pseudo ground truth from Tango at a distance of 50m, 100m and the end point in Fig. 10. Our IONet shows competitive performance over traditional PDR and has the advantage of generating a continuous trajectory at 10 Hz, although its heading attitude deviates from true values occasionally.

To summarize the generalizability of IONet, Section 6.1 stated that IONet gives significantly better performance when applied to different attachments separately. On the other hand, Sections 6.3 and 6.5 demonstrated that the neural network generalizes well across different users, devices, and test environments. As discussed in Section 6.4, IONet can perform well when simultaneously trained and evaluated on multiple motion modes, but may give better performance if the network is trained individually for each motion type.

6.6 Uncertainty Estimation

We tested the Bayesian model proposed in Section 5 to evaluate its ability for uncertainty estimation. The handheld attachment in normal walking was taken as an example. The model was trained on the data collected by iPhone 7Plus, but tested on an Android smartphone Nexus 5, whose IMU has distinct different properties from the iPhone 7Plus, in order to show the model confidence in a different input distribution.

The predicted polar vector (delta location and delta heading) is shown together with its corresponding uncertainty (standard deviation $\boldsymbol{\sigma}$) and the ground truth in Fig. 11a. From the variance of motion transformation, the uncertainties of absolute heading and locations are inferred according to Equations 25 - 27, and their results are presented in Fig. 11b. For heading estimation, the uncertainty captures to what extent the predicted values deviate from the real ones. Although we never provide any labels for training the uncertainty, it automatically learns to represent the likelihood of the predicted results, even with a new input distribution. The corresponding variances of the whole trajectory on the North (y) and East (x) axes increase dramatically. This is because the location sees an integration from the location change in two axes, and it is inferred from both the variances of absolute heading and the delta L. Fig. 12 illustrated the correlation between the errors and uncertainties for the global heading and locations in North and East axes. It demonstrates that the uncertainties are positive correlated with the errors, and capable of reflecting the belief in the model prediction. Note that the small error does not always mean small uncertainty, because the mean errors can cancel each other out. For example, if the location transformation errors are -3m and 3m at previous step and current step, the absolute error would be 0 m, but the uncertainties have to accumulate to reflect the belief drift in system.

6.7 Trolley Tracking

We consider a more general problem without periodic motion, which is hard for traditional step-based PDR or SINS on a limited quality IMU. Tracking wheel-based motion, such as a shopping trolley/cart, robot or baby-stroller is highly challenging and hence under-explored. Current approaches to track wheeled objects are mainly based on visual odometry or visual-inertial odometry (VIO) [12], [13]. However, they fail when the device is occluded or operating in low light environments, such as placed in a bag. Moreover, their high energy- and computation-consumption also constrain further application. Here, we apply our model on a trolley tracking problem *using only inertial sensors*. Due to a lack of comparable techniques, our proposal is compared with the state-of-art visual-inertial odometry Tango.

Our experiment devices, namely an iPhone 7 and the Google Tango are attached on a trolley, pushed by a participant. Detailed experiment setup and results could be found in supplementary video [4]. High-precision motion reference was provided by Vicon. The VIO of Tango device is based on Kalman filtering to fuse the inertial navigation system with the visual features extracted from images. However, in VIO the image features are extracted by hand-designed algorithms. In some scenarios, it is hard to obtain enough visual features to estimate the geometry structure of a scene, for example, when cameras are in front of a blank wall, no useful features can be extracted. In our experiments, the VIO collapsed due to the lost features in the structureless and featureless wall of the experimental room, which further breaks down the entire system. From the trajectories from Vicon, our IONet and Tango in Fig. 13 our proposed approach shows almost the same accuracy as Tango, and even better robustness, because our pure inertial approach suffers less from environmental factors. With the help of visual features, VIO (Tango) can constrain error drift by fusing visual transformations and the inertial system, but it will collapse when capturing erroneous features or no features, especially in open spaces. This happened in our experiment, shown in Fig. 14. Although VIO can recover from the collapse, it still left a large distance error.

7 Conclusion and Future Work

We have presented a novel method for using inertial sensors to estimate displacements over a given time window. Specifically, inertial measurements and ground truth data were input to a neural network to learn the transformation between the raw measurements and the movement of the sensors unit. The method makes no assumption about either sensor placement or user motion and is therefore able to circumvent fundamental limitations of existing methods for inertial indoor navigation. The performance of the method was evaluated through experiments including not only typical pedestrian motions such as walking and running at different speeds, but also trolley tracking. Performance evaluations demonstrated the neural network outperformed competing algorithms based on standard inertial navigation systems and model-based step estimation in several scenarios.

Future work may focus on the estimation of positional and rotational displacements in three dimensions. Another future research direction is to combine the existing physical model of inertial navigation mechanism with deep neural networks. For example, IMUs can be calibrated before double integration by modelling the error distribution of inertial measurements via deep neural networks. Another alternative is to apply the neural network *after* the double integration. This could be implemented using the framework proposed in this paper, but by changing ground truth data in the training of the neural network.

Acknowledgments

We would like to thank our colleagues, Dr. Stefano Rosa, Shuyu Lin, Linhai Xie, Bo Yang, Jiarui Gan, Zhihua Wang, Zihang Lai, Peijun Zhao for their suggestions and help, and Prof. Hongkai Wen at University of Warwick, Dr. Sen Wang at Heriot-Watt University, Dr. Ronald Clark at Imperial College London for useful discussions.

References

[1] R. Harle, “A survey of indoor inertial positioning systems for pedestrians,” *IEEE Commun. Surveys Tuts.*, vol. 15, no. 3, pp. 1281–1293, Mar. 2013.

[2] D. Lymberopoulos, J. Liu, X. Yang, R. R. Choudhury, V. Handziski, and S. Sen, “A realistic evaluation and comparison of indoor location technologies: Experiences and lessons learned,” in *Proc. Int. Conf. Inf. Process. Sensor Netw.*, Seattle, Washington, 2015, pp. 178–189.

[3] J. Prieto, S. Mazuelas, and M. Z. Win, “Context-aided inertial navigation via belief condensation,” *IEEE Trans. Signal Process.*, vol. 64, no. 12, pp. 3250–3261, Jun. 2016.

[4] J. O. Nilsson, A. K. Gupta, and P. Hädel, “Foot-mounted inertial navigation made easy,” in *Proc. IEEE Int. Conf. Indoor Positioning and Indoor Navigation*, Busan, South Korea, Oct. 2014, pp. 24–29.

[5] F. Li, C. Zhao, G. Ding, J. Gong, C. Liu, and F. Zhao, “A reliable and accurate indoor localization method using phone inertial sensors,” in *Proc. ACM Int. Conf. Ubiquitous Comput.*, Pittsburgh, Pennsylvania, 2012, pp. 421–430.

[6] A. Brajdic and R. Harle, “Walk detection and step counting on unconstrained smartphones,” in *Proc. ACM Int. Conf. on Pervasive and Ubiquitous Comput.*, Zurich, Switzerland, 2013, pp. 225–234.

[7] Y. LeCun, Y. Bengio, and G. E. Hinton, “Deep learning,” *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[8] A. Kendall, M. Grimes, and R. Cipolla, “PoseNet: A convolutional network for real-time 6-DOF camera relocalization,” in *Proc. Int. Conf. on Computer Vision (ICCV)*, 2015.

[9] S. Wang, R. Clark, H. Wen, and N. Trigoni, “DeepVO: Towards end-to-end visual odometry with deep recurrent convolutional neural networks,” in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, Singapore, Singapore, May 2017, pp. 2043–2050.

[10] C. Chen, X. Lu, A. Markham, and N. Trigoni, “IONet: Learning to Cure the Curse of Drift in Inertial Odometry,” in *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence (AAAI)*, 2018.

[11] P. G. Savage, “Strapdown inertial navigation integration algorithm design part 1: Attitude algorithms,” *J. Guidance, Control, and Dynamics*, vol. 21, no. 1, pp. 19–28, 1998.

[12] M. Bloesch, S. Omari, M. Hutter, and R. Siegwart, “Robust visual inertial odometry using a direct EKF-based approach,” in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Sep. 2015, pp. 298–304.

[13] M. Li and A. I. Mourikis, “High-precision, consistent EKF-based visual-inertial odometry,” *The Int. J. Robotics Research*, vol. 32, no. 6, pp. 690–711, 2013.

[14] S. Leutenegger, S. Lynen, M. Bosse, R. Siegwart, and P. Furgale, “Keyframe-based visual-inertial odometry using nonlinear optimization,” *Int. J. Robotics Research*, vol. 34, no. 3, pp. 314–334, 2015.

[15] C. Forster, L. Carlone, F. Dellaert, and D. Scaramuzza, “On-Manifold Preintegration for Real-Time Visual-Inertial Odometry,” *IEEE Transactions on Robotics*, vol. 33, no. 1, pp. 1–21, 2017.

[4] Video is available at: https://youtu.be/L5LtE-PQuHk

[16] 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.

[17] E. Foxlin, “Pedestrian tracking with shoe-mounted inertial sensors,” *IEEE Comput. Graphics and Appl.*, vol. 25, no. 6, pp. 38–46, Nov. 2005.

[18] C. Chen, Z. Chen, X. Pan, and X. Hu, “Assessment of Zero-Velo city Detectors for Pedestrian Navigation System using MIMU,” in *IEEE Chinese Guidance, Navigation and Control Conlerence*, 2016, pp. 128–132.

[19] Y. Shu, K. G. Shin, T. He, and J. Chen, “Last-mile navigation using smartphones,” in *Proc. Int. Conf. Mobile Comput. Netw.*, Paris, France, 2015, pp. 512–524.

[20] P. Davidson and R. Pich, “A survey of selected indoor positioning methods for smartphones,” *IEEE Commun. Surveys Tuts.*, vol. 19, no. 2, pp. 1347–1370, 2017.

[21] Z. Xiao, H. Wen, A. Markham, and N. Trigoni, “Lightweight map matching for indoor localisation using conditional random fields,” in *Proc. Int. Symp. Inf. Process. Sensor Netw.*, Berlin, Germany, 2014, pp. 131–142.

[22] S. Hilsenbeck, D. Bobkov, G. Schroth, R. Huitl, and E. Steinbach, “Graph-based data fusion of pedometer and WiFi measurements for mobile indoor positioning,” in *Proc. ACM Int. Conf. Pervasive and Ubiquitous Comput.*, Seattle, Washington, 2014, pp. 147–158.

[23] Q. Xu, R. Zheng, and S. Hranilovic, “IDyLL: Indoor Localization using Inertial and Light Sensors on Smartphones,” in *Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing - UbiComp ’15*, 2015, pp. 307–318.

[24] S. Wang, H. Wen, R. Clark, and N. Trigoni, “Keyframe based large-scale indoor localisation using geomagnetic field and motion pattern,” in *IEEE/RSJ Int. Conf. Intell. Robot. Syste. (IROS)*, Daejeon, South Korea, Oct. 2016, pp. 1910–1917.

[25] C. X. Lu, P. Zhao, C. Chen, R. Tan, and N. Trigoni, “Simultaneous Localization and Mapping with Power Network Electromagnetic Field,” in *The 24th Annual International Conference on Mobile Computing and Networking (MobiCom)*, 2018.

[26] S. Ahuja, W. Jirattigalachote, and A. Tosborvorn, “Improving Accuracy of Inertial Measurement Units using Support Vector Regression,” Tech. Rep., 2011.

[27] A. Parate, M. C. Chiu, C. Chadowitz, D. Ganesan, and E. Kalogerakis, “RisQ: Recognizing smoking gestures with inertial sensors on a wristband,” in *Annual International Conference on Mobile Systems, Applications, and Services (MobiSys)*, 2014, pp. 149–161.

[28] A. Mannini and A. M. Sabatini, “Walking speed estimation using foot-mounted inertial sensors: Comparing machine learning and strap-down integration methods,” *Medical engineering & physics*, vol. 36, no. 10, pp. 1312–1321, 2014.

[29] ——, “Machine learning methods for classifying human physical activity from on-body accelerometers,” *Sensors*, vol. 10, no. 2, pp. 1154–1175, 2010.

[30] A. Valtazanos, D. Arvind, and S. Ramamoorthy, “Using wearable inertial sensors for posture and position tracking in unconstrained environments through learned translation manifolds,” in *2013 ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*. IEEE, 2013, pp. 241–252.

[31] M. Yuwono, S. W. Su, Y. Guo, B. D. Moulton, and H. T. Nguyen, “Unsupervised nonparametric method for gait analysis using a waist-worn inertial sensor,” *Applied Soft Computing*, vol. 14, pp. 72–80, 2014.

[32] X. Xiao and S. Zarar, “Machine Learning for Placement-Insensitive Inertial Motion Capture,” *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 6716–6721, 2018.

[33] A. Graves and N. Jaitly, “Towards end-to-end speech recognition with recurrent neural networks,” in *Proc. Int. Conf. Machine Learning*, vol. 32, no. 2, Bejing, China, Jun. 2014, pp. 1764–1772.

[34] A. M. Dai and Q. V. Le, “Semi-supervised sequence learning,” in *Advances in Neural Inf. Process. Syst.*, Montreal, Canada, Dec. 2015, pp. 3079–3087.

[35] J. Donahue, L. A. Hendricks, M. Rohrbach, S. Venugopalan, S. Guadarrama, K. Saenko, and T. Darrell, “Long-term recurrent convolutional networks for visual recognition and description,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 4, pp. 677–691, Apr. 2017.

[36] K. Konda and R. Memisevic, “Learning visual odometry with a convolutional network,” in *Proc. Int. Conf. Computer Vision Theory and Appl. (VISAPP)*, Berlin, Germany, 2015, pp. 486–490.

[37] T. Zhou, M. Brown, N. Snavely, and D. G. Lowe, “Unsupervised learning of depth and ego-motion from video,” in *Proc. IEEE Int. Conf. Comput. Vision and Pattern Recognition (CVPR)*, Honolulu, HI, Jul. 2017, pp. 6612–6619.

[38] S. Wang, R. Clark, H. Wen, and N. Trigoni, “End-to-end, sequence-to-sequence probabilistic visual odometry through deep neural networks,” *Int. J. Robot. Research*, vol. 37, no. 4-5, pp. 513–542, Oct. 2018.

[39] R. Clark, S. Wang, A. Markham, N. Trigoni, and H. Wen, “VidLoc : 6-DoF video-clip relocalization,” *CVPR*, 2017.

[40] R. Clark, S. Wang, H. Wen, A. Markham, and N. Trigoni, “VINet: Visual-inertial odometry as a sequence-to-sequence learning problem.” *Association for the Advancement of Artificial Intell.*, pp. 3995–4001, 2017.

[41] N. El-Sheimy, H. Hou, and X. Niu, “Analysis and modeling of inertial sensors using Allan variance,” *IEEE Trans. Instrum. Meas.*, vol. 57, no. 1, pp. 140–149, Jan. 2008.

[42] P. Hippe, *Windup in Control*, 2006.

[43] J. M. Hausdorff, “Gait dynamics, fractals and falls: Finding meaning in the stride-to-stride fluctuations of human walking,” *Human Movement Sci.*, vol. 26, no. 4, pp. 555–589, Aug. 2007.

[44] M. Lindfors, G. Hendeby, F. Gustafsson, and R. Karlsson, “Vehicle speed tracking using chassis vibrations,” *IEEE Intelligent Vehicles Symposium, Proceedings*, vol. 2016-Augus, no. Iv, pp. 214–219, 2016.

[45] Z. Xiao, H. Wen, A. Markham, and N. Trigoni, “Robust indoor positioning with lifelong learning,” *IEEE J. Sel. Areas Commun.*, vol. 33, no. 11, pp. 2287–2301, Nov. 2015.

[46] K. Greff, R. K. Srivastava, J. Koutnk, B. R. Steunebrink, and J. Schmidhuber, “LSTM: A search space odyssey,” *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 28, no. 10, pp. 2222–2232, Oct. 2017.

[47] W. Samek, A. Binder, G. Montavon, S. Lapuschkin, and K. Mller, “Evaluating the visualization of what a deep neural network has learned,” *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 28, no. 11, pp. 2660–2673, Nov. 2017.

[48] Y. Gal and Z. Ghahramani, “Dropout as a Bayesian approximation: Representing model uncertainty in deep learning,” *ICML*, vol. 48, 2015.

[49] P. Agarwal, “Robust graph-based localization and mapping,” Ph.D. dissertation, University of Freiburg, 2015.

[50] J. L. Crassidis, “Sigma-point kalman filtering for integrated gps and inertial navigation,” *IEEE Trans. Aerosp. Electron. Syst.*, vol. 42, no. 2, pp. 750–756, Apr. 2006.

[51] A. Kendall and Y. Gal, “What uncertainties do we need in Bayesian deep learning for computer vision?” in *Advances in Neural Information Processing Systems 30*, 2017, pp. 5574–5584.

[52] Vicon. Vicon Motion Capture Systems. 2017.

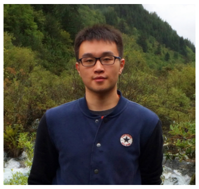
[53] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” in *Proc. Int. Conf. Learning Representations (ICLR)*, 2014.

[54] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, “Dropout: A simple way to prevent neural networks from overfitting,” *J. Machine Learning Research*, vol. 15, pp. 1929–1958, 2014.

[55] Z. Xiao, H. Wen, A. Markham, and N. Trigoni, “Robust pedestrian dead reckoning (R-PDR) for arbitrary mobile device placement,” in *Proc. Int. Conf. Indoor Positioning and Indoor Navigation*, Busan, South Korea, Oct. 2014, pp. 187–196.

[56] Tango. Google Tango Tablet. 2014.

Changhao Chen is currently a PhD student in Department of Computer Science, University of Oxford. Before that, he obtained his MEng degree at National University of Defense Technology, China, and BEng Degree at Tongji University, China. His research interest lies in machine learning for signal processing, and intelligent sensor systems, with applications on ubiquitous localization and pedestrian navigation using mobile devices.



Dr. Xiaoxuan Lu is currently a PostDoctoral researcher at Department of Computer Science, University of Oxford. Before that, he obtained his Ph.D degree at University of Oxford, and MEng degree at Nanyang Technology University, Singapore. His research interest lies in Cyber Physical Systems, which use networked smart devices to sense and interact with the physical world.

Dr. Johan Wahlström is a postdoc researcher at Oxford University since January 2018. He is working on indoor navigation for emergency responders. Johan received his MSc degree in Engineering Physics and PhD degree in Electrical Engineering from KTH Royal Institute of Technology, Stockholm, Sweden, in 2014 and 2017, respectively.

Prof. Andrew Markham is an Associate Professor at the Department of Computer Science, University of Oxford. He obtained his BSc (2004) and PhD (2008) degrees from the University of Cape Town, South Africa. He is the Director of the MSc in Software Engineering. He works on resource-constrained systems, positioning systems, in particular magneto-inductive positioning and machine intelligence.

Prof. Niki Trigoni is a Professor at the Department of Computer Science, University of Oxford. She is currently the director of the EPSRC Centre for Doctoral Training on Autonomous Intelligent Machines and Systems, and leads the Cyber Physical Systems Group. Her research interests lie in intelligent and autonomous sensor systems with applications in positioning, healthcare, environmental monitoring and smart cities.