WalkCompass: Finding Walking Direction Leveraging Smartphone's Inertial Sensors

by

Nirupam Roy

Bachelor of Engineering
Bengal Engineering and Science University, Shibpur 2007

Submitted in Partial Fulfillment of the Requirements

for the Degree of Master of Science in

Computer Science and Engineering

College of Engineering and Computing

University of South Carolina

2013

Accepted by:

Srihari Nelakuditi, Major Advisor

Wenyuan Xu, Committee Member

Manton M. Matthews, Graduate Director and Committee Member

Lacy K. Ford, Jr., Vice Provost and Dean of Graduate Studies

Abstract

Determining moving direction with smartphone's inertial sensors is a well known problem in the field of location service. Compass alone cannot solve this problem because smartphone's compass cannot adapt with the orientation change. GPS, works successfully in outdoor, is not suitable in indoor scenario. Another well known approach called dead-reckoning needs to know phones initial orientation and over time it keeps accumulating errors. An ideal system should be able to calculate the heading direction without any user intervention and it should be accurate and light-weight. Moreover in order to be able to work as a generic system, the system should not be restricted by any strict requirement for its function. Considering all the limitations of the conventional solutions and all the requirements of a generic solution, we propose a smartphone based solution called WalkCompass designed especially for pedestrians. Our system focuses on the variation of force during normal human walk and captures this property with the help of the smartphone's inertial sensors. Therefore, the algorithm is inherently free from errors generated by the orientation of the phone. The performance of the system does not depend on the holding style or location of the phone in the body. The algorithm can work fast enough to determine the direction of movement in real time and because of its low complexity, the complete system can be implemented in a regular smartphone. WalkCompass does not need any bootstrapping and can produce results at each step of a walk. The heading direction estimated by WalkCompass has an average error of 6 degrees, which is half of the contemporary solutions.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

Finding the walking direction of a pedestrian is a long standing problem in the field of indoor localization. A large family of indoor location based applications, like geofencing, personal diaries, pedestrian navigation, automatic map generation and so on, rely on an accurate heading direction estimator for their performance. It acts as an important building block at the lower layers for such applications. The researches on the mention field often assumes the existence of an ideal heading direction estimator module or come up with an application specific solution. However, this problem demands a complete research that can lead to a generic and accurate pedestrian heading direction estimation module. It will benefit the proposed indoor localization based solutions by making them feasible for practical implementation and many upcoming researches to use this as a reliable building block.

In this project we attempt to come up with a stand-alone and generic algorithm for finding the heading angle of a pedestrian with the help of smartphone sensors. In ideal scenario, a generic moving direction estimator application should have the following properties:

- No requirement for known initial orientation of the sensing device
- No restriction on the orientation of the device
- Fast, real-time response

- Little or no bootstrapping
- High accuracy without any correction by the constraints of the map
- Low complexity and energy efficiency that makes it applicable for resource constrained computing devices
- No application specific assumption
- Applicable in indoor as well as outdoor scenarios

The existing IMU based solution uses custom devices, which are mounted on particular parts of the body. Other solutions that use the smartphone as sensing and computing device put restriction on the usage of the smartphone to figure out the initial orientation of it and to track the subsequent orientation change. GPS is one of the conventional solutions of this problem, but it does not work in indoor scenario. Moreover it is extremely power hungry and measurement accuracy is also not suitable for pedestrian. Another well known approach is dead-reckoning. But dead-reckoning also needs to know phones initial orientation and over time it keeps accumulating errors and after some time the estimation becomes to noisy to use. However this is worth mentioning that this technique also stores the sensor readings for the entire dead-reckoning duration which requires a lot of memory for storing and processing the raw data.

With the ambition to propose a solution which meets the requirements of an ideal solution and which is especially suitable for the pedestrian, we propose WalkCompass. Specifically the contributions of this project are threefold:

- A novel algorithm for pedestrian heading direction estimation that follows the properties of a generic solution for the problem.

- Detail analysis of the effect of human walk on the smartphones placed at various parts of the body. It provides a theoretical base to the proposed solution and similar works with smartphone's inertial sensors.
- An accurate algorithm for determining an extended set of motion modes of the smartphone, without the costly frequency domain analysis of data.

In this system we determine the moving direction of a pedestrian carrying a smartphone by analyzing the forces experienced by the device because of various movements during the walk. These forces do not depend on the orientation of the phone. Therefore the algorithm is inherently free from any error generated by the orientation of the phone. Moreover, the performance of the system does not depend on the holding style or location of the phone in the body. The algorithm can determine the direction of movement in real time and because of its low complexity, the complete system can be implemented on a smartphone. WalkCompass does not need any bootstrapping and can produce results with the granularity of each step of a walk. The heading direction estimated by this algorithm has average error of 6 degree, which is half of the contemporary attempts [15].

Chapter 2

Shortcomings of the Conventional Approaches

Digital Compass

The digital compass is ubiquitous in smartphones nowadays. The compass application processes the values of magnetometer sensor to figure out the direction of a specific axis of the smartphone. This compass can give the heading direction if that axis is aligned to the direction of the walk. If the smartphone is not properly aligned with the walking direction then it produces wrong estimation for the heading angle. It creates a great inconvenience to the user to hold the phone steadily throughout the walk. Moreover, this solution is completely useless for the applications that attempt to function without user intervention.

Velocity Vector

Theoretically the velocity of the user can be obtained by integrating the linear acceleration. The velocity vector itself can point out to the heading direction. But in practical scenario, the whole system works in a different way. This is because the smartphone sensors, however, are inherently noisy. The accelerometer sensors used in smartphones are MEMS sensors. They contain micro electro-mechanical parts that convert the acceleration in difference of capacitance. Therefore the source of the noise is partly the mechanical moving elements. The constant noise from internal

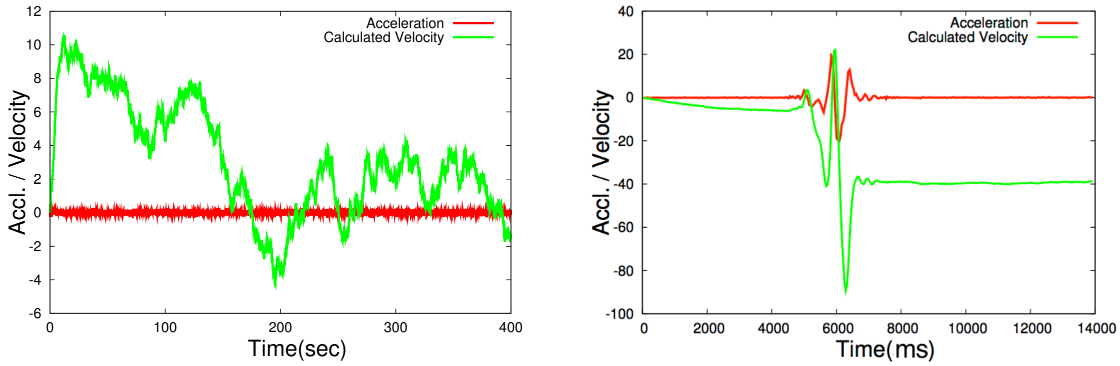

Figure 2.1: These experiments show the error in the velocity estimation by integrating the linear acceleration values obtained from a smartphone sensor. The phone is kept stationary on the table, but the calculated velocity gets a high value because of the noise in smartphone sensor data.

circuitry also adds with this. As a result, the data produced by accelerometer sensors contain high level of noise. In addition, some ambient vibration causing random glitches also affects the data.

The noise present in the accelerometer data makes it almost impossible to get the accurate velocity or position by directly integrating the data. In order to show how worse the scenario is, we conduct a simple experiment. We collected linear acceleration reading for about 400 seconds by keeping the phone still on a table. Then we integrate the data to get the velocity of the phone, which should always be zero in this experiment. As we can see in the Figure 2.1a the noise is amplified by the process of integration and gets accumulated over time. As a result we can see that the calculated velocity reaches as much as 10 m/sec. Similarly in the second experiment, we collected the data for 14 seconds and during the experiment a small jerk was applied to the phone at around 6th second. However the phone was kept still for rest of the time. After the integration, the velocity was found to be negative instead of zero as shown in Figure 2.1b. It clearly indicates that it is not possible in practice to obtain the velocity vector by integrating the values of smartphone accelerometers.

GPS

Modern smartphones are equipped with Global Positioning Systems. GPS can provide the location of the smartphone and therefore can be considered as a solution to get the moving direction. However, the GPS estimation of position can have an error of around 10 meters. It is a small error in case of vehicular navigation but significant for pedestrian navigation. Moreover, GPS does not work in indoor location, which is the primary environment for pedestrian navigation. GPS is also marked by its power inefficiency. All these make GPS a poor choice for determining the direction of walk.

Dead Reckoning

One approximate solution to the problem of finding walking direction is based on dead-reckoning. In navigation, dead-reckoning is the process of estimating the current position by continuously calculating the distance or orientation from a known fixed position. In order to find the moving direction with dead-reckoning process, the system first registers the direction of compass at some known orientation of the phone. After that it constantly records the change in orientation. At the final position the system calculates the current orientation of the phone from the recorded changes and the initial orientation. Once the current orientation is known, it can compensate the current compass value accordingly to find the current moving direction. However this process suffers from errors in calculating the final orientation due to the noise and inaccuracy of the sensors. As a result, the estimate of the direction drifts and the error accumulates over time. Therefore it needs periodic correction with the help of known landmarks. The periodic correction process may not be feasible in most of the cases. The initial known orientation also adds another constraint on this. Overall dead reckoning fails to present an accurate and practical solution.

Chapter 3

Related Work

Estimation of heading direction always remains an important piece of information since the beginning of the history of navigation and positioning. The advancement of navigation is primarily dominated by the advancement of the tools that find the direction of movement. Magnetic compass is the most important among them. It progresses from a floating magnetized needle to a 32 points compass during the thirteenth, fourteenth and fifteenth centuries [18]. The invention of magnetometer in 1833 and the Hall effect [9] in 1879 led to the electronic compass, which opens up the door of electronic world to the navigation system. Various compass enabled electronic devices declared the modern age in navigation and positioning system. Now almost every smartphone is equipped with a digital compass. It is the inbuilt compass and many other sensors including accelerometer and gyroscope that attracts the researcher to develop a smartphone based personal navigation system for the user.

Another important landmark that expedite the advancement of the personal navigation system is GPS. The opening up of GPS for civilian users in 1996 [17] revolutionized the personal navigation and positioning. It was end of 2004 when the assisted GPS was successfully introduced to mobile phones and in 2007 Nokia E90 [1] was the first smartphone to come with an integrated GPS system. GPS, however, could not impact the development of indoor positioning and navigation system because of its poor performance inside a building. Accuracy of GPS, close to 10 meters, is also large compared to the pedestrian movement. Researchers still mostly rely on the built-in inertial sensors for heading direction estimation.

Many early efforts to find the heading direction estimate involve various techniques. Some of them directly use raw compass data after filtering for noise. Some of the papers propose GPS assisted solutions [2] to find the heading angle. James et al. [6] proposed a solution of finding the heading by observing the movements of eyes as the user looks at various objects during the walk. One of the recent researches on this topic exploits the smartphone camera and finds the vanishing point to detect the heading angle [20]. Direction of movement estimation with the help of inertial sensors is also explored in several solutions. A number of researches on Pedestrian Navigation System (PNS) [12, 13, 24] present solution for the heading angle estimation with IMU based device mounted on the shoe or fixed at some other part of the body. Scope of this thesis is distinct from these solutions. We attempt to estimate the heading direction of a pedestrian in an indoor environment with the help of the smartphone carried by her. The smartphone can be under the normal usage and we do not put any restriction on the orientation or usage pattern of the device. As carrying the phone in a pocket is considered as normal usage pattern, we do not involve the camera in our solution. At the same time we also exclude GPS from consideration, as the primary location for pedestrian navigation is indoor. To provide a generic solution, without any infrastructural help from the outside or training of any kind, we keep the WiFi or Bluetooth based solutions outside of the scope of this thesis.

The problem in the heading direction estimation with built-in digital compass of the smartphone is that the change in the orientation of the device adds an offset to the estimation. It becomes worse in pedestrian scenario as different user have different gait and the orientation may change randomly during the walk. Many solutions deal with this offset with the help of dead reckoning, a technique of continuous monitoring the change in orientation from a known initial orientation of the device. Apart from the accumulated error in dead reckoning process, the requirement of the initial known orientation puts some additional constraints on the usage pattern. For example, [16] requires the user to hold the smartphone in a specific orientation as long as the application is functional. The solution in [14] assumes that the smartphone will be in a pocket of the pants with limited variation in the placement. The initial orientation is inferred in [15] from the gesture of the user when she holds the phone to open the application itself. This opportunity is specific to some application and particularly absent in the scenarios when the applications is designed to function without user intervention. Moreover, it assumes that the user will put the smartphone in the pocket before starting to walk. This is a controlled environment too, as she can use the phone during the walk and switch between various positions in between.

Zee et al. [19] presents a solution based on differences in the pattern of frequency domain representation of the accelerometer signal during walk. This technique alone cannot distinguish between forward and backward movement. Moreover the estimation obtained from this technique needs to be corrected using the constraints on the allowable walking space imposed by the map. There are several other solutions, like FootPath [16], that depends on the map to correct the crude estimations of heading angle. The map of the area may not be available for all applications. One example of such applications can be the automatic map building applications [21]. So the availability of the map is not a valid assumption for a generic heading direction estimator.

Chapter 4

Intuition

Analysis of Force, Acceleration, and Velocity

The concept of heading direction estimation in WalkCompass is based on the principles of force and motion in Newtonian physics. According to the principles, all motions of an object are the effect of the forces applied on it. Every change in position or velocity, from the drop of a pen to the complex motion of a car, is actually caused by the forces. The relationship between the force, velocity and acceleration is better explained with a simple 2 dimensional scenario described in the Figure 4.1. Here the object was initially stationary at position A. A constant force (F1) is applied for n seconds on it and after that, at position C, the force reverses its direction (F2) and again acts for n seconds. Even if we cannot observe the force directly, we can sense it through the changes in position, velocity and acceleration of the object. The change in position, velocity and acceleration, caused by this force, are depicted in this figure. First the object starts moving in the direction of the force and its velocity increases as long as the force is applied to the forward direction. The velocity gradually decreases to zero in the second half, when the force is applied in the opposite direction. The change in velocity always happens in the direction of the applied force. The acceleration of the object is actually the rate of change in velocity. The direction of the acceleration, therefore, always points to the direction of the applied force. As we see in this example, a couple of forces applied from the opposite directions of each other create the rise and fall in the velocity of the object. During the first half, the object moved in the direction of the applied force (F1) and in the second half, when the object was still in motion, it moved in the opposite direction of the applied force (F2). Therefore, we can figure out the direction of the movement of this object once we know the direction of any one of these forces.

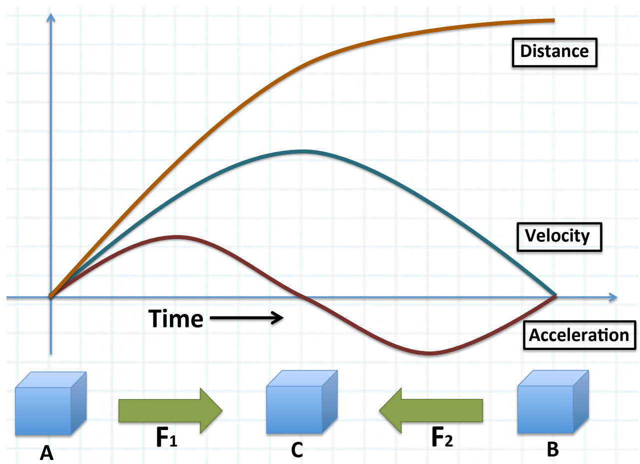

Figure 4.1: The relationship between the force, acceleration, and velocity is explained with an example scenario.

The movement of limbs during walk also follows the basic relationship of the force, velocity and acceleration mentioned above. Each part of the body experiences acceleration due to the force that moves them forward and so does the smartphone placed at that part of the body. The acceleration, recorded by the smartphone at various point of time, can be exploited to get the direction of the force applied to it at that time.

Experiments on Human Walk Cycle

A close analysis of human locomotion reveals that the body does not move with same speed throughout the walk. If we divide a walk into repetitive patterns, called gait cycle [8], we will find that at the beginning and end of the cycle the velocity is low. Actually the velocity of the walk slows down whenever any of the feet touches the ground. This means when the leg swings in the air, it experiences two opposite forces. During the first half, a force is applied to increase the velocity of the limb in the direction of the walk and in the next half it reverses its direction to slow down the velocity before touching the ground. Lets call the first half of this process 'acceleration phase' and the second half 'deceleration phase'. In order to determine the heading direction from the direction of these forces we need to know which of these phases the force corresponds to.

Many researches [7, 8, 11] analyze the dynamic of human walk and show the acceleration of various parts of the body. However, work that explores the acceleration recorded by the smartphones at their normal usage during a walk is rare. We investigated this in detail in our experiments with multiple persons carrying smartphones at various places including inside various kinds of pants pocket, on palm top, in swinging hand, in cases attached to waist, and inside backpack. We record the movement of the subjects and collect the sensor data from smartphones, which are time synchronized with the camera. Initially the room was dark and the cameras were running. The smartphones were also sensing the light of the environment through their ambient light sensors. Then we turn on the light to have a sudden increase in light level sensed by the smartphones as well as the cameras. Later when we analyze, with our OpenCV [3] based custom software, the senor data from the smartphones and video from the camera, we use this hint of change in light level to

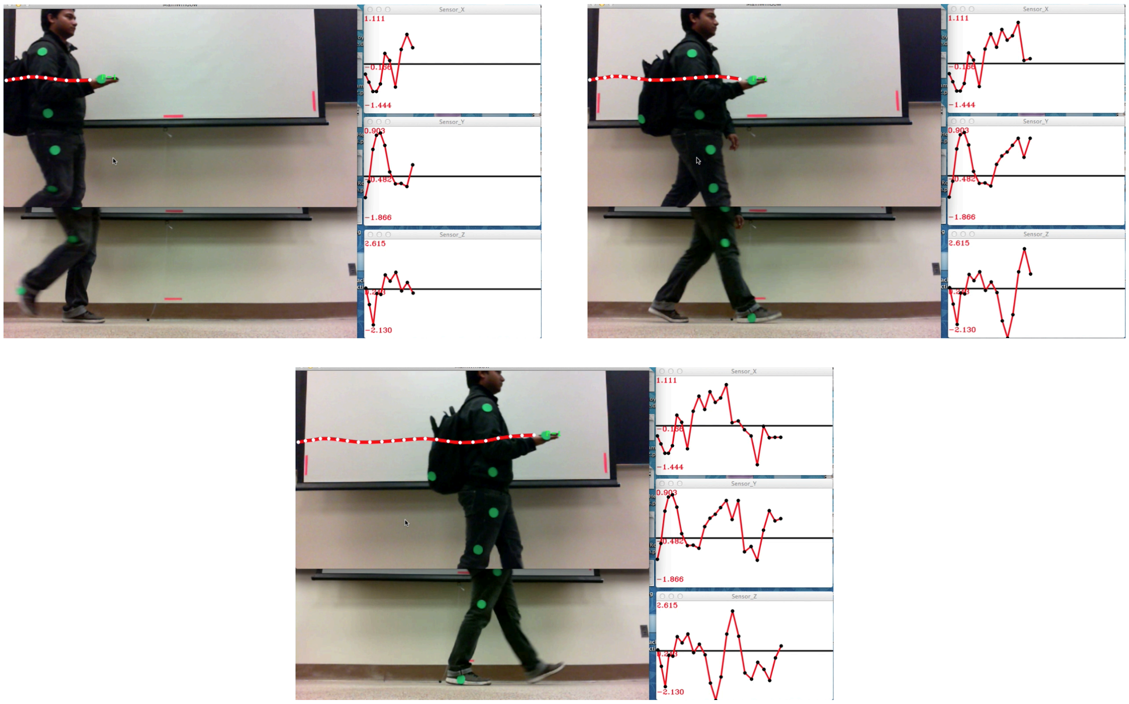

Figure 4.2: Three different stages of the walk cycle and the corresponding values of linear acceleration, as measured by a smartphone held on the palm top.

achieve the millisecond level synchronization between these two sources of data. We analyze frame-by-frame movements during the walk of each person and corresponding changes in the sensor data. Figure 4.2 shows the screen shots of the experiment and the corresponding values of linear acceleration in the three small windows on the right. The user holds the smartphone on the right palm top during the walk. It also tracks the phone in the video to show the vertical movement of the phone. The window on the top left focuses on the hand and the window below it shows the position of the feet. The screen shots of the same experiment with the linear acceleration data from a smartphone carried in the right pocket of the trousers in given in Figure 4.3. In the screen shots we can observe the changes in the sign of the acceleration vales at different phases of walk. It also clearly shows the sudden changes in all three axes of the accelerometer when the leg touches the ground. The results of this experiments leads to the following observations:

1. Most of the time the body moves in the direction of the leg when it swings in the air. This serves as the basic segment of heading direction of a person for WalkCompass.

2. The swinging leg experiences a forward force during the first half of the movement and an opposite force during the second half. The forward force actually shows the heading direction.

3. The movement of the legs and other parts of the body are different in nature. Therefore, for accurate step detection, the accelerometer data obtained from the smartphones placed various parts of the body should be processed in different ways.

Based on the above observation we design an algorithm that can determine the heading of a pedestrian at each footstep, irrespective of the orientation of the smartphone. This algorithm first determines the gait cycle of the user and its various phases. After that it figures out the time frame when the smartphone was under the influence of the forward force. The acceleration under that time frame points to the heading direction of the user. It determines the direction vector from the accelerometer data and thus gets the vector that represents the heading direction in 3-dimensional space. This raw heading is then projected on desired plane of movement and compensated for the possible errors introduced by lateral movements of bipedal locomotion.

The gait cycle determination process of WalkCompass depends on various features obtained from the inertial sensor data. It, therefore, depends on the part of the body

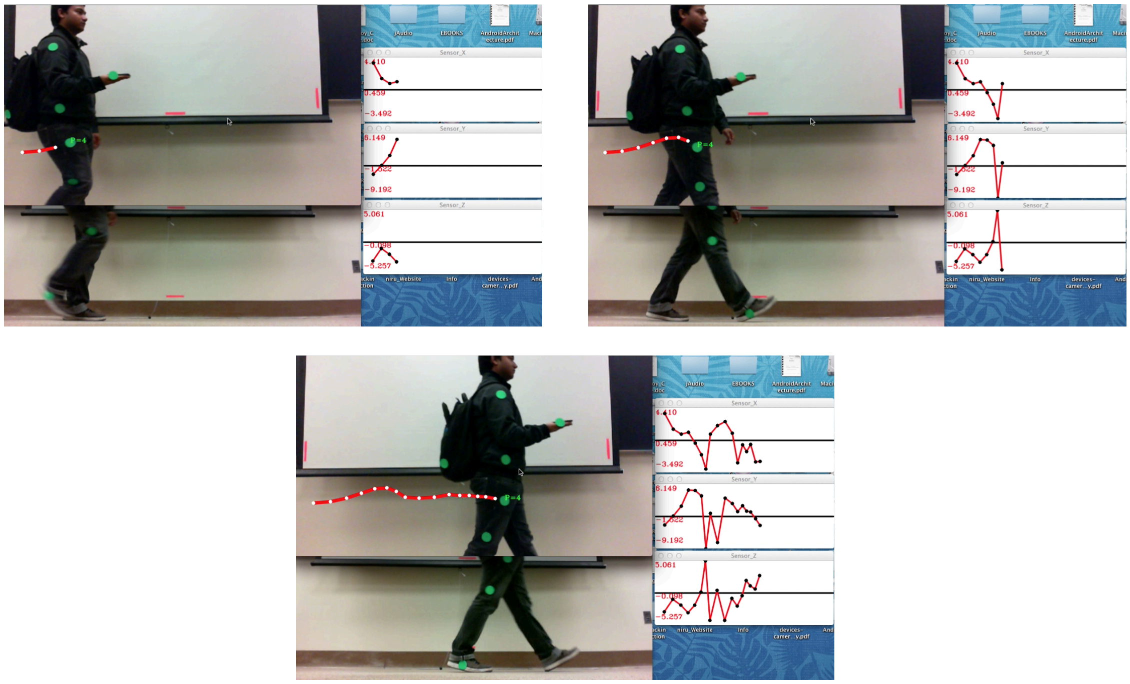

Figure 4.3: In this experiment the user carries a smartphone in a pocket of his trousers. The figures show three different stages of the walk cycle and the corresponding values of linear acceleration, as measured by a smartphone.

the smartphone is placed. To deal with this, WalkCompass distinguishes between various usage patterns or motion modes of the smartphone with the help of a novel motion mode detection algorithm. This algorithm depends only on the various time domain features to accurately determine the motion mode of the smartphone and completely avoids costly frequency domain conversion of the data.

Chapter 5

Design and Implementation

Overview

We design the WalkCompass application in such a way that can calculate the heading direction in real-time. The complete flow is divided into small modules that can be plugged or removed without any turbulence in the flow. The flow, as shown in the Figure 5.1, starts with the collection of available sensor data with the help of standard API provided by the platform, which is Android in this case. As the pedestrian movement analysis does not need very high frequency signals and sampling at higher rates includes high frequency noise in the data, we limit the data collection to a constant rate of 25 samples per second. The collected data goes to the Motion Mode Detection module and a buffer module simultaneously. However, a low pass filter module de-noises the sensor data before sending it to the buffer, whereas the motion mode detection algorithm works with the raw sensor data. The Step Detection module takes hint about the motion mode and generates corresponding intermediate signals to aid the step detection. It then estimates the timings for the sensor data that correspond to the heading vector. The algorithm fetches the heading direction vector and the compass direction vector from the buffered sensor data and feeds it to the Heading Direction Calculation module for generating the heading angle. At the end, the Post-processing module compensates for the errors in generated heading angle and produces the final result. We discuss in details the major modules below.

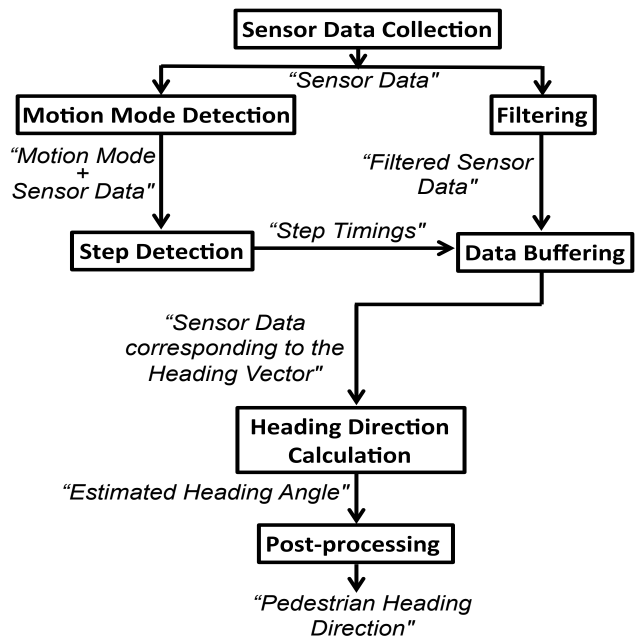

Figure 5.1: The flow diagram of the WalkCompass application.

Motion Mode Detection

We design a novel motion mode detection algorithm to determine various use patterns. The algorithm is based on the energy and various time domain features of the accelerometer signal. We avoided costly frequency domain analysis for this purpose that can potentially slow down the whole chain of heading direction estimation flow. The energy of the accelerometer signal alone can provide coarse division between high and low intensity activities as shown in [23]. The high intensity activities can be a walk with the phone in swinging hand or pants pocket. On the other hand the stationary position, walking with phone on palm top or shirt pocket are the examples of low intensity activities. The activity identification with energy detection alone is not precise and does not meet the requirements of WalkCompass. For example, it fails to distinguish a walk with the phone on the palm top from texting in stationary position. WalkCompass uses the hint of motion mode to accurately identify the timing of the steps, as the phone shows significant diversity in the sensor data depending on the use pattern. According to the scope of the current version of WalkCompass, the Motion Mode Detection module recognizes the following cases:

- Case 1: The phone is held in the hand that swings naturally during the walk.
- Case 2: The phone is in any of the pockets of the pants.
- Case 3: The phone is on the palm top or held in way so that the user can use it. The user can watch something on the screen and click or type on it. However, the motion detection algorithm does not assume that the screen will always face the user. It considers the use pattern to be 'Case 3' as long as the user holds the phone in hand to restrict its natural movement. It does not put any constraint on the orientation of the phone.
- Case 4: The motions when phone experiences a sudden jerk. For example, the user raises the hand that holds the phone.
- Case 5: All other low energy random movements.

To achieve more granularity in activity recognition without the help from frequency domain features, we focus on the periodic nature of the accelerometer signal during a walk in time domain. As we can see in the Figure 5.2, the accelerometer signal shows the periodic ups and downs according to the steps taken by the user. This feature is strongly connected with the walk or run and does not appear in the signal for any other normal activity with the phone. We process the accelerometer signal to find out the 'positive zero crossing' in order to capture this periodic nature

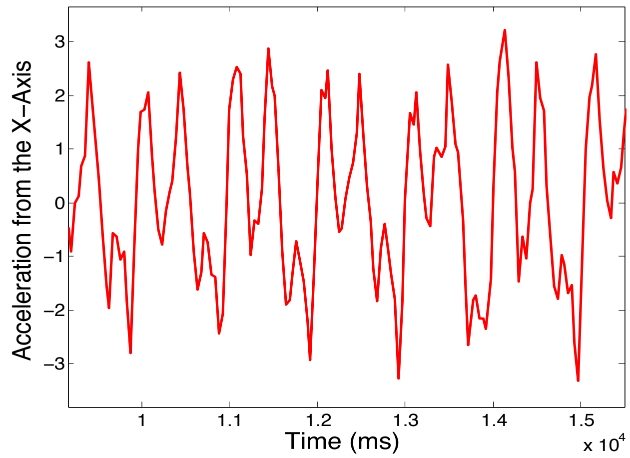

Figure 5.2: The accelerometer signal shows periodic nature during the walk.

in a feature. 'Positive zero crossing' data is actually a series of values that represents the times when the signal crosses (or touches) the value zero line from the negative side. Given the points of the signal are represented by the time and acceleration pair, $<t_i, a_i>$, a point tz_i in the 'positive zero crossing' is calculated with the formula given below:

$$tz_j = t_i - a_i \cdot \frac{t_{i+1} - t_i}{a_{i+1} - a_i} \forall a_i \leq 0 \; and \; a_{i+1} > 0 \tag{5.1}$$

The mean and standard deviation of the 'positive zero crossing' values over a window are the two features, which together with the power of the accelerometer signal distinguishes between 5 different modes of user activity required for the Walk-Compass algorithm. As discussed before, the walk leaves a periodic effect on the accelerometer value. The zero crossing values are also regular in case of walk, compared to any other random activities like texting. So, a very low standard deviation of zero crossing values indicates an action involving repetitive patterns, like walk or run. However, this feature is not prominent in all three axes of the accelerometer for all the time. Depending on the gait and orientation of the phone, an axis of the accelerometer may not have expected positive zero crossing value. Fortunately, in all the cases we find at least one axis shows consistent standard deviation value. We use this feature to distinguish between the 'Case 3' with a relatively stable phone and some random low intensity activity when the user is not walking.

The mean value of the positive zero crossing values over a window provides an estimate of the time interval of the repetition. In high intensity activities, the low mean value signifies the 'Case 2' and a high mean value is for the 'Case 1'. The algorithm first categorizes the quick random jerk, high intensity activity and low intensity activity by looking in to the total energy level of the accelerometer signal from all 3 axes. In the next step it uses the standard deviation and mean of the zero crossing values of accelerometer to distinguish between various cases of user activities as mentioned before.

Step Detection

Step detection is an integral part of the WalkCompass algorithm. It helps the algorithm to find out the status of the gait cycle. Various movements of the body during a walk leave impact on the three axes of the accelerometer data. If the phone is on the right side of the body, it experiences higher impact from the movement of the right limbs than that of the left. The reverse is also true. We call the side of the body, on which the mobile is placed, is the 'primary side'. The other side is called 'secondary side'.

Detecting the beginning and ending of a gait cycle generally involves the detection of the point in time when the primary leg touches ground. This event has a significant impact on the accelerometer value, as it delivers a sharp jerk to the device. During all other time the values from the different axes of the accelerometer may or may be

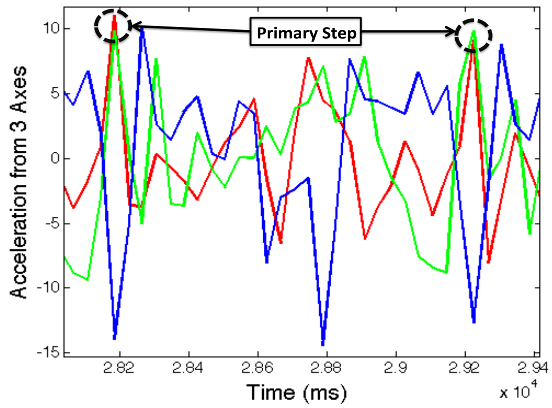

Figure 5.3: The linear acceleration signals from all three axes simultaneously reach extreme values when the primary leg touches the ground during the walk.

correlated depending on the orientation of the smartphone, but at the ground touch of the primary leg makes all the three axes to reach a high value simultaneously. This is shown in the Figure 5.3. We exploit this key observation to create another intermediate signal, which is used for the primary step detection in our algorithm. To generate this step detector signal, we first take the absolute values from individual axes and add them together. Finally we take the envelope of this summation to get the step detector signal.

Thus we get a signal that shows the point when all the three axes of the accelerometer gives a very high value together. The local peaks of the step detector signal indicate the ground touch of the primary leg. The peaks in the step detector signal, as shown in Figure 5.4, indicates the timings of the primary steps. We run a peak detector algorithm to figure out the timing of the steps.

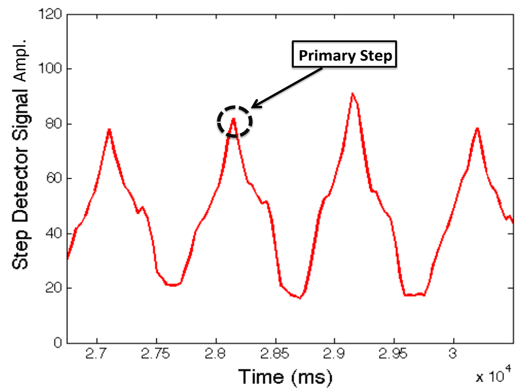

Figure 5.4: The peaks in the step detector signal, an intermediate signal generated from the linear acceleration signal, indicates the timings of primary steps.

Heading Direction Calculation

The time between two primary steps is the gait cycle. A gait cycle involves two sets of acceleration and deceleration phases one for each leg. The cycle starts with the acceleration phase of the secondary leg and followed by its deceleration phase. After that the two phases of the primary leg comes one by one. Therefore, if we divide the gait cycle into four parts, the last part represents the deceleration phase of the primary leg. We chose this part for finding the heading direction vector. We take a window in the middle of this phase and find the average of the values of three axes separately from linear acceleration data within this window. These three average values, one from each axes, represents the three component of the heading direction vector on the phone's coordinate system. As it is a deceleration phase, the calculated vector points exactly opposite to the direction of the walk. We compensate for this error by reversing the direction of the vector.

The calculated heading direction vector shows the movement of the leg in three-dimensional space; we need to project this vector to a horizontal plane to find the actual direction of the walk in user's two-dimensional space of movement. This horizontal plane should be globally constant and must not depend on the coordinate system of the phone. We take the plane perpendicular to the gravity for this purpose. The gravity vector always points to the ground no matter what the orientation of the smartphone is. Therefore it can serve as a global constant. We call this the 'walk plane'. We applied vector operations to find the projection of the heading vector perpendicular of the gravity vector. The projected heading vector lies on the walk plane.

To present the heading direction in angles like a compass, we find out the angle that the projected heading vector makes with the magnetic North of the Earth. We project the magnetic field vector on the 'walk plane' and find the directed angle between from the magnetic field vector to the heading vector on the 'walk plane' to get the direction of the walk as the deviation in angle from the magnetic North.

Post-processing

In bipedal movement the pedestrian takes a step to the left or right of the actual heading direction. The step by step estimated angle is, therefore, does not give the true direction of heading. Moreover, the generated heading angle may be noisy because of occasional random forces applied to the smartphone during the walk. To deal with this issue we post-process the estimated heading angle by taking the average of two consecutive estimates and applying a median filter. The output of the Post-processing module is the final heading direction estimated by WalkCompass.

Chapter 6

Evaluation

In this section, we evaluate our algorithm under various testing scenarios. In the first experiment, we focus on the heading direction estimation in two different usage patterns. The next experiment evaluates the performance of the WalkCompass for its robustness in the scenario where the orientation of the phone changes frequently and the direction of the walk remains unchanged.

Estimated Heading Angle

We have various users to use the WalkCompass application in a predefined path in the corridor and recorded the data including the estimated heading direction. During the walk the user has to go approximately 25 steps at 160 degree before taking a right turn and then again around 30 steps forward at 244 degree. In this experiment we include the 84 degree turn in the path to measure the response of the algorithm when the heading direction changes to a large angle. We consider the 160 degree and the 244 degree orientation of the corridor as the ground truth for this experiment. However, this kind of constant direction value cannot serve as a ground truth for pedestrian walk. In an unrestricted natural walk, the user can not exactly follow a straight line. Most of the time she leans towards a side and constantly corrects the direction to reach the destination. Therefore, an algorithm that estimates the heading direction at the granularity of a single step, cannot use the direction of the line joining the start and end position as the true direction of walk.

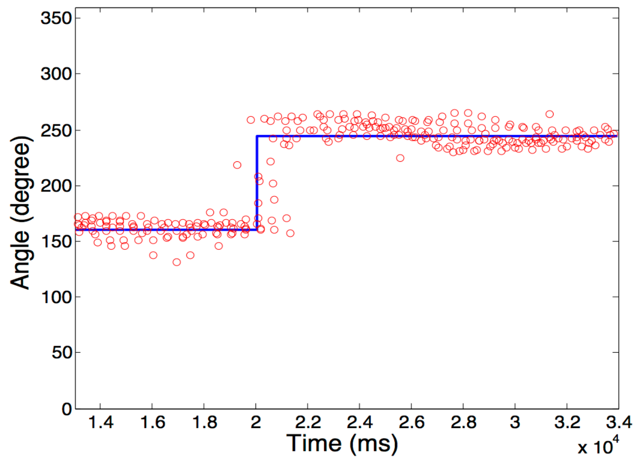

Figure 6.1: The scatter plot of the estimated angles. Here the user carries the phone on her hand during the walk and simultaneously uses it.

We tested the application under two scenarios: on palm top and inside pocket. In the first scenario the user hold the phone in hand in a way so that she can watch the screen while walking. This captures one of the common holding position where the person can use the phone by clicking or typing on the screen or watching something on it. Although there is no restriction on the initial orientation of the phone, the user does not change its orientation during the walk. We analyze the effect of orientation change in our next experiment. We collect several traces of this experiment and the scatter plot in Figure 6.1 shows the estimated heading angles. The red dots show the heading angle of the user at each step and the blue line shows actual angle at various time. As we can see in the figure, the calculated heading angle follows the true direction of the walk and changes accordingly. However, after the turn the application takes four steps to adapt the changed heading angle. The filters used in WalkCompass to produce a stable result from the noisy sensors introduce this delay.

In the second scenario, the user carries the smartphone trouser pocket. We do

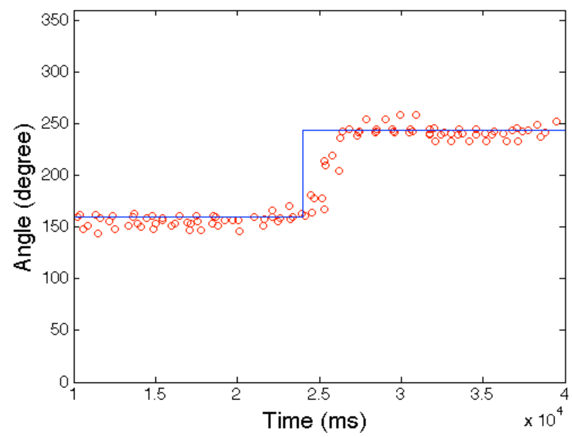

Figure 6.2: The scatter plot of the estimated angles. Here the user carries the phone inside the pocket of her trousers.

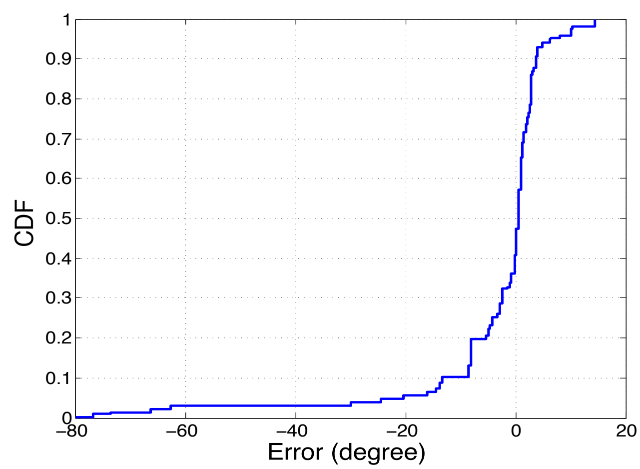

Figure 6.3: The CDF of the errors in estimated heading direction. The error is calculated as the difference between the estimated angle and the ground truth of the heading.

not put any restriction on the choice of the pocket and orientation of the phone inside the pocket. The user, however, does not deliberately change the position of the phone during the experiment. We find a similar scatter plot in Figure 6.2 for this experiment also. We calculate the error of the estimated heading angles by taking the difference from the orientation of the corridor. The CDF plot of the error, as we

Table 6.1: The various statistics of the errors in estimated heading direction.

Statistics	Value
Mean	6.2058
Median	2.7402
Standard Deviation	12.4174
Minimum error	0.0256
Maximum error	80.3090

can see in Figure 6.3, shows that the estimated angle remains within 10 degree of the ground truth for around 70% of the samples. There is a high error region on the plot, from -20 degree to -80 degree, for around 5% of the samples. These high errors come from the delay in adapting the 84 degree turn. The mean value of the error is 6 degree. The Table 6.1 shows various statistics of the error as measured during the experiment.

Impact of the Orientation Change

One of the indispensable properties of a generic heading direction estimator should be the robustness for any change in the device's orientation. It becomes the most challenging part in case of pedestrian heading direction estimation, as the device continuously changes its orientation during the walk. The direction and amount of the orientation change is unpredictable, because we do not put any restriction on the carrying style of the phone. The algorithm of WalkCompass, however, inherently free from the effect of the orientation. It figures out the direction of the forces applied on the device during the walk and in the global co-ordinate, this direction remains unchanged even after the device changes its orientation.

To test the robustness of WalkCompass, we designed experiments where the direction of the walk remains unchanged, but the user changes the orientation of the phone to large angles. In this test the user carries the smartphone, with running

WalkCompass application, in her hand. She walks 60 steps in a straight corridor where after every 15 steps changes the orientation of the phone. Under such scenario, ideally the output of the heading direction estimation algorithm should remain constant throughout the walk and should always point to the actual direction of the walk, unaffected by the devices orientation change.

In this sub-section, all the figures show the direction of the walk as calculated by the WalkCompass in red solid line. The x-axes of the plots represent the time of the walk and y-axis shows the angle in degrees. All the angles are calculated as the deviation from magnetic North in a horizontal plane. The green dashed line shows the compass angle, which is the direction the y-axis of the phone points to. The compass angle gives us a hint about the change of the orientation of the phone. The thin blue horizontal line shows actual orientation angle of the corridor; 244 degree. We show the blue line in the plot at 244 degree to mention the direction of the corridor and can be taken a very rough approximation of the ground truth. In the figure we can see that the plots does not start from time zero. It is because here the time starts when the application is turned on, but the walk does not start at that moment. The user takes some time before she starts walking and again there is a time gap between the finish of the walk and turning off the application.

In the first experiment the change in orientation is approximately 90 degrees after every 15 steps. As we see in Figure 6.4, the calculated orientation is not affected by the orientation change. We can see the change in phone's orientation from the compass angle plot. To test it in more common orientation change scenarios, in the second experiment, the user switches the phone orientation between texting and calling positions. The Figure 6.5 shows the results, which is similar to the first experiment.

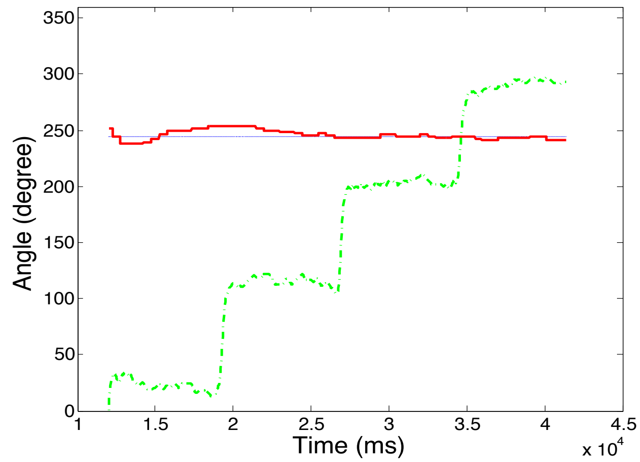

Figure 6.4: The user changes the orientation of the smartphone by 90 degrees after every 15 steps. The user walks along a straight corridor, oriented at 244 degrees. The heading vector calculated by WalkCompass remains unaffected on orientation change of the smartphone.

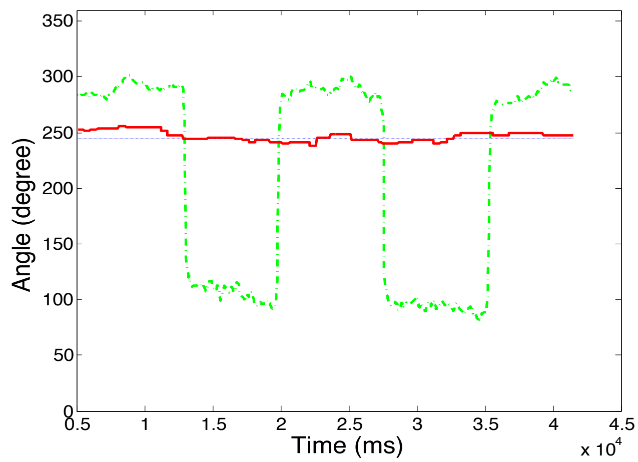

Figure 6.5: The user switches the orientation of the smartphone from texting position to calling position after every 15 steps. The user walks along a straight corridor, oriented at 244 degrees. The heading vector calculated by WalkCompass remains unaffected on orientation change of the smartphone.

We repeat the same experiment with various users and collect the data. The initial orientation of the smartphone and the change in orientation was up to the user and therefore random. We present the calculated orientation for all of these

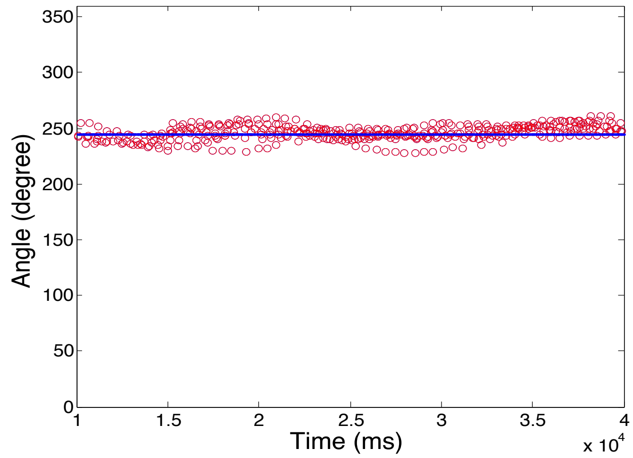

Figure 6.6: The scatter plot of the estimated heading direction for many users walking on a straight corridor. The users change the orientation of the smartphone in any random direction after every 15 steps. The corridor is oriented at 244 degrees. The heading angles calculated by WalkCompass mostly follow the direction of the walk along the corridor without being affected by the random changes in the orientation of the phone.

Table 6.2: The various statistics of the errors in estimated heading direction. The experiments focus on the robustness of the algorithm when the orientation of the phone changes frequently and the direction of the walk remains unchanged.

Statistics	Value
Mean	5.4701
Median	4.5573
Standard Deviation	4.2746
Minimum error	0.0404
Maximum error	22.5499

traces of this sub-section in Figure 6.6. Here the red circles represent the calculated angles and the blue horizontal line shows the orientation of the corridor(244 degree) as mentioned earlier. Here we can see how the trend of the calculated angled follows the general direction of walk. We calculated the error in heading direction estimation by taking 244 degree as the ground truth, although this does not follow the true heading at each step. We plot the cumulative distribution of errors in Figure 6.7. Here we can see that the error is limited to 10 degrees for more than 80% of the samples. In

Figure 6.7: The CDF plot of the error in estimating the heading direction, when the orientation of the phone changes frequently. The error is the difference between the estimated angle and 244 degree, the approximate ground truth.

Table 6.2 we present various statistics, generated on these estimated error values, that summarizes the performance of WalkCompass in withstanding smartphone's orientation change during the walk. We find that the mean of the error remains close to the statistics presented in Table 6.1.

Cold Start Issue

We can notice the existence of high error values in the CDF plots, although in a small percentage. This high error values appear because of the initial empty buffers, which are used for filtering various data. The buffers are used to store a window of values for the moving average calculator or for the low pass filters used to process the raw sensor data or for the median filter used to calculate the final heading angles. In the beginning of the execution of the application, these buffers are filled with zeros and therefore any new value is underestimated until the buffer is sufficiently filled with valid values. We call this the 'cold start issue'. The estimates during this period do not match with the reality and thus leads to high error. This situation can also

Table 6.3: The confusion matrix shows the performance of the Motion Mode Detection algorithm. The figures in the cells show the percentage of the calculated motion pattern.

TrueMode	Case1(SwingHand)	Case2(Pocket)	Case3(PalmTop)	Case4(Jerk)	Case5(Other)
Case 1	82.88	0.75	0	15.85	0.52
Case 2	13.21	84.04	0	0	2.75
Case 3	0	0	97.22	0	2.78

arise when the direction of walk changes abruptly with a high angle. We are working to mitigate this issue by adaptively resizing the buffers and calibrating the optimal sizes for them.

Performance of Motion Mode Detection Algorithm

We evaluate the performance of the Motion Mode Detection algorithm by recording the calculated motion mode at each step of the user. The users walk on a corridor that has one left turn, one about turn, and one right turn. It takes approximately 80 steps for the user to walk on this specific path during the experiment. The users naturally carry the smartphone in one specific pattern for each experiment. We focus on the swinging hand ('Case 1'), pocket ('Case 2') and palm-top ('Case 3') scenarios for this evaluation. The Table 6.3 shows the confusion matrix with the percentage of the calculated motion modes for each of the experiments. The rows correspond to the experiments with one motion mode and the columns shows, in percentage, the number of times the algorithm predicted the corresponding motion mode. We repeat each experiment multiple times for each user and the result shows that more than 80% of the time the algorithm correctly identifies a motion mode.

Chapter 7

Limitations and Future Work

The heading vector determined by the WalkCompass algorithm shows the direction of the walk in the coordinate system of the smartphone. As the orientation of the phone's coordinate system is unknown, we need a global landmark to represent the heading direction relative to the direction of that landmark. The magnetic North of earth serves as the mentioned global landmark for direction. In other word, the direction obtained from the digital compass is used as the global reference direction. WalkCompass also use the direction of the magnetic field obtained from the built-in magnetometer of the smartphone to represent the walking direction of the user as the angle of deviation from this. This turns out to be the biggest source of error for WalkCompass in indoor environment. In indoor environment, the direction of the magnetic compass spatially varies due to the presence of ferrous structural material or contents, electrical power system or electronic appliances that affects the natural geo-magnetic field [4,5,10,22]. It introduces error in heading direction estimated by WalkCompass, although the heading vector computed by the algorithm is correct. We are working to find out an alternative global direction for our algorithm. We are exploring many techniques for this purpose including the WiFi RSS map, magnetic map and direction of Access Points etc.

In our future work we will also focus on the improvement in the response time and mitigating cold start issue. We are working on the idea that predicts the direction of the next step based on previous stable headings and clues from various other sensors to find the orientation change of the user. We call this Micro Dead Reckoning, as it uses the dead reckoning technique for the short intervals and at the points when the user takes a turn to estimate the change in the direction of walk. One of the challenges for this technique is to distinguish between walking direction change and the orientation change of the device only. We can also adaptively maintain the length of the buffers, so that at the beginning of the walk or whenever any turn invalidates the contents of the buffer, the algorithm can shorten the length of it and increase the length up to a limit when a series of stable headings are found.

In future we plan to extend the scope of WalkCompass in various other scenarios. The first among them is to determine heading direction in 3-dimensional space. The heading vector calculated by the WalkCompass actually points the direction in 3-dimensional space. Therefore, the algorithm can inherently produce the direction of movement in 3-dimensional environment. We will be working to adapt the algorithm to exploit this capability. This will be helpful to identify the heading direction when the user is climbing stairs and in many other scenarios where the movement is not restricted to the horizontal plane only.

There are cases where the user takes help from other equipment for locomotion, for example wheel chair, skating board etc. We envision that the heading direction estimator algorithm can work under such scenarios also as long as repetitive pattern of heading forces are applied and the smartphone can sense it during the process of movement. Moreover, WalkCompass can enhance the estimation of walking direction by communicating between multiple sensing devices placed at various part of the body. For example, the combined data from a smartphone in pocket and a tablet in hand can have more information about the user's movement and hence can produce better estimation of the heading direction. To extend the WalkCompass to work under these scenarios is another future direction of work.

Conclusion

Accurate estimation of the moving direction of a pedestrian is a long standing problem in the field of smartphone based indoor localization. Most of the indoor localization concepts assume that some algorithm can accurately find the moving direction of the pedestrian. However, existing solutions put some constraints on the use model of the system and thus make them ineffective in practice. WalkCompass proposes an effective solution in the direction of solving the heading angle problem by analyzing the various forces employed during the walking process. Our proposed algorithm can identify the moving direction of the pedestrian on per step basis. The solution is inherently free from the constraints of the orientation of the smartphone. This system can be used as a generic building block for any navigation system. We also analyzed the human gait pattern in detail and using this analysis we explained why our algorithm should work successfully in any pedestrian navigation system. We achieve a low average error of 6 degrees, which is a significant improvement for a generic solution. We are working on WalkCompass to make the system more robust, fast, and applicable to a wider set of scenarios.

Bibliography

[1] *Nokia wikipedia*, `http://en.wikipedia.org/wiki/Nokia_E90`.

[2] Mark Amundson, *Compass assisted gps for lbs applications*, Proceedings of the 18th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS 2005), 2001, pp. 2965–2968.

[3] G. Bradski, *The OpenCV Library*, Dr. Dobb's Journal of Software Tools (2000).

[4] Jaewoo Chung, Matt Donahoe, Chris Schmandt, Ig-Jae Kim, Pedram Razavai, and Micaela Wiseman, *Indoor location sensing using geo-magnetism*, Proceedings of the 9th international conference on Mobile systems, applications, and services, ACM, 2011, pp. 141–154.

[5] Francois Clinard, Chantal Milan, Mohamed Harb, Paule-Marie Carli, Claire Bonithon-Kopp, Jean-Paul Moutet, Jean Faivre, and Patrick Hillon, *Residential magnetic field measurements in france: comparison of indoor and outdoor measurements*, Bioelectromagnetics **20** (1999), no. 5, 319–326.

[6] James E Cutting, Paula MZ Alliprandini, and Ranxiao Frances Wang, *Seeking one's heading through eye movements*, Psychonomic Bulletin & Review **7** (2000), no. 3, 490–498.

[7] François Faure, Gilles Debunne, Marie-Paule Cani-Gascuel, and Franck Multon, *Dynamic analysis of human walking*, Computer Animation and SimulationâĂŹ97, Springer, 1997, pp. 53–65.

[8] Davrondzhon Gafurov, Kirsi Helkala, and Torkjel Søndrol, *Gait recognition using acceleration from mems*, Availability, Reliability and Security, 2006. ARES 2006. The First International Conference on, IEEE, 2006, pp. 6–pp.

[9] Edwin H Hall, *On a new action of the magnet on electric currents*, American Journal of Mathematics **2** (1879), no. 3, 287–292.

[10] Janne Haverinen and Anssi Kemppainen, *Global indoor self-localization based on the ambient magnetic field*, Robotics and Autonomous Systems **57** (2009), no. 10, 1028–1035.

[11] Verne T Inman, *Human locomotion*, Canadian Medical Association Journal **94** (1966), no. 20, 1047.

[12] J Won Kim, Han Jin Jang, Dong-Hwan Hwang, and Chansik Park, *A step, stride and heading determination for the pedestrian navigation system*, Journal of Global Positioning Systems **3** (2004), no. 1-2, 273–279.

[13] Bernhard Krach and Patrick Robertson, *Integration of foot-mounted inertial sensors into a bayesian location estimation framework*, Positioning, Navigation and Communication, 2008. WPNC 2008. 5th Workshop on, IEEE, 2008, pp. 55–61.

[14] Seon-Woo Lee, Philhwan Jung, and Seong-Ho Song, *Hybrid indoor location tracking for pedestrian using a smartphone*, Robot Intelligence Technology and Applications 2012, Springer, 2013, pp. 431–440.

[15] Fan Li, Chunshui Zhao, Guanzhong Ding, Jian Gong, Chenxing Liu, and Feng Zhao, *A reliable and accurate indoor localization method using phone inertial sensors*, Proceedings of the 2012 ACM Conference on Ubiquitous Computing, ACM, 2012, pp. 421–430.

[16] Jó Agila Bitsch Link, Paul Smith, Nicolai Viol, and Klaus Wehrle, *Footpath: Accurate map-based indoor navigation using smartphones*, Indoor Positioning and Indoor Navigation (IPIN), 2011 International Conference on, IEEE, 2011, pp. 1–8.

[17] Office of Science and Technology Policy National Security Council, *Press release - u.s. global positioning system policy*, `http://clinton4.nara.gov/textonly/WH/EOP/OSTP/html/gps-factsheet.html`.

[18] Douglas T Peck, *The history of early dead reckoning and celestial navigation: Empirical reality versus theory*, New World Explorers (2002).

[19] Anshul Rai, Krishna Kant Chintalapudi, Venkata N Padmanabhan, and Rijurekha Sen, *Zee: Zero-effort crowdsourcing for indoor localization*, Proceedings of the 18th annual international conference on Mobile computing and networking, ACM, 2012, pp. 293–304.

[20] Laura Ruotsalainen, Heidi Kuusniemi, and Ruizhi Chen, *Heading change detection for indoor navigation with a smartphone camera*, Indoor Positioning and Indoor Navigation (IPIN), 2011 International Conference on, IEEE, 2011, pp. 1–7.

[21] Hyojeong Shin and Hojung Cha, *Wi-fi fingerprint-based topological map building for indoor user tracking*, Embedded and Real-Time Computing Systems and Applications (RTCSA), 2010 IEEE 16th International Conference on, IEEE, 2010, pp. 105–113.

[22] William Storms, Jeremiah Shockley, and John Raquet, *Magnetic field navigation in an indoor environment*, Ubiquitous Positioning Indoor Navigation and Location Based Service (UPINLBS), 2010, IEEE, 2010, pp. 1–10.

[23] Melania Susi, Valérie Renaudin, and Gérard Lachapelle, *Motion mode recognition and step detection algorithms for mobile phone users*, Sensors **13** (2013), no. 2, 1539–1562.

[24] Oliver Woodman and Robert Harle, *Pedestrian localisation for indoor environments*, Proceedings of the 10th international conference on Ubiquitous computing, ACM, 2008, pp. 114–123.