

# A Multi-Model Fusion-Based Indoor Positioning System Using Smartphone Inertial Measurement Unit Sensor Data

PRISCILLA ADONG<sup>1</sup>, ODONGO STEVEN EYOBU<sup>1</sup>, TONNY J. OYANA<sup>1</sup> AND DONG SEOG HAN<sup>2</sup>

<sup>1</sup>School of Computing Informatics Technology, Makerere University, Plot 56, Pool Road, P.O. Box 7062, 5 Kampala, Uganda; apriscilla@cis.mak.ac.ug, toyana@cis.mak.ac.ug

<sup>2</sup>School of Electronics Engineering, Kyungpook National University, 80 Daehak-ro, Buk-gu, Daegu 41566, 7 Korea; dshan@knu.ac.kr

Corresponding author: Odongo Steven Eyobu (e-mail: sodongo@cis.mak.ac.ug).

**ABSTRACT** We propose novel multi-model fusion-based step detection and step length estimation approaches that use the Kalman filter. The proposed step detection approach combines results from three conventional step detection algorithms, namely, findpeaks, localmax, and advanced zero crossing to obtain a single and more accurate step count estimate. The proposed step length estimation approach combines results from two popular step length estimation algorithms namely Weinberg's and Kim's methods. In our experiment, we consider five different smartphone placements, that is, when the smartphone is handheld, handheld with an arm swing, placed in the backpack, placed in a trousers' back pocket and placed in a handbag. The system relies on inertia measurement unit sensors embedded in smartphones to generate accelerometer, gyroscope and magnetometer values from the human subject's motion. Results from our experiments show that our proposed fusion based step detection and step length estimation approaches outperform the conventional step detection and step length estimation algorithms, respectively. Our Kalman fusion approach achieves a better step detection, step length estimation for all the five smartphone placements hence providing a better positioning accuracy. The performance of the proposed multi-model fusion-based positioning system was measured using the root mean square error (RMSE) of the displacement errors and step count errors exhibited by all the step length and step count algorithms. The results show that the proposed Kalman fusion approach for step count estimation and step length estimation provides the least RMSE for all the smartphone placements. The proposed approach provides an average RMSE of 0.26 m in terms of the final position estimate for all the smartphone placements.

**INDEX TERMS** Indoor positioning, Kalman filter, Multi-model Fusion, Pedestrian dead reckoning, Smartphone IMU sensors.

## I. INTRODUCTION

Smartphones are one of the most popular computing devices today. According to Statista [1], the current number of smartphone users is approximately 3.5 billion and is expected to grow to 3.8 billion in 2021. Smartphones are embedded with several sensors and communication modules such as global positioning system (GPS) modules, inertia measurement unit (IMU) sensors, light, temperature and proximity sensors. These modules have enhanced smartphone capabilities by enabling the provision of services such as location-based services (LBS) [2], [3]. The deployment of IMU sensors *i.e* accelerometer, gyroscope and magnetometer in smartphones

and the growing number of smartphone users makes smartphone IMU-based indoor positioning feasible. Additionally, smartphone IMU sensor-based positioning systems do not require installation of additional infrastructure making it cheaper as compared to other popular indoor positioning system (IPS) approaches like Wi-Fi-based and ultra-wideband based systems [2], [4], [5]. Pedestrian dead reckoning (PDR) is a popular technique for obtaining a pedestrian's position in indoor environments. With PDR, it is possible to detect steps, estimate step length and heading of a pedestrian using data generated from IMU sensors. PDR based approaches have advantages of high availability, immunity to dynamic

changes in the environment and low deployment and maintenance costs [5]. However, traditional approaches based on PDR suffer from drift errors which increase with walking distance [2], [4].

Indeed, a significant number of studies have been carried out to improve the accuracy of PDR based IPS. Some of the proposed solutions include combining PDR with other technologies such as Wi-Fi and Bluetooth, constraining smartphone placements and using machine learning techniques [2], [5].

In this study, we propose a new multi-model fusion based step detection and step length estimation algorithm that uses the Kalman filter to correct errors from various conventional step detection and step length estimation algorithms, respectively. Our step detection algorithm combines results from three conventional step detection algorithms *i.e.* findpeaks, localmax and advanced zero crossing to obtain a single and more accurate step count estimate. Our step length estimation algorithm combines results from two popular step length estimation algorithms *i.e.* Weinberg's method and Kim's method. To evaluate the performance of our proposed algorithm, we carry out experiments for five popular phone placements *i.e.* hand held in front of the chest, hand held with an arm swing, placed in a backpack, placed in a trousers' back pocket and placed in a handbag. Comparisons with conventional step detection and step length estimation algorithms are presented. The main contributions of this paper are:

- A novel multi-model fusion step detection and step length estimation algorithm using the Kalman filter with capabilities of accurate step detection and step length estimation, respectively with the mobile smartphone placed at various positions.
- A performance analysis of conventional step detection and step length estimation techniques based on different smartphone positions held by the human subject.

The rest of this paper is organized as follows. Related works are discussed in Section II, technical preliminaries are presented in Section III, our proposed approach is discussed in Section IV, experiments and data collection are discussed in Section V, results are presented and discussed in Section VI and finally the conclusion and future works are presented in Section VII.

## II. RELATED WORKS

The PDR algorithm has three major components *i.e.* step detection, step length estimation and heading estimation. The accuracy of each component algorithm has a direct impact on the robustness of an IPS.

Various approaches for step detection exist in previous studies. The most popular approaches are based on constant thresholding commonly referred to as zero crossing, mean, standard deviation, windowed peak detection, and feature clustering. However, it has been reported in [6] that no single algorithm is sufficient for step detection. For instance, in [6], it is noted that methods based on applying a threshold to the standard deviation of the acceleration magnitude were very

robust in identifying steps but produced many false positives. Notably, methods based on windowed peak detection produced the overall optimal results with a median error of 1.3% regardless of smartphone placement. Several studies have therefore implemented hybrid algorithms to improve step detection accuracy. Some intelligent learning [7], [8] and hybrid [4], [9], [10] algorithms are presented and reviewed in the next paragraphs. In [10], a step detection algorithm that combines thresholding and windowed peak detection is presented. A fast Fourier transform (FFT) based smoother was applied on the magnitude of acceleration to remove interference signals before step detection. The average distance estimation error rate reported was 5% for all walking speeds. The proposed approach in [10] exhibits promising results however, too many thresholds and step detection rules are applied to the signal which makes the algorithm very complex. Other studies like [4], [9] have recommended the use of probabilistic methods such as the particle filter and Kalman filter to account for errors of over counting and under counting steps. In [4], the Kalman filter was used to fuse accelerometer and gyroscope data for pitch and roll estimation. Sensor fusion was used to reduce accumulated error from the accelerometer and gyroscope in order to obtain a better pitch and roll estimate. A step was detected if the pitch value was above a predefined threshold. In [9], steps were detected from either the magnitude of accelerometer output or the magnitude of gyroscope output using their novel step detection algorithm. A Kalman filter based step restoration algorithm was used to detect missed steps. They observed that the maximum acceleration peak occurs almost periodically during walking. By using the Kalman filter, they were able to estimate walking step time and determine when a step was taken. Results from their step detection algorithm are used as input for the Kalman filter. Studies in [5] and [8] have explored machine learning techniques for step detection. Machine learning models such as long short-term memory (LSTM) were used to differentiate locomotion modalities, phone placements and interfering activities such as calling and texting before step detection. Results showed that these techniques are robust for step detection and allow unconstrained use of smartphones. However, machine learning tasks are compute-intensive and consume a lot of energy hence reducing battery life of small computing devices like smartphones. The high complexities and/or low accuracy of the existing step detection algorithms motivated us to develop a new step detection algorithm.

In PDR, step length is used to translate steps into distance travelled by an object or pedestrian. Step length varies from one person to another due to varying heights, walking speed and walking style [2]. Step length can vary up to 40% from person to person for the same walking speed and up to 50% for different walking speeds [11]. Simpler approaches to step length estimation assume that step length is fixed which is usually an average for that user. Using a fixed step length results in large cumulative errors in distance estimation and only performs well with dominant walking patterns [2]. On

the other hand, advanced approaches compute step length for each individual step. This matches the reality as steps length may vary even for the same user for a given walking speed.

The renowned dynamic step length estimation algorithm presented by Weinberg [11] is based on the vertical displacement of the hip which is related to difference between the maximum and minimum acceleration. The method contains an adjustable constant which is dependant on height, weight and gender of the pedestrian. Kim *et al.* [12] developed a dynamic step length estimation algorithm that relates step length for each step to the measured acceleration and an adjustable constant. Wang *et al.* [13] present an improved step length estimation method based on Weinberg's method [11] and Kim's method [12]. They include the time for each step which reflects the speed of walking. The formula is

$$\begin{aligned} \text{steplength} = & k_1 \times \text{mean}(\text{delta\_a}) \times t_s \\ & + k_2 \sqrt[4]{\text{acc}^p - \text{acc}^v} + \gamma \end{aligned} \quad (1)$$

where  $k_1$  and  $k_2$  are scale factors,  $t_s$  is the time of the step process and  $\gamma$  is an offset.  $\text{acc}^p$  and  $\text{acc}^v$  are the local maxima and local minima, respectively and  $\text{mean}(\text{delta\_a})$  is the average acceleration at each time step. Results from their experiments showed that the proposed algorithm is more accurate than Weinberg's method [11] and Kim's method [12]. The maximum error in step length was about 5 cm [13].

Mathematical models presented by [11], [12] and [13] estimate step length with decent accuracy but require calibration for each individual user. Some studies have proposed sensor fusion based techniques for step length estimation. Sensor fusion provides a corrective measure for over estimation or under estimation thereby eliminating the need for calibration.

In [14], a step length estimation algorithm based on an indoor navigation system-zero velocity update (INS-ZUPT) Kalman filter is presented. The Kalman filter is used to combine pseudo-measurements of zero-velocity and zero-rotation. A moving variance of acceleration magnitude and a moving average of angular rate magnitude are used to detect ZUPT periods. The major draw back for this algorithms is the accumulated error that results from filter prediction [14]. The study by Poulouse *et al.* [4] combines readings from the accelerometer and gyroscope using the Kalman filter to obtain pitch values. A first-order linear regression model on the pitch amplitude is used to estimate the step length. Sensor fusion was used to reduce the accumulated error from the accelerometer and gyroscope in order to get a more reliable step length estimate [4].

The direction of movement can be obtained using gyroscope and magnetometer data. Integrating gyroscope readings gives the angular displacement of a device while magnetometer readings can be used to obtain absolute orientation with respect to the magnetic north [2]. However the magnetometer is susceptible to magnetic disturbances which cause short term heading estimation errors of up to  $100^\circ$  [2]. The gyroscope is immune to magnetic disturbances but suffers drift error which accumulates over time. This error is a result

of integration. In order to minimise error accumulation, studies have suggested fusing readings from the gyroscope and magnetometer using methods like the particle filter, Kalman filter and complementary filtering.

Li *et al.* [15] used the particle filter to combine data from the accelerometer, gyroscope and magnetometer to obtain heading information at each step. Furthermore an algorithm that corrects the particle direction in the particle filter was developed to minimise heading estimation errors. In their study, a 95 percentile error of  $23^\circ$  was reported. Poulouse *et al.* [4] used the Kalman filter to combine readings from the gyroscope and magnetometer for heading estimation and reported a mean error of  $4.72^\circ$  [4], [16]. Fusion of the magnetometer and gyroscope reading improves heading estimation however, with time the errors are also accumulated [17].

Fusion of estimated heading with virtual landmarks has been proposed to solve the problem of error accumulation [17], [18]. Virtual landmarks vary from distinct patterns in data collected by IMU sensors, to unique RSS from nearby Wi-Fi access points and specified points on the building's floor plan. The precise position of these landmarks are usually known. Once a landmark is detected, corrections are made to the estimated heading. Li *et al.* [17] proposed a real-time heading compensation algorithm that uses the Kalman filter to combine heading information with virtual landmarks. They identify three regions on the map of the building as landmarks and used least squares for comparative analysis while Deng *et al.* [18] used the unscented Kalman filter to fuse heading information with identified landmarks such as stairs, escalators and elevators. Nevertheless, even with the reduced heading error, using landmarks in heading estimation requires that a PDR algorithm be customised to a specific environment which in turn affects the flexibility of the IPS.

### III. TECHNICAL PRELIMINARIES

#### A. FINDPEAKS

Findpeaks is based on mean and standard deviation of the acceleration magnitude. To detect steps, we computed the magnitude of acceleration from raw accelerometer readings using

$$|a| = \sqrt{x^2 + y^2 + z^2} \quad (2)$$

where  $a$  is the acceleration,  $x$ ,  $y$  and  $z$  are sensor readings for the  $x$ ,  $y$ , and  $z$  axes, respectively. The magnitude allows large changes in acceleration to be detected regardless of the phone's orientation. The mean magnitude  $\text{meanmag}$  was subtracted from the magnitude of acceleration  $\text{mag}$  to remove any constant effects such as gravity from the data. We then applied the *findpeaks* function from the Matlab toolbox to detect peaks. A step was detected when the minimum height  $\text{minPeakHeight}$  of the acceleration magnitude was above one standard deviation  $\text{std}(\text{meanmag})$ . This method is simple, has low processing time and is very robust in identifying steps however, it produces many false positives [6], [19].

## B. LOCALMAX

Localmax is based on windowed peak detection of local minimum and local maximum. To detect steps, we smoothed the acceleration magnitude using a moving average filter with a defined window size. Local minimas and local maximas from the smoothed signal are then identified. In this study, a step was detected where there was a strong local maxima between two consecutive local minimas within a predefined window. This algorithm is very robust in detecting steps. However, it detects many false positives especially when the signal is noisy. To eliminate false steps, we set a minimum local maxima separation of 5 samples. Therefore, a local maxima was only considered to be a step if it was at least 5 samples away from the previous local maxima. This algorithm was adopted from [6], [20].

## C. ADVANCED ZERO CROSSING

To detect steps, gravity ( $9.8 \text{ m/s}^2$ ) is subtracted from the magnitude of acceleration. A step is then detected if the magnitude had a zero crossing *i.e* goes from a negative to a positive value and is above the predefined upper or lower threshold. The thresholds are set to eliminate signals generated when the pedestrian is not walking, that is, signal generated due to others random movements. These signals usually have a lower acceleration magnitude. The thresholds were computed using the formula

$$\text{threshold} = \pm c \times STD_{\text{rest}} \quad (3)$$

where  $c$  is a constant such that  $c \geq 1$ , and  $STD_{\text{rest}}$  is the standard deviation of the acceleration when the pedestrian is resting *i.e* sitting or standing. Our  $STD_{\text{rest}}$  is an average of the standard deviation of acceleration when the user is seated and standing. This algorithm has low processing time but produces many false positives if the threshold is too low and misses many steps if threshold is too high. In addition to this it is challenging to find a threshold that is optimal for various smartphone placements. The algorithm was adopted from [21].

## D. WEINBERG'S METHOD

Weinberg's method is related to difference between the maximum and minimum acceleration. A step length is defined as

$$\text{steplength} \simeq l \times \sqrt[4]{A_{\text{max}} - A_{\text{min}}} \quad (4)$$

where  $A_{\text{max}}$  and  $A_{\text{min}}$  are the maximum and minimum acceleration values for a single step, respectively and  $l$  is a parameter whose value is dependant on the height, weight and gender of the pedestrian [11].

## E. KIM'S METHOD

Kim's method relates step length to measured acceleration in each step. The formula is given by

$$\text{steplength} = 0.98 \times \frac{\sqrt[3]{\sum_{i=1}^n |a_i|}}{n} \quad (5)$$

where  $a_i$  is the  $i^{\text{th}}$  sample for a single step and  $n$  is the number of samples corresponding to each step [12].

## F. THE BASIC KALMAN FILTER ALGORITHM

The Kalman filter is a recursive algorithm used to estimate a system state when it cannot be measured directly given the system model and state measurement updates. The Kalman filter is popular and efficient for sensor and data fusion [22], [23], [24]. The Kalman filter is efficient because of its inexpensive computational requirements. The recursive nature of the Kalman filter allows it to be executed in real-time without storing the history of observations. This makes it suitable for applications that require frequent updates for example positioning applications [23]. There are three varieties of the Kalman filter namely, the conventional Kalman filter designed for linear systems, the extended Kalman filter (EKF) and unscented Kalman filter (UKF) designed for non-linear systems. The EKF and UKF are based on the conventional Kalman filter. The Kalman filter algorithm consists of two phases *i.e* the predict and update phase. The conventional Kalman filter algorithm is defined in the following procedure

Step 1: Set the initial values  $\hat{x}_0$  and  $P_0$

Step 2: Predict and error co-variance

$$\hat{x}_k^- = A\hat{x}_{k-1} + Bu_k + w_k \quad (6)$$

$$P_k^- = AP_{k-1}A^T + Q \quad (7)$$

Step 3: Compute the Kalman gain

$$K_k = \frac{P_k^- C^T}{C P_k^- C^T + R} \quad (8)$$

Step 4: Read the Measurement value  $z$

Step 5: Compute the estimate  $\hat{x}$

$$\hat{x}_k = \hat{x}_k^- + K_k(z_k - C\hat{x}_k^-) \quad (9)$$

Step 6: Compute the error co-variance

$$P_k = (I - K_k C) P_k^- \quad (10)$$

Step 7: Go back to step 2 until the estimate converges

where

$\hat{x}$  is the state.

$A$  is the state transition model.

$B$  is the control input model.

$u$  is the control vector.

$P$  is the error covariance.

$Q$  is the process noise covariance.

$C$  is the observation matrix.

$R$  is the measurement noise covariance.

$z$  is the noisy measurement.

$w$  is the process noise with zero mean and with covariance,

$Q_k : \mathbf{w}_k \sim \mathcal{N}(0, \mathbf{Q}_k)$ .

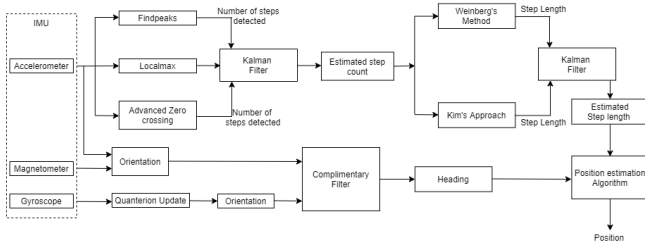


FIGURE 1. The conceptual diagram of our proposed approach.

In this study, for step count estimation, the measurement value and the measurement noise co-variance are customized using step count values realised from the find peaks, localmax and zero crossing approaches. Similarly for the step length estimation, the the measurement value and the measurement noise co-variance are customized using step length values realised from Weinberg's approach and Kim's method.

#### IV. PROPOSED MULTI-MODEL FUSION PDR-BASED IPS

Fig. 1 presents the proposed approach using a block diagram. The next subsections present the application of the step counts and Step lengths achieved from the previously mentioned models in the Kalman filter. Since position estimation requires knowledge on the heading, a conventional heading estimation is also presented for purposes of feeding into the final position estimation equation together with the step length and step count information.

##### A. STEP DETECTION

The Kalman filter was used to estimate the number of steps taken by the participants at regular intervals of 1s. The initial state variables  $x_0$  and initial error co-variance matrix  $P_0$  are given by

$$x_0 = 1 \quad (11)$$

$$P_0 = 1 \quad (12)$$

The state of system was defined by  $x_k$ . For this study, the dynamic equation of the system is given by

$$x_k = Ax_{k-1} + w_k \quad (13)$$

where  $A = 1$  is our system state matrix,  $k = 1, 2, 3, \dots, n$  where  $n$  is the total time in seconds and  $w_k$  is random noise with zero mean and co-variance matrix where  $Q = 0.025$ . The estimated step counts  $z_f$ ,  $z_l$  and  $z_a$  from findpeaks, localmax and advanced zero crossing algorithms respectively, were used to compose the measurement value  $z$  used in the Kalman filter such that:

$$z = \begin{bmatrix} z_f & 0 & 0 \\ 0 & z_l & 0 \\ 0 & 0 & z_a \end{bmatrix} \quad (14)$$

Again, for this study, the measurement noise covariance matrix  $R$  is

$$R = \begin{bmatrix} R_f & 0 & 0 \\ 0 & R_l & 0 \\ 0 & 0 & R_a \end{bmatrix} \quad (15)$$

where  $R_f$ ,  $R_l$  and  $R_a$  are the measurement noise covariances from the findpeaks, localmax and advanced zero crossing algorithms, respectively. The observational matrix  $H$  is defined by

$$H = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (16)$$

##### B. STEP LENGTH ESTIMATION

The initial state variables  $x_0$  and initial error co-variance matrix  $P_0$  are given by

$$x_0 = 0.5 \quad (17)$$

$$P_0 = 1 \quad (18)$$

The state of the Kalman filter is defined by  $x_k$ . The dynamic equation of the system is defined by

$$x_{k+1} = Ax_k + w_k \quad (19)$$

where  $A = 1$  is our system state matrix and  $w$  is random noise with zero mean and co-variance matrix  $Q$  where  $Q = 0.0001$ .

The step length estimates  $z_w$  and  $z_k$  from Weinberg's methods and Kim's method, respectively were used as the measurement  $z$  for the Kalman filter, where

$$z = \begin{bmatrix} z_w & 0 \\ 0 & z_k \end{bmatrix} \quad (20)$$

The measurement noise co-variance matrix  $R$  is given by

$$R = \begin{bmatrix} R_w & 0 \\ 0 & R_k \end{bmatrix} \quad (21)$$

where  $R_w$  and  $R_k$  are the measurement noise co-variances from the Weinberg's methods and Kim's approach, respectively.

The observational matrix  $H$  is defined by

$$H = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad (22)$$

##### C. HEADING ESTIMATION

We used the complementary filter to estimate heading from the accelerometer, gyroscope and magnetometer through sensor fusion.

###### 1) Accelerometer

We use a low pass filter to smooth raw acceleration readings and then compute the pitch and roll values as follows:

$$\phi_{Acc} = \arctan \left( \frac{A_y}{\sqrt{A_x^2 + A_z^2}} \right) \frac{180}{\pi} \quad (23)$$

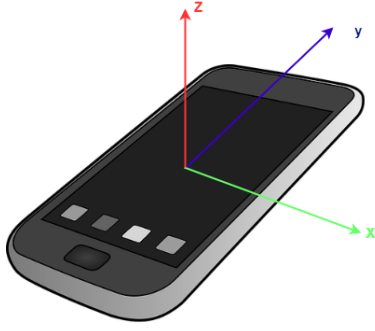


FIGURE 2. A Smartphone's coordinate system.

$$\theta_{Acc} = \arctan \left( \frac{A_x}{\sqrt{A_y^2 + A_z^2}} \right) \frac{180}{\pi} \quad (24)$$

where  $\phi_{Acc}$  is roll angle,  $\theta_{Acc}$  is the pitch angle,  $A_x$ ,  $A_y$ , and  $A_z$  are the raw acceleration readings for the  $x$ ,  $y$ , and  $z$  axes, respectively.

The yaw angle,  $\psi$  cannot be computed directly from acceleration, therefore we computed yaw from magnetometer readings.

#### 2) Magnetometer

The magnetometer determines a target's heading using the smartphone's coordinate system which may not be aligned with the global coordinate system depending on the phones placement. The smartphone coordinate system is depicted in Fig. 2. The global coordinate system is calculated using the formula

$$\begin{bmatrix} H_x \\ H_y \\ H_z \end{bmatrix} = \begin{bmatrix} \cos\phi & \sin\phi\sin\theta & -\sin\phi\cos\theta \\ 0 & \cos\theta & \sin\theta \\ \sin\phi & -\sin\theta\cos\phi & \cos\phi\cos\theta \end{bmatrix} \begin{bmatrix} h_x \\ h_y \\ h_z \end{bmatrix} \quad (25)$$

where  $H_x$ ,  $H_y$ ,  $H_z$  are  $x$ ,  $y$ , and  $z$  axes of the global coordinate system and  $h_x$ ,  $h_y$ ,  $h_z$  are  $x$ ,  $y$ , and  $z$  axes of the phone's coordinate system. Our yaw  $\psi_{mag0}$  is given by

$$\psi_{mag0} = \arctan \left( \frac{H_y}{H_x} \right) \quad (26)$$

#### 3) Sensor fusion of accelerometer and magnetometer

To calculate the tilt compensation for the yaw (heading) angle, we use the roll ( $\phi_{Acc}$ ) and pitch ( $\theta_{Acc}$ ) angles as follows:

$$M_x = H_x \cos(\phi_{Acc}) + H_z \sin(\phi_{Acc}) \quad (27)$$

$$M_y = H_x \sin(\theta_{Acc}) \sin(\phi_{Acc}) + H_y \cos(\theta_{Acc}) - H_z \sin(\theta_{Acc}) \cos(\phi_{Acc}) \quad (28)$$

where  $M_x$  and  $M_y$  are the tilt values along the  $x$  and  $y$  axes, respectively. Heading from the accelerometer and magnetometer  $\psi_{mag}$  is given by

$$\psi_{mag} = \arctan \left( \frac{M_y}{M_x} \right) \quad (29)$$

#### 4) Gyroscope

To obtain pitch, roll, and yaw angles from the gyroscope, We first computed Euler's angles from a rotation matrix using the formula

$$\begin{bmatrix} \phi \\ \theta \\ \psi \end{bmatrix} = \begin{bmatrix} \omega_x + \omega_y \sin(\phi) \tan(\theta) + \omega_z \cos(\phi) \sin(\theta) \\ \omega_y \cos(\phi) - \omega_z \sin(\phi) \\ \omega_y \sin(\phi) / \cos(\theta) + \omega_z \cos(\phi) \cos(\theta) \end{bmatrix} \quad (30)$$

where  $\omega_x$ ,  $\omega_y$ , and  $\omega_z$  are the  $x$ ,  $y$ , and  $z$  axes of the gyroscope, respectively.

We converted Euler's angles to quaternion values using the formula

$$\begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{bmatrix} = \begin{bmatrix} \cos \frac{\phi}{2} \cos \frac{\theta}{2} \cos \frac{\psi}{2} + \sin \frac{\phi}{2} \sin \frac{\theta}{2} \sin \frac{\psi}{2} \\ \sin \frac{\phi}{2} \cos \frac{\theta}{2} \cos \frac{\psi}{2} - \cos \frac{\phi}{2} \sin \frac{\theta}{2} \sin \frac{\psi}{2} \\ \cos \frac{\phi}{2} \sin \frac{\theta}{2} \cos \frac{\psi}{2} + \sin \frac{\phi}{2} \cos \frac{\theta}{2} \sin \frac{\psi}{2} \\ \cos \frac{\phi}{2} \cos \frac{\theta}{2} \sin \frac{\psi}{2} - \sin \frac{\phi}{2} \sin \frac{\theta}{2} \sin \frac{\psi}{2} \end{bmatrix} \quad (31)$$

Heading  $\psi_{gyro}$  estimate from the gyroscope was obtained using the formula

$$\psi_{gyro} = \arctan \left[ \frac{2(q_0 q_3 + q_1 q_2)}{1 - 2(q_2^2 + q_3^2)} \right] \quad (32)$$

#### 5) Complimentary Filter

Heading from the gyroscope is prone to drift errors over long periods of time. To correct the drift errors, we calibrated the gyroscope estimates using values from the accelerometer and magnetometer via the complimentary filter.

The complimentary filter is a combination of low pass filtering of the accelerometer and magnetometer orientation and high pass filtering of the gyroscope orientation. The complimentary filter has the following form

$$Heading = (1 - k)\psi_{gyro} + k\psi_{mag} \quad (33)$$

where  $1 - k$  is the high pass filter weight and  $k$  is the low pass filter weight.

#### D. POSITION ESTIMATION

We used the position estimation algorithm with the formula

$$X_p = X_{p-1} + SL \times \cos(HD) \quad (34)$$

$$Y_p = Y_{p-1} + SL \times \sin(HD) \quad (35)$$

where  $X_p$  and  $Y_p$  are position values on the  $x$  and  $y$  axes, respectively and  $X_{p-1}$  and  $Y_{p-1}$  are the initial position values, HD is heading and  $SL$  is step length.

The algorithm uses step length and heading to estimate the target's position with respect to the previously known position [4]. The initial target's position is assumed to be zero.

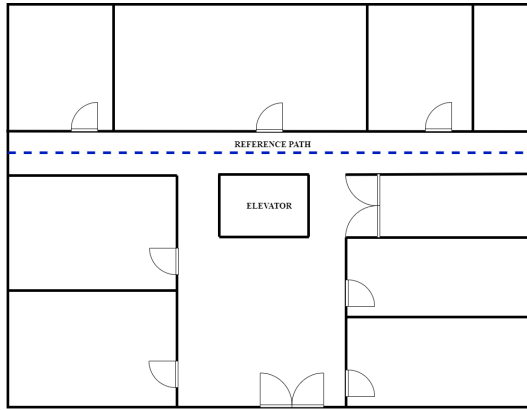


FIGURE 3. Plane view of the experiment scenario.

## V. EXPERIMENTS AND DATA

We collected the IMU sensor data used in this study. The experiment was carried out using an Infinix Note 5 smartphone running android version 9, with a 2.3 GHz Quad-core processor and 3GB RAM. Acceleration data from the smartphone's IMU sensors were retrieved using the AndroSensor app available on google play store at a sampling rate of 0.1 Hz and then processed using Matlab. The data collection environment is characterized by a flat floor/smooth surface in a storied building located in Makerere University College of Computing and Information Sciences where human subjects normally walk on. Sample data is generated from a middle-aged female participant and of height 1.6 m. This human subject was required to hold the phone in two different positions while walking along a straight path in a corridor on the third floor in the building facility. Five data collection scenarios were used. Our experimental design consisted of two scenarios. In scenario one involved her walking with a smartphone held in her hand at the chest angle, in the second scenario, she walked with a smartphone held in her hands while swinging the arm, the third scenario involved her walking with the smartphone placed in the backpack, the fourth scenario involved her walking with the smartphone placed in the trouser's back pocket and the last scenario involved her walking with the smartphone placed in the handbag. The five positions were used to generate acceleration data from the smartphone's IMU sensor. Fig. 3 shows the ground floor plan of the experiment environment and the dotted line shows the reference path.

To evaluate the performance of our proposed step counting algorithm, we compared the ground truth values with values estimated by the findpeaks, localmax, advanced zero crossing and our proposed step counting approach. For step length estimation we compared the ground truth values with the values estimated by the Weinberg method, Kim's method and our proposed step length estimation approach. The error metrics used include step error, absolute error, mean absolute error and the RMSE.

The step error was given by

$$\text{stepError} = TS - ES \quad (36)$$

where  $TS$  is the total number of actual steps counted and  $ES$  is the total number of steps estimated.

The absolute error was given by

$$ae = |est - gt| \quad (37)$$

where  $ae$  is the absolute error,  $est$  is the estimated value and  $gt$  is the ground truth value.

The RMSE given by

$$RMSE = \sqrt{\frac{\sum_{i=1}^n (\hat{p}_i - p_i)^2}{n}} \quad (38)$$

where  $\hat{p}_1, \hat{p}_2, \dots, \hat{p}_n$  are estimated step count at each time step,  $p_1, p_2, \dots, p_n$  are the ground truth values and  $n$  is the number of observed values.

We also evaluated the performance of the overall position estimation algorithm using the RMSE given by eq.(38) and displacement error  $d_e$  given by

$$d_e = x_t - x_o \quad (39)$$

where  $x_t$  is the estimated position at time  $t$  and  $x_o$  is the actual position at time  $t$ .

We evaluated the performance of the heading algorithm using the mean absolute error  $mae$  given by

$$mae = \frac{1}{n} \sum_{j=1}^n |h_j - \hat{h}_j| \quad (40)$$

Where  $n$  - is the number of observed values,  $h_j$  is the actual heading angle and  $\hat{h}_j$  is the estimated heading angle.

## VI. RESULTS

### A. GROUND TRUTH

For each smartphone placement, the participant was required to take 25 steps along the reference path, and multiple data traces were collected for each placement. We excluded all traces where the step count was not clear.

The actual stride lengths were determined by marking the ground for the entire length of the reference path, the participant was required to follow the markings on the ground.

The actual heading was determined using a compass.

### B. STEP DETECTION AND STEP COUNTING

Fig. 4, 5, and 6 graphically show the points at which steps were detected by findpeaks, localmax and advanced zero crossing.

Fig. 4 shows the performance findpeaks (peak detection) for the various smartphone placements used in this study. Out of the 25 steps taken by the participants, the algorithm detected 23, 20, 25, 24, and 24 steps as seen in Fig. 4a, 4b, 4c, 4d, and 4e, respectively. It's clear that the algorithm is robust

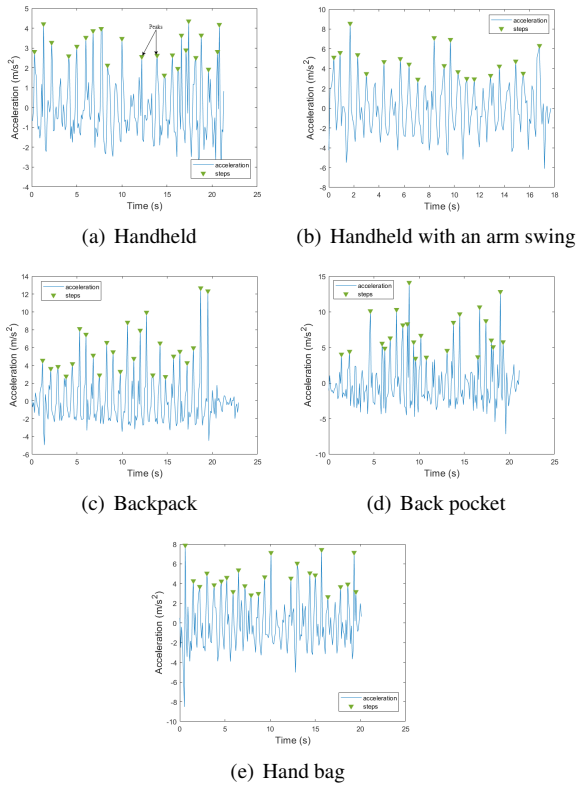


FIGURE 4. Peaks detected by the findpeaks algorithm.

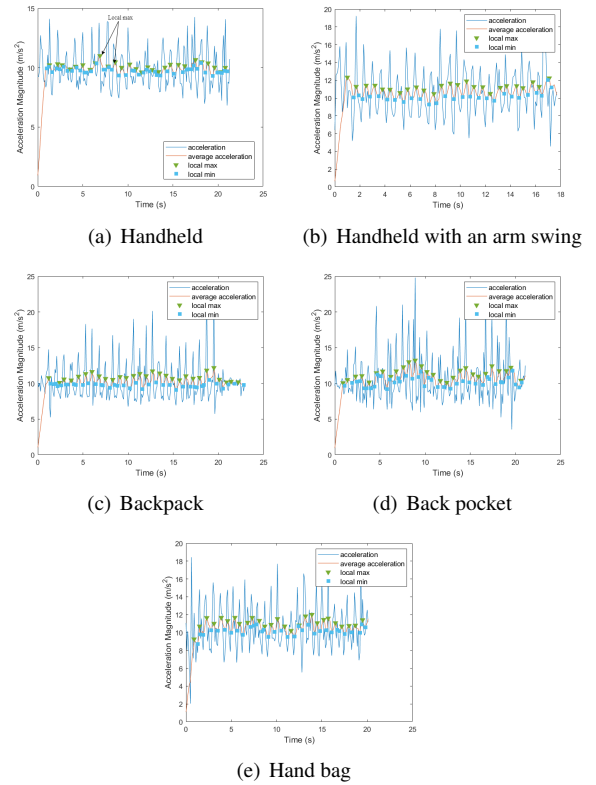


FIGURE 5. Points at which steps were detected by the localmax algorithm.

in identifying steps when the peaks generated are distinct and there is less noise in the signal. However, it misses steps when peaks are less noticeable as shown in Fig. 4b. The combined effects of the arm swing and leg movement generates signals with less conspicuous peaks making peak detection more complex. The algorithm also misses steps when the signal is noisy. The noisy signal overshadows the peaks.

Fig. 5, shows the performance of the localmax algorithm for the various smartphone placements used in this study. Out of the 25 steps taken by the participant, the algorithm detected 28, 25, 29, 28, and 27 steps as seen in Fig. 5a, 5b, 5c, 5d, and 5e, respectively. The algorithm generally detects false steps where signals are noisy. This is due to the fact that a signal even when not distinct, can be classified as a local maxima as long height of that signal is greater or equal to the height of all signals in a specified window. This is especially observable in Fig. 5c where the algorithm detected 4 local maximas/steps after the time 20s even when the signals are clearly not distinct enough to be classified as steps.

Fig. 6 shows the performance of the advanced zero crossing algorithm for the various smartphone placements used in this study. Out of the 25 steps taken by the pedestrian, the algorithm detected 25, 27, 28, 31, and 28 steps as seen in Fig. 6a, 6b, 6c, 6d, and 6e, respectively. The algorithm generally detects false steps where the noisy signals generated are strong enough to surpass the predefined lower and upper

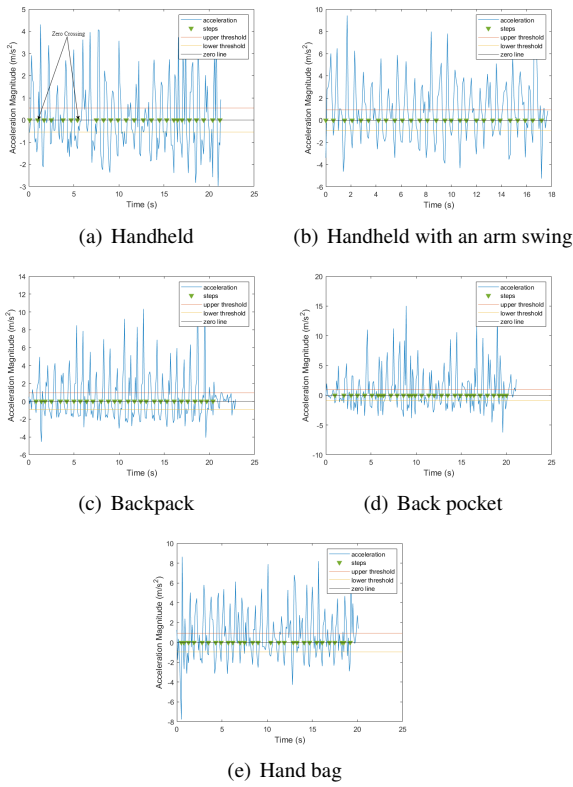
thresholds.

Fig. 7, 8, 9, 10, and 11 show the comparison between the actual number of steps vs the number of steps counted by findpeaks, localmax, advanced zero crossing and our approach for the various smartphone placements used in the study.

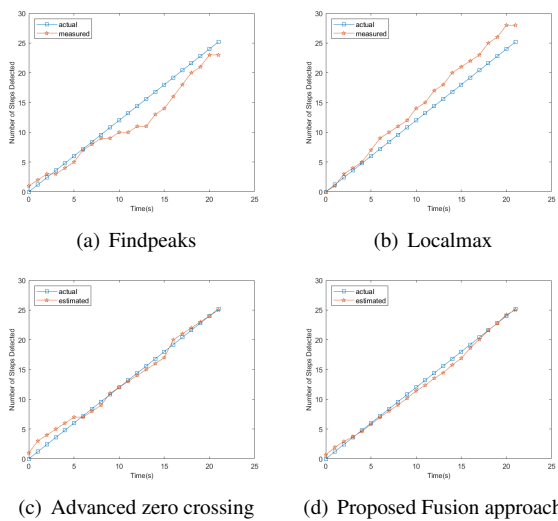
Fig. 7 shows the number of steps counted by each step detection/step counting algorithm when the smartphone was handheld. Our approach is able to correct the under counting by the findpeaks in Fig. 7a and over counting by the localmax algorithm in Fig. 7b and in addition improves the accuracy of step count at every given time interval.

Fig. 8 shows the number of steps counted by each step detection/step counting algorithm when the smartphone was handheld with an arm swing. Our approach is able to correct the under counting by the findpeaks algorithm in Fig. 8a and over counting by the advanced zero crossing algorithm in Fig. 8c and in addition improves the accuracy of step count at every given time interval.

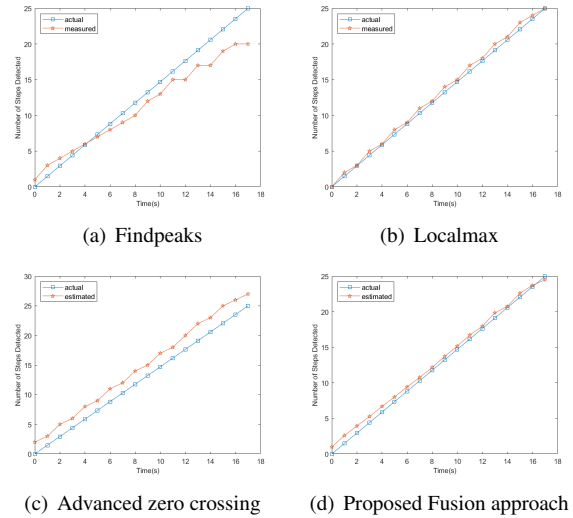
Fig. 9 shows the number of steps counted by each step detection/step counting algorithm when the smartphone was placed in a backpack. Our approach is able to correct the over counting by both localmax in Fig. 9b and advanced zero crossing algorithm in Fig. 9c. The estimated value by our proposed Kalman filter based fusion approach quickly converges to the true value and the approach is more accurate than findpeaks, localmax and advanced zero crossing.



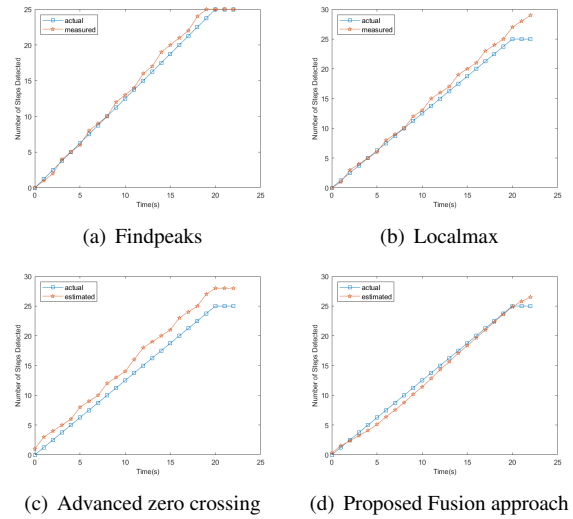
**FIGURE 6.** Points at which steps were detected by the advanced zero crossing algorithm



**FIGURE 7.** Step counting by findpeaks, localmax, advanced zero crossing and our proposed approach when the phone was handheld.



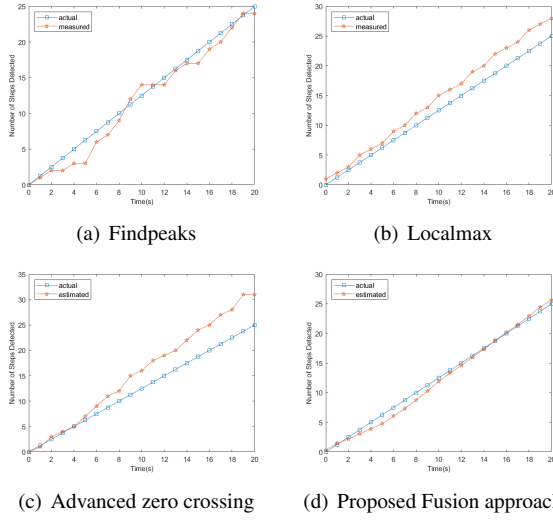
**FIGURE 8.** Step counting by findpeaks, localmax, advanced zero crossing and our proposed approach when the phone was handheld with an arm swing.



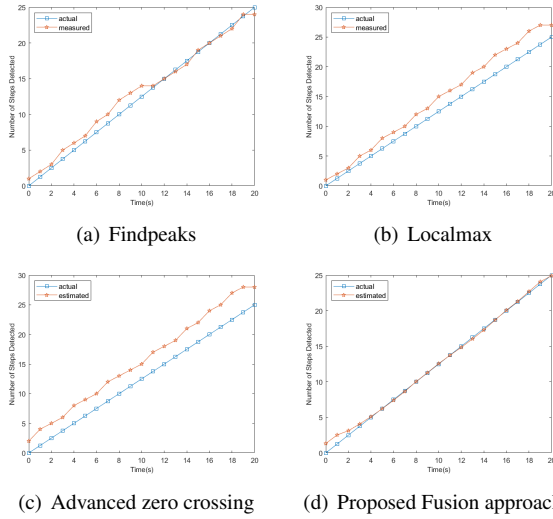
**FIGURE 9.** Step counting by findpeaks, localmax, advanced zero crossing and our proposed approach when the phone was placed in a backpack.

Fig. 10 shows the number of steps counted by each step detection/step counting algorithm when the smartphone was placed in a back pocket. Our approach is able to correct the under counting by findpeaks in 10a while correcting the over counting by both localmax in Fig. 10b and advanced zero crossing algorithm in Fig. 10c. The estimated value by our proposed Kalman filter based fusion approach quickly converges to the true value and is more accurate than findpeaks, localmax and advanced zero crossing.

Fig. 11 shows the number of steps counted by each step detection/step counting algorithm when the smartphone was placed in the handbag. Our fusion based approach is able to correct the under counting by findpeaks in 11a while correcting the over counting by both localmax in Fig. 11b and advanced zero crossing algorithm in Fig. 11c. The estimated



**FIGURE 10.** Step counting by findpeaks, localmax, advanced zero crossing and our proposed approach when the phone was placed in the back pocket.

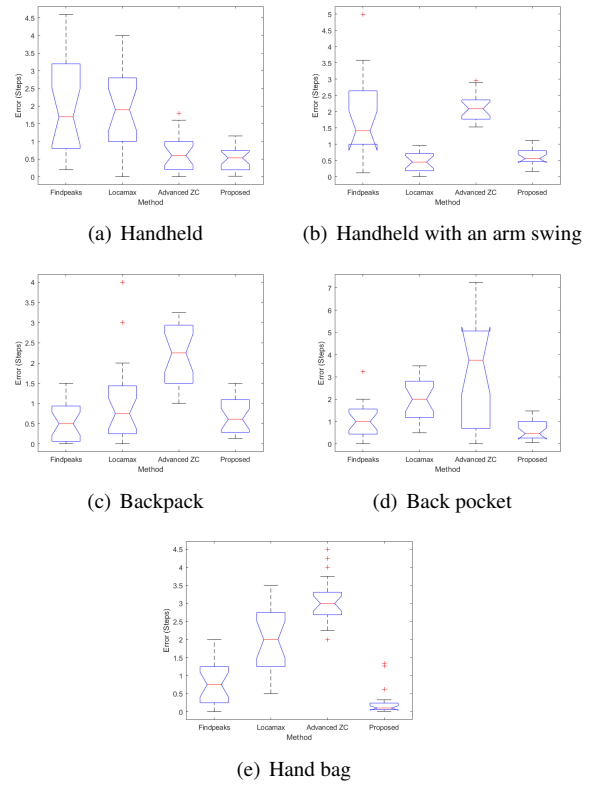


**FIGURE 11.** Step counting by findpeaks, localmax, advanced zero crossing and our proposed approach when the phone was placed in the hand bag.

value by our proposed Kalman filter based fusion approach quickly converges to the true value and is able to estimate the correct count at almost every time interval

We measured the accuracy of each step detection/step counting algorithm by comparing the estimated step count with the ground truth values at each time step. Fig. 12 shows the distribution of step count errors for the various smartphone placements considered in this study. using an error rate metric defined in eq. (37). For each step detection/ step counting approach, the median, lower and upper quartiles and whiskers of the data set are shown.

We observed that findpeaks, localmax and advanced zero crossing perform well for at least one smartphone placement. For instance advanced zero crossing in 12 (a), localmax in 12 (b) and findpeaks in 12 (c). The performance of these



**FIGURE 12.** Absolute step count errors for findpeaks, localmax, advanced zero crossing and the proposed step counting approach for all the smartphone placements considered.

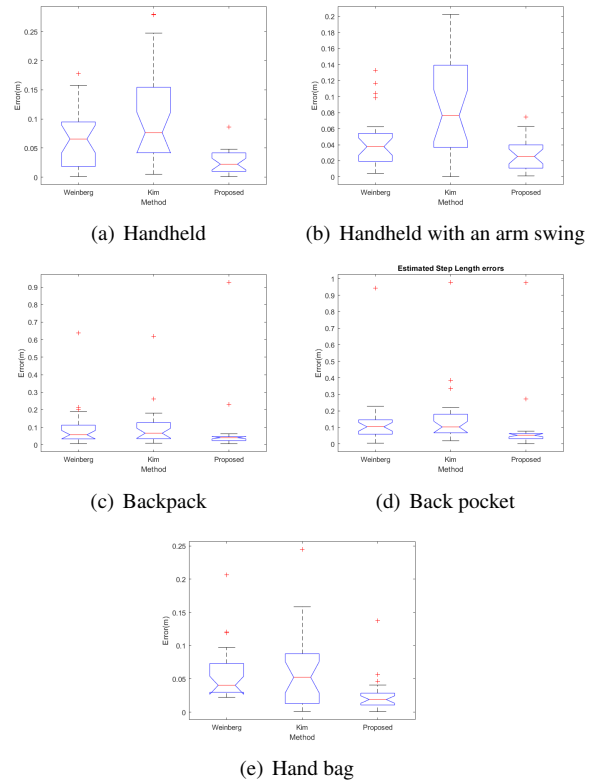
algorithms for each placement depends on the parameters used for example the window size in localmax and the upper and lower thresholds in advanced zero crossing. However, it was challenging to find optimal parameters that favour all smartphone placements.

On the other hand our proposed approach generally outperformed findpeaks, localmax and advanced zero crossing for all smartphone placements. This is true except for localmax when the phone was hand held with an arm swing and findpeaks when the phone was placed in the backpack. Even so, the performance of our algorithm is still good. The median, lower and upper quartiles for localmax when the smartphone was hand held with an arm swing are 0.45, 0.18, 0.71 as compared to 0.55, 0.46, and 0.8, respectively for the proposed algorithm. The median, lower and upper quartiles for findpeaks when the smartphone was placed in the backpack are 0.5, 0.0625, and 0.9 as compared to 0.6, 0.287, and 1, respectively for the proposed algorithm.

The summary of the step error, RMSE and absolute errors of the various step detection/step counting algorithms used in this study are given in Table 1. The step error is obtained by subtracting the estimated number of steps from the actual number of steps as shown in eq. (36). The absolute error is given by eq. (37) and RMSE is given by eq. (38). From Table 1, it's clear that our proposed step detection approach generally has the lowest step error, absolute error and RMSE

**TABLE 1.** Step errors, Absolute errors and RMSE for findpeaks, localmax, advanced zero crossing and the proposed step counting approach for all the smartphone placements considered.

Placement and Method	Actual number of steps	Number of steps estimated	Step Error (steps)	Absolute Error (m)	RMSE (m)
<b>Handheld</b>					
Findpeaks	25	23	2	1.7	2.2647
localmax	25	28	3	1.9	2.2365
AZC	25	25	0	0.6	0.8323
Proposed	25	25	0	0.53	0.6081
<b>Arm Swing</b>					
Findpeaks	25	20	5	1.14	2.1983
localmax	25	25	0	0.45	0.5409
AZC	25	27	2	2	2.1534
Proposed	25	25	0	0.55	0.6589
<b>Backpack</b>					
Findpeaks	25	25	0	0.5	0.7335
localmax	25	29	4	0.75	1.4046
AZC	25	28	3	2.25	2.2686
Proposed	25	26	1	0.6	0.7893
<b>Back pocket</b>					
Findpeaks	25	24	1	1	1.2979
localmax	25	28	3	2	2.2241
AZC	25	31	6	3.75	3.8506
Proposed	25	26	1	0.45	0.7826
<b>Hand bag</b>					
Findpeaks	25	24	1	0.75	0.9728
localmax	25	27	2	2	2.1971
AZC	25	28	3	3	3.1349
Proposed	25	25	0	0.1	0.4480



**FIGURE 13.** Absolute step length errors for Weinberg's method, Kim's method and our proposed approach for the various smartphone placements.

as compared to findpeaks, localmax and advanced zero crossing.

**C. STEP LENGTH ESTIMATION**

We evaluated the performance of our proposed step length estimation approach by comparing the RMSE and absolute errors with the RMSE and absolute errors of Weinberg's and Kim's method. The RMSE was used to determine the overall performance of the various step length estimation algorithms while the absolute error was used to determine the accuracy of the various step length estimation methods at each step. The formula for absolute errors and the RMSE were defined in eq. (37) and eq.(38), respectively.

Fig. 13 shows the distribution of step length estimation errors for the various smartphone placements considered in this study. The error metric used to calculate the errors was the absolute error. For each stride length estimation approach, the median, lower and upper quartiles and whiskers of the data are shown.

The median errors vary between 0.037 m to 0.1 m for Weinberg's approach, 0.05 m to 0.1 m for Kim's approach and 0.018 m to 0.06 m for the proposed approach. Kim's approach has the widest dispersion and tends to over estimate the step length while our proposed approach has the overall best performance.

Table 2 shows the summary of absolute errors and RMSE for Weinberg's method, Kim's approach and our proposed

**TABLE 2.** Absolute step length error and RMSE for Weinberg's method, Kim's method and our proposed approach for the various smartphone placements.

Placement and Method	Absolute Error (m)	RMSE (m)
<b>Handheld</b>		
Weinberg	0.065	0.0846
Kim	0.076	0.1287
Proposed	0.022	0.0327
<b>Arm Swing</b>		
Weinberg	0.037	0.0605
Kim	0.076	0.1091
Proposed	0.025	0.0331
<b>Backpack</b>		
Weinberg	0.057	0.1562
Kim	0.065	0.1575
Proposed	0.040	0.1908
<b>Back pocket</b>		
Weinberg	0.1	0.2186
Kim	0.1	0.2435
Proposed	0.06	0.2038
<b>Hand bag</b>		
Weinberg	0.04	0.0722
Kim	0.05	0.0874
Proposed	0.018	0.0359

approach. From the summary, it is clear that our proposed approach has the overall best performance. Our maximum absolute error was 0.06 m as compared to 0.1 m for both Weinberg and Kim's approaches and the maximum RMSE of 0.2038 m as compared to 0.2186 m for Weinberg's method and 0.2435 m for Kim's approach.

The estimated step lengths were used to calculate the

distance travelled by the participant. The cumulative sum of the step length gave the distance travelled by the participant after every step. To evaluate the performance of our approach, we compared the distances estimated by Weinberg's method, Kim's method and our proposed approach with the actual distance measured on the ground. Fig. 14 compares the distances estimated by Weinberg, Kim and the proposed approach for each smartphone placement. Our proposed approach shows the overall best performance while Kim's approach had the highest error for all smartphone placements. All the three algorithms considered had larger distance estimation errors for when the phone was in the backpack as seen in Fig. 14 (c) and when the phone was in the back pocket as seen in Fig. 14 (d). This is attributed to the accuracy of the step counting component where for both placements, the step error was one step. A step error of one step leads to a distance error equivalent to one step length.

Fig. 15 shows the distribution of distance estimation errors for the various smartphone placements considered in this study. using an error rate metric defined in eq. (37). For each step length estimation approach, the median, lower and upper quartiles and whiskers of the data are shown.

The median errors vary between 0.168 to 0.446 m for Weinberg's approach, 0.339 to 0.850 m for Kim's approach and 0.1 to 0.321 m for the proposed approach. Kim's approach has the widest dispersion and tends to over estimate the distance while our proposed approach has the overall best performance.

Table 3 shows the summary of absolute errors and RMSE for Weinberg's method, Kim's approach and our proposed approach. From the summary, it is clear that our proposed approach has the overall best performance with a maximum absolute error of 0.321 m as compared to 0.446 m for Weinberg approach and 0.85 m for Kim's approach. The maximum RMSE of the proposed approach is 0.4247 m as compared to 0.6611 m for Weinberg's method and 1.1360 m for Kim's approach.

#### D. HEADING ESTIMATION

Fig. 16 shows the estimated roll, pitch and yaw for the various smartphone placements. The mean absolute heading error was  $0.0378^\circ$ ,  $0.0244^\circ$ ,  $0.0233^\circ$ ,  $0.0239^\circ$ , and  $0.0316^\circ$ , handheld, handheld with an arm swing, backpack, back pocket and hand bag, respectively as summarised in Table 4.

#### E. POSITION ESTIMATION

The participants position was computed using the formula defined by eq. (34) and eq. (35). Fig. 17 shows the estimated positions of trajectories in the global frame for the various smartphone placements. The highest displacement errors were recorded for the backpack (1.4544) and back pocket (1.1672) smartphone placements as shown in Fig. 17(c) and 17(d). This is due to the errors of the step counting component where for both placements, the step error was one step leading to a distance error equivalent to one step length. The overall best performance was recorded when

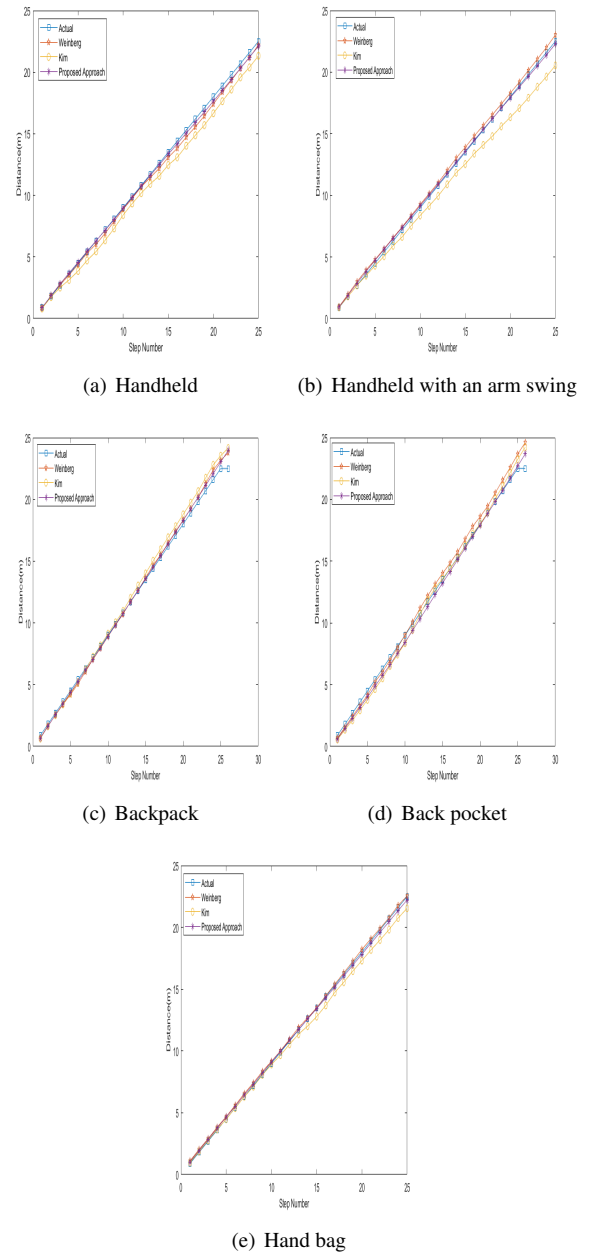
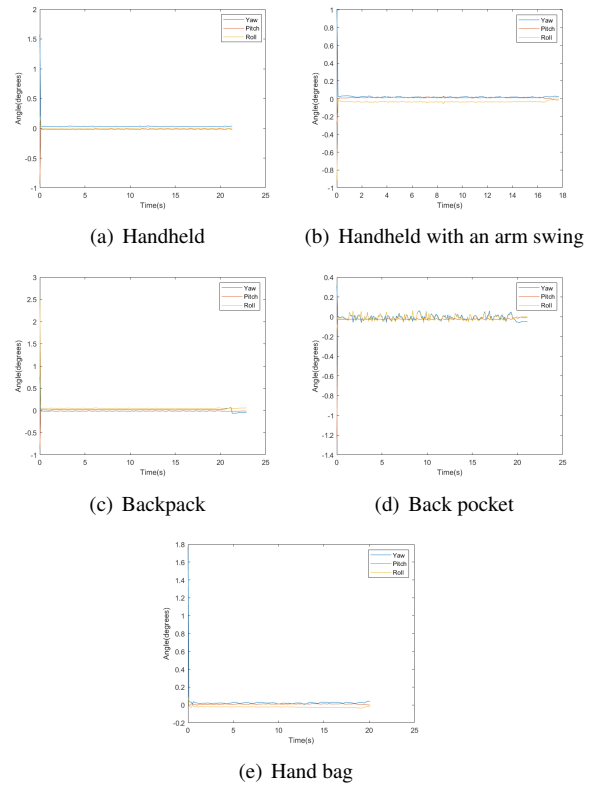
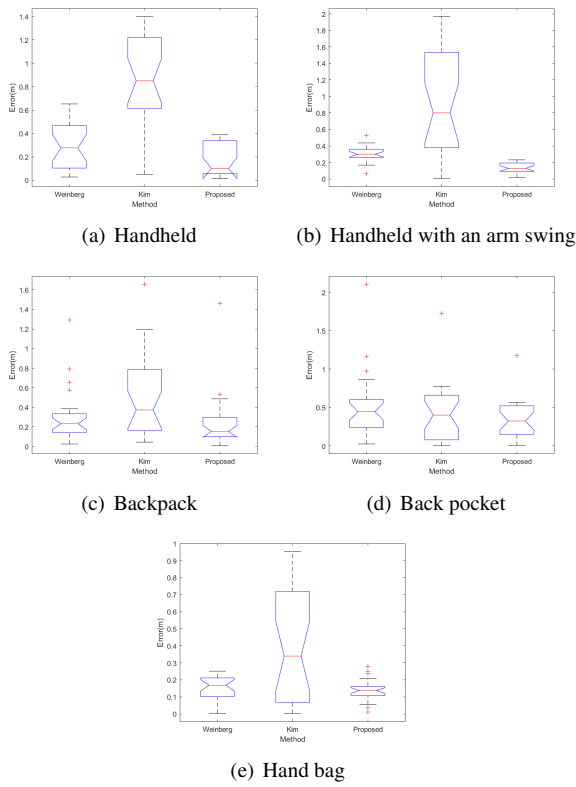


FIGURE 14. Distance estimation by Weinberg's method, Kim's method and our proposed approach for the various smartphone placements.

the phone was handheld with an arm swing. Table 5 shows the summary of the displacement errors and RMSE for the various smartphone placements.

#### VII. CONCLUSION AND FUTURE WORK

We developed a new fusion based approach for step detection and step length estimation using the Kalman filter. The Kalman filter was able to improve accuracy of conventional step detection and step length estimation algorithms. We tested our approach using five different smartphone placements. Our approach was able to achieve high step detection, step length estimation and positioning accuracy for all smart-



**FIGURE 15.** Absolute distance error for Weinberg's method, Kim's method and our proposed approach for the various smartphone placements.

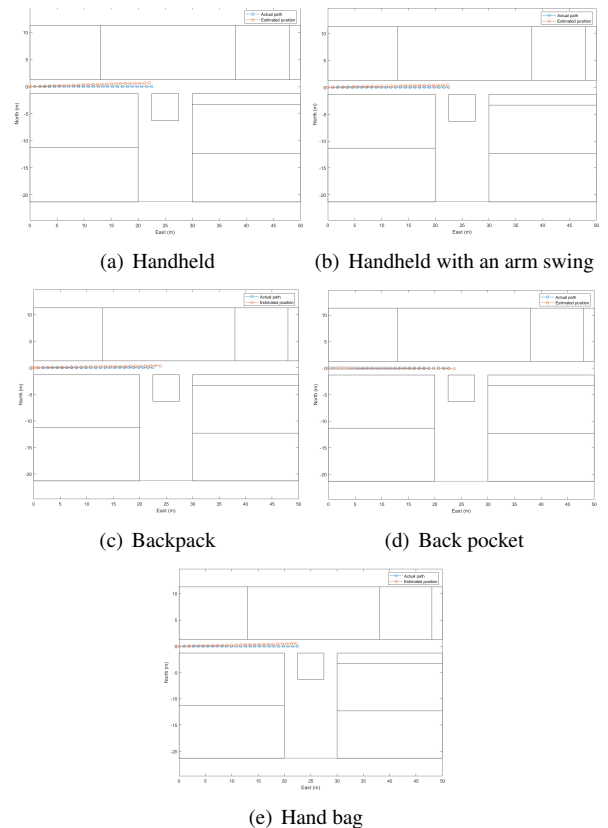
**FIGURE 16.** Estimated roll, pitch and yaw

**TABLE 3.** Absolute distance errors and RMSE for Weinberg's method, Kim's method and our proposed approach for the various smartphone placements.

Placement and Method	Absolute Error (m)	RMSE (m)
<b>Handheld</b>		
Weinberg	0.276	0.3656
Kim	0.85	0.9568
Proposed	0.1	0.2236
<b>Arm Swing</b>		
Weinberg	0.299	0.3215
Kim	0.8	1.1360
Proposed	0.127	0.1490
<b>Backpack</b>		
Weinberg	0.232	0.4023
Kim	0.373	0.6554
Proposed	0.154	0.3710
<b>Back pocket</b>		
Weinberg	0.446	0.6611
Kim	0.4	0.5675
Proposed	0.324	0.4247
<b>Hand bag</b>		
Weinberg	0.168	0.1686
Kim	0.339	0.5232
Proposed	0.137	0.1518

**TABLE 4.** Mean absolute errors by the heading estimation for the various smartphone placements.

Placement	Mean Absolute Error (degrees)
Handheld	0.0378
Handheld with an Arm Swing	0.0244
Backpack	0.0233
Back Pocket	0.0239
Hand bag	0.0316



**FIGURE 17.** Positions of trajectories in the global coordinate frame

**TABLE 5.** Displacement errors and RMSE for our position estimation approach for the various smartphone placements.

Placement	Displacement Error (m)	RMSE (m)
Handheld	0.3911	0.2252
Handheld with an Arm	0.2148	0.1460
Swing		
Backpack	1.4544	0.3623
Back Pocket	1.1672	0.4169
Hand bag	0.2844	0.1509

phone placements considered in this study.

For future work, we intend to experiment when the pedestrian walks at varying speeds and different paths such as rectangular and circular.

**REFERENCES**

[1] S. O’Dea, Number of smartphone users worldwide from 2016 to 2021, Statista. [Online]. Available: <https://www.statista.com/statistics/330695/number-of-smartphone-users-worldwide/> [accessed on 01 September 2020].

[2] Z. Yang, C. Wu, Z. Zhou, X. Zhang, X. Wang, and Y. Liu. "Mobility increases localizability: A survey on wireless indoor localization using inertial sensors." *ACM Computing Surveys (Csur)*, vol. 47, no. 3, pp. 1–34, April. 2015.

[3] P. G. R. B. J. H. David Hanes, Gonzalo Salgueiro. *IoT Fundamentals: Networking Technologies, protocols, and Use Cases for the Internet of Things*. Indianapolis, IN, USA: Cisco Press, 2017, pp. 75–135.

[4] A. Poullose, O. S. Eyobu, and D. S. Han. "An indoor position-estimation algorithm using smartphone imu sensor data." *IEEE Access*, vol. 7, pp. 11165–11177, January. 2019.

[5] G. Hussain, M. S. Jabbar, J.-D. Cho, and S. Bae. "Indoor positioning system: A new approach based on lstm and two stage activity classification." *Electronics*, vol. 8, no. 4, pp. 375, March. 2019.

[6] A. Brajdic and R. Harle, "Walk detection and step counting on unconstrained smartphones," in *Proc. UbiComp*, Zurich, Switzerland, pp. 225–234.

[7] P. Davidsson and R. Piche. A survey of selected indoor positioning methods for smartphones." *IEEE Commun. Surveys & Tuts.*, vol. 19, no. 2, pp. 1347–1370, December. 2016.

[8] X. Shi, X. Shao, Z. Guo, G. Wu, H. Zhang, and R. Shibusaki. "Pedestrian trajectory prediction in extremely crowded scenarios." *Sensors*, vol. 19, no. 5, pp. 1223, March. 2019.

[9] Y. S. Suh, E. Nemati, and M. Sarrafzadeh. "Kalman-filter-based walking distance estimation for a smart-watch," in *Proc. CHASE*, Washington, DC, USA, 2016, pages 150–156.

[10] N.-H. Ho, P. H. Truong, and G.-M. Jeong. "Step-detection and adaptive step-length estimation for pedestrian dead-reckoning at various walking speeds using a smartphone." *Sensors*, vol. 16, no. 9, pp. 1423, September. 2016.

[11] H. Weinberg. "Using the adxl202 in pedometer and personal navigation applications." *Analog Devices AN-602 application note*, vol. 2, no. 2, pp. 1–6, July. 2002.

[12] J. W. Kim, H. J. Jang, D.-H. Hwang, and C. Park. "A step, stride and heading determination for the pedestrian navigation system." *Journal of Global Positioning Systems*, vol. 3, no. 1-2, pp. 273–279, December. 2004.

[13] A. Wang, X. Ou, and B. Wang. "Improved step detection and step length estimation based on pedestrian dead reckoning," in *Proc. ISEMC*, Nanjing, China, 2019, pp. 1–4.

[14] C. Tjhai and K. O’Keefe. "Step-size estimation using fusion of multiple wearable inertial sensors," in *Proc. IPIN*, Sapporo, Japan, 2017, pp. 1–8.

[15] F. Li, C. Zhao, G. Ding, J. Gong, C. Liu, and F. Zhao. "A reliable and accurate indoor localization method using phone inertial sensors", in *Proc. UbiComp ’12*, Pittsburgh, USA, 2012, pp. 421–430.

[16] A. Poullose, O. S. Eyobu, and D. S. Han. "A combined pdr and wi-fi trilateration algorithm for indoor localization," in *Proc. ICAIIC*, Okinawa, Japan, 2019, pp. 072–077.

[17] X. Li, J. Wang, and C. Liu. "Heading estimation with real-time compensation based on Kalman filter algorithm for an indoor positioning system." *ISPRS International Journal of Geo-Information*, vol. 5, no. 6, pp. 98, June. 2016.

[18] Z. Deng, W. Si, Z. Qu, X. Liu, and Z. Na. "Heading estimation fusing inertial sensors and landmarks for indoor navigation using a smartphone in the pocket." *EURASIP Journal on Wireless Communications and Networking*, no. 1 (2017), pp. 160, September. 2017.

[19] IT-JIM, "Watch your steps: a brief review of step detection using mobile sensors", 2017.[Online]. Available: <https://www.it-jim.com/2017/10/24/watch-your-steps> [accessed on 16 December 2019].

[20] S. Sen, J. Lee, K.-H. Kim, and P. Congdon. "Avoiding multipath to revive in building wifi localization," in *Proc. MobiSys ’13*, Taipei, Taiwan, 2013, pp. 249–262.

[21] J. Seo, Y. Chiang, T. H. Laine, and A. M. Khan. "Step counting on smartphones using advanced zero-crossing and linear regression," in *Proc. IMCOM ’15*, Bali Indonesia, 2015, pp. 1–7.

[22] Alex Becker. Kalman Filter, 2019. [Online]. Available: <https://www.kalmanfilter.net/default.aspx> [accessed on 20 November 2019].

[23] B. Alsadik. *Adjustment Models in 3D Geomatics and Computational Geophysics: With MATLAB Examples*. Elsevier Science, 2019, pp. 299–326.

[24] G. Welch and G. Bishop. *An introduction to the Kalman filter*. 1995.



ADONG PRISCILLA received her Bsc. in Computer Science from Makerere University, Uganda in 2014. She is currently pursuing her Msc. in Data Communications and Software Engineering at Makerere University. Her research interests are in indoor localization, Internet of Things, Machine learning and wireless communications.



ODONGO STEVEN EYOBU received his Bsc. in Computer Science from Islamic University, Uganda in 2004. He obtained the M.S. degree in Data Communication and Software Engineering from Makerere University, Uganda in 2007 and a Ph.D. in Electronics Engineering from Kyungpook National University, Korea in 2018. His research interests include deep learning systems, indoor Localization, vehicular communications, intelligent transportation systems, and wireless sensors.

He is currently a lecturer at the School of Computing and Informatics Technology at Makerere University, Uganda.



TONNY J.OYANA is a Professor and the Principal, College of Computing and Information Sciences (CoCIS), Makerere University. He holds a Ph.D in Geography with a concentration in Geographical Information Systems (GIS) from the State University of New York (SUNY) Buffalo. He has previously served as Director of the Undergraduate Programs, Department of Geography and Environmental Resources, Southern Illinois University (SIU) and as the Director of Informatics for the Institute for Research, Innovation, Synergy and Health Equity (iRISE) Community Collaboration Core, University of Tennessee Health Science Center, Memphis both in the U.S.

His research interests are GIS and GIScience; visualization and cartography; spatial epidemiology; applying GIS techniques to the subfields of disease surveillance and monitoring, bioterrorism, and environmental health and exposure; spatial epidemiology; multivariate statistics; spatial statistics.



DONG SEOG HAN received the B.S. degree in electronic engineering from Kyungpook National University (KNU), Daegu, South Korea, in 1987, and the M.S. and Ph.D. degrees in electrical engineering from the Korea Advanced Institute of Science and Technology, Daejeon, South Korea, in 1989 and 1993, respectively. From 1987 to 1996, he was with Samsung Electronics Co., Ltd., where he developed the transmission systems for QAM HDTV and Grand Alliance HDTV receivers.

Since 1996, he has been with the School of Electronics Engineering, KNU, as a Professor. He was a courtesy Associate Professor with the Department of Electrical and Computer Engineering, University of Florida, in 2004. He was the Director with the Center of Digital TV and Broadcasting, Institute for Information Technology Advancement, from 2006 to 2008. His main research interests include intelligent signal processing and autonomous vehicles.

• • •