A 3D indoor positioning system based on low-cost MEMS sensors

Lingxiang Zheng [a],*, Wencheng Zhou [a], Weiwei Tang [a], Xianchao Zheng [a],
Ao Peng [a], Huiru Zheng [b]

[a] School of Information Science and Engineering, Xiamen University, Xiamen, China
[b] School of Computing and Mathematics, Computer Science Research Institute, University of Ulster, Jordanstown Campus, Shore Road, Newtownabbey, County Antrim, UK

ARTICLE INFO

ABSTRACT

A positioning system in the absence of GPS is important in establishing indoor directional guidance and localization. Inertial Measuring Units (IMUs) can be used to detect the movement of a pedestrian. In this paper, we present a three-dimensional (3D) indoor positioning system using foot mounted low cost Micro-Electro-Mechanical System (MEMS) sensors to locate the position and attitude of a person in 3D view, and to plot the path travelled by the person. The sensors include accelerometers, gyroscopes, and a barometer. The pedestrians motion information is collected by accelerometers and gyroscopes to achieve Pedestrian Dead-Reckoning (PDR) which is used to estimate the pedestrian's rough position. A zero velocity update (ZUPT) algorithm is developed to detect the standing still moment. A Kalman filter is combined with the ZUPT to eliminate non-linear errors in order to obtain accurate positioning information of a pedestrian. The information collected by the barometer is integrated with the accelerometer data to detect the altitude changes and to obtain accurate height information. The main contribution of this research is that the approach proposed fuses barometer and accelerometer in Kalman filter to obtain accurate height information, which has improved the accuracy at x axis and y axis. The proposed system has been tested in several simulated scenarios and real environments. The distance errors are around 1%, and the positioning errors are less than 1% of the total travelled distance. Results indicate that the proposed system performs better than other similar systems using the same low-cost IMUs.

© 2016 Elsevier B.V. All rights reserved.

1. Introduction

The Global Positioning System is widely used in tracking locations at outdoors. However, it cannot be used in indoor environments because of its experience of sever signal attenuation. On the other hand, there are huge demands for high precision indoor positioning systems. Examples of the applications include tracking children's locations in shopping malls or large supermarkets, locating people with elderly dementia in providing support, and finding a car in a big parking garage.

Current indoor positioning technologies can be divided into two types: (1) infrastructure-based approaches and (2) infrastructure-free approaches.

* Corresponding author. Tel.: +86 13515967334.
 E-mail addresses: lxzheng@xmu.edu.cn (L. Zheng), h.zheng@ulster.ac.uk (H. Zheng).

Infrastructure-based approaches obtain the indoor positioning based on the information collected from external infrastructure or external equipment, such as network nodes, WiFi signals, Bluetooth signals, radio frequency (RF) signals, magnetic signals and video signals.

Infrastructure-free approaches eliminate the need for external signals. Most of these approaches are based on inertial sensors, *i.e.*, accelerometers and gyroscopes. These sensors can accurately collect data in a harsh environment. However, the drift and bias errors of these sensors cause serious problems. Recently, the zero velocity update (ZUPT) algorithm [1] was proposed to overcome the sensors' errors and improve their accuracy significantly.

Infrastructure-based solutions require the installation of infrastructure and tends to be more expensive [2], by contrast, infrastructure-free solutions are more flexible and low cost [3]. Recent years have seen the trend of moving toward to infrastructure-free solutions, however, the accuracy is too low to be used in real world applications [4]. To address this challenge, this paper proposes an infrastructure-free approach to achieve high precision 3D indoor positioning using a low cost sensor. Pedestrian dead-reckoning (PDR) is used in this paper to sequentially estimate the pedestrian's position. PDR consists of the double integration of current inertial sensor readings. When using PDR to calculate the position and velocity, the system will eventually be led to divergent because of the sensor error [5]. To solve this issue, we use Kalman filter model as basis model. It consists a prediction process and an update process. When there is an appropriate observation, Kalman filter can compensate the error by the update equation to keep the system relatively stable. ZUPT can very accurately detect the time when a pedestrian is standing still, *i.e.* the velocity of the pedestrian is zero. We use this information as the observation in Kalman filter, so it can eliminate system errors greatly. Comparing to the existing solutions, we use the barometer fused with accelerometer to obtain z-axis information by combine integrating ZUPT with Kalman filter. The existing solutions for height information achievement include the fusion of barometer and accelerometer, but as far as we know, none of them added ZUPT in the model. Because our system is foot-mounted, so ZUPT is available and we use it as our observation to eliminate error to get more accurate results. And the accurate height information will improve the results at *x*-axis and *y*-axis in Kalman filter, which will outperform than some other existing solutions.

The rest of this paper is organized as follows: Section 2 describes the related work. Section 3 presents the proposed mathematical model of the system, with the details of the design, framework and the processing methods. The simulation experiments to test the model and the results are detailed in Section 4. The paper is concluded in Section 5 with a discussion of limitations and future work.

2. Related work

There is a large amount of work in the area of indoor positioning with infrastructure-based approaches. A typical infrastructure-based approach is described in [6,7], which located a pedestrian using WiFi signal strength measurements. Bacak and Celebi [2] proposed an indoor positioning system based on the RF signal fingerprinting method. Aversa [8] used the heterogeneous wireless networks in localization. The IndoorAtlas and the DLR proposed indoor positioning solutions based on magnetic field information [9,10]. The advancement of Internet of Things presents another opportunity for different kinds of sensors and devices to obtain a variety of services to help them in positioning and in other functionalities [11,12]. The infrastructure-based system suffers from the signal interference and the installation and maintenance of a full set of signal transmitting and receiving devices can increase the system cost. It limits their usability, especially when there are no external sources of signal or when these signals are difficult to setup.

There are also a number of authors proposed kinds of infrastructure-free approaches to achieve indoor positioning. Krach [13], detected the state of motion of the pedestrian by measuring acceleration and angular velocity collected by an accelerometer and a gyroscope. In [14], authors estimated the velocity, position and attitude information of the pedestrian to create a pedestrian movement model by inertial calculation. In [15], the author achieved positioning by utilizing multiple cameras. There are other publications that combined the position information calculated by inertial sensors with other information sources, such as WiFi [16], magnetic field [17], and indoor plan [18,19], to obtain a higher accuracy. The infrastructure-free approaches remove the dependence from the signal source and reduce their maintenance cost significantly. Nevertheless, these methods still use expensive sensors to obtain high precision achievements and the majority are based on the two-dimensional plane [20–22].

In our previous work, Zheng et al. [23] developed a 3D indoor positioning system based on the Kalman filter model using low cost Micro-Electro-Mechanical System(MEMS) sensors. MEMS is the technology of very small devices, and merges at the nano-scale into nanoelectromechanical systems (NEMS) and nanotechnology [24]. The inertial sensors computed the position using the Pedestrian Dead-Reckoning (PDR) algorithm. The position, velocity, and altitude information of the pedestrian was then calculated. The acceleration and angular velocity were used to detect the time when the pedestrian was in contact with the ground while walking through the zero velocity update (ZUPT) algorithm. The velocity equals zero at the time when the ZUPT is satisfied. This was input into a Kalman filter as an observation value to eliminate systematic errors. A barometer was introduced to obtain height information to improve 3D indoor positioning system.

In this paper, we further improve the precise of the work. The barometer is integrated with the accelerometer and input to the Kalman filter for height information. We also use the height information calculated to distinguish whether the pedestrian is walking on flat ground, climbing stairs or taking an elevator. The Kalman filter is extended to 15-dimensions, and the acceleration deviation and gyroscope bias is added in the state vector. Combining these methods are beneficial for
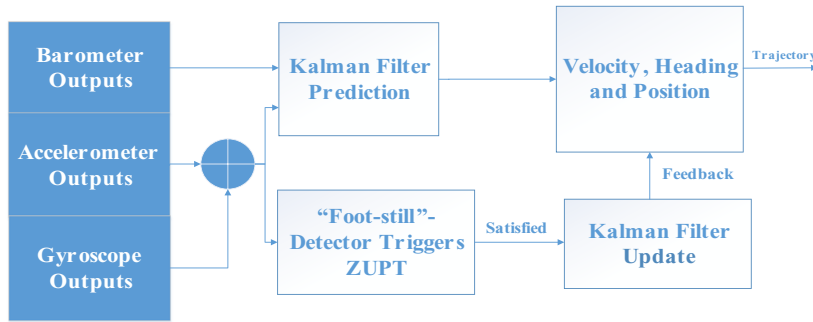
Fig. 1. Algorithm architecture.

Fig. 2. System architecture.

obtaining higher precision positioning results. The particle filter (PF) was removed from previous work to reduce calculation costs.

3. Mathematical model

The overall design and the algorithm architecture is shown in Fig. 1. The system consists of four modules: data acquisition (accelerometer, gyroscope and barometer), strapdown inertial calculation (Kalman filter prediction), ZUPT algorithm, and error elimination (Kalman filter update).

The data acquisition module gathers data from accelerometers, gyroscopes and barometer. The outputs of the accelerometers and gyroscopes are sent to Kalman filter prediction module to calculate the movement of the foot. The barometer is used to measure the height information. The acceleration and angular velocity information is also used for foot-still detection, which detect the time when the velocity of pedestrian is zero. When ZUPT is satisfied, the module of Kalman filter update will complete the correction of velocity. The corrected velocity is then fed back to the trajectory of velocity, heading and position information and output.

Fig. 2 illustrates the system architecture. Walking data are collected by a foot-mounted sensor and is sent to the smart phone via blue tooth. And all calculations are carried out on the smart phone. Then the users can view the trajectory on their phone, and send the trajectory data to cloud to share their experience.

3.1. Zero velocity update algorithm

The ZUPT (zero velocity update) algorithm has been proved to be an effective method to control and eliminate data drift errors. ZUPT is triggered when the foot is stationary on the ground. The ZUPT algorithm applied in paper [1] considered multiple conditions using the information output from accelerometers and gyroscopes. Three conditions ($C1$, $C2$ and $C3$) were used to determine whether the foot is stationary on the floor. The conditions are defined as

$$C1 = \begin{cases} 1 & th_{a_{min}} < a_{k\text{-}total} < th_{a_{max}} \\ 0 & otherwise \end{cases} \tag{1}$$

$$C2 = \begin{cases} 1 & \sigma^2_{a_k} > th^2_{\sigma_{a\text{-}min}} \\ 0 & otherwise \end{cases} \tag{2}$$

$$C3 = \begin{cases} 1 & |w_{k\text{-}total}| < th_{w_{max}} \\ 0 & otherwise \end{cases} \tag{3}$$

where $a_{k\text{-}total}$ represents the magnitude of the acceleration at time k, $\sigma^2_{a_k}$ is the local acceleration variance, and $w_{k\text{-}total}$ represents the magnitude of the heading. The parameter $th_{a_{min}}$, $th_{a_{max}}$, $th^2_{\sigma_{a\text{-}min}}$ and $th_{w_{max}}$ are defined based on the observation to make the ZUPT detector be active only when the pedestrian is stand-still. Condition $C1$ indicates that the acceleration is within a limited range, which is close to the gravity value. Condition $C2$ means the fluctuation of acceleration cannot be too large, while $C3$ means the magnitude of angular velocity is smaller than a threshold. In theory, the angular velocity is zero when pedestrian stands, ZUPT would only be triggered when all of the three logical conditions are satisfied simultaneously.

3.2. Height information calculation model

Titterton and Weston [5] described the method to obtain the velocity and distance information of pedestrian, solely based on accelerometer and gyroscope outputs. In addition to these two sensors, we introduced a barometer module to detect floor level changes. Multiple sensors are mounted on the shoes of the pedestrian, and the values are read when the pedestrian is stationary on the floor. The mean value of the pressure is computed as Eq. (4):

$$\overline{P(k)} = \frac{\sum_{i=stepLocs_k(1)}^{stepLocs_k(n)} P(i)}{n} \tag{4}$$

where $P(i)$ represents the value of mean pressure detected by sensors, and $stepLocs_k(n)$ is the nth point on the Kth step.

The following formula calculates the pressure difference:

$$P(k) = \overline{P(k)} - \overline{P(0)} \tag{5}$$

where $P(0)$ represent the pressure value of the first step, and here we use the value $P(k)$ to calculate the relative altitude to the first position.

Maik and Olive [25] proposed a relation between the barometer measurements and the relative altitude as shown in Eq. (6):

$$p_h(k) = p_0(k) \cdot \left(1 - \frac{a \cdot \Delta h(k)}{T_0(k)}\right)^{\frac{M \cdot g(\varphi(k))}{R \cdot a}} \tag{6}$$

where $\Delta h(k) = h(k) - h_0$, $h(k)$ denotes the height at time instance k, h_0 denote reference height, and where M is the molar mass, $g(\varphi)$ is the gravity, depending on the current latitude φ of the user and R is the universal gas constant, respectively.

Here we use the constant values $p_0(0)$, $T_0(0)$ and $\varphi(0)$ to replace $p_0(k)$, $T_0(k)$ and $\varphi(k)$, respectively and apply a Taylor series expansion around the triple $(h(0), T_0(0), p_0(0))$, as shown in Eq. (7).

$$p_h(k) = p_0(0)\left(1 - \frac{a \cdot \Delta h(0)}{T_0(0)}\right)^{\frac{M \cdot g(\varphi(0))}{R \cdot a}} - p_0(0)\frac{M \cdot g(\varphi(0))}{T_0(0) \cdot R}\left(1 - \frac{a \cdot \Delta h(0)}{T_0(0)}\right)^{\frac{M \cdot g(\varphi(0))}{R \cdot a} - 1} \cdot \Delta h(k) \tag{7}$$

We use $p_h(k)$ minus $p_h(0)$, and obtain an approximate relation between the barometer measurements and the relative altitude $\Delta p_h(k)$:

$$\Delta p_h(k) = -p_0(0)\frac{M \cdot g(\varphi(0))}{T_0(0) \cdot R}\left(1 - \frac{a \cdot \Delta h(0)}{T_0(0)}\right)^{\frac{M \cdot g(\varphi(0))}{R \cdot a} - 1} \cdot \Delta h(k) \tag{8}$$

where $\Delta p_h(k) = p_h(k) - p_h(0)$,

Define N as

$$N = -p_0(0)\frac{M \cdot g(\varphi(0))}{T_0(0) \cdot R}\left(1 - \frac{a \cdot \Delta h(0)}{T_0(0)}\right)^{\frac{M \cdot g(\varphi(0))}{R \cdot a} - 1} \tag{9}$$

N is a constant and Eq. (8) can be written as

$$\Delta p_h(k) = N\Delta h \tag{10}$$

The relative altitude information $\Delta p_h(k)$ can then be obtained from barometer measurements using Eq. (10).

The relative height information can also be calculated by accelerometer as Eq. (11)

$$h = \frac{1}{2}acc \cdot t^2 \tag{11}$$

where acc is the vertical acceleration.

Kalman filter is then used to integrate the relative altitude information and height information, obtained using Eq. (10) with Eq. (11) respectively. The process is detailed as follows:

The state equation can be modeled as

$$X_{h,k} = F_h X_{h,k-1} + B_h u_k + w_k \tag{12}$$

where

$X_{h,k} = \begin{bmatrix} h_k & v_k \end{bmatrix}^T$, h_k is the relative height at time point k and v_k represents the vertical speed at time point k;

$F_h = \begin{bmatrix} 1 & Ts \\ 0 & 1 \end{bmatrix}$, Ts represent the sampling interval;

$B_h = \begin{bmatrix} \frac{Ts^2}{2} & Ts \end{bmatrix}^T$;

$u_k = acc_k$; and

$w_k \sim N(0, Q_k)$, w_k is the process noise which is assumed to be drawn from a zero mean multivariate no normal distribution with covariance Q_k

The observation equation is defined in Eq. (13).

$$Z_{h,k} = H_h X_{h,k} + v_k \tag{13}$$

where

$$H_k = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}; \text{ and}$$

$v_k \sim N(0, R_k)$, v_k is the observation noise which is assumed to be zero mean Gaussian white noise with covariance R_k.

The update equations for the Kalman filter are defined as the time update and the measurement update.

Time update is computed in Eqs.(14) and (15):

$$\widehat{x}_{k|k-1} = F_h \widehat{x}_{k-1|k-1} + B_h u_{k-1} \tag{14}$$

$$P_{k|k-1} = F_h P_{k|k-1} F_h^T + Q_k \tag{15}$$

Measurement update starts when ZUPT is true, i.e. the vertical speed is equal to zero at the time. The measurement update is presented in Eqs. (16)–(18).

$$K_k = P_{k|k-1} H_k^T \left(H_k P_{k|k-1} H_k^T + R_k \right)^{-1} \tag{16}$$

$$P_{k|k} = (I - K_k H_k) P_{k|k-1} \tag{17}$$

$$\widehat{x}_{k|k} = \widehat{x}_{k|k-1} + K_k \left(\begin{bmatrix} h_{bar} & 0 \end{bmatrix} - H_k \widehat{x}_{k|k-1} \right) \tag{18}$$

where h_{bar} is derive from Eq. (10)

The height information calculated from Kalman filter shown above can be used to distinguish the pedestrian is going stairs or not. Two threshold γ_h and μ_h are introduced. If difference of two consecutive height Δh is smaller than γ_h, It indicates that the pedestrian is walking on a flat route. Otherwise, the pedestrians is climbing stairs or taking elevator. To distinguish the later, the statistics variance is used. The specific method is that if the variance of pressure calculated from previous step to next step is larger than μ_h, the pedestrian is considered taking elevator. The algorithm is showed below:

1: **if** $|h_{next} - h_{pre}| < \gamma_h$ **then**
2:   Pedestrian is walking on a flat route
3: **else if** $var\left(h_{pre} : h_{next} \right) > \mu_h$ **then**
4:   Pedestrian is taking elevator
5: **else**
6:   Pedestrian is climbing stairs
7: **end if**

3.3. Pedestrian movement model

In this paper, we use Kalman filter to build a pedestrian movement model. Navigation parameters used in the study are pedestrian position, velocity, and attitude information. The detailed description of the predicted state $\widehat{x}_k^-$, covariance P_k, state transition matrix F and observation matrix H can be found in our previous work [26]. Here we introduce the IKZ (Inertial Navigation System & Kalman Filter & ZUPT) model designed in this paper.

Firstly, The Inertial Navigation System calculated the position s_k^T, velocity v_k^T, and altitude ϕ_k^T of pedestrian. The state $\widehat{X}_k^-$ predicted by KF is

$$\widehat{X}_k^- = \begin{bmatrix} s_k^T & v_k^T & \phi_k^T \end{bmatrix} \tag{19}$$

Secondly, the ZUPT is true, when the pedestrian was standing still, which means that the velocity of the pedestrian is zero. This information is used as a measurement value of Kalman filter, and the posterior estimation $\widehat{X}_{k|k}$ is calculated in Eq. (20):

$$\widehat{X}_{k|k} = \widehat{X}_{k|k-1} + K_k \left(O_{3*1} - H_k \widehat{x}_{k|k-1} \right) \tag{20}$$

where O_{3*1} is a 3∗1 zero matrix, i.e. each element of the matrix is 0.

Through the design above, the drift error can be minimized effectively. As for the bias error in the MEMS sensor, the existing approaches assume that the bias error is a constant of the system under static conditions, and thus a one-time compensation is made in the system application. In reality, the bias error is not a constant after a long time of continuously working or with influence of the temperature change [27]. To address this issue, we use Kalman filtering to estimate necessary compensation. Firstly, we expand the Kalman filter state matrix X_k from 9 dimensions to 15 dimensions, where the

new entrants are accelerometer bias error ε_b for 3 dimensions and gyroscope bias error α_b with another for 3 dimensions as described in Eq. (21)

$$X_k = \begin{bmatrix} s_k^T & v_k^T & \phi_k^T & \varepsilon_b^T & \alpha_b^T \end{bmatrix}^T \tag{21}$$

Secondly, we expand the one step prediction covariance matrix P, the process noise covariance matrix Q and the observation matrix H as shown in Eq. (22):

$$\begin{cases} P = \begin{bmatrix} P_{11} & O_{3*3} & O_{3*3} & O_{3*3} & O_{3*3} \\ O_{3*3} & P_{11} & O_{3*3} & O_{3*3} & O_{3*3} \\ O_{3*3} & O_{3*3} & P_{11} & O_{3*3} & O_{3*3} \\ O_{3*3} & O_{3*3} & O_{3*3} & P_{11} & O_{3*3} \\ O_{3*3} & O_{3*3} & O_{3*3} & O_{3*3} & P_{11} \end{bmatrix} \\ Q = \begin{bmatrix} Q_{11} & O_{3*3} & O_{3*3} & O_{3*3} \\ O_{3*3} & Q_{22} & O_{3*3} & O_{3*3} \\ O_{3*3} & O_{3*3} & Q_{33} & O_{3*3} \\ O_{3*3} & O_{3*3} & O_{3*3} & Q_{44} \end{bmatrix} \\ H = \begin{bmatrix} O_{4*2} & I_{4*4} & O_{3*3} & O_{3*3} \end{bmatrix} \end{cases} \tag{22}$$

Here, O_{i*j} represents an $i*j$ zero matrix. I_{i*j} is an $i*j$ scalar matrix, with all diagonal elements of the matrix equals to 1 and others are 0. P_{11}, P_{22} and P_{33} are the experience value of the initial covariance matrix of the pedestrian navigation parameters [28]. O_{11} and O_{22} represent the experience values of the initial covariance matrix of the accelerometer and gyroscope process noise [28]. P_{44} and p_{55} represent the initial mean square errors (MEMs) of the accelerometer bias drift and the gyroscope bias drift respectively, while Q_{33} and Q_{44} represent the process noise covariance matrix of the accelerometer bias modeling and the gyroscope bias modeling respectively.

Finally, the state transition matrix F and the noise gain matrix Γ in the updating phase are expanded as described in Eq. (23):

$$\begin{cases} F = \begin{bmatrix} & & & O_{3*3} & O_{3*3} \\ & F_{9D} & & Rb2t*Ts & O_{3*3} \\ & & & O_{3*3} & -Rb2t*Ts \\ O_{3*3} & O_{3*3} & O_{3*3} & I_{3*3} & O_{3*3} \\ O_{3*3} & O_{3*3} & O_{3*3} & O_{3*3} & O_{3*3} \end{bmatrix} \\ \Gamma = \begin{bmatrix} O_{3*3} & O_{3*3} & O_{3*3} & O_{3*3} \\ Rb2t*Ts & O_{3*3} & O_{3*3} & O_{3*3} \\ O_{3*3} & -Rb2t*Ts & O_{3*3} & O_{3*3} \\ O_{3*3} & O_{3*3} & I_{3*3}*Ts & O_{3*3} \\ O_{3*3} & O_{3*3} & O_{3*3} & I_{3*3}*Ts \end{bmatrix} \end{cases} \tag{23}$$

Through this process, the accelerometer bias error ε_b and the gyroscope bias error α_b in 15D Kalman filter can be obtained, and they are used to eliminate the bias error in the MEMS sensor.

4. Simulation experiments and results

In this section, we present the data collection from real walkings and the simulation experiments using MATLAB. APM 2.5 [29] is used in the experiments to evaluate the proposed algorithm, which is implemented with a small low cost six degree of freedom (6DOF) inertial sensor (MPU6000) and a pressure sensor (MS5611). The pedestrian participated in the experiment is a 22 years old healthy male, and he walked through a building wearing the shoe-mounted sensor module. The data were recorded at the 100 Hz clock rate. All the experiments were conducted at No. 2 Research Building at Haiyun Park in Xiamen University, China. The experimental data were recorded in text format and can be downloaded from GitHub (https://github.com/ECG-XMU/ShoeMountIMU-Dataset).

4.1. Simulation of the system components

4.1.1. Simulation experiment of the zero-velocity detector
Four simulation experiments were conducted to detect the number of steps. The pedestrian in the experiment had walked around randomly in a regular way.
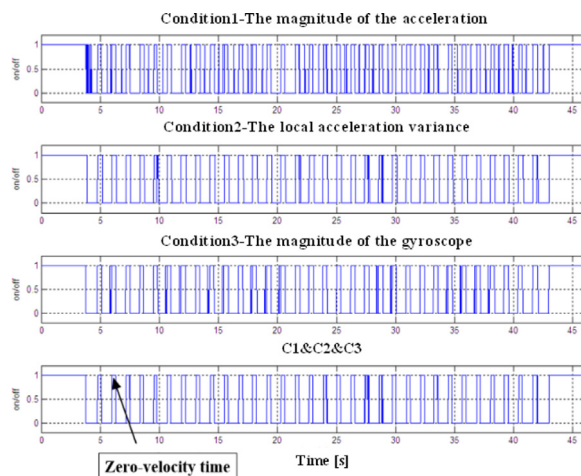
Fig. 3. The condition of the zero-velocity.

Fig. 4. Schematic diagram of pedestrian horizontal position with 9D Kalman filter state model.

Table 1
The detection of steps, the elements represent detected steps under the left conditions.

Steps	Ex1	Ex2	Ex3	Ex4
C1	45	118	149	222
C2	42	75	110	141
C3	32	67	94	133
C1&C2&C3	34	70	102	135
True steps	34	69	100	136

The four simulation experiments were conducted, they detect step number by four ZUPT conditions: $C1$, $C2$, $C3$ and combined conditions $C1\&C2\&C3$ (described in ZUPT algorithm), respectively. Fig. 3 shows the data of $C1-C3$ and zero velocity time ($C1\&C2\&C3$) from accelerometer and gyroscope when the pedestrian walking in the corridor.

Table 1 lists the simulation results. The actual steps walked by the pedestrian were counted manually. The steps were measured using the three sensors C from individual sensors and all the three sensors together. As can be seen from the data in the table that any one single condition cannot detect the number of steps precisely, whereas the steps detected with the combined conditions $C1\&C2\&C3$ are much more accurate.

4.1.2. Comparison of Kalman filter models

Two experiments were carried out to verify the performance of the 9D Kalman filters and 15D Kalman filters. The pedestrian participant walked from the left side of the building to the right side of the building, then returned to the original location, and repeated the walk for four times. Schematic diagrams of pedestrian horizontal position for the 9D and 15D Kalman filters are shown in Figs. 4 and 5, respectively.

Fig. 4 shows that the track of pedestrian was consistent with the practical path. However there was still an angular drift error in whole track. After three exhumations, final positioning had a vertical shifting about 2 m. In Fig. 5, vertical drift was improved notably, and the track of pedestrian coincides with the real path more closely.

Specific parameters used in the experiments are listed and compared in Table 2.

Fig. 5. Schematic diagram of pedestrian horizontal position with 15D Kalman filter state model.

Table 2
Parameter comparison between 9D KF and 15D KF.

	9D state module	15D state module
Heading angle drift	0.02	$\approx 0 (< 0.01)$
Maximum speed	3.23	3.24
Speed drift	$\approx 0 (< 0.01)$	$\approx 0 (< 0.01)$
Total walking distance	247.93	251.95
Actual coordinates of the end	(0,0)	(0,0)
Positioning coordinates of the end	(0.21,−2.03)	(−0.49, −0.06)
Error of the end	2.04	0.49

Table 3
Time cost and error reduce comparison between system with PF and system without PF.

Time cost/error	Track 1	Track 2	Track 3
System without PF	3.42 s/0.66%	13.34 s/0.38%	3.06 s/0.34%
System with PF	6.29 s/0.53%	23.34 s/0.30%	6.25 s/0.29%
PF time increase	83.92%	74.96%	104.24%
PF error reduce	0.13%	0.08%	0.05%

The comparisons in the table shows that the whole system has been greatly improved by using the 15D Kalman filter state module. It has successfully eliminated zero bias error of the angular velocity and achieved the elimination of heading angle drift. The original 2 m vertical drift error has been eliminated. It greatly enhanced positioning accuracy in the 2D plane.

4.1.3. Assessment of the particle filter

In this paper, the particle filter (PF) was removed from previous work, because of its high computing cost. Here we carried out three experiments to compare the performance of system with or without using the PF. We assessed the increase of time cost and the decrease of system error when using the PF. As shown in Table 3. the time increase was 87.71% in average, while the error was only reduced by 0.09%. Therefore, the removing the PF will reduce the time cost significantly with marginal loss of the accuracy.

The Track 3 is shown in Fig. 6. A subject walked from the entrance of 2th floor and walked to the stairs, then he downstairs to 1st floor and go back to 2nd floor at start point. The whole distance calculated 120.55 m. It can be found that both of trajectories are match to true path. And in the system without PF, the distance between start point and end point is 0.41 m, the error rate is 0.34%. At the same time, in the system with PF, the distance between start point and end point is 0.35 m, the error rate is 0.29%. It can be conclude that the PF only improved a little performance in the above algorithm.

4.1.4. Simulation of height information calculation

To test the algorithm on the height information calculated, which was presented in Section 3. We carried out an experiment which included three ways of walking, including walking on a flat ground section, taking elevator and climbing stairs, respectively. And in this simulation experiment, the constant γ_h was set as 0.16, and μ_h was set as 300 Pa.

The participant walked on the flat ground from the entrance of 3rd floor of No. 2 Research Building to the front of the elevator and took the elevator to 1st floor, then went through corridor to the upstairs, finally he walked up to the 3rd floor. To compare the performance of the proposed algorithm, we tested the data using the previous algorithm which is used to obtain the height information only using the barometer. The paths of height information compared are shown in Fig. 7.

Both of the height distance between start point and end point in two trajectory are close to zero. With the presented algorithm, the distance is 0.04 m, and with the previous algorithm the distance is 0.18 m. And we can found that the two
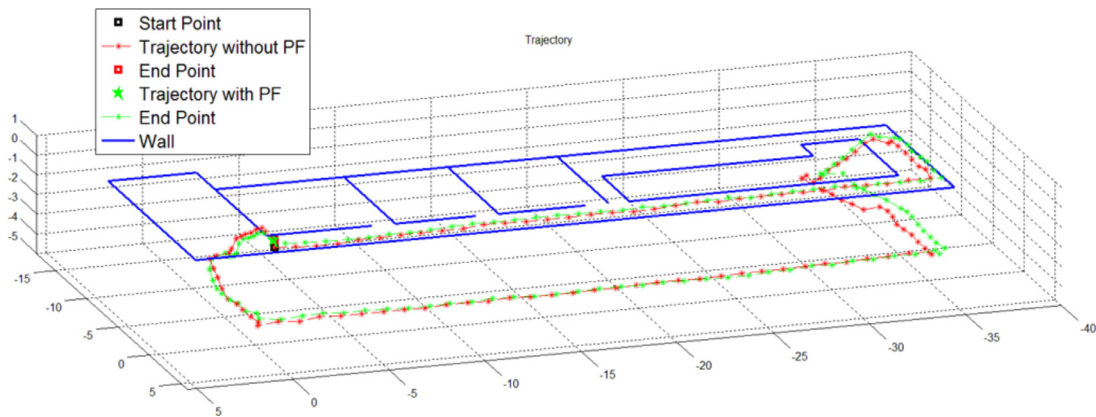
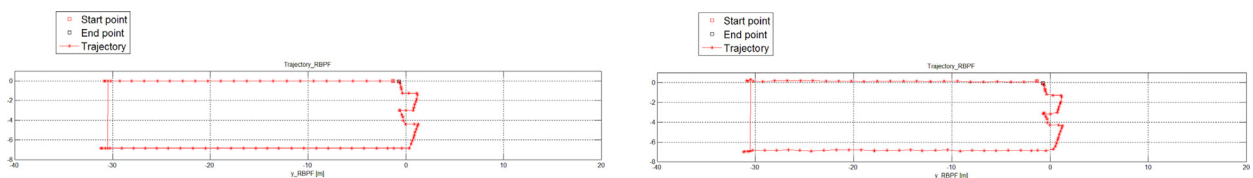**Fig. 6.** The trajectory of the system with PF and without PF.

Fig. 7. Experiment of height information (left for presented algorithm and right for previous algorithm).

Table 4
The distance measured in given tracks.

Subjects	Trajectory				
	Straight	Rectangle	Triangle	Circular	Mean error
Subject A	0.7091%	1.0567%	0.3146%	0.3267%	0.6018%
Subject B	0.5667%	1.0467%	0.8433%	0.5000%	0.7392%
Subject C	1.4233%	2.1667%	0.3867%	1.0773%	1.2635%
Subject D	1.1900%	1.7000%	0.9633%	0.8633%	1.1791%
Subject E	1.1267%	1.1367%	0.7233%	0.3304%	0.8293%
Mean error	1.0032%	1.4214%	0.6462%	0.6195%	0.9226%

algorithm can both recognize the motion of take elevator. However, trajectory with the present algorithm is very stable when the pedestrian walked in flat, and there are some jitter in the latter trajectory.

### 4.2. Simulation of the system

4.2.1. Simple trajectory

In this section, we employed five subjects to test the performance of the system. The five subjects includes 22–24 years old male and female participants. The test locations are the No. 2 Research Building and the Experimental Building at Xiamen University. The test consists of four different trajectories. These tracks are straight, triangle, rectangular, and circular. The real length of the track of straight is 28.48 m. The rectangular track is a square with the side length of 5.6 m. The triangle track is an isosceles triangle with the bottom side length of 12 m and the height of 6 m. The circular track is a circle with a radius of 3 m. Labels are placed on the ground to make sure the participant walked on the track. Each participant walked three times for each track.

We calculated the distance through the system algorithm and assessed the error using Eq.(24).

$$Error\ rate = \frac{\left|True\ distance - Distance\ calculated\right|}{True\ distance} \qquad (24)$$

The values in Table 4 present the average error of one participant walking on one track, i.e. the mean value of errors over the participant's three walks of the same track. The real distance of the trajectories of straight, rectangle, triangle, and circular are 28.48 m, 25.60 m, 28.97 m and 18.85 m, respectively. The average errors of all subjects over each track were calculated. The overall error was calculated over all participants and tracks to assess the performance, which is 0.9226%, as shown in the last column and the last row in Table 4. The results demonstrate that the proposed algorithm has performed well in simple trajectories. Most complex trajectories in real life can be decomposed as many small regular trajectories, such
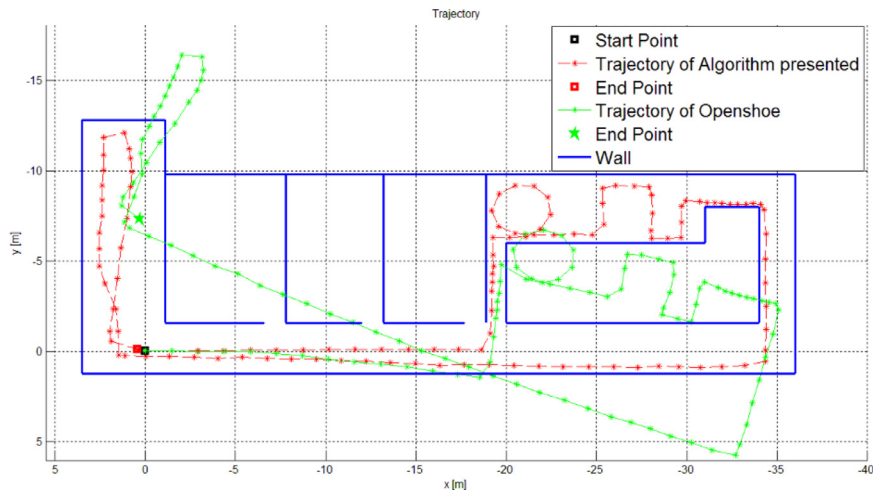
Fig. 8. Experiment of 3D pedestrian navigation compared to Openshoe. (For interpretation of the references to color in this figure, the reader is referred to the web version of this article.)
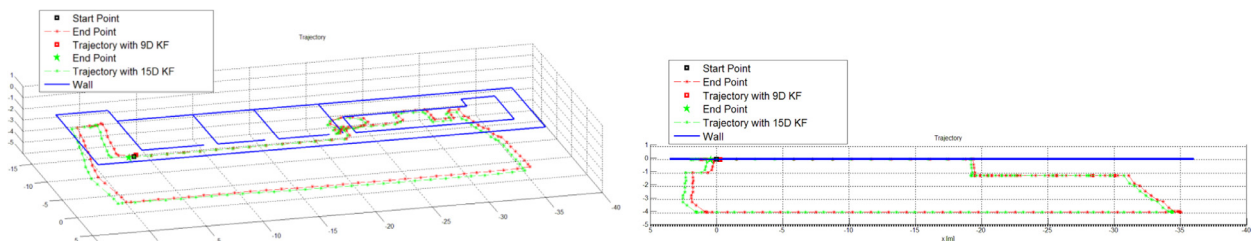
Fig. 9. Experiment of 3D pedestrian navigation (left for overall and right for front view) with presented algorithm.

as rectangle, triangle, and circular. Therefore the inertial navigation system presented in this paper can also be applied to the complex trajectories.

4.2.2. Complex trajectory

In this experiment, Subject C wore the Sensor Module on the right side of the right foot, walking from the entrance of 2nd floor of No. 2 Research Build. Firstly he walked straight to the middle of 2nd floor, then went down stairs to the landing place. After that he paced along a circular track and a rectangular track, then he went down stairs to 1st floor, and went through the corridor and went upstairs, finally he walked up to the start position on the 2nd floor. This complex trajectories was used to simulate walking in a complex building in order to test the overall performance of proposed algorithm in complex environments. In Eq. (9), no knowledge about $p_0(0)$, $T_0(0)$ and other constants were described, so we obtained the constant N via experiments. In this experiment N was set to 0.09.The height information was calculated by the accelerometer and barometer, and fused by Kalman filter which as presented in Section 3.

To assess the performance of the proposed algorithm, we test the data using the algorithm presented by Openshoe [28].

Openshoe does not have height information, so we only compared the top view. Here, we added 15D KF into our system. The paths compared are shown in Fig. 8. The red line presents the results from the proposed algorithm, and the green line is obtained from the Openshoe's algorithm. Results show that the algorithm we proposed in this paper had much less heading error in comparison to the algorithm proposed by Openshoe. The end point of the Openshoe's trajectory is far away from the start point. There is very small heading error in the trajectory of our algorithm, so it matched the real trajectory and the start point is closed to the end point. This indicates that the proposed algorithm performed well in complex trajectories.

We then compared the performance of the proposed algorithm with our previous algorithm in 3D view using both 9D Kalman filter and 15D Kalman filter. The path of proposed algorithm is shown in Fig. 9, and the path of the previous algorithm is shown in Fig. 10. The figures show that both of the two tracks calculated matched the real track closely. The error rate we calculated is the distance between the start point and the end point (DBSE) divided by the total distance. The distance of whole track is 138.65 m and the error rates are summarized in Table 5. There is huge heading error in Openshoe's algorithm and the distance between the start point and the end point is 8.74 m with the error rate of 6.30%. The start point and the end point in the presented algorithm with 15D KF is 0.23 m, the error rate is only 0.17%. This experiment shows that proposed algorithm, in comparison to the other algorithms, has significantly improved in accuracy, and the trajectory of the proposed algorithm is more stable than other algorithms. Here, we speculate that the three-axis coordinate interacts
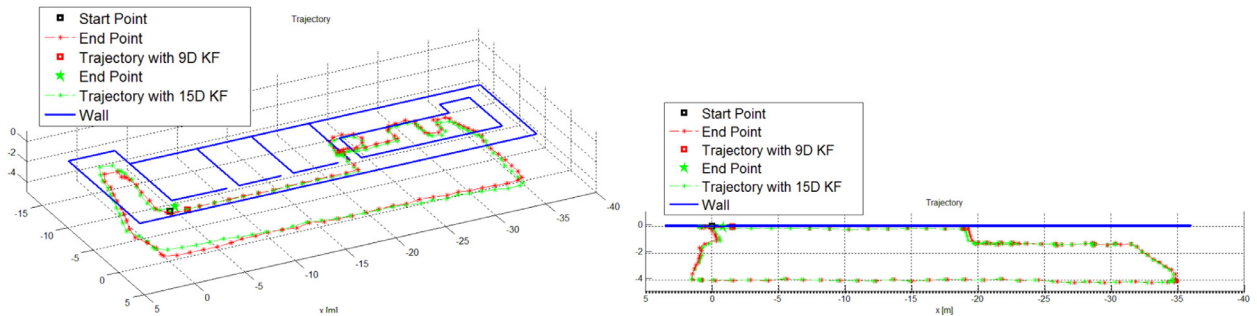
Fig. 10. Experiment of 3D pedestrian navigation (left for overall and right for front view) with previous algorithm.

Table 5
The error rate in different algorithm.

	DBSE	Total distance	Error rate
Previous algorithm with 9D KF	2.60 m	138.65 m	1.88%
Previous algorithm with 15D KF	1.07 m	138.65 m	0.77%
Presented algorithm with 9D KF	0.44 m	138.65 m	0.32%
Presented algorithm with 15D KF	0.23 m	138.65 m	0.17%
Openshoe algorithm	8.74 m	138.65 m	6.30%

in PDR with the coordinate rotation matrix [5], so the accurate height information is helpful for improving the accuracy in the x,y plane. As for 9D KF and 15D KF, we found that from the beginning of the track, there is no significant difference in the trajectory between the two schemes (Figs. 9 and 10); however, in the latter part of the walk, the algorithm with 15D KF started to fix gyroscope and accelerometer bias, mainly gyroscope bias caused by the drift. But no difference in the short time walking between 9D KF and 15D KF.

As discussion above, we can conclude that the accuracy of the proposed algorithm performs better in the complex trajectory, which includes continuous turns and resemble real-life routes. The 15D KF is helpful to achieve more accurate results in a long time walking.

5. Conclusion and future work

This paper presented a model for achieving 3D indoor positioning based on foot-mounted sensors. We used ZUPT algorithm to detect the time when a pedestrian was standing still, and this information was used in Kalman filer to eliminate systematic errors. Based on our previous work, a 3D indoor positioning barometer was added and fused with accelerometer through a Kalman filter to obtain accurate height information, and we also extended the 2D model to the 3D model. We removed the PF due to its high computational time cost and the difficulty to implement in wearable devices, although it could help for positioning to some extent. The experiment results of the Simple trajectories showed that the proposed algorithm has performed well with different pedestrians. As for the experiments of complex trajectory, our algorithm performed better than the method proposed in [28] in the display of the track. Our trajectory matched the real route closely and could return to the start point after a round trip walk. In addition, we compared the proposed algorithm to our previous algorithm under the conditions of 9D KF and 15D KF, respectively. The new algorithm performed better with lower error rates and the higher trajectory stability. We determined that the main reason was that we obtained the accurate height information by the fusion of barometer and accelerometer data in the Kalman filter presented in Section 3.2, which was one of the main contributions in this paper. This method helped for getting a better result at both x axis and y axis of the position. Because of the rotation matrix, the three-axis coordinate interacts in the PDR algorithm. The 15D KF had a better performance than that of the 9D KF for a longer trip, because it could effectively eliminate the heading error caused by the gyroscope drift. In a shorter trip, the 9D KF and the 15D KF had a similar performance. The 15D KF consumed more computational cost. Therefore it was important to choose a cost-effective algorithm according to the trip distance. The testing of the system had demonstrated that the proposed algorithm could be effective used in indoor navigation. Future work will focus on the implementation of the Kalman filter to fuse gyroscope data and magnetic sensor data to correct the orientation error, to reduce the gyroscope drift on a long trip, so that a long term stable orientation solution can be achieved and the position information within a large area can be further improved. Since the technology will be used on wearable devices, we will continue working toward reducing the computational time in this system.

Acknowledgment

References

[1] F. Seco, J. Prieto, J. Guevara, et al., Indoor pedestrian navigation using an INS/EKF framework for yaw drift reduction and a foot-mounted IMU, in: Proceedings of 2010 7th Workshop on Positioning Navigation and Communication (WPNC), IEEE, 2010, pp. 135–143.

[2] A. Bacak, H. Celebi, Practical considerations for RSS RF fingerprinting based indoor localization systems, in: Proceedings of 2014 22nd Signal Processing and Communications Applications Conference (SIU), IEEE, 2014, pp. 497–500.

[3] E. Foxlin, Pedestrian tracking with shoe-mounted inertial sensors, IEEE Comput. Graph. Appl. 25 (6) (2005) 38–46.

[4] M. Nilsson, J. Rantakokko, M.A. Skoglund, G. Hendeby, Indoor positioning using multi-frequency RSS with foot-mounted INS, in: Proceedings of the Fifth International Conference on Indoor Positioning and Indoor Navigation (IPIN2014), 27–30 October 2014, Busan, Korea, 2014.

[5] D. Titterton, J.L. Weston, Strapdown Inertial Navigation Technology, vol. 17, IET, 2004.

[6] C. Feng, W.S.A. Au, S. Valaee, Z. Tan, Received-signal-strength-based indoor positioning using compressive sensing, IEEE Trans. Mobile Comput. 11 (12) (2012) 1983–1993.

[7] B. Ferris, D. Fox, N.D. Lawrence, WiFi-SLAM using Gaussian process latent variable models., in: Proceedings of IJCAI, vol. 7, 2007, pp. 2480–2485.

[8] R. Aversa, B. Di Martino, M. Ficco, S. Venticinque, A simulation model for localization of pervasive objects using heterogeneous wireless networks, Simul. Model. Pract. Theory 19 (8) (2011) 1758–1772.

[9] L. IndoorAtlas, Ambient Magnetic Field-based Indoor Location Technology: Bringing the Compass to the Next Level, IndoorAtlas Ltd. (2012).

[10] I. Wieser, A.V. Ruiz, M. Frassl, M. Angermann, J. Mueller, M. Lichtenstern, Autonomous robotic SLAM-based indoor navigation for high resolution sampling with complete coverage, in: Proceedings of 2014 IEEE/ION Position, Location and Navigation Symposium, PLANS 2014, IEEE, 2014, pp. 945–951.

[11] Y. Chen, H. Hu, Internet of intelligent things and robot as a service, Simul. Model. Pract. Theory 34 (2013) 159–171.

[12] Y. Chen, Z. Du, M. García-Acosta, Robot as a service in cloud computing, in: Proceedings of 2010 Fifth IEEE International Symposium on Service Oriented System Engineering (SOSE), IEEE, 2010, pp. 151–158.

[13] B. Krach, P. Robertson, Integration of foot-mounted inertial sensors into a Bayesian location estimation framework, in: Proceedings of the 5th Workshop on Positioning, Navigation and Communication, 2008. WPNC, IEEE, 2008, pp. 55–61.

[14] P. Robertson, M. Angermann, B. Krach, Simultaneous localization and mapping for pedestrians using only foot-mounted inertial sensors, in: Proceedings of the 11th International Conference on Ubiquitous computing, ACM, 2009, pp. 93–96.

[15] Y. Nam, J.W. Chong, Efficient simulation for positioning and utilizing of multiple cameras, Simul. Model. Pract. Theory 34 (2013) 37–47.

[16] L. Bruno, P. Robertson, WiSLAM: Improving FootSLAM with WiFi, in: Proceedings of 2011 International Conference on Indoor Positioning and Indoor Navigation (IPIN), IEEE, 2011, pp. 1–10.

[17] J. Bird, D. Arden, Indoor navigation with foot-mounted strapdown inertial navigation and magnetic sensors [emerging opportunities for localization and tracking], IEEE Wirel. Commun. 18 (2) (2011) 28–35.

[18] P. Davidson, J. Collin, J. Takala, Application of particle filters for indoor positioning using floor plans, in: Ubiquitous Positioning Indoor Navigation and Location Based Service (UPINLBS), 2010, IEEE, 2010, pp. 1–4.

[19] Y. Chen, A. Sabnis, M. Garcia-Acosta, Design and performance evaluation of a service-oriented robotics application, in: Proceedings of the 29th IEEE International Conference on Distributed Computing Systems Workshops, 2009, ICDCS Workshops'09, IEEE, 2009, pp. 292–299.

[20] M.G. Puyol, M. Frassl, P. Robertson, Collaborative mapping for pedestrian navigation in security applications, in: Future Security, Springer, 2012, pp. 49–60.

[21] P. Robertson, M. Angermann, B. Krach, M. Khider, Inertial systems based joint mapping and positioning for pedestrian navigation, in: Proceedings of International Conference on ION GNSS, 2009.

[22] O. Bebek, M. Suster, S. Rajgopal, M.J. Fu, X. Huang, M.C. Cavusoglu, D.J. Young, M. Mehregany, V. Den Bogert, A.J. Ton, et al., Personal navigation via shoe mounted inertial measurement units, in: Proceedings of 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), IEEE, 2010, pp. 1052–1058.

[23] L. Zheng, W. Zhou, W. Tang, X.Z. Yang, S. Pu, C. Li, B. Tang, Y. Chen, A foot-mounted sensor based 3D indoor positioning approach, Autonomous Decentralized Systems (ISADS), 2015 IEEE Twelfth International Symposium on, IEEE, 2015, pp. 145–150.

[24] N. Maluf, K. Williams, Introduction to Microelectromechanical Systems Engineering, Artech House, 2004.

[25] M. Bevermeier, O. Walter, S. Peschke, R. Haeb-Umbach, Barometric height estimation combined with map-matching in a loosely-coupled Kalman-filter, in: Proceedings of 2010 7th Workshop on Positioning Navigation and Communication, WPNC, IEEE, 2010, pp. 128–134.

[26] X. Zheng, H. Yang, W. Tang, S. Pu, L. Zheng, H. Zheng, B. Liao, J. Wang, Indoor pedestrian navigation with shoe-mounted inertial sensors, in: Multimedia and Ubiquitous Engineering, Springer, 2014, pp. 67–73.

[27] X. Chen, P. Yang, Y. Chen, Analysis and processing on zero position error of MEMS gyroscope, Chin. J. Sens. Actuators 5 (2012) 017.

[28] J.-O. Nilsson, I. Skog, P. Handel, K. Hari, Foot-mounted ins for everybody-an open-source embedded implementation, in: Proceedings of 2012 IEEE/ION Position Location and Navigation Symposium, PLANS, IEEE, 2012, pp. 140–145.

[29] APM Copter, APM 2.5 and 2.6 overview – arducopter, http://copter.ardupilot.com/wiki/common-apm25-and-26-overview/ (accessed 10.12.15).