January 2015

Using topography to aid cell phone geolocation

Jaeyoung Kim
Purdue University

Graduate School Form
Updated 1/15/20

PURDUE UNIVERSITY
GRADUATE SCHOOL
Thesis/Dissertation Acceptance

This is to certify that the thesis/dissertation prepared

By Jaeyoung Kim

Entitled
USING TOPOGRAPHY TO AID CELL PHONE GEOLOCATION

For the degree of Master of Science in Mechanical Engineering

Is approved by the final examining committee:

Kartik B. Ariyur
Chair

Galen B. King

Raymond A. Decarlo

To the best of my knowledge and as understood by the student in the Thesis/Dissertation Agreement, Publication Delay, and Certification Disclaimer (Graduate School Form 32), this thesis/dissertation adheres to the provisions of Purdue University's "Policy of Integrity in Research" and the use of copyright material.

Approved by Major Professor(s): Kartik. B. Ariyur

Approved by: Jay P. Gore 12/3/2015
Head of the Departmental Graduate Program Date

USING TOPOGRAPHY TO AID CELL PHONE GEOLOCATION

A Thesis

Submitted to the Faculty

of

Purdue University

by

Jaeyoung Kim

In Partial Fulfillment of the

Requirements for the Degree

of

Master of Science in Mechanical Engineering

December 2015

Purdue University

West Lafayette, Indiana

To my parents, Hyunsup Kim and Jinbum Shin

ACKNOWLEDGMENTS

I appreciate my advisor, Professor Kartik B. Ariyur, for helping and sincerely discussing whenever I faced problems. I also appreciate my committees, Professor Galen B. King and Professor Raymond A. DeCarlo. In addition to the faculty members, I would like to thank from lab mates and especially thank to Yan Cui for his suggestions and helping me find good solutions. Lastly, I always thank my parents for their support and encouragement that helped me to motivate and focus.

TABLE OF CONTENTS

Page

LIST OF TABLES

LIST OF FIGURES

Figure Page

ABBREVIATIONS

GPS	Global Positioning System
RSS	Recieved Signal Strength
IMU	Inertial Measurement Unit
RFID	Radio Frequency Identification
TDOA	Time Difference of Arrival
FFT	Fast Fourier Transform
KF	Kalman Filter
IA	Interval Analysis
RSSI	Received Signal Strength Indication
RMSE	Root Mean Square Error

ABSTRACT

Kim, Jaeyoung. M.S.M.E, Purdue University, December 2015. Using Topography to Aid Cell Phone Geolocation. Major Professor: Kartik Ariyur, School of Mechanical Engineering.

In daily life, people demand accuracy of the Global Positioning System (GPS) receiver. The current problem of GPS on mobile phones is that it is not available in areas such as urban, natural canyons, forests, and indoor environments. Several methods have been developed to obtain more accurate position estimation over the past years. The received signal strength (RSS) and time difference of arrival (TDOA) are the main approaches to use available mobile signals and errors around $4 \sim 12$ dB and $10 \sim 60$ meters, respectively. Another approach to make a better performance of the sensor is to use radio frequency identification (RFID) with indoor Wi-Fi. A new method from our group shows that using magnetic field intensity maps based on interval analysis can perform better than the RSS, TDOA and RFID and reduce error for geolocation in some areas where GPS is not accessible. In our study, we develop a novel algorithm where sensor measurements on the cell phone are used to construct the topographic maps and aid cell phone geolocation which focuses on the angles of inclination in user's pathway when GPS is spotty. This can be particularly useful on uneven terrain outdoors. For sensor characterization, we use application in android operating system of smartphone by name of sensor stream IMU+GPS. The sensor stream allows for users to observe, select or record the current values of various measurements such as accelerations, angular rates (gyroscope), magnetic fields, GPS position and received signal strength indication (RSSI) in 3-dimensional coordinate system. We firstly develop algorithms of fast fourier transform and low pass filter to find the accurate vertical acceleration measurements which impact values are corresponded to step occurrences. Before analyzing position estimation, we use

the relationship between the stride length and stride interval and the methodology of detecting peak values to find the user's step. In order to reduce uncertainty and find the user's walking direction in our navigation system, we apply the Kalman filter and rotation matrix. We then develop optimization algorithms to bound the local position estimation into small 2-dimensional intervals using the interval analysis and dynamic estimation. After transforming the history of gravitational vectors to a fixed local-coordinate frames, we are able to construct a topographical map of pathway. We test our methodology in controlled conditions on an instrumented treadmill and also outdoors where GPS is available. We then use our topographic mapping to augment the results from pedometry and magnetic mapping to obtain better geolocation.

1. INTRODUCTION

1.1 Background and Motivation

The current problem in Global Positioning System (GPS) on mobile phones is that it is not available in areas such as urban, natural canyons, forests, and indoor environments [1]. In daily life, many civilians demand accuracy of the GPS receiver. Over the past years, several ways have been developed to obtain more accurate position estimation of the sensor. The received signal strength (RSS) [2] and time difference of arrival (TDOA) [3] are the main approaches to use available mobile signals and errors around $4 \sim 12$ dB and $10 \sim 60$ meters, respectively. Another approach to improve sensor performance is to use radio frequency identification (RFID) with indoor Wi-Fi [4]. A new method from our group shows that magnetic field intensity maps based on interval analysis has smaller errors than the RSS, TDOA and RFID for estimating geolocation in some areas where GPS is not accessible [5]. Magnetic intensity map is one of the possible cases that can be constructed by collecting sensor data on the meshed grid. In this paper, we develop novel algorithms where sensor measurements on the cell phone are used to construct topographic maps and aid cell phone geolocation. We firstly figure out the vertical accelerations using FFT analysis and low pass filter to obtain clear steady data. Before analyzing position estimation, we need to use stride detection method in order to find the user's steps. We then develop optimization algorithms [6] to bound the local position estimation into small 2-dimensional intervals and find position estimation using interval analysis and dynamic estimation. After transforming the history of gravitational vectors to a fixed local-coordinate frame, we are able to construct a topographical map of the region. We test our methodology in controlled conditions on a instrumented treadmill and also outdoors where

GPS is available. We then use our topographic mapping to augment the results from pedometry and magnetic mapping to obtain better geolocation.

1.2 Outline of Thesis

Chapter one presents the background of a priori in obtaining accurate geolocation, the motivation and methodology of this research. Chapter one also includes the overall outline of thesis and overall algorithmic framework.

Chapter two presents the sensor characterization and data collection from the sensors in the cell phone.

Chapter three presents the methodology of data extraction and selecting the specific range of data to be used for stride detection and for virtual pedometry.

Chapter four presents the methodology of stride detection by peaks and the relationship between the stride length and stride interval. This chapter also shows the optimal estimation of yaw angles using the Kalman filtering analysis, compensation of rotation using Euler angle and position estimation using interval analysis and dynamic estimation.

Chapter five presents the idea behind topographic mapping, gravity analysis and construction of locally absolute coordinate frames to build topographical map.

Chapter six presents testing for various angles of inclination such as flat surface, inclined surface, declined surface and unknown outdoor environment. It also shows the maps constructed.

Chapter seven presents the summary, conclusions of the thesis, error analysis and future recommendations. Figure 1.1 summarizes our overall algorithmic framework.

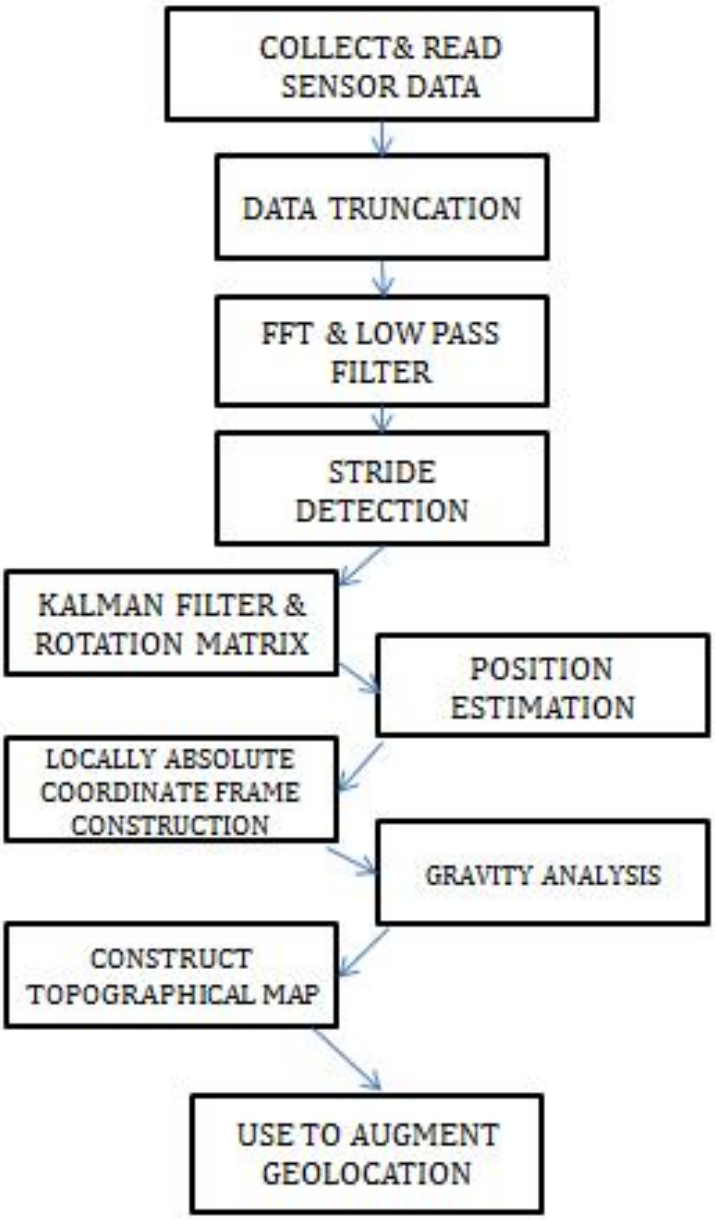

Figure 1.1. Overall Algorithmic Framework.

According to overall algorithmic framework as shown in figure 1.1, we define each section for clear understanding and it is followed by the steps:

1. Collect and read sensor data: The data produced by sensor from the cell phone is stored as an excel file for offline processing.

2. Data truncation: Steady state walking data is collected to avoid transients of starting and stopping.

3. FFT and low pass filter: Given the steady walking from step 3, we use FFT to determine a cutoff frequency to eliminate other high frequencies. We then use a low pass filter with that cufoff frequecy to obtain clean steady data.

4. Stride detection: Given clear steady data, peak values of the vertical acceleration are corresponded to step occurrences as based on the threshold and conditional input value.

5. The Kalman filter and rotation matrix: Estimate yaw angles to find users present walking direction with the vertical angular rate measurements and compensate rotation effect using Euler angle when users body moves.

6. Position estimation: Based on the interval analysis and dynamic estimation, the lateral and horizontal positions are estimated.

7. Locally absolute coordinate frame construction: The sensor measurements on the cell phone transform to locally absolute coordinate frame using the cross product operation with gravitational and magnetic field vectors.

8. Gravity analysis: The angles of inclination in pathway are estimated using the local standard gravity and local calibrated gravitational vectors.

9. Construct topographical map: Given the position estimation and the angles of inclination, the construction of topographical map algorithms are developed using the slope of line in 3-dimensional coordinate frames.

10. Use to augment geolocation: Topographic mapping will augment the results from pedometry and magnetic mapping to obtain better geolocation.

2. SENSOR CHARACTERIZATION AND DATA COLLECTION

This chapter presents the sensor characterization and the methodology for collecting data from the sensor in the cell phone.

2.1 Sensor Characterization

In this research, we use an application in the android operating system, by the name of Sensor stream IMU+GPS, which can collect the sensor data in a certain period of time. The sensor stream IMU+GPS allows for user to observe, select or record the current values of accelerometer, gyroscope (angular rate), magnetic field, GPS position and received signal strength indication (RSSI) in 3-dimensional coordinate frames. The sampling frequency of the sensor can be adjusted by selecting one of several available frequencies. It is also available for users to save sensor data to SD card and PC for offline data processing. For real tests in this research, we use the Samsung Galaxy S4 and its application of sensor as shown in figure 2.1.

Figure 2.1. Samsung Galaxy S4 and sensor stream IMU+GPS.

2.2 Data Collection

The steps of sensor data collection are followed by:

Step 1. Collect and read the vertical acceleration measurements from the sensor in the cell phone which will be used both for stride detection and topographical mapping.

Step 2. The x- and y-coordinate acceleration measurements on the sensor from the cell phone are used to estimate positions in the lateral and horizontal coordinate frames.

Step 3. The gyroscope measurements are used to estimate walking direction by applying the Kalman filtering and to compensate rotation effect by using Euler angle.

Step 4. The magnetic field measurements are used as an unit vector, which will be fused with gravitational measurements, to construct the locally absolute coordinate frames.

Step 5. The gravitational measurements are used to find the angles of inclination by comparing to the local standard gravity.

Step 6. The GPS location measurements are used as reference values comparing to our methodology in outdoor environment test.

For the real test in this research, we setup 50 Hz as sampling frequency which is one of available frequecies that users select. Since the frequency of human walking and running cannot be over 4 Hz, it is available to avoid aliasing in this system. we use 50 Hz of sampling frequency to be used in the process of fast fourier transform and low pass filtering. After the process of low pass filtering is completed, we then estimate the acceleration measurements which are intial sources to find step occurrences and position estimation in the horizontal and lateral coordinate frames.

Figure 2.2. Smartphone location for measurement.

We collect measurements from the sensor in the cell phone that is located inside the shoes during the test for convenient and comfort as shown in figure 2.2. For the orientation of inertial sensor in the cell phone, it is perpendicular to the surface in order to establish real test. The reason for selecting the location of the cell phone is to estimate the accurate gravitational measurements from the sensor which will be compared to the local standard gravity and find the angles of inclination as shown in the figure 2.3.

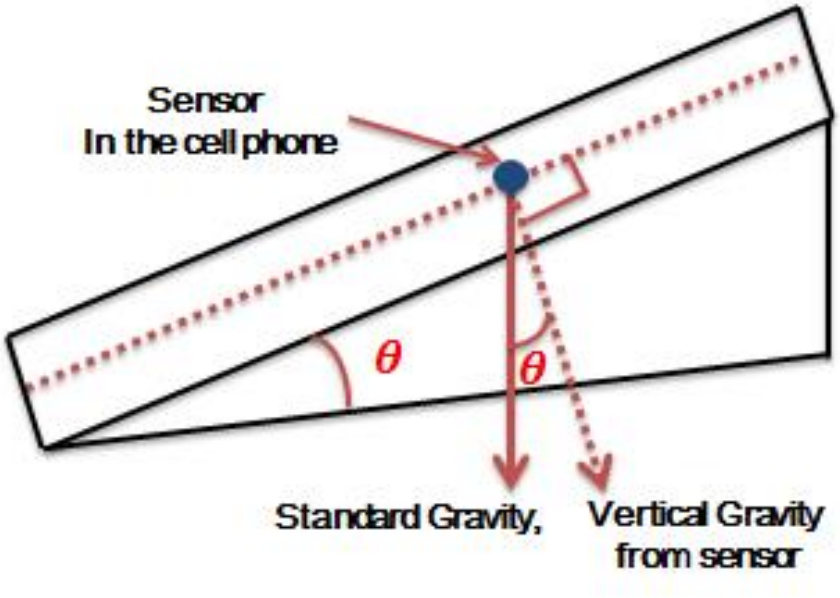

Figure 2.3. Estimation of angle of inclination using gravity vectors.

To avoid the oscillation, shoes is tightened with string as much then the inertial sensor in the cell phone can be fixed in shoes. To get more accurate measurements, we will use rotation matrix using Euler angle to compensate oscillation of human body, equations of International Gravity Formular 1980 and the Free Air correction to find the local standard gravity at specific location. In addition, we will transform the gravititional measurements to the locally absolute coordinate frames which are suitable for our research.

3. DATA ANALYSIS

In this chapter, we present the methodology of data analysis to get accurate measurements from the sensor in the cell phone and reduce errors. We then find the accurate vertical acceleration that will be used to find step occurrences in our research.

3.1 Data Truncation

We truncate irrelevant measurements of the vertical acceleration from the sensor in the cell phone. As shown in figure 3.1, the initial and terminal states of the vertical acceleration are assumed that begin and stop walking period. Since we need to find the vertical acceleration in steadily walking period which is corresponded to proper steps, the initial and terminal states are redundant in our research.

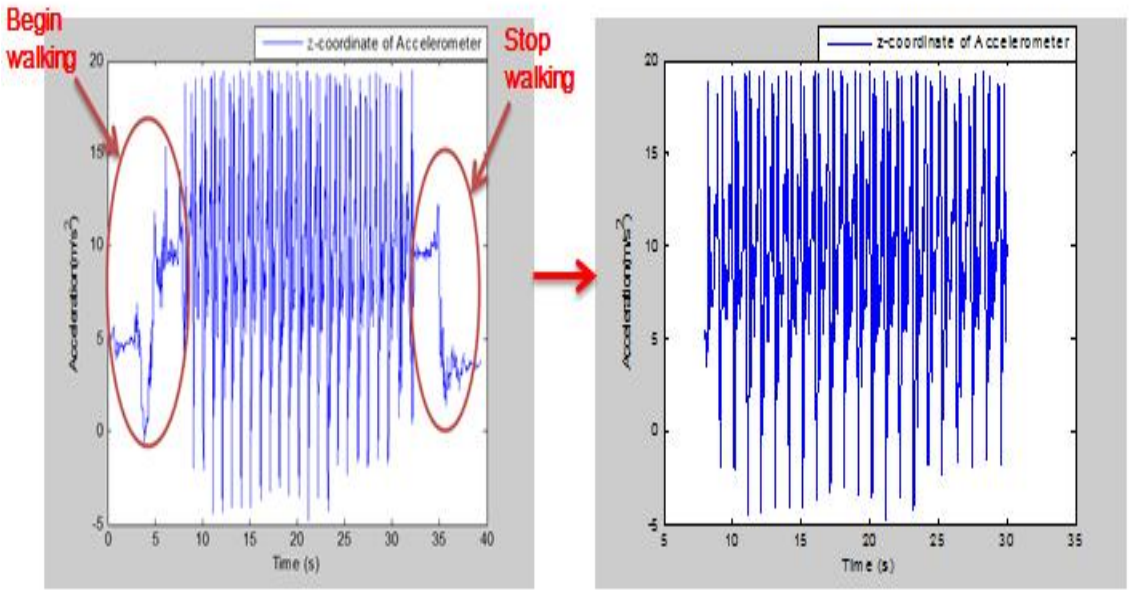

Figure 3.1. The vertical acceleration measurements truncation.

According to the vertical acceleration measurements truncation as shown in figure 3.1, the time intervals in a range between 0 to 8 seconds and 32 to 40 seconds assume that begin and stop walking. After the process of truncation is completed, we can obtain the steadily walking period which is also shown in figure 3.1. We then apply the vertical acclerations in steadily walking into the fast fourier transform and low pass filtering with a selection of 50 Hz sampling frequency. Therefore, we can more effectively minimize the sensitive noise and estimate the accurate vertical accelerations that indicate user's walking steps.

3.2 Fast Fourier Transform Analysis and Low Pass Filtering

Fast fourier transform (FFT) is a widely used in the data signal processing and data analysis. Fourier transform analysis converts signal from time domain to frequency domain and vice versa in order to do data processing. Fast fourier transform (FFT) can rapidly compute such as the transformations by factorizing the discrete fourier transform matrix into a product of sparse factors [7]. The mathematical equations of forward and inverse discrete fourier transform show below.

$$X_k = \sum_{n=0}^{N-1} x_n e^{\frac{-i2kn}{N}}, \quad k = 0, 1, 2, ..., N-1 \tag{3.1}$$

Forward Discrete Fourier Transform(DFT)

$$x_n = \frac{1}{N} \sum_{k=0}^{N-1} X_k e^{\frac{2kn}{N}}, \quad k = 0, 1, 2, ..., N-1 \tag{3.2}$$

Inverse Discrete Fourier Transform(IDFT)

Let $x_0, x_1, ...x_{N-1}$ be complex numbers, X_k be the result of DFT and N is number of output.

The most common used algorithm for the FFT is the Cooley-Tukey which expresses as discrete fourier transform (DFT) of size $N = N_1 * N_2$ where N_1 and N_2

are small size of outputs in DFT to save the computation time O(N log N), recursively[8][9]. Since the Cooley-Tukey algorithm is to divide the transform into two pieces of size $\frac{N}{2}$ at each step, N is commonly expressed as a power of two. In our research, we use N as the length of the vertical acceleration measurements from the embedded sensor on the cell phone. From the Cooley-Tukey algorithms, the DFT of the function can be rearraged into a sum over the even-number and a sum over the odd-number as shown below

$$X_k = \sum_{n=0}^{N/2-1} x_{2n} e^{\frac{-i2kn}{N/2}} + \sum_{n=0}^{N/2-1} x_{2n+1} e^{\frac{-i2kn}{N/2}} = Y_k + Z_k, \quad k = 0, 1, 2, ..., \frac{N-1}{2} \tag{3.3}$$

Where,

Y_k is the DFT of the even-indexed inputs

Z_k is the DFT of the odd-indexed inputs

As equation above re-expresses in terms of Y_k and Z_k,

$$X_k = Y_k + e^{\frac{-i2k}{N}} Z_k, \quad k = 0, 1, 2, ..., \frac{N-2}{2} \tag{3.4}$$

Since Y_k and Z_k are periodic sequences with $\frac{N}{2}$, these two terms can be expressed as

$$Y_k = Y_{k+\frac{N}{2}} \tag{3.5}$$

$$Z_k = Z_{k+\frac{N}{2}} \tag{3.6}$$

The X_k with periodicity is shown below

$$X_{k+\frac{N}{2}} = Y_k + e^{\frac{-i2k}{N}} Z_k, \quad k = 0, 1, 2, ..., \frac{N-2}{2} \tag{3.7}$$

According to the Cooley-Tukey algorithm, the splitting algorithm can be used to compute Y_k,Z_k and X_k to reduce the time and we use MATLAB programming to compute the splitting alogrithm automatically.

The transformation from x_n to X_k indicates a conversion from configuration to frequency space, and this process is useful in seeing the characteristic spectrum of a signal. As based on the theoretical equations above, we find the single-sided amplitude spectrum of the vertical accleration measurements over frequencies. According to the analysis for single-sided amplitude spectrum of the vertical acceleration measurements, we estimate a cutoff frequency, peak to peak passband and stop band of attenuation to be used in low pass filtering and find the filtered vertical acceleration measurements.

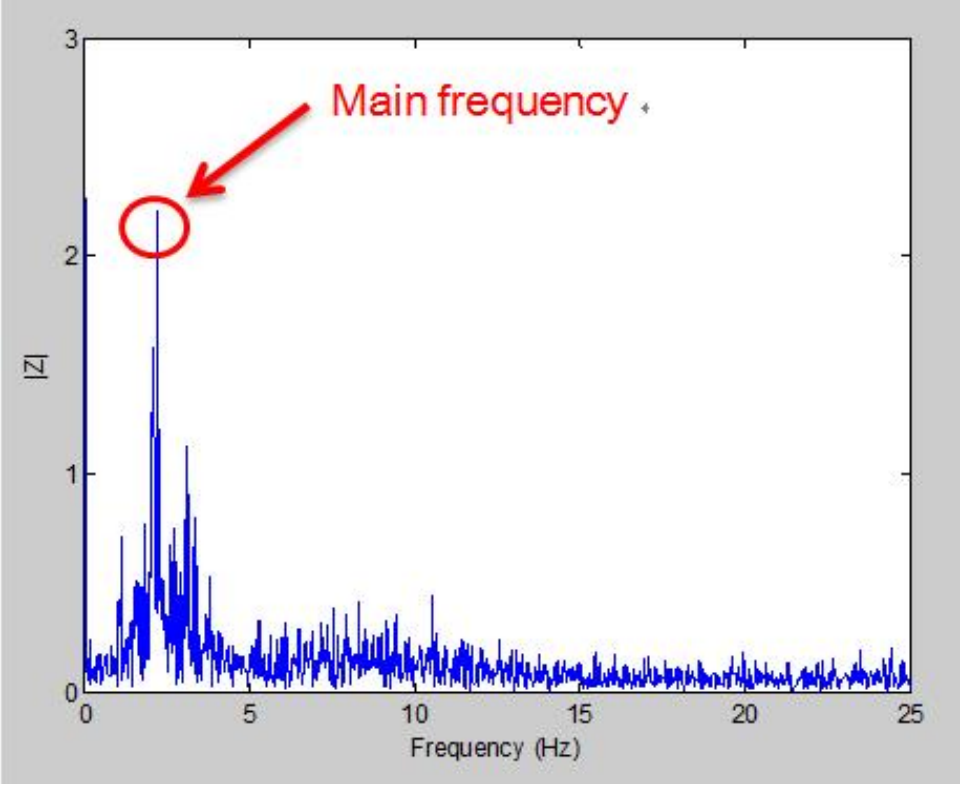

Figure 3.2. One sample in a single-sided amplitude spectrum of vertical acceleration.

According to a single-sided amplitude spectrum of the vertical acceleration which is shown in figure 3.2, the main frequency is about 2 Hz and it indicates that most of the walking period is about 0.5 seconds. Other high frequencies indicate the measurement noise which is caused by oscillation of the cell phone bonded to the user's body

and shoes. After most of the walking period is estimated, we use low pass filtering to find the filtered vertical acceleration measurements which will be used for the stride detection. We then compare the acceleration measurements using low pass filter with the acceleration measurements using truncation method.

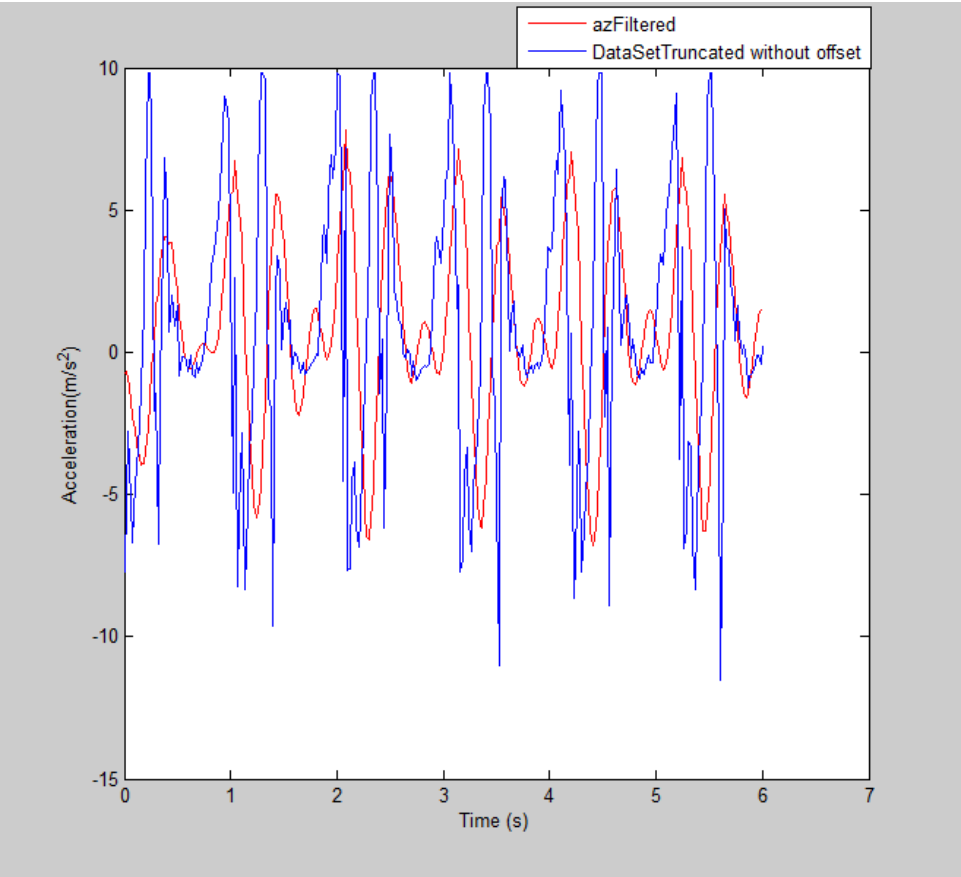

Figure 3.3. Filtered acceleration vs. acceleration using truncation method.

As shown in figure 3.3, the filtered vertical acceleration becomes shift up, start at zero and compensate for gravity effect as compared to acceleration data using truncation method. Therefore, low pass filtering reduces the disturbances and noises that it helps to figure out the accurate vertical acceleration measurements which are corresponded to step occurrences. After data analysis from chapter 3, we then detect steps and find position estimation in next chapter.

4. DETECTION AND ESTIMATION FROM MEASUREMENTS

In this chapter, we present the discrete system for stride detection, analysis of peak values and the relationship between the stride lengths and stride intervals . This chapter also presents the analysis of the walking direction using the Kalman filtering analysis, rotation effect with the Euler angle, position estimation in the lateral and horizontal coordinate frames.

4.1 Discrete Time System for Stride Detection

The discrete time system is a model which the state variables are only available to change at a countable number of points over time [10]. We select the stride detection as discrete model and analyze the peak values of the vertical acceleration which are corresponded to the step occurrences.

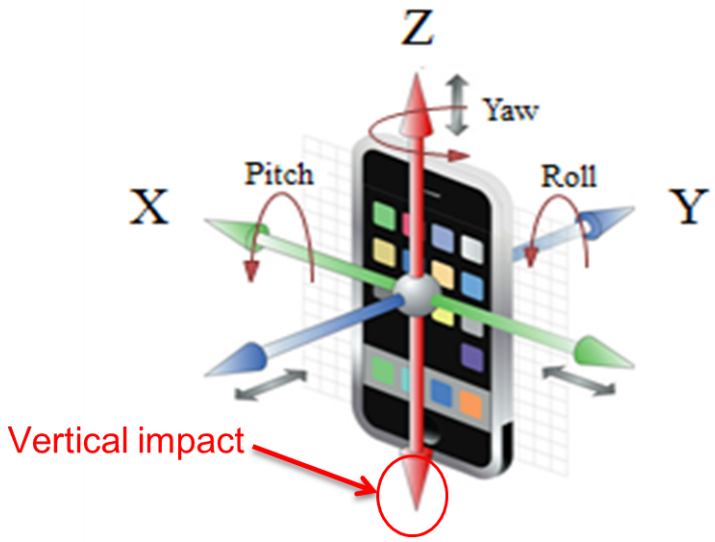

Figure 4.1. The orientations of inertial sensor in the cell phone.

The basic concept of stride detection is that the vertical acceleration of the walking is produced by z-directional impact when the users foot hit on the ground as shown in figure 4.1. We assumed that z-directional impact is considered as step when it is larger than the threshold of the vertical acceleration measurements. The detailed steps for selecting the threshold are presented in 4.1.2.

4.1.1 Relationship between Stride Length and Stride Interval

In our research, we notice that the relationship between stride length and stride interval help us find the accurate cell phone user's steps in various cases such as walking, running, hiking, etc. Since various cases have different walking characters, finding the relationship between stride length and stride interval is important factor to figure out the accurate position estimation. Before we estimate the velocity and position from the acceleration measurements, it is necessary to analyze the relationship between the stride length and stride interval. The relationship between the stride length and stride interval in theoretical biology has been introduced [11] and the previous researches have been developed that the users total walking activity can be divided into the designated pieces to reduce the error in long distance navigation [12]. As based on previous researches, we select the section of walking path and measure different walking steps in the designated path as shown in figure 4.2.

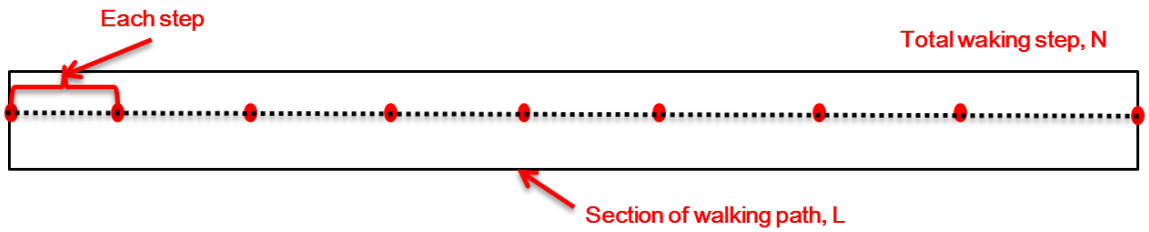

Figure 4.2. Section of walking path.

We then divide total lengths of walking path by total steps, collect data from various cases of walking steps and figure out the fitted curve between the stride length and stride interval. The fitted curve has a power function and it is re-calibrated by real test to obtain precise data. The relationship between stride length and stride interval has been expressed as

$$\Delta t_k = t_i - t_{i-1} \tag{4.1}$$

$$\Delta s_k = C\Delta {t_k}^{b-1/b} + C_1 \tag{4.2}$$

Where,

Δt_k = The stride interval which is detected for each stride

Δs_k = The estimated stride length which is applied to estimate correct position

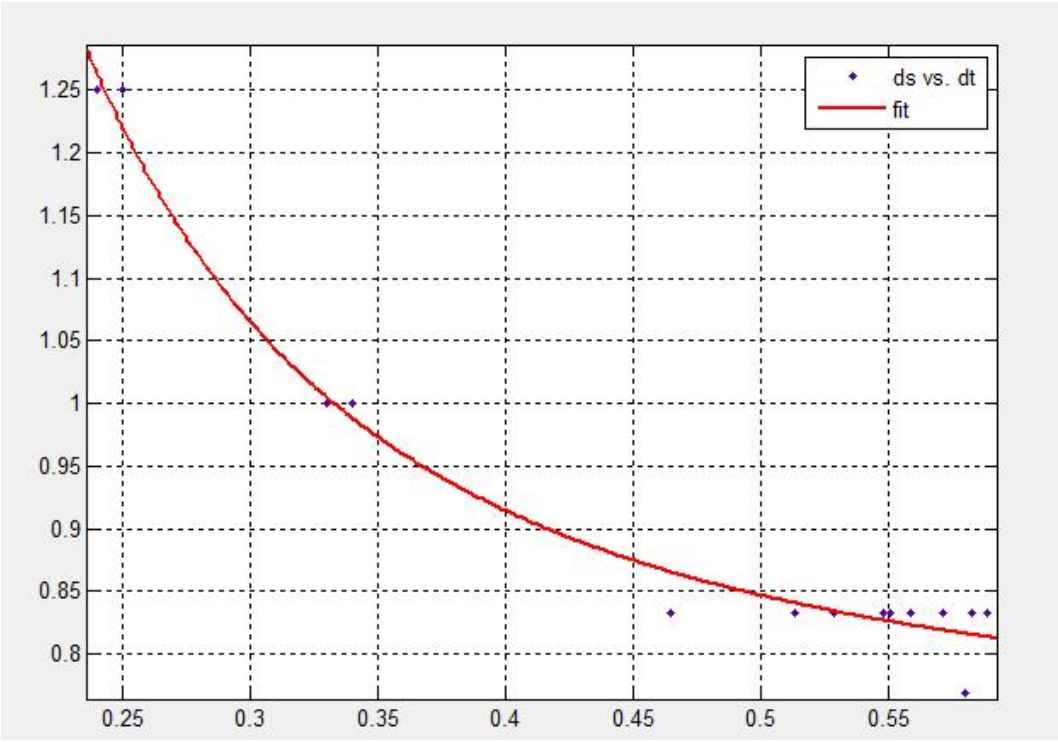

Figure 4.3. The fitted curve between the stride length and stride interval.

From the fitted curve between stride length and stride interval, we find the level of quantity in fitting such as R^2, Adjust R^2, SSE, and the root mean square error (RMSE). Each variable from above equations, variable C, b and C1, is 0.02637, -2.103 and 0.7333, respectively. The variables also have minimum and maximum values which are accessible in the fitted curve equation. Compared to the results in the previous research [12], we obtain more precise variables with the 95 percentages confidence bound and 98 percentages of confidence interval (R^2). The new fitted function is a better approach than former results, which has trends of human walking character as shown table 4.1.

Table 4.1. The index variables of stride model.

Variables	Former Numerical values[12]	Current Numerical values
b	$1.1060e-3$ $(0.4259e-3, 2.6380e-3)$	0.02637 $(-0.009851, 0.06259)$
C	0.1311 $(0.1158, 0.1670)$	-2.103 $(-2.97, -1.235)$
C_1	0.7707 $(0.7406, 0.8007)$	0.7333 $(0.6483, 0.8184)$
R^2 value	0.9507	0.9861
Adjust R^2 value	0.9476	0.9838
RMSE	0.04427	0.02223

4.1.2 Detecting by Peak Value Points

There are some previous researches to detect each step such as detection of zero-crossing points and detection of peak values [12]. In this research, we use to detect by peak values since the detection of zero-crossing points from previous research [12] is not appropriate for larger stride length in real test, detection by peak values has

a little disturbance affect and it does not rely on initial compensation. The steps for detecting by peak values are followed by:

Step 1: Detect the peak values of the vertical accelerations which were estimated using fast fourier transform and low pass filtering as shown in chapter 3.

Step 2: Set the condition value which is an input for the function by examining the peak values along the input data with a pre-defined threshold value, τ.

Step 3: If the default condition is 1, we have initial two threshold values as 40 percent of the original input data.

Step 4: If the default condition is 0, then the threshold value holds for all through the process.

Step 5: If the default condition is 2, then the algorithms use stop sign to fix backward.

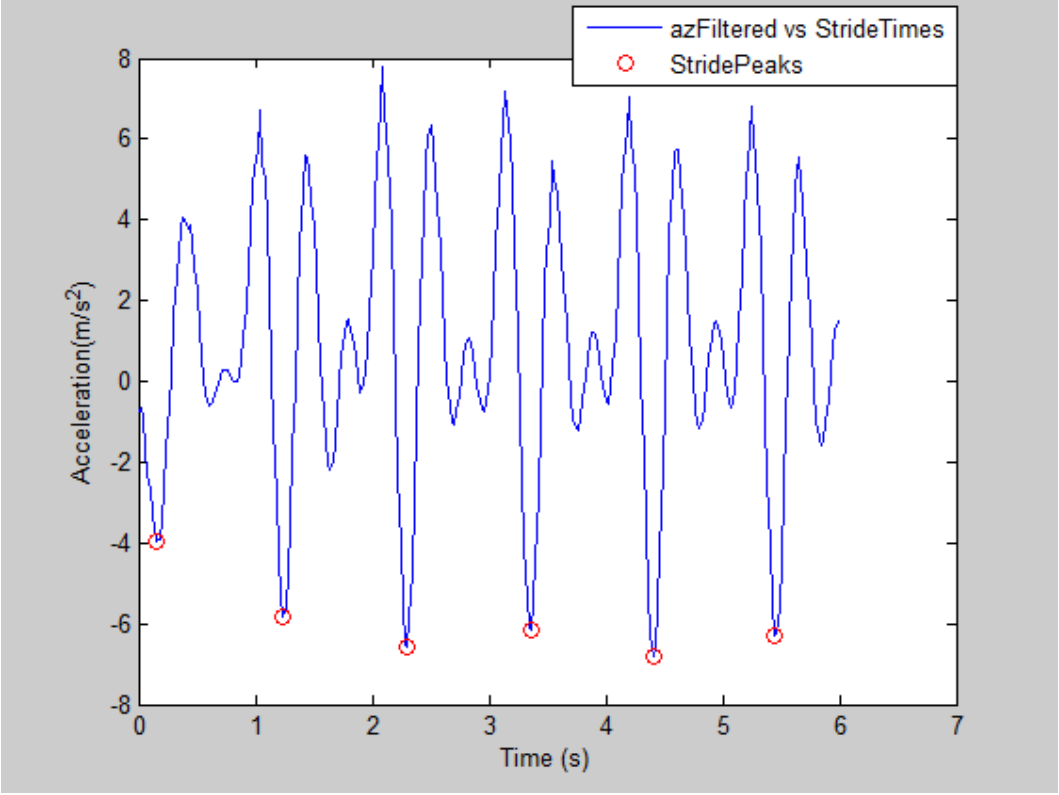

Figure 4.4. The stride detection for peak values.

According to figure 4.4, the vertical acceleration has a minimum value below -4(g) so it needs to be filtered by $\tau = -4(g)$ to eliminate other interferences. We can express a range of the vertical accelerations to be filtered as below and find the peak values.

$$a_z = \begin{cases} a_z, & a_z \leq \tau \\ 0, & a_z > \tau \end{cases} \tag{4.3}$$

Having the stride detection by peak values, we can figure out the accurate step occurrences. The step occurrences will be used as initial sources for estimating user's position and finding the angles of inclination using the gravitational vectors. In the next chapter, the Kalman filtering analysis and rotation matrix with Euler angle are applied to estimate user's present walking direction and compensate for rotation effect.

4.2 Kalman Filtering Analysis

The purpose of using the Kalman filtering analysis in our research is to find the accurate yaw angles of gyroscope measurements which can figure out the users walking direction at certain time. The Kalman filtering is also available to update the yaw angle measurements to estimate the overall walking direction. In general, the random noise are occurred in the systems and measurement procedure. The Kalman filtering has developed many years for navigation systems to reduce the random noises in the IMUs [13]. For the walking paths, the cell phone users can walk in various directions such as straight, backward, deviated paths, and, etc. Having the Kalman filtering analysis, We reduce the random noises of yaw angles from the sensor in the cell phone and estimate the user's walking direction. Figure 4.5 shows the example of the Kalman filter process.

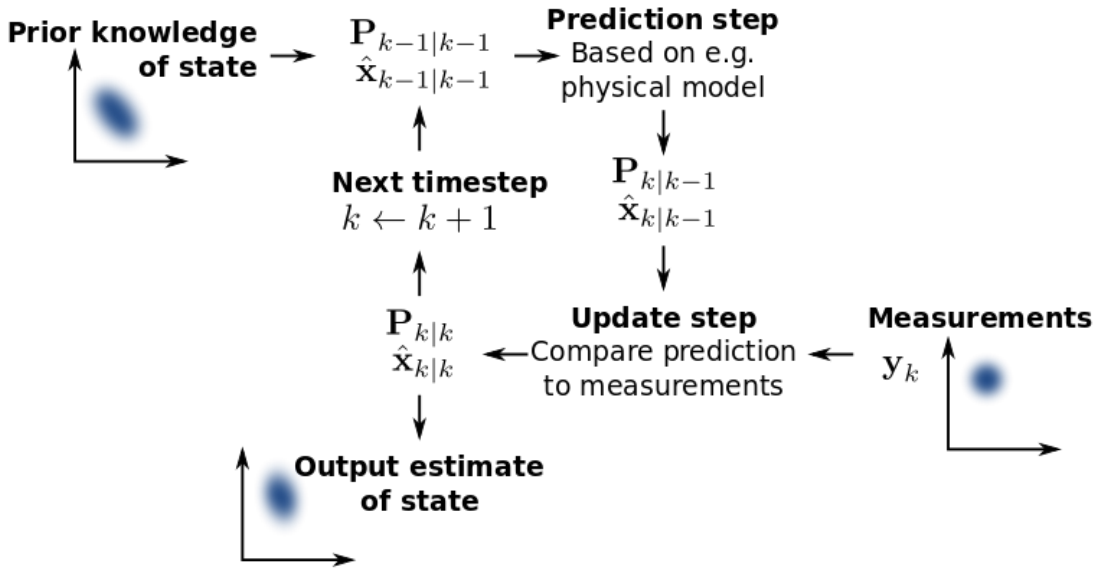

Figure 4.5. The process of Kalman filter.

The Kalman filtering on the discrete times is applied into linear dynamic systems. The state of the Kalman filter system represents in terms of a vector of real numbers and generates the new state at each incremental time. We firstly introduce the basic equations of the Kalman filter which is expressed as

$$x_k = F_k x_{k-1} + B_k u_k + w_k \tag{4.4}$$

Where,

x_k= process state at time t_k

F_k= the state transition model which is applied to the x_{k-1}

B_k= the control-input model which is applied to the u_k

w_k= the process noise (zero mean multivariate normal function with covariance,Q_k)

In this system, $x(k)$ consists of the vertical rotation angle and the yaw angle (gyroscope) measurements from the sensor which can be expressed as $x(k) = (\phi_k, w_{zk})^T$ with initial condition, $x(0) = (\phi_0, w_{z0})^T$. The $B_k u_k$ term does not exist in our system since we don't have any known control inputs. The z_k of the state x_k which is collected directly from the sensor as shown below

$$z_k = H_k x_k + v_k \tag{4.5}$$

Where,

z_k = measurement at time t_k

H_k = the observation model which connects between measurement and state vector at t_k

v_k = the noise in the system itself and measurement

Having the $x(k)$ and $z(k)$ equations, following invariants are defined with initial estimation of $E[x(0)]_{t=0} = \hat{x_0}$ and $E[(x_0 - \hat{x_0})(x_0 - \hat{x_0})^T]_{t=0} = P_0$.

$$E[w_k {w_i}^T] = \begin{cases} Q_k & i = k \\ 0 & i \neq k \end{cases} \tag{4.6}$$

$$E[v_k {v_i}^T] = \begin{cases} R_k & i = k \\ 0 & i \neq k \end{cases} \tag{4.7}$$

$$E[w_k {v_i}^T] = \begin{cases} 0, & \forall \text{k and i} \end{cases} \tag{4.8}$$

Since the w_k and v_k noises are considered as white having zero correlations to each other, we can make the simplified system. The Kalman filter model assumes the true state at time k is developed from the state at time, $(k-1)$ and the covariance is propagated as shown below

$$\tilde{x}_k = \tilde{x}(k \mid k-1) = F_k \hat{x}(k-1 \mid k-1) \tag{4.9}$$

$$\tilde{P}_k = \tilde{P}(k \mid k-1) = F_k \tilde{P}_(k-1 \mid k-1) F_k^T + Q_k \tag{4.10}$$

Where,

$\tilde{x}(k \mid k-1)$ = process state vector at time k given observations up to and including at time $(k-1)$

F_k= state transition model which is applied to the x_{k-1}

$\tilde{P}(k \mid k-1)$ = the covariance at time k given observations up to and including at time $(k-1)$

Q_k= the noise covariances

Having the updated state $\hat{x}(k \mid k-1)$ with the covariance $\tilde{P}(k \mid k-1)$, we can find the updated measurements which are applied to with z_k and use each variable as shown below

$$\hat{x}_k = \tilde{x}_k + K_k[z_k - H_k x_k] \tag{4.11}$$

$$\hat{P}_k = [I - K_k H_k]\tilde{P}_k \tag{4.12}$$

$$K_k = P_k {H_k}^T (H_k \hat{P}_k {H_k}^T + R_k)^{-1} \tag{4.13}$$

Where,

K_k = the optimal Kalman gain at time t_k

In our research, each matrix is determined as shown below.

$$F_k = \begin{bmatrix} 1 & 0.02 \\ 0 & 1 \end{bmatrix} H_k = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} Q_k = \begin{bmatrix} 0.004 & 0 \\ 0 & 1 \end{bmatrix} R_k = \begin{bmatrix} 0.004 & 0 \\ 0 & 4 \end{bmatrix} \tag{4.14}$$

Having the Kalman filtering analysis, the random noises of yaw angles can be minimized then find the accurate values. We apply the results of the Kalman filtering with rotation matrix to estimate user's walking direction at certain time as shown in outdoor environment test in chapter 6.

4.3 Rotation Effect

Since the sensor in the cell phone is bonded directly to user's foot inside shoes, rotation effect in a certain angle needs to be considered when user's body moves. We apply the acceleration and angular rates measurements from the sensor in the cell phone into Euler angles in order to compensate the rotation effect and estimate the accelerations and angular rates precisely [12]. The rotation angle can be expressed as matrix which is shown below [14].

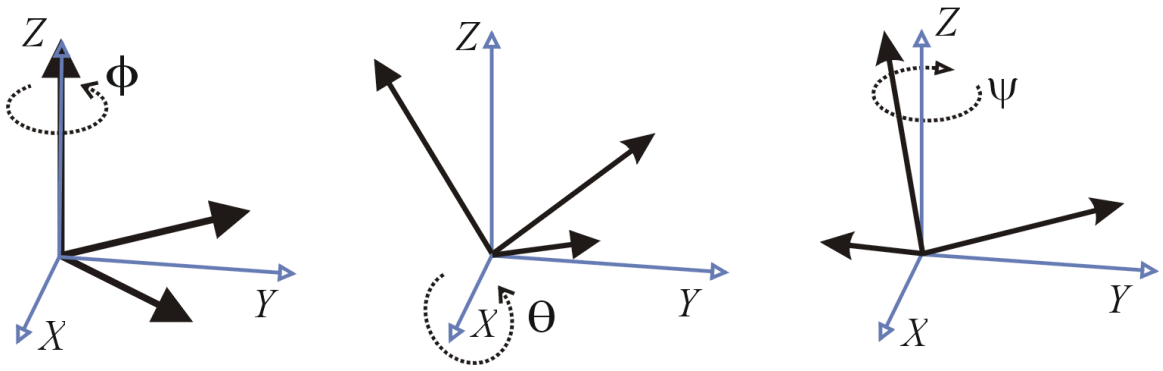

Figure 4.6. Rotation coordinate frames.

$$R_{3x3} = \begin{bmatrix} C\theta C\phi & S\psi S\theta C\phi - C\psi S\phi & C\psi S\theta C\phi + S\psi S\phi \\ C\theta S\phi & S\psi S\theta S\phi + C\psi C\phi & C\psi S\theta S\phi - S\psi C\phi \\ -S\theta & S\psi C\theta & C\psi C\theta \end{bmatrix} \tag{4.15}$$

Where ψ, θ and ϕ represent the rotation angle from x-, y- and z- coordinate frames. C and S represent *cosine* and *sine*. The rotation matrix is considered as the sequence of rotations that we analyze about x-, y- and z-coordinate axis. As we apply rotation matrix to the acceleration and angular rate, these updated variables can be expressed as

$$a_k = R_{3x3} \cdot (\phi_k \cdot a_{k-1}) \tag{4.16}$$

$$w_k = R_{3x3} \cdot (\phi_k \cdot w_{k-1}) \tag{4.17}$$

Where a_k and w_k are the updated acceleration and angular rate at each sample times from previous state. We can also apply rotation matrix into stride length for each dimensional displacement and it is expressed as

$$\begin{pmatrix} \Delta x \\ \Delta y \\ \Delta z \end{pmatrix} = R_{3x3} \cdot ds \tag{4.18}$$

Where,

Δx = x-axis displacement using rotation matrix and stride detection

Δy = y-axis displacement using rotation matrix and stride detection

Δz = z-axis displacement using rotation matrix and stride detection

We also apply rotation matrix into the gravitational vectors in the locally absolute coordinate frames to find the accurate angles of inclination in the user's pathway. The way to construct the locally absolute coordinate frames and estimate the accurate gravitational vectors in the locally absolute coordinate frames will present in chapter 5.

$$\begin{pmatrix} {G_x}' \\ {G_y}' \\ {G_z}' \end{pmatrix} = R_{3x3} \cdot [G_x, G_y, G_z]^T \tag{4.19}$$

Where,

${G_x}'$ = x-axis gravitational vector in the locally absolute coordinate frame using rotation matrix

${G_y}'$ = y-axis gravitational vector in the locally absolute coordinate frame using rotation matrix

${G_z}'$ = z-axis gravitational vector in the locally absolute coordinate frame using rotation matrix

Having a consideration of rotation effect using Euler angle, we then develop algorithms of static estimation using interval analysis and dynamic estimation for the

accurate position estimation in the lateral and horizontal coordinate frames on the topographical map.

4.4 Position Estimation

Position estimation is necessary to work on the process of constructing the topographical map which is corresponded to the gravitational measurements on the same location. To obtain the accurate position estimation, it is also necessary to deal with the static estimation and the dynamic estimation which aids to improve the estimation accuracy for keeping the positions inside the region of estimated uncertainties. For static estimation, we use interval analysis (IA) to reduce random noises and integrate topographical map easily. There has been introduced interval analysis which is able to estimate system state [6][15][16] and fuse the magnetometer with the IMUs to improve the accuracy of estimation and to be stable the system [5]. Since our task deals with random fields and the Kalman filter or the particle filter are the process for stationary noise, interval analysis is more fitted than the Kalman filter or the particle filter in our research. The concept of interval analysis can be expressed as

$$[x] = [\underline{x}, \bar{x}] \tag{4.20}$$

$$\underline{x} = \inf\{a \in R \cup \{-\infty, \infty\} \forall x \in [x], a \leq x\} \tag{4.21}$$

$$\bar{x} = \sup\{b \in R \cup \{-\infty, \infty\} \forall x \in [x], x \leq b\} w([x]) = \bar{x} - \underline{x} \tag{4.22}$$

Where,

$$\underline{x} = \text{Lower bound on one dimension R}$$
$$\bar{x} = \text{Upper bound on one dimension R}$$
$$w([x]) = \text{Width on one dimension R}$$

Interval analysis is extended to the four classical operations of real arithmetic such as addition, subtraction, multiplication and division which can be defined as

$$[x] \cdot [y] = [\{x \cdot y \in R \mid x \in [x], y \in [y]\}] \tag{4.23}$$

The $\cdot$ indicates any binary operator on the interval [x] and [y]. We can also apply the interval analysis into intersection and union which are expressed as

$$[x] \cap [y] = [\{z \in R \mid z \in [x], z \in [y]\}] \tag{4.24}$$

$$[x] \cup [y] = [\{z \in R \mid z \in [x], z \in [y]\}] \tag{4.25}$$

In order to apply an interval set of certain space in the interval analysis to 2-dimensional estimation, we use the definition of interval real vector.

Definition (Interval Real Vector)[6] The interval vector [x], which is a subset of R^n, can be defined as n closed intervals in the Cartesian product and it is expressed as

$$[x] = [x_1] \times [x_2] \cdots \times [x_n] \tag{4.26}$$

Having the interval vector using Cartesian product, estimation of position with x and y intervals can be defined as a non-empty box.

$$[P_0] = [x_0] \times [y_0] \tag{4.27}$$

$$[P_n] = [P_0] + \sum_{i=1}^{n} \begin{pmatrix} dx_i \\ dy_i \end{pmatrix} \tag{4.28}$$

P_0 represents the initial estimation of position with initial x_0 and y_0. P_n is corresponding to update inertial position estimation. It is also found the width vector of box, $w([p])$ as shown below.

$$w([p]) = (w_x([p]), w_y([p]))^T = (w([x]), w([y]))^T \tag{4.29}$$

Having applied interval analysis, we obtain rectangular axis-aligned two dimensional intervals to express walking path model in static estimation in order to make the accurate system modelling. The result of static estimation for two dimensional position intervals is shown below.

As a result of the static estimation, we are available to figure out the crossing of paths which can be considered as intersections. To obtain more accurate position estimation, we use dynamic estimation to remove the intersection of redundant spaces in

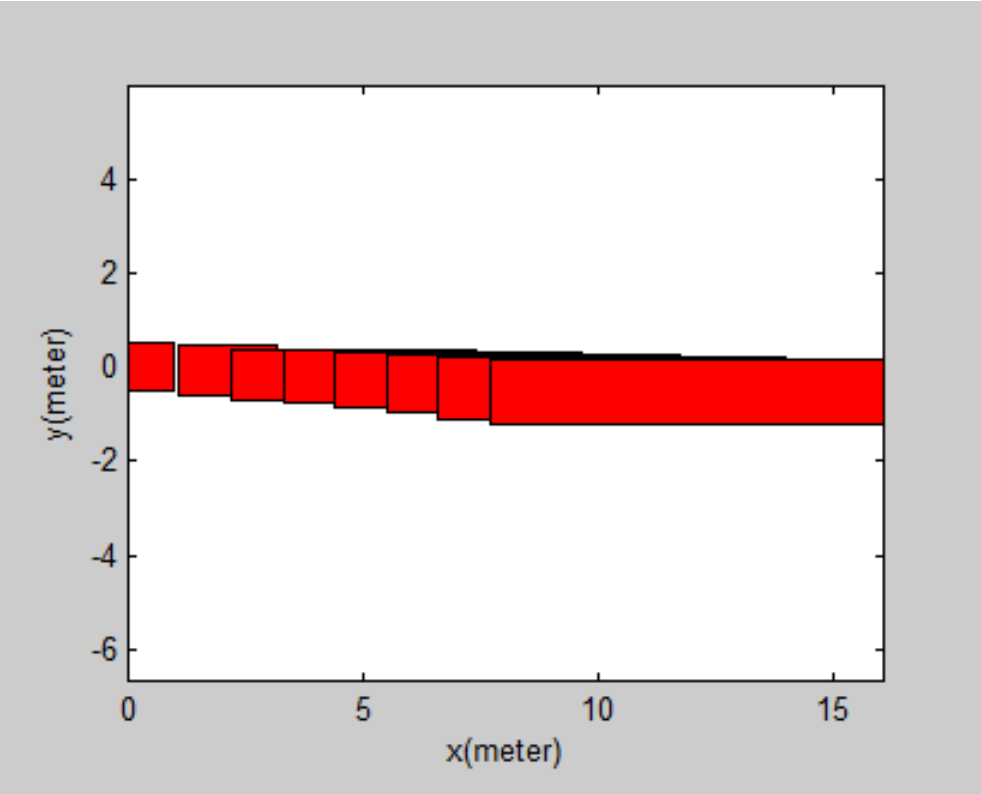

Figure 4.7. Result of static estimation for two dimensional position intervals.

static estimation and to improve the estimation accuracy using the size of estimation error for keeping the positions inside the estimated space. In order to construct the dynamic path, we use the result of static estimation (S_t) from starting point ($t = 0$) to current point,($t = n \cdot \Delta t$) then remove the redundant ones by examining their propagation. We assume that the smallest estimated width, $w([p]) = sup([p]) - inf([p])$ is larger than the true displacement between two grid points, Δd. We shows theorem of relationship between d and $w([p])$ and its proof to get clear understanding of theoretical method.

Theorem Suppose that there are two grid points on 2-dimensional space, (x_{k+1}, y_{k+1}). The displacement between these two grid points can be expressed as

$$\Delta d = (\Delta d_x, \Delta d_y)^T = (x_{k+1} - x_k, y_{k+1} - y_k)^T$$

$$max[\Delta d] < w([p]) \quad then \quad (x_k, y_k) \in [p_k] \quad and \quad (x_{k+1}, y_{k+1}) \in [p_k]$$

Proof Assume that there is displacement along x-direction, $x_{k+1} = x_k + \Delta d_x$ where Δd_x is a range of $[inf\{\Delta d_x\}, sup\{\Delta d_x\}]$. Then, we can define a term x_c as

$$x_c = inf\{x_{k+1}\} - sup\{x_k\} = inf\{x_k + \Delta d_x\} - sup\{x_k\} = inf\{x_k\} + inf\{\Delta d_x\} - sup\{x_k\}$$

$$= (sup\{x_k\} - w(x)) + inf\{\Delta d_x\} - sup\{x_k\} = inf\{d_x\} - w(x) \leq sup\{\Delta d_x\} - w(x) < 0$$

$$\Rightarrow [p_k] \cap [p_{k+1}] \neq \emptyset$$

The displacement along y-direction is same as above equation. So, if there is intersection between the current estimation of interval set and next estimation of interval set, we can remove redundant part and identify the accurate interval of position estimation using above theorem and its proof. Having an integration above theorem with dynamic process, we then obtain the acceleration, velocity and position precisely which are shown below.

$$[a_t] = [a_t - 3\mu_{cc}, a_t + 3\mu_{cc}] \tag{4.30}$$

$$[v_t] = [v_{t-1}] + [a_t] \cdot \triangle t \tag{4.31}$$

$$[p_t] = [p_{t-1}] + \frac{1}{2}[a_{t-1}] \cdot \triangle t^2 + [v_{t-1}] \cdot \triangle t \tag{4.32}$$

Where,

$\triangle t$ = The time interval

a_t = The acceleration including the error vector, μ_{cc}

$[v_t]$ = The updated velocity estimation

$[p_t]$ = The updated position estimation

For error vector, it is not enough to bound the drift of position estimation by applying the Kalman filtering into our navigation system. The measurements from the sensor on the cell phone have uncertainties which are caused by the sensor itself. Especially, the bias in the acceleration and gyroscope measurements propagates to the position estimation as we integrate the measurements as the cube of time. After

the integration from the accelerations in long time, error can be a huge as the way of $Err = \frac{1}{2}v_{err}t^2$. Therefore, we use the results from the virtual pedometry with the stride detection to figure out the accurate position estimation and velocity in every step.

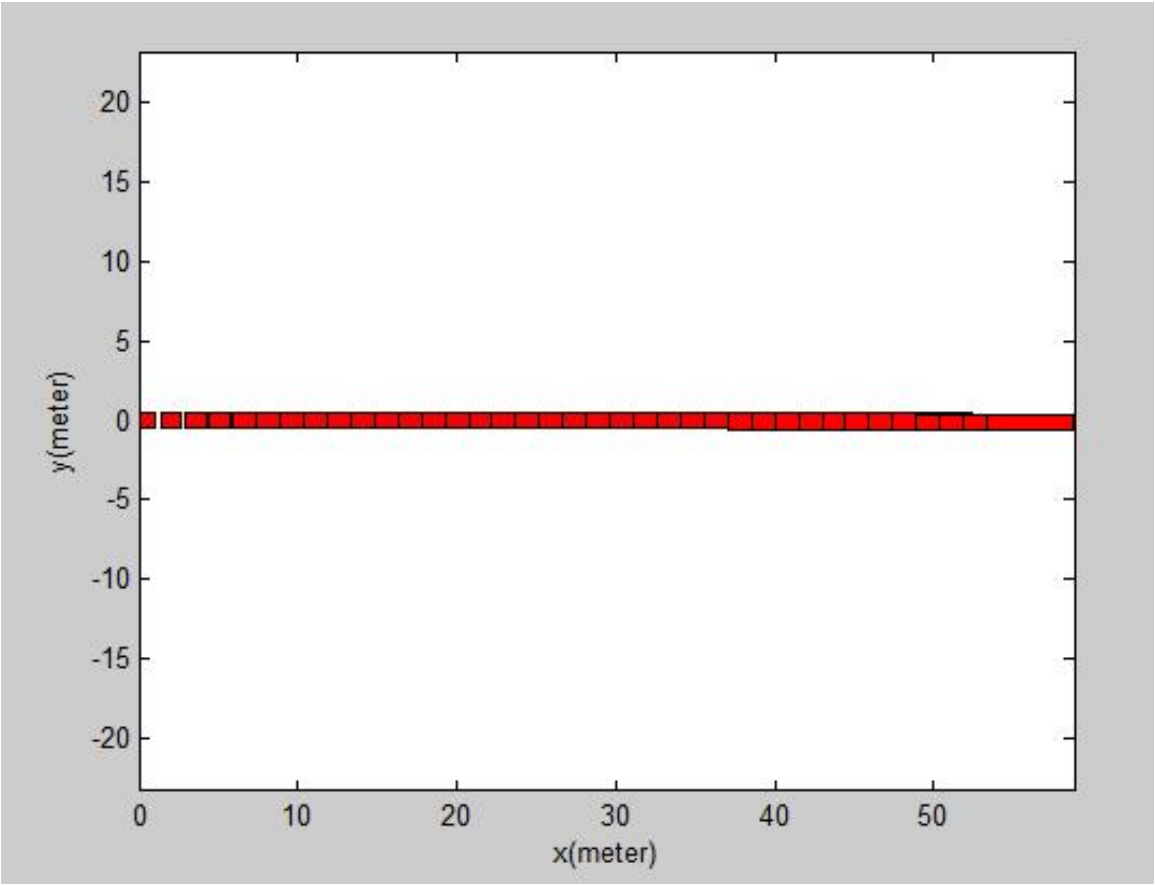

Figure 4.8. Result of dynamic estimation for two-dimensional position intervals.

The result of dynamic estimation for 2-dimensional position estimation is shown in figure 4.8. According to the result of dynamic estimation for 2-dimensional postion intervals, the size of each region becomes smaller and each region has few intersection. Therefore, we can estimate both of indoor and outdoor geolocation without fluctations.

5. CONSTRUCTION OF TOPOGRAPHICAL MAP

The main objective of our research is to find the angles of inclination are the vertical coordinate axis of pathway which aid to construct topographical map. In the previous chapters, we focus on analyzing the position estimation using interval analysis and dynamic estimation. We then develop the algorithms to build the locally absolute coordinate frame and find the angle of inclination using gravity analysis. Finally, we estimate the vertical coordinate axis of pathway and build topographical map.

5.1 Construct Locally Absolute Coordinate Frames

Since the coordinate frames of cell phone and those of the locally absolute coordinate frames are different, the measurements in the cell phone's coordinate frames are necessary to transform into the locally absolute coordinate frames. We then obtain the accurate gravitational vectors to compare the local standard gravity and find the angles of inclination for user's pathway. To construct the locally absolute coordinate frames, we use the gravitational measurements and the magnetic field measurements on the embedded sensor in the cell phone. The basic idea of constructing the locally absolute coordinate frames comes from creating the coordinate frames using two unit vectors [17]. Using the cross product operation with two measurements, it is convenient to construct a new coordinate frames as shown below.

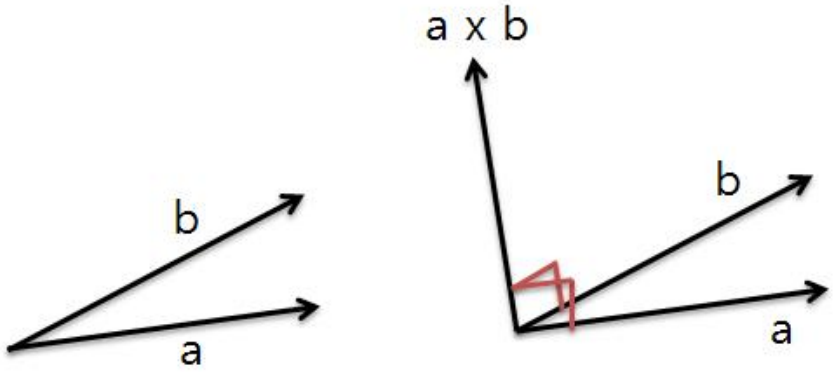

Figure 5.1. The cross product operation with two vectors.

The cross product of unit vector a and b ($a \times b$) is perpendicular to both a and b. We can select either a or b vector to represent one of the coordinate axes. If we select a vector aligned with x-coordinate axis of the locally absolute coordinate frames, the direction of a vector can be corresponded to the x-coordinate of axis in the locally absolute coordinate frames. The z-coordinate axis in the locally absolute coordinate frames is same as the unit vector of cross product operation. The y-coordinate axis in the locally absolute coordinate frames must be perpendicular to both x and z-coordinate axis in the locally absolute coordinate frames. Three unit vectors in the locally absolute frame can be expressed below and figure 5.2 shows the construction of new coordinate frames using two vectors.

$$u_x = \frac{a}{\|a\|} \tag{5.1}$$

$$u_y = u_z \times u_x \tag{5.2}$$

$$u_z = \frac{a \times b}{\|a \times b\|} \tag{5.3}$$

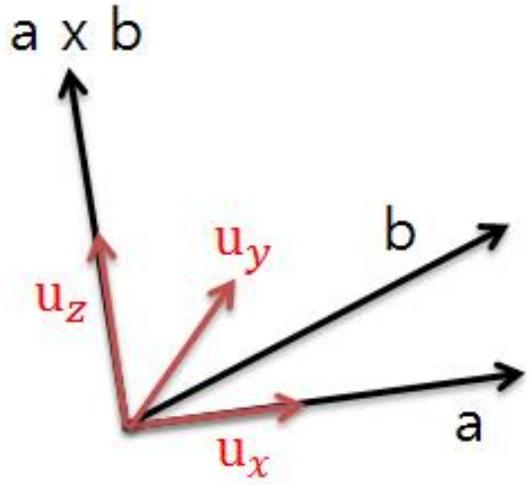

Figure 5.2. The locally absolute coordinate frame construction using two vectors.

In our research, we use the x- and y-coordinate magnetic field measurements to build the x-coordinate axis in the locally absolute coordinate frames because these two vectors can be one of the coordinate axes as corresponding to the reference direction. So, the a vector can be defined as $[m_x, m_y, 0]$ and unit vector of a is expressed as

$$u_a = u_x = \frac{[m_x, m_y, 0]}{\sqrt{m_x^2 + m_y^2}} = [m'_x, m'_y, 0] \tag{5.4}$$

For the b vector, we use the gravitational measurements from the embedded sensor in the cell phone and it is defined as $[g_x, g_y, g_z]$. The unit vector of b can be expressed as

$$u_b = \frac{[g_x, g_y, g_z]}{\sqrt{g_x^2 + g_y^2 + g_z^2}} = [g'_x, g'_y, g'_z] \tag{5.5}$$

The cross product operation between unit vector of a and b that is corresponded to the z-coordinate axis in the locally absolute coordinate frames is expressed as

$$u_{a\times b} = u_z = \frac{\begin{vmatrix} i & j & k \\ m'_x & m'_y & 0 \\ g'_x & g'_y & g'_z \end{vmatrix}}{\|[m'_x m'_y 0] \times [g'_x g'_y g'_z]\|} \tag{5.6}$$

$$= \frac{m'_y g'_z i - m'_x g'_z j + (m'_x g'_y - m'_y g'_x)k)}{\sqrt{(m'_y g'_z)^2 + (-m'_x g'_z)^2 + (m'_x g'_y - m'_y g'_x)^2}} \tag{5.7}$$

$$= \frac{(m'_y g'_z, -m'_x g'_z, (m'_x g'_y - m'_y g'_x)]}{\sqrt{(m'_y g'_z)^2 + (-m'_x g'_z)^2 + (m'_x g'_y - m'_y g'_x)^2)}} \tag{5.8}$$

Finally, the y-coordinate axis in the locally absolute coordinate frames, which is perpendicular to both u_x and u_z, simply expresses as $u_z \times u_x$ using 5.4 and 5.6 equations. As based on the results of finding three unit vectors in the locally absolute coordinate frames, each unit vector has a 3×1 matrix and total unit vectors can be expressed as 3×3 matrices.

$$\begin{bmatrix} u_x \\ u_y \\ u_z \end{bmatrix} = \begin{bmatrix} u_{xx'} & u_{xy'} & u_{xz'} \\ u_{yx'} & u_{yy'} & u_{yz'} \\ u_{zx'} & u_{zy'} & u_{zz'} \end{bmatrix} \tag{5.9}$$

Where x, y and z represent the x-, y- and z-coordinate axis of unit vector components in the locally absolute coordinate frames. Having three unit vectors, we then combine these three unit vectors with gravitational measurements to find the accurate gravity vectors in the locally absolute coordinate frames. The gravitational vectors in the locally absolute coordinate frames are expressed as matrix below

$$g_{locally\ absolute} = g_{sensor} \cdot \begin{bmatrix} u_x \\ u_y \\ u_z \end{bmatrix} = [g_x, g_y, g_z] \cdot \begin{bmatrix} u_{xx'} & u_{xy'} & u_{xz'} \\ u_{yx'} & u_{yy'} & u_{yz'} \\ u_{zx'} & u_{zy'} & u_{zz'} \end{bmatrix} = [G_x\ G_y\ G_z] \tag{5.10}$$

Where,

G_x represents the x-coordinate axis of gravity in the locally absolute coordinate frame

G_y represents the y-coordinate axis of gravity in the locally absolute coordinate frame

G_z represents the z-coordinate axis of gravity in the locally absolute coordinate frame

Having the integration of rotation matrix with the gravitational vectors in the locally absolute coordinate frames in chapter 4, it is available to compare to the local standard gravity and find the angles of inclination in user's path way. It is also appropriate for testing both indoor and outdoor environments in our research.

5.2 Gravity

In the history of gravity, gravity has been introduced in Galileo Galileis experiment of dropping balls from the Tower of Pisa. For planets on the galaxy, they are surrounded by own gravitational field that exerts attractive force on all objects. The standard numerical gravity value, which is defined by the International Bureau of Weights and Measures under SI, is $g = 9.80665 m/s^2$ without consideration for the air resistances, latitudes, surface features and densities. Since we develop to build a topographical map on the Earth's shape, it is necessary to find the precise value of gravity with distance from equator on specific location. So, we use the theoretical equations of International Gravity Formula 1980 [18] and the Free Air Correction to calculate the accurate value of gravity as a function of latitude and height above sea level.

$$g = 9.780327(1 + A\sin^2 L - B\sin^2 2L) - 3.086 \times 10^{-6} H \tag{5.11}$$

Where,

$$\text{A} = 0.0053024$$
$$\text{B} = 0.0000058$$
$$\text{L} = \text{latitude(meter)}$$

H = height above sea level (meter)

According to U.S. climate data [19], we can obtain average values of latitude (40.475) and height above sea level (712 feet) in West Lafayette and we plug these values into equation above. We then obtain the local gravity value ($9.80145 m/s^2$) in West Lafayette to be used as the reference gravity value in our research. The local gravity value is compared to the gravitational measurements from the embedded sensor in the cell phone and find the angles of inclination.

5.3 Gravity Analysis

In the gravity analysis, we express the local standard gravity as 3×1 matrices and it acts on the vertical direction in the locally absolute coordinate frames. The embedded sensor in the cell phone collects the user's gravitational vectors in the 3-dimensional coordinate frames and it can be expressed as

$$g_{reference} = [0\ 0\ g]^T \tag{5.12}$$

$$g_{sensor} = [g_x\ g_y\ g_z]^T \tag{5.13}$$

In the inclined plane in physics, there is the key to solve the problems which is related to inclined or declined plane. The forces acting on the objects that are parallel to the surface and the objects are in contact with the plane. In our research, we use the idea for analyzing the forces acting on object to the inclined surface. We assume that the angle between the vertical gravitational measurement from embedded sensor and the local standard gravity is same as the angle of inclination in the user's pathway as shown in figure 2.3. In order to find the angle between the vertical gravitational vector and the local standard gravitational vector, the mathematical equation is used with these two vectors in 3-dimensional spaces which can be expressed as

$$\cos\theta = \frac{v \cdot w}{|v| \cdot |w|} \tag{5.14}$$

Where,

v = the vertical gravitational vector from the embedded sensor

w = the local standard gravitational vector

θ = the angle between the vertical gravitational and the local standard gravitational vector

As we re-write above equation with the gravitational vectors,

$$\cos\theta = \frac{[0\ 0\ g]^T \cdot [g_x\ g_y\ g_z]^T}{|\sqrt{g^2}| \cdot |\sqrt{g_x^2 + g_y^2 + g_z^2}|} \tag{5.15}$$

Since we assume that the magnitude of gravitational measurement vectors is same as the magnitude of the local standard gravitational vector, the equation above can be simplified as

$$\cos\theta = \frac{g \cdot g_z}{|g| \cdot |g|} = \frac{g \cdot g_z}{|g|^2} \tag{5.16}$$

$$\theta = \cos^{-1} \frac{g \cdot g_z}{|g|^2} \tag{5.17}$$

5.4 Construct Topographical Map

Having all analysis of position estimation and the angles of inclination, we then develop the algorithms of constructing topographical map. Since we develop to find the position estimation in the horizontal and lateral coordinate frames, the position estimation in the vertical coordinate axis is also available to find using the angles of inclination. We use the idea from the way to find slope in the 3-dimensional space [20],

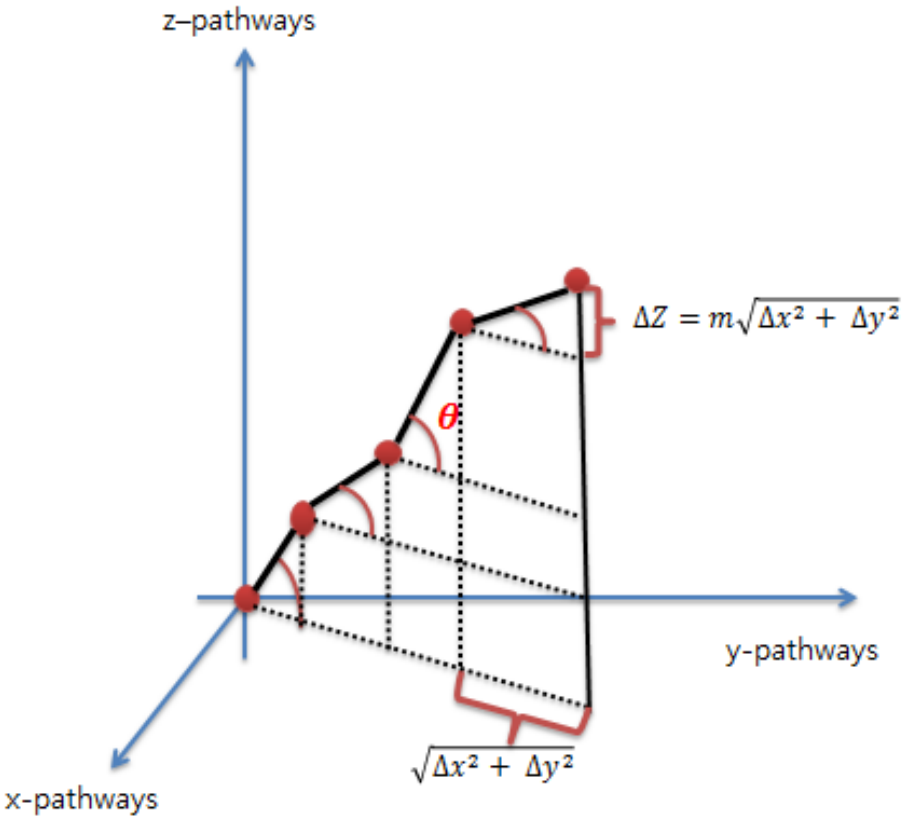

Figure 5.3. Example of slope in 3-dimensional space.

$$\tan\theta = slope(m) = \frac{\Delta z}{\sqrt{\Delta x^2 + \Delta y^2}} \tag{5.18}$$

$$\Delta z = \tan\theta \times \sqrt{\Delta x^2 + \Delta y^2} \tag{5.19}$$

Since we have a discrete time system and collect the step occurrences to be path node, the difference between each step indicates the path intervals. We then develop algorithms that z-directional path intervals will be accumulated and it can be visualized as topographical map. We can express above equations as discrete system modelling below,

$$z_{t+1} = z_t + \sqrt{(x_{t+1} - x_t)^2 + (y_{t+1} - y_t)^2} * \tan(\theta_{t+1} - \theta_1) \tag{5.20}$$

Where,

x_{t+1} = The updated position estimation in the x-coordinate axis

y_{t+1} = The updated position estimation in the y-coordinate axis

z_{t+1} = The updated position estimation in the z-coordinate axis

θ_{t+1} = The updated angle between z-directional gravitational vector and the local standard gravitational vector

θ_1 = The initial angle between z-directional gravitational vector and the local standard gravitational vector

6. EXPERIMENTAL RESULTS FOR VARIOUS TOPOGRAPHIES

Having all the algorithms of data analysis, stride detection, the Kalman filtering analysis, compensation of rotation effect, gravity analysis, the angles of inclination and map construction, we set up series of tests including flat surface area, inclined surface area, declined surface area and outdoor environment in this chapter. In order to test for convenience, we use an instrumented treadmill which can adjust the walking speed, angle of inclination and measure times and total distances of walking. For outdoor environment, we select the location of walking paths and test our methodology. The results will be compared to GPS location elevation measurement to verify our methods.

Figure 6.1. Instrumented treadmill for testing, Precor 966i Experience.

6.1 Flat Surface Area

Before we start to test, the embedded sensor in the cell phone needs to be stable at a few seconds and begins to collect measurements. We then set up the distance of pathway, put the cell phone inside shoes and measure sensor data for flat surface area. Figure 6.2 and 6.3 illustrate that the angle of inclination at each stride point and histrogram of angle at each stride point. Figure 6.4 and Table 6.1 show that the result from the test for flat surface area.

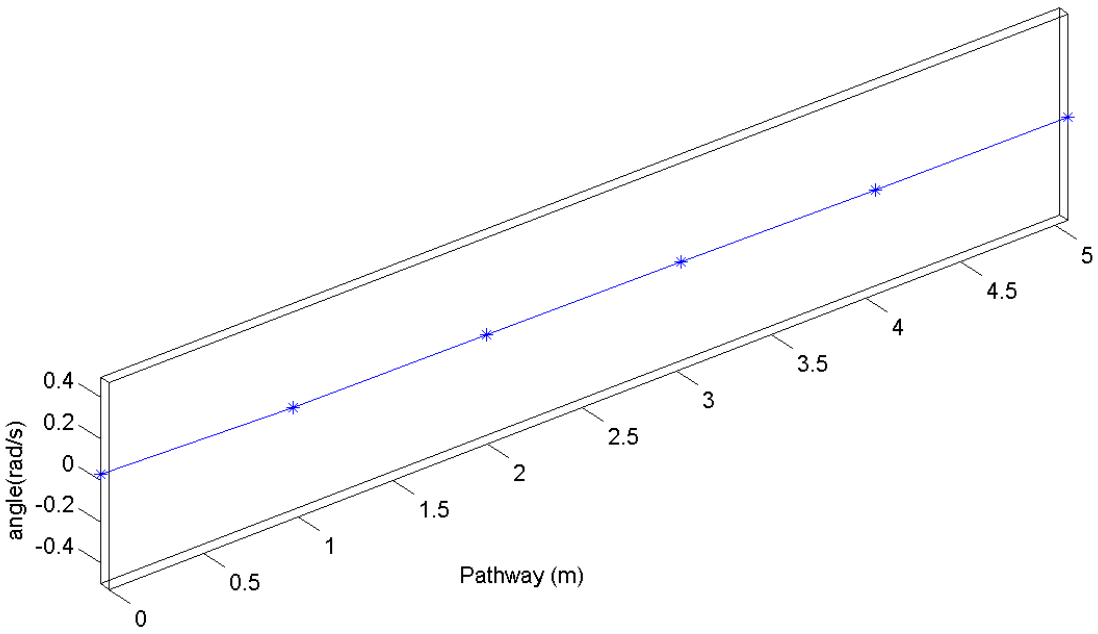

Figure 6.2. Angle of inclination at each point for flat surface area.

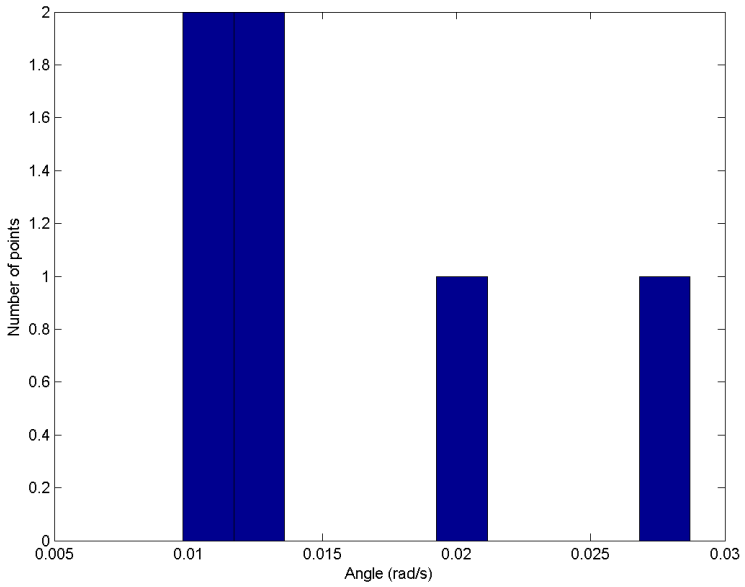

Figure 6.3. Histogram of angle at each point for flat surface area.

From histogram chart, each angle at stride point can be compared with reference value that is assumed as zero degree. We then find the average angle of inclination for testing and calculate the maximum height from flat surface compared to reference value.

Table 6.1. The results of testing for flat surface area.

	Testing	**Reference**	**Difference**
Average value of Inclination of angle(Degree)	0.896	0.0	0.896
Max height from flat surface (meters)	0.078	0.0	0.078

According to the results of testing for flat surface area, the average value of inclination of angle is 0.896 degree and the maximum height from flat surface is 0.078 meter. As shown in figure 6.4, The shape of topographical map looks almost flat and this result can be satisfied with the reference shape model that we need to obtain.

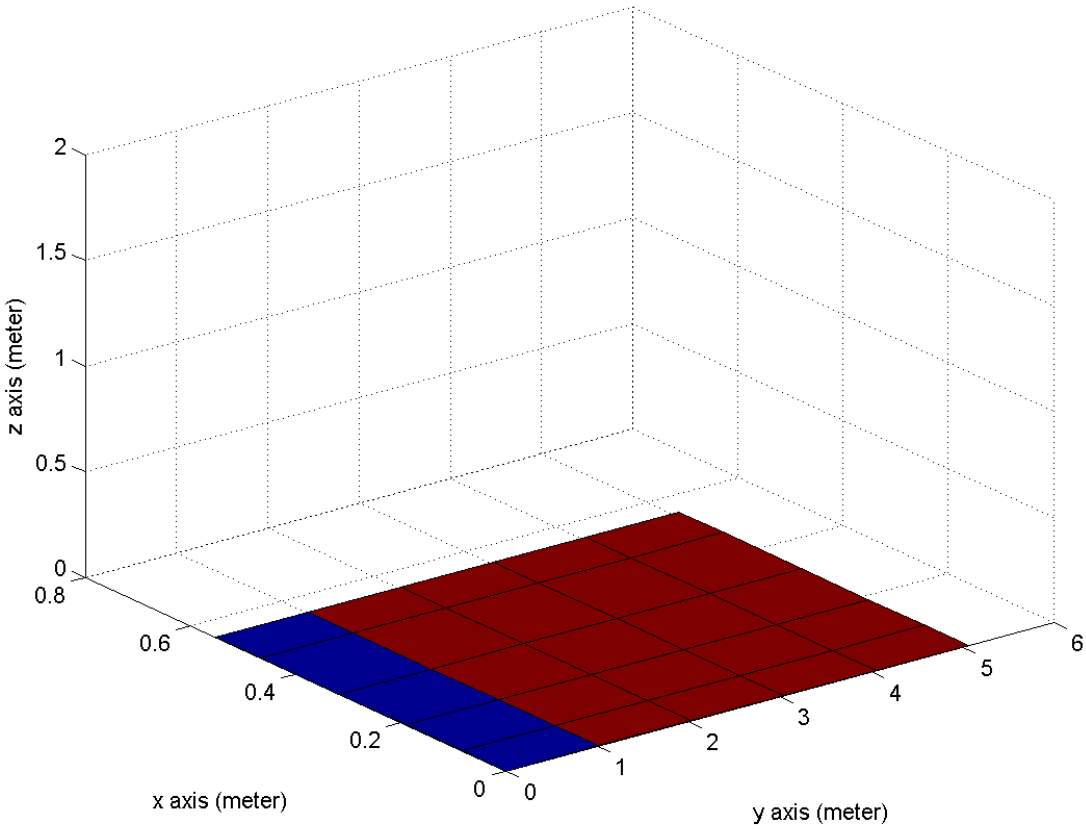

Figure 6.4. Topographical map for flat surface area.

6.2 Inclined Surface Area

In order to test for inclined surface area, we firstly set up the 2.5 miles per hour for walking speed and 10 meters pathways on the treadmill which is installed in Corec-gym at Purdue University. We then set up the 15 percentage for angle of inclination, which is same as 8.53 degree (0.149 rad/s) and collect sensor measurements. The result for inclined surface area is shown below.

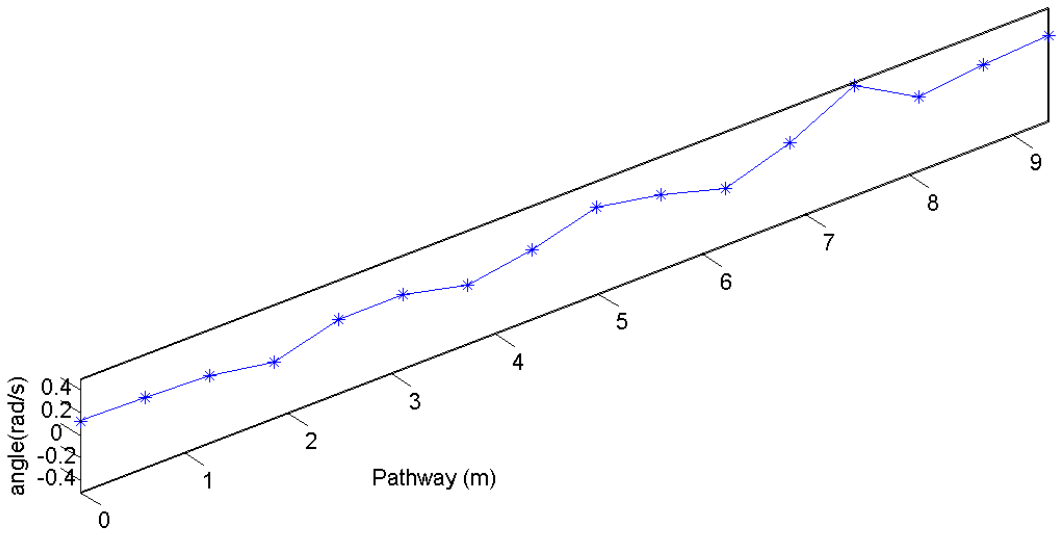

Figure 6.5. Angle of inclination at each stride point for inclined surface area.

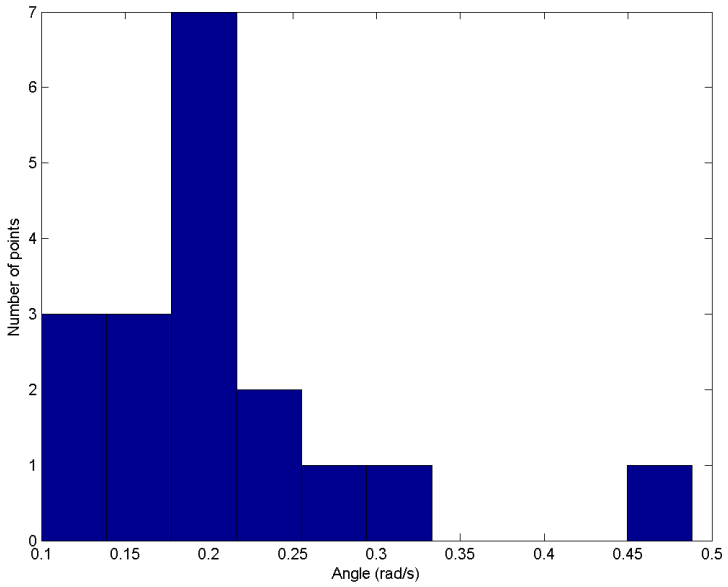

Figure 6.6. Histogram of angle at each point for inclined surface area.

From histogram chart for inclined surface area, each angle at stride point is compared with reference value that is assumed as 8.53 degree (0.149 rad/s). We then calculate the average angle of inclination from testing and find the maximum height from flat surface that will be compared to reference value.

Table 6.2. The results of testing for inclined surface area.

	Testing	**Reference**	**Difference**
Average value of Inclination of angle(Degree)	8.67	8.53	0.14
Max height from flat surface (meters)	1.4243	1.4093	0.015

According to the result of testing for inclined surface area, the average value of inclination of angle is 8.67 degree and the maximum height from flat surface is 1.4243 meters. As compare to reference values, the differences for average angle and maximum height from flat surface are 0.14 degree and 0.015 meters, respectively. So, we can obtain a good shape model of topographical map for inclined surface area.

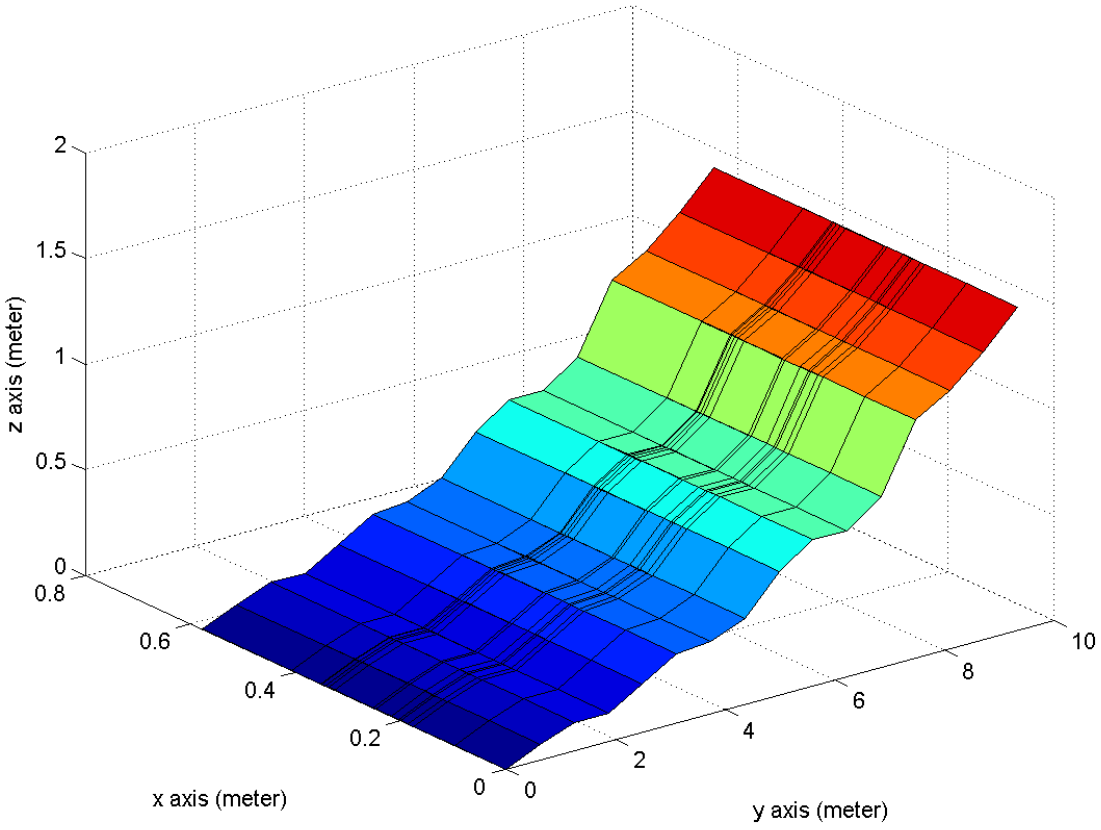

Figure 6.7. Topographical map for inclined surface area.

6.3 Declined Surface Area

In order to collect sensor measurements for declined surface area, we set up the 2.5 miles per hour for walking speed and 10 meters pathways on the treadmill which is as same as the testing for inclined surface area. We then adjust the 15 percentage for angle of inclination and test backward walking (- 8.53 degree). The result for declined surface area is shown below.

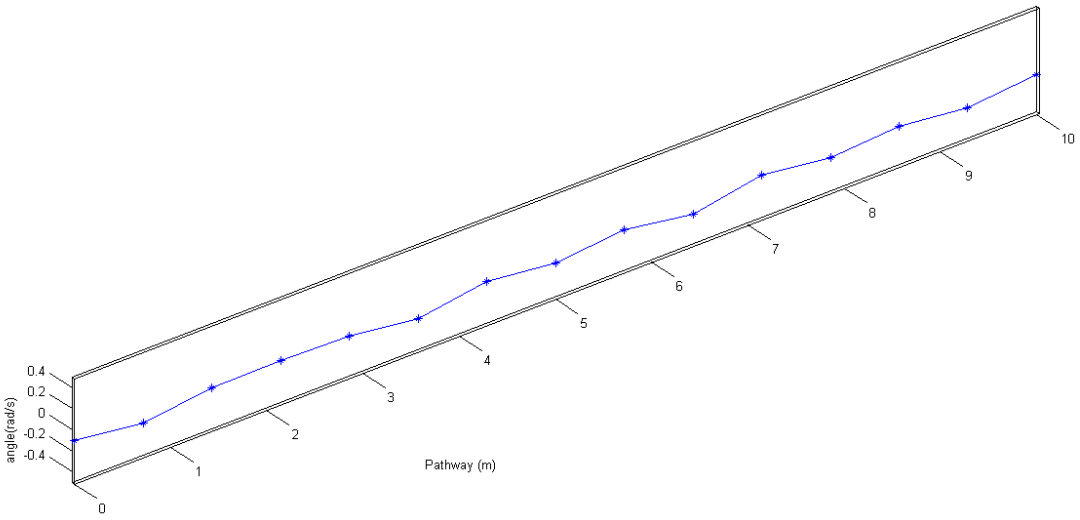

Figure 6.8. Angle of inclination for declined surface area.

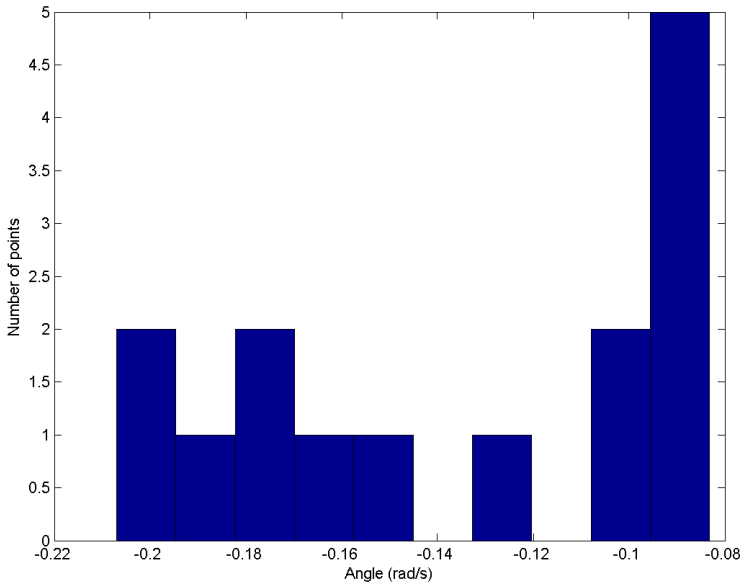

Figure 6.9. Histogram of angle at each point for declined surface area.

From histogram chart for declined surface area, each angle at stride point is compared with reference value that is assumed as - 8.53 degree (- 0.149 rad/s). We then calculate the average angle of inclination from testing and find the maximum height from flat surface that will be compared to reference value.

Table 6.3. The results of testing for declined surface area.

	Testing	**Reference**	**Difference**
Average value of Inclination of angle(Degree)	-7.75	-8.53	0.78
Max height from flat surface (meters)	-1.4046	-1.4093	0.0047

From the result of testing for declined surface area, the average value of inclination of angle is -7.75 degree and the maximum height from flat surface is -1.4046 meters. As compared to reference, there are slightly different between testing and reference. Therefore, we can obtain the accurate topographical map for declined surface area as same as a series of test for flat and inclined surface area.

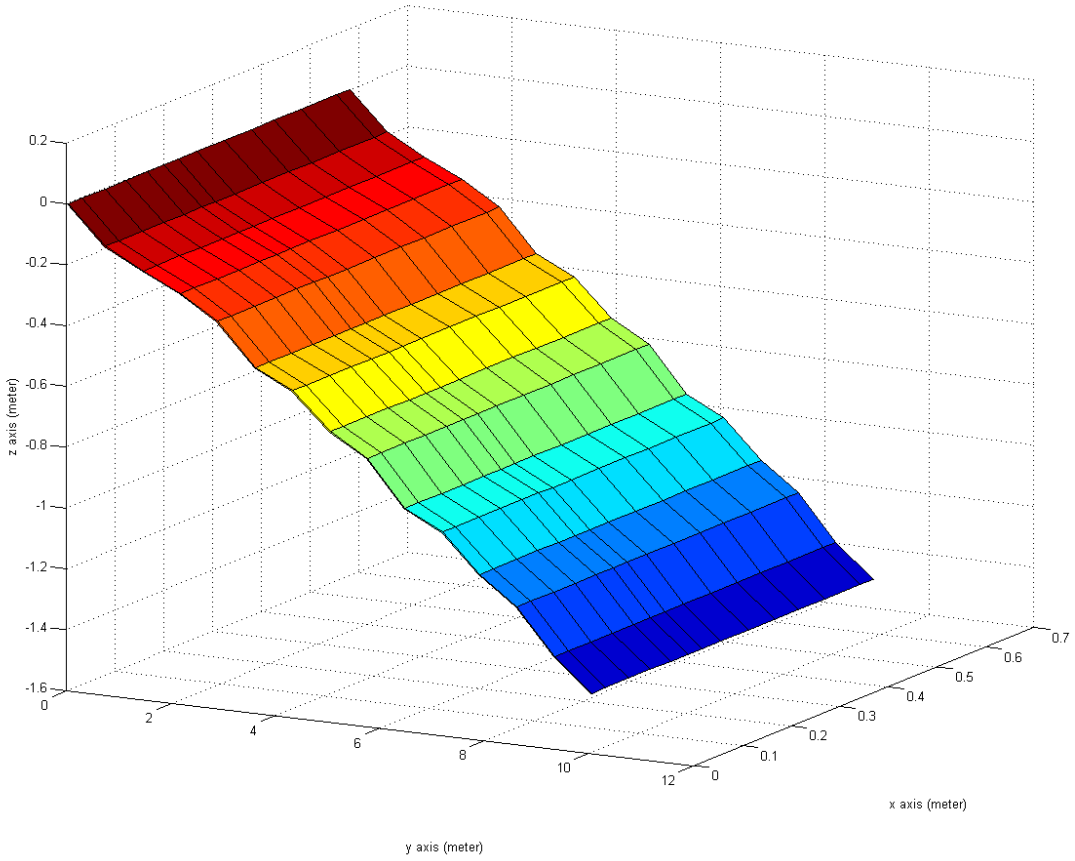

Figure 6.10. Topographical map for declined surface area.

6.4 Outdoor Test

Based on the results from a series of test established in the flat, inclined and declined surface area using an instrumented treadmill, we then test outdoor environment in order to verify our methodology and compare to GPS elevation measurements. We select the location of the pathway in Tower Drive at Purdue University which has a length of 80 meters in one direction with curve and it shows in figure 6.11. Having the algorithms of constructing topographical map, we use the elevation of position estimation from the GPS location measurements to compare our topographical map as shown in figure 6.14. The main results from outdoor test are listed in table 6.4.

Figure 6.11. Tower drive at Purdue University.

According to figure 6.12, we estimate the user's walking direction using Kalman filtering analysis in chapter 4. To analyze the walking direction specifically, we show the 2- and 3-dimensional spaces. As compared the results with pathway in the real-test, our methodology of estimating a walking direction is a quite correspondence. Figure 6.13 shows the topographical map for outdoor environment.

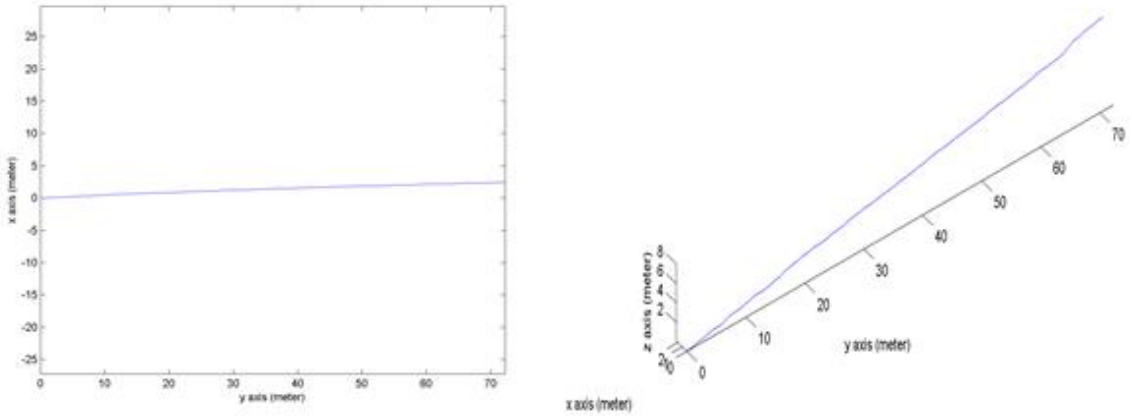

Figure 6.12. The walking direction in 2-dimensional and 3-dimensional spaces.

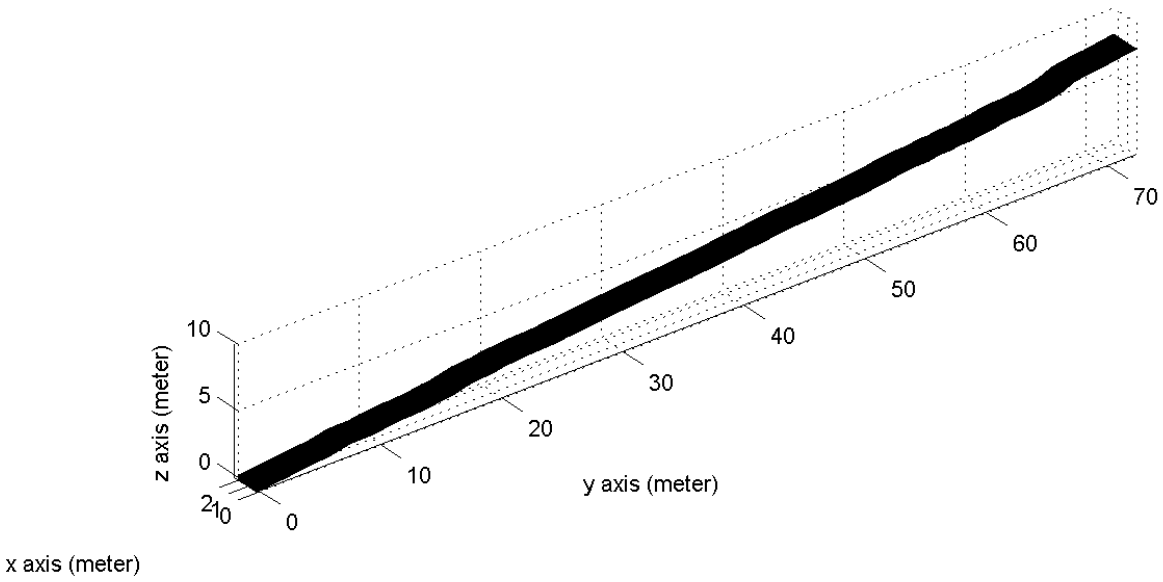

Figure 6.13. Topographical map for outdoor test.

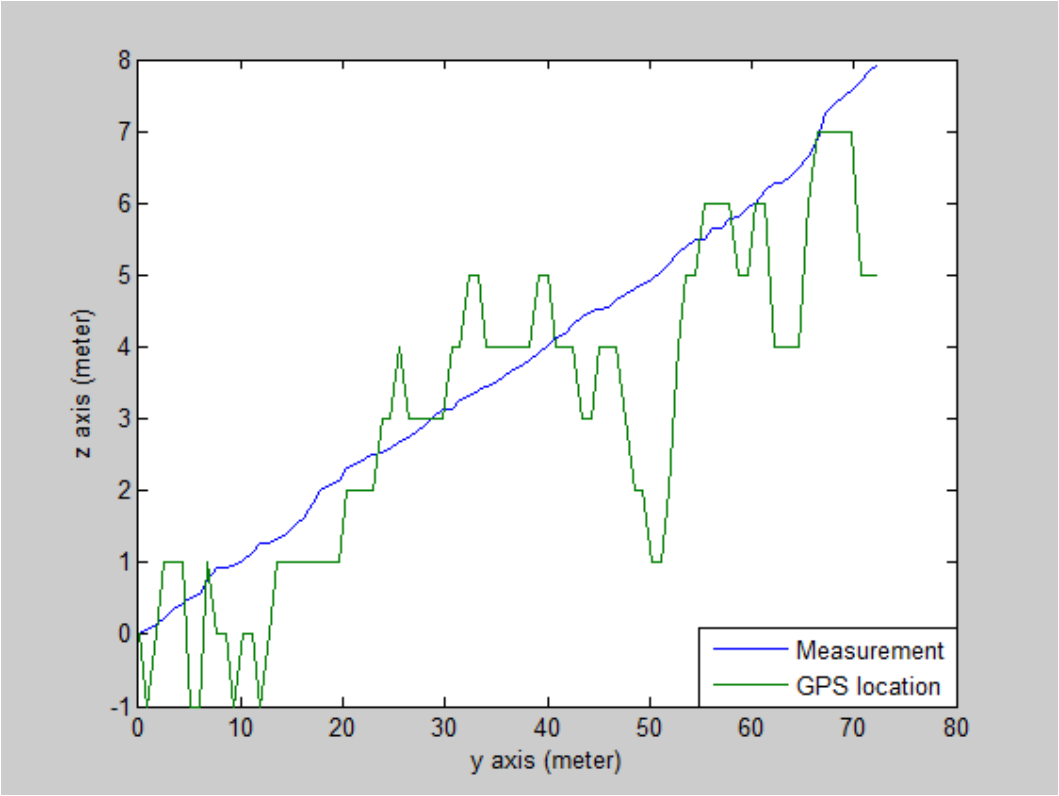

Figure 6.14. Comparison between maps and GPS location.

Having the results from Figure 6.14 and Table 6.4, there is difference between the topographical map and GPS location elevation. For average slope of pathway, the difference between map and GPS is 0.6282. For the maximum and minimum length of elevation, the differences between testing and GPS location elevation are 0.9338 meter and 1 meter, respectively. Since the GPS is hard to find the accurate position estimation at every step occurrences, we can use the overall slope performance in pathway, maximum and minimum length of elevation to compare the results of testing with GPS location measurements.

Table 6.4. The results of testing for outdoor environment.

	Testing	**GPS location elevation**	**Difference**
Average slope of pathway	3.7096	3.0814	0.6282
Maximum height of elevation (meter)	7.9338	7	0.9338
Minimum height of elevation (meter)	0	-1	1

7. CONCLUSIONS AND RECOMMENDATIONS

7.1 Conclusions

In this research, we proposed the novel algorithms to construct a topographical map for cellphones geolocation when GPS is spotty. We firstly collected the IMUs and RSSI measurements from the embedded sensor in the cell phone. We then used truncation method to obtain the vertical accleration measurements in steadily walking period without the sensitive noise. Moreover, we used the fast fourier transform and low pass filtering to obtain more clear measurements which are corresponded to the impact values of users step. We then analyzed the stride detection of the users walking path as based on the relationship between stride length and stride interval. We also applied the Kalman filter and rotation matrix to remove random noise in the vertical gyroscope measurements and estimate the users overall walking direction. We also developed the position estimation using interval analysis and dynamic estimation to reduce intersection of regions and fluctuations. Having a gravity analysis and construction of the locally absolute coordinate frames, topographical map was constructed. After the process of testing various cases such as flat, inclined, declined and outdoor environment, our methodology was verified and augmented the results from pedometry and magnetic mapping to estimate geolocation more precisely.

7.2 The Error Analysis

From the methodology of constructing a topographical map, there is still existed the estimation error. The one of main reasons for propagating the estimation error in our research comes from the relationship between the stride intervals and stride lengths. Since the relationship is expressed as the power function with the numerical variables, it is hard to estimate all steps using our power function. Another reason

for estimation error is from data analysis. The measurements from the sensor are not perfectly truncated. Therefore, redundant can still be remained. Even though we use low pass filter to estimate the vertical accleration measurements precisely, noise can be existed for high frequencies. Detecting by peak values can occur errors in this system. Although the methodology of detecting by peak values is filtered by threshold and it is fitted for larger stride length in real test, peak values are not perfectly corresponded to step occurrences. Lastly, the oscillation and shaking of cell phone in the shoes can produce some unpredictable errors.

7.3 Future Recommendations

As based on the current results, there are some issues that are going to be developed in the future. Since uncertainties can be accumulated along the long distance of pathway, our methodology for constructing a map is not available to find the exact position estimation alone. We will extend to fuse our methodology with the magnetic mapping algorithms so that topographical map can be constructed more precisely even if the pathway is uneven surface. Secondly, we apply our methodology to more various tests such as different walking speed and different walking paths. We can also try to use different IMUs on the sensor and compare with current results. Having the novel algorithms of the constructing topographical map, we can develop to make a sensor chip which aids to apply measurements such as the radiation effect from the user's walking paths.

LIST OF REFERENCES

LIST OF REFERENCES

[1] Bajaj, R., Ranaweera, S. L.,& Agrawal, D. P. (2002). GPS: location-tracking technology, *Computer, 4*, 9294.

[2] Rappaport, T.(1996). GWireless communications, principles and practice. *IEEE Press, Prentice Hall.*

[3] Sayed, A. H., & Tarighat, A.(2005). Network-based wireless location, challenges faced in developing techniques for accurate wireless location information. *IEEE Signal Processing Magazine 22(4)*,2440.

[4] Bekkali,A., Sanson,H., & Matsumoto,M.(2007). RFID indoor positioning based on probabilistic RFID map and Kalman filtering. *Proceedings of IEEE International Conference on Wireless and Mobile Computing, Networking and Communications* (pp.21-27).

[5] Cui, Y., An, R., & Ariyur, K. B. (2015). Cellphone geolocation via magnetic mapping. *Automatica,51* 70-79.

[6] Jaulin, L., Kieffer, M., Didrit, O., & Walter, E. (2001). Applied interval analysis, with examples in parameter and state estimation. *robust control and robotics*, Springer.

[7] Van Loan, C. (1992). Computational Frameworks for the Fast Fourier Transform. SIAM.

[8] Cooley, J.W., Tukey, J.W.(1965). An algorithm for the machine calculation of complex Fourier series. *Mathematics of computation 19.90*, 297-301.

[9] Mersereau, R.M., and Speake, T.C.(1981). A unified treatment of Cooley-Tukey algorithms for the evaluation of the multidimensional DFT. *Acoustics, Speech and Signal Processing, IEEE Transactions on 29.5*, 1011-1018.

[10] Heimo H., Lazansky, J., & Marik, V.(1995). *Information Management in Computer Integrated Manufacturing*, 220.

[11] Kuo, A.D.(2001). A Simple Model of Bipedal Walking Predicts the Preferred Speed-Step Length Relationship. *Journal of Biomechanical Engineering*, Vol.123.

[12] Cui, Y., & Ariyur, K.B.(2011). Bounding inertial drift with human gait dynamics for personal navigation. *Proceedings of IEEE International System Conference*, Montreal, Canada (pp.28-33).

[13] Kalman, R. E.(1960). A New Approach to Linear Filtering and Prediction Problems. *Journal of Basic Engineering 82 (1)* 3545

[14] Slabaugh,G.G.(1999). Computing Euler angles from a rotation matrix. 39-63.

[15] Moore, R.E., Kearfott, R.B.& Cloud, M.J. (2009). Introduction to interval analysis. SIAM.

[16] Kieffer, M., Jaulin, L., Walter, E. (2000).Robust autonomous localization using interval analysis. *Reliable Computing, 6(3)*,337362.

[17] Widnall, S. & Bonnifait,Ph.(2009). Lecture L3- Vectors. Matrices and Coordinate Transformations. *Dynamics*

[18] Hinze, W.J., Von Frese, R.R., & Saad, A.H.(2013). *Gravity and Magnetic Exploration: Principles, Practices, and Applications. Cambridge University Press. p. 130.*, ISBN 978-1-107-32819-8.

[19] U.S. climate data.(2015). Map of West Lafayette - Indiana. Retrieved from http://www.usclimatedata.com/map.php?location=USIN0707

[20] Clapham, C & Nicholson, J. (2009). Oxford Concise Dictionary of Mathematics. *Gradient. Addison-Wesley*, p.348.

APPENDIX

Appendix: Algorithms of building topographical map

```
%% Read sensor measurements on the cellphone

dataSet = csvread('sensor_data.csv'); % read through data
t0 = dataSet(1,1); % initial time is not zero in the file.
figure(1) % check the plot
plot(dataSet(:,1)-t0,dataSet(:,4),'b')
legend('z-coordinate of Accelerometer')
xlabel('Time (s)')
ylabel('Acceleration(m/s^2)')


%% Sampling freq is 50 Hz, and after truncation

dataSetTruncated = dataSet(1000:3000,:);
figure(2) % check the plot
plot(dataSetTruncated(:,1)-t0,dataSetTruncated(:,4),'b')
title('Acceleration with respect to Truncated time')
legend('Z-coordinate of Acceleration')
xlabel('Time (s)')
ylabel('Acceleration(m/s^2)')


%% FFT analysis of the z directional signal

az = dataSetTruncated(:,4)-9.8;
Fs = 50; % sampling freq
NFFT = 2^nextpow2(length(az)); % Next power of 2 from length of z
Z = fft(az,NFFT)/length(az); % fft process
```

```
f = Fs/2*linspace(0,1,NFFT/2+1);

%% Plot single-sided amplitude spectrum

figure(3)
plot(f,2*abs(Z(1:NFFT/2+1)))
% title('Single-Sided Amplitude Spectrum of a_z')
xlabel('Frequency (Hz)')
ylabel('|Z|')

%% Low pass filter with a Fs = 50Hz, fpass = 5Hz, fstop = 10Hz.

t = dataSetTruncated(:,1)-t0;
t = t-t(1);
order = 4;
cutoff_freq = 2;
passband_peak_to_peak_db = 0.5;
stopband_attenuation = 20;
[B,A] = ellip(order,passband_peak_to_peak_db, stopband_attenuation,
cutoff_freq/(0.5*Fs),'low');
azFiltered = filter(B,A,az);
figure(4) % filtered ay signal
plot(t,azFiltered,'r',t,az,'b')
title('azFiltered vs DataSetTruncated without offset')
xlabel('Time (s)')
ylabel('Acceleration(m/s^2)')

%% Stride detection

azThreshold = 2; % this is pre-examined from the figure
condition = -5; % input argument, detecting from starting point
stridePointers = strideDetection(-azFiltered,azThreshold,condition);
strideTimes = [];
stridePeaks = [];
for i = 1:length(stridePointers)
```

```
strideTimes(i) = t(stridePointers(i));
stridePeaks(i) = azFiltered(stridePointers(i));
end
figure(5)
plot(t,azFiltered,'b',strideTimes,stridePeaks,'ro')
legend('azFiltered vs StrideTimes', 'StridePeaks')
xlabel('Time (s)')
ylabel('Acceleration(m/s^2)')

%% Walking direction check, based on w_x, the rotation rate on
x direction with Kalman Filter

wz = dataSetTruncated(:,9); % angular rate z
thetaz = zeros(501,1); % Kalman filter implementation
Phi = [1 0.02;0 1]; % state
Hk = [1 0;0 1];     % state
Qk = [0.004 0;0 1]; % state
Rk = [0.004 0;0 4]; % state
Pk = [0.01 0;0 4];  % state
zk = [thetaz';wz']; % state
xkstate = [0;0];
wzstate = [];
i = 1;
while i <501
    xk = Phi*xkstate;
    Pk = Phi*Pk*Phi'+Qk;
    Sk = Hk*Pk*Hk'+Rk;
    Kk = Pk*Hk*inv(Sk);
    xkstate = xk+Kk*(zk(:,i)-Hk*xk);
    wzstate(:,i) = xkstate;
    Pk = (eye(2,2)-Kk*Hk)*Pk;
    i = i+1;
end

%% Position estimation via stride model.
```

```
%%Stride model parameters

Cinterval = infsup(-0.009851,0.06259);
C = -2.103;
bindex = 0.02637;
C1interval = infsup(0.6483, 0.8184);
sigmaAccx = 0.5; % acc x
sigmaAccz = 0.5; % acc z

%%Stride update, initial set-up

pathInterval(:,1) = [infsup(0,1);infsup(-0.5,0.5)];
fusionInterval(:,1) = pathInterval(:,1);
positionXZ = [infsup(0,1);infsup(-0.5,0.5)];

%% Sensor fusion with IMUs and Dynamic process

accx = dataSetTruncated(:,3);
accz = dataSetTruncated(:,5);
for i = 2:length(stridePointers) % scan all stride detections
dtInterval = t(stridePointers(i))-t(stridePointers(i-1));
dsInterval = Cinterval*dtInterval^bindex+C1interval; % ds update
rotate2D = [cos(wzstate(1,stridePointers(i))),
-sin(wzstate(1,stridePointers(i)));...

sin(wzstate(1,stridePointers(i))),
cos(wzstate(1,stridePointers(i)))];
pathInterval(:,i) = pathInterval(:,i-1)+2*[dsInterval*
[cos(wzstate(1,stridePointers(i))),...
sin(wzstate(1,stridePointers(i)))]]'; %iteration
velocityInterval = pathInterval(:,i)-pathInterval(:,i-1);
for j = stridePointers(i-1):stridePointers(i)
accInterval = [infsup(accx(j)-3*sigmaAccx,accx(j)+3*sigmaAccx);...
infsup(accz(j)-3*sigmaAccz,accz(j)+3*sigmaAccz)];
positionXZ = positionXZ+0.5*accInterval*(t(j)-t(j-1))^2+
```

```
velocityInterval*(t(j)-t(j-1));
end
[crossing,adjacent,faraway] = IntervalJudge(real(positionXZ),
real(pathInterval(:,i))); % check crossing
if crossing == 1 % if crossing is non-empty, then use the intersection
fusionInterval(:,i) = intersect(real(positionXZ),real(pathInterval(:,i)));
positionXZ = intersect(real(positionXZ),real(pathInterval(:,i)));
intersectionFlag = 1;
else % otherwise, use initial estimations
fusionInterval(:,i) = real(pathInterval(:,i));
positionXZ = real(pathInterval(:,i));
intersectionFlag = 0;
end
end
figure(6)
plot(real(pathInterval(1,:)),real(pathInterval(2,:)))
axis equal
xlabel('x(meter)')
ylabel('y(meter)')

%% Map generation, firstly by stride model, with estimation of nodes.

pathNodes = [];
pathNodes(:,1) = [0;0];
C = -2.103;
C1 = 0.7333;


%% Construct Global frames

x = (dataSetTruncated(:,19)); %x-component of gravity vector
x_1 = length(x);
y = (dataSetTruncated(:,21)); %y-component of gravity vector
z = (dataSetTruncated(:,20)); %z-component of gravity vector
a = power(x,2); %x^2
```

```
b = power(y,2); %y^2
c = power(z,2); %z^2
scalarg =sqrt(a+b+c); %sqrt(x^2+y^2+z^2)
gvector = -[x,y,z]; % gravity vector
m_x = (dataSetTruncated(:,11)); %x-component of magnetic field vector
m_x_1 = length(m_x);
m_y = (dataSetTruncated(:,13)); %y-component of magnetic field vector
m_z = (dataSetTruncated(:,12)); %z-component of magnetic field vector
a_1 = power(m_x,2); %mx^2
b_2 = power(m_y,2); %my^2
scalarm = sqrt(a_1+b_2); %sqrt(mx^2+my^2)
magvector = [m_x,m_y,repmat(0,m_x_1,1)]; % magnetic field vector

u_1 = repmat(scalarg,1,3); % denominator of unit vector g
v_1 = repmat(scalarm,1,3);% denoninator of unit vector m

u = rdivide(gvector,repmat(scalarg,1,3)); % unit vector g
v = rdivide(magvector,repmat(scalarm,1,3)); % unit vector m

v_x = repmat(v(:,1),1,3); % x component of unit vector of magnetic field
v_y = repmat(v(:,2),1,3); % y component of unit vector of magnetic field
v_z = repmat(v(:,3),1,3); % z component of unit vector of magnetic field

cros = cross(u,v); %cross product of unit vector g and m
cros_x = cros(:,1); % x-component of cross product
cros_y = cros(:,2); % y-component of cross product
cros_z = cros(:,3); % z-component of cross product
cros_x_2 = power(cros_x,2); %x^2 of cross product
cros_y_2 = power(cros_y,2); %y^2 of cross product
cros_z_2 = power(cros_z,2); %z^2 of cross product
mag_cros = sqrt(cros_x_2 + cros_y_2 + cros_z_2); %magnitude
cros_unit = rdivide(cros,repmat(mag_cros,1,3)); % unit vector

x_global = v; % x-component of unit vector
z_global = cros_unit; % z-component of unit vector
```

```
y_global = cross(x_global,z_global); % y-component of unit vector

new_g = dot(gvector,x_global,3) + dot(gvector,y_global,3) +
dot(gvector,z_global,3);
new_g_matrix = [dot(gvector,z_global,2)  dot(gvector,x_global,2)
dot(gvector,y_global,2)]; % new g-vector in global frame
new_g_z = dot(gvector,y_global,2);% new z-axis g vector

%% using new gravity vector to find the angle of inclination
g1 = 9.81;
A = [0 0 g1];
A_1 = repmat(A,x_1,1);
B = new_g_matrix;
g2 = sqrt(power(dot(gvector,z_global,2),2)+
power(dot(gvector,x_global,2),2)+power(dot(gvector,y_global,2),2));

n1 = g1*g2;
n2 = dot(A_1,B,2);
theta = acos(rdivide(n2,n1)); % theta in radian
magOnNodes(1) = real(theta(stridePointers(1),1));

for i = 2:length(stridePointers)
dsInterval = C*dtInterval^bindex;
pathNodes(:,i) = pathNodes(:,i-1)
+dsInterval*[cos(wzstate(1,stridePointers(i))),...
sin(wzstate(1,stridePointers(i)))]';
magOnNodes(i) = real(theta(stridePointers(i),1));
end
figure(7)
plot3((((pathNodes(1,:)))),pathNodes(2,:),magOnNodes-magOnNodes(1),'*-');
box on
xlabel('Pathway (m)')
ylabel('PathNode2')
zlabel('angle(rad/s)')
```

```
%% Construct a Topographical Map

for i = 2:length(pathNodes)
S = tan(magOnNodes); %slope of topographical map
R = times(sqrt((pathNodes(2,i)-pathNodes(2,i-1))^2 + (pathNodes(1,i)-
pathNodes(1,i-1))^2),S); % length of z in y interval
Y(1) = R(1);
Y(i) = Y(i-1)+R(i);
Z = repmat(Y,size((pathNodes(1,:)),2),1);
K = repmat(pathNodes(2,:),size(pathNodes(1,:),2),1);
end

%% Check walking direction from z- and y-coordinate frame

figure(22)
plot((pathNodes(1,:)), abs(pathNodes(2,:)));
axis equal
xlabel('y axis (meter)')
ylabel('x axis (meter)')
saveas(gcf,'2dwalkingdirection.png')
figure(23)
plot3(pathNodes(1,:), abs(pathNodes(2,:)),Y);
axis equal
xlabel('y axis (meter)')
ylabel('x axis (meter)')
zlabel('z axis (meter)')
trans_x = transpose(pathNodes(2,:));
trans_y = transpose(pathNodes(1,:));
trans_z = transpose(Y);
trans_mat = [trans_x,trans_y,trans_z];

%% Topographical Map
map_x = pathNodes(1,:);
map_y = abs(pathNodes(2,:));
map_z = Z-Z(1);
```

```
n = surf(map_x,map_y,map_z);
xlabel('y axis (meter)')
ylabel('x axis (meter)')
zlabel('z axis (meter)')

%% GPS location

gps_location = dataSetTruncated(:,29);
figure(9)
plot((pathNodes(1,:)) , gps_location(stridePointers)-
gps_location(stridePointers(1)));

%% Comparison with GPS and measurements

figure(10)
plot((pathNodes(1,:)), Y-Y(1), (pathNodes(1,:)) ,
gps_location(stridePointers)-gps_location(stridePointers(1)))
xlabel('y axis (meter)')
ylabel('z axis (meter)')
legend('Measurement','GPS location')
```

VITA

VITA

Jaeyoung Kim was born in Daejeon, Republic of Korea, in 1989. He received his B.S. in Mechanical Engineering at Purdue University, West Lafayette, IN, USA, in 2014. He has developed the algorithms of estimating cellphone's geolocation on navigation system for two years at Purdue University. He is currently pursuing the Master degree in Mechanical Engineering at Purdue University under Professor Kartik B. Ariyur.