

Robust Indoor Positioning with Lifelong Learning

Zhuoling Xiao, Hongkai Wen, Andrew Markham, and Niki Trigoni

Abstract—Inertial tracking and navigation systems have been playing an increasingly important role in indoor tracking and navigation. They have the competitive advantage of leveraging not requiring expensive infrastructure - only existing smart mobile devices with embedded inertial measurement units. When aided with other sources of information, such as radio data from existing WiFi/BLE infrastructure, and environment constraints from floor plans or radio maps, they often report great performance of 0.5-2 meters. Given the promising results, what is it that prevents the widespread adoption of this tracking solution? We argue that pedestrian dead reckoning (PDR) techniques are often evaluated in a specific context, and are not mature enough to handle variations in user motion, device type, device placement or environment. They typically use a number of parameters that require careful context-specific tuning, which is labor intensive and requires expert knowledge. In this paper, we propose two novel approaches to addressing these problems: Our first contribution is a Robust Pedestrian Dead Reckoning algorithm (R-PDR), that is based on general physics principles that underpin human motion and is by design robust to context changes. The second contribution is a novel way of interaction between the PDR and map matching layers based on the principle of lifelong learning. Unlike traditional approaches where information flows unidirectionally from the PDR to the map matching layer, we introduce a feedback loop that can be used to automatically tune the parameters of the PDR algorithm. This is not dissimilar to the way that people improve their navigation skills when they repeatedly visit the same environment. Extensive experiments in multiple sites, with a variety of users, devices and device placements show that the combination of a robust PDR with a lifelong learning tracker can achieve sub-metre accuracy with no user effort for parameter tuning.

Index Terms—Indoor positioning, pedestrian dead reckoning, lifelong learning.

I. INTRODUCTION

Indoor tracking and navigation is a fundamental need for pervasive and context-aware smartphone applications. Applications of indoor positioning include smart retail, navigation through large public spaces like transport hubs, and assisted living. The ultimate objective of an indoor positioning system is to provide continuous, reliable and accurate positioning on smartphone class devices. Inertial tracking with or without the aid of existing maps has received a lot of attention recently, but existing solutions lack generality and robustness. They are typically targeted at achieving high accuracy in specific contexts, and are fine tuned for specific phone placements, users, devices or environments. When tested in different conditions than the ones designed for, their performance degrades significantly.

For example, the majority of existing solutions assume that the user holds the device in a specific way (e.g. texting mode) [1]. More advanced solutions first use a classifier to infer the user walking pattern and phone placement (e.g. hand swinging, texting, etc.) [2], [3], and then exploit this information to optimize the inertial tracking system [4]. However, even the state of the art approaches can only handle a limited number of phone placement options. When the user deviates from this predefined set, the tracking system is at a loss for how to handle the new case. Another major issue is that the inertial tracking system typically assumes knowledge of a number of parameters that account for user, device and environment variability. Examples of user variability parameters include those that relate the user height, step frequency and acceleration variance to their step length [5]–[8]. Further parameters are needed to model the noise of inertial sensors, and environment-specific magnetic distortions [9]. The vast tuning effort involved in optimizing PDR systems is one of the major hurdles that prevent inertial tracking systems from becoming mainstream. In this paper, we move away from context-specific fine-tuning, and pursue the vision of general-purpose robust tracking with lifelong learning. Our proposed approach is based on two overarching principles:

Robustness: We propose a phone placement classification scheme that is simple, yet comprehensive, i.e. covers all possible ways of attaching the phone to the human body. Based on this scheme, we propose R-PDR, a robust pedestrian dead reckoning (PDR) algorithm that is based on general physics principles underpinning human motion, and avoids using context-specific heuristics, such as if phone is in a pant pocket, use technique A; else if in text mode, use technique B. By design, R-PDR is robust to user, device and phone placement variability.

Lifelong learning: We appreciate the fact that PDR parameters often require context-specific tuning, but we propose that this information should be learnt, rather than manually tuned or assumed as input. To this end, we exploit the interplay between the pedestrian dead reckoning (PDR) and the map-matching (MM) layers. Existing work typically feeds PDR output into a map matching technique (e.g. particle filter/HMM/CRF), which exploits map information to correct the inertial trajectory. Our proposed system named Lifelong Learning Tracker (LL-Tracker) introduces a reverse link between the two layers. It augments the functionality of the map-matching layer to learn context-specific parameters, which are then fed back to the underlying PDR layer. Although in this paper we demonstrate the efficacy of LL-Tracker in learning parameters of our robust pedestrian dead reckoning

algorithm (R-PDR), we argue that the idea of lifelong learning is broadly applicable to improve the performance of any PDR implementation that requires parameter tuning.

To summarise, the contributions of this paper are as follows:

- A novel comprehensive way of classifying human motion.
- A robust pedestrian dead reckoning algorithm (R-PDR) that exploits the new classification scheme and is robust to user, device and phone placement variability.
- A lifelong learning tracker (LL-Tracker) that eases the effort of parameter tuning by introducing a feedback loop between the map matching and PDR layer.
- Evaluation of R-PDR and LL-Tracker in a wide range of contexts, varying device type, user, phone placement and building.

The remainder of this paper is organized as follows: Sec. II outlines the proposed system architecture, and Sec. III and IV delve into the details of the proposed algorithms, R-PDR and LL-Tracker, and compare them with related work. Sec. V evaluates the performance of proposed and competing tracking algorithms and Sec. VI concludes the paper.

II. SYSTEM ARCHITECTURE

The main motivation of our work is to build an indoor tracking system that is robust to noise introduced by inertial measurement unit (IMU) and to context variability. Note that PDR algorithms typically include tasks for motion mode recognition, step detection, step length estimation and step heading estimation. In accurately accomplishing these tasks, we have addressed the following key challenges:

Motion classification: Despite a plethora of work on recognizing different motion modes, to our knowledge, there is still a lack of a comprehensive classification scheme that captures the wide range of human motion and device attachment. Moreover, solutions that capture many motion modes are typically complex, since they have to include a large number of motion-specific optimisation rules, especially for solving the problem of step detection.

Device orientation estimation: The performance of the PDR algorithm largely depends on the ability to accurately track the orientation of the mobile device. This task is particularly challenging in the context of unconstrained devices embedded with low-cost IMU sensors. Generally, the gyro drift and magnetic disturbances, especially in indoor environments, are the major obstacles to accurate orientation estimation. Addressing this problem is essential, because orientation information is used by nearly all tasks of the PDR algorithm (Sec. III).

Acceleration long/short term drift: Another major challenge is the correction of short-term and long-term drift in the acceleration signal. The noise that distorts the acceleration signal is a major issue because it creeps into all four PDR tasks, as will be explained in Sec. III.

Parameter tuning: The inertial tracking system generally needs careful tuning of parameters such as sensor bias and step length estimation parameters, to name a few. The fact that certain parameters are context-specific, i.e. dependent on the environment, device or individual, poses a great challenge to the widespread application of inertial tracking systems.

Fig. 1 shows the proposed system architecture that was designed to address the challenges discussed above. The layer referred to as Robust Pedestrian Dead Reckoning (R-PDR), is equipped with three novel capabilities: Firstly, it features a novel motion classification scheme, which only includes two classes, but captures a wide spectrum of human motion and device placement combinations. Secondly, leveraging simple observations about human motion, R-PDR provides a robust mechanism for correcting short-term and long-term acceleration bias. Thirdly, R-PDR features a novel and robust mechanism for estimating device orientation. Sec. III-A first describes in detail the three new capabilities of R-PDR, while Sec. III-B explains how they become instrumental to reinforcing its four steps.

The output of the robust R-PDR algorithm is then fed into the Map Matching layer, which takes into account map constraints to correct long-term drift. Unlike existing systems where information flows upwards only, we introduce a feedback loop, wherein the output of map matching is used to fine tune the PDR parameters. The resulting end-to-end algorithm is a Lifelong Learning Tracker (LL-Tracker), which *learns from experience* and gradually gets better at handling familiar tracking scenarios. In Sec. IV, we describe the feedback loop in more detail and provide examples that illustrate the benefits of cross-layer optimisation.

III. ROBUST PEDESTRIAN DEAD RECKONING

A. Novel Capabilities

We are now in a position to drill down to the three novel capabilities of R-PDR, starting with a novel scheme devised to address the current challenges in classifying human motion. **Novel Motion Classification:** Existing work has proposed techniques for identifying a variety of motion modes, including phoning, texting, hand swinging, static, etc. [10], [11]. However, these motion modes are only a subset of all possible human motion modes, and suffice in controlled experiment settings. To make the motion mode recognition work in unconstrained real world settings, a large number of new motion modes has to be defined, leading to a steep rise if we aim to count all possible motion modes of human beings. In this paper, instead of separating motion modes according to their differences, we take the approach of grouping them together using the features they have in common. Specifically, we define a very simple classification scheme consisting of only two modes: symmetric and asymmetric. Informally, the symmetric mode is the class of motions that are impacted in a similar manner by the movement of the left and right leg; the asymmetric mode includes all other types of motion.

Examples of typical symmetric motion modes include, but are not limited to: *Texting*: The mobile device is held in front of the user while walking; *Phoning*: The mobile device is held close to the head while walking; *Heavy bag*: The mobile device is put in a heavy bag (so the hand of the pedestrian is not swinging) while walking; *Shirt pocket*: The mobile device is put in a shirt pocket while walking; *Static*: The mobile device is not moving, wherever it is put.

Similarly, some typical asymmetric motion modes include: *Hand swinging*: The mobile device is held in a swinging hand

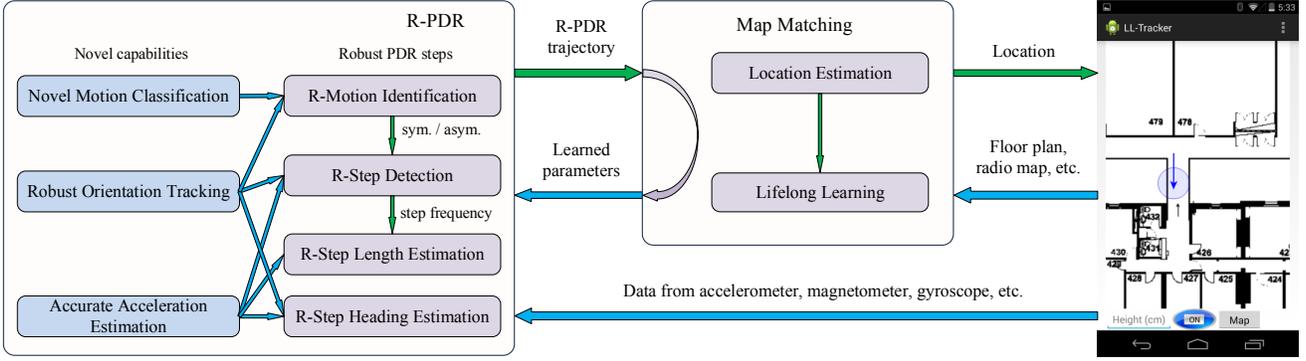


Fig. 1. System architecture of LL-Tracker.

while walking; *Trouser pocket*: The device is put in a trouser pocket (front or back) while walking; *Belt*: The device is fastened to the belt while walking; *Handshaking*: The device is (possibly periodically) shaken while not moving.

Fig. 2 illustrates a few examples of symmetric and asymmetric motion modes. Notice that all symmetric motion modes have similar periodic acceleration signals, with a period corresponding to a single step, while all asymmetric motion modes have similar periodic orientation patterns, with a period corresponding to two steps. The benefits of classifying motions into symmetric and asymmetric are three fold. First, it is a comprehensive scheme that leaves no motion out. Second, the motion recognition step itself is a very easy task, which requires simple cross correlation with sine waves (Sec. III-B). Lastly, the fact that symmetric motions have periodic acceleration signals while asymmetric motion modes has periodic orientation signals greatly facilitates the task of step detection also discussed in Sec. III-B.

Robust Orientation Estimation: Being able to accurately track the device’s orientation is the foundation for accurate long term inertial tracking. This is because most steps of PDR, including step detection, and step length and heading estimation, are based on the orientation of the device. However, tracking device orientation remains a major issue with low-cost IMU sensors embedded in unconstrained mobile devices. The gyro drift and magnetic disturbances especially in indoor environments are the major obstacles on the way to accurate orientation tracking. Most existing work uses Kalman filters, e.g. indirect Kalman filters [12], extended Kalman filters (EKF) [13] and Unscented Kalman filters (UKF) [14], to track device orientation with angular velocities from gyroscope to periodically update the orientation. The gyro drift in horizontal plane (yaw) can be compensated with magnetic field measurements especially if we have a map of the area’s magnetic distortions. However, the gyro drifts in roll and pitch angles are still a pressing problem for devices with motion acceleration [15]. We argue that this problem can be addressed by accurately estimating the gravitational acceleration from the raw accelerometer signal. We can then use the gravity vector as an observation in the Kalman filter to correct drift in roll and pitch.

Currently there are two dominant approaches to estimate the gravity component of accelerometer measurements. The first approach, proposed by Mizell et. al [16], uses the mean

over a window of fixed duration to estimate the gravity. This approach is elegant and simple but it is not suitable for mobile devices used by pedestrians. The major reason is that a change in orientation introduces a considerable lag to accurate gravity estimation. To reduce the lag, it is possible to shorten the time window over which the gravity is estimated [17]. However, this comes with a decrease in the accuracy of gravity estimates. The second approach, proposed by Hemminki et. al [18], estimates gravity only when the variation of the acceleration signals over a period of time is below a certain threshold. This approach reduces the lag introduced by the first approach, but also suffers from two fundamental limitations. First, this approach works only when the mobile device is static for a period of time; it does not work when the user is in motion, which is when gravity estimates are mostly needed. Secondly, the thresholds of variations in different motion modes are different, making it hard to apply in real-world settings. A similar approach [19] opportunistically estimates the gravity for the device’s attitude estimation when the angular velocity of the device is detected to be less than a threshold.

To address these limitations, we propose a novel algorithm that can accurately estimate the gravity vector without lag and works all the time. The key fact that it exploits is that gyro sensors are reliable over a short period of time. Assume we want to estimate the gravity at time t given a window of historical accelerometer data $\mathbf{a}_{t-T:t}$ of size T . We first rotate acceleration signals $\mathbf{a}_{t-T:t-1}$ to the same orientation as \mathbf{a}_t , and then estimate the gravity at time t as the mean of the acceleration readings after the rotation:

$$g_t = \mathbf{R}_t \sum_{\tau=t-T}^t \mathbf{R}_\tau^T \mathbf{a}_\tau / T, \quad (1)$$

where \mathbf{R}_τ is the rotation matrix from the earth coordinate system to the device coordinate system at time τ . The rotation \mathbf{R}_τ can be easily converted from the orientation at time τ .

Experiments have been conducted to compare the proposed algorithm with the state-of-the-art gravity estimation techniques. In the experiments the pedestrian walked normally with the mobile device in texting mode. The gravity vector is estimated using the proposed and competing approaches and then rotated to the earth coordinate system according to the current orientation estimate of the device. The gravity value, as shown in Fig. 3(a), is then estimated as the acceleration in the Z axis. It is observed that Mizell’s approach is the

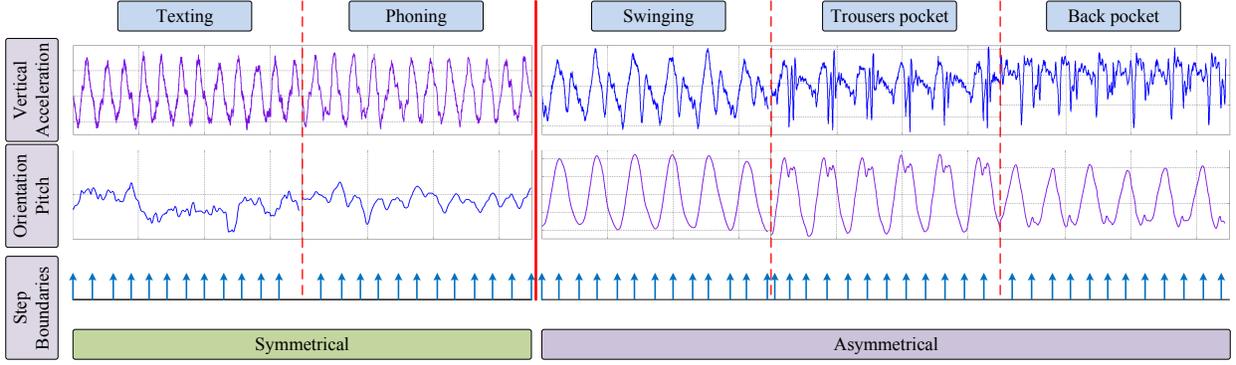


Fig. 2. Acceleration and orientation signals of some typical motions. Note how symmetric motions have strong periodicity in acceleration, whereas asymmetric motion has strong periodicity in pitch.

most inaccurate due to its inability to manage the orientation change of the device. Hemminki’s approach provides more accurate, yet very sparse gravity estimates, because in most cases the motion of the pedestrian makes the variation in acceleration bigger than the threshold. In comparison, the proposed approach offers accurate gravity estimates and works continuously.

The gravity vector estimates are then fed into the Kalman filter as additional observations (along with magnetometer measurements) to estimate the orientation of the device. We can indirectly see the benefits of the proposed orientation tracking approach as follows. We first rotate every acceleration vector to the earth coordinate system according to the device orientation estimate at that moment. Then the gravity magnitude is simply estimated as the mean of the acceleration in Z axis in a short window. Fig. 3(b) shows the long term estimate of gravity with and without the proposed orientation tracking algorithm. As expected, the drift of device orientation grows quickly without the proposed approach, leading to the gravity magnitude estimate drifting away from the ground truth. This acceleration drift caused by orientation drift can result in miscounting user steps, as will be explained in Sec. III-B. However, this drift has been successfully corrected by the proposed approach which computes and relies on accurate gravity vector estimates.

Accurate Acceleration Estimation: Though the proposed gravity estimation algorithm can guarantee the robust long term orientation tracking of the device, it is not sufficient to accurately estimate acceleration on the horizontal plane, which is important for step detection. The major reason is that even a very small tilt of the device, e.g. two degrees, which may not be corrected by the above algorithm, can significantly impact the acceleration signal on the horizontal plane. For instance, a tilt of two degrees can only change the acceleration reading in Z axis from 9.806 m/s^2 to 9.801 m/s^2 , but it can offset the acceleration on the horizontal plane (either X or Y) by 0.342 m/s^2 . To make things worse, this offset is significantly amplified through double integration which results in large errors in distance estimation. For example, the two degree tilt error can result in an error of up to 20 meters within 10 seconds.

To address this problem, we have further explored the physical constraints from human walking. It is demonstrated

in [20] that humans have surprisingly consistent walking patterns for consecutive steps. Inspired by [20], we design an algorithm to estimate and correct the orientation error using the repetitiveness in walking patterns, called repetitive pattern update (REPUT). We rely on two key assumptions: first, we consider the velocity of the mobile device to be almost the same after each step in symmetric motion mode or every two steps in asymmetric motion mode¹. Second, we assume that the orientation drift, described as the reverse rotation \mathbf{R} , is the same during one or two steps. Let a single/double step start at time t_0 with velocity \mathbf{v}_0 and end at time t_e with velocity \mathbf{v}_e . The end velocity is expressed as follows:

$$\mathbf{v}_e = \mathbf{v}_0 + \sum_{t=t_0}^{t_e} \mathbf{R} \cdot \mathbf{a}_t \cdot \Delta t, \quad (2)$$

where \mathbf{R} is the rotation matrix that corrects the orientation drift, $\mathbf{a}_t = [a_t^x, a_t^y, a_t^z]$ is the acceleration vector at time t , and Δt is the acceleration sampling interval. Then based on our REPUT assumption, we have

$$\Delta \mathbf{v} = \mathbf{v}_e - \mathbf{v}_0 = \mathbf{R} \sum_{t=t_0}^{t_e} \mathbf{a}_t \cdot \Delta t = \mathbf{0}. \quad (3)$$

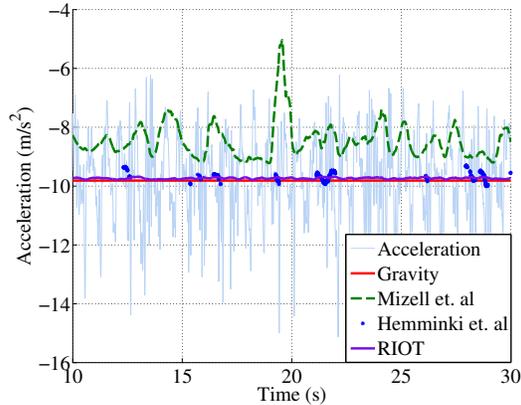
Other repetitive patterns can also be formulated. For instance, in 2D positioning the device has the same distance off the ground before and after one step, which can be formulated as

$$\Delta l = \mathbf{R} \sum_{t=t_0}^{t_e} \left(\sum_{\tau=t_0}^t a_t^z \Delta \tau \right) \Delta t = 0. \quad (4)$$

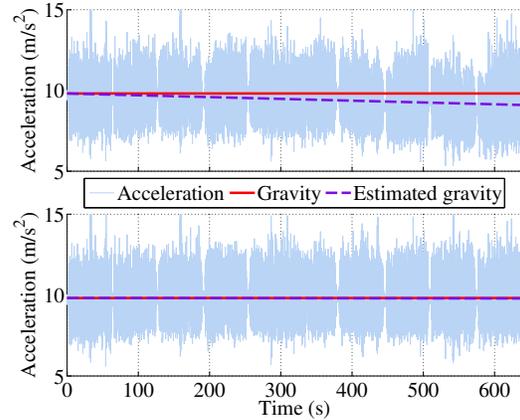
where a_t^z is the vertical acceleration along Z axis. It is possible that these conditions could contradict each other because either the velocity difference $\Delta \mathbf{v}$ or distance difference Δl cannot be strictly zero. Therefore, least-square solutions of the counter-drift rotation matrix \mathbf{R} should be obtained from simple optimization.

To test the effectiveness of the REPUT algorithm, we have conducted a simple experiment during which the pedestrian walks normally with the device at hand. Fig. 4 shows a typical example of acceleration signals before and after the REPUT algorithm. We can observe that the acceleration

¹The start and end of every step or two can be easily detected with an enhanced zero crossing detector in Sec. III-B.



(a) Gravity estimated using different techniques.



(b) Vertical acceleration without (top) and with robust gravity estimation (bottom).

Fig. 3. Illustrative examples of robust orientation estimation.

signals along X and Y axis drift significantly over time, if we do not exploit the repetitive pattern assumption; the drift is successfully corrected by REPUT.

B. R-PDR steps

We are now in a position to show how they are employed to design robust versions of the four pedestrian dead reckoning steps: namely R-Motion Identification, R-Step Detection, R-Step Length Estimation, and R-Heading estimation.

R-Motion Identification: This step exploits the novel classification scheme, which consists of two motion classes only - symmetric and asymmetric. In addition, it leverages the corrected orientation and acceleration signals in order to identify the motion class. Fig. 2 illustrates an interesting pattern that we exploit for motion identification. In symmetric motion modes, e.g. texting and phoning, acceleration signals resemble sine waves, whereas in asymmetric motion modes, orientation signals are closer to sine waves. Thus, our classifier is based on assessing which one of the vertical acceleration or orientation waveforms has the strongest periodic component. More precisely, we compute the discrete Fourier transform (DFT) of acceleration and orientation over the frequency range that corresponds to human walking, typically between 1.2 Hz and 2.5 Hz. We then use an energy detector to decide whether the device placement is resulting in symmetric (vertical acceleration has stronger periodicity than orientation) or asymmetric motion (orientation has stronger periodicity than vertical acceleration).

R-Step Detection: Once the motion mode is identified, the next task of a PDR algorithm is to detect human steps. Typical existing step detection algorithms detect steps by identifying the cyclic patterns in acceleration signal, e.g. peak detection [5], [7], [21], zero-crossings [22]–[24], spectral analysis like Fourier transform [25], [26], short time Fourier transform [27], and continuous wavelet transform [8], and auto/cross correlation [28], [29]. The aforementioned techniques assume that only walking behavior generate periodic acceleration signals but this is not true. Hence, they cannot distinguish between a real step and a simple hand swing

without the user really moving forward. This is because the two actions can actually generate very similar acceleration signal patterns. The problem of false positive step detection is worse when applying algorithms that use the magnitude of the acceleration signals [30], and is further exacerbated with less constrained motion modes, like crawling, stumbles, side-steps, and shuffles [31].

To summarise, existing techniques only focus on identifying the boundaries of *step-like* signal segments, referred to as step pattern extraction. In comparison, the proposed R-Step Detection algorithm not only proposes a novel way of performing step pattern extraction, but also introduces an additional classification step for distinguishing whether the extracted step is real or fake. To clarify, a real step is defined as a step with locomotion, as opposed to a fake step without locomotion. An example of a fake step is one detected when a person swings the phone in her hand without actually moving her legs. Details of the novel step pattern extraction and step classification modules are provided below.

Part 1: Step pattern extraction: The proposed approach here aims to identify the boundaries of step-like signal segments in a lightweight online fashion. We leverage our novel motion mode classification scheme to extract steps very accurately with time domain analysis. The approach that we use depends on whether the motion is symmetric or asymmetric. If it is symmetric, we use the acceleration signal for step detection, because it has a simple sine-wave-like pattern in symmetric motion mode. Step boundaries here are detected with enhanced zero crossing detector similar to [24]. However, as shown in Fig. 2, the acceleration patterns of the asymmetric motion modes differ significantly, making it extremely difficult to identify steps with temporal features like peaks or zero crossings. Therefore, if the motion is asymmetric, we identify the orientation signal as the key to the step extraction, because the change of orientation instead of acceleration is highly periodic in asymmetric motion modes, as shown in Fig. 2. In this case, step boundaries are the peaks and troughs of the device orientation signal - e.g. first step is from trough to peak, second step is from peak to trough etc. (provided that they are

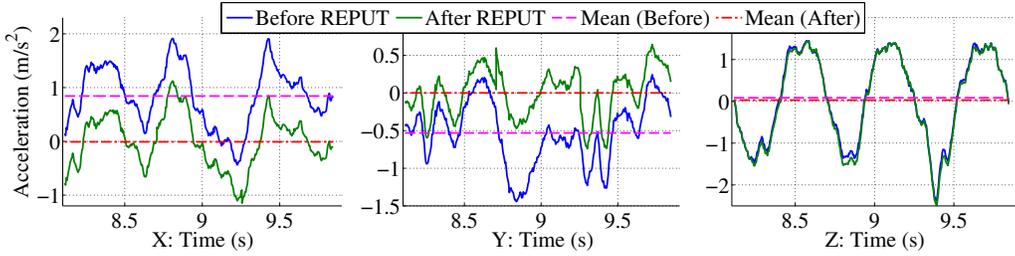


Fig. 4. Acceleration signals before and after REPUT. According to the walking nature of human beings, the X and Y acceleration signals are expected to have mean values close to zero in a window. However, the mean acceleration drifts significantly away from zero without REPUT.

far apart to avoid local fluctuations).

Part 2: Step classification: The step pattern extraction step is not sufficient to distinguish between real and fake steps. We argue that a further classification task is needed, that takes into account informative features of the orientation and acceleration signals. In particular, the horizontal and vertical displacements are identified as informative features in telling whether real locomotion happens to the mobile device. They are derived from the double integration of the acceleration signal during one step (in the x-y plane / z axis resp.). These two features are extremely informative because they can capture the fundamental difference between real and fake steps. In real steps, the horizontal displacement is higher and the vertical displacement is lower than in fake steps. The cross correlation between the acceleration signal and the orientation signal is also identified as an effective feature to reject fake steps because the acceleration and orientation signals tend to be strongly correlated if we shake the device, which is not true for normal walking behaviour. The orientation offset is another feature used for real/false step classification. The orientation offset measures how much the orientation has changed during one step. This feature is proposed to prevent the acceleration generated by turns from being taken into account in calculating the horizontal and vertical displacement.

The strong features defined above allow us to accurately distinguish between real and fake steps using a simple decision tree. We run a series of experiments to compare the proposed R-Step Detection algorithm with state-of-the-art step detection algorithms, including peak detection (PD) [21], zero-crossing detection (ZD) [24], short time Fourier transform (STFT) [32], continuous wavelet transform (CWT) [33], and auto-correlation (AC) [29]. Our experiments involve different users, attachments and a total of over 15,000 real steps. The step detection accuracy of the competing approaches in our experiments is similar to the accuracy reported in [27]. Fig. 5(a) shows the step detection accuracy of the proposed and competing approaches. It is observed that the proposed approach consistently features very high accuracy regardless of the device attachment. We also demonstrate the performance of existing algorithms in determining whether a step has actually been taken or not, using data from a number of behaviours that are repetitive but do not correspond to forward displacement. Each behaviour, such as tapping, nodding, and foot tapping, was performed 200 times. It is observed from Fig. 5(b) that the proposed algorithm reports close to 0 real steps while all other algorithms report approximately 200 steps, which

demonstrates the effectiveness of the proposed algorithm in rejecting false positive steps.

R-Step Length Estimation: Unlike the foot-mounted inertial navigation system [34], double integration of the acceleration to get the step length is not applicable to handheld devices, reason being that zero velocity update (ZUPT) is not applicable to handheld devices because only the heel can be regarded as static when one foot is planted on the floor during walking.

Various models have been proposed to address the step length estimation problem. The simplest approach is to make it a constant [35] because pedestrians have surprisingly constant step length with a natural walking pace [20]. However, when walking with others, people are likely to adjust their natural walking pace, which leads to a significant change of the step length [31]. The most common approaches used to estimate the step length are linear [6], [8] or nonlinear [5], [7] model to relate the step length to variables like pedestrian height, step frequency, acceleration variance, etc. These models are easy to implement and thus widely adopted in practice. An alternative of the linear/nonlinear model is regression. Machine learning regression techniques like support vector machine [3], [30] and neural networks [36] have also been applied based on various features extracted from the sensor data.

In this study we use a simple step length model [10], which uses acceleration signal to detect the step frequency (denoted with f_t at time t) which is later used to estimate step length l_t :

$$l_t = h(\alpha f_t + \beta) + \gamma. \quad (5)$$

where h is the pedestrian height, α is the step frequency coefficient, β is the step constant given pedestrian height, and γ is the step constant. The accuracy of this model is not satisfactory by itself because step length parameters vary across different users and environments. The proposed R-Step Length Estimation algorithm however benefits from being fed with carefully configured parameter values thanks to the feedback loop between R-PDR and Map Matching. This unsupervised learning approach, referred as LL-Tracker, is described in detail in Sec. IV.

R-Heading Estimation: To estimate the heading of the pedestrians, most existing work assumes the knowledge of the position where the mobile device is attached, thereby making the assumption that the heading of the pedestrian is always consistent with the heading of the mobile devices [37], which is not true in real world settings. Zee [29] demonstrated that the heading can be estimated by observing the frequency spectrum of a typical walk. Although this is an interesting

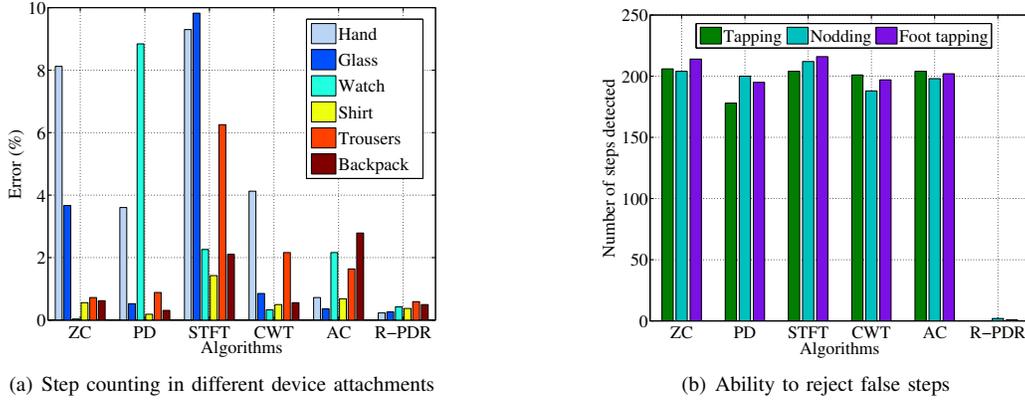


Fig. 5. Experiment results showing the step counting accuracy and ability to reject false steps of the R-PDR algorithm.

idea, it suffers from direction ambiguity. WalkCompass [4] samples the acceleration signal during the segment of a step with maximum deceleration. After removing noise and rotating samples to a stable coordinate system, it averages them to obtain the walking direction in 3D. It then projects the estimated direction to the horizontal *walk* plane. This approach works for various device placements, but is sensitive to noise as shown below.

In this paper, instead of averaging like WalkCompass, we fit the straight line that best matches the acceleration readings of a single step in the horizontal plane (X and Y) using simple least squares. We disambiguate between the two headings of the straight line using the first 50% of acceleration samples. If the residual error is small (Fig. 6(a) top and middle) we use the resulting estimate as the offset between device and user heading; otherwise (6(a) bottom), we use the most recent offset that was reliably estimated. Figs. 6(b) and 6(c) compares the proposed and competing techniques for tracking a user in a sports centre for 110 seconds. Notice that the proposed R-PDR approach can detect and avoid noisy heading estimates, thus yielding higher accuracy than the competing approaches.

The proposed heading estimation approach relies on being able to correctly rotate acceleration data to the earth coordinate system. This requires accurate device orientation estimates, which in turn depends on the quality of gyro data as discussed in Sec. III-A. Our approach to addressing the issue of long term drift of the gyro sensors, and further improving the accuracy of R-Heading Estimation, is to learn the sensor bias via the lifelong learning approach of LL-Tracker, as discussed in Sec. IV.

IV. LIFELONG LEARNING TRACKER

Background: Although the proposed R-PDR is robust to device, attachment and user motion variability, it does not fully address the problem that all PDR algorithms suffer from - the fact that over time, the estimated position drifts further away from the true position. A well established approach to correcting the long-term drift in PDR is to use map constraints, such as those encoded in floor plans, radio RSSI maps and magnetic distortion maps. This is a reasonable assumption given the wide availability of maps (esp. floor plans), and the intensive effort invested by major players, such as Google,

Microsoft, Apple and Qualcomm, in indoor mapping. The map matching layer can be implemented in a number of ways. Traditionally, it has used recursive Bayesian filters such as HMMs, Kalman Filters and Particle Filters, the latter being the most widely used approach due to its simplicity, ease of implementation and accuracy when a large number of particles is used. More recently, a novel approach based on undirected graphical models (specifically, conditional random fields) was introduced, and shown to be more lightweight and accurate than competing Bayesian filters [1].

In this paper, we argue that although the map matching layer has already a key role in indoor positioning, it has yet to achieve its full potential. It can be used not only for correcting the long-term drift in the PDR trajectory, but also for the novel purpose of lifelong learning. Similar to humans who become better at finding their whereabouts as they visit the same environment multiple times, so should an indoor positioning algorithm as it runs repeatedly in the same context. This idea is not entirely new in indoor positioning. An illustrative example is the implementation of map matching using a Simultaneous Localisation And Mapping approach, such as Wi-FiSLAM [38]. Bayesian filters are also adapted in map matching, e.g. Zee [29], [37], where the radio map or the magnetic distortion map [39] are learned after the trajectories have been matched to a given map. In principle, the more a user visits an environment, the more accurate the inferred radio maps become, and the better the performance of map matching. However, so far, the concept of lifelong learning has only been used to optimise the functionality of the map matching layer itself.

In stark contrast, in this paper, we suggest that information from the map matching layer could play a key role in optimising the performance of the R-PDR layer. It can be used to learn a variety of parameters such as those that depend on the device / device attachment (e.g. sensor bias), or on the user/environment (e.g. step length parameters). We refer to our proposed end-to-end lifelong learning tracking system as LL-Tracker. This consists of R-PDR, map matching and a feedback loop from map matching to R-PDR, as shown in Fig. 1.

Note that we have implemented MapCraft map matching algorithm as described in [1]. MapCraft can flexibly fuse various sensor observations such as inertial and WiFi sensor

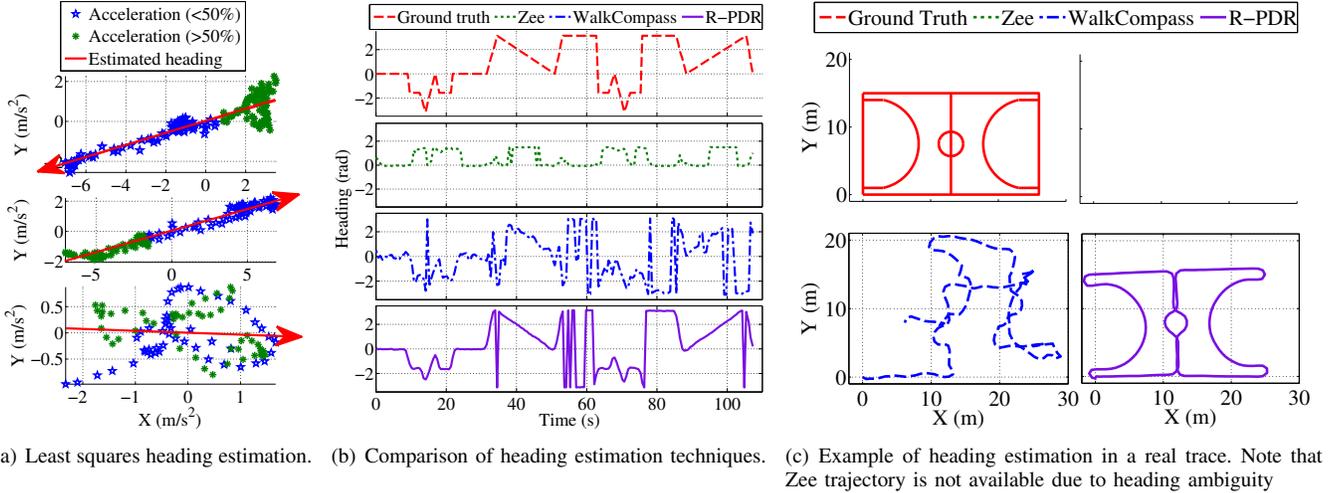


Fig. 6. Heading estimation: algorithm and illustrative example.

data, as well as leverage constraints drawn from a variety of maps, such as floor plans and radio maps. For details on how sensor observations and constraints are fused please refer to MapCraft [1]. In what follows, we first describe the parameter learning process, which is central to the feedback loop, and then provide a couple of illustrative examples.

Feedback Loop and Parameter Tuning: The accuracy of R-PDR depends on a number of parameters employed in its four steps. These are hard to tune manually because they may depend on the device, attachment, user and environment. To address this issue, we propose to use an unsupervised approach to learning R-PDR parameters. This approach is triggered every time the map constraints are such that there is only one and only one matched trajectory that can best support the raw trajectory with high probability. This guarantees the existence and uniqueness of the matched trajectory solution. When this condition is met, R-PDR parameter tuning becomes the following optimization problem:

$$\mathbf{x}^* = \operatorname{argmax}_{\mathbf{x}} \ln p(\mathcal{S}(\mathbf{x})|\mathcal{Z}(\mathbf{x})), \quad (6)$$

where \mathbf{x} is the parameter (or vector of parameters) of R-PDR (or indeed another PDR implementation) that requires tuning, $\mathcal{S}(\mathbf{x})$ is the matched trajectory and $\mathcal{Z}(\mathbf{x})$ is the raw trajectory fed from R-PDR to map matching. The key idea is that only when the estimated parameter values are the same as (or very close to) the correct ones, can we maximize the conditional probability of the matched trajectory given the raw trajectory $p(\mathcal{S}(\mathbf{x})|\mathcal{Z}(\mathbf{x}))$.

The solution to this optimization can be obtained with the expectation maximization (EM) approach. However, the soft EM approach does not work because the optimization of parameters \mathbf{x} in the M-step actually changes the state space of the model, which makes the E-step unable to evaluate the expectation in the next iteration [1]. Therefore, the hard EM approach, also known as Viterbi training is employed to solve the optimization problem [40].

Illustrative examples: We now show a couple of examples that demonstrate the effectiveness of the feedback loop between R-PDR and map matching for parameter tuning.

Learning of gyro bias: The learning of sensor bias, especially for low-cost sensors, is crucial to the performance of PDR algorithms. The motion sensor bias is the bottleneck that stops the inertial tracking from being widely used because the sensor bias is accumulated, leading to significant errors in the trajectory, especially when used in open space where no additional constraints can be applied. To make things worse, motion sensors also suffer from both time drift and thermal drift, which makes the bias varying with time and temperature. Therefore, the *lifelong* learning of sensor bias is essential to accurate PDR tracking. In practice all three motion sensors available, including accelerometer, magnetometer, and gyroscope, have drift or error. However, of all three motion sensors, the gyroscope plays the key role in terms of determining the accurate orientation of the device, which builds the foundation for long-term tracking. In addition, the calibration of gyroscope sensors is especially difficult compared to the calibration of accelerometers and magnetometers. Therefore, the learning of the gyroscope bias is an essential capability.

The key idea of learning the gyro bias is that only when the estimated heading is the same as (or very close to) the real heading can we maximize the conditional probability $p(\mathcal{S}(\mathbf{x})|\mathcal{Z}(\mathbf{x}))$ in Eqn. (6). Since the orientation of the device $\mathbf{q}(\boldsymbol{\omega}, \mathbf{b})$ can be easily derived from the gyro bias \mathbf{b} and the angular velocity $\boldsymbol{\omega}$ [15], the learning of gyro bias can be achieved by optimizing

$$\mathbf{b}^* = \operatorname{argmax}_{\mathbf{b}} \ln p(\mathcal{S}|\mathcal{Z}(\mathbf{q}(\boldsymbol{\omega}, \mathbf{b}))), \quad (7)$$

in which the formulation of $p(\mathcal{S}|\mathcal{Z}(\mathbf{q}(\boldsymbol{\omega}, \mathbf{b})))$ is discussed in detail in [1]. Since MapCraft uses conditional random fields to perform map matching, this optimization problem can be solved by iterating the forward-backward algorithm [1].

We have conducted experiments to test the effectiveness of the sensor bias learning. Fig. 13 shows the trajectory with one Nexus 5 device in an indoor basketball court without any floorplan or other map constraints. It is observed that the trajectory largely deviates from the ground truth without sensor bias learning. It is greatly improved in terms of heading estimation after the sensor bias of this device has

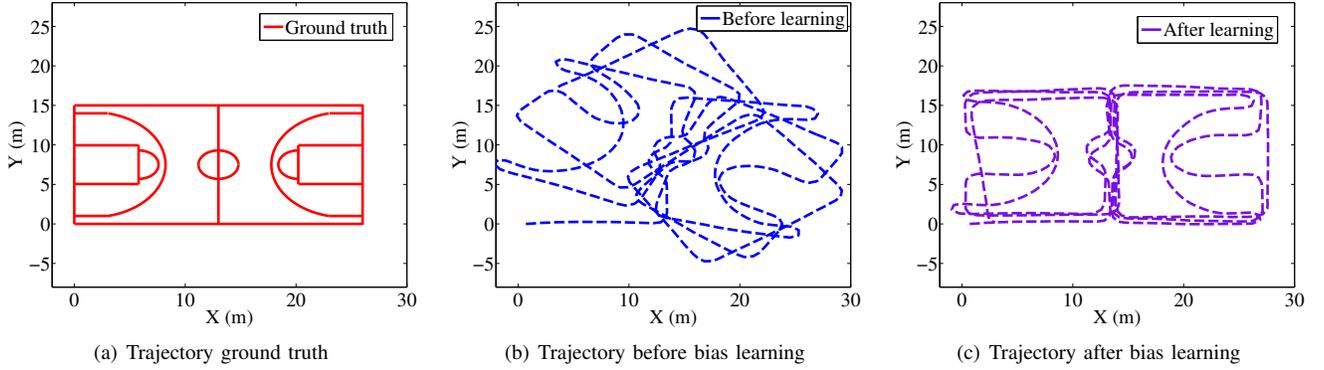


Fig. 7. Inertial trajectories of 0.5 km in a basketball court without any map constraints before and after sensor bias learning, showing an improvement in loop closing loop error from 17.3m to 2.3m, and demonstrating the importance and effectiveness of the learning algorithm.

been learned in a completely different (office) environment shown in Fig. 15(a). This example serves to show how lifelong learning can exploit one environment’s structure to benefit navigation in more challenging open space environments. The implications of cross-environment learning can extend to other users: the high tracking accuracy of a user with “lifelong learning experience” can be exploited to create an accurate radio map of the environment, which can in turn benefit other algorithms like WiFiSLAM [38] and users tracking with only radio maps.

Learning of step length constant: The second example shows how we can use the feedback loop between R-PDR and map matching to improve the performance of R-Step Length Estimation, the third step of R-PDR. In particular, we learn the step constant γ (Eqn. (5)) for different individuals because 1) the average step length plays a crucial role in the tracking accuracy due to the high consistency of step length in human walking patterns; and 2) the parameters α and β are very similar for different individuals in our experiments.

The step length constant learning is achieved in the same way as the learning of the gyro bias by replacing learning variable from \mathbf{b} in Eqn. (7) to the step length constant γ . In practice, \mathbf{b} and γ are learned simultaneously as a vector of variables.

We have conducted experiments in an office environment, a museum environment, and a market to show the effectiveness of the feedback loop on step length estimation. Notice in Fig. 8(a) that the step constants γ are very different for different pedestrians (U1 to U4), and very different even for the same pedestrian in different environments (office, museum, and market) as shown in Fig. 8(b). Fig. 8(c) shows that the step constant that maximizes the conditional probability $p(\mathbf{S}|\mathbf{Z})$ also minimizes the RMS error of tracking, allowing us to cope with user and environment variability.

Learning of Environment: In addition to the device or individual-specific parameters, LL-Tracker can also learn environment specific parameters including various radio and magnetic distortion maps. Traditionally, these maps can be obtained via either labor-intensive manual mapping [41] or SLAM [38]. Most SLAM approaches are unable to provide global sensible maps without external information or require the manual setting of landmarks [42] or fingerprinting points [38].

Compared with existing mapping techniques, the proposed LL-Tracker can globally map the environment-specific parameters with zero-effort from users. Based on the estimated trajectories from the map matching algorithm, we can then build the radio map or magnetic distortion map easily. These maps can later on be used by R-PDR and the map matching layer to improve positioning accuracy for other users.

Experiments have been conducted to learn the WiFi radio map and magnetic distortion map. Fig. 9(a) compares the radio map learned and the one manually built to test the accuracy of the learning technique. It is observed that the learned map is highly consistent with the one manually built. Figs. 9(b) and 9(c) show the magnitude and heading distortion of the magnetic field of the earth learned from an office building. It is observed that the magnetic field of the earth suffers from significant distortion in indoor environments. These distortions can then be used as unique signatures for localization, similar to radio maps. In addition, together with the radio map, the magnetic distortion map can also improve the heading estimation accuracy of PDR, especially the initial heading.

V. EVALUATION

Sites: LL-Tracker is evaluated and compared against competing approaches in three real-world settings with known floor plans: an office building ($65 \times 35m^2$), a museum ($109 \times 89m^2$), and a sports center hall ($30 \times 20m^2$). For the majority of the tests, the office building is used as it has the most distorted trajectories. Overall, 224 trajectories of average length over 400 m were collected over 30 days. Error is expressed in [m] RMS.

Participants: The variations between different people are taken into account by acquiring data from 15 people of different genders, heights, and ages. During the experiments, the subjects are mounted with several devices simultaneously in different parts of the body, typically hand, watch, glasses, shirt pocket, and trousers pocket. Then they walk anywhere in the building without planned routes, to realistically capture real pedestrian motion, rather than artificial, constant speed trajectories. They are told to move freely and may have different motion modes including walking, standing still, sitting down, bending to pick up something on the floor, etc.

Devices and Implementation: Different types of mobile devices and wearable sensors are involved in experiments,

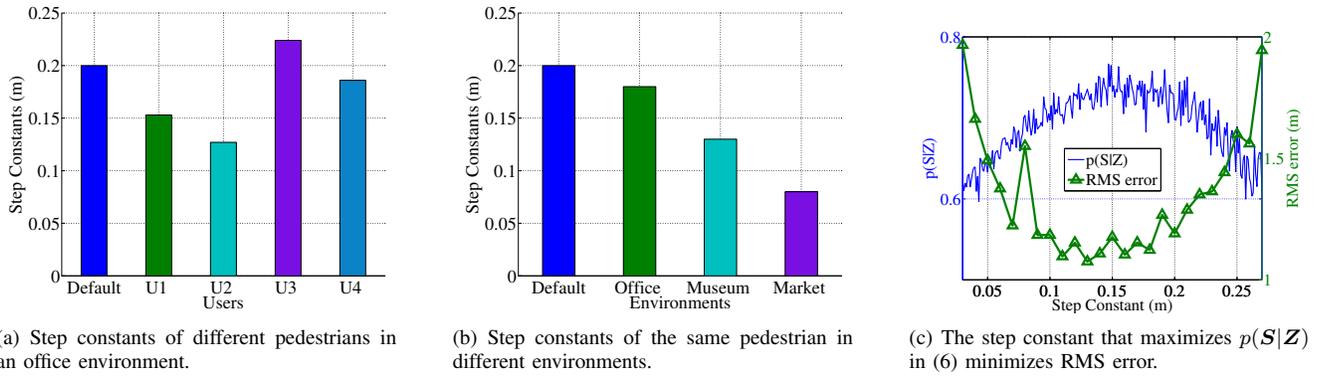


Fig. 8. Experiments results showing the importance and effectiveness of the proposed lifelong learning approach on step length estimation.

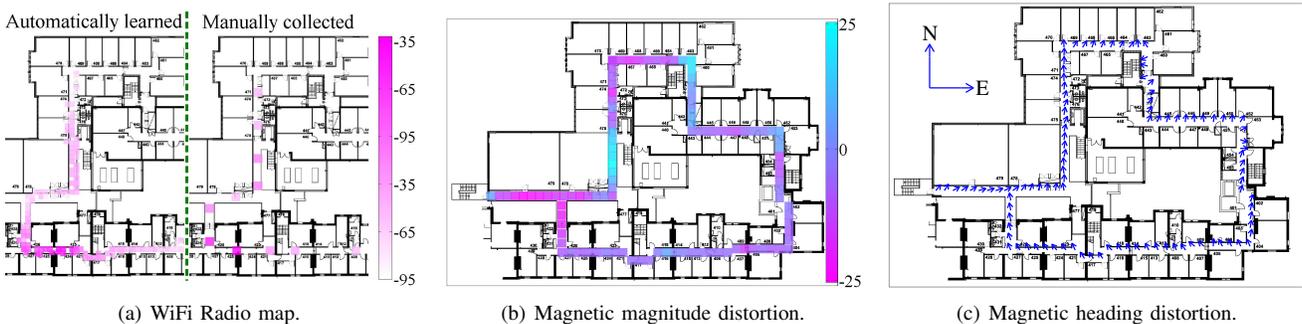


Fig. 9. Example experiments showing the effectiveness of the LL-Tracker in learning parameters of the environments.

including LG Nexus 4, LG Nexus 5, Asus Nexus 7, Samsung Galaxy S IV, x-IMU-1, and x-IMU-2. These devices differ greatly in terms of sensors, functionality and price. A snapshot of our application prototype has been shown in Fig. 1.

Ground Truth: To provide accurate ground truth, numbered labels were placed along corridors and within rooms on a 2 m grid. During the experiments, the test subject always held one camera at hand. These labels were filmed at the same time experiments were conducted. The time-synchronized video streams were then mapped to locations on the floorplan, and intermediate locations interpolated using footstep timing, also obtained from the video.

Proposed Algorithms: In this section we evaluate the performance of two proposed and two competing end-to-end positioning solutions. The proposed ones are LL-Tracker and R-Tracker. In particular, *LL-Tracker* combines R-PDR and map matching with lifelong learning, i.e. with the feedback loop from map matching to fine tune the parameters of R-PDR. On the other hand, *R-Tracker* consists of R-PDR and map matching without lifelong learning.

Competing Algorithms: Practical indoor positioning algorithms are required to be infrastructure-free or use existing infrastructures like WiFi and Bluetooth low energy (BLE). Existing practical algorithms fall into two categories: RF category where only WiFi/BLE are used and fusion category where inertial data and WiFi/BLE data are fused to perform positioning. Typical examples in the RF category are HORUS [43], RADAR [41], and EZ [44]. The state-of-the-art algorithms in the fusion category are MapCraft [1], Zee [29], UnLoc [42], WifiSLAM [38], and the algorithm in [37]. To evaluate the performance of LL-Tracker, we select the algo-

rithm that reports the best accuracy from each category, HORUS from RF category and MapCraft from fusion category, as competing approaches. In addition, we have implemented NAWalkCompass as a state of the art competing algorithm. NAWalkCompass combines the normalized autocorrelation [29] for step detection with the WalkCompass approach [4] for heading estimation.

HORUS [43] is a fingerprinting-based localization approach which uses an existing radio map, e.g. RSS-location pairs denoted with $\langle \mathbf{R}_i, \mathbf{L}_i \rangle$, $i = 1 \dots N$. To localize mobile devices, HORUS 1) takes a set of RSS \mathbf{r} , 2) estimates the likelihood of this set given RSS-location pairs in the radio map as

$$p(\mathbf{r}|\mathbf{R}_i) = \prod_{j=1}^M p(r_j = R_{i,j}|\mathbf{L}_i), \quad (8)$$

where M is the dimension of \mathbf{r} and $p(r_j = R_{i,j}|\mathbf{L}_i)$ is the distribution of the RSS at location \mathbf{L}_i (assumed to be Gaussian) and then 3) selects the location from the pair with the highest likelihood or a weighted mean of all locations based on the likelihood as the estimated location.

MapCraft [1] is a recently proposed lightweight indoor positioning algorithm, which employs conditional random fields for map matching. It has reported an RMS error of around 1 ~ 2 meters but, so far, it has been designed and implemented with the assumption of mobile devices held by the user in texting mode. To avoid penalising it for this reason, we compare the performance of the proposed algorithms using a variety of mobile device attachments, to that of MapCraft using a single device attachment (text mode).

The combination of Normalized Autocorrelation [29] and

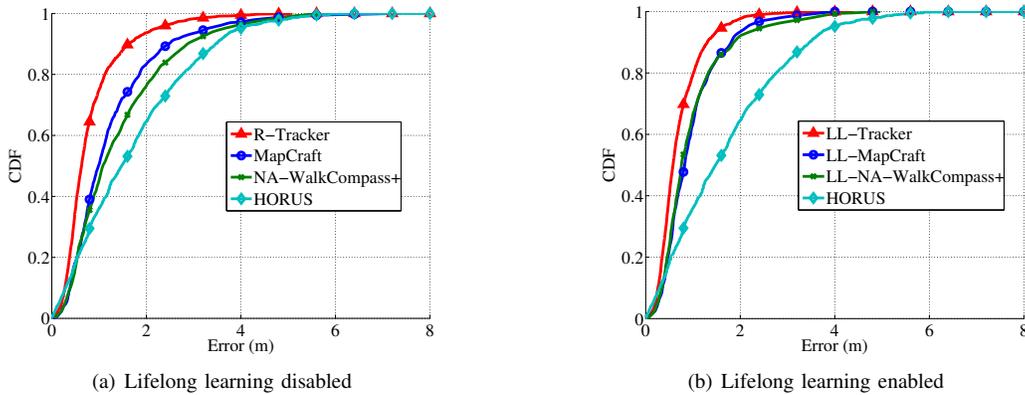


Fig. 10. Accuracy comparison of various tracking algorithms.

TABLE I
COMPARISON OF TRACKING ACCURACY (RMS [M]).

Lifelong Learning	PDR algorithm			HORUS
	MapCraft*	AC-WalkC+	R-PDR	
Disabled	1.61	1.78	1.07	4.62
Enabled	1.16	1.23	0.86	

* MapCraft implements naive PDR algorithm which only works with device held by hand in texting mode.

WalkCompass [4], called NA-WalkCompass, has also been implemented for comparison. The normalized autocorrelation has reported the best step detection results compared with other existing approaches, as discussed in Sec. III-B. However, since normalized autocorrelation cannot distinguish between real steps and other repetitive motions like nodding and tapping, we only evaluate the accuracy of NA-WalkCompass using trajectories without such repetitive motions. In addition, WalkCompass, the most practical existing heading estimation approach, still has two limitations. The first limitation lies in the fact that WalkCompass has no mechanism to correct the long term drift of inertial sensors, which indicates that the performance of WalkCompass can be significantly degraded in long-term tracking. Since we have some fairly long trajectories (>2 km) which WalkCompass cannot manage, we enhance WalkCompass with the proposed robust orientation estimation to make it capable of working for long term tracking (called NA-WalkCompass+). The other limitation is that WalkCompass does not work well with devices attached to the shirt pocket or head, as the authors also suggested. Therefore, we only evaluate the performance of this approach with devices at hand or in the pocket.

A. Accuracy

To evaluate the accuracy of the proposed tracking system, we have conducted experiments with 15 users, 5 different devices and 5 different attachments in the office and museum environments. The same IMU and WiFi data were fed into the proposed R-Tracker, LL-Tracker, and the competing approaches including MapCraft, NA-WalkCompass+, and HORUS. Fig. 10 compares the error cumulative distribution function of the four approaches without lifelong learning ability. It is observed from Fig. 10(a) that R-Tracker outperforms all other approaches, followed by MapCraft, NA-WalkCompass+, and lastly HORUS which is used as a baseline

for tracking accuracy. Note that all PDR algorithms compared in this section are fused with the map matching algorithm used in MapCraft to estimate the locations while HORUS only exploits knowledge of the radio map, but does not perform map matching.

The tracking accuracy of these algorithms is further improved when the lifelong learning capability is enabled, as shown in Fig. 10(b). The tracking errors of these combinations of approaches are compared in Table. I. It can be observed that the lifelong learning ability can improve the tracking accuracy of not only the proposed approach from 1.07m (R-Tracker) to 0.86m (LL-Tracker), but also other PDR-based approaches from 1.61m (MapCraft) to 1.16m (LL-MapCraft), and from 1.78m (AC-WalkCompass+) to 1.23m (LL-AC-WalkCompass+). The reason lies in the fact that the proposed lifelong learning algorithm optimizes generic PDR parameters, e.g. sensor bias, step constant, etc. instead of algorithm-specific parameters.

LL-Tracker outperforms competing approaches because it combines robust pedestrian dead reckoning (i.e. the ability to distinguish between real and fake steps) with lifelong learning of R-PDR parameters. Unlike LL-Tracker, the performance of both MapCraft and NA-WalkCompass+ can be easily degraded under non-ideal experiment conditions. For example, motions of the test subjects like occasionally shaking the device, opening/closing the door, etc. generate signals similar to steps and thus lead to over counting of steps using these algorithms, which thereby degrades the tracking accuracy.

B. Robustness

The next set of experiments is designed to validate the robustness of the proposed LL-Tracker as we change the experimental conditions.

User Variability: To evaluate the robustness over the user variability, 15 different people with different genders, ages from 20 to 55, and height from 150cm to 190cm are involved in the experiments. To minimize the impacts from other factors, we fixed the experiment site to be the office environment, the device to be Nexus 5 phone and attachment to be handheld only. Different step constant parameters are learned for different pedestrians. Fig. 11(a) shows some typical step constants for different people in the office environment

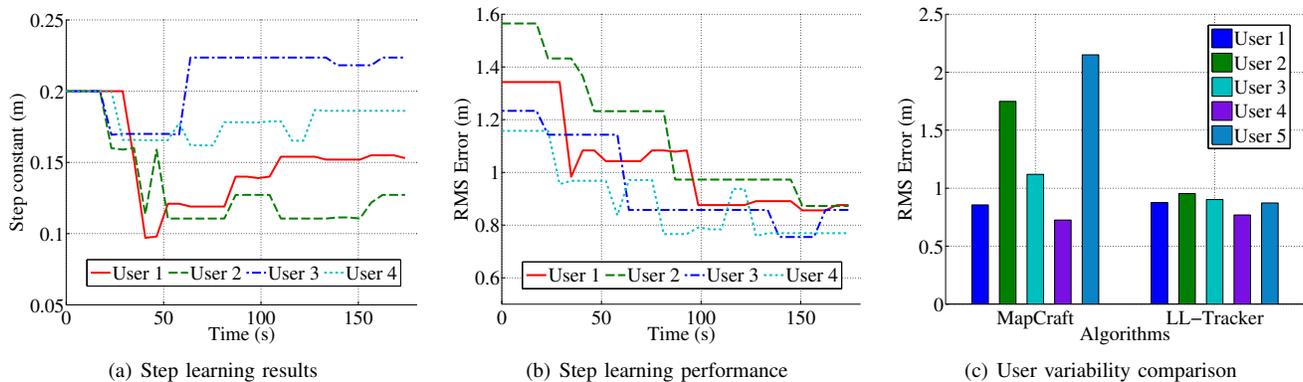


Fig. 11. Tracking accuracy of LL-Tracker with different devices, showing the effectiveness of the lifelong learning algorithms.

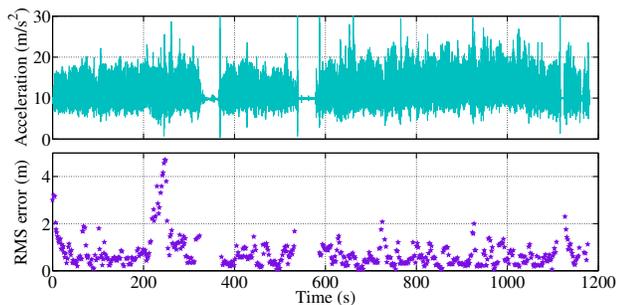


Fig. 12. Tracking accuracy with various behaviors and varying walking speed.

and the time taken to learn these parameters. Fig. 11(b) shows how the tracking performance has been improved with the convergence of the parameter learning. It is also shown in Fig. 11(c) that the proposed LL-Tracker has almost the same RMS tracking error regardless of different users while the counterpart, MapCraft, has individual specific tracking errors.

To further demonstrate the robustness of LL-Tracker, we have performed experiments taking into account various motion behaviors and time-varying walking speed. A set of behaviors including walking, sitting down, standing up, standing still, accelerating, decelerating, bending down to pick up things on the floor were performed by the pedestrian. We can observe some of these behaviors from the acceleration shown at the top of Fig. 12 while the bottom of Fig. 12 shows the tracking accuracy over the whole trajectory. Note that the big tracking error shortly after 200 seconds is due to the deliberate sudden significant change of walking speed and random shaking of the device. In this case the learning algorithm takes around 20 seconds to converge again.

Device Variability: To evaluate the robustness over the device variability, five different devices, including Nexus 5, Nexus 4, Samsung S4, X-IMU 1 and X-IMU2 are involved in the experiments. Again, we fixed other variables like the experiment site (office environment), users (User 1), and attachment (handheld). It is shown in Fig. 13(b) that the tracking accuracy for different devices are remarkably stable after each device has been used for 5 ~ 10 minutes. The trend is especially apparent for the devices used in open space like the sports centre where no map constraints are available to improve the tracking accuracy. It is observed that after walking for 0.5km in the basketball court, the RMS tracking error can be as high

as 15m without the bias learning algorithms while the error can decrease to around only 2m if the gyro bias has been learned in the office environments before it is used in the basketball court. In addition, in the office environment where the map matching can compensate for the sensor bias, the sensor bias learning still improves the tracking accuracy by over 20% (from 0.96m to 0.75m).

Attachment Variability: Five typical attachments are tested in this experiment to evaluate the robustness of LL-Tracker, including handheld, watch, glasses, shirt pocket, and trousers pocket. This experiment is also conducted by User 1 with Nexus 5 phone in the office environment. The traveled distance in this set of experiments is over 12 km in total. The test subject had different motion modes during the experiments including walking, standing still, sitting down, bending to pick up something on the floor, etc. It is observed from Fig. 13(c) that the RMS errors of LL-Tracker are extremely similar for different attachments. Please note that MapCraft only works for handheld attachment.

Environment Variability: We then evaluate the performance of LL-Tracker in a variety of environments, namely an office environment, a museum, and a sports center. All of these environments have different floor plans and methods of construction which affect the obtained sensor data. The museum is a multi-storey stone building with large, open spaces. Testing was conducted on the ground floor. The office environment (where the majority of the tests have been conducted) is a multi-storey office building with a stone and brick construction, reinforced with metal bars; testing was conducted on the fourth floor. The sports center is a big multi-functional hall where the experiments were conducted in the basketball court for the convenience of ground truth collection. In this set of experiments, data from all 15 experiment subjects, 5 different devices, and 5 different attachments are taken into account.

We have taken very complex and tortuous trajectories which typically are the weakness of inertial tracking systems, due to drift and the absence of absolute anchor measurements. Fig. 15 shows a couple of illustrative examples of how LL-Tracker succeeds in accurately tracking a pedestrian through the office and museum environments. The cumulative distribution functions of these environments are shown in Fig. 14(a). The RMS errors are 0.86m in the office and 0.90m in the museum. Note that LL-Tracker starts tuning R-PDR parameters after

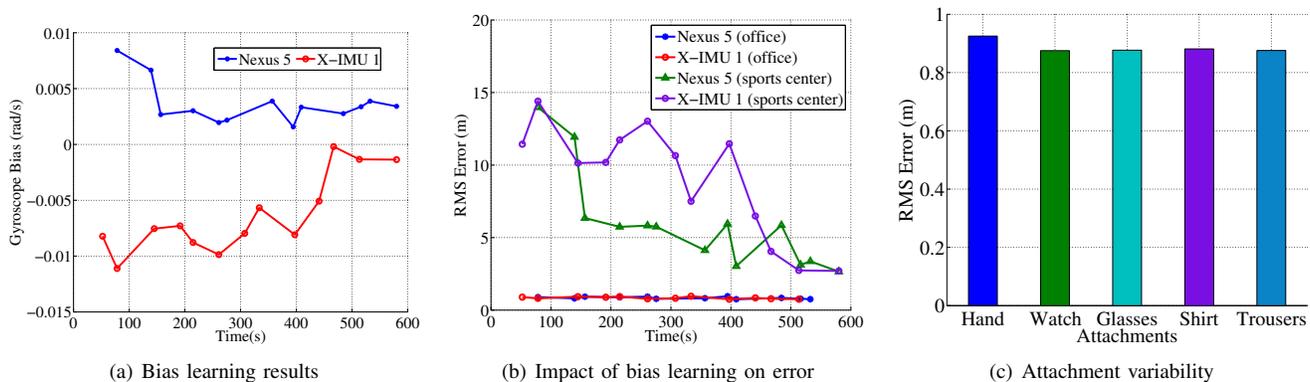


Fig. 13. Tracking accuracy of LL-Tracker with different devices and attachments, showing the effectiveness of the lifelong bias learning algorithms. The bias was learned in the office environment for a period of time, and then tested both in the same office and a completely different sports centre environment.

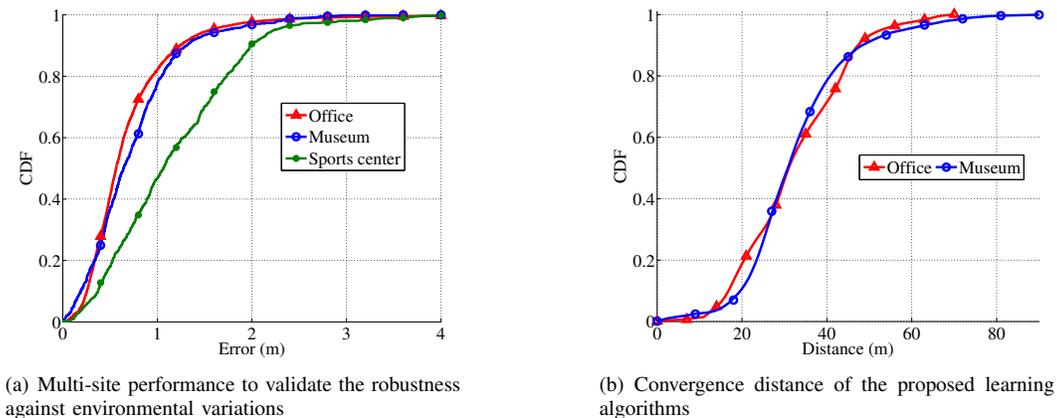


Fig. 14. The performance of proposed LL-Tracker in multi-sites.

it converges to a single trajectory on the map with high probability; the CDF of the convergence distance in the office and museum environments is presented in Fig. 14(b).

The performance in open space is always the limitation of PDR algorithms because the errors can be accumulated quickly within a short time when no external constraints, e.g. the floor plan, can be utilized to correct the drift, especially with low-cost IMU sensors. In order to understand the limitation of the proposed R-PDR algorithm, we have tested the performance of various devices in a sports center with results reported in Fig. 14(a). The devices used in this experiment have been calibrated with the feedback loop from map matching layer in the office environment shown in Fig. 15(b). It is observed that the proposed LL-Tracker only has a RMS error of around 1.27 meters even in open space (basketball court) with trajectories as long as 0.5 km. An example trajectory in this environment can be found in Fig. 7(c).

The automatic learning of parameters is crucial to the performance of PDR-based tracking systems. We have observed from the extensive experiments that PDR parameters especially the gyro bias have much greater impact on the tracking performance in environments with less constraints like the sports center than in more constrained environments like the office building and the museum where the floor plan can provide sufficient constraints. The gyro bias plays a crucial role in the tracking performance because the proposed system heavily relies on the accurate orientation estimation

of the device. In comparison, the impact of learning and tuning the step constant offset is less significant when the map matching algorithm is applied in environments with rich floorplan constraints.

VI. CONCLUSION AND FUTURE WORK

In this paper, we have demonstrated the merits of a novel indoor positioning system LL-Tracker, consisting of a robust PDR and a lifelong learning component. On the one hand, the new motion classification scheme and the use of general motion principles to counteract sensor noise have enabled us to move away from artificial and controlled settings, and develop a system that works reliably across a diverse range of real world settings. On the other hand, the lifelong learning aspect of LL-Tracker is a step change in optimising positioning systems: it is a simple yet general idea that can be applied to any positioning system implementation. Unlike existing work, where lifelong learning has only been used to optimize the map matching layer, in this paper we generalise this idea to optimize the entire system, reaping the fruits of cross-layer optimisation. LL-Tracker achieves sub meter accuracy more than 80% of the time, compared to the state of the art MapCraft algorithm, even when LL-Tracker is tested in unconstrained settings and MapCraft uses a single known device attachment. We have further shown that LL-Tracker is robust not only to device and its attachment, but also to

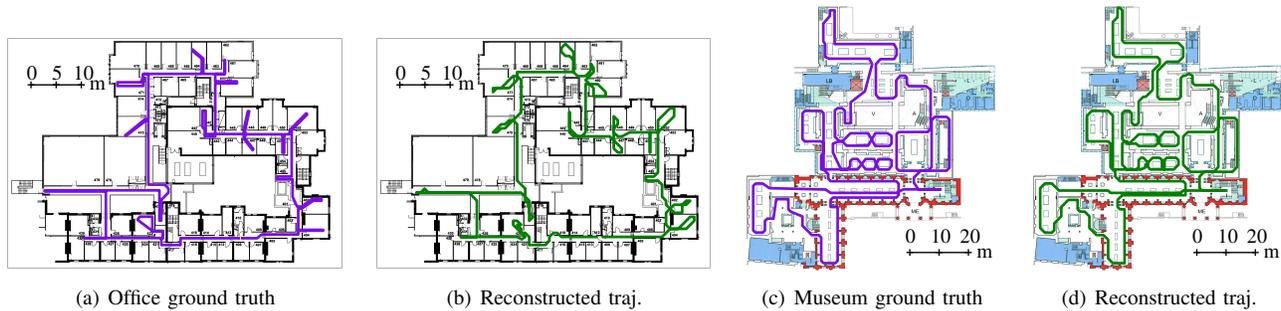


Fig. 15. Experiments in the office and museum sites, showing the ground truth and reconstructed trajectories.

user and environment variability. It learns PDR parameters in an unsupervised manner and thus requires zero user effort. Notably, we have illustrated cases in which its operation in one environment can boost its performance in a new and more challenging environment. LL-Tracker has widespread application, as it can be used with a wide variety of sensors and different types of maps. In the future, we plan to learn a wider variety of parameters, and to study in depth the sensitivity of each parameter to context (user, device, attachment, or environment) variability. Our next step is to incorporate algorithms that detect context changes, and select appropriate parameter settings.

ACKNOWLEDGMENT

The authors would like to thank the anonymous reviewers for their comments and suggestions to improve the paper. They also acknowledge the support of the EPSRC through grants EP/L00416X/1.

REFERENCES

- [1] Z. Xiao, H. Wen, A. Markham, and N. Trigoni, "Lightweight map matching for indoor localization using conditional random fields," in *IPSN'14*, pp. 131–142, 2014.
- [2] J. Yang, "Toward physical activity diary: motion recognition using simple acceleration features with mobile phones," in *Proc. 1st Int. workshop Interactive multimedia for consumer electronics*, pp. 1–9, 2009.
- [3] J. S. Wang, C. W. Lin, Y. T. Yang, and Y. J. Ho, "Walking pattern classification and walking distance estimation algorithms using gait phase information," *IEEE Trans. Biomedical Engineer.*, vol. 59, no. 10, pp. 2884–2892, 2012.
- [4] N. Roy, H. Wang, and R. Roy Choudhury, "I am a smartphone and i can tell my user's walking direction," in *MobiSys'14*, pp. 329–342, 2014.
- [5] L. Fang, P. Antsaklis, L. Montestruque, M. B. McMickell, M. Lemmon, Y. Sun, and H. Fang, "Design of a wireless assisted pedestrian dead reckoning system—the NavMote experience," *IEEE Trans. Instrument. Measure.*, vol. 54, no. 6, pp. 2342–2358, 2005.
- [6] S. H. Shin, C. G. Park, and J. W. Kim, "Adaptive step length estimation algorithm using low-cost MEMS inertial sensors," in *SAS'07*, pp. 1–5, 2007.
- [7] I. Bylemans, M. Weyn, and M. Klepal, "Mobile Phone-Based Displacement Estimation for Opportunistic Localisation Systems," in *UbiComp'09*, pp. 113–118, Oct. 2009.
- [8] V. Renaudin, M. Susi, and G. Lachapelle, "Step length estimation using handheld inertial sensors," *Sensors*, vol. 12, pp. 8507–8525, Jan. 2012.
- [9] J. Chung, M. Donahoe, C. Schmandt, I.-J. Kim, P. Razavai, and M. Wiseman, "Indoor location sensing using geo-magnetism," in *MobiSys'11*, p. 141, 2011.
- [10] V. Renaudin, M. Susi, and G. Lachapelle, "Step length estimation using handheld inertial sensors," *Sensors*, vol. 12, no. 7, pp. 8507–8525, 2012.
- [11] M. Susi, V. Renaudin, and G. Lachapelle, "Motion Mode Recognition and Step Detection Algorithms for Mobile Phone Users," *Sensors*, vol. 13, no. 2, pp. 1539–1562, 2013.
- [12] Y. S. Suh, "Orientation estimation using a quaternion-based indirect Kalman filter with adaptive estimation of external acceleration," *IEEE Trans. Instrument. Measure.*, vol. 59, no. 12, pp. 3296–3305, 2010.
- [13] X. Yun and E. R. Bachmann, "Design, Implementation, and Experimental Results of a Quaternion-Based Kalman Filter for Human Body Motion Tracking," *IEEE Trans. Robotics*, vol. 22, pp. 1216–1227, Dec. 2006.
- [14] B. Huyghe, J. Doutreloigne, and J. Vanfleteren, "3D orientation tracking based on unscented Kalman filtering of accelerometer and magnetometer data," in *SAS'09*, pp. 148 – 152, 2009.
- [15] T. Harada, T. Mori, and T. Sato, "Development of a Tiny Orientation Estimation Device to Operate under Motion and Magnetic Disturbance," *The Int. J. Robotics Res.*, vol. 26, pp. 547–559, June 2007.
- [16] D. Mizell, "Using gravity to estimate accelerometer orientation," in *ISWC'03*, pp. 252–253, 2003.
- [17] H. Lu, J. Yang, Z. Liu, N. D. Lane, T. Choudhury, and A. T. Campbell, "The Jigsaw continuous sensing engine for mobile phone applications," in *SenSys'10*, pp. 71–84, ACM Press, 2010.
- [18] S. Hemminki, P. Nurmi, and S. Tarkoma, "Accelerometer-based transportation mode detection on smartphones," in *Sensys'13*, pp. 1–14, 2013.
- [19] P. Zhou, M. Li, and G. Shen, "Use it free: Instantly knowing your phone attitude," in *MobiCom'14*, (New York, NY, USA), pp. 605–616, 2014.
- [20] J. Rose and J. G. Gamble, *Human Walking*. Baltimore, PA, USA: Lippincott, Williams and Wilkins, 3rd ed., 2006.
- [21] J. Kim, H. Jang, D. Hwang, and C. Park, "A step, stride and heading determination for the pedestrian navigation system," *J. Global Pos. Syst.*, vol. 3, no. 1, pp. 273–279, 2004.
- [22] S. Beauregard and H. Haas, "Pedestrian dead reckoning: A basis for personal positioning," in *WPNC'06*, pp. 27–36, 2006.
- [23] S. H. Shin, M. S. Lee, and P. C. G., "Pedestrian dead reckoning system with phone location awareness algorithm," in *PLANS'10*, pp. 97–101, 2010.
- [24] P. Goyal, V. J. Ribeiro, H. Saran, and A. Kumar, "Strap-down Pedestrian Dead-Reckoning system," in *IPIN'11*, pp. 1–7, Sept. 2011.
- [25] N. Ravi, N. Dandekar, P. Mysore, and M. Littman, "Activity recognition from accelerometer data," in *AAAI'05*, pp. 1541–1546, 2005.
- [26] J.-g. Park, A. Patel, D. Curtis, S. Teller, and J. Ledlie, "Online pose classification and walking speed estimation using handheld devices," in *UbiComp'12*, pp. 1–10, 2012.
- [27] A. Brajdic and R. Harle, "Walk detection and step counting on unconstrained smartphones," in *UbiComp'13*, pp. 225–234, ACM Press, 2013.
- [28] H. Ying, C. Silex, A. Schnitzer, S. Leonhardt, and M. Schiek, "Automatic step detection in the accelerometer signal," in *BSN'07*, pp. 80–85, 2007.
- [29] A. Rai and K. Chintalapudi, "Zee: Zero-effort crowdsourcing for indoor localization," in *MobiCom'12*, pp. 1–12, 2012.
- [30] M. Alzantot and M. Youssef, "UPTIME: Ubiquitous pedestrian tracking using mobile phones," in *WCNC'12*, pp. 3204–3209, 2012.
- [31] R. Harle, "A Survey of Indoor Inertial Positioning Systems for Pedestrians," *IEEE Commun. Surveys Tutorials*, vol. 15, pp. 1281–1293, Jan. 2013.
- [32] P. Barralon, N. Vuillerme, and N. Noury, "Walk detection with a kinematic sensor: frequency and wavelet comparison," in *Proc. Ann. Int. Conf. IEEE Eng. Med. Bio. Soc.*, vol. 1, pp. 1711–4, Jan. 2006.
- [33] M. N. Nyan, F. E. H. Tay, K. H. W. Seah, and Y. Y. Sitoh, "Classification of gait patterns in the time-frequency domain," *J. Biomech*, vol. 39, pp. 2647–56, Jan. 2006.
- [34] E. Foxlin, "Pedestrian tracking with shoe-mounted inertial sensors," *IEEE Comput. Graph. App.*, vol. 25, no. 6, pp. 38–46, 2005.
- [35] R. Faragher and R. Harle, "SmartSLAM: an efficient smartphone in-

door positioning system exploiting machine learning and opportunistic sensing,” in *ION GNSS+’13*, pp. 1–14, 2013.

- [36] W. Chen, R. Chen, Y. Chen, H. Kuusniemi, and J. Wang, “An effective Pedestrian Dead Reckoning algorithm using a unified heading error model,” in *PLANS’10*, pp. 340–347, May 2010.
- [37] 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 *UbiComp’12*, pp. 421–430, 2012.
- [38] J. Huang, D. Millman, M. Quigley, D. Stavens, S. Thrun, and A. Agarwal, “Efficient, generalized indoor WiFi GraphSLAM,” in *ICRA’11*, pp. 1038–1043, May 2011.
- [39] M. H. Afzal, V. Renaudin, and G. Lachapelle, “Assessment of indoor magnetic field anomalies using multiple magnetometers,” in *ION GNSS+’10*, pp. 21–24, 2010.
- [40] H. Trinh, *A machine learning approach to recovery of scene geometry from images*. Ph.d thesis, Toyota Technological Institute at Chicago, 2010.
- [41] P. Bahl and V. N. Padmanabhan, “RADAR : An in-building RF-based user location and tracking system,” in *INFOCOM’00*, pp. 775–784, 2000.
- [42] H. Wang, S. Sen, A. Elgohary, M. Farid, M. Youssef, and R. R. Choudhury, “No need to war-drive: unsupervised indoor localization,” in *MobiSys’12*, pp. 197–210, 2012.
- [43] M. Youssef and A. Agrawala, “The Horus WLAN Location Determination System,” in *MobiSys’05*, pp. 205–218, 2005.
- [44] K. Chintalapudi, A. Padmanabha, and V. Padmanabhan, “Indoor localization without the pain,” in *MobiCom’10*, pp. 173–184, 2010.



Zhuoling Xiao is currently a PhD candidate in Computer Science at University of Oxford. His research interests focus on sensor networks, including localization, communication and coordination protocols for networked sensor nodes, and machine learning techniques for sensor networks and localization.



Dr. Hongkai Wen received his PhD in Computer Science from the University of Oxford. He is currently a postdoctoral researcher at the Department of Computer Science, University of Oxford. His research interests are in sensor networks, localization and navigation, and probabilistic machine learning.



Dr. Andrew Markham received the Bachelor’s (2004) and PhD (2008) degrees in Electrical Engineering from the University of Cape Town, South Africa. He is currently an Associate Professor in the Department of Computer Science, at the University of Oxford, working in the Sensor Networks Group. His research interests include low power sensing, embedded systems and magneto-inductive techniques for positioning and communication.



Dr. Niki Trigoni is an Associate Professor at the Department of Computer Science, University of Oxford. She obtained her PhD at the University of Cambridge (2001), became a postdoctoral researcher at Cornell University (2002-2004), and a Lecturer at Birkbeck College (2004-2007). Since she moved to Oxford in 2007, she established the Sensor Networks Group, and has conducted research in communication, localization and in-network processing algorithms for sensor networks. Her recent and ongoing projects span a wide variety of sensor networks applications, including indoor/underground localization, wildlife sensing, road traffic monitoring, autonomous (aerial and ground) vehicles, and sensor networks for industrial processes.