Smartphone-based Pedestrian Dead Reckoning and Orientation as an Indoor Positioning System

Tinh Do-Xuan *, Vinh Tran-Quang *, Tuy Bui-Xuan *, Vinh Vu-Thanh †
*School of Electronics and Telecommunications, Hanoi University of Science and Technology, Hanoi, Vietnam
†University of Information and Communication Technology, Thai Nguyen University, Thai Nguyen, Vietnam
Email: doxuantinh.fet@gmail.com, vinhtq@hust.edu.vn

***Abstract*—Indoor positioning system (IPS) is becoming more and more important and necessary in our daily life. Based on the IPS, we may develop a variety of location based services (LBS) such as navigation and orientation in large commercial centers, tracking and monitoring objects, or locating the elderly and children. With the development of new advanced technologies, the hand-held devices such as smart-phones and tablets have been widely used. These devices are all equipped with common sensors such as accelerometer, magnetic field, gyro and sound sensors. Therefore, the development of sensors-based positioning system is becoming practical and easy to implement while GPS or Cell-ID works inaccurately in the indoor environment. In this paper, we propose a solution and develop an application to position user's location in an indoor environment by pedestrian dead reckoning (PDR) and orientation. The proposed system just uses some available sensors of smart-phones: accelerometer sensor, gyro sensor and gravity sensor for PDR which detects steps as well as to measure the length and the orientation of his/her steps and then computes his/her location.**

***Keywords*—sensor-based indoor positioning system; pedestrian dead reckoning; smart-phone; Quaternion.**

I. INTRODUCTION

Positioning is a technique that used to get object's position in a frame of reference. Commonly, positioning can be done using global positioning system (GPS) or Cell-ID. But these techniques require a large infrastructure, such as GPS satellites or base transceiver station (BTS) cell-phone service provider. Moreover, the above systems work inefficiently in the indoor environment. The GPS signal strength is so weak and will be affected by many obstacles in the indoor environment that the positioning is greatly affected. Positioning with BTS cell-phone allows locating the user around the BTS within the radius from 100 m up to 35 km [1] depending on the cell radius. A better way to locate an indoor user is to use Wi-Fi. However, this method needs many access points (APs) as reference points for calculating the user's location. Concurrently, APs must be aligned with the available fixed location. In addition, it is difficult to identify the user's orientation of immediate motion. As we mentioned above, positioning using Pedestrian Dead Reckoning (PDR) approach and orientation is a simple and effective way to locate user's location. This technique can be used at anywhere and at any time with just a smart phone or a tablet instead of requiring a large infrastructure. Whenever a user steps, the step will be automatically detected by the system, the step length and orientation then computed to provide the user's position. In order to ensure the smart-phone-person interaction, we only take into account of the case that the users hold their cell phone and the yaw angle of phone surface (the plane xOy in the body's coordinate as plotted in Fig. 1) ranging from 0 to 90 degree.

The remainder of this paper is organized as follows. Section II demonstrates how to using accelerometer sensor and gravity sensor as well as PDR algorithms to calculate the acceleration of user movement. Next, in Section III we also introduce gyro sensor and orientation theory to estimate rotation angle and then experiment will be implementing to test the result. The Section IV proposes an Indoor Position System that uses the proposed algorithms to determine the user's position. In the final section, we present the conclusion and the future work.

II. PEDESTRIAN DEAD RECKONING

A. Step Detection

1) Accelerometer Sensor in Smart-phone: At present, most of smart-phones are equipped with accelerometer sensors. These sensors allow determining the acceleration of receivers, gravitational acceleration. In Android OS, accelerometer sensors are declared as TYPE_ACCELEROMETER TYPE_GRAVITY, and TYPE_LINEAR_ACCELERATION [2]. The acceleration values are determined in the body's coordinates shown in Fig. 1. The LINEAR_ACCELERATION sensor allows identifying acceleration value relative to the earth but excluding gravitational acceleration value. So we will use output values of this sensor for PRD approach instead of ACCELEROMETER sensor. However, it is just the acceleration value relative to the body's coordinate, with different yaw, angle; the acceleration value is different at each correlative axis in the same movement. Therefore, it is necessary to build a formula for determining the acceleration value relative to the fixed system coordinate or the earth's coordinate.

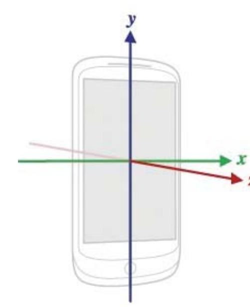

Fig. 1. The body's coordinate system.

2) Calculation of acceleration motion: When stepping, a person makes two motions including go-ahead (the force parallel to the ground) and the person pushed by feet (the force perpendicular to the ground) as plotted in Fig. 2 . So we will calculate the acceleration value in these two ways. Due to the postures and the way of holding cell phone, we only consider the acceleration value in the z'Oy' plane of the body's coordinate system. The zOy plane has Oz perpendicular to the ground, and Oy is tangential axis to the ground. It is assumed that at time t, the acceleration vector of the device is $\overrightarrow{OP}$, with $\overrightarrow{OP} = \overrightarrow{OZp'} + \overrightarrow{OYp}$, where Zp' and Yp' are acceleration values measured by the LINEAR_ACCELERATION sensor. Let θ is the angle of two planes (z'Oy') and (zOy) as plotted in Fig. 3. We have,

$$\begin{aligned} \mathrm{Zp} &= \sin(\theta) * \mathrm{Yp}' + \cos(\theta) * \mathrm{Zp}' \\ \mathrm{Yp} &= \cos(\theta) * \mathrm{Yp}' - \sin(\theta) * \mathrm{Zp}'. \end{aligned} \tag{1}$$

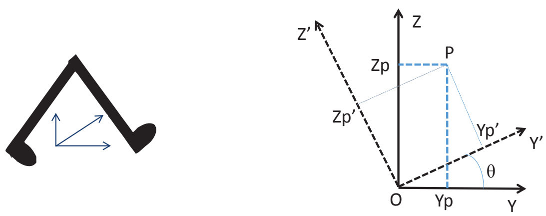

Fig. 2. The footstep analysis and acceleration in two system coordinates.

Figure 3 shows the gravity in each directions of device with θ that is the angle between z'Oy' plane of body's coordinate system and zOy plane as explained above. To determine the θ, we use GRAVITY sensor with the following equation:

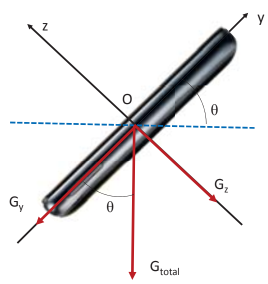

Fig. 3. The gravity in each directions of device.

$$\begin{aligned} \theta &= \arccos\left(\frac{G_y}{G_{Total}}\right) - \frac{\pi}{2} \\ G_{Total} &= \sqrt{G_x^2 + G_y^2 + G_z^2} \end{aligned} \tag{2}$$

The acceleration values measured after taking 40 steps at each direction (Oy and Oz) which were determined through the equation (1) are plotted in Fig. 4 and Fig. 5. As can be seen from the graphs and similar experiments, we find that the acceleration value in the direction of Oz (perpendicular to the ground) has more stable values than that in the direction of Oy. Therefore, we use the acceleration value in the direction of Oz to evaluate steps.

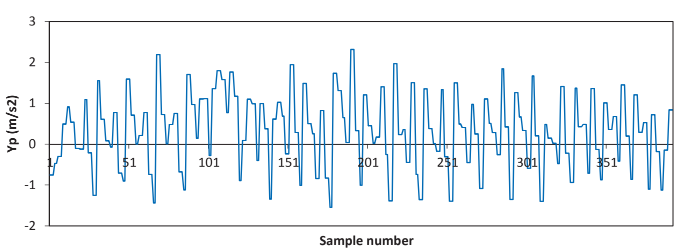

Fig. 4. The acceleration in the direction of Oy.

3) Footstep detection algorithm: There are many step detection techniques such as: the local variance method, zero crossing method, finite-state machine method. The local variance method [3] is based on filtering magnitude of acceleration followed by applying a threshold on the variance of

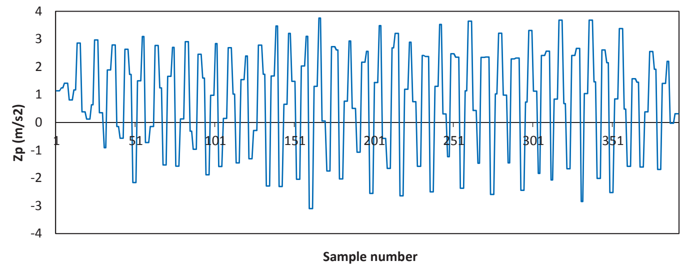

Fig. 5. The acceleration in the direction of Oz.

acceleration over a sliding window. The zero crossing method is discussed in [4]. This method computes the magnitude of the acceleration signal. The step boundaries are defined by the positive-going zero crossing of a low-pass filtered version of the magnitude of acceleration signal. However, both the local variance method and the zero crossing method have been applied to foot-mounted sensors.
In this paper, we just present a step detection techniques: Finite-state machine (FSM) that was discussed in [5]. In [5] the average error in estimating the number of steps using FSM is 2.66% in case handling smart-phone compare with the 27.18% of average error using Local Variance method [3] and 30.19% of average error using Zero Crossing method [4]. So, the FSM method is the best method for detecting step in the three methods.

From calculated acceleration values, we will use a combination of FSM and threshold levels defined by statistical process. Each step is relative to 4 states: State 0, State 1, State 2, and State 3.

- State 0: starting step
- State 1: the user heels up, and move a foot towards, take a half of step. At this time, the acceleration reaches the maximum.
- State 2: The acceleration will decrease to State 2 when the user is about to heel up the other foot.
- State 3: Complete one step. After the acceleration reduces to the State 3, the user will return to the State 0 - starting state for the next step.

Finite-state machine diagram: Figure 6 shows the FSM diagram which includes four states are state 0, state 1, state 2, state 3 as explained above, input as the acceleration value in the direction of Oz determined through the equation (1), up threshold and low threshold. The advantages of this FSM approach is: It does not require any pre-processing or filtering operations on the accelerations signal. Therefore, it is suitable for implementation on the resource-limited mobile phones.
Determination of up threshold: to identify the up threshold, we implement two experiments.

- 988 step samples are obtained, the maximum and minimum value of acceleration were recorded at each step. The maximum values of acceleration are shown in the Fig. 7.
- 2116 samples of acceleration values when the device fluctuates mildly at random are shown in the Fig. 8

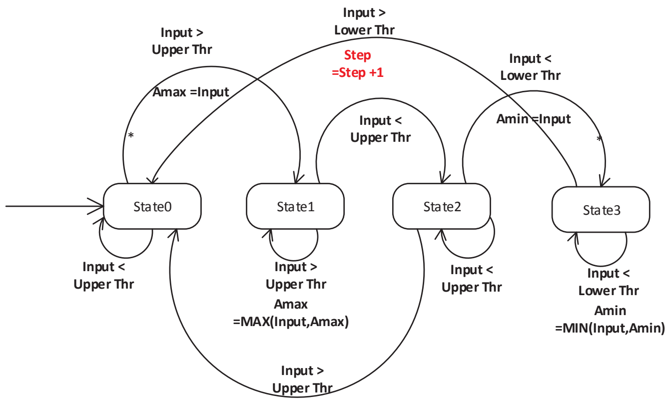

Fig. 6. The FSM diagram for step detection.

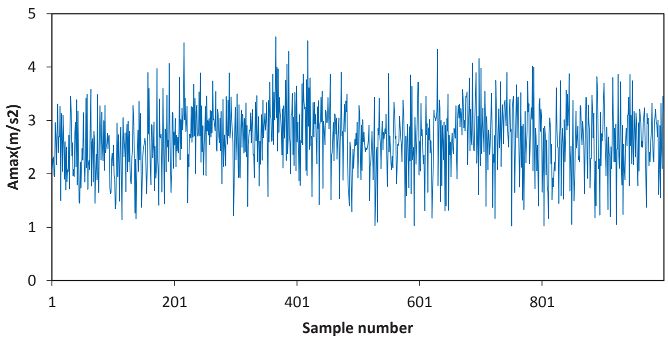

Fig. 7. The maximum value of acceleration during walking with smart-phone in hand.

To evaluate threshold levels, we use receiver operating characteristic (ROC) curve for analysis [6]. ROC graphs are two-dimensional graphs in which true positive rate (TPR) or sensitivity(%) is plotted on the Y axis and false positive rate (FPR) or 100-specificity(%) is plotted on the X axis. TPR and FPR are discussed in [6]. To facilitate analysis, we use software MedCalc to draw curves [7]. As can be seen from the Fig. 9 and the Table I that show threshold values, sensitivity, specificity value, the value of up threshold is selected = 1.278. With this value, the true positive rate of the classifier is 98.1% and false positive rate is 1.7%, which are evaluated best by ROC Curve.

Determination of Low Threshold The minimum values of acceleration for the 988 steps are shown in the Fig. 10. By the same way, the value of low threshold is set to -0.0689. With this value, the true positive rate of the classifier is 97.5% and false positive rate is 1.7%, which are evaluated best by ROC Curve.

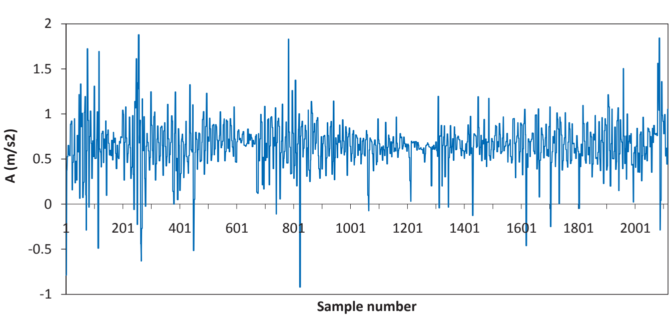

Fig. 8. The acceleration when the device fluctuates mildly at random.

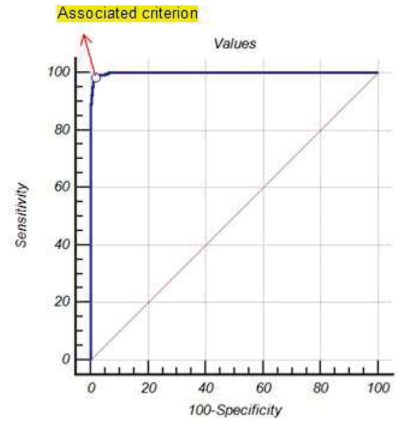

Fig. 9. The ROC Curve of up threshold.

TABLE I. THE CRITERION VALUES AND COORDINATES OF THE ROC CURVE FOR UP THRESHOLD.

Criterion	Sensitivity	Specificity
...	...	...
>1.259	98.20	98.16
>1.263	98.10	98.16
>1.278	**98.10**	**98.30**
>1.307	97.80	98.30
...	...	...

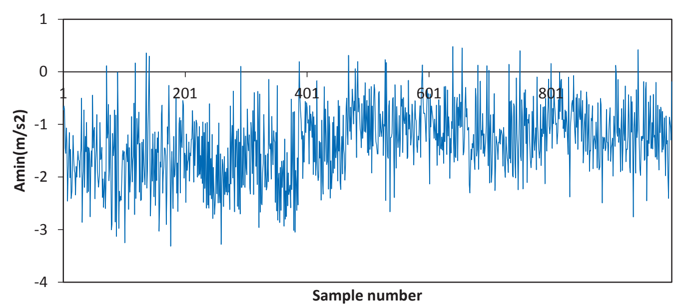

Fig. 10. The minimum value of acceleration during walking with smart-phone in hand.

B. Step length Estimation

We can approximate the distance traveled by estimating the step length. Normally, there are two methods for estimating the step length: static method and dynamic method. The static method allows determining the step length by the height of the user. However, this method is not accurate when the user moves quickly or slowly. Therefore, we use the dynamic method that the step length depends on the motion acceleration:

$$L = f(A) \tag{3}$$

where A is acceleration value determined through the equation (1). In this subsection, We present three dynamic methods to estimate the step length:

- **Weinberg Method:** This method determines a step length by the difference between the maximum acceleration and minimum acceleration of each step [8].

$$Step\_size = k\sqrt[4]{A_{\max} - A_{\min}} \tag{4}$$

- **Scarlet method:** This method determines the step length by a correlation between the value of maximum, minimum, and average acceleration of the step length [9].

$$Step\_size = k\frac{\frac{\sum_{k=1}^{N}|A_k|}{N} - A_{\min}}{A_{\max} - A_{\min}} \tag{5}$$

- **Kim method:** This method provides the determination of a correlation between step length and average acceleration values [10].

$$Step\_size = k\sqrt[3]{\frac{\sum_{k=1}^{N}|A_k|}{N}} \tag{6}$$

Experiment

In order to identify constant value for each method, we carry out 10 experiments with different ways of carrying a smartphone and different walking rate to get the constant value k for the each method. They are shown in Table II. After determining the constant value for the each method as shown in Table II, we just used these values to estimate the distance traveled for the each method in 10 times checking test as shown in Table III.

TABLE II. THE CONSTANT VALUES FOR EACH METHOD IN 10 EXPERIMENTS.

K-Weinberg Average	K-Scarlet Average	K-Kim Average
0.497168502	1.0180004	0.603300921

TABLE III. DISPLACEMENT ESTIMATION ERROR

Method	Displacement Estimation Error	
	Average(m)	Std.Deviation (%)
Weinberg	0.636	1.99
Scarlet	0.729	2.29
Kim	0.723	2.266

C. Evaluation of deviation and selection of the best method

Table IV show the percentage of error is 1.946% in step detection when applying FSM with two threshold that were evaluated by ROC Curve. As can be seen in Table III the Weinberg algorithm shows the result of distance estimation with the average error 1.99% which is the lowest error. Therefore, we select the finite-state machine diagram and the two threshold levels: up threshold = 1.278 and low threshold = -0.0689 to detect steps, and we use Weinberg algorithm to estimate the step length.

III. ESTIMATE ROTATION ANGLE

There are many ways to calculate movement orientation of the device such as using magnetic field sensors and gravitational sensor to estimate the orientation of device relative to three coordinate axes placed on earth [2]. Nevertheless, due to unstable gravity and magnetic field at different points, there is a big tolerance when determining the orientation on the earth plane (horizontal plane). Another way to estimate the orientation of device is using a gyro sensor. This sensor will measure the angular momentum in 3 body's axis and these parameters are used to predict the movement orientation with low tolerance.

TABLE IV. ERROR PERCENTAGE OF STEP DETECTION 10 TIMES CHECKING TEST.

Test No	Step		
	Real step	Detected step	Deviation (%)
1	46	46	0
2	46	45	2.18
3	47	46	2.13
4	47	45	4.26
5	47	46	2.13
6	46	47	2.18
7	47	47	0
8	47	46	2.13
9	48	48	0
10	45	43	4.45
	Average error		1.946

A. Gyroscope Sensor Overview

A gyroscope is a device used primarily for navigation and measurement of angular velocity expressed in Cartesian coordinates.

$$\omega = (\omega_x, \omega_y, \omega_z) \tag{7}$$

With smart devices like smart-phones, the values provided by the sensor are angular momentum (rad/s) of the device in the body's coordinate system as shown in Fig. 1. There are three parameters provided by the Gyroscope sensor including: pitch, roll, and yaw. The spinning angle corresponding to each axis during the $t_0 - t$ time is as equation:

$$w_p(t) = \int_{t_0}^{t} \omega_p(\tau)d\tau + w_{p_0} \tag{8}$$

Where p is direction (x, y or z). We will determine w_x, w_y and w_z. The approaches to map the rotation of device for the earth axis will be introduced below.

B. 3D Rotation

As we know, Euler angles, Rotation matrices and Quaternions are rotational modalities that can be used to describe orientation of objects in 3D. As discussed in [11], the quaternion has many advantages that are disadvantages of Euler angles and Rotation matrices such as: coordinate system independence, simple interpolation methods, simple composition and especially no gimbal lock. So, we use quaternions for representation of rotation.

Quaternion Theorem

A quaternion q is defined as the sum of a scalar q_0 and a vector $q = (q_1, q_2, q_3)$ According to Quaternion Theorem [12] we have:

- For any unit quaternion:

$$q = q_0 + q = \cos\frac{\theta}{2} + u\sin\frac{\theta}{2} \tag{9}$$

and for any vector $v \in \mathrm{R}^3$ the action of the operator

$$\begin{aligned} Lq(v) &= q.v.q^* \\ &= (q_0^2 - \overrightarrow{q}.\overrightarrow{q}).v + 2q_0.\overrightarrow{q} \times v + 2\overrightarrow{q}(\overrightarrow{q}.v) \end{aligned} \quad (10)$$

where **v** is equivalent to a rotation of the vector through an angle θ about **u** as the axis of rotation.

- Quaternion Operator Sequences: Let p and q be two unit quaternions. We first apply the operator L_p to the vector **u** and obtain the vector **v**. To **v** we then apply the operator L_p and obtain the vector **u**.

$$w = L_q(v) = q.v.q^* = L_{qp}(u) \quad (11)$$

Quaternion algebra for representation of rotation

Assume the vector of angular speed during the time from t to $t + \Delta t_1$ is $\overrightarrow{\omega}_1$, with $\overrightarrow{\omega}_1 = \overrightarrow{\omega}_{1x} + \overrightarrow{\omega}_{1y} + \overrightarrow{\omega}_{1z}$. With very small Δt, we may regard the velocity of device at each axis as constant motion as a result, the rotation vector of motion is $\overrightarrow{w}_1 = \overrightarrow{\omega}_1.\Delta t_1$. The direction of $\overrightarrow{w}_1$ corresponds to the rotation axis, $\overrightarrow{w}_1$ is a rotation angle. According to the equation (9), we get Quaternion correlative to the above rotation to be:

$$q = \cos\frac{|\overrightarrow{w_1}|}{2} + \frac{\overrightarrow{w_1}}{|\overrightarrow{w_1}|}\sin\frac{|\overrightarrow{w_1}|}{2} = q_0 + \overrightarrow{q}. \quad (12)$$

Supposing that at time t, we have $\overrightarrow{OA}$ in the Descartes with the coordinate (x_A, y_A, z_A). According to the equation (10), after the period $t + \Delta t$ we have

$$\begin{aligned} \overrightarrow{OA_1} &= q.\overrightarrow{OA}.q* \\ &= (q_0^2 - \overrightarrow{q}.\overrightarrow{q}).\overrightarrow{OA} + 2q_0.[\overrightarrow{q}, \overrightarrow{OA}] + 2.\overrightarrow{q}(\overrightarrow{q}.\overrightarrow{OA}) \\ &= (x_{A_1}, y_{A_1}, z_{A_1}). \end{aligned} \quad (13)$$

Let α is the yaw angle of the two vectors $\overrightarrow{OA}$ and $\overrightarrow{OA_1}$ in the plane (xOy), we have

$$\cos\alpha = \frac{x_A * x_{A_1} + y_A * y_{A_1}}{\sqrt{x_A^2 + y_A^2} * \sqrt{x_{A_1}^2 + y_{A_1}^2}} \quad (14)$$

Similarly, assume the vector of angular speed during the time from $t + \Delta t_1$ to $t + \Delta t_1 + \Delta t_2$ is $\overrightarrow{\omega_2}$. The rotation vector of motion is $\overrightarrow{w_2} = \overrightarrow{\omega_2}.\Delta t_2$. We get Quaternion correlative to the above rotation to be:

$$p = \cos\frac{|\overrightarrow{w_2}|}{2} + \frac{\overrightarrow{w_2}}{|\overrightarrow{w_2}|}\sin\frac{|\overrightarrow{w_2}|}{2} = p_0 + \overrightarrow{p}. \quad (15)$$

At the time $t + \Delta t_1 + \Delta t_2$ the $\overrightarrow{OA}$ will be $\overrightarrow{OA_2}$. According the equation (11) we have:

$$\overrightarrow{OA_2} = (q.p).\overrightarrow{OA}.(q.p)^* \quad (16)$$

C. Experiment

To evaluate the accuracy of Quaternion algebra that uses for representation rotation, we implemented 180 experiments with the angle 5^o, 10^o, 15^o, 20^o...80^o, 85^o. The real angle and computed angle of all experiments is shown in the Fig. 13. The result representing the average deviation of measurements is 3.25^o.

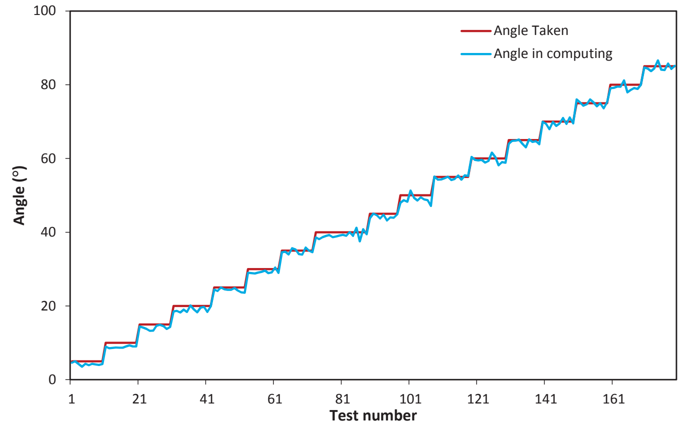

Fig. 11. The angle taken and angle in computing.

IV. Indoor Positioning System

A. System diagram

The user's position is identified by the motion orientation and distance compared with the benchmark. The two figures are identified by step length and angle estimation as mentioned in the Section (II) and Section (III). The diagram of the system is shown in the Fig. 12. Right after a step is detected, the system will estimate the step length and compute orientation to get current position. A Indoor positioning algorithm which is explained below will execute to get the position of the user who hold the smart-phone.

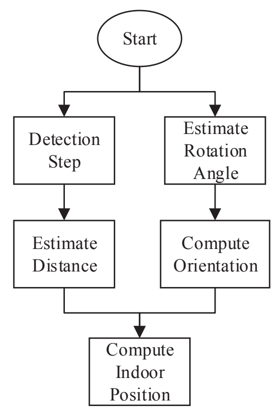

Fig. 12. Flow chart of overall the positioning system.

Indoor positioning algorithm:

Assumed that the user take n steps from the position $S_0(x_o, y_0)$, to the position $S_n(x_n, y_n)$, each step is correlative to the vectors $\overrightarrow{S_1}$, $\overrightarrow{S_2}$, $\overrightarrow{S_3}$... $\overrightarrow{S_n}$. Assume that the angles measured by the device at the starting time and step detection times are $\gamma_0, \gamma_1, \gamma_2, ..., \gamma_{n-1}, \gamma_n$. Let α_i is a yaw angle of the vector $\overrightarrow{s_i}$ and $\overrightarrow{s_0}$. We have $\alpha_i = \gamma_i - \gamma_0$ with i=1, 2, 3, 4, ..., n-1, n The length of vector $\overrightarrow{s_i}$ is the length of step i estimated by the system. The user's position at the time of step i detection is:

$S_i = (x_i; y_i)$

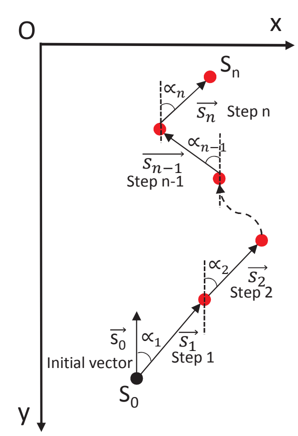

Fig. 13. Descriptions motion according to vectors.

where $x_i = x_{i-1} + |\vec{s}_i| * \sin\alpha_i$ and $y_i = y_{i-1} - |\vec{s}_i| * \cos\alpha_i$ with $i = 1, 2, 3, ..., n-1, n$

B. Experimental Result

We carry out a test for the system at Room 618-Ta Quang Buu library-Hanoi University of Science and Technology. The dimension of room is 19m x 32m. The result is plotted in the Fig. 14. In Fig. 14, the blue line is actual pathway of the user and device. 3 green lines with red dots are the position of steps measured by the application. Deviation distance is calculated for each test as shown in Fig. 15. The maximum tolerance of these tests is 1.89m. Results measurement error caused by internal error of device and the motion of person no corresponding to motion of device so that during the moving process will cause error accumulation.

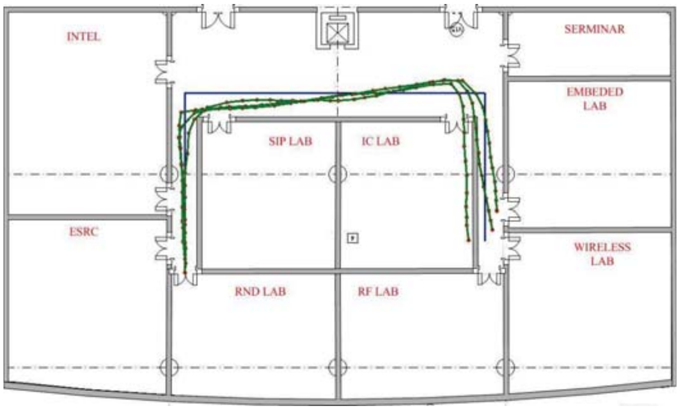

Fig. 14. Experimental results

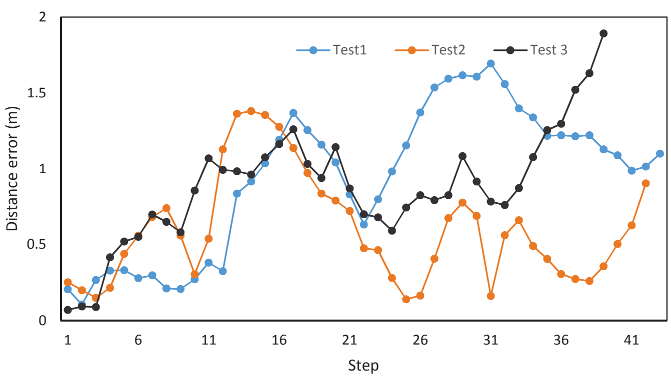

Fig. 15. Deviation distance for each test.

To dissipate this error, we build a interested point in a fixed position and use the Quick Response Code (QR Code) to read the actual coordinates. A QR Code contains three parameters: x, y, β; where x, y are coordinates of the QR Code's position, β is yaw angle between the direction of smart-phone when it scans the QR Code and the original direction. After the smart-phone have completed scanning a QR Code, user's location user's orientation will be replaced by x, y and β.
Using QR code to read the actual coordinate has reduced error distance down to 1 meter. As you can see, we set 4 interested points as plotted in Fig. 16. The first interested point is initial point. The result are listed following the Table V.

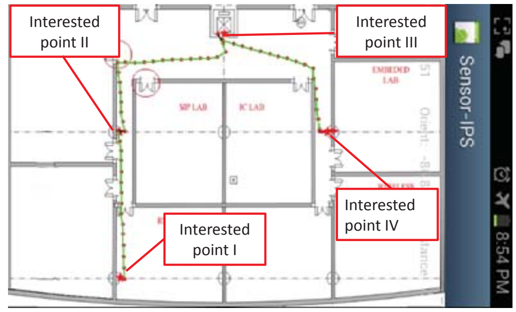

Fig. 16. Using QR code to read the actual coordinates

TABLE V. TABLE SHOWS THE DISTANCE ERROR AFTER USING THE INTERESTED POINTS.

Interested point	Real position (x,y)	Computed position (x,y)	Distance error (m)
II	(8, 23)	(7.902, 23.427)	0.438
III	(2.3, 16)	(2.047, 16.761)	0.82
IV	(8, 9)	(8.057, 8.513)	0.490

V. CONCLUSION

This paper presents a positioning system that not only can be used in indoor environment but also outdoor environment with smart-phone. The system focused on displacement estimation by utilizing the accelerometer sensor and gyroscope sensor built-in a smart-phone which is placed in the hand. Step detection on walking and holding the smart phone process results an average error of 1.946% and average error of rotation angle is 3.52%. Using QR code to read the actual coordinate has reduced error distance from 1.89 m down to 1 m.

REFERENCES

[1] N. Deblauwe, "Gsm-based positioning: Techniques and applications," in *Proc. IEEE Globecom '99*, Ravensteingalerij 28,B-1000 Brussels, 2008, p. 18.

[2] http://developer.android.com/reference/android/hardware/Sensor.html.

[3] A. Jimenez, F. Seco, C. Prieto, and J. Guevara, "A comparison of pedestrian dead-reckoning algorithms using a low-cost mems imu," in *Intelligent Signal Processing, 2009. WISP 2009. IEEE International Symposium on*, Aug 2009, pp. 37–42.

[4] H. H. StÂť'ephane Beauregard, "Pedestrian dead reckoning: A basis for personal positioning," in *Proceeding of the 3rd Workshop on positioning, navigation and communication*, pp. 27–35.

[5] M. Alzantot and M. Youssef, "Uptime: Ubiquitous pedestrian tracking using mobile phones," in *Wireless Communications and Networking Conference (WCNC), 2012 IEEE*, April 2012, pp. 3204–3209.

[6] T. Fawcett, "An introduction to roc analysis," *Pattern Recognition Letters.*, vol. 2, pp. 24–33, December 2005.

[7] http://www.medcalc.org/index.php.

[8] H. Weinberg, "Using the adxl202 in pedometer and personal navigation application," *Analog Devices AN-900 Application Note.*, 2002.

[9] J. Scarlett, "Enhancing the performance of pedometers using a single accelerometer," *Analog Devices AN-900 Application Note.*, 2005.

[10] J. W. Kim, H. J. Jang, and D.-H. Hwang, "A step, stride and heading determination for the pedestrian navigation system," *Global Positioning Systems.*, vol. 3, pp. 273–279, 2004.

[11] M. L. Erik B.Dam, Martin Koch, "Quaternions, interpolation and animation," in *Technical Report DIKU-TR-98/5,Department of Computer Science at the University of Copenhagen*, 1998.

[12] Y.-B. Jia, "Quaternions and rotations," vol. 1, Sep. 2013.