Pedestrian dead reckoning employing simultaneous activity recognition cues

2012 Meas. Sci. Technol. 23 025103

(http://iopscience.iop.org/0957-0233/23/2/025103)

Meas. Sci. Technol. **23** (2012) 025103 (20pp) doi:10.1088/0957-0233/23/2/025103

Pedestrian dead reckoning employing simultaneous activity recognition cues

Kerem Altun[1] **and Billur Barshan**

Department of Electrical and Electronics Engineering, Bilkent University, Bilkent, TR-06800 Ankara, Turkey

E-mail: kaltun@cs.ubc.ca, billur@ee.bilkent.edu.tr

Received 14 September 2011, in final form 18 November 2011
Published 11 January 2012
Online at stacks.iop.org/MST/23/025103

Abstract
We consider the human localization problem using body-worn inertial/magnetic sensor units. Inertial sensors are characterized by a drift error caused by the integration of their rate output to obtain position information. Because of this drift, the position and orientation data obtained from inertial sensors are reliable over only short periods of time. Therefore, position updates from externally referenced sensors are essential. However, if the map of the environment is known, the activity context of the user can provide information about his position. In particular, the switches in the activity context correspond to discrete locations on the map. By performing localization simultaneously with activity recognition, we detect the activity context switches and use the corresponding position information as position updates in a localization filter. The localization filter also involves a smoother that combines the two estimates obtained by running the zero-velocity update algorithm both forward and backward in time. We performed experiments with eight subjects in indoor and outdoor environments involving walking, turning and standing activities. Using a spatial error criterion, we show that the position errors can be decreased by about 85% on the average. We also present the results of two 3D experiments performed in realistic indoor environments and demonstrate that it is possible to achieve over 90% error reduction in position by performing localization simultaneously with activity recognition.

Keywords: inertial sensing, wearable computing, pedestrian dead reckoning, human localization, human activity recognition

(Some figures may appear in colour only in the online journal)

1. Introduction

Dead reckoning is the process of estimating the current position of a moving entity using the position estimate (or fix) calculated at previous time instants and the velocity (or speed) estimate at the current time instant. It can also be used to predict the future position by projecting the current known position and speed to a future instant [1]. Since the past position estimates are projected through time to obtain new estimates in dead reckoning, position errors accumulate over time. Because of this cumulative error propagation, dead-reckoning estimates are unreliable if calculated over long periods of time. Hence, dead reckoning is seldom used alone in practice and is often combined with other types of position sensing to improve position accuracy.

Historically, dead reckoning has been used in ship navigation for centuries. Reference [1] explains its use in ship navigation in detail. It has been used in air navigation since the beginning of 1900s; a thorough survey appears in [2, 3]. A survey on the positioning and navigation methods for vehicles appears in [4]. Dead reckoning is employed in mobile robotics through the use of odometry [5] and/or inertial navigation systems (INSs).

INSs [6] can be used for both indoor and outdoor positioning and navigation. Fundamentally, gyroscopes provide angular rate information and accelerometers provide velocity rate information. Although the rate information is reliable over long periods of time, it must be integrated to provide position, orientation and velocity estimates. Thus,

[1] Present address: Department of Computer Science, University of British Columbia, Vancouver, BC, Canada.

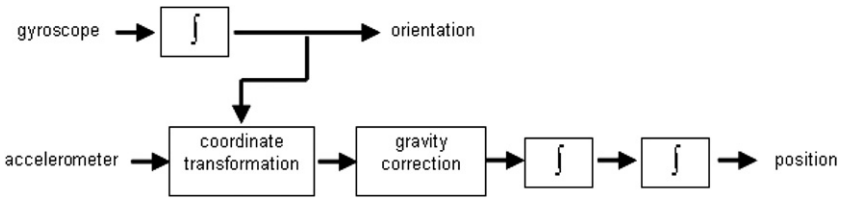

Figure 1. Strap-down INS integration.

even very small errors in the rate information provided by inertial sensors cause unbounded growth in the error of the integrated measurements. As a consequence, an INS by itself is characterized by position errors that grow with time and distance, usually referred to as the 'drift error.' One way of overcoming this problem is to periodically reset inertial sensors with external absolute sensing mechanisms and to eliminate this accumulated error. Thus, in most cases, data from an INS must be integrated with absolute location-sensing mechanisms to provide useful information about position.

An inertial measurement unit (IMU) consists of orthogonally mounted accelerometers and gyroscopes in three spatial directions. If the IMU is directly mounted on the moving object, the system is called a strap-down INS [6]. The IMU provides three acceleration and three angular velocity (or angular rate) outputs in the object coordinate frame. A basic block diagram of a strap-down INS is given in figure 1. To estimate the orientation (or attitude) of the moving object, the gyroscope outputs should be integrated. Then, using the estimated orientation, accelerometer outputs should be transformed to the Earth coordinate frame. The acceleration values in the Earth coordinate frame are integrated twice to get the position. Because of the integration operations involved in the position calculation, any error in the sensor outputs accumulates in the position output, causing a rapid drift in both the gyroscope and accelerometer outputs. Thus, the reliability of position estimates decreases with time. For example, a constant bias in the gyroscope will cause an error in the position that grows proportional to the cube of time, and a constant bias in the accelerometer will cause an error that grows proportional to the square of time [7]. For this reason, inertial sensors are usually used in conjunction with other sensing systems that provide absolute external reference information.

One application of INSs is in pedestrian dead reckoning (PDR). PDR systems are generally used in GPS-denied environments such as inside buildings, tunnels, underground or dense forests and around tall buildings in urban areas where GPS data are not accurate or always available. References [8] and [9] provide brief surveys on PDR systems. Such systems are usually developed for security personnel and emergency responders [10]. Unlike land vehicles and robots, a method called 'zero velocity update' (ZUPT) enables the stand-alone usage of INSs on pedestrians, without any external reference sensor. The ZUPT method exploits the fact that during walking, the velocity of the foot is zero at some time interval during the stance phase (see section 3.1). If this time interval is correctly detected, the drift in the velocities calculated in strap-down integration can be reset to zero and the drift in one step will not be carried over to the next step. As an alternative, instead of directly resetting the velocities to zero, this information can be used as a measurement in a Kalman filter [11, 12]. In [10], the ZUPT method is used to estimate the distance travelled and a high-grade gyroscope is employed to estimate the orientation. Alternative methods for orientation estimation also exist in the literature. In [13], a Kalman filter is used to estimate the orientation. Accelerometers and magnetometers can also be used interchangeably with gyroscopes depending on whether the body is in motion or not [14]. Another approach is to use the orientation output of a commercially available sensor module that integrates accelerometer, gyroscope and magnetometer measurements [15]. An extensive survey on orientation estimation methods using body-worn sensors appears in [16]. Heuristic methods that exploit the usual walking patterns of people can also be applied for drift reduction [17] and elimination [18] in gyroscopes.

In order to apply the ZUPT method, correct detection of gait events such as the stepping instants and correct estimation of gait parameters such as stride length are crucial for many PDR systems. This detection can be performed using only inertial sensors as in [19]. Zero-velocity detection algorithms using inertial sensors are compared in [20, 21]. In [22], an external pressure sensor is used to detect the steps. It is also possible to perform activity recognition with inertial sensors to detect the stepping instants and estimate the stride length [23, 24].

Integrating external reference sensors with PDR systems is also common in the literature. In [13, 25], a shoe-mounted inertial/magnetic system is used together with a quaternion-based extended Kalman filter (EKF) to estimate the 3D path travelled by a walking person. Magnetic sensors are used in the initialization of the EKF. Reference [26] combines dead reckoning with GPS in outdoor environments. For indoor environments, WiFi fingerprinting method is used for localization. Reference [27] uses the GPS data for error correction. The pedestrian trajectory is estimated using a PDR system and a wireless sensor network in [8].

Another alternative for integrating external references is map matching. If a map of the environment is available, this information can be used to provide drift error correction. In [28], this idea is applied in an outdoor environment, combined with a heuristic drift elimination procedure described in [18]. In indoor environments, activity-based map matching can be used [24]. This idea exploits the fact that the activity context of the pedestrian gives information about his location. For example, if the pedestrian is ascending stairs, most locations on an indoor map can be ruled out, improving the position estimate. Here, we follow a similar approach.

In this study, we perform pedestrian localization using five inertial and magnetic sensor units worn on the body [29]. Localization is performed simultaneously with activity recognition, where activity recognition cues are used in the position updates to correct the drift errors of inertial sensors. Apart from being inherent in inertial sensors, drift errors and offsets in body-worn systems can also arise from initial misplacement, occasional slips from the initial position and orientation during operation, or loose mounting on the body. Even though the initial errors are expected to be small, they are accumulated and result in larger errors over long periods of time. To the best of our knowledge, these issues have not been addressed before in the literature. We demonstrate that using a given map of the environment and activity recognition cues, these errors can be reduced considerably and accurate localization can be achieved without having to use any external reference sensor. In practice, the proposed method can be used in applications where a map is available and GPS data are not reliable or not available at all (e.g., underground mines, indoor areas and urban outdoor areas with tall buildings). We note that although here we use activity recognition information to improve localization performance, the converse is also possible, i.e. localization information and a map can improve the activity recognition performance. However, in our previous studies, since we have observed that activity recognition with high accuracy can already be achieved using proper signal processing and pattern recognition techniques [30], we focus on only one side of the loop in this paper. In a very recent study, activity recognition and body pose estimation are combined in a very similar way [31].

We have performed experiments in both 2D and 3D environments. In the 2D experiments, walking, standing and turning activities are considered. In 3D localization experiments, ascending/descending stairs activity is added to these activities. We assume that a map of the environment is available and that the switches between these activities usually correspond to multiple locations on the map. For example, in an indoor environment, switching from walking to turning activity might correspond to the end of a corridor or to the front of a room, whereas switching from walking to standing activity might correspond to a location in front of a lift. Therefore, activity switches usually correspond to several discrete locations in the environment. If one can detect the activity switches correctly, it is possible to use the corresponding position information in order to correct the drift in the position.

The rest of this paper is organized as follows: in section 2, we describe the sensors used in this study. Section 3 explains the theoretical background of the applied methods. Sections 4 and 5 present the results of 2D and 3D experiments, respectively. We provide a discussion of the results, limitations of the proposed method and related issues in section 6 and conclude with section 7, providing some future research directions.

2. Inertial/magnetic sensing equipment

In this study, we use five MTx three-degree-of-freedom (3-DOF) orientation trackers (figure 2), manufactured by Xsens Technologies [32]. Each MTx unit has a tri-axial accelerometer, a tri-axial gyroscope and a tri-axial magnetometer so that the sensor units acquire 3D acceleration, rate of turn and the strength of the Earth's magnetic field. Accelerometers of two of the MTx trackers can sense in the range $\pm 50\ \mathrm{m\,s^{-2}}$ (standard range) and the other three can sense in the range of $\pm 180\ \mathrm{m\,s^{-2}}$ (customized range). All gyroscopes in the MTx units can sense in the range of $\pm 1200^{\circ}\ \mathrm{s^{-1}}$ angular velocities; magnetometers can sense magnetic fields in the range of $\pm 75\ \mu\mathrm{T}$. Additionally, each sensor unit has a built-in Kalman filter that outputs the orientation of the sensor with respect to a global coordinate frame (see section 3.1). Three orientation output modes can be used for the output: direction cosine matrix, quaternion and Euler angles. In this study, we use the quaternion output mode.

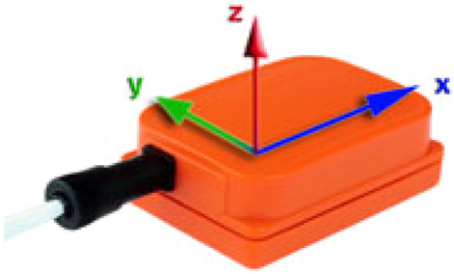

Figure 2. MTx 3-DOF orientation tracker (reprinted from http://www.xsens.com/en/general/mtx).

The sensors are placed on five different positions on the subject's body as shown in figure 3. Two of the customized sensor units are placed on the feet, the remaining customized unit is placed on the subject's chest and the standard units are placed on the sides of the knees (the right side of the right knee and the left side of the left knee). The customized units are used on the feet to avoid saturation in the sensor outputs, because feet accelerations are expected to be larger than knee accelerations (up to $\pm 90\ \mathrm{m\,s^{-2}}$ in our experiments). The sensor units on the feet and chest are used to estimate the distance travelled and the heading, respectively. The sensor units on the legs are not used in the localization process; they are used for activity recognition in the 3D experiments.

3. Methodology

In the following, we refer to several different coordinate frames, which are the global coordinate frame, local navigation coordinate frames and the sensor coordinate frames (figure 4). There is a single global coordinate frame. In the default configuration of this coordinate frame, the z axis points upward along the vertical (opposite to the direction of the gravity vector g), the x axis points towards the magnetic north and the y axis points to the west, completing the right-handed coordinate frame (figure 4(a)). The local navigation coordinate frames are translated versions of the global frame to the position of each sensor unit, and therefore, in the default case, also have their z axes pointing upwards along the vertical, x axes pointing in the magnetic north direction and y axes pointing to the west. In other words, there is a single global coordinate frame but five local navigation coordinate frames, one for each sensor

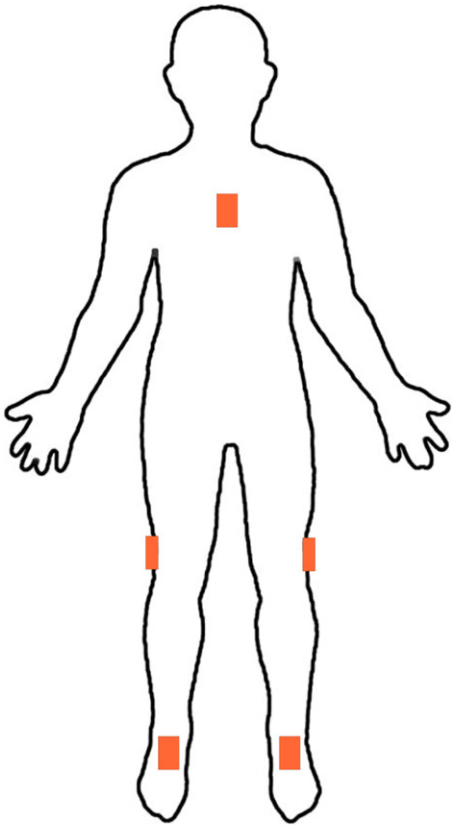

Figure 3. The locations of the sensor units on the body. (The outline of the human body is taken from http://www.anatomyacts.co.uk/learning/primary/Montage.htm.)

unit. The axes of the local navigation coordinate frames always remain parallel to the axes of the global coordinate frame but their origins are shifted to the locations of the sensor units. The sensor coordinate frames also have their origins at the positions of the sensor units but their three axes have arbitrary orientation initially, as shown in figure 4(*a*).

As stated above, the MTx units provide raw acceleration, angular velocity and magnetic field data, in addition to the orientation data that are calculated by the built-in Kalman filter. In this section, the steps used for processing these data are explained. The processing is done in two separate tracks, one of which is for localization and the other is for activity recognition.

3.1. Localization

The processing for localization is done in two main steps. In the first step, the trajectories are found using the ZUPT method, mentioned in section 1. In the second step, a Kalman filter-like state estimation procedure is employed to utilize the activity recognition cues and improve the results.

We perform the regular strap-down integration procedure, using the orientation data output from the MTx sensor and ZUPTs. A block diagram that summarizes this procedure is depicted in figure 5. As shown in the diagram, calculations for the distance travelled and the heading are performed separately. To estimate the heading, it is possible to use the orientation output of the MTx unit either on the chest or on the feet. We use the chest sensor output because during walking, the chest is a relatively stable reference to measure the person's heading as opposed to the feet. That is, the signals recorded on the chest are less oscillatory than the signals acquired from other locations. The quaternion output mode is used for orientation to avoid the occurrence of any singularities possible in the Euler angle mode, even though this is unlikely for the chest. At the beginning of the experiments, a reset operation is performed on the coordinate frames such that the yaw angle is initially set to zero and is measured with respect to the vertical axis during the motion (see section 4.1). Then, the orientation data are converted to Euler angles (see, for example, [33]). In the Euler angle domain, the yaw angle (ψ) represents the instantaneous heading. Here, it is assumed that the left and right turns performed during motion are about the vertical axis.

To estimate the distance travelled, the sensor signals on either foot can be used. First, using the orientation output of the sensor unit, the accelerations are transformed from the sensor coordinate frame to the local navigation coordinate frame. The transformation can simply be performed as

$$\mathbf{a}_L = q_{LS}\,\mathbf{a}_S\,q_{LS}^* = q_{LS}\,\mathbf{a}_S\,q_{SL}, \tag{1}$$

where $\mathbf{a}_L$ is the acceleration vector in the local navigation frame, $\mathbf{a}_S$ is the acceleration vector in the sensor coordinate

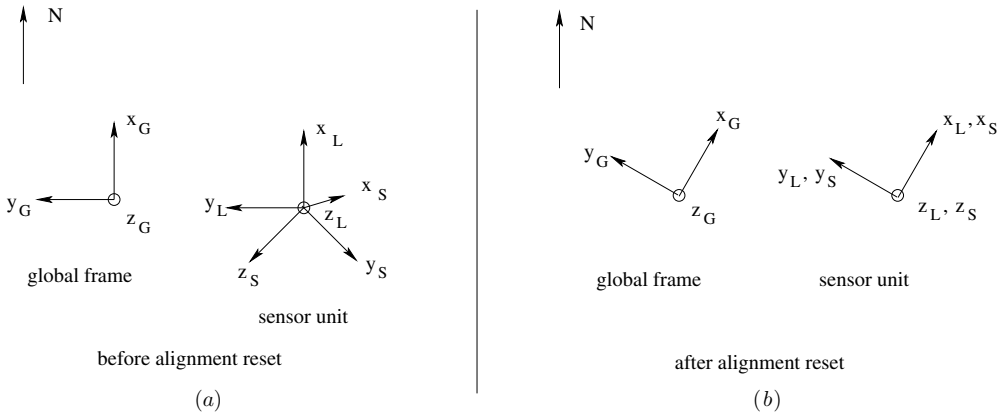

Figure 4. Top views of the global (G), local navigation (L) and sensor (S) coordinate frames (*a*) before and (*b*) immediately after the alignment reset operation.

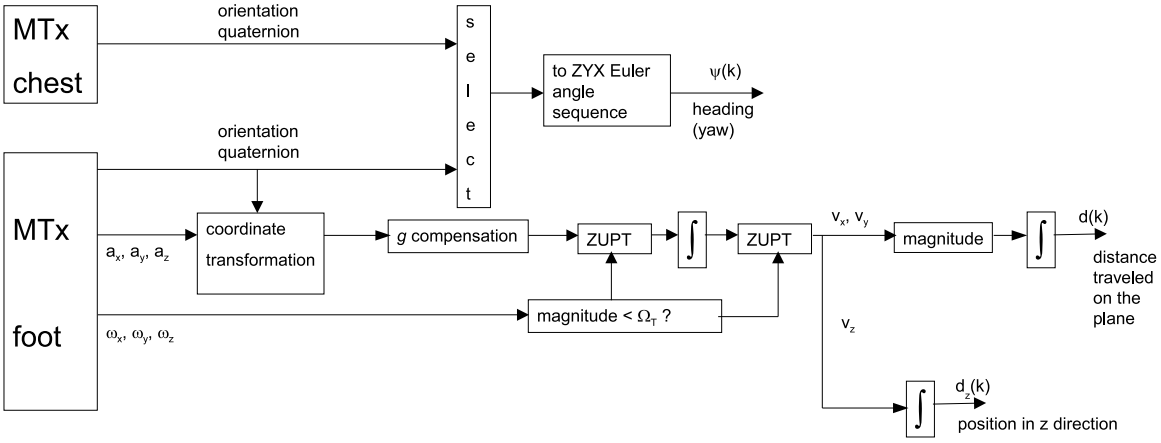

Figure 5. Block diagram for the first processing step.

Figure 6. The human gait cycle
(figure from http://www.sms.mavt.ethz.ch/research/projects/prostheses/GaitCycle).

frame and q_{LS} is the quaternion representing the orientation of the sensor coordinate frame with respect to the local navigation frame. To estimate the position from the acceleration signal, the acceleration data must be integrated twice. Because of this integration procedure, the errors in the sensor readings are accumulated, causing unbounded drift in the position.

We use the ZUPT method [10] to reduce the drift in position. When a person is walking, the motion of the leg is quasiperiodic. The collection of these motions within one period is called the gait cycle. The human gait cycle is roughly divided into two phases called the stance phase and the swing phase. The stance phase is defined as the time interval during which the foot is in contact with the ground, and the swing phase is the time interval during which the foot does not touch the ground. Stance phase takes approximately 60% of the gait cycle, as shown in figure 6. During a sub-interval ΔT of the stance phase, the foot velocity and acceleration are expected to be zero. Thus, the true values of the velocity and acceleration are known. If one can successfully detect this sub-interval, the sensor signals can be reset to zero and the drift error in one step will not be carried over to the next step.

The problem is now converted to successfully detecting the ΔT interval where the foot velocity is exactly zero. There are a number of detectors used in the literature for this purpose: acceleration moving variance detector, acceleration magnitude detector and angular rate magnitude detector [20]. In a recent study, an alternative detector was proposed that gives slightly better results than the angular rate magnitude detector [20]. However, in most of the studies, the angular rate magnitude detector outperforms the others. We use the angular rate magnitude detector in this study because of its performance and simplicity of implementation. Using the magnitude of the angular velocity (rate), the following binary signal is constructed:

$$I_{\text{step}}(k) = \begin{cases} 1, & |\boldsymbol{\omega}(k)| \leqslant \Omega_T \\ 0, & |\boldsymbol{\omega}(k)| > \Omega_T, \end{cases} \tag{2}$$

where k is the time step, $|\boldsymbol{\omega}(k)| = \sqrt{\omega_x(k)^2 + \omega_y(k)^2 + \omega_z(k)^2}$ and Ω_T is a pre-set threshold value. This signal is constructed separately for the left foot and the right foot sensors. When this signal is 1, the foot is assumed to be in the stance phase; otherwise it is assumed to be in the swing phase. To eliminate possible instantaneous 0-1-0 or 1-0-1 switches in this signal, a median filter is used. Then, the velocities and accelerations are set to zero when this signal is 1, and the integrations in the block diagram in figure 5 are performed. Note that the integrations on the plane and in the z direction are performed separately, resulting in the signals $d(k)$ and $d_z(k)$, which correspond to the distance travelled on the $x-y$ plane and the position on the z axis, respectively.

Because of the slight movement of the chest during walking, the heading signal contains ripples, as shown by the blue-dashed line in figure 7(*a*). This signal can be smoothed using the gait phase data obtained using the aforementioned method. The I_{step} signal of the right foot is superimposed on this plot in the green-solid line in the same figure. These data are obtained in an experiment where the subject stands for 5 s, then starts walking along a straight line, then turns 90° to the right at about $t = 25$ s and continues walking. As can be observed in the figure, when a right step is taken (i.e. when $I_{\text{step}} = 0$ for the right foot), the chest angle swings slightly to the left, and vice versa. To remove the ripples, the mean of the heading data between rising edges of the I_{step} signal can be calculated and replaced as a corrected heading signal. This is

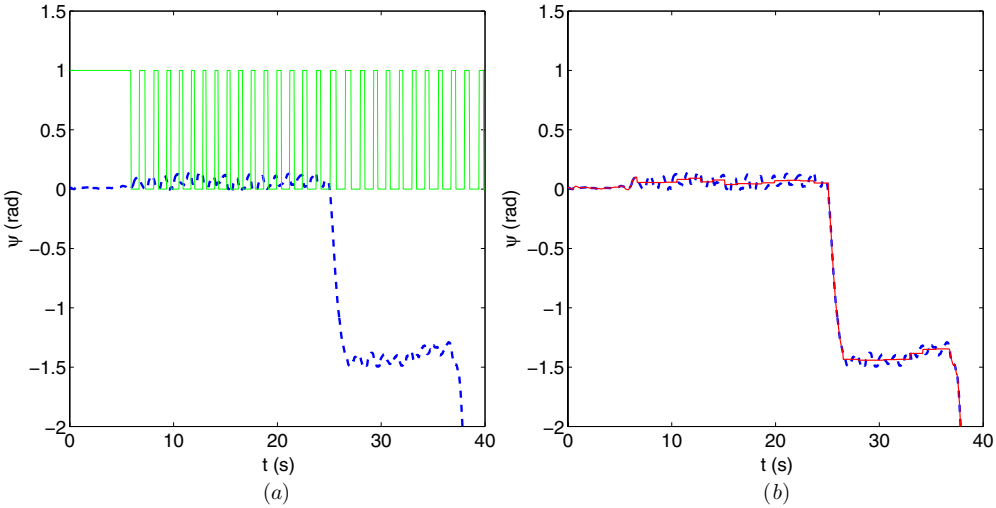

Figure 7. (*a*) Original heading signal (blue-dashed line) and swing-stance phase indicator variable (green-solid line) superimposed; (*b*) original heading signal (blue-dashed line) and corrected heading signal (red-solid line).

shown in figure 7(*b*). In this figure, the original heading data are shown by the blue-dashed line and the corrected heading is shown by the red-solid line. Obviously, this correction should be made separately for either foot depending on which foot's data are used in evaluating $d(k)$, using the I_{step} indicator for that foot. In this case, the correction is made using the right foot data. The corrected heading data are denoted as $\psi(k)$ in the rest of this text.

After determining $d(k)$, $d_z(k)$ and $\psi(k)$, the path can be reconstructed using the simple state model given below:

$$\begin{aligned} x(k) &= x(k-1) + \Delta d(k-1)\cos[\psi(k-1)] \\ y(k) &= y(k-1) + \Delta d(k-1)\sin[\psi(k-1)] \\ z(k) &= z(k-1) + \Delta d_z(k-1), \end{aligned} \tag{3}$$

with the initial conditions $x(0)$, $y(0)$ and $z(0)$. Here, $\Delta d(k-1) = d(k)-d(k-1)$ represents the distance travelled on the plane and $\Delta d_z(k-1) = d_z(k)-d_z(k-1)$ represents the displacement in the z direction, during the kth time step.

By defining a state vector $\boldsymbol{\xi}(k) = [x(k),\ y(k),\ z(k)]^T$ and an input vector $\mathbf{u}(k) = [\Delta d(k)\cos\psi(k),\ \Delta d(k)\sin\psi(k),\ \Delta d_z(k)]^T$, the equation becomes

$$\boldsymbol{\xi}(k) = \boldsymbol{\xi}(k-1) + \mathbf{u}(k-1) \tag{4}$$

with the initial condition $\boldsymbol{\xi}(0) = [x(0),\ y(0),\ z(0)]^T$. In the 2D experiments, we do not consider the z direction. That is, $d_z(k)$ is not calculated and the state $z(k)$ is deleted from the state vector in these experiments.

The performance of the above model depends on the performances of the distance and the heading estimation methods. In our experiments, we observed that both have errors, which causes the reconstructed path to drift over time. This drift is naturally amplified as the length of the walking path increases. The most dominant cause of error is the dislocation of the mounted sensors during the experiments, especially the heading sensor. For example, a slight dislocation of the chest sensor causes a slight measurement error in the heading that causes the path to drift drastically over long periods of walking. This could be caused by attaching the sensors to loose rather than tight clothing. Magnetic disturbance caused by the ferromagnetic materials in the environment is another source of error for the magnetometers that directly affects the heading. Accelerometer data can be used to estimate the inclination angle, but the only external reference available for determining the heading is the magnetic field data. Furthermore, the thresholds that we use are fixed constants, i.e. they are not selected specifically for the person wearing the sensors. Considering the age, height and weight variations among people, such errors are unavoidable. Therefore, we use cues obtained from activity recognition and perform position updates when such cues are available, in order to improve the results.

3.2. Activity recognition

In our earlier work [30], we demonstrated that it is possible to distinguish between various activities using body-worn inertial and magnetic sensors and provided an extensive comparison between various classifiers. Simple Bayes classifiers with Gaussian probability density functions are sufficient to obtain over 95% correct classification rates if training data from that specific person are available. However, if such training data are not available to the classifiers, more complex classifiers such as the k-nearest neighbour method (k-NN) or support vector machines (SVM) can be utilized that have expected correct classification rates of about 85%. The reader is referred to [30, 34–36] for surveys of the literature on activity recognition using body-worn sensors.

In our experiments in 2D, we consider a reduced activity set, comprised of walking, standing and turning activities. Since these three activities are quite different from each other, using complex classifiers is not necessary. We use a rule-based classifier for these three activities, in which the following rules are applied in the given order:

(i) if the filtered heading value is above a certain threshold, the activity is classified as turning;
(ii) if both feet are stationary, then the activity is classified as standing;

(iii) if the above conditions do not hold, then the activity is classified as walking.

For the first rule, the heading signal is passed through a first-order difference filter of length 1 s and thresholded. The second rule is realized by performing an AND operation on the I_{step} indicator variables (equation (2)) for the left and the right feet.

For our 3D experiments, we introduce the 'stairs' activity to the activity set that represents the activity state of the subject while ascending or descending stairs. Distinguishing between walking and stairs activities is not straightforward, and a simple rule-based method like the one applied above cannot be used in this case. Therefore, we use the k-NN classifier. The data acquired in our previous work [30] are employed as the training data for the classifier. From that article, we combine the data of the activities walking in a parking lot (A9) and walking on a treadmill (A10) to get the 'walking' class, and data of ascending stairs (A5) and descending stairs (A6) to get the 'stairs' class. We use the standing (A2) activity for the 'standing' class directly. To recognize these three activities, we use the sensors on the right and the left legs, since they are mounted at the same position as in that article. Therefore, the data are expected to be similar. We calculate the running mean and running variance values from the test data as features, using a sliding window of length 5 s. This length is chosen since the same length is also used in the training data for feature extraction. We do not use magnetometer data, since the accuracy of magnetometers is known to degrade in indoor environments [16]. The k-NN classifier is used to distinguish between walking, standing and stairs activities, whereas the turning activity is recognized using the same rule as in the rule-based method described above. Then, the switches between activities and corresponding time values are determined and used for position updates, as explained in the following section.

3.3. Simultaneous localization and activity recognition

In this section, we combine the localization results with position updates simultaneously obtained from activity recognition cues. We assume that a map of the environment is available and some of the switches between recognized activities correspond to multiple locations on the map, in general. That is, knowledge of an activity switch provides information about the possible positions on the map.

Suppose that, for a given map, a switch from activity A to activity B can occur at N_{AB} different points. The placeholders A and B can stand for any activity in our activity set, i.e. walking (W), standing (S), turning (T) or stairs (R). For example, a walking-to-standing activity switch is denoted as WS and a walking-to-stairs activity switch is denoted as WR. In the following, the nth AB activity switch point is modelled as a Gaussian random vector with mean $\boldsymbol{\mu}_{\text{AB},n}$ and covariance $\mathbf{P}_{\text{AB},n}$, where $n = 1, \ldots, N_{\text{AB}}$. The mean corresponds to the coordinates of the expected location on the given map, and the covariance models the uncertainty of the location.

In the previous section, we use the state equation (4) to predict the position. To model the uncertainty in the position, consider the state equation

$$\boldsymbol{\xi}(k) = \boldsymbol{\xi}(k-1) + \mathbf{u}(k-1) + \mathbf{R}_{\psi(k)}\mathbf{w}(k), \tag{5}$$

with the initial condition $\boldsymbol{\xi}(0)$ modelled as a Gaussian random vector with mean $\boldsymbol{\mu}_{\boldsymbol{\xi}}(0)$ and covariance matrix $\mathbf{P}_{\boldsymbol{\xi}}(0)$. Note that here $\boldsymbol{\xi}(k)$ is a random process and is different from the deterministic state vector in equation (4). However, we use the same notation for simplicity. The input $\mathbf{u}(k)$ is the same as in equation (4). In equation (5), $\mathbf{R}_\theta$ represents a rotation on the plane by an arbitrary angle θ:

$$\mathbf{R}_\theta = \begin{pmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{pmatrix}, \tag{6}$$

and $\mathbf{w}(k)$ is the process noise modelled as a white Gaussian noise with a diagonal covariance matrix $\mathbf{Q}$. In equation (5), the noise vector is rotated by $\psi(k)$ at each time step k. This way, the noise introduced to the system is modelled such that it is uncorrelated (and independent, since it is Gaussian) in the current heading direction and in the perpendicular direction to the heading. If there were no rotation, the noise would be uncorrelated in the global x and y directions, as long as the covariance matrix $\mathbf{Q}$ is diagonal. We believe that introducing this rotation matrix is a more realistic assumption for our model than assuming the noise in the x and y directions as being uncorrelated.

Suppose that an AB activity switch is detected and a position update is performed at a previous time $k = k_1$. Until the next position update, equation (5) can be used to model the position. The prediction equations using this forward model are given as

$$\begin{aligned} \hat{\boldsymbol{\xi}}_f(k|k_1) &= \hat{\boldsymbol{\xi}}_f(k-1|k_1) + \mathbf{u}(k-1) \\ \boldsymbol{\Sigma}_f(k|k_1) &= \mathbf{R}_{\Delta\psi(k)}\boldsymbol{\Sigma}_f(k-1|k_1)\mathbf{R}^T_{\Delta\psi(k)} + \mathbf{R}_{\psi(k)}\mathbf{Q}\mathbf{R}^T_{\psi(k)} \end{aligned} \tag{7}$$

for $k > k_1$, where the subscript f stands for the forward model and $\Delta\psi(k) = \psi(k) - \psi(k-1)$. The initial conditions for these prediction equations depend on the activity switch at $k = k_1$. They are given as $\hat{\boldsymbol{\xi}}_f(k_1|k_1) = \boldsymbol{\mu}_{\text{AB},n}$ and $\boldsymbol{\Sigma}_f(k_1|k_1) = \mathbf{P}_{\text{AB},n}$, where n is the index of the corresponding activity switch point on the map. If no position update is performed up to time k, then $k_1 = 0$ and the initial conditions for the forward filter are the initial conditions of the state model. That is, $\hat{\boldsymbol{\xi}}_f(0|0) = \boldsymbol{\mu}_{\boldsymbol{\xi}}(0)$ and $\boldsymbol{\Sigma}_f(0|0) = \mathbf{P}_{\boldsymbol{\xi}}(0)$.

When an activity switch from activity C to activity D (i.e. a CD switch) is detected at $k = k_2$, we run the same system backwards in time, all the way back to the previous activity switch AB and position update at $k = k_1$. The backward filter equations are

$$\begin{aligned} \hat{\boldsymbol{\xi}}_b(k-1|k_2) &= \hat{\boldsymbol{\xi}}_b(k|k_2) - \mathbf{u}(k-1) \\ \boldsymbol{\Sigma}_b(k-1|k_2) &= \mathbf{R}_{\Delta\psi(k-1)}\boldsymbol{\Sigma}_b(k|k_2)\mathbf{R}^T_{\Delta\psi(k-1)} \\ &\quad + \mathbf{R}_{\psi(k-1)}\mathbf{Q}\mathbf{R}^T_{\psi(k-1)} \end{aligned} \tag{8}$$

for $k_1 < k \leqslant k_2$, where the subscript b stands for the backward model and $\Delta\psi(k-1) = \psi(k-1) - \psi(k)$. The initial conditions for these prediction equations again depend on the current activity switch at $k = k_2$, and are given by $\hat{\boldsymbol{\xi}}_b(k_2|k_2) = \boldsymbol{\mu}_{\text{CD},n^*}$ and $\boldsymbol{\Sigma}_b(k_2|k_2) = \mathbf{P}_{\text{CD},n^*}$. The subscript n^* indicates the predefined CD switch location on the map that is the closest to the forward state estimate just before the position update. More precisely,

$$n^* = \arg\min_n ||\hat{\boldsymbol{\xi}}_f(k_2|k_1) - \boldsymbol{\mu}_{\text{CD},n}||. \tag{9}$$

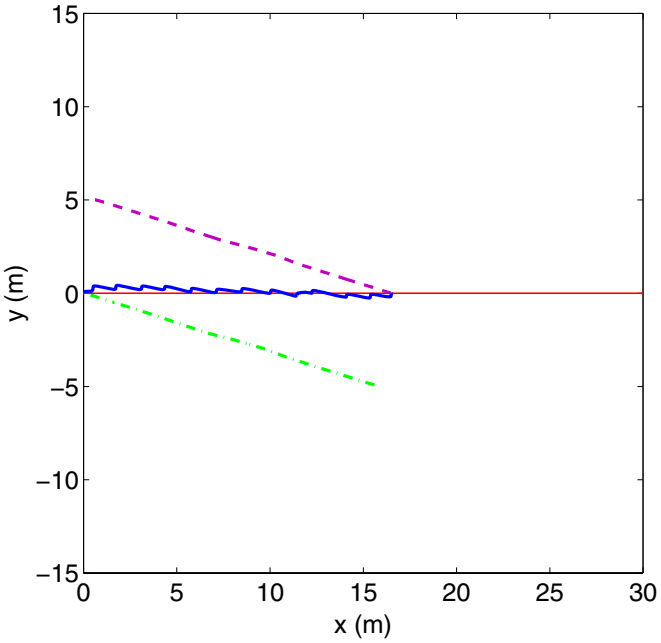

Figure 8. Optimal combination (blue-solid line) of the forward (green-dash-dotted line) and backward (magenta-dashed line) estimates. The thin red-solid line shows the true path.

At this point, for each $k = k_1+1, \ldots, k_2-1$, we have two estimates available for the position. The linear combination of these two estimates with the minimum covariance is (see the appendix)

$$\hat{\boldsymbol{\xi}}(k|k_1,k_2) = \boldsymbol{\Sigma}(k|k_1,k_2) \times [\boldsymbol{\Sigma}_f(k|k_1)^{-1}\hat{\boldsymbol{\xi}}_f(k|k_1) + \boldsymbol{\Sigma}_b(k|k_2)^{-1}\hat{\boldsymbol{\xi}}_b(k|k_2)], \quad (10)$$

where $\boldsymbol{\Sigma}(k|k_1,k_2) = [\boldsymbol{\Sigma}_f(k|k_1)^{-1} + \boldsymbol{\Sigma}_b(k|k_2)^{-1}]^{-1}$ is the covariance of the combined estimate.

In practice, we run the forward filter in a causal manner until an activity switch is detected. When an activity switch is detected at $k = k_2$, the backward filter is run all the way back to the previous position update at $k = k_1$, and the position estimates for $k = k_1 + 1, \ldots, k_2 - 1$ are calculated. If there is no previous position update, then $k_1 = 0$. After the update and the smoothing operation, the new k_1 value is assigned as k_2. This is illustrated in figure 8 that includes a portion of one of our 2D experiments. In the experiment, the subject starts from point $(0, 0)$ and walks in the $+x$ direction, which is shown by the thin red-solid line and represents the ground truth. The green-dash-dotted line shows the reconstructed path until an activity switch is detected, which occurs at point $(16.5, 0)$. The reconstructed path is drifting from the actual path, as shown in the figure.

The average heading error is about 18°. Such large heading errors are not frequently observed in our experiments; however, this experiment is chosen to demonstrate the performance of combining activity recognition cues. After the activity switch, the backward filter should be run all the way back to the previous activity switch. Since there is no previous activity switch, the backward filter is run to the beginning, $k = 0$. This path is shown by the magenta-dashed line. Then, these estimates are combined to get the improved estimate, which is shown by the blue-solid line in the figure. The reconstruction almost coincides with the ground truth after the update, as confirmed by the figure.

4. 2D Experiments

4.1. Experimental setup

A total of 11 experiments are performed in 2D, in two different environments. The first set of experiments is performed outdoors on a straight line of 66 m length. The line is divided into four segments of equal length, and the endpoints of each segment are marked with a + or a × sign. The path is illustrated in figure 9.

A coordinate frame is assigned in this environment such that the line coincides with the x axis. The origin of the coordinate frame is at the leftmost point of the line. The × marks indicate possible locations to perform the 'walking-to-standing' (WS) activity switch, and the + marks indicate the locations to perform the 'walking-to-turning' (WT) activity switch. Note that point $(66, 0)$ is marked with both symbols meaning that it is possible to perform both WS and WT activity switches at this location.

In this outdoor environment, four experiments are performed:

(1) start from point $(0, 0)$, stop at $(16.5, 0)$, stop at $(49.5, 0)$, stop at $(66, 0)$;
(2) start from point $(0, 0)$, stop at $(16.5, 0)$, turn back at $(33, 0)$, stop at $(16.5, 0)$, stop at $(0, 0)$;
(3) start from point $(0, 0)$, stop at $(16.5, 0)$, stop at $(49.5, 0)$, turn back at $(66, 0)$, stop at $(49.5, 0)$, stop at $(16.5, 0)$, stop at $(0, 0)$;
(4) start from point $(0, 0)$, stop at $(49.5, 0)$, turn back at $(66, 0)$, stop at $(16.5, 0)$, stop at $(0, 0)$.

Note that it is not required to stop at every × mark, or turn back at every + mark, but these marks indicate some nonzero likelihood that these events will occur at that location.

The sports hall of Bilkent University is used as the second environment. The subjects are required to walk on lines drawn on the floor. The map of this indoor environment is shown in figure 10. Similar to the first setup, the × marks indicate possible locations to perform the standing activity. Each corner in the figure indicates a possible location to perform the turning activity. Thus, the WS and WT activity switch points on the map are assigned manually; all corners are defined as WT switch points and WS switch points are assigned arbitrarily.

The seven experiments performed in this environment are as follows:

Figure 9. The path followed in the first four experiments (all dimensions in m).

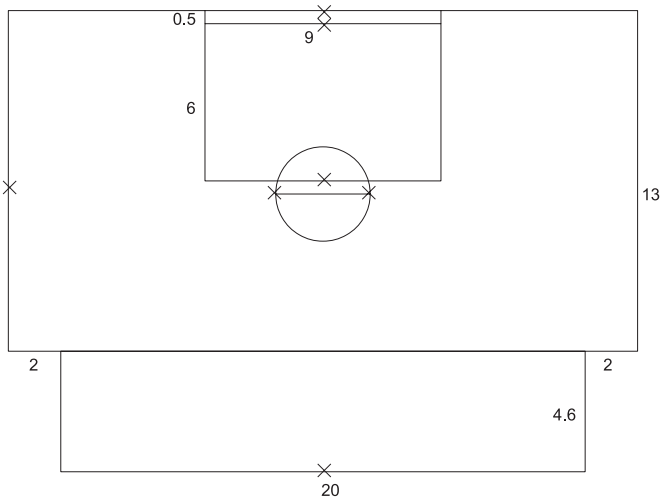

Figure 10. The path followed in the second set of experiments (all dimensions in m).

Table 1. Total path lengths of the experiments.

Experiment no	Path length (m)
1	66
2	66
3	132
4	132
5	222
6	222
7	90
8	90
9	33.9
10	96.2
11	96.2

(5) walk for three laps on a rectangle of size 24 m × 13 m;
(6) walk for three laps on a rectangle of size 24 m × 13 m, stopping at the midpoint of the longer side;
(7) walk for three laps on a rectangle of size 9 m × 6 m;
(8) walk for three laps on a rectangle of size 9 m × 6 m, stopping at the midpoint of the longer side;
(9) walk for three laps on a circle of diameter 3.6 m, stopping each time at the endpoints of the diameter;
(10) walk for one lap on a rectilinear polygon;
(11) walk for one lap on a rectilinear polygon, stopping at three different points.

The total path lengths of these experiments are tabulated in table 1. These 11 experiments are performed by four male and four female subjects, whose ages, heights and weights are presented in table 2.

Before starting the experiments, an 'alignment reset' is performed on each sensor unit to reset the coordinate frames such that the initial orientation transformation corresponds to the unit operator (that is, the initial orientation output is $\mathbf{I}_{3\times3}$ in the direction cosine matrix mode, $q = 1$ in the quaternion output mode or zero Euler angles in the Euler angle output mode), and the z axes are in the vertical direction. The top views of the global, local navigation and the sensor coordinate frames before and immediately after the alignment reset are shown in figure 4. Note that before the alignment reset, the global and local navigation frames are in their default configuration. However, at the reset instant, the x–y orientation of these frames may change arbitrarily, while their z axes remain perpendicular to the horizontal plane, opposite to the direction of the gravity vector. Immediately after the alignment reset, the local navigation and the sensor coordinate frames are coincident. All orientation outputs during the experiments are obtained with respect to the local navigation coordinate frames, illustrated in figure 4(*b*) for a single sensor unit. After the alignment reset, the sensor coordinate frames may rotate and translate with the motion of the person, whereas the global frame remains fixed and the local navigation frames may translate but not rotate.

Table 2. Profiles of the eight subjects.

Subject no	Gender	Age	Height (cm)	Weight (kg)
S1	f	32	158	45
S2	f	34	161	51
S3	m	25	180	79
S4	f	22	166	47
S5	f	24	178	60
S6	m	33	175	95
S7	m	22	187	75
S8	m	25	182	75

4.2. Experimental results

In this section, we present and compare the results of the reconstruction with and without using any activity recognition cues. We calculate the error between the reconstructed path and the true path by discretizing the true path with equally spaced points on the path, and consider either path as a finite set of points. We use a symmetric error criterion between two point sets P and Q, proposed in [37]. The well-known Euclidean distance $d(\mathbf{p}_i, \mathbf{q}_j) : \mathbb{R}^3 \times \mathbb{R}^3 \to \mathbb{R}^{\geqslant 0}$ of the ith point in the set P with the position vector $\mathbf{p}_i = (p_{xi}, p_{yi}, p_{zi})^T$ to the jth point $\mathbf{q}_j = (q_{xj}, q_{yj}, q_{zj})^T$ in set Q is given by

$$d(\mathbf{p}_i, \mathbf{q}_j) = \sqrt{(p_{xi} - q_{xj})^2 + (p_{yi} - q_{yj})^2 + (p_{zi} - q_{zj})^2}, \tag{11}$$

where $i \in \{1, \ldots, N_1\}$ and $j \in \{1, \ldots, N_2\}$. In [37], we consider and compare three different metrics to measure the similarity between two sets of points, each with certain advantages and disadvantages. In this work, we use the most favourable of them to measure the closeness or similarity between the sets P and Q:

$$\mathcal{E}_{(P-Q)} = \frac{1}{2} \times \left(\frac{1}{N_1} \sum_{i=1}^{N_1} \min_{\mathbf{q}_j \in Q} \{d(\mathbf{p}_i, \mathbf{q}_j)\} + \frac{1}{N_2} \sum_{j=1}^{N_2} \min_{\mathbf{p}_i \in P} \{d(\mathbf{p}_i, \mathbf{q}_j)\} \right). \tag{12}$$

According to this criterion, we take into account all points in the two sets and find the distance of every point in the set P to the nearest point in the set Q and average them, and vice versa. The two terms in equation (12) are also averaged, so that the criterion is symmetric with respect to P and Q.

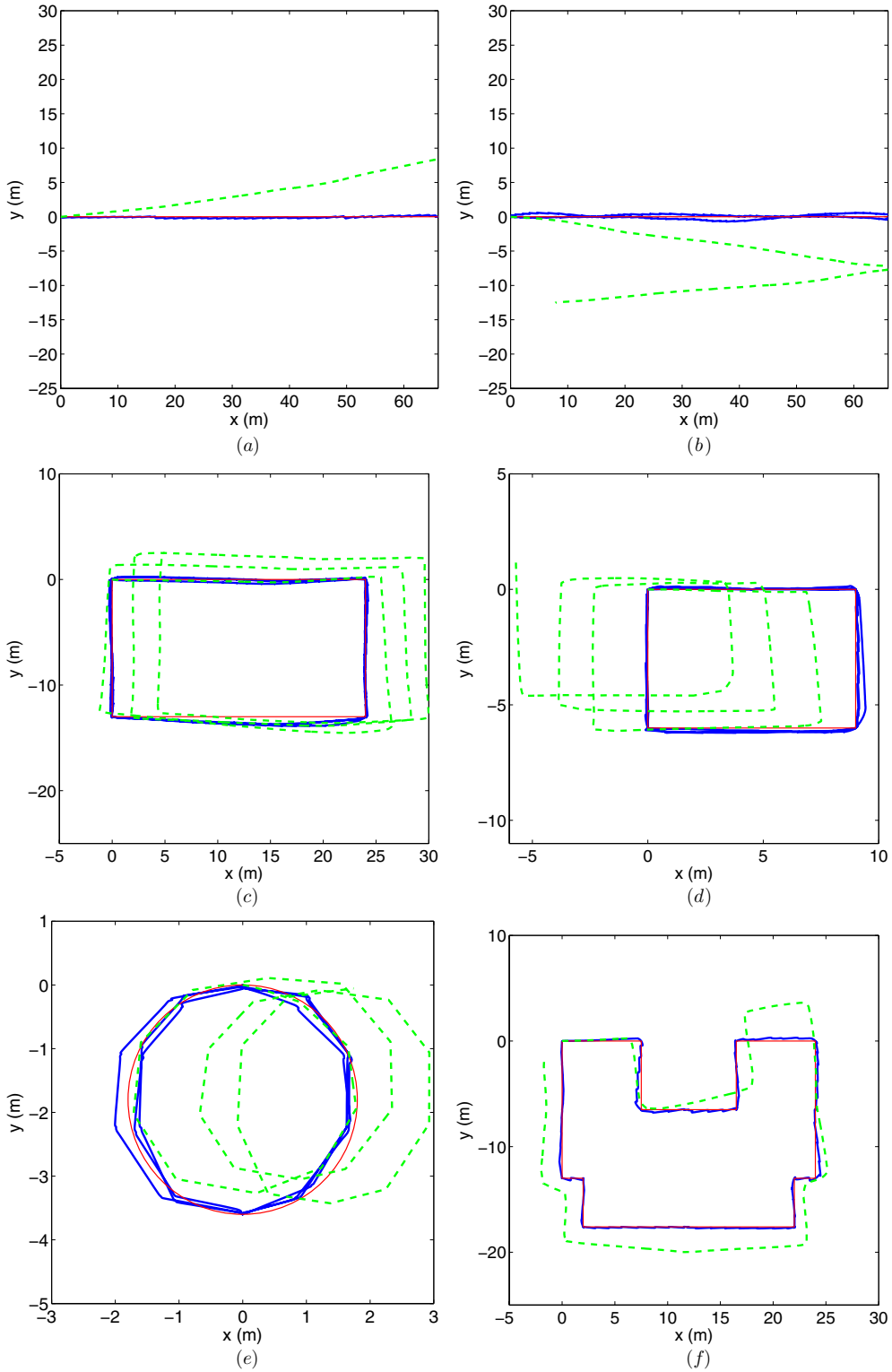

Figure 11. Sample reconstructed paths for experiments (*a*) 1, (*b*) 3, (*c*) 5, (*d*) 8, (*e*) 9, (*f*) 11, without (green-dashed line) and with (blue-solid line) activity recognition cues. The true path is indicated with the thin red-solid line.

The parameters selected for the experiments are tabulated in table 3. Each of the first set of experiments (1–4) is performed on the map given in figure 9. For the second set of experiments (5–11), we first consider each experiment separately. That is, the possible activity switch locations are not defined for the whole map, but only for the activity switch points on the walked path. Examples of reconstructed paths are presented in figure 11. In this figure, reconstructed paths without (with) activity recognition cues are shown by the green-dashed (blue-solid) line. In other words, the green-dashed line shows the result of using ZUPT only, whereas the blue-solid line shows the result of using the proposed method. It can be observed that the reconstruction improves considerably when activity recognition cues are utilized.

Table 3. Parameter values used in the experiments.

Parameter	Value
Ω_T	$1\ \text{rad}\,\text{s}^{-1}$
$\mathbf{P}_\xi(0)$	$0.01\mathbf{I}_{2\times2}$
$\mathbf{Q}$	$\begin{pmatrix} 0.01 & 0 \\ 0 & 0.1 \end{pmatrix}$
$\mathbf{P}_{WS,n}$	$0.01\mathbf{I}_{2\times2}, \quad \forall n$
$\mathbf{P}_{WT,n}$	$0.04\mathbf{I}_{2\times2}, \quad \forall n$

The errors between the true path and the reconstructed path without and with activity recognition updates are presented in tables 4 and 5, respectively. In the tables, the calculated errors using equation (12) are divided by the length of the path covered in each experiment (table 1) and then multiplied by 100 to convert to centimetres. Therefore, the values are in terms of cm m^{-1}, interpreted as centimetre error per unit metre of path length. The last columns in both tables show the averages of the other columns and represent the resulting average error in a given experiment. The reduction in the average error values by introducing activity recognition position updates is illustrated in figure 12, in which the percentage decrease in the errors can be visualized. For experiments 1–4 performed outdoors along a straight line, the average error without the updates is 1.92 $\text{cm}\,\text{m}^{-1}$. With the updates, this error is reduced to 0.14 $\text{cm}\,\text{m}^{-1}$, for which the percentage decrease in the average error can be calculated as $\frac{1.92-0.14}{1.92} \times 100 = 92.7\%$. For indoor experiments 5–11, the average error without the updates is 0.96 $\text{cm}\,\text{m}^{-1}$, which is reduced to 0.20 $\text{cm}\,\text{m}^{-1}$ after the updates. Similarly, the average percentage decrease can be calculated as $\frac{0.96-0.20}{0.96} \times 100 = 79.1\%$. On average, the error is reduced by about 85%. We also calculate the error values at the activity switch locations. That is, when a position update is performed, the corresponding error is calculated. Then, these errors are averaged, yielding the values presented in table 6. However,

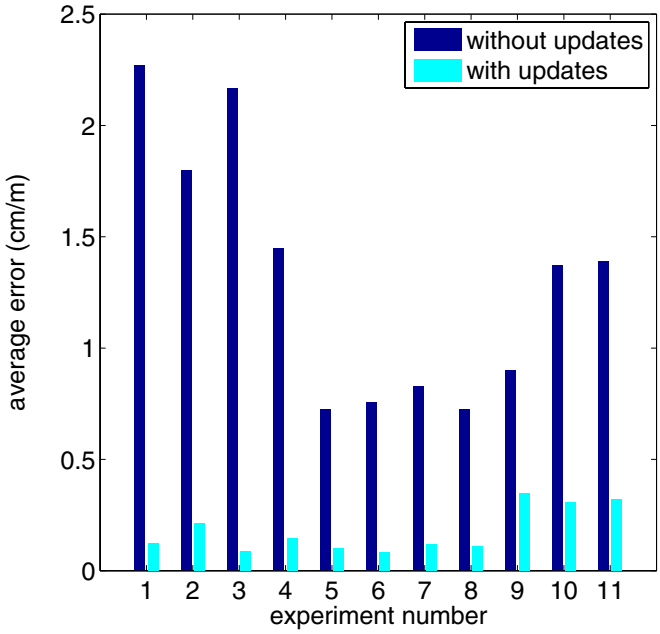

Figure 12. Average error values for all experiments without and with applying activity recognition position updates.

Table 4. Error values without activity recognition updates (in cm m^{-1}).

Experiment no	S1	S2	S3	S4	S5	S6	S7	S8	Average
1	1.21	0.31	4.33	0.71	2.56	5.30	1.02	2.71	2.27
2	3.76	4.32	1.04	1.93	1.32	0.69	0.74	0.59	1.80
3	3.70	6.26	1.17	4.32	0.67	0.46	0.21	0.54	2.17
4	1.77	1.76	1.39	3.14	0.92	0.81	1.08	0.72	1.45
5	0.45	0.87	0.21	0.67	1.31	1.00	0.77	0.53	0.73
6	0.94	1.20	0.50	0.68	0.74	0.33	1.13	0.52	0.76
7	0.56	1.92	1.00	0.64	0.30	0.30	0.75	1.16	0.83
8	0.73	0.51	0.24	1.47	0.53	0.60	1.30	0.42	0.73
9	0.84	1.04	0.83	0.65	0.95	0.49	1.29	1.12	0.90
10	1.47	1.76	1.18	1.64	1.77	0.64	0.78	1.74	1.37
11	1.35	1.31	1.17	2.09	2.40	1.26	0.77	0.78	1.39
Overall average									1.31

Table 5. Error values with activity recognition updates (in cm m^{-1}).

Experiment no	S1	S2	S3	S4	S5	S6	S7	S8	Average
1	0.10	0.11	0.11	0.12	0.11	0.18	0.10	0.14	0.12
2	0.16	0.50	0.08	0.34	0.23	0.18	0.11	0.08	0.21
3	0.08	0.16	0.04	0.13	0.09	0.06	0.08	0.05	0.09
4	0.17	0.09	0.14	0.23	0.19	0.09	0.12	0.13	0.15
5	0.10	0.15	0.09	0.12	0.07	0.11	0.09	0.09	0.10
6	0.09	0.13	0.04	0.10	0.11	0.08	0.09	0.04	0.08
7	0.06	0.20	0.08	0.11	0.12	0.14	0.15	0.09	0.12
8	0.10	0.18	0.09	0.15	0.08	0.14	0.08	0.06	0.11
9	0.21	0.29	0.62	0.15	0.22	0.55	0.55	0.19	0.35
10	0.16	0.30	0.20	0.45	0.22	0.23	0.44	0.48	0.31
11	0.64	0.13	0.12	0.30	0.14	0.13	1.02	0.10	0.32
Overall average									0.18

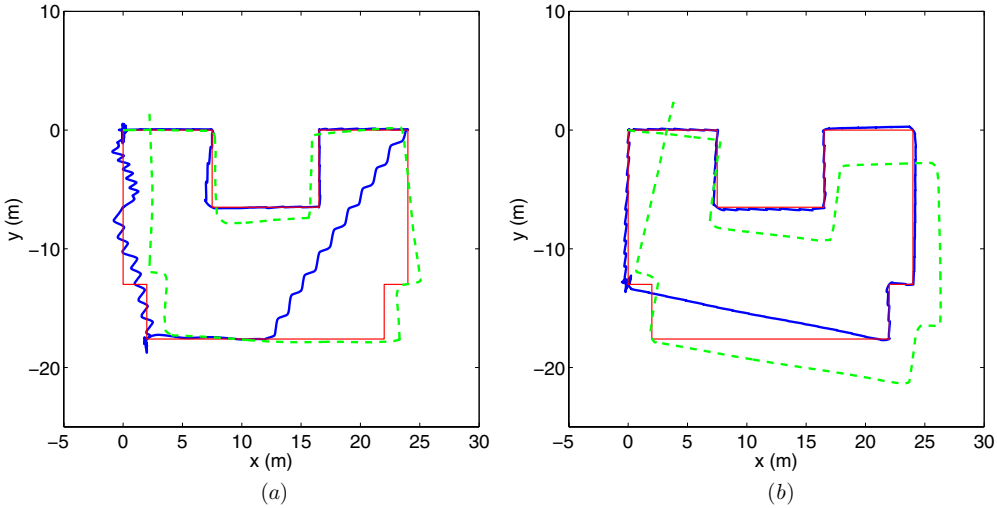

Figure 13. Incorrectly reconstructed paths caused by (*a*) incorrect activity recognition and (*b*) offsets in sensor data.

Table 6. Averaged position errors at the position update locations (in cm m^{-1}).

Experiment no	S1	S2	S3	S4	S5	S6	S7	S8	Average
1	1.48	0.48	3.00	4.21	3.29	4.34	2.65	3.36	2.85
2	2.76	5.63	2.04	4.62	1.86	1.51	2.59	2.06	2.88
3	1.61	4.62	1.25	3.29	0.91	0.98	0.65	1.22	1.82
4	2.04	2.32	1.22	6.29	1.21	1.38	2.93	2.19	2.45
5	1.41	0.99	0.41	1.78	0.73	1.63	1.14	1.02	1.14
6	2.50	0.94	0.51	1.04	0.69	0.64	0.57	0.79	0.96
7	0.45	5.53	0.74	1.42	0.63	1.22	0.81	0.96	1.47
8	0.59	0.87	0.95	1.40	0.69	0.92	0.83	0.70	0.87
9	1.39	1.05	4.55	1.33	1.27	3.27	2.40	1.19	2.06
10	1.19	1.43	1.01	1.90	1.00	0.75	1.44	1.54	1.28
11	1.79	1.29	0.73	2.04	1.03	0.80	5.32	0.89	1.73
Overall average									1.77

in a few cases, the positions are not updated to the correct location, as explained below.

The activity recognition performance is perfect for the WS switches, i.e. all WS switches are correctly recognized for all subjects in all experiments. Some instantaneous false alarms (type I errors[1]) are observed but they have been eliminated by employing a simple median filter. For the WT switches, no false alarms are observed. However, some of the WT activity switches are not correctly recognized (type II errors[2]), since the thresholds are not set individually for each subject. These type II errors in WT switches sometimes cause the subsequent updates to be made at incorrect locations, such as the example shown in figure 13(*a*). Here, the two WT switches while walking on the lower-right corner in the figure are not correctly detected. Over the $8 \times 11 = 88$ experiments performed in this part, this problem occurs only once. Even if there is no incorrect detection of activity, the same problem can still occur, as shown in figure 13(*b*). Here, the offset in the angle measurement causes the forward filter to diverge from the actual path, and when a WT switch is detected, the calculated closest WT switch point (equation (9)) is not the actual turning point. This phenomenon is observed five times in all 88 experiments.

For experiments 5–11, we also reconstruct the paths using the whole of the map in figure 10. That is, we define all corners on the map as WT switch points, and the points marked with × as WS switch points. The error values without activity recognition updates are the same as in table 4. The results with activity recognition updates are given in table 7, and the changes in the average error are given as a bar chart in figure 14. The average errors for most of the experiments are reduced for this case as well, with the exception of the experiment involving walking on a circle (experiment 9). In table 7, it can be observed that the errors have increased only for three of the subjects. In these cases, the paths are not correctly reconstructed. This is caused by the fact that the circle experiment involves continuous turning activity, although not as sharp as turning at the corners. In fact, the thresholds for detecting turning activity should be chosen such that the slow turning motion on the circular path is not detected as an activity switch, but the sharp turning motion at the corners is detected. This will, of course, depend on the radius of curvature of the circle, and the smaller it is, the larger will be the error. Based on the experimental results, we can state that it is not

[1] In the context of this work, a type I error means that an activity switch has not actually occurred, but the recognition algorithm falsely detects that it has occurred.

[2] Conversely, a type II error means that an activity switch has actually occurred, but the recognition algorithm fails to detect the activity switch. These terms are borrowed from the statistics terminology.

Table 7. Error values with activity recognition updates using the whole map (in cm m^{-1}).

Experiment no	S1	S2	S3	S4	S5	S6	S7	S8	Average
5	0.13	0.15	0.09	0.12	0.65	0.44	0.09	0.09	0.22
6	0.39	0.20	0.04	0.25	0.20	0.08	0.17	0.04	0.17
7	0.07	0.80	0.16	0.16	0.18	0.21	0.17	0.13	0.23
8	0.21	0.28	0.18	0.20	0.13	0.25	0.08	0.09	0.18
9	0.77	0.29	16.21	0.15	7.99	2.32	0.71	0.20	3.58
10	0.16	0.29	0.18	0.45	0.20	0.23	0.44	0.49	0.31
11	0.79	0.13	0.13	0.31	0.14	0.12	2.55	0.10	0.53
Overall average									0.75

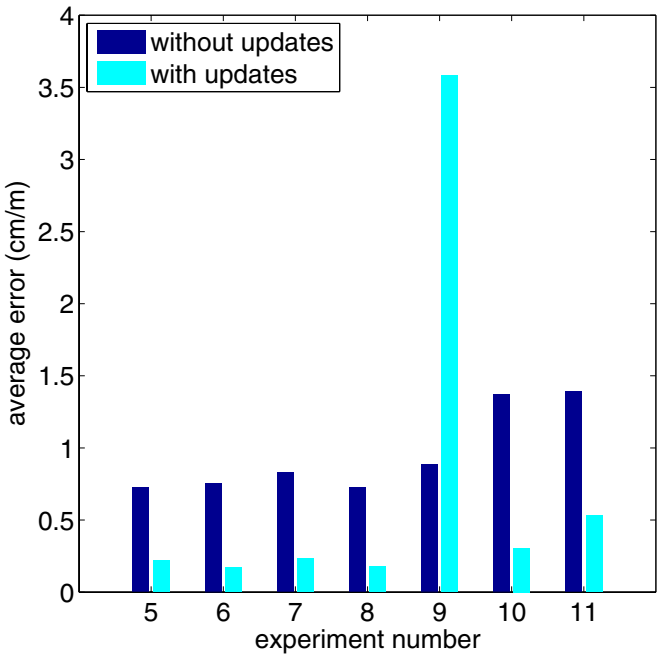

Figure 14. Average error values for experiments 5–11 without and with activity recognition position updates when the whole map is used.

possible to choose a single threshold that performs perfectly for all subjects, because every subject performs the walking motion uniquely in his/her own style. This problem can easily be solved by introducing uniformly spaced WT switch points on the circle. By defining 36 additional WT switch points on the circle that are 10° apart, we reduce the average error to 0.32 cm m^{-1}. However, since the radius of curvature of the circle in this experiment is too small and such sharp turns would very rarely be encountered on locations other than corners in a realistic situation, such a procedure would not be necessary in most cases. Sample reconstructions for this method are shown in figure 15.

After introducing these additional WT switch positions, the errors between the true and reconstructed paths are given in table 8, and the average position errors at the update locations are given in table 9. In this case, the average error without the updates is again 0.96 cm m^{-1}, which is reduced to 0.28 cm m^{-1} using the activity updates and defining new WT switch points on the circle. In other words, the percentage reduction in the average error is $\frac{0.96-0.28}{0.96} \times 100 = 70.8\%$.

Note that the errors of experiments 10 and 11 increased slightly after the addition of more WT switch locations. This is illustrated in the reconstruction in figure 15(*d*), which belongs to the same experiment as in figure 11(*f*). Here, it can be observed that the performance of the latter is better. The degradation in the performance of the former results from the addition of more WT switch points on the circle in order to improve the incorrect reconstructions of the circular path. This causes the closest WT switch point (equation (9)) to differ from the actual turning point in figure 15(*d*). This means that the addition of more switch points may cause degradations in the performances of other path reconstructions and may affect the overall error negatively. Therefore, using more activity switch points on a map does not necessarily improve the overall performance.

5. 3D experiments and results

5.1. Experiment in indoor building environment

To demonstrate the applicability of our method in a realistic setting, we performed an experiment on two consecutive floors of an indoor environment. The experiments are conducted in the Electrical and Electronics Engineering building on the Bilkent University campus.

In addition to the walking, standing and turning activities of the 2D experiments, we introduce the stairs activity in the 3D experiments. We denote the walking-to-stairs and stairs-to-walking activity switches as a WR switch, using a single label. This is because at each walking-to-stairs switch location, a stairs-to-walking switch can also occur, and vice versa. In other words, walking-to-stairs and stairs-to-walking activity switch locations correspond to the same points on a given map.

This experiment is performed by subjects S1, S3 and S8. In [30], we demonstrated that including training data from an individual improves the classification performance considerably. This is also confirmed in this study. The subject S8 in this study was also one of our test subjects in [30], and the best classification performance in this experiment is achieved with subject S8.

The activity recognition performances are presented in figure 16. The blue thick lines in the figures represent the activity detected by the *k*-NN classifier, and the red thin lines represent the true activity, which is determined manually by observing the signals and the video recording of the experiment. We count the number of samples where the true activity is the same as the recognized activity and

Table 8. Error values with activity recognition updates using the whole map, after defining more WT switch locations (in cm m^{-1}).

Experiment no	S1	S2	S3	S4	S5	S6	S7	S8	Average
5	0.13	0.15	0.09	0.12	0.65	0.44	0.09	0.09	0.22
6	0.39	0.20	0.04	0.25	0.20	0.08	0.17	0.04	0.17
7	0.07	0.79	0.16	0.16	0.18	0.21	0.17	0.13	0.23
8	0.21	0.28	0.18	0.20	0.13	0.25	0.08	0.09	0.18
9	0.22	0.29	0.34	0.15	0.19	0.46	0.71	0.20	0.32
10	0.16	0.37	0.18	0.45	0.20	0.23	0.44	0.49	0.32
11	0.88	0.21	0.13	0.31	0.14	0.12	2.54	0.10	0.55
Overall average									0.28

Table 9. Averaged position errors at the position update locations (in cm m^{-1}).

Experiment no	S1	S2	S3	S4	S5	S6	S7	S8	Average
5	1.41	0.99	0.41	1.78	2.19	1.63	1.14	1.02	1.32
6	2.34	0.89	0.51	1.04	0.67	0.64	0.65	0.79	0.94
7	0.45	4.00	0.70	1.37	0.63	1.14	0.79	0.89	1.25
8	0.50	0.76	0.89	1.30	0.70	0.88	0.83	0.62	0.81
9	1.15	1.05	2.12	1.33	0.92	2.19	1.78	1.19	1.47
10	1.19	1.43	0.92	1.90	1.02	0.75	1.44	1.54	1.27
11	1.88	1.40	0.73	2.02	1.03	0.81	4.04	0.89	1.60
Overall average									1.24

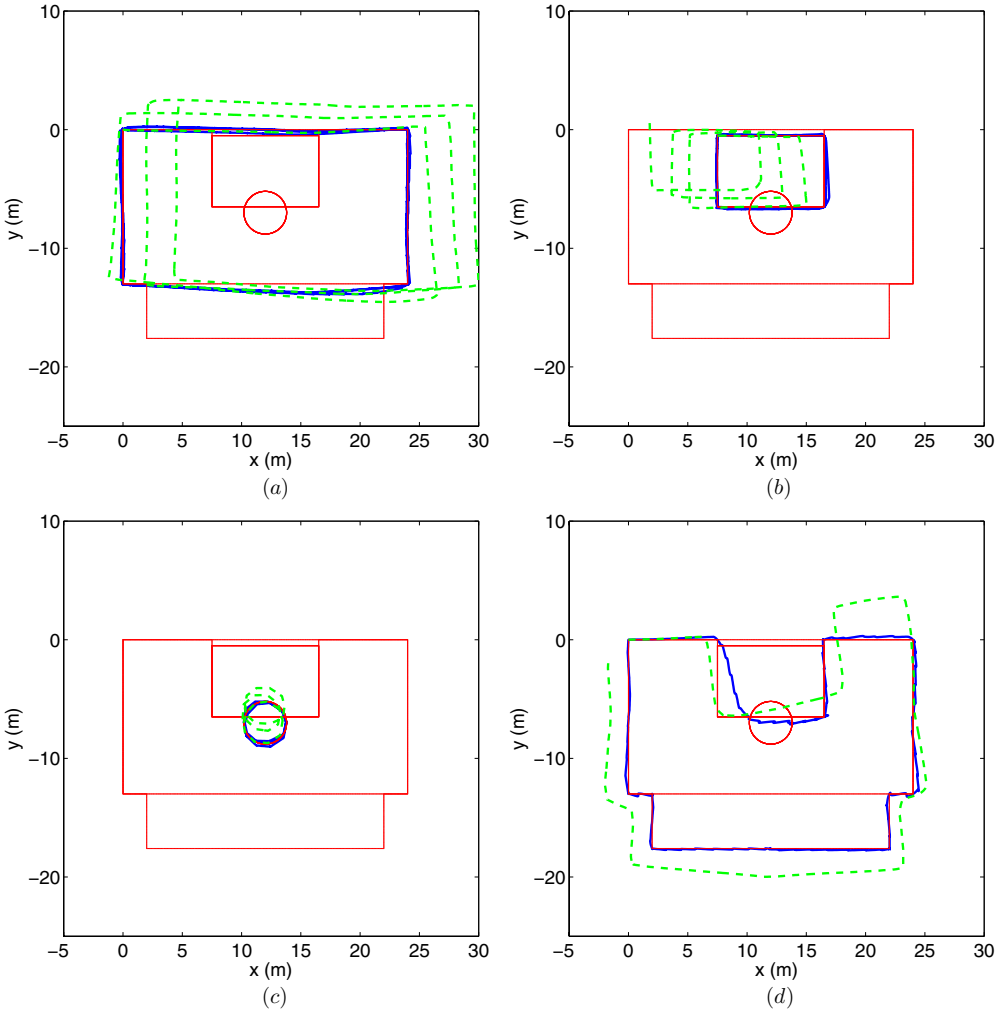

Figure 15. Sample reconstructed paths for experiments (*a*) 5, (*b*) 8, (*c*) 9, (*d*) 11, without (green-dashed line) and with (blue-solid line) activity recognition cues on the whole map. The true path is indicated with the thin red-solid line.

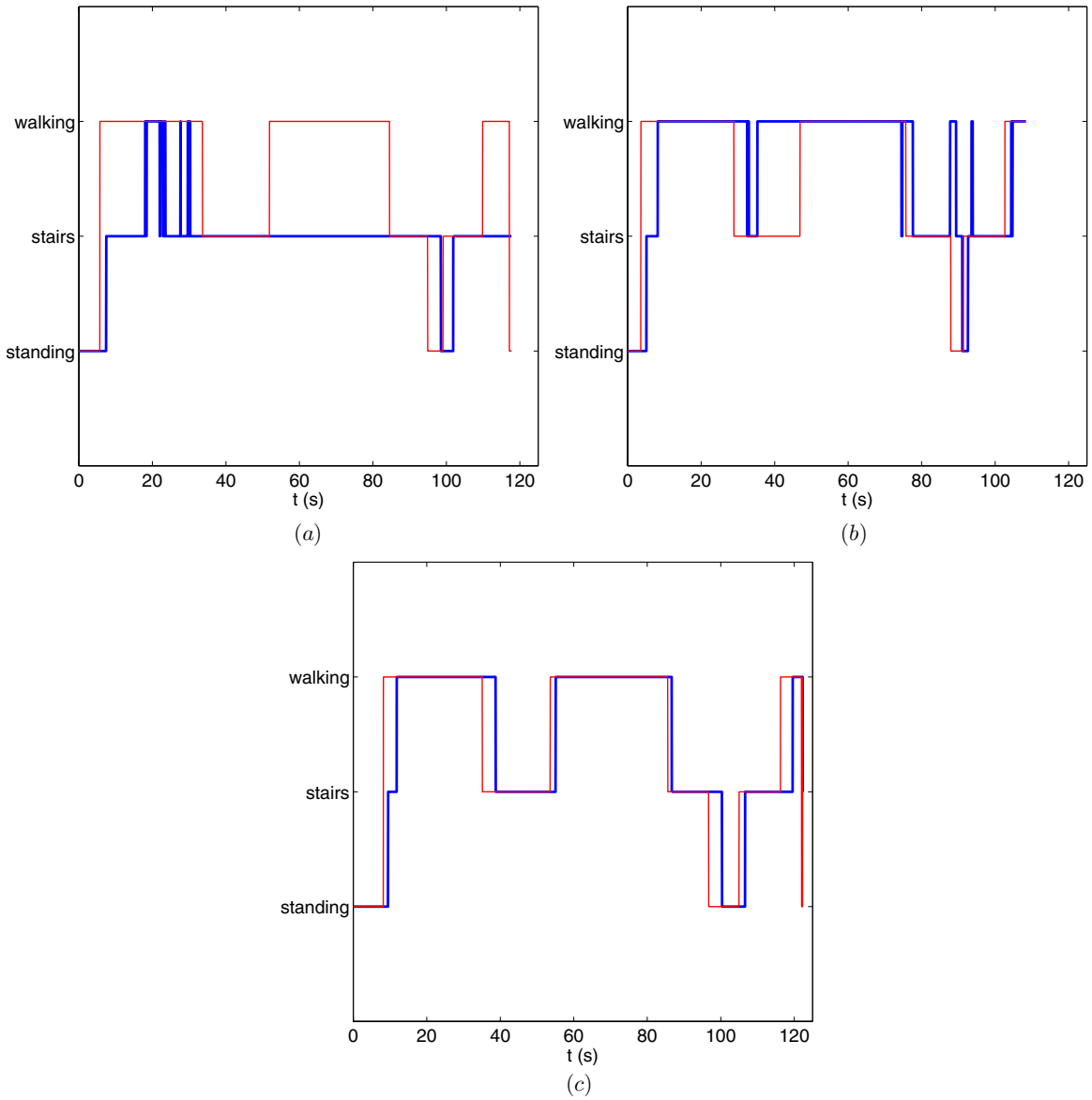

Figure 16. Activity recognition performance for subjects (*a*) S1, (*b*) S3 and (*c*) S8. The blue thick lines represent the activity recognized by the *k*-NN classifier, whereas the red thin lines represent the true activity determined manually.

divide this number by the total number of samples to evaluate the activity recognition performance. The performance is found to be 40.7% for S1, 73.0% for S3 and 84.6% for S8. We conclude that the performance of S8 is the best because the training data of S8 are already available to the *k*-NN classifier. Since the profiles (such as age, height and weight) of S3 and S8 are similar (table 2), the activity recognition performance of subject S3 is also good. The mediocre performance for S1 can be explained by the fact that the profiles of the subjects in the training data do not resemble the profile of subject S1. The profiles of the subjects in the training data can be found in [38].

The results of the reconstruction before and after activity recognition updates are presented in figure 17. In the figure, the red thin line represents the true path, the green-dashed line represents the reconstructed path without activity recognition updates and the blue-solid line represents the reconstructed path after applying the activity recognition updates. We also run the localization algorithm assuming that the activity recognition performance is perfect, i.e. we use the red thin lines in figure 16 as the activity recognition result. The reconstruction with this approach is shown in the black-dash-dotted line. The localization result improves with accurate activity information as expected, indicating that the more accurate the activity recognition is, the more accurate will be the position estimation.

We set the initial position of the subject as the origin, and the initial walking direction as the x direction. In this setting, the only WS switch point is $(0, 0, 0)$. We do not introduce any additional artificial WS switch locations since this experiment is performed in a realistic environment. The WT and WR switch points are presented in table 10 in matrix form for compactness, whose rows correspond to the coordinates of activity switch locations. These locations are determined considering the walked path and the construction

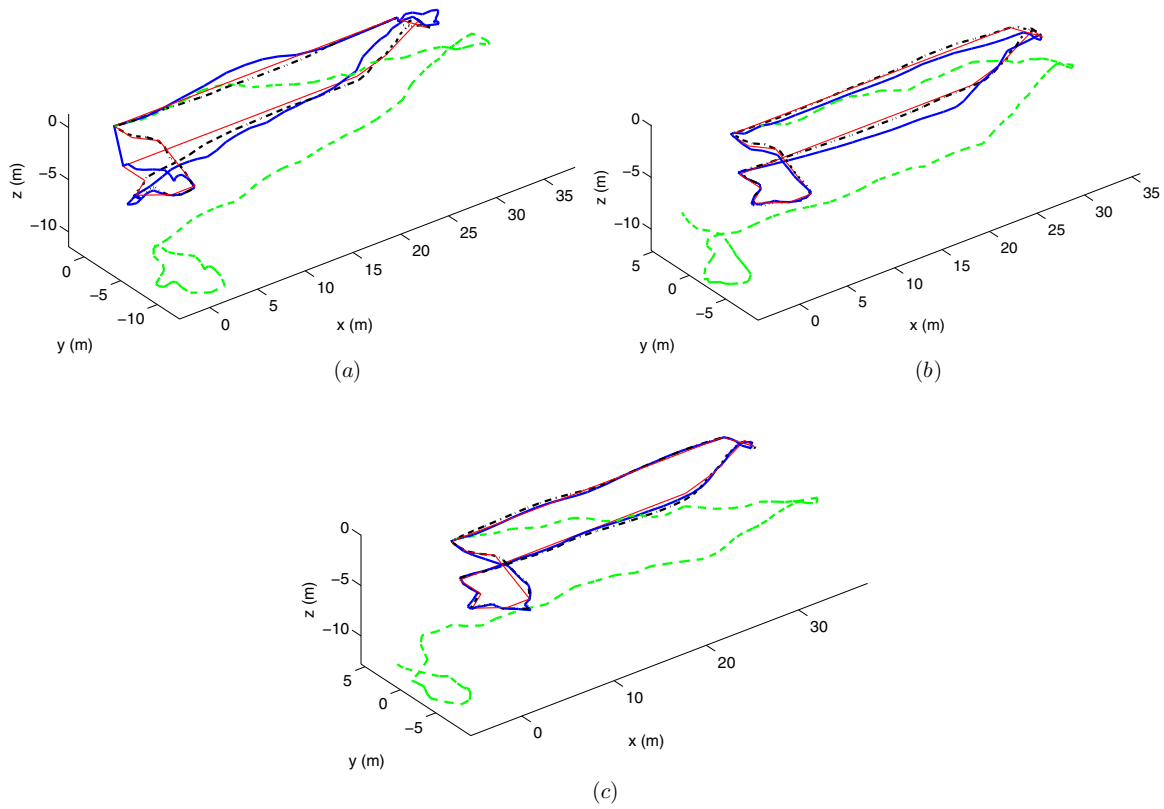

Figure 17. Localization results for subjects (*a*) S1, (*b*) S3 and (*c*) S8. The reconstructions are calculated with ZUPT only (green-dashed line), using *k*-NN activity recognition updates (blue-solid line) and using the true activity recognition updates (black-dash-dotted line). The thin red-solid line shows the true path.

Table 10. Walking-to-turning (WT) and walking-to-stairs (WR) activity switch locations.

WT			WR		
0	0	0	29.40	0	0
32.78	0	−2.08	29.40	1.30	−4.16
32.78	1.30	−2.08	−1.20	−4.50	−4.16
0.90	0	−4.16	1.50	−4.20	0
0.90	−3.00	−4.16	−0.90	−9.10	−2.08
−1.20	−4.50	−4.16	1.50	−9.10	−2.08
−0.90	−9.10	−2.08			
1.50	−9.10	−2.08			
1.50	−4.20	0			
0	−2.40	0			

plans of the building. We also used a tape measure to determine the coordinates of some of the waypoints on the path.

Several heuristics are used in the simultaneous localization and activity recognition process in 3D. We observed that there are some instantaneous WR switches while the subject is ascending or descending stairs (figure 16(*a*)). That is, occasionally the activity classifier instantaneously decides that the subject is walking although he is actually on the stairs. The converse also occurs, i.e. the classifier detects the 'stairs' activity, while the subject is walking on the level floor. To avoid an incorrect position update at these instants, we introduce a condition on the WR switches such that the switched activity (in this case, walking) must go on for at least 3 s for a position update to be applied. Another heuristic is that if the current activity is detected as walking, we do not modify the position in the z direction in the prediction equation. This is fair because on the given map, walking activities only take place on the horizontal plane. If a map was given with possible uphill or downhill walking platforms (which is quite unlikely in an indoor building environment), this rule would lead to incorrect results and should not be used.

Table 11. Error values for the 3D experiment.

Subject no	ZUPT error	Error with *k*-NN	Error with perfect activity recognition
S1	4.82	1.11	0.48
S3	4.80	0.48	0.26
S8	5.84	0.35	0.33

As shown in figure 17, the path reconstruction is almost perfect for S8 after introducing the updates. Using the error measure in equation (12), we calculate the errors between the reconstructed paths and the true path. These error values are given in table 11. Here, it can be observed that the errors decrease considerably when activity recognition updates are introduced. The average ZUPT error is 5.15 cm m^{-1}, which is reduced to 0.65 cm m^{-1} with the *k*-NN activity recognition updates. This corresponds to a decrease of 87% in the average error. For S8, whose training data are available, the decrease is 94%. Therefore, it can be concluded that, in general, improved

activity recognition performance results in a larger decrease in the error. The last column in the table gives the error values if the activity recognition were done perfectly, i.e. it corresponds to the error between the red thin line and the black-dash-dotted line. The reason for the degradation in the activity recognition performance is that each person has a different style of walking on the stairs as well as on a straight path. Distinguishing between walking and stairs activities is not possible with high accuracy if the classifiers are trained with the data of other subjects. Therefore, in a practical application, the classifier must be trained with the data of the user, which is an operation to be performed only once. Then, our simultaneous localization and activity recognition method can be used, which improves the localization performance by reducing positioning errors about 90%. However, in general, we would like to note that if physical features such as height and weight of the training and test subjects are similar, the classification results improve.

The results of our 3D experiments suggest that if the classifiers are trained with data from a person with similar physical features to the person to be localized, the performances of both the localization and activity recognition processes improve. In the 2D experiments where a simple rule-based activity classifier is used, there seems to be no correlation between the physical features of the participants and the localization performance.

5.2. Experiment on spiral stairs

To test the performance of the 3D algorithm with continuous turning activity, we also performed an experiment on spiral stairs with subject S8. The subject ascends the stairs on a fire escape for eight storeys. We detect the turning activity using the rule-based algorithm in our 2D experiments. Even though there is continuous turning activity, the preset threshold defined in the rule-based algorithm is exceeded only occasionally, resulting in a stairs-to-turning (RT) activity switch. Therefore, 80 equally spaced RT activity switch locations are defined on the spiral stairs. The results are presented in figure 18. Similarly, the green-dashed and blue-solid lines represent the reconstructed path without and with activity recognition updates, respectively. The thin red-solid line represents the actual path. For this experiment, the error is decreased from 2.08 to 0.24 cm m^{-1} with the activity recognition updates, resulting in 88% error reduction.

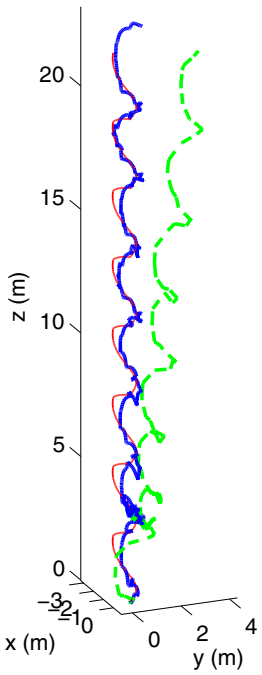

Figure 18. Sample reconstructed path for the spiral stairs.

6. Discussion

The proposed method and its experimental verification demonstrate that activity recognition provides useful cues for localization when combined with a known map of the environment. Path reconstruction improves significantly when the activity switch cues are used for position updates so that localization is performed simultaneously with activity recognition. Considering the whole of the maps for both sets of 2D experiments, the average percentage decrease in the error is 79%. The errors at the final point of the experiments are zero for all experiments since the subjects stop at the end of the experiment at a WS switch point where a final position update is performed.

The errors calculated using equation (12) represent the average distance between the true and the reconstructed paths. This is a spatial error measure between two sets of points that comprise the curves. If the true position of the subject as a function of time were available, a more reliable error criterion would be to calculate the error between the true and the estimated positions at all time values, and then to take the time average. However, in our experiments, the true positions of the subjects are not available. Obtaining accurate true position data as a function of time is a difficult task outdoors because low-cost handheld GPS equipment has accuracies in the order of several metres. In indoor environments, it might be necessary to configure accurate WiFi- or RFID-based positioning systems.

In our experiments, we have observed mainly two phenomena as the source of path reconstruction errors. These two phenomena impose some limitations on the potential applications of our method.

Some of the errors are caused by incorrect activity recognition. This can be observed either in the form of incorrect position updates (caused by type I errors) or in the form of prevention of a required position update from being made (caused by type II errors). An example of the latter is shown in figure 13(*a*). Our method can fail to reconstruct the path correctly if such errors are likely to occur. However, if the activities defined are sufficiently well differentiated, or more precisely, if the selected features for different activities are well separated in the feature space, activity recognition errors can be reduced considerably. In real-life applications, features should be extracted in a way to make the activities easily differentiable. Distinguishing between similar activities such as ascending/descending stairs and walking is not an

easy task that can be accomplished by using simple rule-based classifiers and may require more complex classifiers. To improve the performance, training data of the test subject should be made available to the classifiers; otherwise similar activities can be confused.

The number of activity classes should be selected according to the environmental setting. For example, underground mines are one of the most suitable environments to apply the methods developed in this study since it is not possible to use external positioning methods such as GPS. By defining activities specific to the given setting and implementing suitable activity recognition techniques, accurate positioning can be achieved underground. Activities such as putting a helmet on and taking it off, operating machinery, walking, sitting, resting, etc can be defined. These activities can be easily distinguished using sensor units at different positions on the body and properly selected features. Potential applications exist in other GPS-denied environments such as inside buildings, tunnels, underground or dense forests and around tall buildings in urban areas where GPS data are not accurate or always available. In this study, we consider a rather limited set of daily activities such as walking, standing and turning, because our purpose here is to demonstrate the potential of incorporating activity recognition cues in a PDR system that can be used in any general setting. We conducted controlled and well-defined experiments and gave the participants accurate instructions on where to perform these activities on the map. However, in a practical setting, standing and turning activities can be performed irregularly at any location. In this case, a separate class for such irregular motions can be defined as in [39] and false position updates can be prevented by training the classifiers properly.

Errors are also caused by finding and matching the *nearest* predefined activity switch point to the forward estimate when an activity switch is detected (equation (9)). One example is illustrated in figure 13(*b*). When the position update intervals are long, the errors in estimated heading and estimated distance can make the forward position estimate drift away towards a predefined activity switch point which is different from the correct one. To avoid these errors, the map data should be interpreted and processed intelligently when defining the activity switch points. The predefined activity switch points on the map should be sufficiently close to each other so that the distance between position updates is not too large. However, increasing the number of possible activity switch locations on a map does not always improve the localization results. If the points are too close, the drifts in the estimated position and/or heading may lead to incorrect position updates as shown in figure 13(*b*), which might not have occurred if there were fewer activity switch points. Even though the number of correct position updates increases with increasing number of activity switch locations, the number of incorrect position updates is expected to increase as well. Therefore, the separation between the activity switch points should be optimized to minimize the error. Methods of manual or automated determination of the number and positions of activity switches need to be developed. Activity switch points exist naturally in many environments. If it is possible to observe the environment through the use of camera surveillance systems, activity switch locations can be determined by analysing the activity patterns of individuals over sufficiently long periods of time. This could be easily done in indoor office environments, shopping malls and factories for instance. In urban outdoor settings, critical road junctions with pedestrian crossings and traffic lights can be observed. The positions of some activity switches are well defined with small uncertainty. For example, transitions from walking to ascending/descending stairs and vice versa, taking the lift and opening/closing doors often take place at well-known positions. Daily activities such as washing hands, brushing teeth and bathing also take place at well-defined locations. However, locations and timing of some activities such as stopping, turning, kneeling, picking up something from the floor may be more uncertain and irregular. It would be more advantageous to exploit and benefit from those activity switch locations with smaller uncertainty by specifying them as position update locations. Both manual and automated procedures should also take into account possible activity misclassifications for a given map as well as the drift characteristics of the sensors used. These extensions of the work reported here deserve dedicated and detailed further investigation.

7. Conclusion

In this paper, we present the results of introducing activity recognition cues to perform position updates in a pedestrian dead reckoning (PDR) system, thus performing localization and activity recognition simultaneously. We use the well-known ZUPT method to estimate the distance travelled, and an orientation sensor mounted on the chest to estimate the heading. Position errors occur because of the characteristic offset or the bias errors at the output of these inertial sensors. Other causes of error are initial misplacement of the sensor units and slippage on the body during operation because of loose mounting. These errors are corrected using a map of the environment with predefined activity switch locations at which position updates are performed. We demonstrate that it is possible to perform pedestrian localization without any external aid other than a map. Experimental verification in indoor and outdoor environments is provided. When a current position update is performed using the activity recognition cues, optimal estimation techniques are used also to correct the past estimates. Activity recognition cues considerably improve the performance of the PDR system, reducing the position errors drastically. If training data of the subjects are available, it is possible to reduce the localization error by about tenfold when activity recognition cues are utilized simultaneously in localization.

Similar to the way activity recognition cues aid the localization process in our approach, location cues can be utilized to improve the accuracy of the activity recognition process. Even though activity recognition is quite accurate when training and test data from the same user are available, the accuracy degrades when training data from the same user are not available. However, for a given position on a map, the possible set of activities that can be performed is

limited. By utilizing the location information in the activity recognition algorithm, this problem can be eliminated. In this context, localization and activity recognition can be envisioned as a loop, very much like simultaneous localization and mapping (SLAM) methods in mobile robotics, where activity recognition aids localization and vice versa.

Acknowledgments

This work was supported by the Scientific and Technological Research Council of Turkey (TÜBİTAK) under grant number EEEAG-109E059. The first author gratefully acknowledges the support of a doctoral scholarship from this grant and from TÜBİTAK-BİDEB during this study. The authors would like to thank the anonymous subjects who performed the experiments.

Appendix. Combining unbiased estimators

Suppose $\hat{\mathbf{x}}_1$ and $\hat{\mathbf{x}}_2$ are two estimators of a random vector $\mathbf{X}$, with error covariances $\boldsymbol{\Sigma}_1 = E[(\hat{\mathbf{x}}_1 - \mathbf{X})(\hat{\mathbf{x}}_1 - \mathbf{X})^T]$ and $\boldsymbol{\Sigma}_2 = E[(\hat{\mathbf{x}}_2 - \mathbf{X})(\hat{\mathbf{x}}_2 - \mathbf{X})^T]$, respectively. We assume that the estimators are unbiased, i.e.

$$E(\hat{\mathbf{x}}_1 - \mathbf{X}) = E(\hat{\mathbf{x}}_2 - \mathbf{X}) = 0 \tag{A1}$$

and the estimation errors are uncorrelated, i.e.

$$E[(\hat{\mathbf{x}}_1 - \mathbf{X})(\hat{\mathbf{x}}_2 - \mathbf{X})^T] = \mathbf{0}. \tag{A2}$$

A linear combination of these two estimators is given as

$$\hat{\mathbf{x}} = \mathbf{W}_1\hat{\mathbf{x}}_1 + \mathbf{W}_2\hat{\mathbf{x}}_2, \tag{A3}$$

where $\mathbf{W}_1$ and $\mathbf{W}_2$ are weighting matrices.

We would like to find the optimal weighting matrices such that the covariance matrix of the combined estimate given by

$$\boldsymbol{\Sigma} = E[(\hat{\mathbf{x}} - \mathbf{X})(\hat{\mathbf{x}} - \mathbf{X})^T] \tag{A4}$$

is 'minimum', which should be interpreted in the sense of the following definition [40].

Definition 1. *Let $\boldsymbol{\Sigma}_1$ and $\boldsymbol{\Sigma}_2$ be symmetric non-negative definite matrices such that the difference $\boldsymbol{\Sigma}_2 - \boldsymbol{\Sigma}_1$ is non-zero and non-negative definite. Then, $\boldsymbol{\Sigma}_1$ is said to be less than $\boldsymbol{\Sigma}_2$.*

If the combined estimate is to be unbiased, the following condition should hold:

$$E(\hat{\mathbf{x}}) = \mathbf{W}_1E(\hat{\mathbf{x}}_1) + \mathbf{W}_2E(\hat{\mathbf{x}}_2) = E(\mathbf{X}). \tag{A5}$$

Since $E(\hat{\mathbf{x}}_1) = E(\hat{\mathbf{x}}_2) = E(\mathbf{X})$, the condition reduces to $\mathbf{W}_1 + \mathbf{W}_2 = \mathbf{I}$, where $\mathbf{I}$ is the identity matrix of appropriate size. The combined estimate can be expressed as

$$\hat{\mathbf{x}} = \mathbf{W}_1\hat{\mathbf{x}}_1 + (\mathbf{I} - \mathbf{W}_1)\hat{\mathbf{x}}_2 \tag{A6}$$

and the optimization criterion is given as

$$\begin{aligned}\mathbf{W}_1^* &= \arg\min_{\mathbf{W}_1} \boldsymbol{\Sigma} \\ &= \arg\min_{\mathbf{W}_1} \mathbf{W}_1(\boldsymbol{\Sigma}_1 + \boldsymbol{\Sigma}_2)\mathbf{W}_1^T - \mathbf{W}_1\boldsymbol{\Sigma}_2 - \boldsymbol{\Sigma}_2\mathbf{W}_1^T + \boldsymbol{\Sigma}_2.\end{aligned} \tag{A7}$$

In this case, the minimizing $\mathbf{W}_1$ can be found as [40]

$$\mathbf{W}_1^* = \boldsymbol{\Sigma}_2(\boldsymbol{\Sigma}_1 + \boldsymbol{\Sigma}_2)^{-1} = \left(\boldsymbol{\Sigma}_1^{-1} + \boldsymbol{\Sigma}_2^{-1}\right)^{-1}\boldsymbol{\Sigma}_1^{-1} \tag{A8}$$

and $\mathbf{W}_2^*$ is given as

$$\mathbf{W}_2^* = \mathbf{I} - \mathbf{W}_1^* = \left(\boldsymbol{\Sigma}_1^{-1} + \boldsymbol{\Sigma}_2^{-1}\right)^{-1}\boldsymbol{\Sigma}_2^{-1}. \tag{A9}$$

Therefore, the optimal combination of estimators $\hat{\mathbf{x}}_1$ and $\hat{\mathbf{x}}_2$ is

$$\hat{\mathbf{x}} = \left(\boldsymbol{\Sigma}_1^{-1} + \boldsymbol{\Sigma}_2^{-1}\right)^{-1}\left(\boldsymbol{\Sigma}_1^{-1}\hat{\mathbf{x}}_1 + \boldsymbol{\Sigma}_2^{-1}\hat{\mathbf{x}}_2\right) \tag{A10}$$

and the error covariance can be calculated as

$$\boldsymbol{\Sigma} = \left(\boldsymbol{\Sigma}_1^{-1} + \boldsymbol{\Sigma}_2^{-1}\right)^{-1}. \tag{A11}$$

Note that in equation (A10), individual estimates are weighted by the inverses of the corresponding covariance matrices. This result also agrees with physical intuition. For example, if one estimate is more reliable than the other (i.e. its covariance matrix is less than the other), its weight will be larger compared to the other weight, and vice versa.

References

[1] Bowditch N 1995 *The American Practical Navigator* (Bethesda, MD: National Imagery and Mapping Agency)
[2] Ayliffe A 2001 The development of airborne dead reckoning. Part I: before 1940—finding the wind *J. Navig.* **54** 223–33
[3] Ayliffe A 2001 The development of airborne dead reckoning. Part II: after 1940—staying on track *J. Navig.* **54** 463–76
[4] Skog I and Händel P 2009 In-car positioning and navigation technologies—a survey *IEEE Trans. Intell. Transp. Syst.* **10** 4–21
[5] Borenstein J and Feng L 1996 Measurement and correction of systematic odometry errors in mobile robots *IEEE Trans. Robot. Autom.* **12** 869–80
[6] Titterton D H and Weston J L 2004 *Strapdown Inertial Navigation Technology* 2nd edn (London: IEE)
[7] Sukkarieh S, Nebot E M and Durrant-Whyte H F 1999 A high integrity IMU/GPS navigation loop for autonomous land vehicle applications *IEEE Trans. Robot. Autom.* **15** 572–8
[8] Fang L, Montestruque L A, McMickell M B, Lemmon M, Sun Y, Koutroulis I, Haenggi M, Xie M and Xie X 2005 Design of a wireless assisted pedestrian dead reckoning system—the NavMote experience *IEEE Trans. Instrum. Meas.* **54** 2342–58
[9] Fischer C and Gellersen H 2010 Location and navigation support for emergency responders: a survey *IEEE Pervasive Comput.* **9** 38–47
[10] Ojeda L and Borenstein J 2007 Non-GPS navigation for security personnel and first responders *J. Navig.* **60** 391–407
[11] Foxlin E 2005 Pedestrian tracking with shoe-mounted inertial sensors *IEEE Comput. Graph. Appl.* **25** 38–46
[12] Godha S and Lachapelle G 2008 Foot mounted inertial system for pedestrian navigation *Meas. Sci. Technol.* **19** 075202
[13] Sabatini A M 2009 Dead-reckoning method for personal navigation systems using Kalman filtering techniques to augment inertial/magnetic sensing *Kalman Filter: Recent Advances and Applications* (Vienna, Austria: I-Tech) pp 251–68
[14] Sabatini A M 2005 Quaternion-based strap-down integration method for applications of inertial sensing to gait analysis *Med. Biol. Eng. Comput.* **43** 94–101
[15] Yun X, Bachmann E R, Moore H and Calusdian J 2007 Self-contained position tracking of human movement using small inertial/magnetic sensor modules *Proc. IEEE Int. Conf. on Robotics and Automation (Rome, Italy, 10–14 April)* pp 2526–33
[16] Sabatini A M 2011 Estimating three-dimensional orientation of human body parts by inertial/magnetic sensing *Sensors* **11** 1489–525

[17] Borenstein J, Ojeda L and Kwanmuang S 2009 Heuristic reduction of gyro drift for personnel tracking systems *J. Navig.* **62** 41–58
[18] Borenstein J and Ojeda L 2010 Heuristic drift elimination for personnel tracking systems *J. Navig.* **63** 41–58
[19] Sabatini A M, Martelloni C, Scapellato S and Cavallo F 2005 Assessment of walking features from foot inertial sensing *IEEE Trans. Biomed. Eng.* **52** 486–94
[20] Skog I, Händel P, Nilsson J and Rantakokko J 2010 Zero-velocity detection—an algorithm evaluation *IEEE Trans. Biomed. Eng.* **57** 2657–66
[21] Jimenez A R, Seco F, Prieto C and Guevara J 2009 A comparison of pedestrian dead-reckoning algorithms using a low-cost MEMS IMU *IEEE Int. Symp. on Intelligent Signal Processing (Budapest, Hungary, 26–28 August)* pp 37–42
[22] Bebek O, Suster M A, Rajgopal S, Fu M J, Huang X, Çavuşoğlu M C, Young D J, Mehregany M, van den Bogert A J and Mastrangelo C H 2010 Personal navigation via high-resolution gait-corrected inertial measurement units *IEEE Trans. Instrum. Meas.* **59** 3018–27
[23] Sun Z, Mao X, Tian W and Zhang X 2009 Activity classification and dead reckoning for pedestrian navigation with wearable sensors *Meas. Sci. Technol.* **20** 015203
[24] Gusenbauer D, Isert C and Krösche J 2010 Self-contained indoor positioning on off-the-shelf mobile devices *Int. Conf. on Indoor Positioning and Indoor Navigation (IPIN) (Zurich, Switzerland, 15–17 September)*
[25] Sabatini A M 2006 Quaternion-based extended Kalman filter for determining orientation by inertial and magnetic sensing *IEEE Trans. Biomed. Eng.* **53** 1346–56
[26] Retscher G 2007 Test and integration of location sensors for a multi-sensor personal navigator *J. Navig.* **60** 107–17
[27] Jirawimut R, Ptasinski P, Cecelja F and Balanchandran W 2003 A method for dead reckoning parameter correction in pedestrian navigation system *IEEE Trans. Instrum. Meas.* **52** 209–15
[28] Aggarwal P, Thomas D, Ojeda L and Borenstein J 2011 Map matching and heuristic elimination of gyro drift for personal navigation systems in GPS-denied conditions *Meas. Sci. Technol.* **22** 025205
[29] Altun K 2011 Intelligent sensing for robot mapping and simultaneous human localization and activity recognition *PhD Thesis* Department of Electrical and Electronics Engineering, Bilkent University, Ankara, Turkey
[30] Altun K, Barshan B and Tunçel O 2010 Comparative study on classifying human activities with miniature inertial and magnetic sensors *Pattern Recognit.* **43** 3605–20
[31] Schwarz L A, Mateus D and Navab N 2012 Recognizing multiple human activities and tracking full-body pose in unconstrained environments *Pattern Recognit.* **45** 11–23
[32] Xsens Technologies 2009 *MTi and MTx User Manual and Technical Documentation* (Enschede, Holland: Xsens Technologies BV) http://www.xsens.com
[33] Kuipers J B 1999 *Quaternions and Rotation Sequences* (Princeton, NJ: Princeton University Press)
[34] Mathie M J, Coster A C F, Lovell N H and Celler B G 2004 Accelerometry: providing an integrated, practical method for long-term, ambulatory monitoring of human movement *Physiol. Meas.* **25** R1–20
[35] Sabatini A M 2006 Inertial sensing in biomechanics: a survey of computational techniques bridging motion analysis and personal navigation, *Computational Intelligence for Movement Sciences: Neural Networks and Other Emerging Techniques* (Hershey, PA: Idea Group Publishing) pp 70–100
[36] Preece S J, Goulermas J Y, Kenney L P J, Howard D, Meijer K and Crompton R 2009 Activity identification using body-mounted sensors—a review of classification techniques *Physiol. Meas.* **30** R1–33
[37] Barshan B 2008 Objective error criterion for evaluation of mapping accuracy based on sensor time-of-flight measurements *Sensors* **8** 8248–61
[38] Altun K and Barshan B 2010 Human activity recognition using inertial/magnetic sensor units *Human Behavior Understanding (Lecture Notes on Computer Science* vol 6219*)* ed A A Salah (Berlin: Springer) pp 38–51
[39] Susi M, Renaudin V and Lachapelle G 2011 Detection of quasi-static instants from handheld MEMS devices *Int. Conf. on Indoor Positioning and Indoor Navigation (IPIN) (Guimarães, Portugal, 21–23 September)*
[40] Odell P L, Dorsett D, Young D and Igwe J 1989 Estimator models for combining vector estimators *Math. Comput. Modelling* **12** 1627–42