Map matching and heuristic elimination of gyro drift for personal navigation systems in GPS-denied conditions

2011 Meas. Sci. Technol. 22 025205

(http://iopscience.iop.org/0957-0233/22/2/025205)

IOP PUBLISHING MEASUREMENT SCIENCE AND TECHNOLOGY

Meas. Sci. Technol. **22** (2011) 025205 (12pp) doi:10.1088/0957-0233/22/2/025205

Map matching and heuristic elimination of gyro drift for personal navigation systems in GPS-denied conditions

Priyanka Aggarwal[1], David Thomas, Lauro Ojeda and Johann Borenstein

The University of Michigan, 2260 Hayward Street, Ann Arbor, MI 48109, USA

E-mail: aggarwalpp@gmail.com davidlt@umich.edu lojeda@umich.edu johannb@umich.edu

Received 23 May 2010, in final form 9 December 2010
Published 18 January 2011
Online at stacks.iop.org/MST/22/025205

Abstract

This paper introduces a method for the substantial reduction of heading errors in inertial navigation systems used under GPS-denied conditions. Presumably, the method is applicable for both vehicle-based and personal navigation systems, but experiments were performed only with a personal navigation system called 'personal dead reckoning' (PDR). In order to work under GPS-denied conditions, the PDR system uses a foot-mounted inertial measurement unit (IMU). However, gyro drift in this IMU can cause large heading errors after just a few minutes of walking. To reduce these errors, the map-matched heuristic drift elimination (MAPHDE) method was developed, which estimates gyro drift errors by comparing IMU-derived heading to the direction of the nearest street segment in a database of street maps. A heuristic component in this method provides tolerance to short deviations from walking *along* the street, such as when crossing streets or intersections. MAPHDE keeps heading errors almost at zero, and, as a result, position errors are dramatically reduced. In this paper, MAPHDE was used in a variety of outdoor walks, without any use of GPS. This paper explains the MAPHDE method in detail and presents experimental results.

Keywords: map-matching techniques, heuristic drift elimination, personal navigation systems, GPS, inertial sensors

(Some figures in this article are in colour only in the electronic version)

1. Introduction

Pedestrian tracking is the process of determining and maintaining positional information for a person traveling on foot. Pedestrian tracking systems (PTSs) are useful for tracking movements of military personnel, finding and rescuing incapacitated first responders, or for location-aware computing and personal navigation assistance [1]. For most PTSs, GPS is the primary source for obtaining position and velocity data. Under ideal conditions, GPS provides long-term accurate and absolute measurements. Outdoors, where there is a clear line of sight to four or more satellites, GPS provides location with accuracies ranging from tens of meters to tens of centimeters, depending on the type of GPS receiver [14]. However, GPS is often unavailable in urban areas, where the urban canyon effect may reduce the number of visible satellites. Even when GPS is available, it is often less accurate because of multipath effects. Alternative approaches for absolute PTSs require active radiation sources such as infrared light [16], ultrasound [17], magnetic fields [18], ultrawideband (UWB) [19] or other radio frequencies [20]. Common to these methods is that they require initial setup of the corresponding sources. Many applications exist in which the initial setup time is not feasible (e.g., firefighting or military) or where the installation of such infrastructure is too costly (e.g., city blocks).

Another approach is to use light emitting and capturing sources such as cameras or light detection and ranging (LIDAR) systems [21]. LIDAR suffers from disturbances when the user walks among other pedestrians, as would be the case in downtown areas of large cities. Another technology

[1] Author to whom any correspondence should be addressed.

that uses cameras and/or LIDAR is simultaneous localization and mapping (SLAM), which is based on building a map within an unknown environment (without *a priori* knowledge) or updating a map within a known environment (with *a priori* knowledge from a given map) while at the same time keeping track of the current location. SLAM systems may accumulate errors over time and can fail when used on a busy street with pedestrians walking by. A recent development [22] uses head-mounted laser range and inertial sensors to localize a person in indoor environments, without use of any *a priori* building or floor map data. All LIDAR-based technologies have the following limitations in common: have active source of radiation which might reveal the military users' location to the enemy and the equipment pose is altered when firefighters crawl on hands and knees for rescue mission. A problem common to LIDAR and computer vision systems is that they do not work well in smoke-filled environments, making them unsuitable for firefighters.

There are a few other radio frequency (RF)-based technologies, such as those using cellular telephones (exploiting time delays) or triangulation methods based on wireless networks for obtaining navigation parameters. The main limitation of cellular technologies is that they are not accurate enough for most pedestrian tracking purposes. Wireless local area networks (WLANs) can be used for indoor navigation, where the location fix is determined by measuring signal strengths from several access points [21]. However, like GPS, this method suffers from multipath and fading effects, and often pre-mapping of signal strength is required. Other methods may include active badges or optical tracking systems that employ specific markers installed in and around the buildings and can assist in detecting the person's current location [23].

An entirely different approach is the use of relative sensor modalities, predominantly IMUs. IMUs do not require any external infrastructure and do not actively radiate. The problem with IMUs is that the position and heading estimates derived from the accelerometers and gyros of the IMU develop unbounded errors. Moreover and besides their astronomical cost, very high-end IMUs that develop errors only slowly are typically too large and heavy to be worn by a person, especially warfighters and first responders who already carry a lot of heavy gear. Smaller and lighter IMUs, typically using micro-electro-mechanical systems (MEMS) technology, develop large position and heading errors quickly. A common approach to making MEMS-based IMUs feasible in PTSs applications is therefore to find ways that allow the bounding of heading and position errors.

The main components of an IMU-based PTS algorithm are footfall detection, step-length along with altitude estimation, and heading estimation [2–6, 20]. Footfall detection may be accomplished by examining accelerometer or gyroscope variances after pre-processing (such as noise reduction by averaging) [2, 3, 7]. The estimation of step-length can be accomplished either by GPS updates [3] (when GPS is at least temporarily available), as a function of step period, acceleration magnitude or acceleration variance [6, 10], by direct inertial sensor integration [1, 11, 12], or by other advanced techniques [15].

One key distinction of IMU-based PTSs is whether the IMU is located on the foot or elsewhere on the body. Location of the IMU on the body makes system integration more practical, but existing systems all struggle with the problem of estimating distance traveled, especially when users apply different gaits, as well as when walking backward or sideways. Foot-mounted IMUs are more cumbersome to use because there are limited options for placing the IMU on or in the user's footwear, and typically a cable connection to a body-worn computer is required. However, the great advantage of foot-mounted IMUs is that they allow compensation for accelerometer drift and thus accurate direct step-length measurements [7] as will be explained in section 2. In most IMU-based systems, altitude is not measured by the IMU but rather by barometric pressure sensors [20].

In IMU-based PTSs, the heading or direction of motion is most commonly obtained from the gyroscopes of the IMU. However, in MEMS-based systems bias drift of the gyros is significant and if left uncorrected can completely mask the navigation solution [1, 11, 12]. Therefore, alternative equipment or additional data sources are required to achieve reasonable accuracy. Since heading errors from inertial sensors can grow without bound, reducing heading errors can dramatically improve position accuracy. Another method for obtaining heading information is by using a magnetic compass [2, 10]. A magnetic compass offers absolute heading information by measuring the earth's magnetic field. However, these measurements can be easily distorted by nearby steel structures or electric fields. For these reasons, magnetometers generally do not work reliably enough inside buildings and work only marginally in urban environments [13].

Map-matching algorithms [24, 25] can also be used for correcting heading errors in PTS. Map matching is the process of comparing personal tracking data with the digital map of streets to match the pedestrian's navigation data to the street segment on which the pedestrian is walking. There are several map-matching algorithms such as geometric point-to-point, geometric point-to-curve and geometric curve-to-curve methods. These methods and their shortcomings will be explained in section 3.2. This paper proposes a modified geometric point-to-curve method to overcome some of the limitations of these existing map-matching algorithms.

The work was motivated by the goal of creating a PTS that is capable of tracking walking persons in GPS-denied urban environments. The focus was on mitigating heading errors by means of the so-called map-matched heuristic drift elimination method that was developed under this study. This method has two components: (1) obtaining the direction of the nearest street segment through map matching and (2) correcting the walker's estimated heading based on the inertial sensors and the present best estimate of the heading calculated by the previously developed heuristic drift elimination (HDE) technique.

It should be noted that the term 'drift elimination' as used in this paper is meant to be taken figuratively, not literally. Of course, our method cannot correct gyro drift at the core, namely inside the gyro. A more precise but stylistically cumbersome name for our method would be 'heuristics-based elimination of

(*a*)

(*b*)

Figure 1. The foot-mounted IMU of the PDR system has two mounting options: (*a*) side-mounted IMU (the IMU itself is covered by the beige-colored thermal insulation), (*b*) in-heel IMU with temperature-controlled, shock-resistant housing.

the ill-effects of drift and of other slow-changing error sources on heading estimation'. In the interest of brevity, we chose the shorter term 'heuristic drift elimination'.

The remainder of this paper is structured as follows. Section 2 describes the earlier developed PDR system. Section 3 describes the TIGER database and OpenStreetMaps (OSMs), where map data are freely available, along with previous map-matching techniques. The proposed MAPHDE method is described in section 4 and results are presented in section 5.

2. The personal dead reckoning system

In earlier work, the Mobile Robotics Laboratory at the University of Michigan developed the personal dead reckoning (PDR) system [7–9, 11]. Using only an IMU for all measurements, the PDR system has a zero-radiation signature, i.e. it does not emit any signals. This makes the system 'invisible' to sensors in hostile environments and immune to interference or jamming.

2.1. Components of the PDR system

The PDR system uses an IMU strapped to the side of the user's boot or embedded in the heel of the user's boot, as shown in figure 1. The side-mounted IMU can be transferred among different users while the in-heel version better protects the IMU from damage and cannot be dislocated easily. The PDR system's computations are performed on a PC-104 computer that is located inside a belt pack, together with batteries and support electronics. The MEMS-based IMU used in the PDR system is the nano-IMU ('nIMU' in short), made by Memsense. Some key specifications for the nIMU are listed in table 1.

Table 1. Key specifications of the Memsense nIMU.

Size (mm)	45 × 23 × 13
Weight (g)	15
Bandwidth (Hz)	75
Gyroscope	
Range (deg s^{-1})	±1200
Angle random walk (deg h$^{-1/2}$)	4.2
Bias drift (deg h^{-1})	80
Accelerometer	
Range (g)	±10

The computer runs the Linux operating system patched with a real-time extension. An IMU-based position estimation system combines two functional components: the estimate of distance traveled and the estimate of heading. In the PDR system, the accuracy of both components is predominantly affected by bias drift ('drift', in short). This is especially true if a relatively low-performance MEMS-based IMU is used. Drift rates for both accelerometers and gyroscopes in a MEMS-based IMU are several orders of magnitude higher than what is found in high-grade aviation IMUs. Of course, the cost of high-grade IMUs is also one or two orders of magnitude higher than that of the nIMU in the PDR system. However, the reason for using a low-grade IMU in the PDR system is not just cost. Rather, in order to embed the IMU in the heel of a regular firefighter or military-style boot, the device must be very small. Because of this size limitation, the only suitable IMU technology is that of MEMS. Another limiting factor is the fact that the peak accelerations and rates of turn of the foot, even at normal walking speed, are significantly higher than those found on the torso of a person. This limits the choice of suitable IMUs to less than a handful of models that offer the large dynamic range required by that particular IMU location.

A foot-mounted IMU makes wiring more difficult, requires greater dynamic ranges of the IMU, and may add difficulties in heading estimation. However, there is a compelling reason for choosing this mounting location after all: it is the only way to reset drift of the accelerometers almost as frequently as once every second and thereby allows the accurate determination of step-length, as explained below.

2.2. Estimate of distance traveled

In the PDR system, the estimate of distance traveled is based on the accelerometers of an IMU. It is well known that measured accelerations must be integrated twice to yield distance traveled. This double integration turns even small amounts of drift into large position errors. In order to overcome this limitation, a technique known as zero velocity updates (ZUPTs) is used. ZUPT is a well-established technique for estimating and counteracting drift in inertial sensors. However, ZUPT is often inconvenient because it can be applied only while the sensor has zero velocity. A foot-mounted IMU, on the other hand, lends itself perfectly to the application of ZUPT because during walking and many other modes of legged motion, the sole of the instrumented boot has zero velocity during parts of a phase of walking called 'midstance.' To exploit this fact, the PDR system examines the incoming data from the IMU and flags the instance of minimal foot motion during the midstance phase. This flagged instance of presumed zero velocity is called 'footfall' in the context of this paper. The ZUPT method is then implemented by comparing the accelerometer-derived velocity of the foot to the velocity of the foot during footfall. Since at footfall the velocity should be zero, any accelerometer-derived non-zero velocity estimate at the instance of footfall is considered to be the result of drift and is compensated for.

Another helpful aspect of walking is that footfalls of the instrumented foot occur with every second step, and that is almost as frequently as once per second. This assures that accelerometer drift is compensated for once per second, thereby not allowing significant drift and position errors to accumulate. As a result, the PDR system estimates distance traveled consistently with errors of less than 1% of distance traveled, on almost all terrains, except for ice, sand, or other non-solid surfaces. The ZUPT method and its application in the PDR system are explained in more detail in the authors' earlier papers [7–9, 11].

2.3. Estimate of change in heading

The PDR system estimates changes of heading from the IMU's gyro data. However, bias drift in the gyros of a MEMS-based IMU is on the order of tens to hundreds of degrees per hour (about 80° h^{-1} in the case of the nIMU). Unfortunately, a method 'similar to the ZUPT approach' that works so well for the PDR system's accelerometers cannot be used to compensate for the gyro drift. The main difficulty is that under realistic walk conditions the shoe heading continues to change during the footfall; therefore, it is 'non-zero' and a method such as ZUPT would not immediately be applicable.

In earlier work [8] the authors introduced a method, called 'heuristic drift reduction' (HDR) that partially compensates for gyro drift by exploiting the fact that pedestrians often walk along approximately in straight lines. This is mainly because pedestrian routes (e.g., sidewalks in urban centers) are often straight and because straight lines are the shortest way to get from point A to B. The method of [8] reduced average heading errors up to fivefold.

In subsequent work [9], the authors introduced a significantly more effective method, called 'heuristic drift elimination'. This method is applicable only in indoor environments in which corridors and walls are straight and intersect at right angles. In practice, this seemingly severe limitation is met by the vast majority of all buildings. In such indoor environments, HDE effectively compensates for gyro drift and other slow-changing heading errors.

Similarly, the key hypothesis or heuristic assumption of the work in this paper is that when pedestrians walk along streets, they do so predominantly in the general direction of these streets. Furthermore, the proposed method presumes that these streets are mapped with good accuracy and that the maps can be obtained in electronic form. Based on these assumptions, the heuristic method proposed in this paper compares the momentary heading of a walker based on gyro estimates to the known heading of the street segment that the walker is currently walking on. Any discrepancy is assumed to be due to gyro drift and is compensated for.

However, these heuristic assumptions do not hold true all the time. In reality, pedestrians do not walk on perfectly straight lines. Pedestrians cross streets at angles, avoid obstacles and pick occasional shortcuts. Nonetheless, this paper argues that despite these local aberrations, streets and walkways channel the motion of pedestrians so that on *average*, the heading of a street segment is a good predictor for the average heading of the pedestrian along that segment. The challenge is to define a practical method that uses the known heading of street segments to correct gyro drift, while being tolerant to these local, temporary deviations. The remainder of this paper introduces one such method.

3. Map matching

As a source for digital street maps, the freely available OSM [27] database is used. This database claims a positional accuracy range of 7.6 m or better.

3.1. The OSM database

The database of roads used in this study was obtained from the OSM database, which is mainly populated by the Master Address File/Topologically Integrated Geographic Encoding and Referencing (MAF/TIGER®) database [26]. This OSM database is a single XML file that is hierarchically composed of ways (including streets, paths, rivers, lakes, legal boundaries, etc), covering the entire United States.

The OSM database is collaboratively updated by volunteers and is more inclusive and more up-to-date than the TIGER database. In the OSM database, straight street segments are represented by two points, while curved street segments are represented by multiple, rather densely spaced points, as shown in figures 2 and 3. The OSM database contains digital vector data describing geographic features but it does not include graphic images for visualization.

In preparation for this study, the latitude and longitude of street centerline points were extracted from OSM database and stored in the flash memory of the PDR system. These data were parsed ahead of time to populate a SQLite database with the required data. In doing so, streets were split into segments of a maximum length of 100 m, to significantly cut down the number of viable segments.

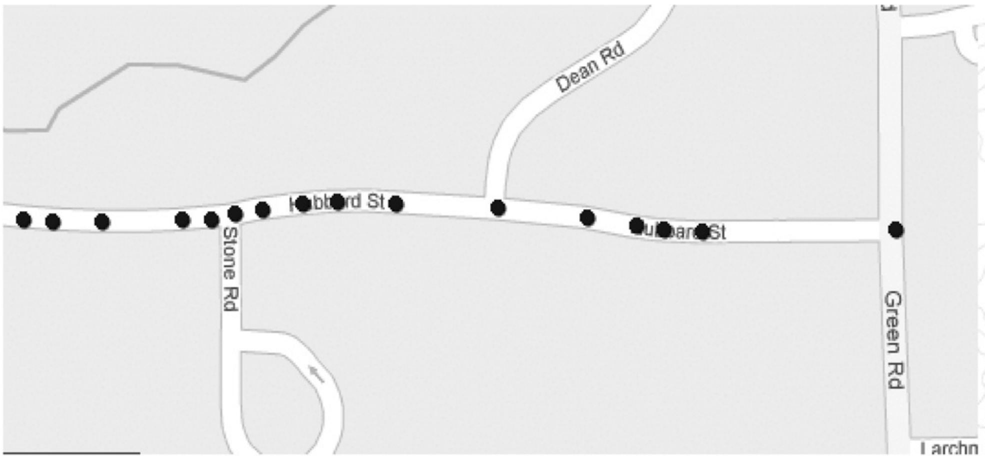

Figure 2. Typical frequency of points used in the OSM database representation of curved street segments (overlaid over a conventional map, for visualization).

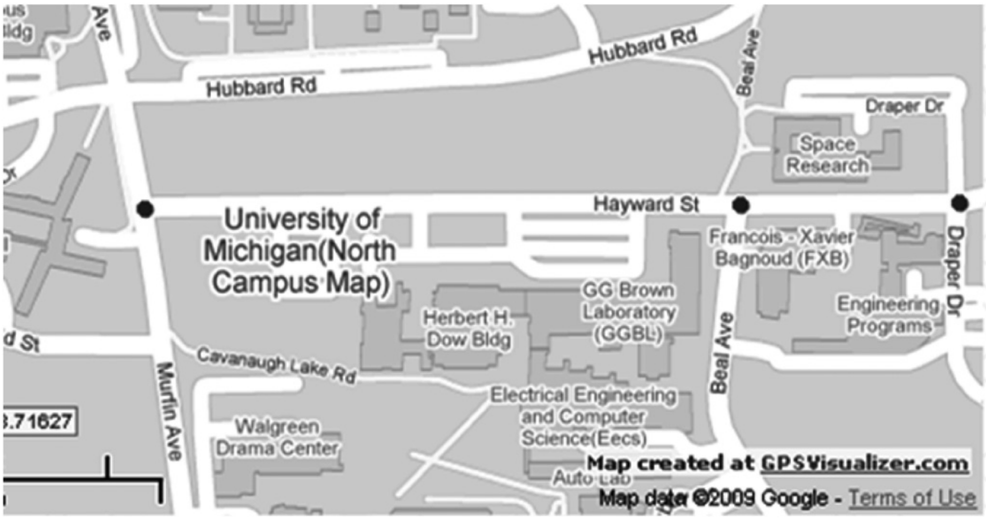

Figure 3. Typical OSM database representation of straight street segments, using only two points (overlaid over a conventional map, for visualization).

3.2. Map-matching techniques and integration methods developed by others

A map-matching algorithm identifies the street segment that the user travels along. There are several map-matching algorithms and some of them are briefly discussed here [28].

3.2.1. Geometric point-to-point matching. In this approach, the closest node in the street network database to the user is obtained. This is the simplest, easiest to implement, and fastest method, but the accuracy of the solution depends on the way the street network is digitized. Streets with a greater number of nodes are more likely to be matched with the user.

3.2.2. Geometric point-to-curve method. With this method a street segment is identified that is closest to the user, rather than the street nodes. From geometry, the minimum distance between a point and a line is the perpendicular distance, and this distance can be readily computed. One problem with this method is illustrated in figure 4, where the trajectory is nearest

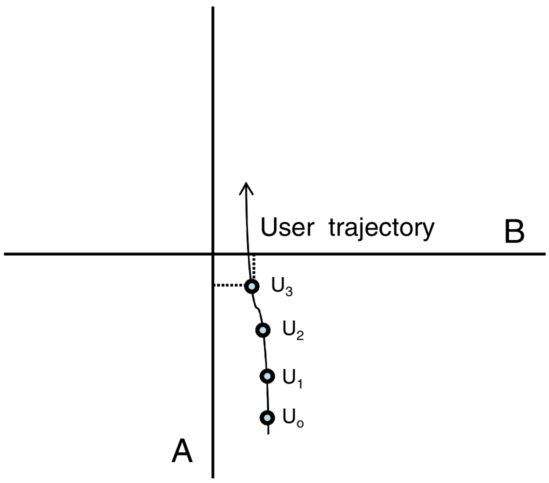

Figure 4. Error in the geometric point-to-curve method near intersection.

to street A but at point U_3 it is equally distant to streets A and B. In this situation the above-explained algorithm fails to select the correct street segment. Hence, this approach does not work well near intersections where different streets can be equidistant from the walker. Furthermore, the obtained perpendicular point may be unstable, oscillating back and forth between two closely spaced parallel streets common in urban environment [28–31]. A single reading may be matched with several street segments, thereby creating an ambiguity in the solution and causes the user to drift from one parallel street to another.

3.2.3. Geometric curve-to-curve matching. With this method the previous m positions of the user are matched with the street curve and the nearest street curve is selected as the solution. This method is very sensitive to outliers and there is no good way to compare curves of different lengths without incorporation of GPS data. On connecting GPS data points, a sequence of piecewise linear curves is formed which is then matched with a set of possible street segments [25]. Generally, the selected street segment will have the smallest norm distance to the GPS curve. However, if GPS data are not accurate, this matching method will produce a wrong match. Furthermore, there should be a sufficient number of GPS readings to form a linear curve for the user's trajectory.

In this study, a modified geometric point-to-curve method was implemented for identifying the nearest street segment to the walker. In addition, only the direction of the nearest street was matched to the walker's estimated heading, since the algorithm does not attempt to correct the walker's position directly.

There are many possible ways of optimally combining the map information with other sources of position information, such as IMU. One such method utilizes the maximum *a posteriori* (MAP) estimator to optimally translate raw position measurements of a walker onto the street network [32]. A MAP position estimate is obtained by solving the equation

$$x = \arg\max[p(x)p(y|x)] \qquad (1)$$

where x is the position estimate of the walker with probability $p(x)$, y is the available noisy measurement and $p(y|x)$ represents the likelihood function.

In cases where the street is a long straight line, it is modeled by the centerline. Curved streets are modeled by a sequence of piecewise straight-line segments, where each piece is treated as a separate street and a locally optimum position estimate is determined for that piece. However, it is possible that the solution obtained by the unbiased MAP estimator is not unique as the likelihood function may have multiple local maxima and a wrong street may be identified. The resolution of this problem requires other sources of position information to be identified and incorporated into the estimation process.

One such source is to include information contained in the walker's trajectory such that a probability is associated with each street. This probability represents the likelihood of the walker being on that particular street. The easily way to accomplish this is by means of a Kalman filter (KF) as it associates a probability with each street such that larger errors produce smaller probabilities of being the selected street segment [3]. However, this approach is not practical because a KF assumes state variables and noises to have a Gaussian distribution, which is not always true in a real scenario [3].

Other possible methods are to use fuzzy logic-based map-matching and probabilistic algorithms using inertial sensors [32, 33]. In [32], a set of eight rules is defined for obtaining position using the map-matching algorithm, inertial sensors, and GPS data. However, in certain cases error growth can completely mask the true solution. A confidence region in the shape of an ellipse or rectangle is defined around the navigation sensor output, for the probabilistic approach [33]. This error or confidence region is then superimposed on a map database to obtain the correct street segment. Even so, additional knowledge may be essential to identify the correct street segment from a number of possibilities.

The MAPHDE method uses a modified map-matching technique, but in conjunction with the aforementioned HDE engine, as will be explained in the following section. The HDE technique is capable of fully compensating for all slow-changing errors including gyro drift, thereby providing near-zero heading errors at steady state. In the context of this study, the steady state is typically reached within a few seconds of walking straight along a straight street segment. HDE needs only four pieces of information to function: distance traveled per footfall, time elapsed since the last footfall, change in heading since the last footfall, and the direction of the nearest street segment [5]. Furthermore, MAPHDE does not need any models or model parameters, as would be required in the case of KFs. In its simplest form, MAPHDE requires the tuning of just three parameters, the gain i_c, the distance of the walker from the nearest street segment $d_{\min}$, and a threshold ψ_{thres}, for the difference between estimated heading from HDE and the present best estimate of the direction of the nearest street segment. Furthermore, HDE takes just a few lines of C-code to implement. The MAPHDE method is explained next.

4. MAPHDE method

This paper presents a novel method for correcting heading errors in position tracking systems based on the proposed MAPHDE method. The required real-time input for the proposed method is the rate of rotation about the Z-axis or change in heading, step-length, time interval between subsequent footfalls, and initial starting position and heading. Of course, this system also requires its map database to be populated with pertinent map data. Generally, on 3D terrain, or with the foot-mounted sensor of the PDR system, a full 6-DOF (degree-of-freedom) IMU is required to estimate heading. Here the notion of a 'virtual Z-axis gyro' is introduced to express the walker's rate of rotation about the navigation Z-axis. The PDR system uses the quaternion attitude representation. The output of the virtual Z-axis gyro may be affected by errors such as offset bias, run-to-run bias, in-run bias, scale factor error, misalignment error, sensor drifts, and random noises, etc, which are difficult to identify and model or compensate. However, the HDE method treats the combination of all these sources of error as a single error term which overall behaves as drift and the HDE method estimates and compensates for it in real-time. This is the main advantage of the HDE technique as it does not require individual error sources to be identified and quantified.

When a person equipped with a virtual Z-axis gyro is walking straightforward, the output of that gyro should be exactly zero throughout the walk. However, due to a variety of slow-changing and random errors (such as bias drift, sensitivity to linear accelerations, scale factor nonlinearity, and with integrated white noise), the actual output is off by some small value ε. For simplicity, all these errors are collectively called 'drift' in this paper. The total travel distance is divided into smaller intervals, i.e. from footfall to footfall, as defined in section 2.2.

Due to the drift error ε, in each interval the rate of turn computed based on the Z-axis gyro is

$$\omega_{\text{meas},i} = \omega_{\text{true},i} + \varepsilon_i \tag{2}$$

where $\omega_{\text{meas},i}$ is the rate of turn around the Z-axis during the sampling interval i, $\omega_{\text{true},i}$ is true but the unknown rate of turn around the Z-axis during the sampling interval i and ε_i is the sum of all unknown errors during the sampling interval i.

The change of heading in the sampling interval i, denoted as $\Delta\psi_i$, is computed by numerically integrating $\omega_{\text{meas},i}$:

$$\Delta\psi_i = \omega_{\text{meas},i} T_i = (\omega_{\text{true},i} + \varepsilon_i) T_i \tag{3}$$

where T_i is the duration of the time interval i in seconds. As T_i is the time between footfall $i-1$ and footfall i, the new estimated heading is

$$\psi_i = \psi_{i-1} + \Delta\psi_i. \tag{4}$$

It is apparent from equations (2) through (4) that the estimated heading, ψ_i, represents the true heading plus the accumulated sum of all heading errors, $\psi_{e,i}$.

4.1. Error correction approaches

A brute force approach to correcting heading errors from known street segment directions is to assume that any difference between the gyro-derived heading and the street segment direction is due to gyro-induced errors. Consequently, one can correct the gyro-derived heading by simply replacing it with the appropriate street segment direction. This brute force method will fail as soon as the walker actually deviates from the direction of the street, for example, to cross the street or avoid an obstacle.

In order to allow for the walker's temporary deviations from the direction of the street, a more subtle approach is needed. For example, instead of completely replacing the gyro-derived heading by the street direction in every sampling interval, one could add a small correction to the gyro-derived heading. This correction could be proportional to the difference between street and gyro-derived heading. However, this approach would still react too strongly to actual deviations. For example, when crossing a street the difference between the direction of the street and the gyro-derived direction can be as large as 90°.

A more deviation-tolerant approach is to add a repetitive and small, *constant-magnitude* correction to the gyro-derived heading. As long as constant-magnitude heading corrections accumulate faster than drift-induced heading errors, this approach can effectively neutralize drift-induced heading errors. It is also apparent from this discussion that the proposed approach will not react strongly to temporary deviations, because corrections in response to temporary deviations cannot accumulate significantly.

The following section formalizes this proposed approach in the form of a feedback control system. Specifically, it will be shown how the proposed algorithm

- models $\psi_{e,i}$ as a disturbance in a feedback control system;
- implements a method for accumulating small, constant magnitude corrections by means of a so-called Binary I-controller;
- remains largely insensitive to changes in ψ that have large amplitudes but short duration.

4.2. MAPHDE algorithm

The basic MAPHDE algorithm functions essentially like a feedback control system. This is different from most other measuring systems, where signals pass from the sensor to the instrument's output in open-loop fashion. Figure 5 shows a block diagram of the feedback control system for the MAPHDE algorithm.

The explanation of the feedback control system starts with the signal from the gyro, which is modeled as a disturbance in the block diagram of figure 5. Suppose the user is walking straight, along a straight section of a linear map feature (for simplicity, it is assumed that all linear map features are streets, but the proposed algorithm works equally well with floor plans of tunnels or buildings). Immediately after a footfall, the measured rate of turn is integrated to yield the change in heading, $\Delta\psi_i$. $\Delta\psi_i$ has two components, the true but unknown change of heading, $\Delta\psi_{\text{true},i}$, and the change of heading error, $\Delta\psi_{e,i}$. Next, $\Delta\psi_i$ is added to ψ_{i-1} and to the output of the binary I-controller, which is explained later in this section. Initially, the output of the I-controller is zero. The label 'z_{-}^{1}' in the feedback loop is the common notation for a pure delay of one sampling interval. After the first iteration, when $i > 1$, the control loop can be closed by comparing ψ_{i-1} to the direction of the closest map feature, $\psi_{\text{map},i-1}$. The difference between them is the error signal E.

$$E_i = \psi_{\text{map},i-1} - \psi_{i-1}. \tag{5}$$

Unlike conventional integral (I) or proportional-integral (PI) controllers, the binary I-controller is designed not to respond at all to the magnitude of E; rather, it only responds to the sign of E. If E is positive (i.e. the measured direction of walking points to the right of the nearest street segment), then a counter (called 'integrator' or 'I') is incremented by a small fixed increment, i_c, where i_c is the gain of the controller and a tunable parameter in MAPHDE. If E is negative, then I is decremented by i_c. In this fashion, *repeated* instances of E having the same sign will result in repeated increments or decrements of I by i_c. The reason for using a *binary* I-controller is that E_i can differ from zero by tens of degrees, for example, when the walker is turning. In that case a conventional I-controller would not work well, since it would respond

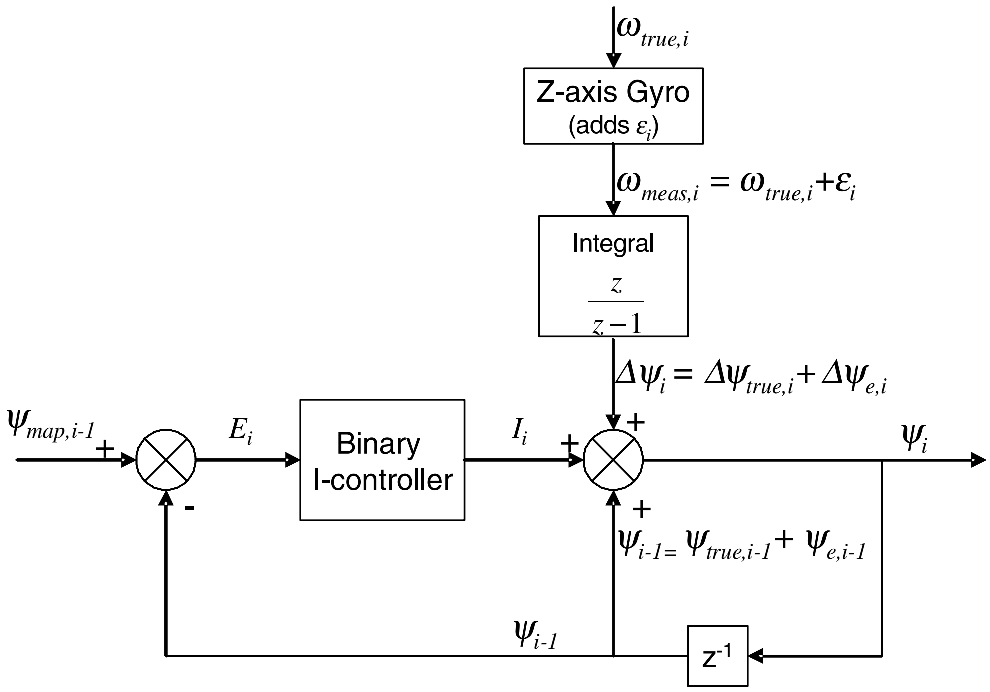

Figure 5. The HDE algorithm viewed as a feedback control system. The block labeled 'Binary I-controller' is explained in the narrative.

strongly to the large value of E, even though a large E is not an indication for a large accumulated heading error. The proposed binary I-controller, on the other hand, is insensitive to the magnitude of E. Rather, the controller reacts, slowly, to E having the same sign *persistently*.

As established by equation (5), if $\psi_{i-1} > \psi_{\text{map},i-1}$ (and thus, $E < 0$), then ψ_{i-1} is immediately to the left of $\psi_{\text{map},i-1}$, and if $\psi_{i-1} < \psi_{\text{map},i-1}$, then ψ_{i-1} is immediately to the right of $\psi_{\text{map},i-1}$. During straight-line walking along a street, a heading to the left of $\psi_{\text{map},i-1}$ suggests that the accumulated heading error, $\psi_{e,i-1}$, had a positive value. To counteract this error, the binary I-controller reduces the content of the integrator by a small value, i_c. Conversely, if $\psi_{i-1} < \psi_{\text{map},i-1}$, then the integrator is increased by i_c. Now the binary I-controller can be formulated,

$$I_i = \begin{cases} I_{i-1} - i_c & \text{for} \quad E < 0 \\ I_{i-1} & \text{for} \quad E = 0 \\ I_{i-1} + i_c & \text{for} \quad E > 0 \end{cases} \qquad (6a)$$

where i_c is the fixed increment, also considered the gain of the binary I-controller in units of degrees.

An alternative way of writing equation (6a) is

$$I_i = I_{i-1} + \text{SIGN}(\psi_{\text{map},i-1} - \psi_{i-1})i_c \qquad (6b)$$

where SIGN() is a programming function that determines the sign of a number. SIGN(x) returns '1' if x is positive, '0' if $x = 0$, and '-1' if x is negative. The next element in the control loop adds the controller output to the raw measurement to give the corrected heading:

$$\psi_i = \psi_{i-1} + \Delta\psi_i + I_i. \qquad (7)$$

4.3. Discussion on the MAPHDE algorithm

As long as i_c is of greater magnitude than typical increments in gyro errors, the integrator continuously tracks the heading error (but with an opposite sign), just like the integrator in a conventional I-controller tracks slow-changing disturbances. At steady state, i.e. when walking straight along a straight street, the content of the integrator will oscillate about the accumulated heading error, $\psi_{e,i}$, with an amplitude of i_c, which is typically smaller than 1°. The strength of the MAPHDE algorithm lies in the fact that it can readily tolerate short deviations from walking in the same direction as a street. For example, when dodging other pedestrians on a busy city street, when crossing a street, when turning at an intersection, or when walking along a curving street, short deviations from the nominal direction of the nearest street segment are incurred. As a result, MAPHDE may increment or decrement the integrator in the wrong direction—but only for the duration of the maneuver. However, once motion is aligned again with the direction of the nearest street segment, it takes just a few steps for MAPHDE to correct the incurred error. The street direction is obtained by the modified map-matching algorithm as explained in the following section.

It is quite possible that other controllers or filters, such as a finite impulse response (FIR) filter or a KF, can be designed and tuned to act in a way similar to the binary I-controller. However, the intention in this study was not to compare the performance of different methods. Rather, the intention was to demonstrate that heading errors due to gyro drift and other slow-changing errors in a position tracking system can be compensated for and effectively reduced to near-zero (at steady state) from map data. The proposed MAPHDE control system seems advantageous because it does not require the explicit

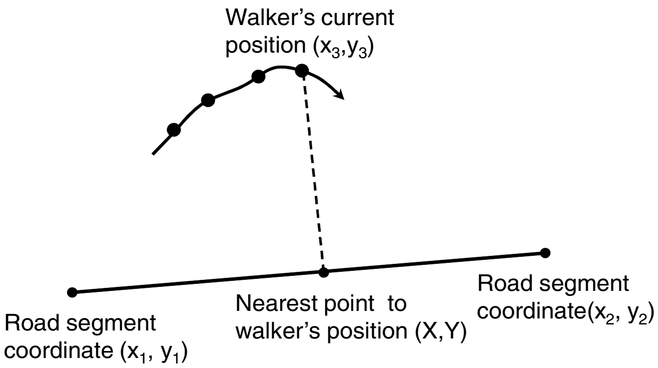

Figure 6. Coordinates of the street segment, the walker and the perpendicular point.

definition of a model, nor does it require the controlled process to be linear.

4.4. Modified map-matching algorithm

This map-matching algorithm searches the on-board database for street segments that meet two conditions.

(1) The segment lies within a pre-defined distance from the walker ($d_{\min}$).
(2) The segment has a direction that differs by no more than ψ_{thres} from the walker's heading.

Once these conditions are met, the perpendicular distance between the selected street segments is calculated as follows:

$$X = \frac{m(y_3 - y_1 + mx_1) + x_3}{m^2 + 1} \tag{8}$$

$$Y = y_1 + m(X - x_1) \tag{9}$$

$$D = \sqrt{(X - x_3)^2 + (Y - y_3)^2} \tag{10}$$

where (x_1, y_1), (x_2, y_2) are the coordinates of the selected street segment as illustrated in figure 6; (X,Y) is the perpendicular point coordinate on the street segment; (x_3, y_3) is the walker's present position and m is the slope of the line formed by two $((x_1, y_1), (x_2, y_2))$ coordinates.

This map-matching algorithm selects the street segment closest to the user's position and corrects the estimated heading based on the selected street direction. The estimate of heading errors is adjusted based on a comparison between the estimated heading from the HDE engine and the present best estimate of the direction of the nearest street segment. This adjustment occurs only when the estimated heading and the direction of the street agree to a great extent, i.e. when $|E_i| < \psi_{\text{thres}}$. ψ_{thres} is a tunable parameter for the algorithm, typically between 20° and 45°.

In the modified map-matching algorithm, only the walker's heading is matched to the nearest street direction, without matching the user's position to the street coordinates. Also, at intersections, all HDE corrections are temporarily suspended by setting i_c to zero as the confidence drops off.

An intersection is detected by the sudden changes in the minimum distance of the user from the nearest street. As the user approaches an intersection, the nearest street segment changes and so does the distance to the nearest street segment. In figure 7, the continuous marked yellow (light-colored) region indicates the intersection where the modified map-matching algorithm was temporary suspended.

4.5. Initialization

The proposed method requires that users follow an initialization procedure that has two components: specification of (1) an initial position and (2) an initial heading. Initialization is a significant problem on its own and this paper does not attempt to offer a comprehensive, commercialization-ready solution. Nonetheless, a few possibilities are discussed below.

4.5.1. Initial position. For a commercial position tracking system for police or other first responders, a possible solution

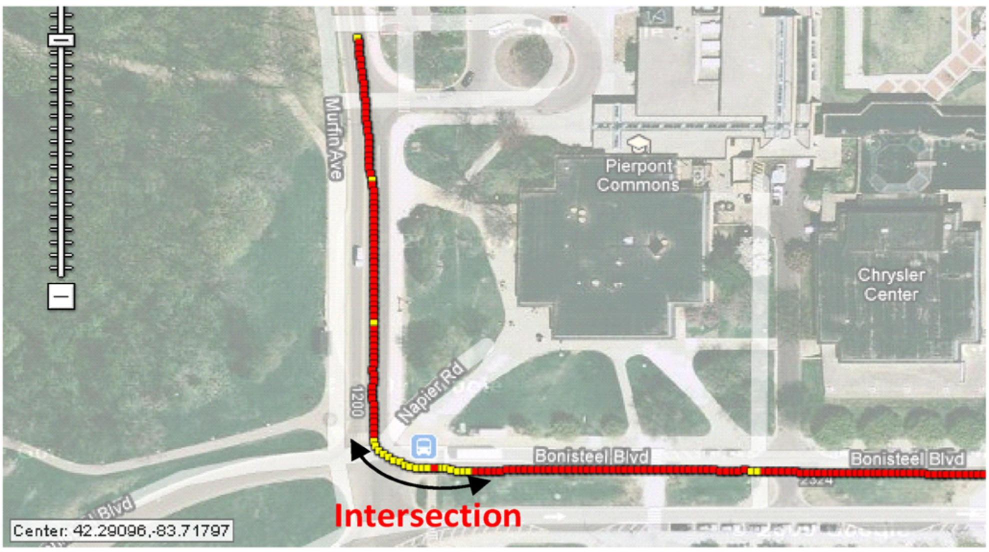

Figure 7. Suspension of the map-matching algorithm near an intersection, indicated by the light-yellow dots.

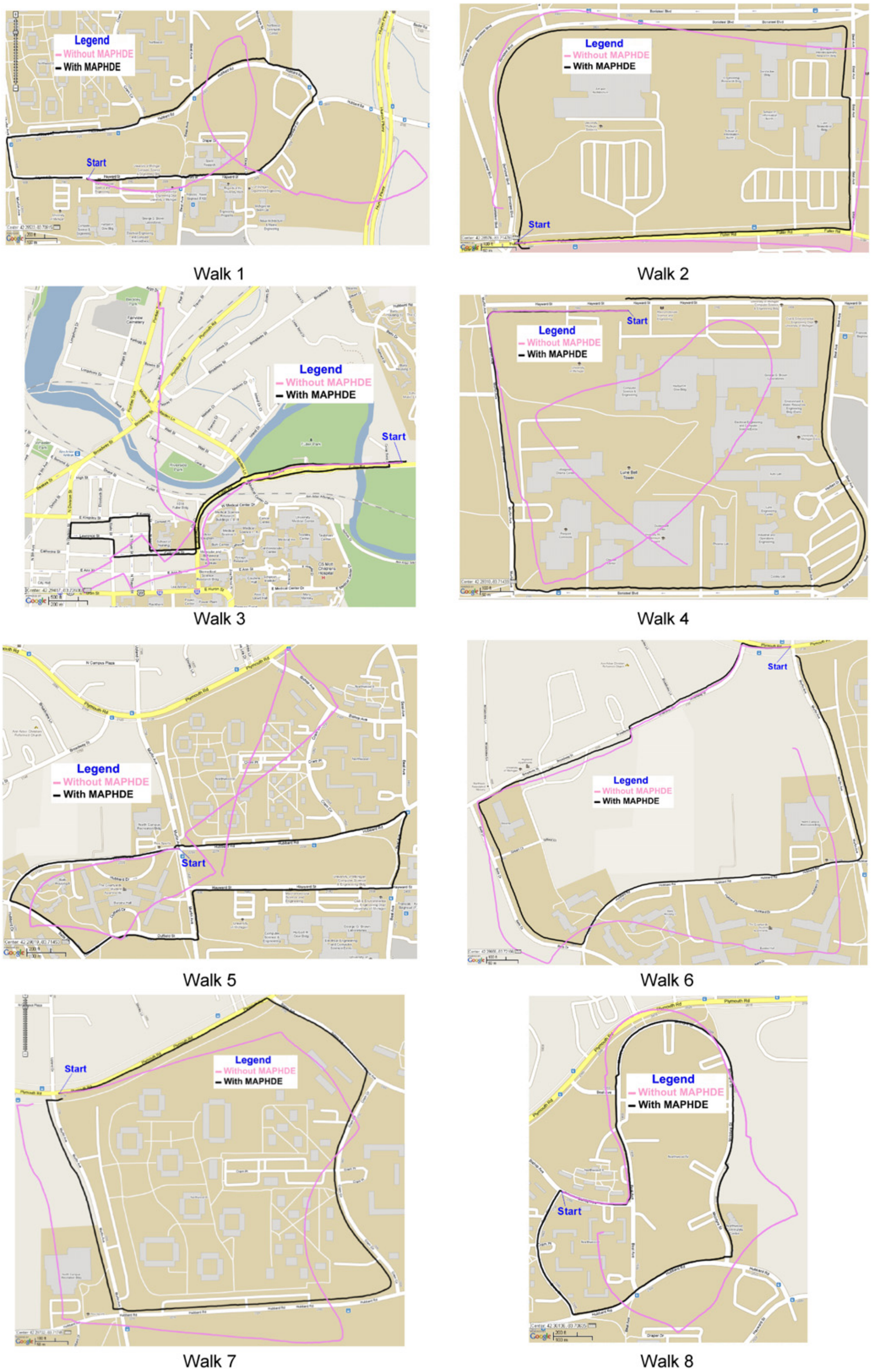

Figure 8. Trajectory of eight outdoor walks. Light-magenta curve: trajectory from PDR data without MAPHDE correction. Black curve: trajectory from PDR data with MAPHDE correction.

is to implement the tracking system in a hand-held device with a touch screen, similar in size and appearance to a handheld GPS device. The user then indicates his/her initial position by tapping the starting position on the screen that displays a local area map. Of course, it would be advised to start a walk at an easily identifiable landmark, such as the intersection of two streets. If a walk started in an area where GPS was available, but then transitioned into a GPS-denied area, then the last GPS-based position could serve as the initial position for the IMU-tracked portion of the walk. Conceivably, a comprehensive

Table 2. Quantitative results for the eight test walks.

Walks	Total duration (min)	Total travel distance (m)	Return position error (RPE) for for uncorrected trajectory (m)	Relative RPE for uncorrected trajectory as percentage of distance traveled (%)	RPE for corrected trajectory (m)	Relative RPE for corrected trajectory as percentage of distance traveled (%)	Improvement provided by MAPHDE (ratio of uncorrected to corrected RPE)
Walk 1	25.48	2197	890.77	40.6	16.10	0.7	55.3-fold
Walk 2	48.12	4344	1739.75	40.1	20.29	0.5	85.7-fold
Walk 3	31.56	1814	360.77	19.9	16.51	0.9	21.9-fold
Walk 4	31.93	2477	133.21	5.4	11.73	0.5	11.3-fold
Walk 5	24.76	1724	160.25	9.3	16.61	1.0	9.6-fold
Walk 6	24.12	1575	53.46	3.4	21.18	1.3	2.5-fold
Walk 7	23.63	1700	43.68	2.6	10.20	0.6	4.3-fold
Walk 8	26.83	2276	148.37	6.5	10.50	0.5	14.1-fold
Average	29.55	2263	441.28	16.4	15.39	0.7	25.6-fold

tracking system could transition from GPS-based to IMU-based tracking automatically, without user intervention.

4.5.2. Initial heading. In a similar approach, initial heading can be indicated by tapping the same touch screen a second time (the first time was for indicating position), at a different point but on the same street. Since a street has only two possible directions, the second tap would unambiguously specify which of the two directions the user wishes to follow initially. In addition, the user would have to take the first few steps in that direction along the street.

5. Experimental results

This section presents the experimental results obtained with the MAPHDE algorithm by a walking subject in real-time. In all cases, two trajectories are shown. The light-magenta curve shows the output of the PDR system without MAPHDE, while the black curve represents the trajectory after MAPHDE was applied. These results focus on experiments consisting of a set of long, complex, closed-loop outdoor walks.

A total of eight walks of different durations and lengths were performed on partially sloped streets, as illustrated in figure 8. The three tunable parameters were kept the same for all of the walks. The walk data are shown using the GPS visualizer toolbox [34]. Note that ground truth is not shown in these figures due to the unavailability of accurate GPS data. An alternative ground truth, based on the average distance of the walker from the centerline of the streets (as could be derived from the OSM database) was also not feasible, because the subjects usually walked on sidewalks, typically several meters away from the streets' centerlines. Also, the OSM data specify only a few fixed points to express the position of curving streets, which is not sufficient to derive ground truth for a curving street (figure 2). In the absence of useable ground truth, graphical results for each of the eight walks are provided in figure 8. A deliberate effort was made to challenge the MAPHDE method by choosing routes with long curving segments. For straight streets, MAPHDE would have performed even better.

A particular challenge in walk 3 was the routing through a downtown area, where the risk of 'snapping' to the wrong street was elevated due to the tight grid of streets. It is clearly visible that the MAPHDE results did not oscillate back and forth between two closely spaced parallel streets common in this dense urban environment.

It is apparent from all these trajectories that MAPHDE substantially improved position estimation. In order to express the performance of MAPHDE qualitatively, a metric called 'return position error' (RPE) was defined which represents the difference between the end point as estimated by the PDR system and the starting point of the trajectory. Since the subjects started and stopped each walk at the exact same point, the estimated trajectories should also start and stop at the same point. Relative RPE, which is the percentage of RPE divided by the total traveled distance, was also calculated. Table 2 provides quantitative result for the eight walks. One can see from these results that the uncorrected RPEs vary dramatically, whereas corrected RPEs are consistently small. Other differences in the magnitude of the uncorrected errors are due to differences in the duration of the walks.

Although this paper focuses exclusively on the correction of heading errors, it should be noted that the results in figure 8 and table 2 reflect heading errors as well as errors in step-length estimation. The latter, however, are very small in the PDR system thanks to the foot-mounted IMU, which allows the application of ZUPT with every footfall (as was explained in section 2).

6. Conclusion

This paper introduced a method for using known street maps for correcting heading errors in an IMU-based personnel tracking system. The proposed MAPHDE method combines a map-matching technique with a heuristics-based correction algorithm. The strength of the heuristics algorithm is that it is tolerant to short-term deviations from walking along the directions of streets. Such deviations occur routinely when the walker dodges other pedestrians, crosses streets, or turns.

Because of the difficulty of recording accurate ground truth data (a high-accuracy differential GPS system would have to be used), plots of all eight outdoor walks overlaid over a map are provided. The plots show the uncorrected and MAPHDE-corrected trajectories and thereby illustrate the effectiveness of the MAPHDE method. It is also evident from the plots that by reducing heading errors to near-zero at steady state, position errors are reduced substantially. The quantitative results in table 2 show that MAPHDE reduces return position errors on average to less than 1% of distance traveled, providing a 25-fold reduction in return position errors on average.

Acknowledgments

This research was supported in part by the Ground Robotics Reliability Center (GRRC) at the University of Michigan, with funding from government contract DoD-DoA W56H2V-04-2-0001 through the Joint Center for Robotics.

References

[1] Foxlin E 2005 Pedestrian tracking with shoe-mounted inertial sensors *IEEE Comput. Graph. Appl.* **25** 38–46

[2] Mezentsev O, Collin J, Kuusniemi H and Lachapelle G 2004 Accuracy assessment of a high sensitivity GPS based pedestrian navigation system aided by low-cost sensors *11th Saint Petersburg Int. Conf. on Integrated Navigation Systems (St Petersburg, 24–26 May 2004)*

[3] Basnayake C, Mezentsev O, Lachapelle G and Cannon M E 2005 An HSGPS, inertial and map-matching integrated portable vehicular navigation system for uninterrupted real-time vehicular navigation *Int. J. Veh. Inform. Commun. Syst.* **1** 131–51

[4] Kim J W, Jang H J, Hwang D-H and Park C 2004 A step, stride and heading determination for the pedestrian navigation system *J. Glob. Positioning Syst.* **3** 273–9

[5] Ladetto Q, Gabaglio V and Meminod B 2001 Combining gyroscopes, magnetic compass and GPS for pedestrian navigation *Int. Symp. on Kinematic Systems in Geodesy, Geomatics and Navigation (Banff, AB, Canada, 5–8 June)*

[6] Stirling R, Collin J, Fyfe K and Lachapelle G 2003 An innovative shoe-mounted pedestrian navigation system *CD-ROM Proc. GNSS, the European Navigation Conf.* pp 103–12

[7] Ojeda L and Borenstein J Personal dead-reckoning system for GPS-denied environments *IEEE Int. Workshop on Safety, Security, and Rescue Robotics (SSRR2007) (Rome, Italy, 27–29 Sept. 2007)*

[8] Borenstein J, Ojeda L and Kwanmuang S 2009 Heuristic reduction of gyro drift in a personal dead reckoning system *J. Navig.* **62** 41–58

[9] Borenstein J and Ojeda L 2010 Heuristic drift elimination for personnel tracking systems *J. Navig.* **63** 591–606

[10] Judd T 1997 A personal dead reckoning module *Institute of Navigation's ION 97 (Kansas City, MO, Sept. 1997)*

[11] Ojeda L and Borenstein J 2006 Non-GPS navigation for emergency responders *Int. Joint Topical Meeting: Sharing Solutions for Emergencies and Hazardous Environments (Salt Lake City, UT, 12–15 Feb.)*

[12] Brand T and Philips R 2003 Foot-to-foot measurement as an aid to personal navigation *Institute of Navigation Annual Meeting 59th (Albuquerque, NM, 23–25 June)*

[13] Stirling R, Collin J, Fyfe K and Lachapelle G 2003 An innovative shoe-mounted pedestrian navigation system *Proc. European Navigation Conf. (Graz, Austria, 22–25 April)*

[14] Misra P and Enge P 2006 *Global Positioning System, Signals, Measurements, and Performance* (Lincoln, MA: Ganga-Jamuna Press)

[15] Moafipoor S, Grejner-Brzezinska A D and Toth C K 2008 A fuzzy dead reckoning algorithm for a personal navigator *Navig. ION J.* **55** 241–55

[16] Butz A, Baus J and Kruger A 2000 Augmenting buildings with infrared information *Proc. Int. Symp. on Augmented Reality* (IEEE Computer Society Press) pp 93–6

[17] Cho S Y and Park C G 2006 MEMS based pedestrian navigation system *J. Navig.* **59** 135–53

[18] Newman J, Ingram D and Hopper A 2001 Augmented reality in a wide area sentient environment *Proc. IEEE and ACM Int. Symp. on Augmented Reality* pp 77–86

[19] Saeed R A and Khatun S 2006 Performance of ultra-wideband time-of-arrival estimation enhanced with synchronization scheme *ECTI Trans. Electr. Eng., Electron. Commun.* **4** 78–84

[20] Eggert R and Raquet J 2004 Evaluating the navigation potential of the NTSC analog television broadcast signal *Proc. ION GNSS-2004 (Long Beach, CA, Sept. 2004)* pp 2436–46

[21] Jarvis A M Y 2008 Integration of photogrammetric and LiDAR data for accurate reconstruction and visualization of urban environments *MSc Thesis* University of Calgary, UCGE Report 20282

[22] Cinaz B and Kenn H 2008 HeadSLAM—simultaneous localization and mapping with head-mounted inertial and laser range sensors *ISWC (Pittsburgh, PA)*

[23] Retscher G 2004 Multi-sensor systems for pedestrian navigation *ION GNSS 2004 Conf. (Long Beach, CA, 21–24 Sept. 2004)*

[24] Pahlavan K, Li X and Makela J-P 2002 Indoor geolocation science and technology *IEEE Commun. Mag.* **40** 112–8

[25] Bernstein D and Kornhauser A 1998 Map matching for personal navigation assistants *Proc. 77th Annual Meeting of the Transportation Research Board (Washington DC, 11–15 Jan.)*

[26] US Census Bureau Geography 2002 TIGER page: cartographic boundary files, available at http://www.census.gov/geo/www/tiger (accessed 2 May 2009)

[27] OpenStreetMap, available at http://www.openstreetmap.org (accessed 10 July 2009)

[28] Greenfeld J S 2002 Matching GPS observations to locations on a digital map *Proc. 81st Annual Meeting of the Transportation Research Board (Washington, DC)*

[29] Kim W, Jee G and Lee J 2000 Efficient use of digital road map in various positioning for ITS *IEEE Symp. on Position Location and Navigation (San Diego, CA)*

[30] Fu M, Li Jie and Wang M 2004 A hybrid map matching algorithm based on fuzzy comprehensive Judgment *IEEE Proc. on Intelligent Transportation Systems* pp 613–7

[31] Quddus M A, Ochieng W Y, Zhao L and Noland R B 2003 A general map matching algorithm for transport telematics applications *GPS Solut.* **7** 157–67

[32] Scott C A and Drane C R 1994 Increase accuracy of motor vehicle position estimation by utilizing map data: vehicle dynamics and other information sources *Proc. Vehicle Navigation and Information Systems* pp 585–90

[33] Retscher G 2006 An intelligent multi-sensor system for pedestrian navigation *J. Glob. Positioning Syst., GPS Solut.* **5** 110–8

[34] GPS Visualizer: Do-It-Yourself Mapping 2003 Available at http://www.gpsvisualizer.com (accessed 31 Dec. 2009)