

Ka Kei Yeung

Smartphone app for Fused Indoor Localization

Master Thesis (MA)-2014-20
November 2014 to May 2015

Advisors: Prof. Dr. Domenico Giustiniano and Dr. Vincent Lenders
Supervisor: Prof. Dr. Bernhard Plattner

Abstract

Localization using the Time-of-Flight (ToF) of RF signals is one of the most popular techniques to track moving objects, such as the Global Positioning System. However, its success for WiFi based indoor tracking remains a challenge. WiFi based indoor localization techniques produce large errors that are a result of multipath and hardware inaccuracies. In order to tackle this issue, we have developed an Android application that retrieves various sensor data, such as accelerometer and gyroscope from smartphones. Using these sensor data, we fuse them with WiFi ToF measurements and improve the localization results from pre-existing systems.

Acknowledgement

Firstly, I would like to thank Prof. Dr. Plattner from Communication System Group of the Computer Engineering and Networks Laboratory (TIK) for giving me this great opportunity to work on this interesting Master Thesis on indoor localization. I would also like to thank my advisors Prof. Dr. Domenico Giustiniano and Dr. Vincent Lenders for helping me through the process. Their advises were very helpful and got me through many hurdles. Lastly I would also like to thank Monika Graf who helped me with all the crucial information regarding the ToF system at Armasuisse, and Aymen Fakhreddine for helping me with some of the coding aspects.

Contents

1	Introduction	11
1.1	Motivation	11
1.2	Related Work	12
1.2.1	Time-of-Flight System	12
1.2.2	Timing Imprecision	12
1.2.3	Multipath	13
1.2.4	Multilateration	14
1.3	Assignment	14
1.4	Overview	14
2	Inertial Sensors	15
2.1	Problem	15
2.2	Accelerometer	15
2.3	Magnetometer	16
2.3.1	Magnetic Declination	16
2.3.2	Magnetic Offset	17
2.3.3	Phone Orientation	17
2.4	Orientation and World Coordinates	18
2.5	Dead Reckoning	19
2.6	Step Counting	19
2.6.1	Gait Profile	20
2.6.2	Step Counter	21
2.7	Gyroscope	21
3	Smartphone App	23
3.1	Data Collection	23
3.1.1	Gait Profile	23
3.1.2	GPS	23
3.2	Interface	24
4	Evaluation	27
4.1	Speed	27
4.1.1	Setup	27
4.1.2	Results	28
4.2	Direction	30
4.2.1	Setup	30
4.2.2	Results	30
4.3	Step Counter	34
4.4	Calibration	36
5	Time-of-Flight System and Sensors Fusion	39
5.1	Kalman Filter	39
5.2	Results and Error Analysis	40
5.2.1	Error Analysis	40
5.2.2	Outliers	40
5.2.3	Ranging Error	41
5.2.4	Calibration Error	41

5.3 Simulation	43
6 Conclusion and Outlook	47
6.1 Conclusion	47
6.2 Outlook	47
A Additional Grpaphs	49
A.1 Outdoor Test: Gait Profile (sensor) vs GPS Speeds vs Ground Truth. Individual Smoothing Factor	49
A.2 Direction Error Test (02-06)	54
A.3 Distance Error Test (02-06)	54
B Additional Tables	63
C Original Assignment	65

List of Figures

1.1 ToF: Multilateraion	11
1.2 Principles of WiFi ToF echo technique	12
2.1 Acceleration Magnitude vs Elapsed Time: Accelerometer Data	16
2.2 Magnetic Offset: Stationary Phone	16
2.3 Magnetic Declination: Armasuisse, Thun	17
2.4 Phone Orientation	18
2.5 Accelerometer and Gravity	18
2.6 Phone Axes (black) vs World Coordinates (red)	19
2.7 Dead Reckoning. Ground Truth (black) versus Estimated Positions (grey)	20
2.8 Hip Bounce	21
3.1 Bosch PLR 50	23
3.2 Gait Profile of the Test User	24
3.3 Velocity Comparison, DoP and Number of Satellites	25
3.4 Interface of the App	25
4.1 Map and Waypoints for Controlled Test: Evaluation of Gait Profile and GPS Speeds	27
4.2 Gait Profile (sensor) vs GPS Speeds vs Ground Truth	28
4.3 ECDF: GPS Speed	28
4.4 ECDF: Gait Profile (Sensor)	29
4.5 Gait Profile (sensor) vs GPS Speeds (MSE)	29
4.6 Testbed I at Armasuisse Thun, Switzerland	30
4.7 Testbed at Armasuisse Thun, Switzerland (Satellite Map). Building Oreintation Measurement	31
4.8 Direction Comparison. Sesnor vs Ground Truth	32
4.9 ECDF: Direction Comparison. Sensor vs Ground Truth	32
4.10 MSE: Sensor Directional Error with Exponential Smoothing	33
4.11 Direction Plot: Raw Compass Data vs Averaging vs Ground Truth (Test 05)	33
4.12 Direction Plot: Raw Compass Data vs Averaging vs Ground Truth (Test 06)	34
4.13 ECDF Direction: Raw Compass Data vs Averaging vs Ground Truth (Test 05)	34
4.14 ECDF Direction: Raw Compass Data vs Averaging vs Ground Truth (Test 06)	35
4.15 Accelerometer (Test 06)	35
4.16 Gait Profile: Updated with 200 Steps using GPS Data	37
4.17 Gait Profile: Updated with 1000 Steps using GPS Data	37
5.1 Flow Chart: Kalman Filter	39
5.2 ECDF: Distance Error (Test 01)	41
5.3 Distance Error vs Elapsed Time (Test 01)	41
5.4 ECDF: Distance Error (Test 01). Thompson Tau: 1000 iterations	42
5.5 Ranging Error vs Elapsed Time (Test 01)	42
5.6 ECDF: Ranging Error (Test 01)	43
5.7 Recalibration Values Distribution	43
5.8 ECDF: Distance Error with New Median Calibration Values (Test 01)	44
5.9 Y Coordinates. Ground Truth vs ToF Measurement and Kalman Filtered (Test 01)	44
5.10 Distance Error ECDF. Simulation (Iterations: 10000). Measurement Noise: 3.47 m	45

List of Tables

1.1	Calibration Values for Samsung Galaxy S5	13
4.1	Way Point Coordinates of Testbed 1	31
5.1	Kalman Filter: Fused Sensor Data and ToF Measurements	40
5.2	Calibration Values: Original versus New	44
B.1	Way Point Coordinates of Testbed 2	63

Chapter 1

Introduction

Localization using the Time-of-Flight (ToF) of RF signals is today one of the most popular techniques to track moving objects. One prominent example is the Global Positioning System, which exploits the differences in signal propagation times between different satellites to provide location services to mobile devices on earth. While the ToF technique has been highly successful in these outdoor domains, its application in WiFi-based indoor localization remains to be a challenge. The research community has instead focused more intensively on other approaches such as signal strength [1, 2, 3, 4, 5], and the angle of arrival [6, 7, 8]. Previous efforts to make use of the ToF of WiFi signals have reported relatively inaccurate results both in static positioning [9, 10, 11, 12, 13] and mobile tracking conditions [14, 15].

In this thesis, we aim to improve the accuracy of ToF positioning by fusing it with data of inertial sensors such as accelerometer and gyroscopes, which are largely available in on-the-shelf smartphones [16, 17, 18]. Similar work has been done in [15], which however requires sophisticated APs. This hinders the adoption of these techniques for wide-spread and low-cost deployments. Therefore, one of the goals of the project is to move from infrastructure-centric system to device-centric system. Since a smartphone has functionalities to geo-locate itself, we expect the fusion of such functions with the infrastructure should help increase positioning accuracy. This includes the use of GPS for outdoor environment or WiFi ToF system for indoor, depending on availability.

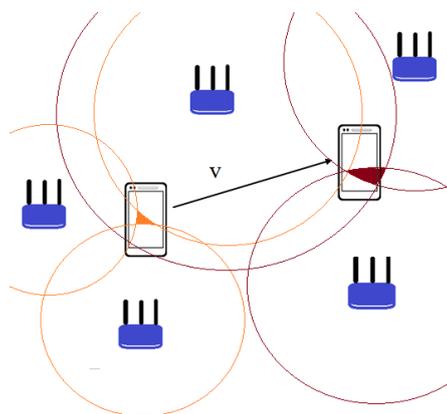


Figure 1.1: ToF: Multilateration

1.1 Motivation

In recent research [19], it was demonstrated that ToF based indoor tracking running exclusively on commercial on-the-shelf (COTS) WiFi hardware is feasible. In April 2014, at the IEEE/ACM IPSN 2014 indoor localization competition in Berlin which involved 22 independent approaches from different research teams, this system was able to achieve a 3.47m average positioning error. While such a two-way ranging method has proven to work, it has two main drawbacks. First, it produces a lot of traffic since every access points needs to inject its own packets to determine the ToF to the targets. Therefore when multiple targets are requesting packets simultaneously, the overhead can get quite significant. Secondly, the Short Interframe Space (SIFS) in many WiFi devices produces jitter in its timing accuracy and therefore distance estimation based on timing precision becomes difficult.

In the first stage, the ToF system will compute the initial position of the smart phone as in [19]. In the second stage, the smart phone uses its inertial sensors to compute its velocity and heading which determines its consecutive positions. By fusing ToF with inertial sensors, we may be able

to reduce the traffic overhead, such that it only queries when ToF information is strictly needed. This means that when the sensor based positioning system is suffering from dead-reckoning, then the measurements from the ToF system could be used, such as by means of applying Kalman Filter [15], to correct it, and therefore increase its accuracy.

1.2 Related Work

Previously, we introduced two similar projects that use inertial sensors for indoor localization. In the ZEE paper [18], the restriction is the necessity of a priori knowledge of the environment. In order to improve accuracy, a particle filter is applied to the measurements according to the floor plan of the test environment, and therefore reducing the possibilities of error (e.g. walking through walls). This method, despite producing an error as low as 2m, is only feasible when a relatively long period of tracking is performed. In our thesis, we aim to account for both long and short periods of tests, as well as a system that requires no prior knowledge of the floor plan. In the SAIL paper [15], it was shown that while using only one sophisticated access point and inertial sensors (accelerometer, gyroscope and magnetometer), a distance error of 0.63m was achieved on a Samsung Galaxy S4. Similarly in the testbed at Armasuisse, the eight COTS access points can provide position estimate in combination with the inertial sensors on the phone.

1.2.1 Time-of-Flight System

The Time-of-Flight system implemented in the Armasuisse testbed is based on the work in [19]. The testbed consists of eight anchors. All ToF measurement data can be found in the server's MySQL. A visual representation of the estimated position of the target device is also available. The concept of WiFi ToF ranging technique uses the signal travelling time between an access point and the target, and therefore estimates the distances by multiplying by the speed of light ($3 * 10^8 m/s$). However, in the real world, this technique suffers from noises coming from two main sources.

1.2.2 Timing Imprecision

The first source comes from the timing imprecision in COTS WiFi devices. For example, for a WiFi router, a specified lag time called Short Interframe Space (SIFS) denotes the time between the reception of a DATA and the transmission of an acknowledgement (ACK) as a fixed interval. This means that the SIFS shall be subtracted from the two way ranging total ToF in the calculation of the distance between the devices (Fig.1.2). Therefore, the measured time between the

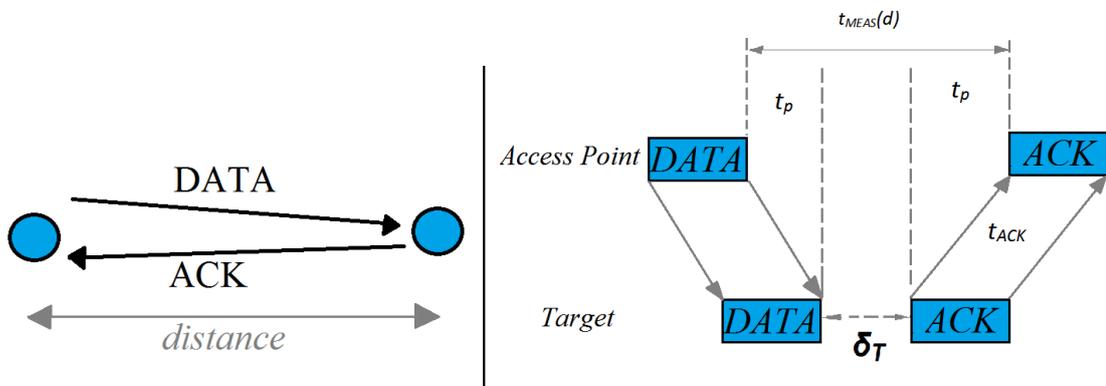


Figure 1.2: Principles of WiFi ToF echo technique

access point and its target are given by:

$$t_{MEAS}(d) = 2 * t_p + t_{ACK} + \delta_T \quad (1.1)$$

Then the corresponding ToF is

$$t_{TOF}(d) = \frac{t_{MEAS}(d) - t_{ACK} - \delta_T}{2} \quad (1.2)$$

$$\hat{d} = c * t_{TOF}(d) \quad (1.3)$$

Here \hat{d} denotes the estimated distance based on measured time and SIFS specification, while c is speed of light. However, this SIFS comes with a tolerance, and in many cases cause a huge deviation from the true distance ($\delta_T = SIFS + tolerance$). For example, a tolerance of even $1\mu s$ translates to an error of $300m$. In order to reduce the contribution of this noise, the paper [19] employed a filtering technique that uses chipset-dependent calibration. The access points in the testbed in Armasuisse use the Broadcom AirForce54G 4318 Wireless Card and the measurements are done in IEEE 802.11b standard. Conceptually, such chipset calibration is done simply by measuring the processing delay and ToF (in clock cycles) for known distances between two chipsets. Since the processing delay also depends on the PHY rate, the same calibration process is done for each of the four rates in Table 1.1. The values in the second column denote the number of clock cycles needed for the ToF measurement to reach a $0m$ distance based on the rate used. During the calibration process, the phone is placed on a location in the testbed where its distances from all eight anchors are known. Each anchor would then send multiple packets of DATA to the smartphone and measure the total time in clock cycles. Lastly, by subtracting the known distance from the measurement, the reference $0m$ time per rate was found. Since the clock rate for the chipset is $88MHz$, the measured distance is

Rate (Mb/s)	1	2	5.5	11
ref ₀ (clock cycles)	27908	22994	19932	19040

Table 1.1: Calibration Values for Samsung Galaxy S5

given by:

$$\hat{d} = \frac{c * (ToF_{MEAS} - ref_0)}{2 * 88MHz} \quad (1.4)$$

$$\text{or } = (ToF_{MEAS} - ref_0) * 1.7045m$$

1.2.3 Multipath

The second source of noise comes from multipath. Intuitively, we can see that the distribution of ToF measurements would be statistically left-skewed if multipath reflections have occurred in the indoor environment. This is because the reflected paths would cause an over-estimation of the ToF compared to its direct-path counterpart. To overcome this contribution of noise, they [19] designed an adaptive filter. Using the large samples of the median ToF measurements, this filter is able to provide a statistical profile specific to the Testbed at Armasuisse. Such a filter is characterized by a linear model (1.5), which given the median ToF measurement, the user can find the optimal percentile that minimizes distance error.

$$y = -0.341668 + 21.384868 * x(\text{clock cycles}) \quad (1.5)$$

where y : Median ToF

x : Optimal Percentile

Although a median of large sample size of measurements should be statistically more accurate than a smaller one, the paper found that using this linear model, even a sample size as small as ten is sufficient.

1.2.4 Multilateration

While the previous filters and linear model provide the distances between the target and the anchors, multilateration is used. Since the estimated position of the target is only visually displayed on the server's website and not available in the MySQL data, we used simple multilateration MATLAB codes to compute the positions offline. In terms of measurement data, it takes roughly $800ms$ for all anchors to measure using all of the rates, providing that they are reachable. This means that, taking $800ms$ as an interval, we could collect all available data from all anchors to estimate the position of the target. On the other hand, in our offline calculation, we also explored the effect of reducing the time interval and therefore using less anchor measurements (minimum of three).

1.3 Assignment

1. Study the current WiFi based indoor localization system developed by advisors [19] and the literature on localization using smartphone sensors [16, 17, 18].
2. Implement a smartphone app running on Android devices. The first step will be to gather data from the internal inertial sensors and implement appropriate smoothing filters of the noisy data.
3. Evaluate the accuracies of sensor data, as well as selecting the appropriate sensors to use where power consumption should be minimized.
4. Fuse collected sensor data (speed, direction) with the ToF system in Armasuisse, by means of Kalman Filter.
5. Evaluate the accuracy of the fused system, particularly if it improves beyond the accuracy achieved without the use of sensor data, as well as compare to a positioning using only sensor data.

1.4 Overview

Chapter 2 introduces the various inertial sensors available in COTS smartphones and explains their potential to supply data information needed for our system. Chapter 3 presents the smartphone app developed in order to collect the appropriate sensor data, and presents its interface and design. Chapter 4 evaluates the data collected from the inertial sensors. Chapter 5 describes the fusion process of the sensor data and the WiFi ToF localization system, evaluates its accuracy, and describes measures carried out to improve error. Chapter 6 concludes our report and presents possible future development based on our findings.

Chapter 2

Inertial Sensors

2.1 Problem

In order to use the inertial sensors to compute relevant data (heading, velocity etc), we must first evaluate their accuracies. In terms of resources, there exists very little information on the sensor accuracies of Android phones. Furthermore, these accuracies are model dependant and we can only evaluate based on the particular model we have on hand - a Samsung Galaxy S5.

2.2 Accelerometer

The accelerometer in the Samsung Galaxy S5 (as well as various other smartphone models) measures acceleration experienced along the three axes of the phone (X, Y and Z, see Fig. 2.6), as well as the offset due to gravity. Thus, when the phone is placed stationary, its Z axis should ideally measure 9.8 m/s^2 . On the contrary, if the phone is in free fall, the magnitude of acceleration upward would measure 0 m/s^2 . Let us first assume that in the direction of the motion (e.g. user walking), the magnitude of acceleration measured is found. Mathematically, we can calculate the distance travelled with the kinematic equations:

$$v = \int a dt$$

$$d = \int v dt$$

where a = Acceleration

v = Velocity

d = Distacne

t = Time

In order to obtain velocities, one can simply integrate accleration over the time interval between samples. Alternatively, it can be integrated again to obtain average distance. This implies that the accelerometer and an accurate time synchronisation are enough to obtain the necessary information to compute distances travelled. However, in practice, an accelerometer produces noisy measurements. Fig. 2.1 is an example of the accelerometer data recorded from the Galaxy S5. Note that since the sensors report timestamps as established nanoseconds since the application started. In order to utilize the highest precision possible, we logged all sensor data in nanoseconds. During this test, the phone is placed stationary on a flat surface. As seen from the figure, the mean value has an offset of roughly 0.53 m/s^2 . Additionally, very noisy data can be observed (in this example, a variance of $3.2 * 10^{-4} \text{ m}^2/\text{s}^4$). Although the noise figure does not appear to be significant in acceleration, it is indeed so after integrations. For example, if we integrate over time once while removing the offset of 9.8 m/s due to gravity, we obtain an average velocity of -8.5964 m/s over a time interval of 16.3049 s . This phenomenon is called integration drift and other corrective measures need to be implemented.

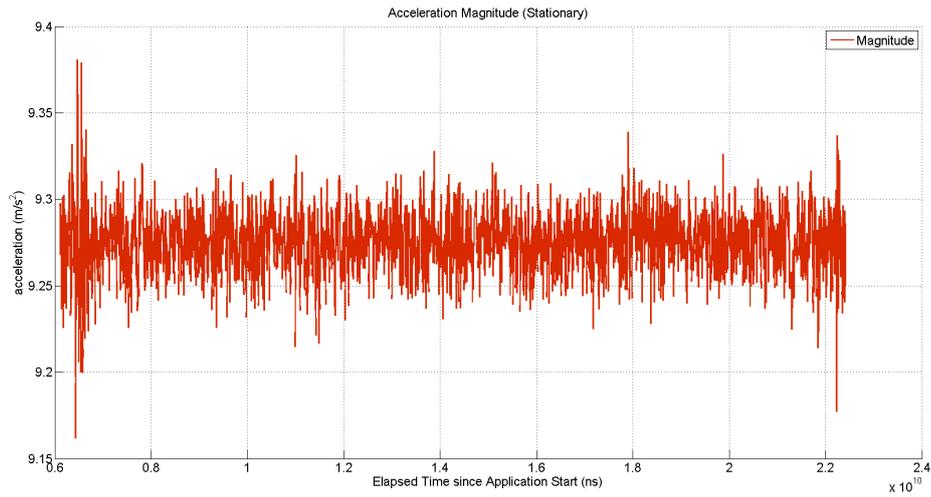


Figure 2.1: Acceleration Magnitude vs Elapsed Time: Accelerometer Data

2.3 Magnetometer

Another sensor widely available in most latest smartphone models is the magnetometer (or compass). It measures the earth magnetic field at a point in space. In other words, it gives a directional measurement in degrees with respect to the north pole.

2.3.1 Magnetic Declination

The first considerable error that comes with the magnetometer is the magnetic declination. This is the angle on the horizontal plane (parallel to earth's surface) between magnetic north (i.e. when a compass needle reads 0°) and true north (the direction along a meridian towards the geographic North Pole). Since the world's magnetic field changes with time, this error is therefore geographically and temporally dependent. For our tests performed at Armasuisse, the declination is 1.8° , while the small yearly increase of 0.12° is neglected. This data is obtained from National Centers for Environmental Information [20] (see Fig. 2.3).

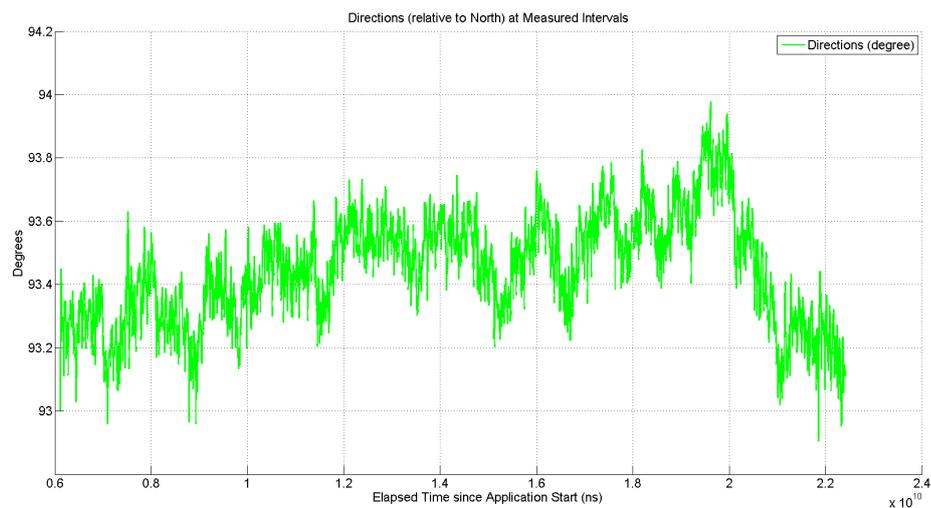


Figure 2.2: Magnetic Offset: Stationary Phone

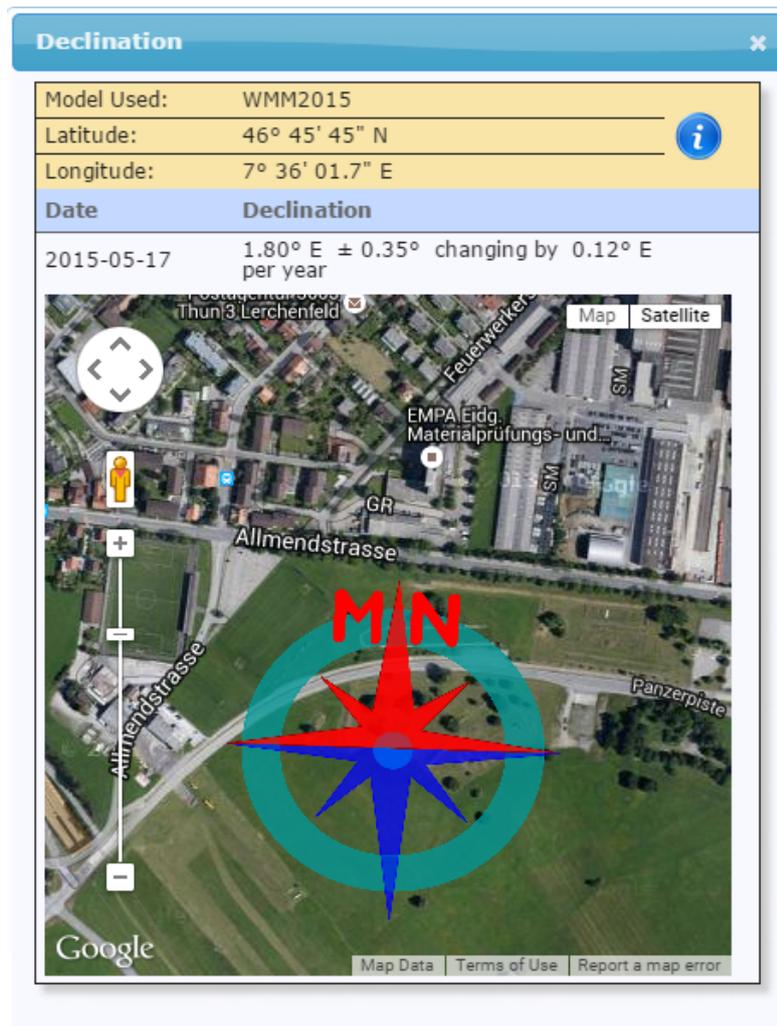


Figure 2.3: Magnetic Declination: Armasuisse, Thun

2.3.2 Magnetic Offset

The second considerable error is the magnetic offset. This offset is present, in particular indoor environment, where magnetic materials (e.g., metal and electronic devices) in close proximity of the mobile phone disturb its perception of "north" [21]. Unlike magnetic declination, magnetic offset is not constant with time, as can be seen in Fig. 2.2. The variation of this offset depends on the the material in proximity of the device. Electronic appliances such as TV with varying EM interference are amongst the common indoor items found even in common households. In this scenario, the mobile phone is again placed stationary and pointed at the same direction, but the magnetic offset causes a time-variant error in the range of 5.4057° . Since each indoor environment would have its uniquely different amount of magnetic materials, it is not possible to denote a constant value to compensate for this offset (as opposed to the case of magnetic declination).

2.3.3 Phone Orientation

The last error comes from the orientation of the phone. The SAIL paper [15] evaluated different poses of the user carrying the mobile phone. This includes:

1. The phone is held in one hand while facing the user
2. The phone is held against the ear

3. The phone is held in one hand while swinging along with the walking motion
4. The phone is placed in a pocket
5. The phone is placed in a purse

The paper also evaluated situations when one pose transitions to another. This is done by incorporating gyroscope and accelerometers to detect pose changes, understanding that using more sensors will inevitably increase measurement errors. While the SAIL paper can detect pose changes accurately (> 95%), other abrupt motional changes during pose changes may not be reliably detected. For example, a sudden sharp turn or when the user stops walking may not be detected if the user simultaneously moves the phone away from the ear and place the phone in the purse, since both types of motion changes require the same sensors. Therefore, in order to simplify our tests, we chose position one and further restrict that the phone is in its "portrait" mode (middle position in Fig. 2.4). Another benefit for selecting such a restriction will be described in the next section (2.4).

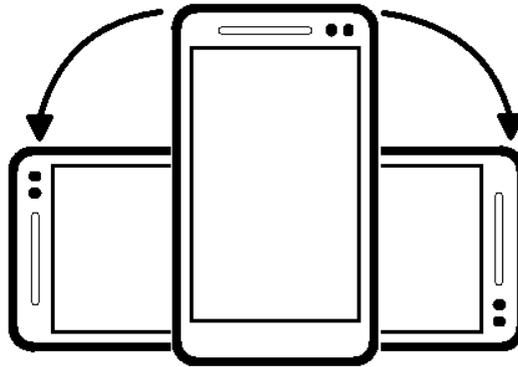


Figure 2.4: Phone Orientation

2.4 Orientation and World Coordinates

In order to determine the heading of the user, we utilize compass data as described in 2.3. However, even in the absence of magnetic offset, the axis with which the compass computes its heading must be the same as that of the user. The smartphone compass uses its Y-axis (see Fig. 2.6) to measure its magnetic difference from the north. The benefit of the restriction of orientation described in 2.3 solves the problem of when the top of the phone is pointing $\pm 90^\circ$. However, it is not feasible to further restrict the user to hold the phone horizontally. Even when the user is willing to do so, there is no guarantee that his/her hand is steady enough to ensure it is perfectly parallel to the ground. This means we cannot reduce the tilted angle θ to 0° .

Let us assume that, as illustrated in Fig. 2.6, the user is holding the phone steadily at an unknown angle θ with respect to the ground. Using accelerometer data:

$$\theta = 90^\circ - \cos^{-1}\left(\frac{g}{a_y}\right)$$

In other words, we can use accelerometer value a_y to compute θ at any time instance and therefore use this angle to find the compass measurement along the world's XY-plane. Conveniently, Android has built-in functions (`getRotationMatrix()` and `getOrientation()`) that perform similar operations. `getRotationMatrix` computes a rotational matrix using accelerometer inputs for computing the phone's orientation compared to the world's coordinates. It then uses

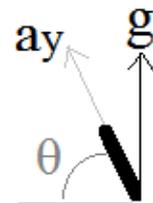


Figure 2.5: Accelerometer and Gravity

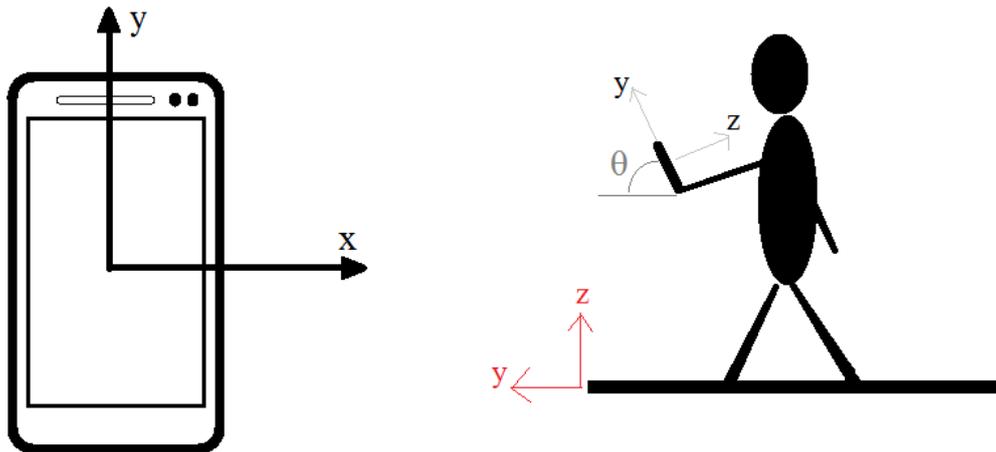


Figure 2.6: Phone Axes (black) vs World Coordinates (red)

magnetometer values to also compute its heading. For example, if the X-axis of the phone is pointing towards East and the Y-axis is pointing towards North, then R is the Identity Matrix. `getOrientation` then returns the value `myHeading` where it represents the angle between the Y-axis of the phone and the north (Azimuth).

```
\lstlistingname{Java Code Snippet: Re-Orientation of Smartphone Axes to
  align with World Coordinates}
boolean success = SensorManager.getRotationMatrix(float[] R, float[] I,
  float[] accelerometer_values, float[] compass_values)

  if (success) {
    SensorManager.getOrientation(R, myHeading);
    float azimuthInRadians = obj.myHeading[0]; // Radians between
      Y-Axis and North
```

Since this reduces the number of necessary sensors to two (accelerometer and compass), we could then evaluate their contributing and combined error.

2.5 Dead Reckoning

Information on direction is necessary for navigation and positioning. While speed and time give information of the distance travelled, on the 2D plane it is necessary to know the direction in order to locate the new X- and Y- coordinates as compared to the previous ones (Fig. 2.7). As the figure shows, when the estimate of the direction is incorrect ($\hat{\theta} > \theta$), the new fix calculated (\hat{x}_k, \hat{y}_k) is also wrong. The same applies when the velocity or timing estimation is incorrect. Consequent estimations based on the wrong new fixes would accumulate the error as seen from the picture. This phenomenon is called dead reckoning. This is an issue common to positioning based only on inertial sensors. Therefore, in our thesis, we aim to correct this error by using ToF system to estimate these fixes. Although the ToF fixes would also produce error, each measurement is independent to the previous ones and therefore does not suffer from dead reckoning.

2.6 Step Counting

Due to the integration drift mentioned earlier, we need to implement other methods to compute kinematic data (velocity, distance). As opposed to integration, the ZEE paper [18] proposes to

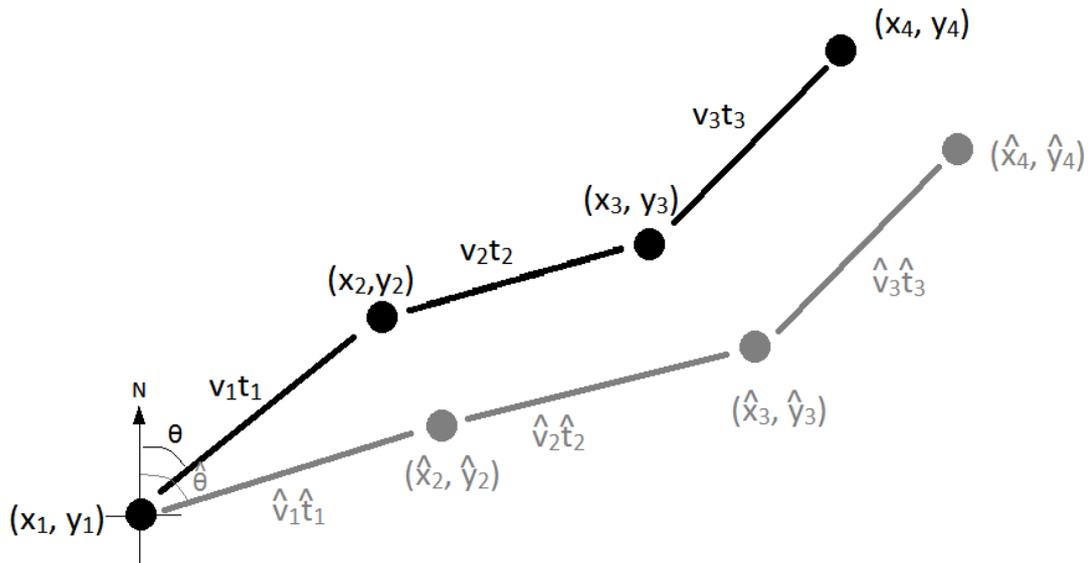


Figure 2.7: Dead Reckoning. Ground Truth (black) versus Estimated Positions (grey)

use auto-correlation function to compute velocity. The intuition behind this methodology is that if the user is walking, the autocorrelation of accelerometer data will peak periodically relative to the step speed of walker. On the other hand, a sudden and large increase of the peak time intervals could imply that the walker has stopped. The ability to accurately detect stops is also critical in terms of tackling one of the the issues attributed to dead reckoning. When the device is confident that its speed is 0 m/s then a new fix will not be needed until the walking begins again. However, the drawback of this implementation is that it does not measure the speed in terms of distance. Although this is compensated for with the particle filtering in the ZEE paper, our thesis aims to avoid using a priori knowledge of the floor plan. Additionally, particle filters are computationally expensive. A rather novel idea is presented in SAIL as described in the next subsection.

2.6.1 Gait Profile

The human gait is the locomotion achieved by the movements of the limbs. In terms of walking, this refers to the movements of hips, legs, feet and the joints. The SAIL paper [15] proposes that it is possible to compute walking speed in terms of the characteristics of human gait. This characteristic is called the bounce factor and can uniquely define the walking gait characteristics of each individual. The paper claims that the hip movement, as illustrated in Fig. 2.8, is linearly related to the step size and to the step frequency of the individual (the faster the person walks, the more their hip bounces).

It further claims that step lengths tend to have correlations between users with similar physical builds. When a new user uses its application, a few measurements are computed (e.g. bounce and step frequency) and then it selects a profile from collected database containing measurements of other users that best approximates this current user. While this method is feasible with a large available database, our thesis does not have access to it. This means our option is to select a specific user and test if these linear relationships are valid. Additionally, we neglected the bounce factor in our calculation and only focused on the linear relationship between step frequency and step size (2.1), for which we call the Gait Profile. In this equation, the left hand side is the walking speed, the first term on the right hand side is step frequency and the last term is the inverse of step size. The experiment and evaluation of the Gait Profile

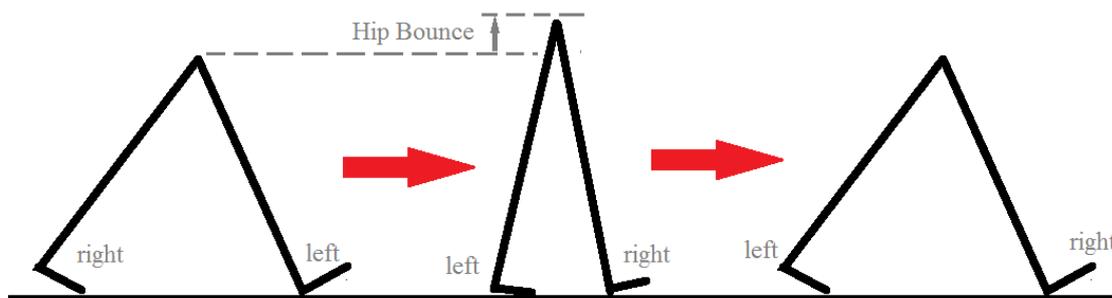


Figure 2.8: Hip Bounce

will be discussed in the next Chapter (??).

$$\frac{Distance}{Time} = \frac{Steps}{Time} * \frac{Distance}{Steps} \quad (2.1)$$

2.6.2 Step Counter

Now that we are able to use Gait Profile to calculate speed, the input we need is the step frequency. Android offers four levels of sensor input delay and in descending order they are:

1. public static final int SENSOR_DELAY_FASTEST
2. public static final int SENSOR_DELAY_GAME
3. public static final int SENSOR_DELAY_NORMAL
4. public static final int SENSOR_DELAY_UI

There are no official resource documenting precisely how one delay setting compares to another. However, from our experiment when our sensors are set to report in the "FASTEST" range, we consistently find the delay between consecutive measurements to have a mean of $5ms$. As it will be seen later in the test conducted at Armasuisse, each position estimation takes around $800ms$ to $1s$, and therefore our sensor reporting delay is sufficiently fast.

Since Android API 4.4 was released in 2013, two new types of sensors were added: Step Counter and Step Detection. According to official documents [22]:

A sensor of this type returns the number of steps taken by the user since the last reboot while activated. The value is returned as a float (with the fractional part set to zero) and is reset to zero only on a system reboot. The timestamp of the event is set to the time when the last step for that event was taken. This sensor is implemented in hardware and is expected to be low power.

Instead of performing real time computation of steps (e.g. auto-correlation of accelerometer datas for all three axes with unknown time delay τ in the ZEE paper), step counter values and their corresponding time stamps are sufficient to calculate step frequency.

2.7 Gyroscope

The original reason for using the gyroscope was such that it adds information to the turning of the user. However, as discussed earlier, using more sensors inevitably introduces more errors. Furthermore, as we have restricted the pose of how the user holds the phone, this becomes unnecessary. Furthermore, part of our goals is to reduce power consumption on the smartphone and therefore a reduction in the amount of sensors used is beneficial.

Chapter 3

Smartphone App

There are two main functions of the smartphone app.

1. Collect and log relevant sensor data for fusion with the ToF system
2. Provide an interface for user interaction

3.1 Data Collection

3.1.1 Gait Profile

In order to build the Gait Profile for the user, we designed an environment where we can know the ground truth. This environment was a relatively straight path outdoor, where the user would walk at different distances at various speeds. It is designed in order to collect data on step sizes ($\frac{\text{Distance}}{\text{Steps}}$) and step frequency ($\frac{\text{Steps}}{\text{Time}}$) which are the data needed for the Gait Profile. The distances are measured using a BOSCH Digital laser measure PLR 50 (Fig.3.1). Using linear regression model on the collected data, we obtained the following Gait Profile (Fig.3.2).

The equation for the linear model is:

$$\text{Step Size} = 0.5254 * \text{Step Frequency} - 0.0967 \quad (3.1)$$

And therefore, using only step counter, we first compute Step Frequency. Then we find Step Size from the linear model. Lastly we can obtain the walking speed (v):

$$v = \text{Step Size} * \text{Step Frequency} \quad (3.2)$$

$$\text{or } v = (0.5254 * \text{Step Frequency} - 0.0967) * \text{Step Frequency} \quad (3.3)$$

One of the goals of the project was to migrate indoor localization from infrastructure-centric system to device-centric system. Providing that the gait profile accurately computed the walking speed of the user, the same application could not be used by a user with different physical build. This meant the linear regression model needed new calibration values and this was not ideal. Specifically, asking a smartphone user to precisely measure and record his/her walking speed to rebuild a new linear model was not the best solution. To tackle this issue, we explored the possibility of retrieving speed information using GPS system.

3.1.2 GPS

Android can retrieve various information related to each GPS location fix from what is known as the NMEA Message. This was done by setting up a listener in our app with the code:



Figure 3.1: Bosch PLR 50

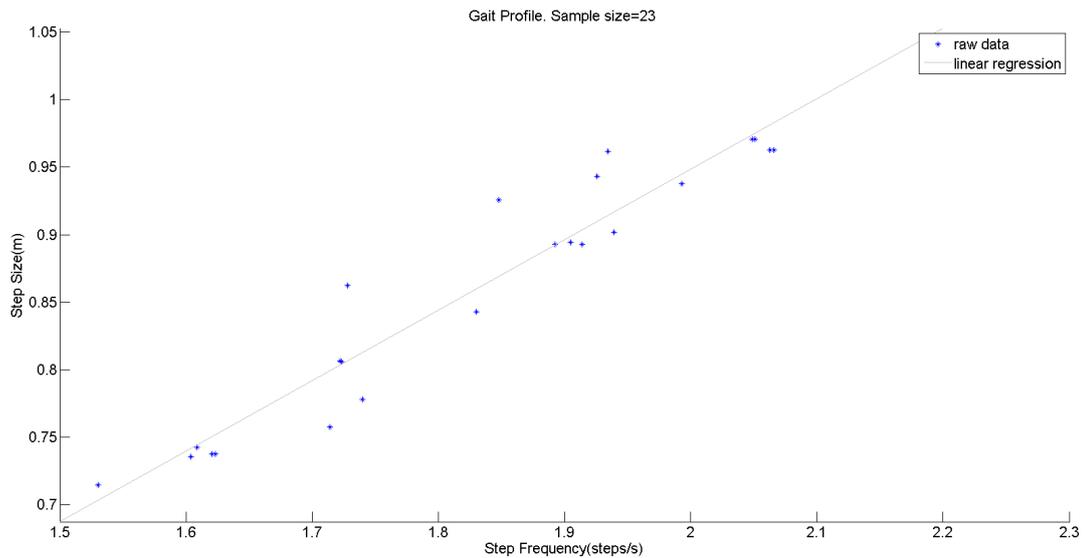


Figure 3.2: Gait Profile of the Test User

```
GpsStatus.NmeaListener()
```

Each NMEA Message carries a header which indicates the specific information in the message. For the purposes of our tests, the relevant informations are:

- Time Stamps
- Number of Satellites used. (headers: "\$GPGSA" or "\$GNGSA")
- Dilution of Precision. (headers: "\$GPGSA" or "\$GNGSA")
- Target Speed Reported by GPS. (headers: "\$GPVTG" or "\$GNVTG")

The time stamps are found when the NmeaListener receives the message. The first idea is to make use of Dilution of Precision in the GPS system [23]. The DOP value measures the accuracy of GPS positioning accuracy rating (where $DOP < 1$: Ideal. $DOP > 20$: Poor). However, our Gait Profile is a measure of speed, while DOP only indicates the accuracy of positioning and it does not directly differentiate into speed values (and accuracies). Evaluating whether DOP can translate into a figure to evaluate GPS speed accuracy or not is a non-trivial task and is beyond the scope of our thesis. Since GPS speed is measured based on the Doppler Effect [24], the number of GPS satellites used for the measurement could be another indicator of the accuracy. For one of our outdoor controlled measurements, we plotted the following (Fig.3.3)

As can be seen in the figure, the number of satellites used to calculate the GPS Reported Speed are mostly more than seven with only one occasion of four. On the other hand, the DoP values are all between 0.9 and 1.2. These numbers do not give us enough information to differentiate a good measurement from a bad one.

3.2 Interface

In order to explain how we designed the application to collect sensor data in real time, let us look at the interface design (Fig.3.4). The reported values on the interface do not necessarily reflect the sampling frequency of the sensor. In particular, direction, acceleration and gravity values are only updated every second. The acceleration value differs from the gravity value in that gravity will always return a magnitude ("M9.721" in this example) close to earth's gravity 9.8 m/s^2 (due to sensor noise) where as acceleration will only have a magnitude close to gravity when the phone is being held steady (in this case, on the hand) and the user is not moving.

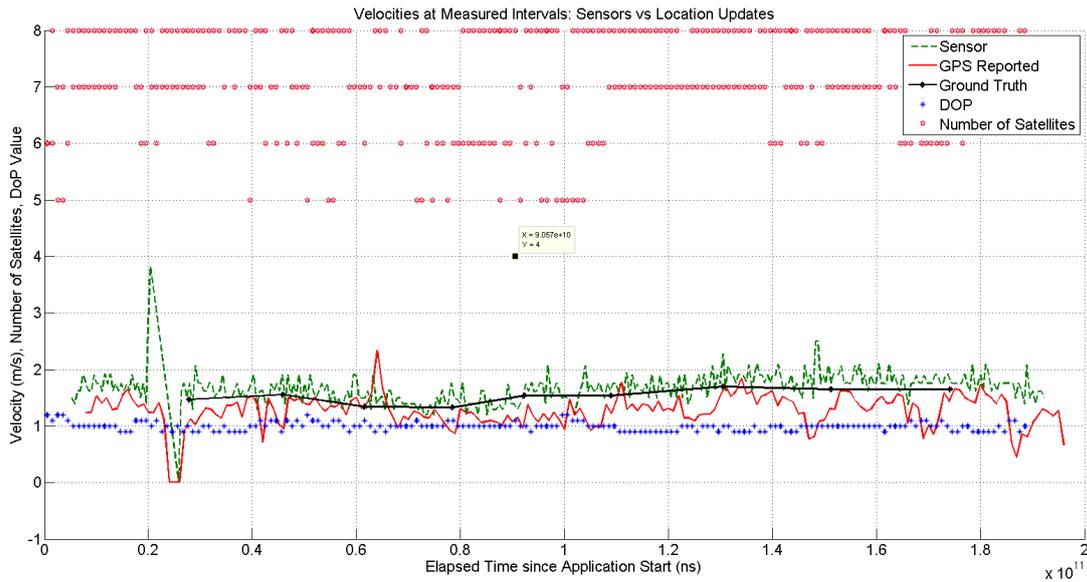


Figure 3.3: Velocity Comparison, DoP and Number of Satellites

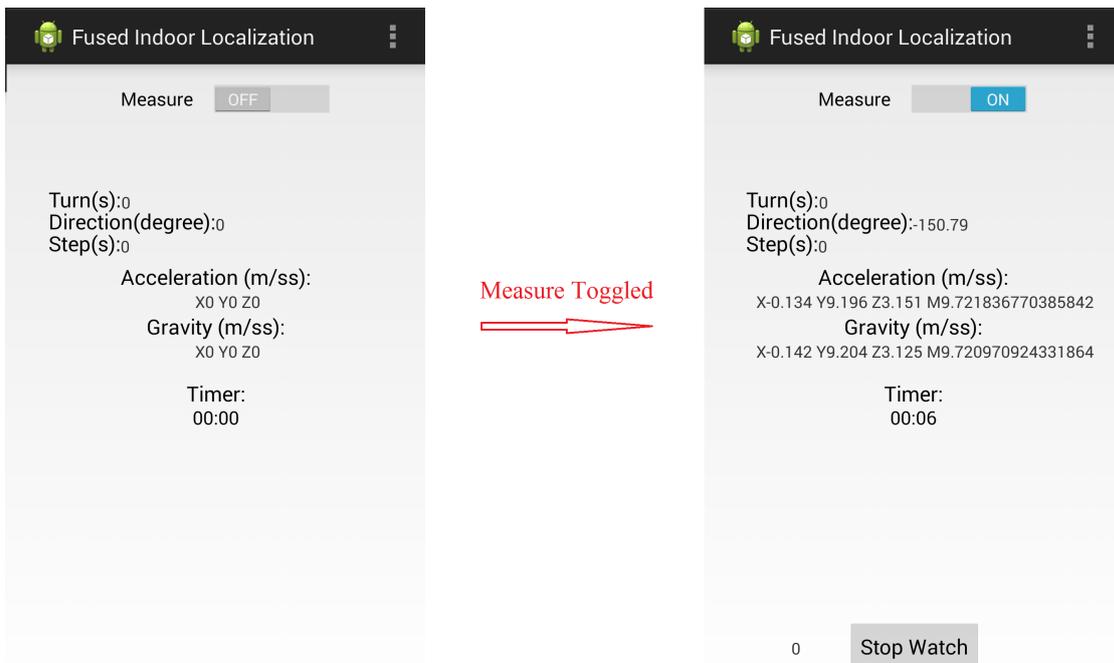


Figure 3.4: Interface of the App

In order to evaluate the accuracy of the step counter, not only do we need to log its data (time stamps and counted step values), we also need the ground truth to compare it with. When the "Measure" button is toggled, the app begins to listen for changes from all the sensors and would update the values on the display accordingly. The "Stop Watch" button would also appear, which logs the elapsed time and accumulated steps. While these interface values are only for reference, the application also saves all the logs into csv files when "Measure" is toggled off. Note that the field "Turns" is a measure using gyroscope and compass. Whenever the compass indicates a directional change larger than a threshold (in our last implementation the threshold

is set to 10) then the app checks if the previous angular speed change (within 500 ms prior) experienced by the gyroscope is larger than another threshold (0.52 rad/s), then it indicates a turn was made. However, as mentioned earlier, directional compass values were sufficient and "Turns" were no longer used in our implementation.

Chapter 4

Evaluation

In order to fuse the sensor data collected with the WiFi ToF system, we evaluated their accuracy and whether each set of data is necessary for the fusion process.

4.1 Speed

4.1.1 Setup

From the velocity plots (using the app's sensors and gait profile versus GPS reported speed) in Fig. 3.3 (preliminary walking experiment), noisy data can be observed, especially in the case of sensor based velocity. Therefore, aside from conducting experiments to evaluate velocity accuracies versus ground truth, we also evaluated if an Exponential Smoothing filter would improve accuracy.

The controlled experiment is done by marking specific waypoints on an outdoor path. A satellite

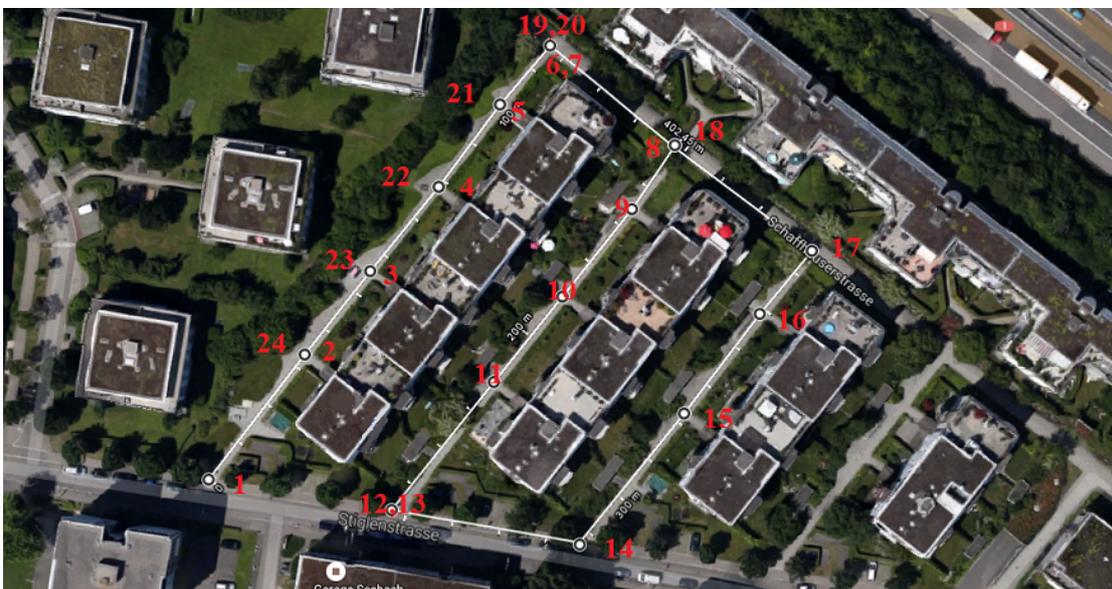


Figure 4.1: Map and Waypoints for Controlled Test: Evaluation of Gait Profile and GPS Speeds

map from Google Map shows this outdoor location where we conducted the test (Fig. 4.1). The red numbers indicated on the maps are the waypoints of this particular path that the user walked. Note that a tuple (e.g. 19,20) indicates that the user stopped at that particular waypoint. Distances between waypoints are measured using distance a measurement tool from Google Map. Ther user, whose Gait Profile was installed in the app, walked from waypoint 1 to waypoint 20 and logged the step counts and time stamps by clicking "Stop Watch" on the app.

4.1.2 Results

Figure 4.2 shows the results of this test. While analysing the test results, we also applied exponential function with smoothing factor $\alpha : \{0.1, 0.2, \dots, 0.9\}$. Plots with individual smoothing factor can be found in Appendix A.1. In order to further investigate the effect of smoothing, we plotted the Empirical Cumulative Distribution Function (ECDF) for both Sensor and GPS velocities against ground truth, as shown in Figure 4.4 and Figure 4.3, as well as their corresponding Minimum Squared Error (MSE) in Figure 4.5.

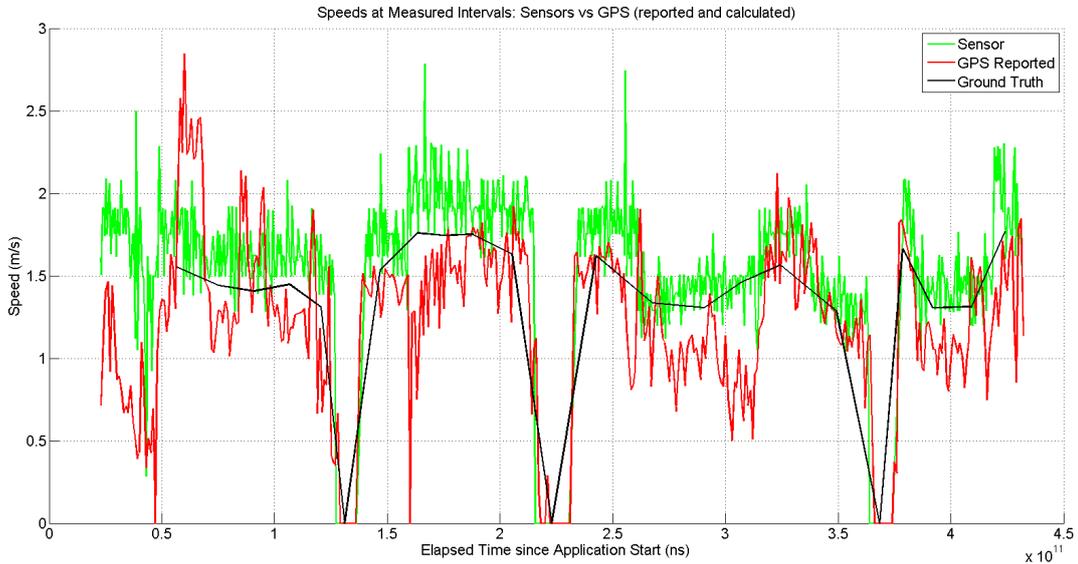


Figure 4.2: Gait Profile (sensor) vs GPS Speeds vs Ground Truth

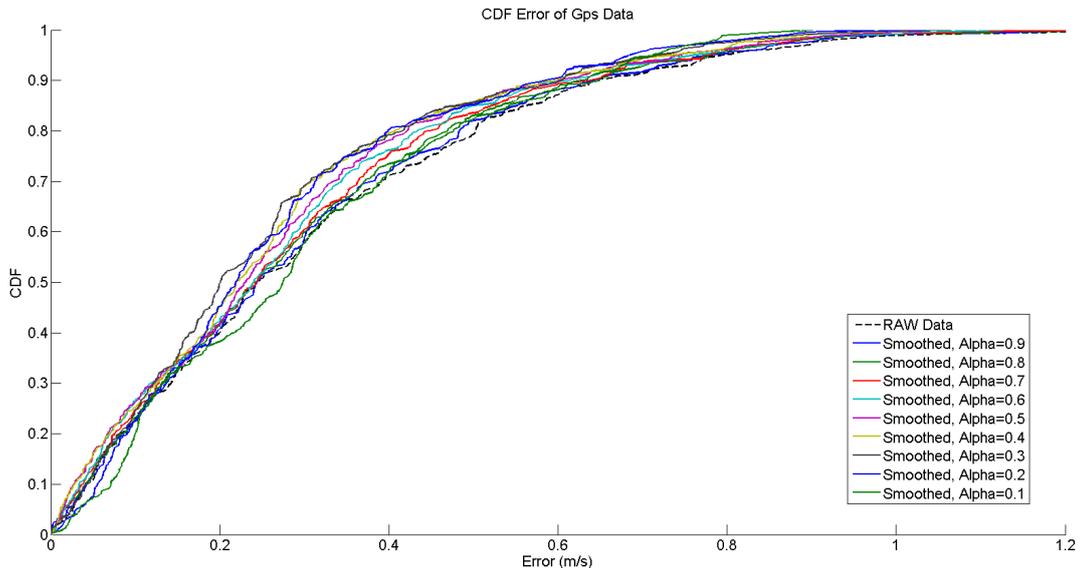


Figure 4.3: ECDF: GPS Speed

From these figures, a few observations were made:

1. The speed computed using Gait Profile is mostly higher than the ground truth.

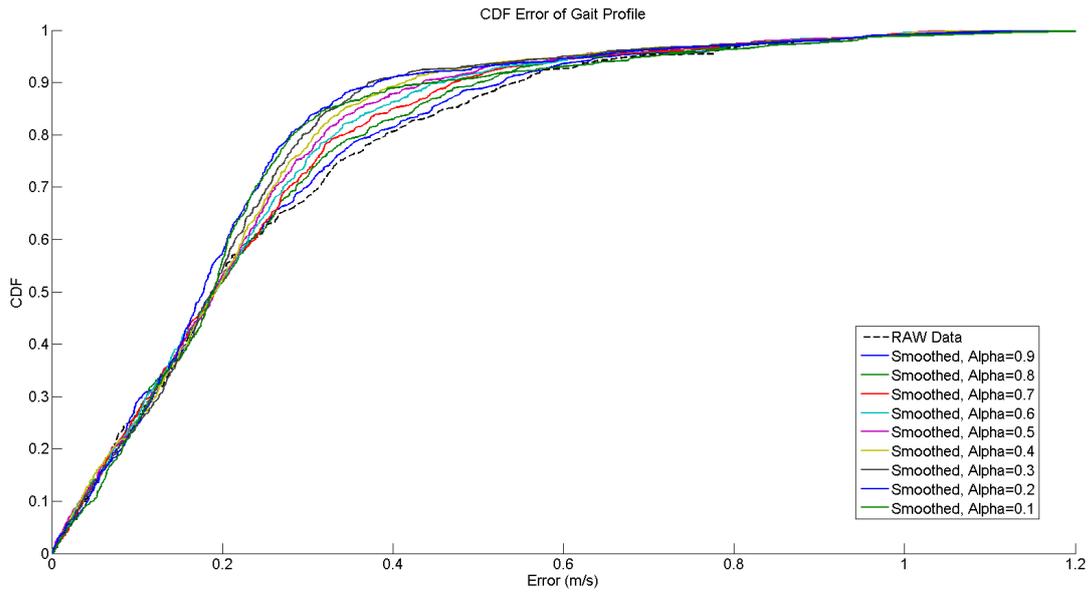


Figure 4.4: ECDF: Gait Profile (Sensor)

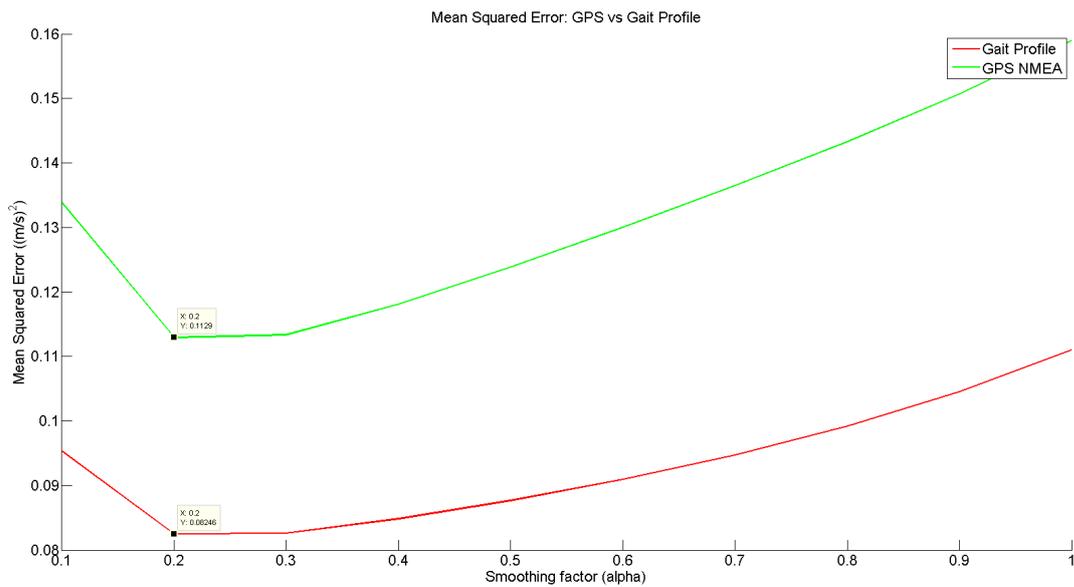


Figure 4.5: Gait Profile (sensor) vs GPS Speeds (MSE)

2. The speed computed using GPS is mostly lower than ground truth.
3. Gait Profile speed has less error overall compared to GPS speed.
4. The smoothing factors of $\alpha = 0.2$ and $\alpha = 0.3$ for both cases minimize the MSE.

The MSE for Gait Profile speed and GPS speed with $\alpha = 0.2$ are $0.082 (m/s)^2$ and $0.113 (m/s)^2$ respectively. The ground truth velocities are calculated by:

$$v_k = \frac{\|p_{k+1} - p_k\|}{(t_k + t_{k+1}) * 0.5} \quad (4.1)$$

where p : waypoint position

k : current sample

t : elapsed time

This explains why during the intervals when the user has stopped moving (around 13 s, 22 s and 37 s), it only reaches 0 m/s in the middle of the intervals. This means the actual error for both Gait Profile and GPS are less.

4.2 Direction

4.2.1 Setup

As mentioned earlier, the data needed for fusing with the ToF system are both speed and its direction. While GPS data can reliably determine the heading of the smartphone, in an indoor environment, we relied on the Compass. We designed a test at the testbed in Armasuisse Thun (Fig.4.6).

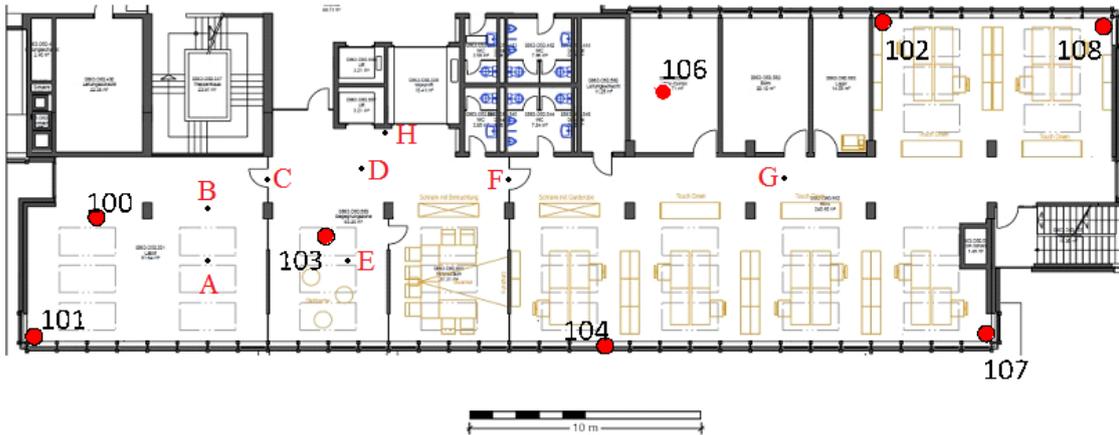


Figure 4.6: Testbed I at Armasuisse Thun, Switzerland

In this setup, we first randomly selected the waypoints A to G. Similarly to 3.2, we used the Bosch digital laser measure to measure the distances from each waypoint to their closest walls or obstacles in both X and Y direction. Taking the bottom left corner as the origin with coordinates $(0,0)$, we then used the scale provided to measure the corresponding X or Y distances from the origin (measuring actual distances were needed for calculation of ground truth speed in later tests). Lastly, the corresponding coordinates of each waypoint were found.

The building where this floorplan is located forms an angle from the north direction. In order to find this direction offset, we used the online tool <http://googlecompass.com/>. In Figure. 4.7, outlined in red is the testbed and the measured orientation of the building is $N7^\circ E$.

4.2.2 Results

Therefore, with a simple rotation, we obtained the following results:

With the coordinates aligned with the world's North and East direction, we then designed walking

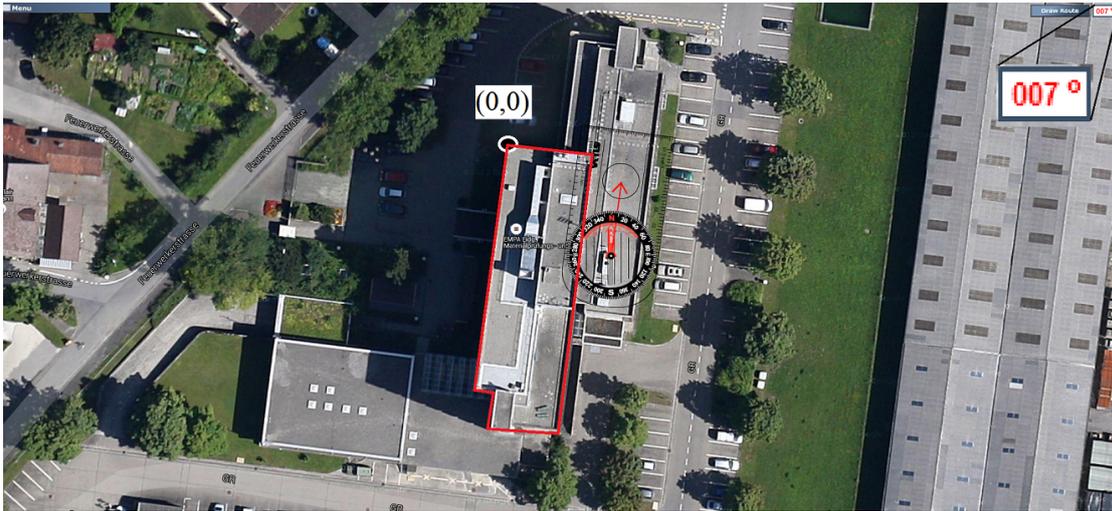


Figure 4.7: Testbed at Armasuisse Thun, Switzerland (Satellite Map). Building Orientation Measurement

Way Point	Coordinates (Before Rotation)	Coordinates (After Rotation)
A	(7.623, 3.573)	(2.6174, -8.0016)
B	(7.786, 6.49)	(5.4927, -8.5189)
C	(10.13, 7.482)	(6.1917, -10.9663)
D	(13.421, 7.227)	(5.5375, -14.2017)
E	(13.672, 7.227)	(1.5675, -13.9671)
F	(20.532, 7.219)	(4.6630, -21.2587)
G	(32.56, 7.29)	(3.2676, -33.2057)
H	(15.22, 10.2834)	(8.3525, -16.3599)

Table 4.1: Way Point Coordinates of Testbed 1

paths with these waypoints. Before analyzing the error statistics, we used a simple algorithm to upperbound the error (ϵ) to 180° :

$$\epsilon = \min(360^\circ - |deg_{sensor} - deg_{truth}|, |deg_{sensor} - deg_{truth}|) \quad (4.2)$$

This is essential since a Compass data is measured with respect to absolute North. For example, if the sensor evaluated a direction of $N359^\circ E$ and the ground truth was $N1^\circ E$ the absolute difference would be $N358^\circ E$ but the true error was merely $N2^\circ E$. As an example, Figure. 4.8 shows the comparison between our sensor data versus ground truth, and Figure.4.9 shows the corresponding ECDF. The median error is 7.29° . This was lower than our expectation since this indoor environment contains many metallic materials, as well as radio signal devices. We expected the error caused by magnetic interference to be large.

In the ZEE paper [18], they measured a magnetic offset at the highest of 30° with probability 2.5% and where the highest probability 30% is found to be an error of 30° . Although our error was comparatively lower, we aimed to improve it further. Similar to the case of velocity, we applied exponential smoothing to the sensor data with a range of smoothing factor α . As can be seen also in Figure. 4.10, the smoothing did not improve accuracy (see A.2 for additional test results).

One notable difference between our approach and from ZEE is that we only used the most recent reading on the Compass to the instance when velocity is computed, whereas ZEE used the mean of an interval of samples around the timestamp where a measurement is in question. A velocity computation can happen only after at least two consecutive steps have occurred,

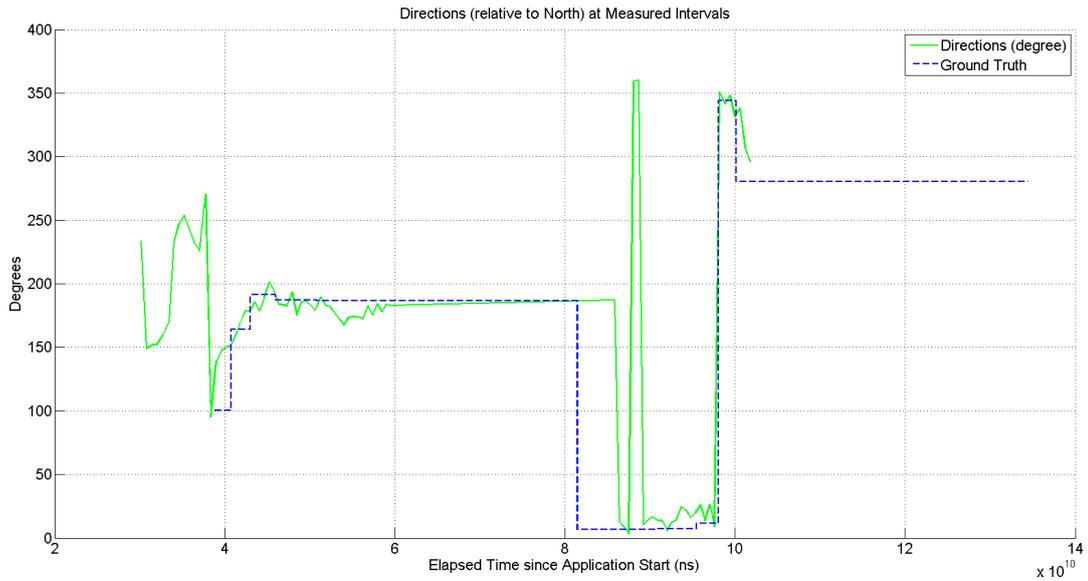


Figure 4.8: Direction Comparison. Sensor vs Ground Truth

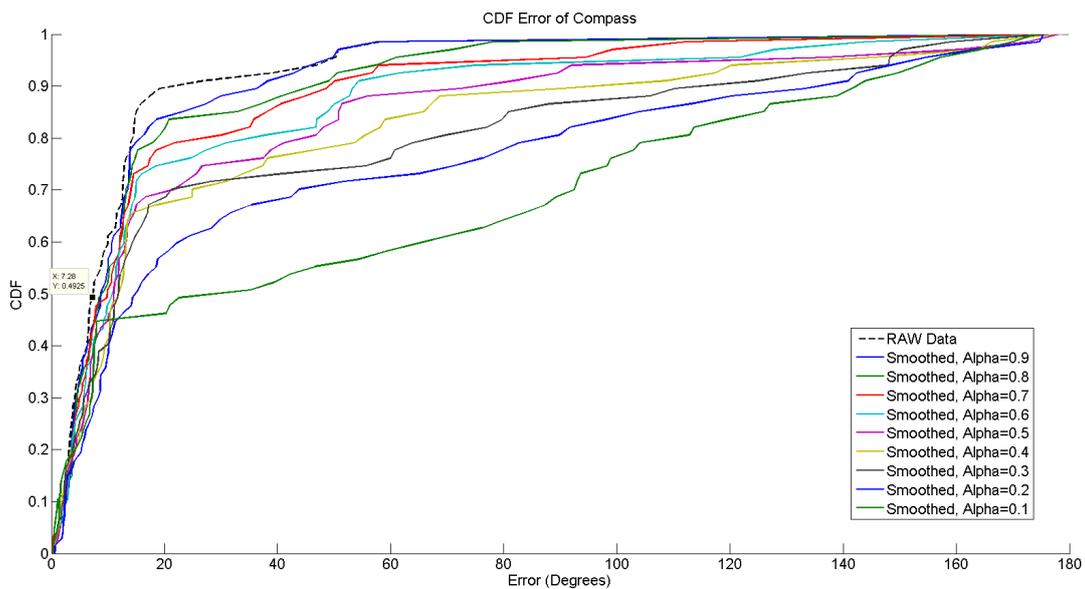


Figure 4.9: ECDF: Direction Comparison. Sensor vs Ground Truth

which is relatively slow compared to the possible maximum sampling frequency when our Magnetometer sampling frequency is set to "public static final int SENSOR_DELAY_FASTEST" (see 2.6.2).

Therefore, we investigated the benefits of averaging compass data. Similar to the test in Chapter.4.2.1, we repeated the walking test several times, but logging all compass readings instead of only the ones closest to instances when speeds were logged. Then we took the average values including all readings before the next instance of speed logs. In Figures. 4.11 and 4.12, we see that averaging, while removing noise, only appeared to improve error on Test 05. From Figures. 4.13 and 4.14, it can be further seen that only in Test 05 did the averaging improve accuracy (from a median of 23.14° to 21.24°). Therefore we did not adopt averaging in our ToF system fusion.

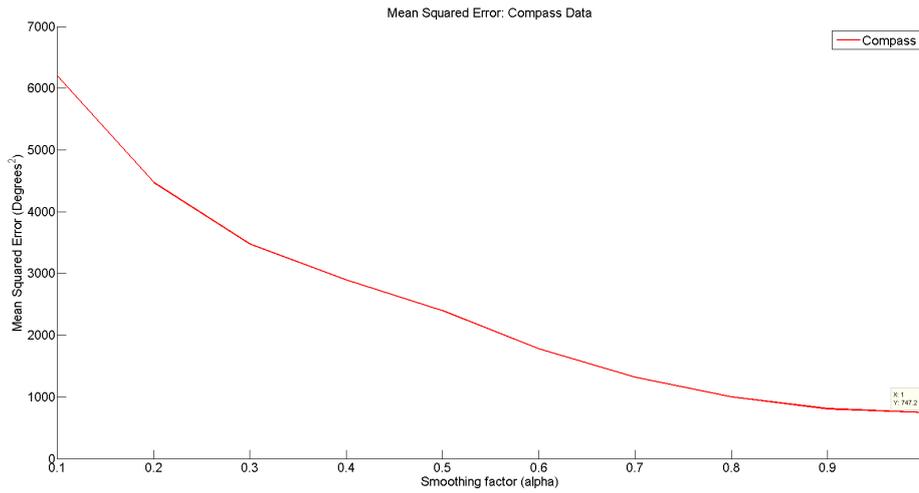


Figure 4.10: MSE: Sensor Directional Error with Exponential Smoothing

Another possible contribution to the error would be the Heading Offset. This is the dif-

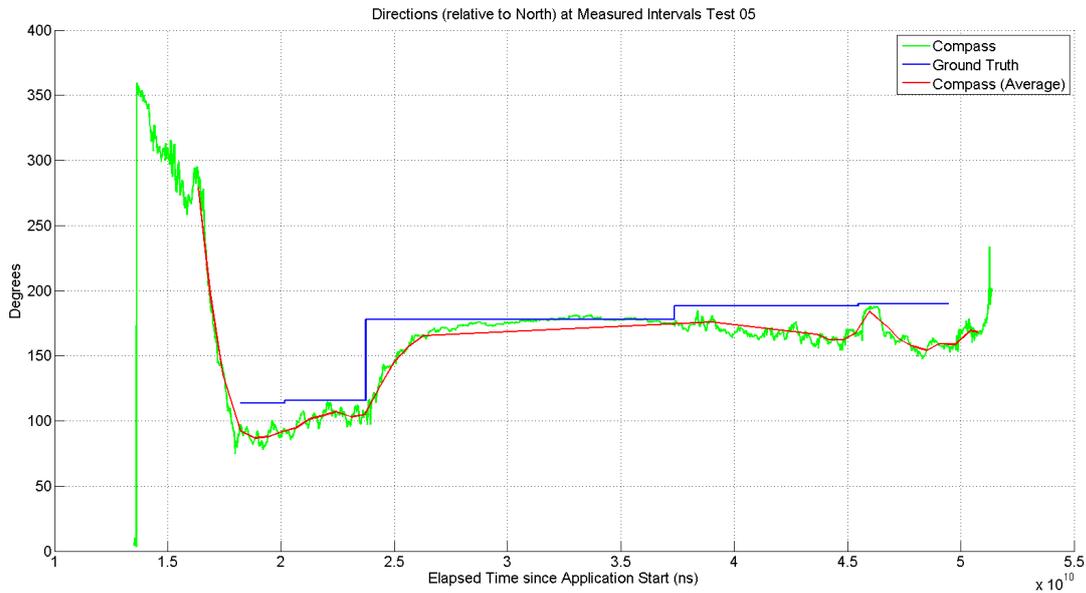


Figure 4.11: Direction Plot: Raw Compass Data vs Averaging vs Ground Truth (Test 05)

ference between the actual heading of the user and the tilt of the phone (where the Y-axis is actually pointing) due to the position of the hand. For example, a right handed user would usually tilt the phone slightly counter-clockwise. While this offset is difficult to measure and not necessarily constant (both user-dependent and motion-dependent), we would model our Kalman Filter assuming the offset as noise.

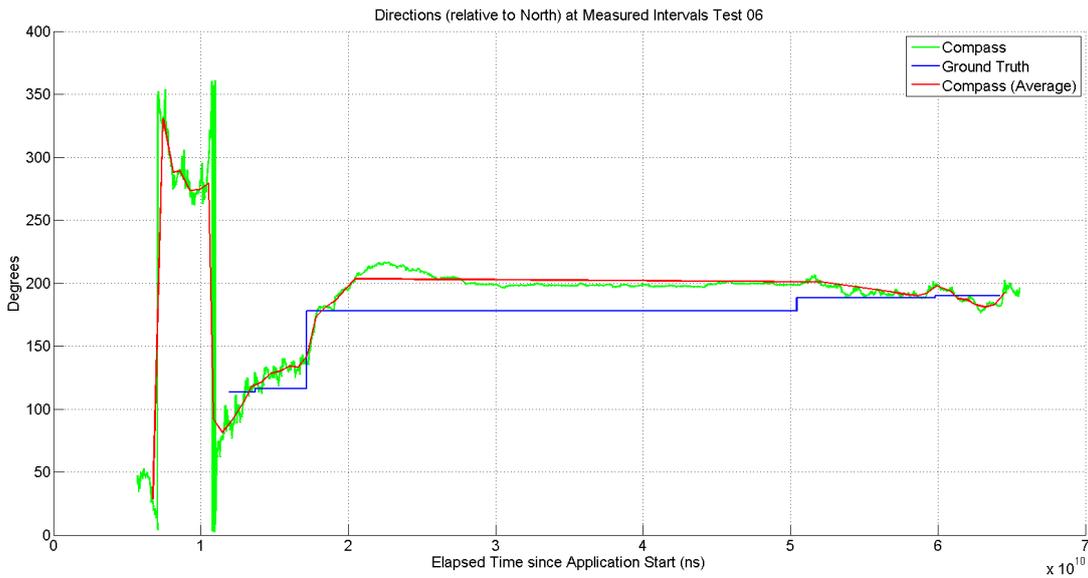


Figure 4.12: Direction Plot: Raw Compass Data vs Averaging vs Ground Truth (Test 06)

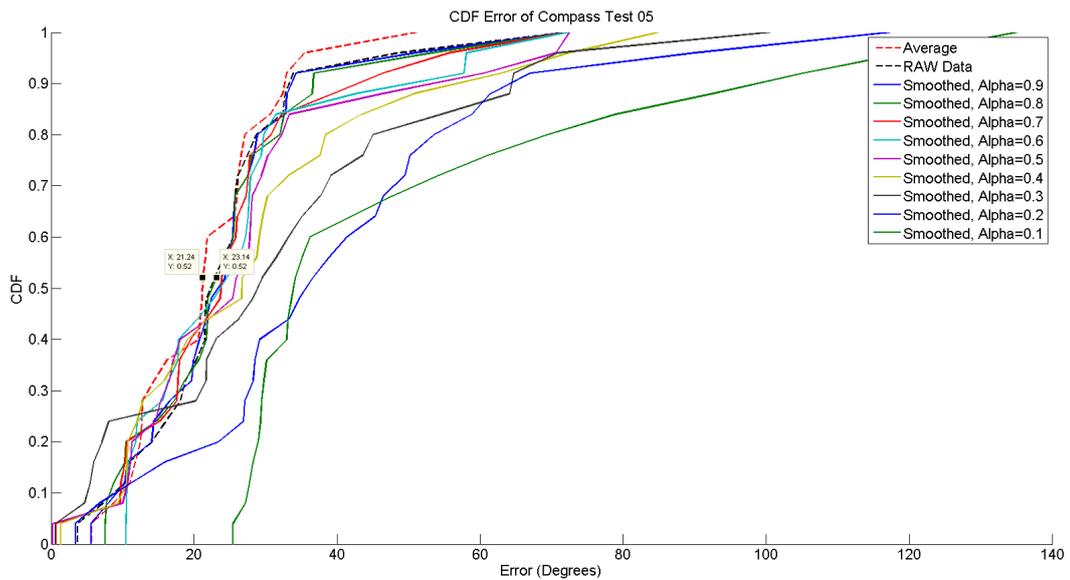


Figure 4.13: ECDF Direction: Raw Compass Data vs Averaging vs Ground Truth (Test 05)

4.3 Step Counter

One of the issues concerning the internal step counter sensor from the phone is its initialization. Once the sensor listener is registered, it does not detect the first step by the user. Instead, it takes roughly six steps before the first step is reported. Since the algorithm for step counter is not officially documented, we made use of the accelerometer as the prior step counts. Since our design fixated the orientation of the phone, we needed to use only the Z direction accelerometer data. This is done by the following:

1. Collect all data prior to the first reported step by Step Counter
2. Smooth data using exponential smoothing to remove noise

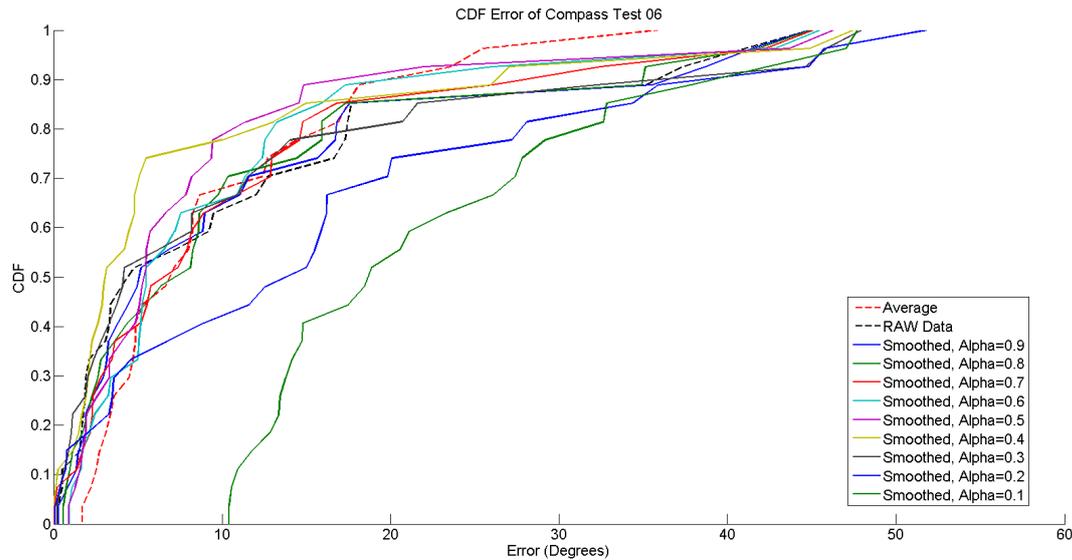


Figure 4.14: ECDF Direction: Raw Compass Data vs Averaging vs Ground Truth (Test 06)

3. Compute mean and for every instance the data crosses the mean, one step is counted

While the algorithm is relatively simple, there are a few disadvantages. Firstly this is not a real time calculation and therefore the user cannot immediately see if the first steps are counted. Secondly, the smoothing factor depends on the noisiness on the data. For cases when the smoothing factor is too low, the algorithm would count extra steps (see Fig. 4.15). If the smoothing factor is too high, some of the spikes in the data might be smoothed out and less steps would be counted. Since the step counter is kept "awake" once the first step is counted, we decided to use this sensor exclusively for our app. This was made possible by a simple change in the interface where the "Measure" button remained inactive until Step Counter had started.

The last issue regarding Step Counter appeared when the user stopped moving. Since speed

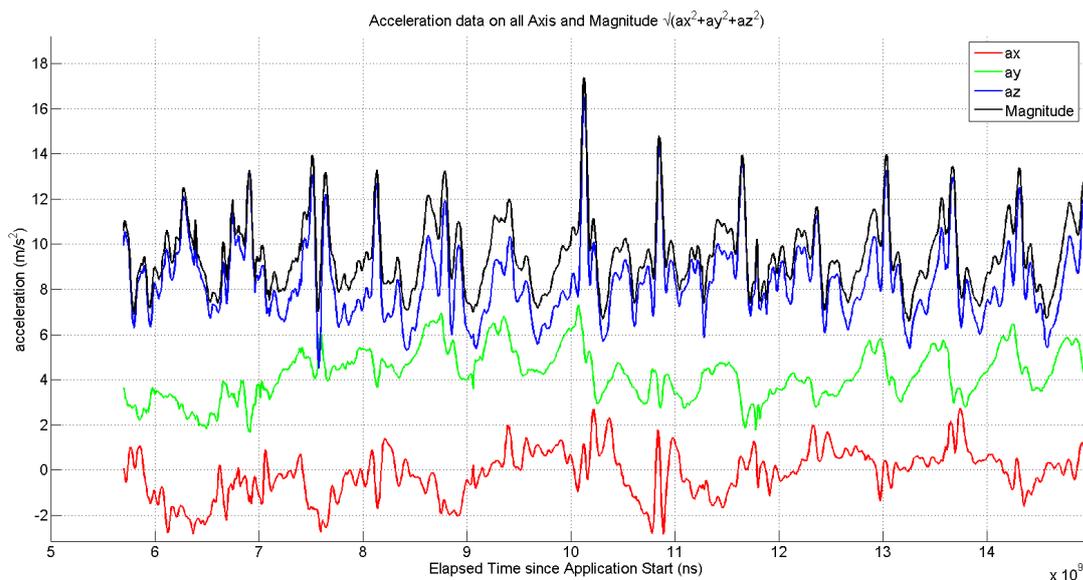


Figure 4.15: Accelerometer (Test 06)

is calculated based on the registered step count, when the user stopped moving, the app sim-

ply waits until the next step and considered the stopped interval as part of one step. Therefore it would register a low speed instead of zero. Using data found in [25], we lower-bounded the possible walking speed to 0.5 m/s . Whenever a speed slower than this is calculated, it would return 0 m/s .

4.4 Calibration

Previously, we evaluated the accuracy of a manually calibrated Gait Profile (default values) versus that from GPS NMEA. Here we added a calibration function to the app that updates the Gait Profile according to the current user of the phone. This calibration is expected to be performed outdoor as the data input relies on GPS data. In order to make sure the user is already in an area where GPS location update is feasible, we modified the interface such that the function was only available when the smartphone had locked into at least one GPS location update.

In the smartphone app, the check box for Gait Profile update function does not appear until "Last Known Location Updated" is displayed. Additionally, when the app is undergoing measurement, changing the Gait Profile from "update" to the default values is also not possible. This function is cumulative. This means that the longer the user has the phone, the more personalized data will be entered into the Gait Profile.

In order to evaluate this calibration function, we reused the original test user and did a short 200 step Gait Profile Update, where we logged the Step Frequency and Step Size every 1 second. The resulting Gait Profile can be seen in Figure. 4.16. As the figure shows, the resulting linear progression is in fact corrupting the data by assuming an inverse linear relationship between Step Size and Step Frequency.

Our first intuition was that the sample size was not sufficient and the sampling rate was perhaps too fast. In our manual calibration, the minimum amount of steps was 35 and the maximum was 154. In order to imitate the same measuring algorithm, we modified the app so that it would first generate a random number within this range of step counts (we chose $\{20, 200\}$). Everytime a new data point was logged (speed, step frequency), a new random number was generated given the same range. We proceeded to do a much longer test in order to collect a larger sample size.

A second experiment was conducted with a sample size of 1000 steps, the result of which can be seen in Figure. 4.17. A similar pattern was observed here and therefore the hypothesis of low sample size did not seem correct. By plotting the corresponding GPS Velocity, we observed that its value appeared to be noisy with a constant mean. Since

$$v = \text{Step Size} * \text{Step Frequency} \quad (4.3)$$

it explained why Step Size and Step Frequency formed an inverse linear relationship. We therefore concluded the GPS online calibration is not feasible and we could only use the manually calibrated Gait Profile for further experiments.

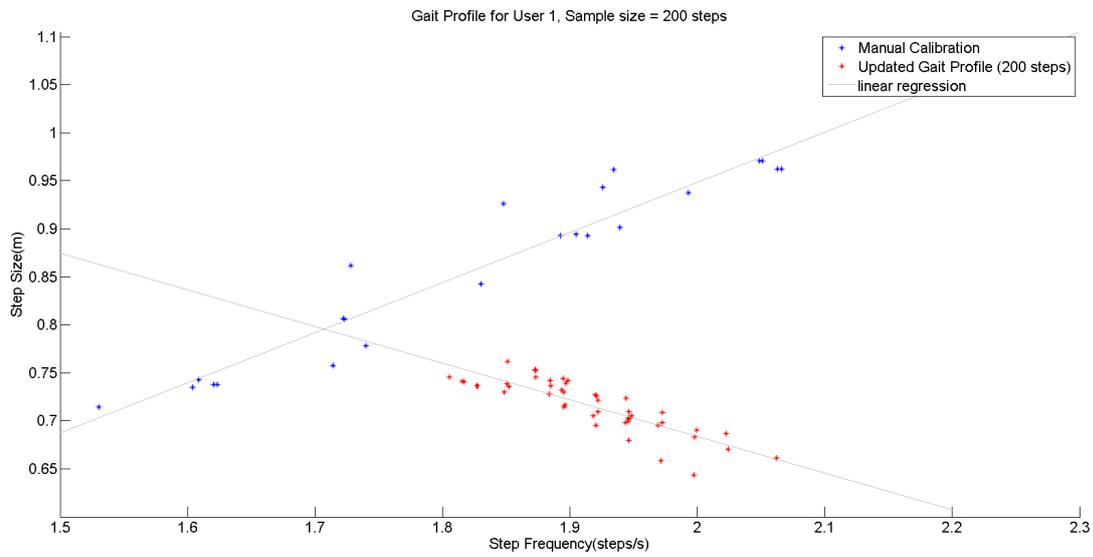


Figure 4.16: Gait Profile: Updated with 200 Steps using GPS Data

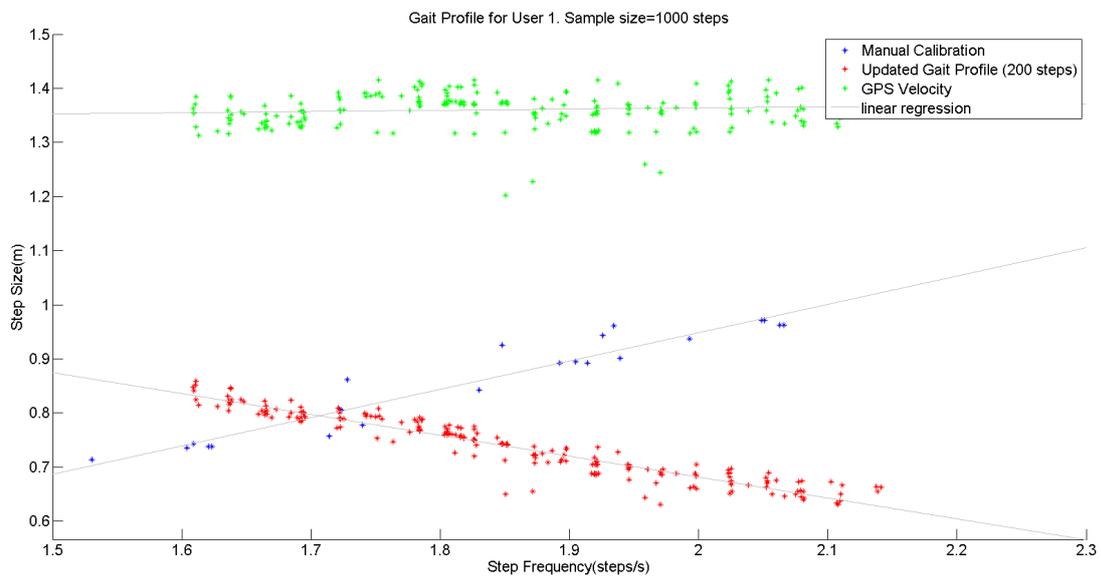


Figure 4.17: Gait Profile: Updated with 1000 Steps using GPS Data

Chapter 5

Time-of-Flight System and Sensors Fusion

5.1 Kalman Filter

Using data from sensors and the WiFi ToF system, we could apply Kalman Filter as illustrated in Figure. 5.1. For each iteration of the Kalman Filter, we first have

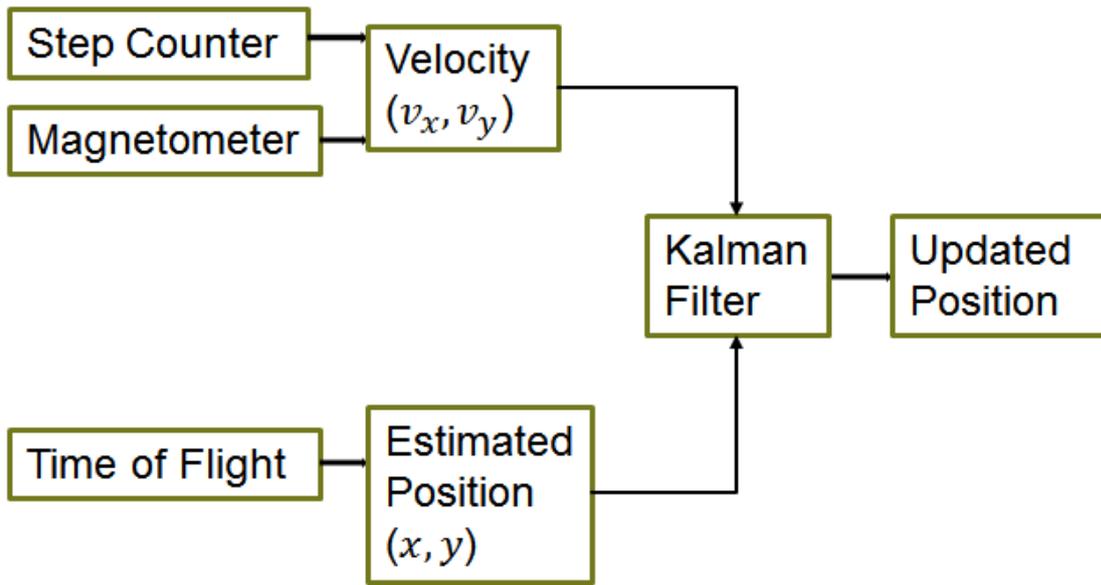


Figure 5.1: Flow Chart: Kalman Filter

$$\begin{bmatrix} x_{k+1} \\ y_{k+1} \end{bmatrix} = \begin{bmatrix} x_k \\ y_k \end{bmatrix} + \begin{bmatrix} dT_k & 0 \\ 0 & dT_k \end{bmatrix} \begin{bmatrix} v_{x,k} \\ v_{y,k} \end{bmatrix} + \begin{bmatrix} q_x * dT_k & 0 \\ 0 & q_y * dT_k \end{bmatrix} \quad (5.1)$$

or simply written as

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \mathbf{B}_k \mathbf{v}_k + \mathbf{Q} \quad (5.2)$$

Here we have N total states ($k = 1, 2, \dots, N$), and in each state we have the two-dimensional velocity computed from Step Counter and Compass ($v_{x,k}, v_{y,k}$). dT_k is the elapsed time between consecutive states where q_x, q_y are the noise variances originated from noisy sensor data. Therefore we can predict the next coordinates x_{k+1}, y_{k+1} . Table. 5.1 summarizes the steps in our Kalman Filter implementation. The measurements from the ToF system is $\mathbf{D}_k = [d_{x,k}; d_{y,k}]$,

the ground truth \mathbf{x}_k and the measurement error $\tilde{\mathbf{D}}_k = [\tilde{d}_{x,k}; \tilde{d}_{y,k}]$ are related by the equation:

$$\begin{bmatrix} d_{x,k} \\ d_{y,k} \end{bmatrix} = \begin{bmatrix} x_k \\ y_k \end{bmatrix} + \begin{bmatrix} \tilde{d}_{x,k} \\ \tilde{d}_{y,k} \end{bmatrix} \quad (5.3)$$

Lastly, the measurement noise matrix is:

$$\mathbf{R} = \begin{bmatrix} r_x & 0 \\ 0 & r_y \end{bmatrix} \quad (5.4)$$

The values q_x, q_y and r_x, r_y are not known since we only have limited empirical data from previous tests to evaluate both sensor and ToF measurement noise. Therefore, we tested different values for our Kalman Filter. Table 5.1 summarizes the steps in our Kalman Filter implementation.

Step 1: Project the State ahead	$\hat{\mathbf{x}}_{k+1} = \mathbf{x}_k + \mathbf{B}_k \mathbf{v}_k$
Step 2: Project the error covariance ahead	$\hat{\mathbf{P}}_{k+1} = \mathbf{P}_k + \mathbf{Q}$
Step 3: Compute the Kalman gain	$\mathbf{K} = \hat{\mathbf{P}}_{k+1} (\hat{\mathbf{P}}_{k+1} + \mathbf{R})^{-1}$
Step 4: Update estimation with measurements	$\mathbf{x}_{k+1} = \hat{\mathbf{x}}_{k+1} + \mathbf{K} (\mathbf{D}_k - \hat{\mathbf{x}}_{k+1})$
Step 5: Update the error covariance	$\mathbf{P}_{k+1} = (\mathbf{I} - \mathbf{K}) \hat{\mathbf{P}}_{k+1}$

Table 5.1: Kalman Filter: Fused Sensor Data and ToF Measurements

5.2 Results and Error Analysis

Most of our experiments were conducted using the same waypoints as shown in 4.2.1 (Test01 - 04) whereas two further tests (05 06) were performed with new waypoint positions, whose coordinates can be found in Appendix B.1. By walking different predefined paths between these waypoints, we obtained the Ground Truth, ToF measurements from the system and logged the relevant sensor data.

5.2.1 Error Analysis

As an example, a resulting ECDF of the distance error can be found in Figure 5.2 (18955 measurements) (see A.3 for additional distance error plots). As can be seen, the Kalman Filter did not improve the position estimation, but rather produced a much larger error than its unfiltered counterpart. However, when we used only sensor data to perform position estimations, we obtained a much lower error (3.5 m). This implied that the large error came from the ToF measurement. In order to identify the source of this large error, we first explored the effect of changing the number of anchors used for multilateration. In all of our additional computation, changing the allowable amount of anchors used for multilateration did not produce noticeable changes in distance error. We further plotted the distance error against time in Figure 5.3.

5.2.2 Outliers

We found many large errors coming from the beginning of the measurements (reaching $6.359 \times 10^3 \text{ m}$ in this example). While this could be an issue of outliers not being removed from the data, the lowest error was still 14.02 m. This meant that the ToF ranging estimation alone was not performing as expected. It is also worth noting that the error found by using only sensors increased over time as a result of dead reckoning. We iterated the Thompson Tau method to remove outliers again and repeated the process to observe if the accuracy would improve. The resulting ECDF after 1000 iterations is shown in Figure 5.4. While this method did remove most outliers, the remaining position estimation both before and after applying Kalman Filter still had a large error (a median of around 18 m). We also observed that the X and Y positions appeared to be only noisy data and did not display similar patterns as the ground truth.

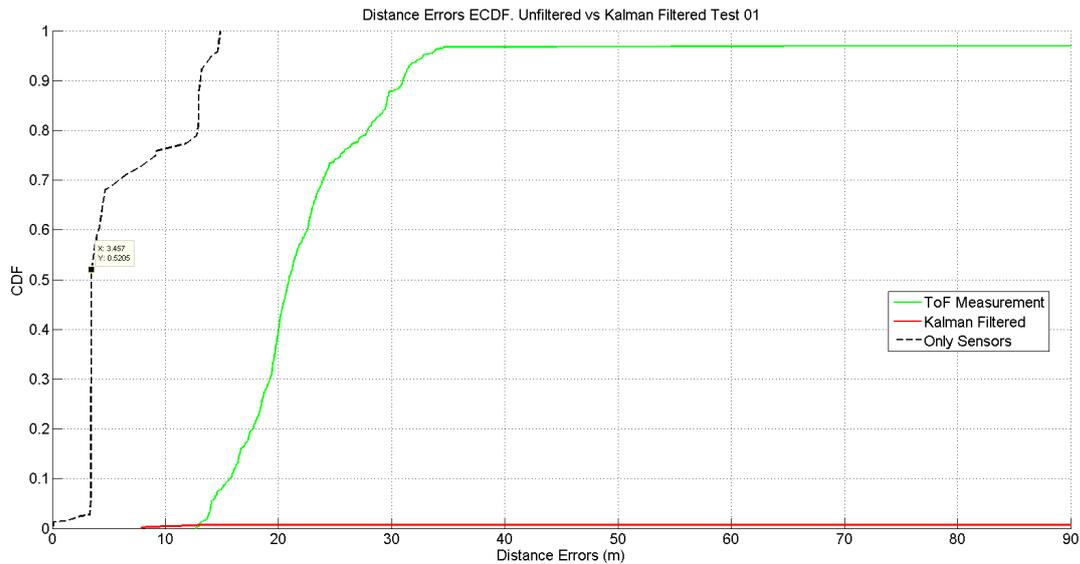


Figure 5.2: ECDF: Distance Error (Test 01)

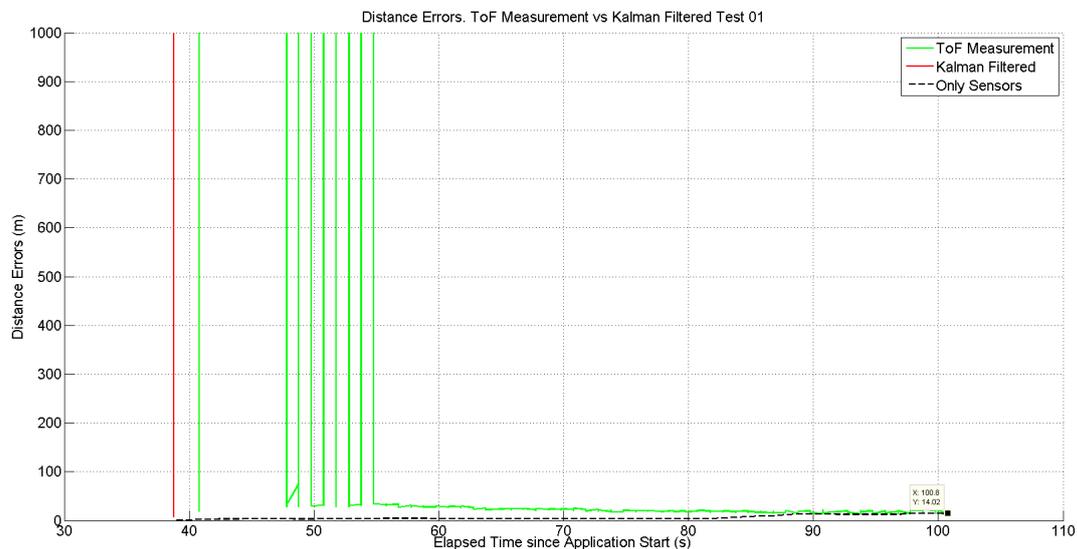


Figure 5.3: Distance Error vs Elapsed Time (Test 01)

5.2.3 Ranging Error

Next, we analysed the ranging error from the anchors. We compared the distance calculated from the anchors using their ToF measurements both before and after the linear regression filter (see 1.2.3 and [19]) was applied. The results can be seen in Figure. 5.5 and Figure. 5.6. The first figure shows that the linear regression filter did improve accuracy, however, the filtered ranging error still reached as large as 100 m . The second graph further shows the reduction in error. This meant that there were still unidentified issues in the system.

5.2.4 Calibration Error

The last source we analysed was the accuracy of the chipset calibration. As described in 1.2.2, these are values that represent the number of clock cycles it takes the anchors to reach the chipset (of the smartphone) when they are 0 m apart (ref_0). Since we did have the ground truth

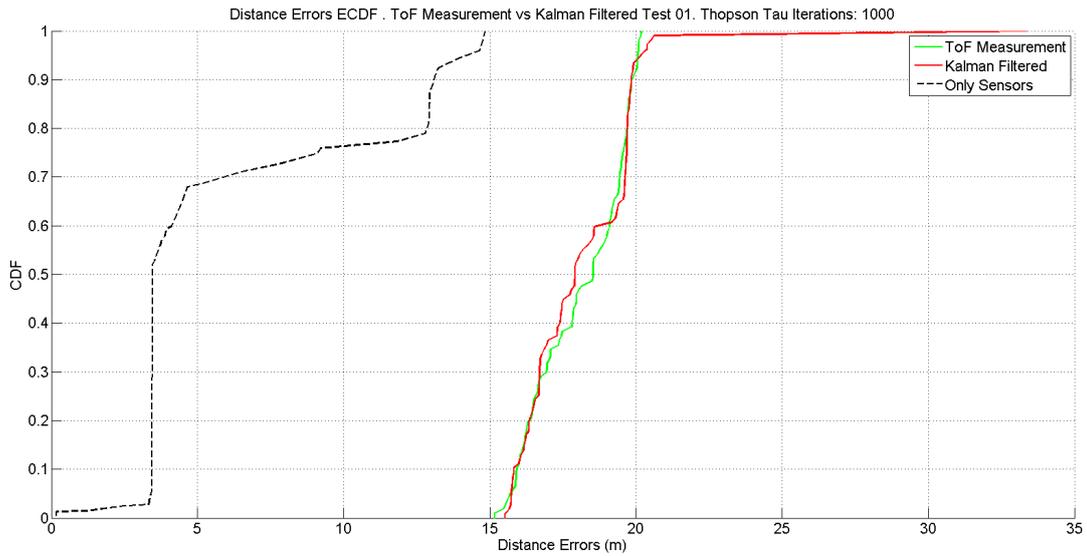


Figure 5.4: ECDF: Distance Error (Test 01). Thompson Tau: 1000 iterations

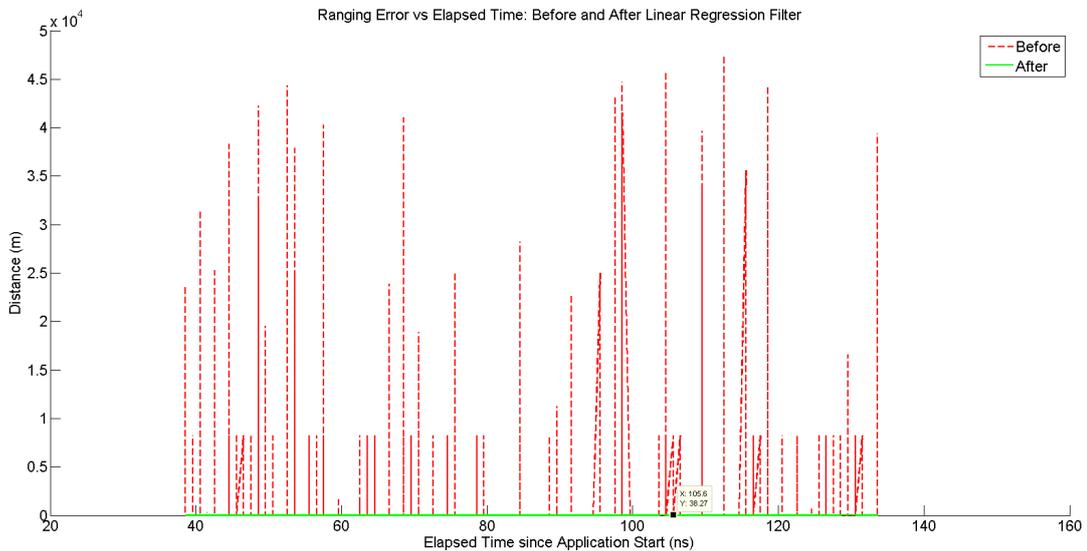


Figure 5.5: Ranging Error vs Elapsed Time (Test 01)

from all our tests, we could reversely calculate the calibration values by the equation:

$$ref_0 = TOF_{MEAS} - \frac{\text{True Distance (m)}}{1.7}$$

For each of the four rates used, we plotted the resulting distribution of the recalibration values in Figure 5.7. Table 5.2 shows the original calibration values as well as the mean and median of the new values. We then attempted to use both the mean and median of the new values to estimate the positioning. While the mean caused the multilateration to fail at converging to position estimate using measurements (it simply returned NaN values in MATLAB), the median improved the distance error as shown in Figure 5.8. However, the error was still quite large, especially compared to the achievement of 3.47 m as described in 1.1. Furthermore, the Kalman Filter still did not improve the distance error. Additionally, as seen in Figure 5.9, the changes in Y coordinate appeared to follow the same pattern as the ground truth with a lag time. This lag could be due to the fact that the user was walking faster than all the anchors could complete their measurements at each position.

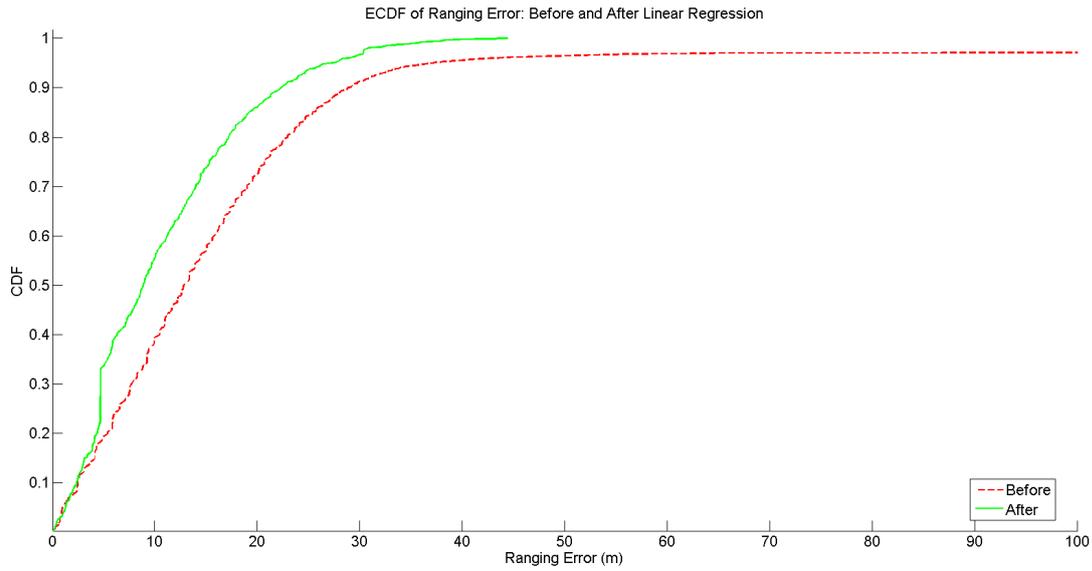


Figure 5.6: ECDF: Ranging Error (Test 01)

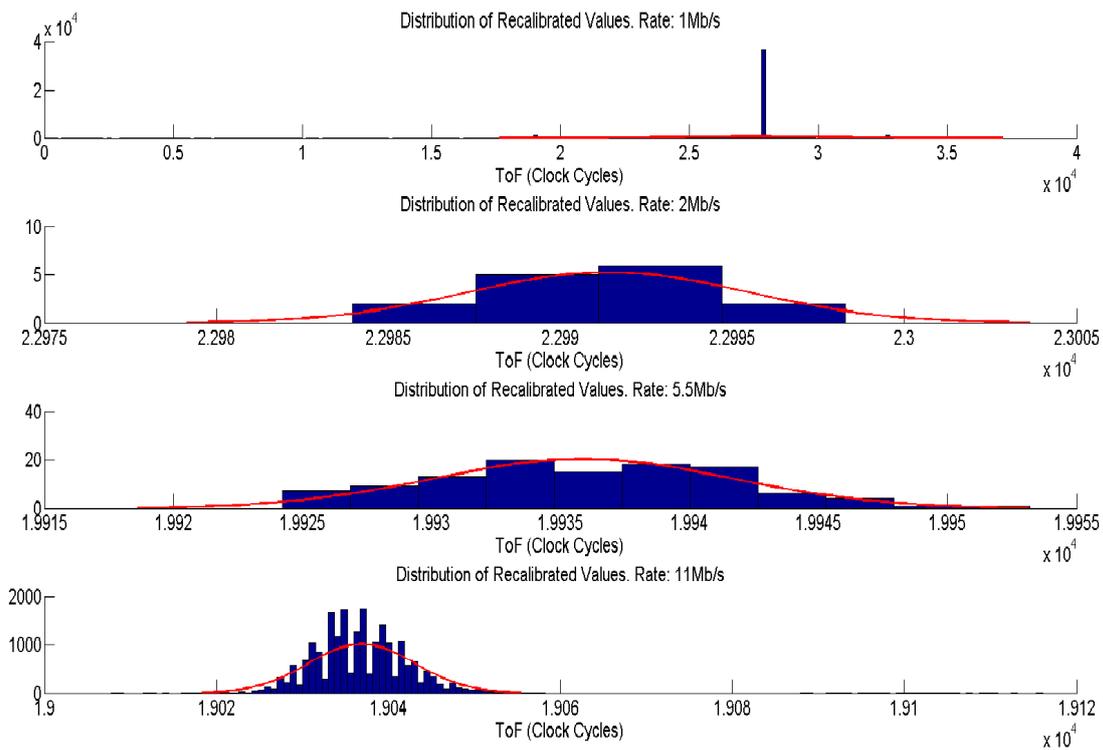


Figure 5.7: Recalibration Values Distribution

5.3 Simulation

Since the source of errors in the WiFi ToF system was too complex to identify and correct, and we still wanted to see the effect of Kalman Filter, we simulated the scenario based on Test 01. As we already knew the ground truth, a simulation based on such could be made with selected measurement noise figures. In our simulation, we used binary search to select a sensor noise figure that best improved Kalman Filter accuracy. Then we compared different intervals

Rates	1 Mb/s	2 Mb/s	5.5 Mb/s	11 Mb/s
Original Values	27908	22994	19932	19040
New Values (Mean)	27386	22991	19936	19037
New Values (Median)	27920	22993	19936	19036

Table 5.2: Calibration Values: Original versus New

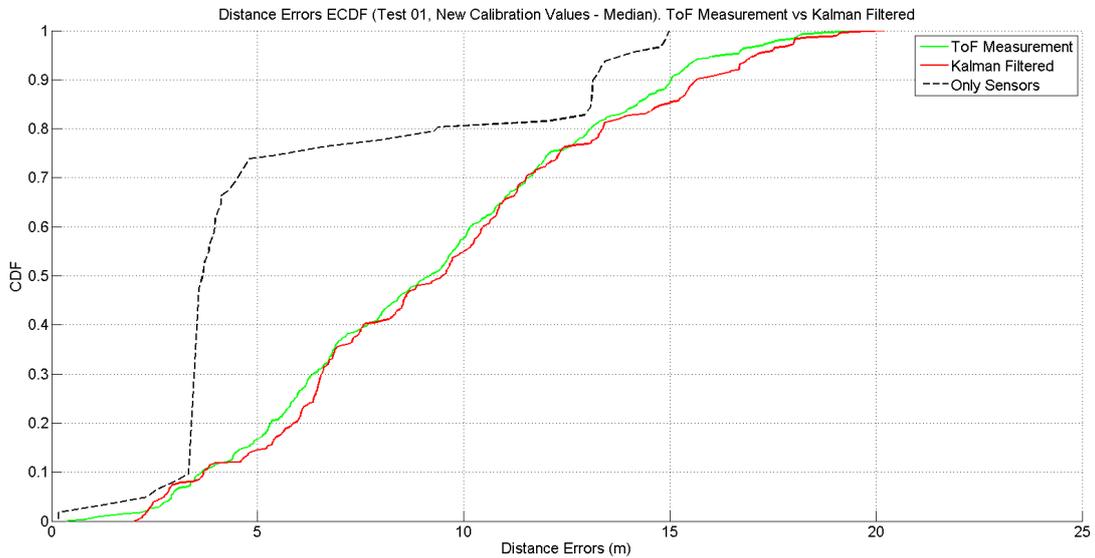


Figure 5.8: ECDF: Distance Error with New Median Calibration Values (Test 01)

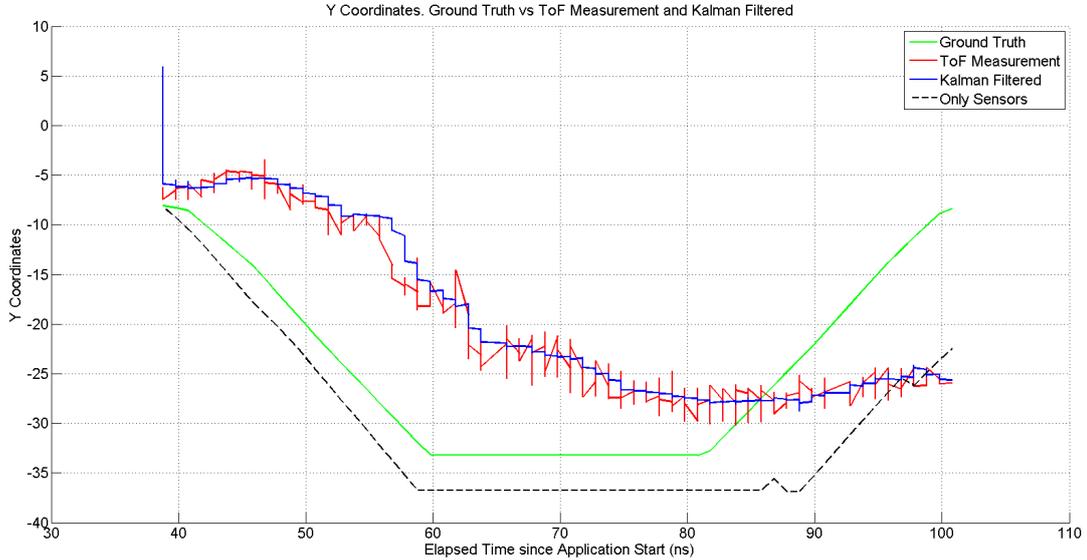


Figure 5.9: Y Coordinates. Ground Truth vs ToF Measurement and Kalman Filtered (Test 01)

of measurements to be used as input to the Kalman Filter. This means that we simulated situations where the fused algorithm request ToF information only at the intervals specified. At other instances, it used only sensor data to update its positions. Figure 5.10 shows the simulation with a ToF measurement noise of 3.47 m . It can be seen that when every 3 measurements are used, it had the highest reduction in distance error. However, when every measurement or every second measurement were used, the accuracy did not improved. While it showed accu-

racy improvements using Kalman Filter, we lacked crucial data, such as true noise variances, to simulate a real life environment. Lastly, a larger database could provide more insight into the influence of the fusion algorithm.

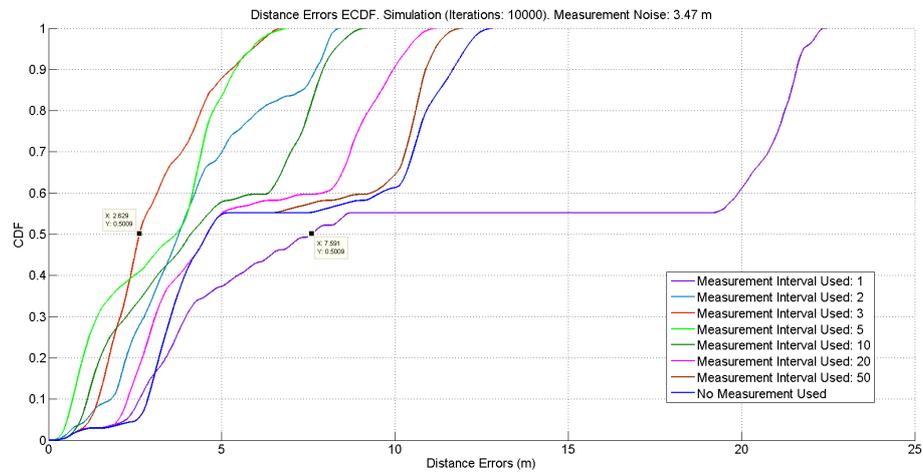


Figure 5.10: Distance Error ECDF. Simulation (Iterations: 10000). Measurement Noise: 3.47 m

Chapter 6

Conclusion and Outlook

6.1 Conclusion

Our thesis first demonstrated and evaluated the use of inertial sensors on commercially available smartphones. In order to reduce power consumption, we used the minimum amount of sensors possible. Firstly, by restricting the user pose, we eliminated various sources of error. This include sudden movement or re-orientation of the phone when the user changes pose. This allowed us to orient the phone to align with the world's coordinate and magnetic field, which also accomodated the use of the magnetometer to compute direction. The availability of Step Counter in the newest smartphones meant that we did not need to design complex computation to count steps. Not only did it save computing power consumption, it was also more accurate than relying on noisy accelerometer data. With the use of a Gait Profile, we could therefore measure the walking speed of the user.

With inertial sensors giving us information of speed and direction, we fused this collected data with the pre-existing WiFi ToF system at Armasuisse. However, due to system instability and unknown issues, we could not reliably obtain positioning measurements as described in [19]. We have reversely attempted to locate the source of error by evaluating the calibration process of the system, ranging errors of anchors, as well as the linear regression model used to decrease the effect of multipath. Unfortunately, none of our methods significantly improved the measurement. In order to seek insight into the effect of Kalman Filter, with which we fused sensor data and ToF measurements, we simulated the walking test using ground truth we previously obtained. While improvement could be observed in some simulations, our database was not sufficient on which to make conclusions.

6.2 Outlook

Although we could not successfully create a fused system for indoor positioning, we have developed some basic structure for further development. While our Gait Profile was calibrated to a single user, when larger databse is collected, it may be possible to assign gait profile to individuals who have similar physical characteristics. This means that a user may simply enter his/her height and weight and the smartphone would select a gait profile from its databse that best approximates the user.

One of the main challenges in our thesis was to select the correct noise figure to evaluate the accuracy of either our simulation or our fused positioning system. For sensor noises, large controlled experiments may need to be conducted that involve many users of different physical shapes. Another issue is to repeat such controlled experiments on different models of smartphones in order to study the difference in sensor inaccuracies. When this data is available in the future, a more robust system can be designed.

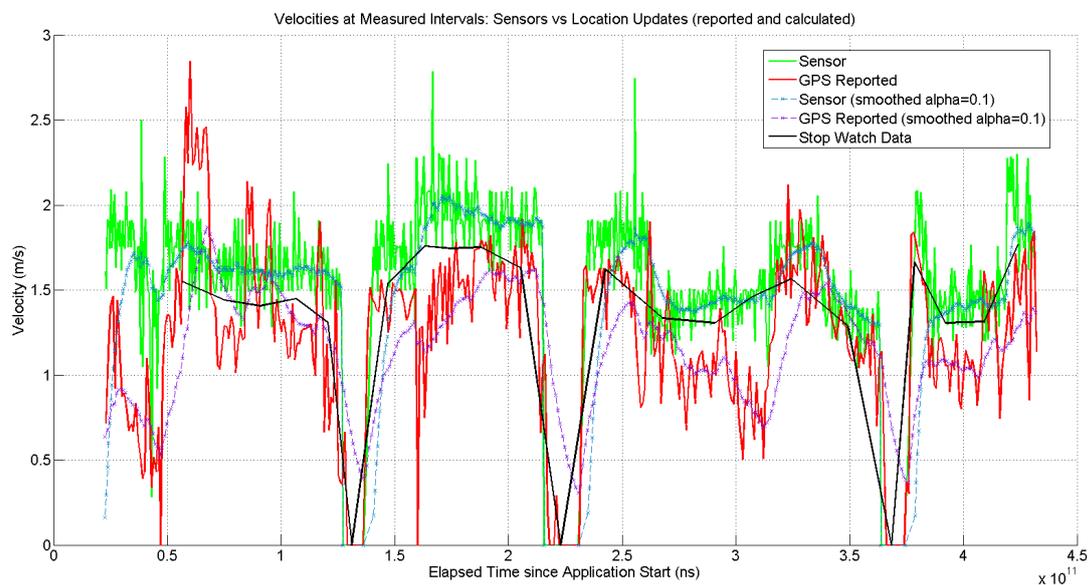
Our ToF system requires roughly 800 *ms* to complete one cycle of measurements. Since our test was based on walking, an improved ToF sytem that is less sensitive to position changes

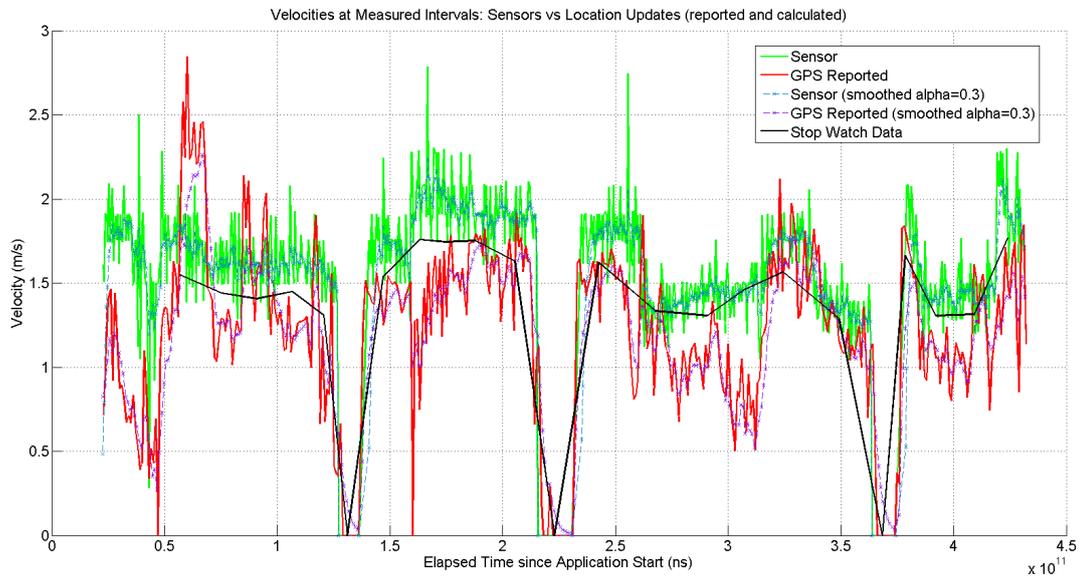
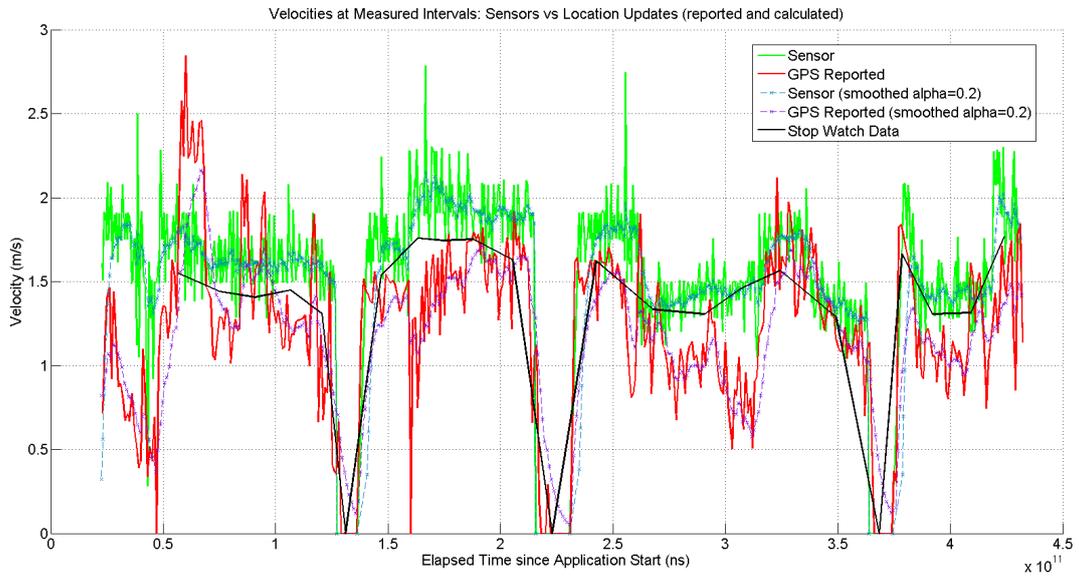
in time could be more beneficial.

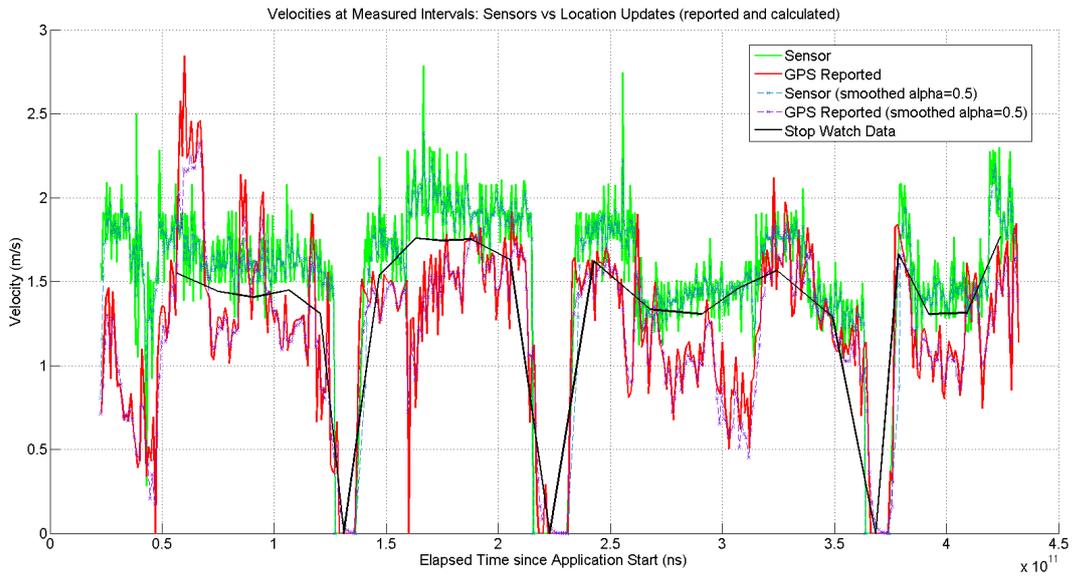
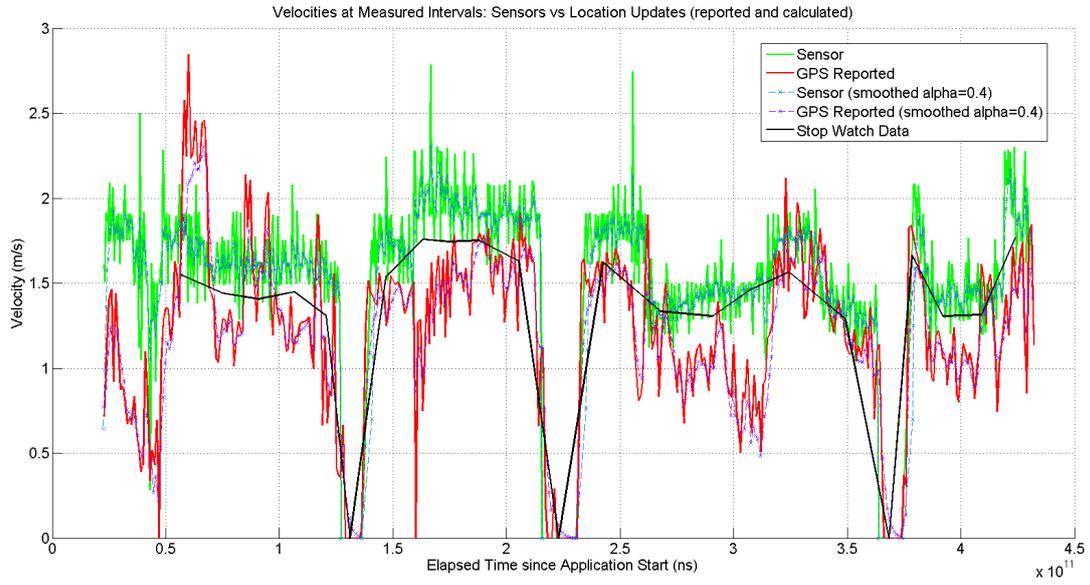
Appendix A

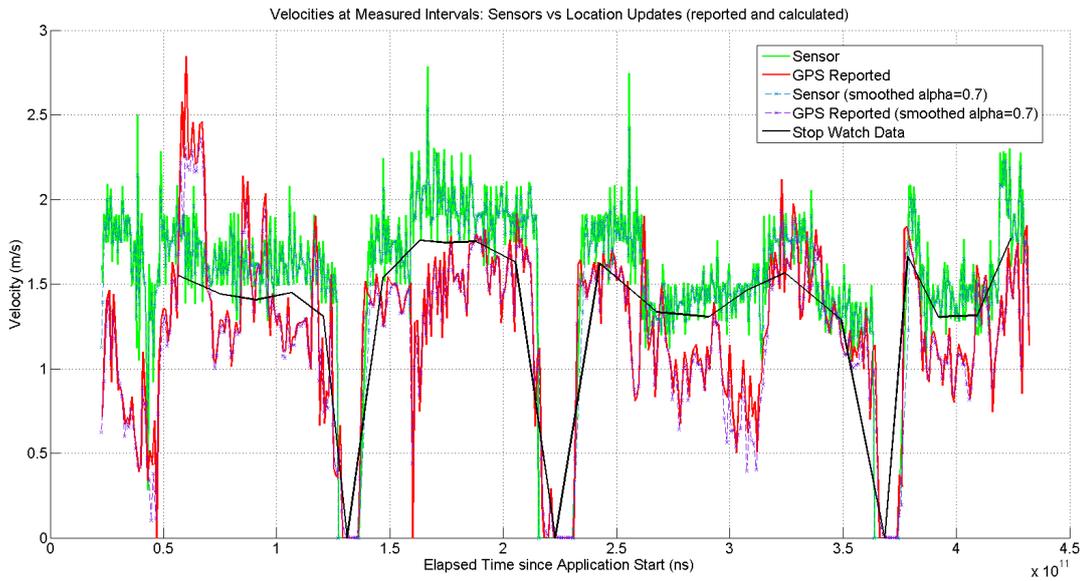
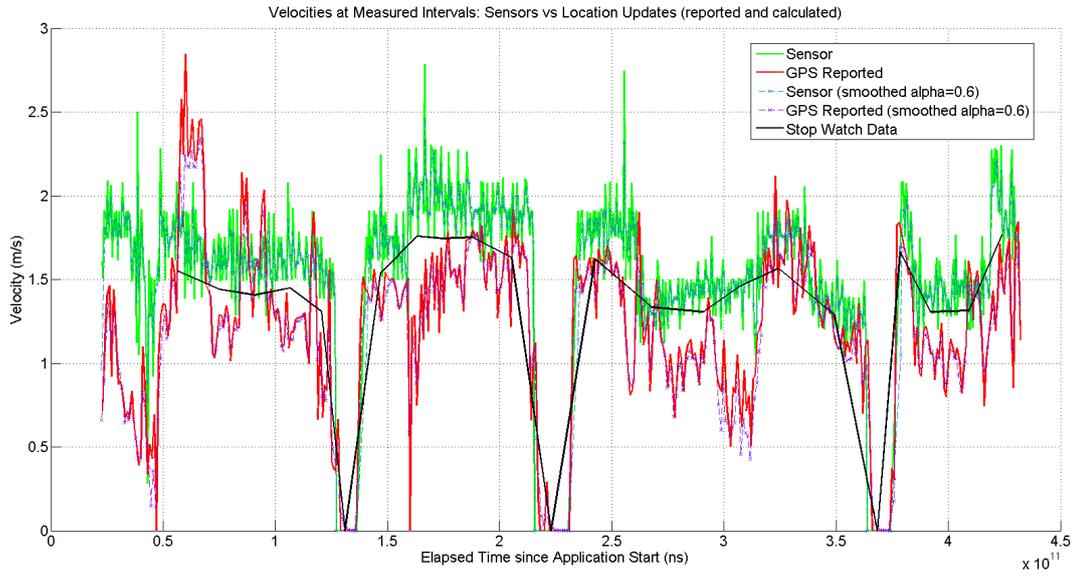
Additional Grpaphs

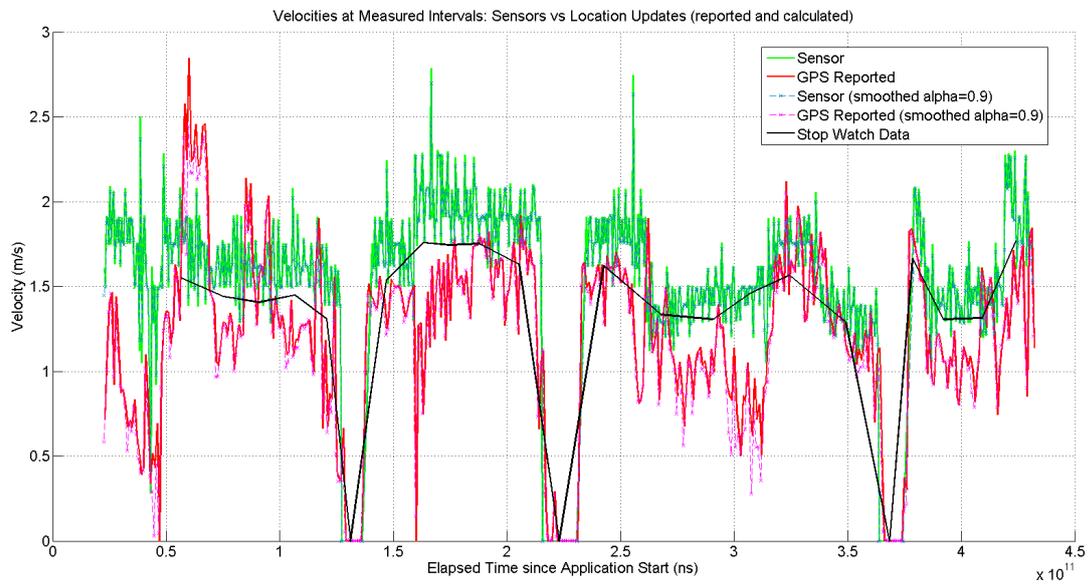
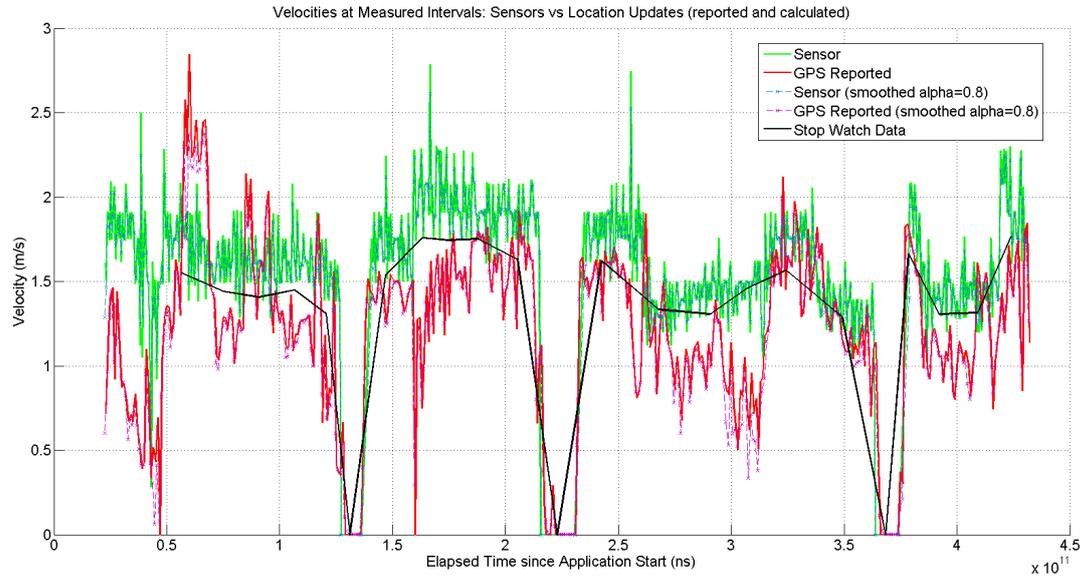
A.1 Outdoor Test: Gait Profile (sensor) vs GPS Speeds vs Ground Truth. Individual Smoothing Factor



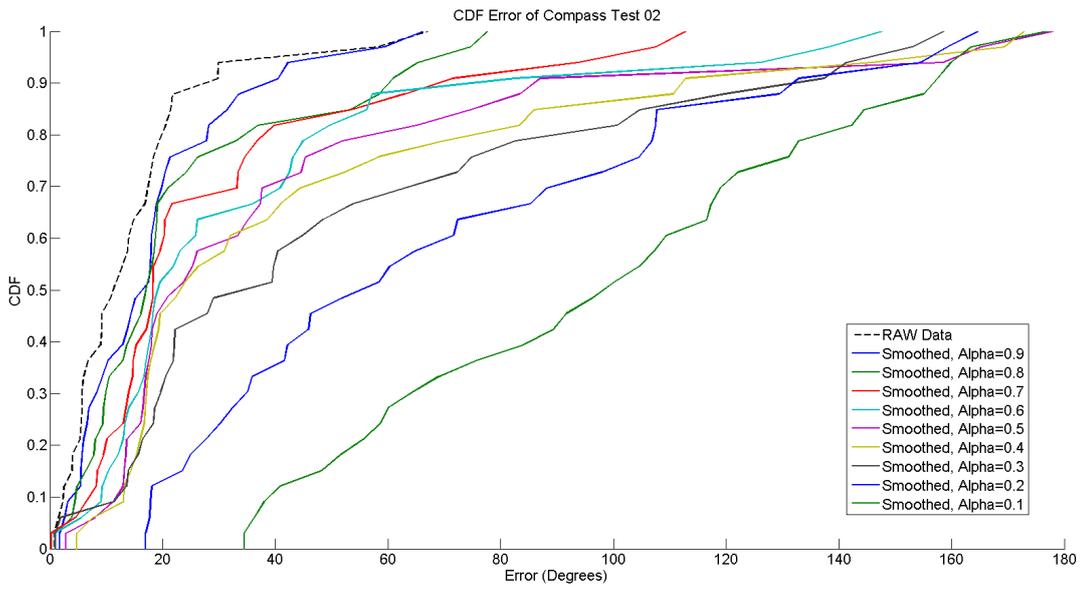
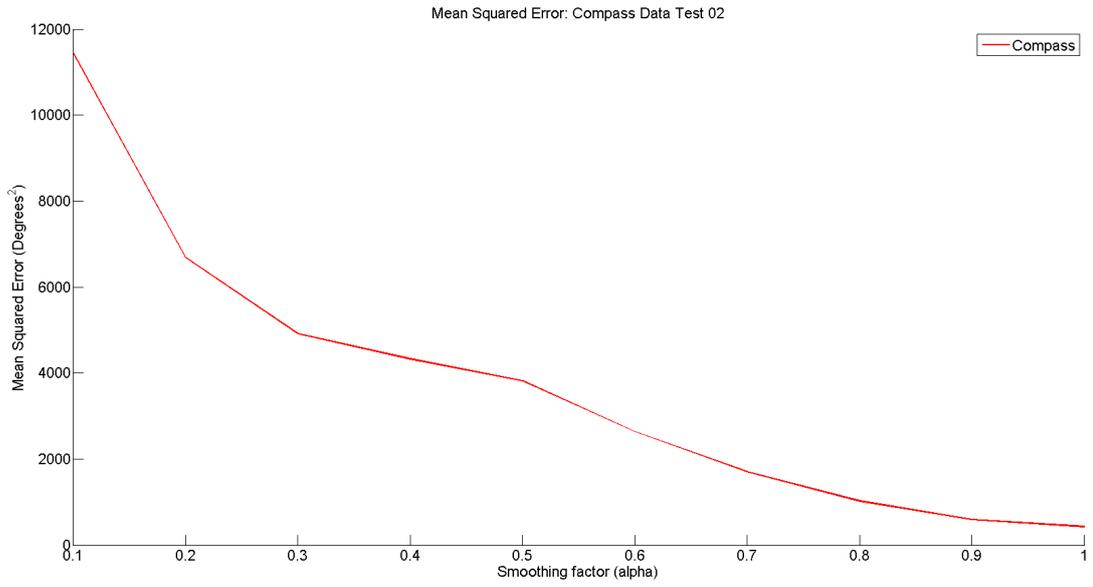




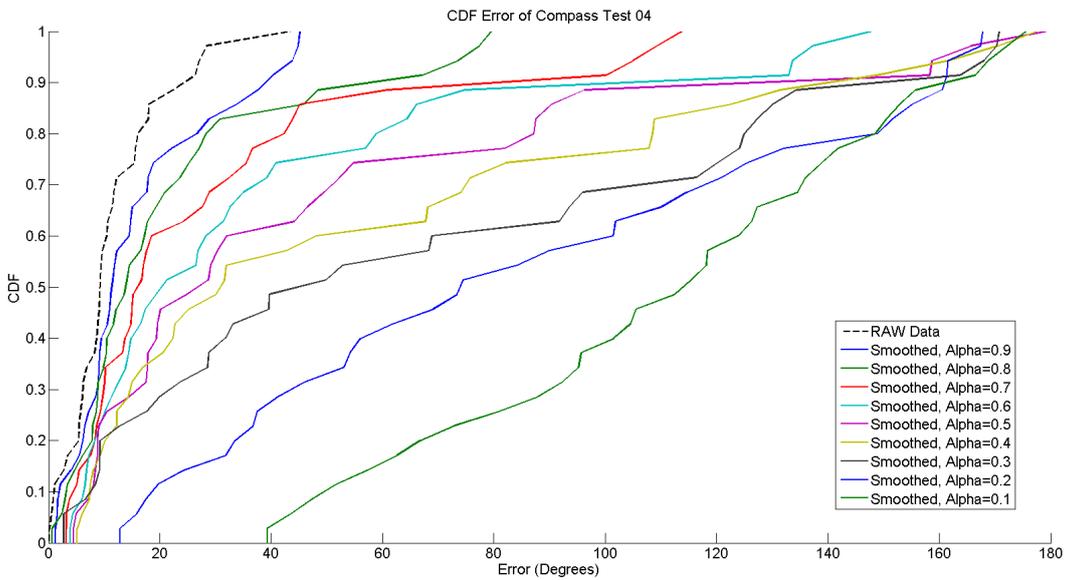
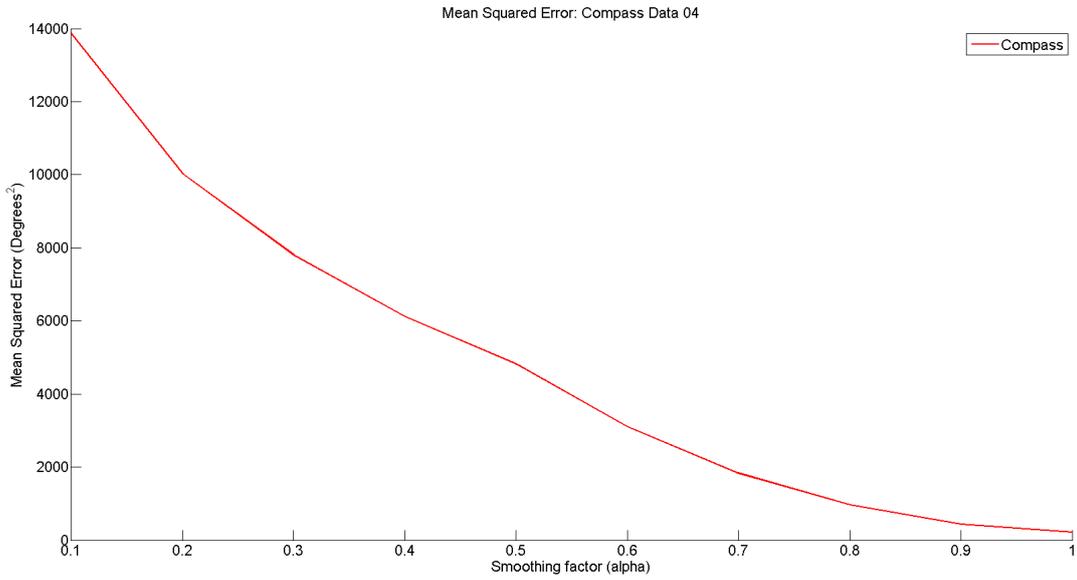


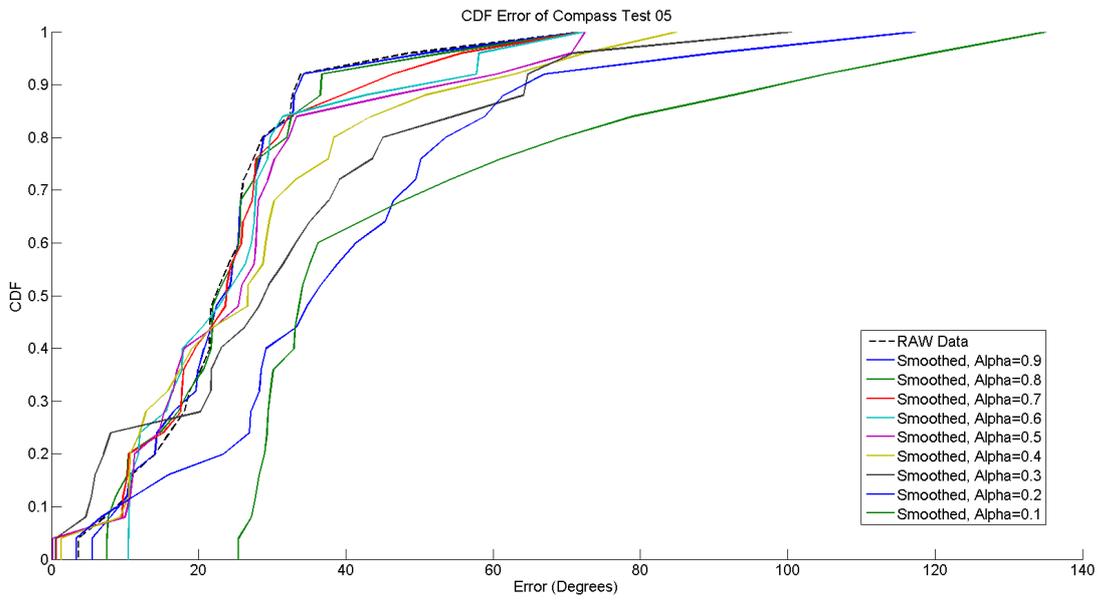
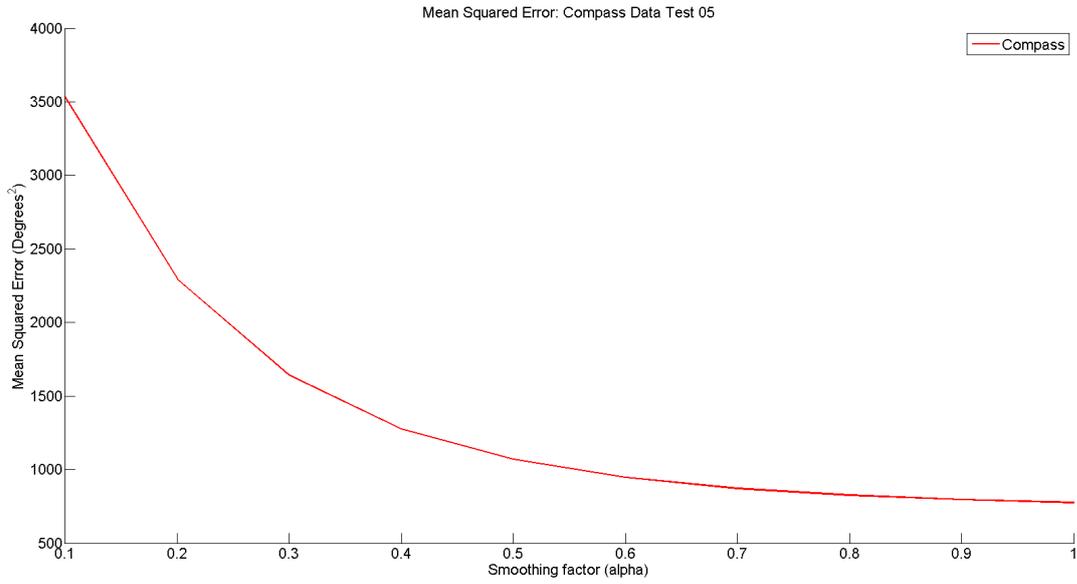


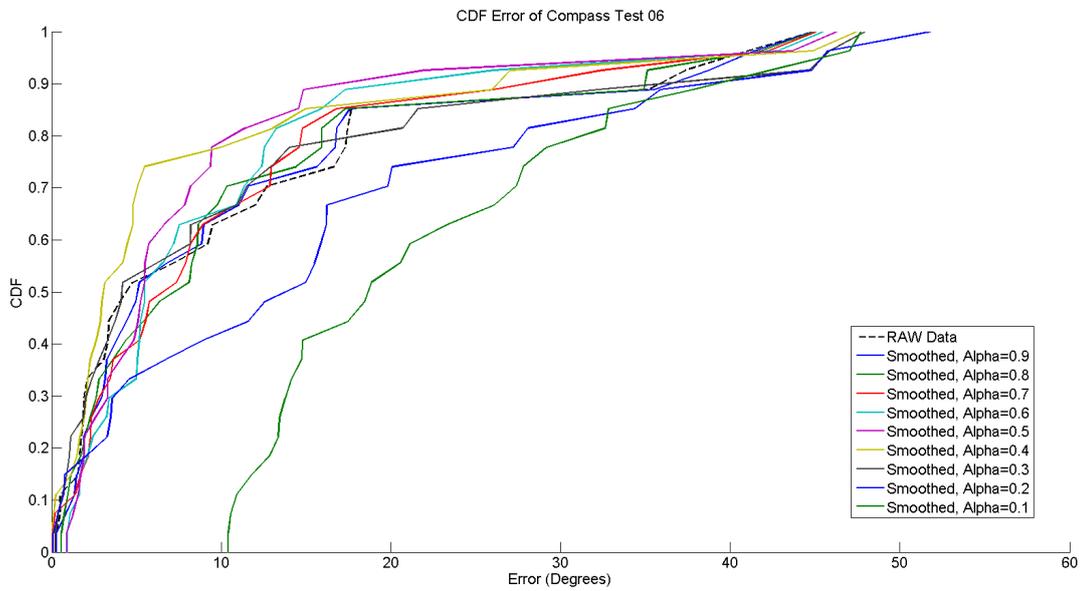
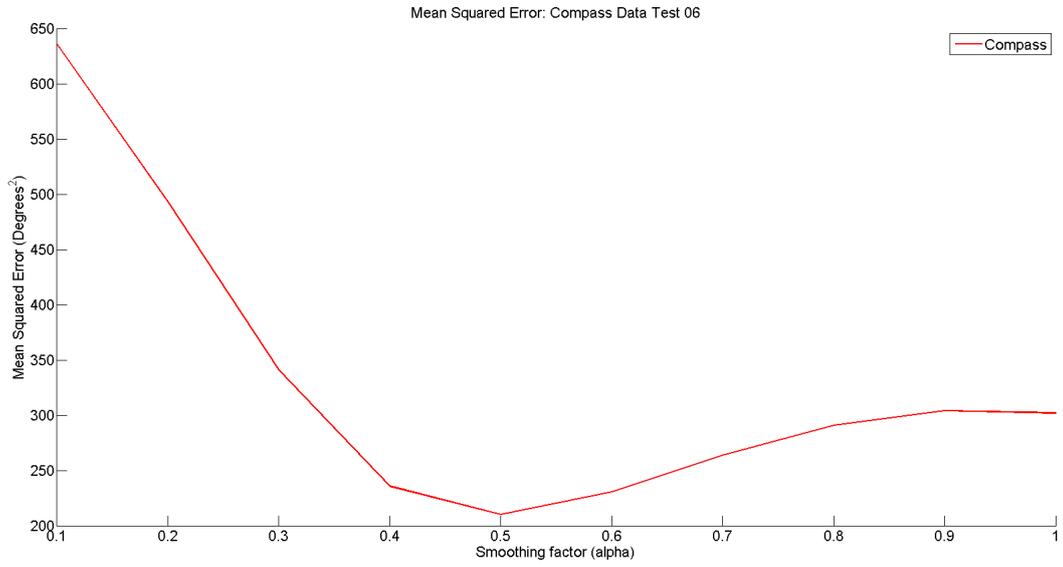
A.2 Direction Error Test (02-06)

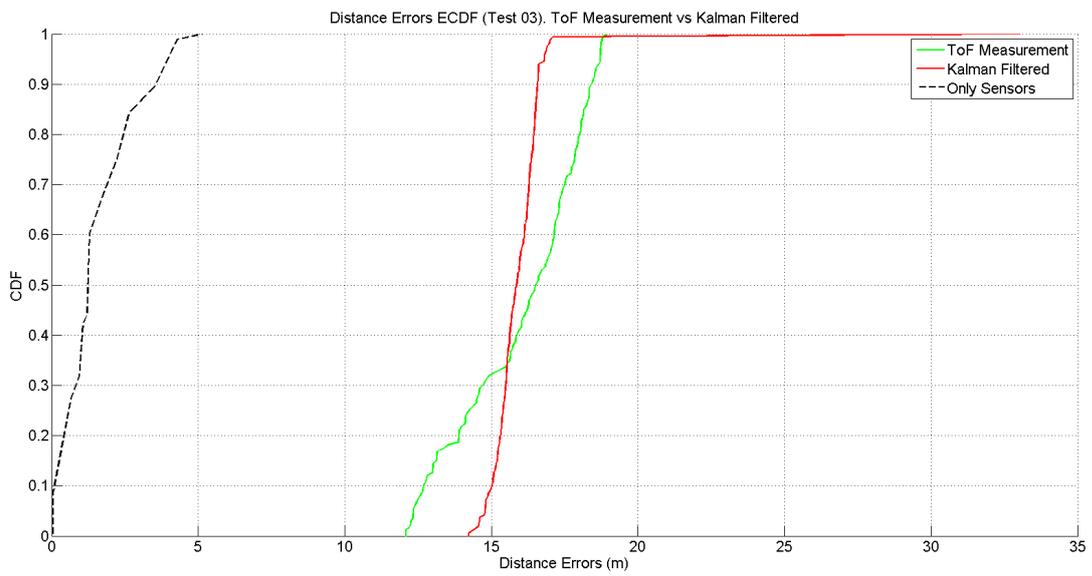
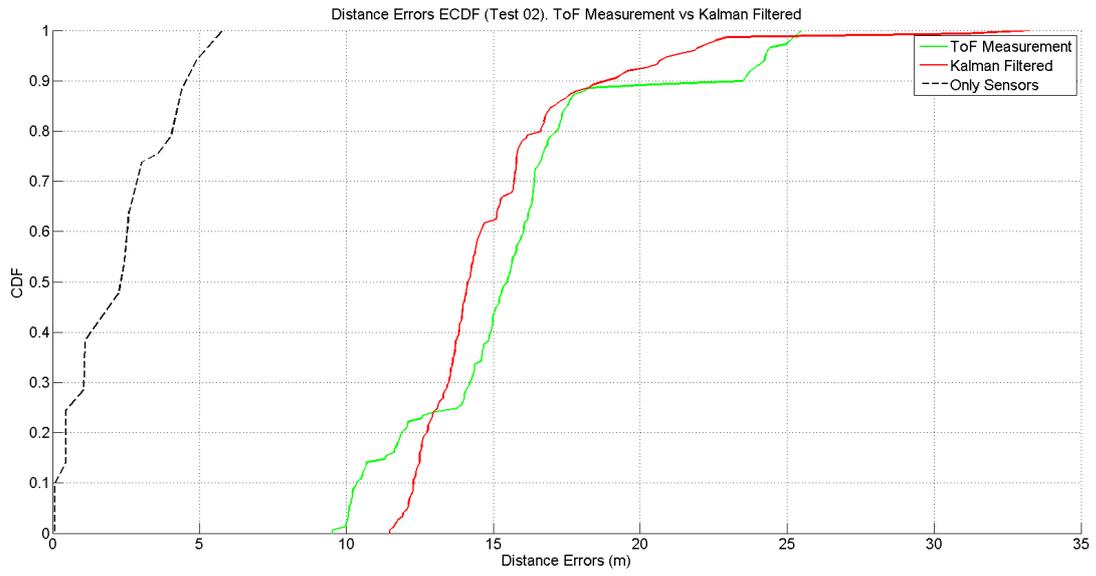


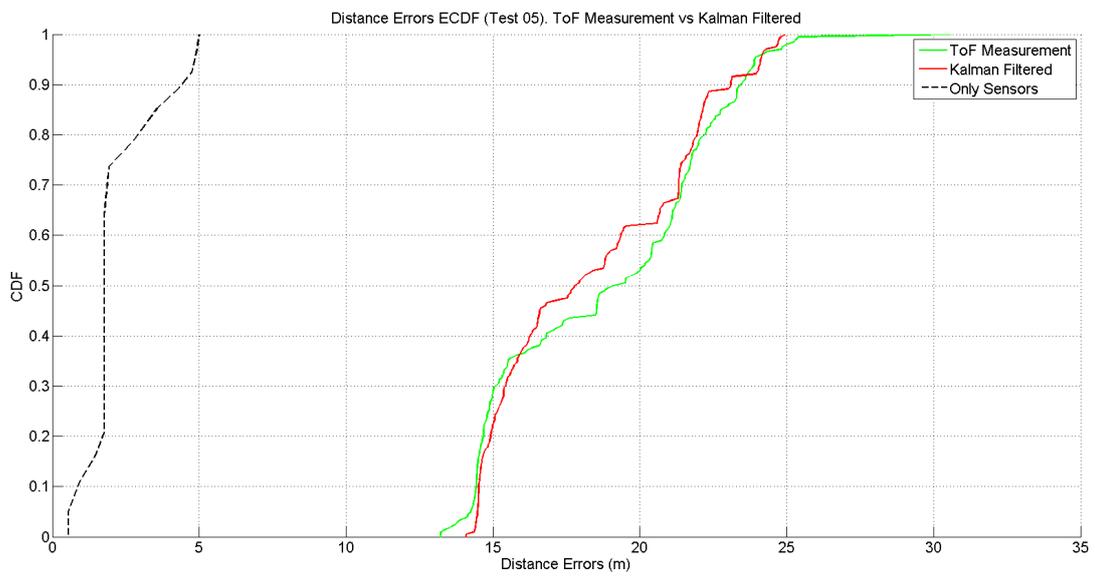
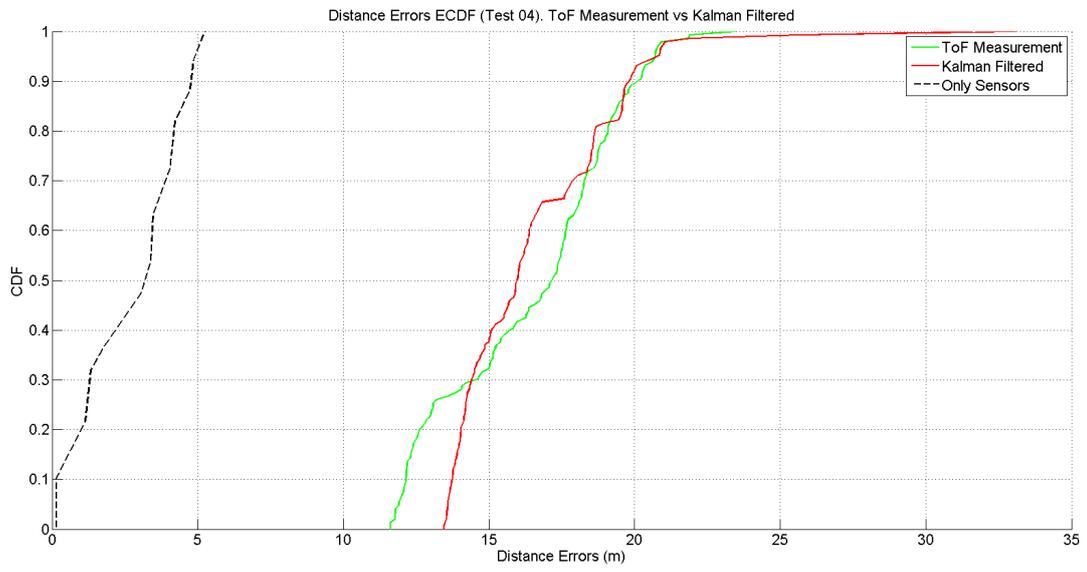
A.3 Distance Error Test (02-06)

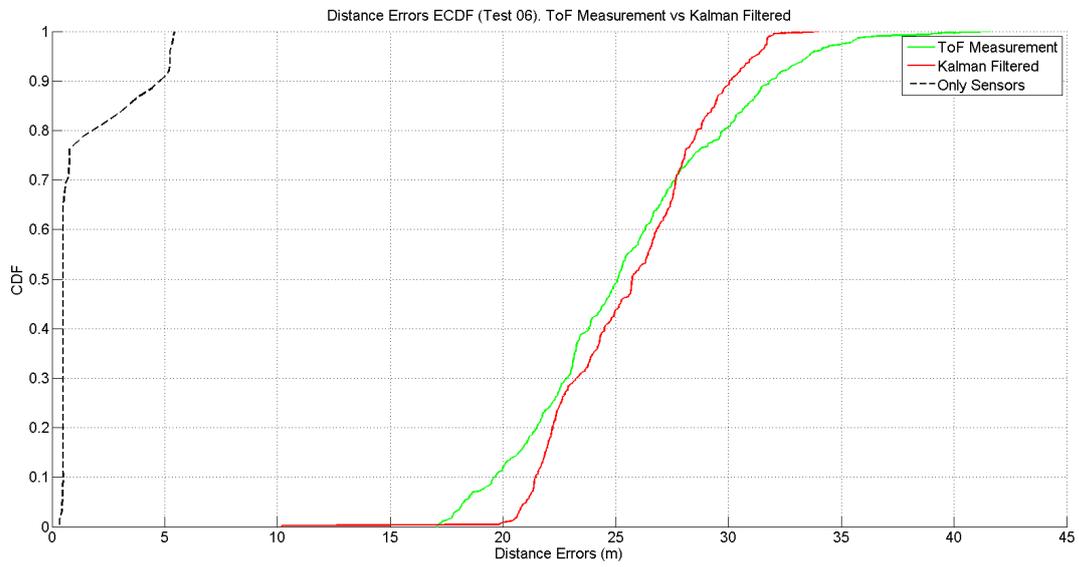












Appendix B

Additional Tables

Way Point	Coordinates (Before Rotation)	Coordinates (After Rotation)
A	(7.1145, 1.4200)	(0.5424, -7.2345)
B	(7.6875, 3.3750)	(2.4130, -8.0415)
C	(8.9585, 7.0951)	(5.9504, -9.7564)
D	(11.8905, 7.5575)	(6.0521, -12.7229)
E	(11.5689, 9.8175)	(8.3344, -12.6791)
F	(13.2295, 7.2190)	(5.5529, -14.0107)
G	(17.2936, 8.2865)	(6.1172, -18.1746)
H	(20.8850, 7.3451)	(4.7451, -21.6245)
I	(25.4150, 7.1301)	(3.9796, -26.0945)
J	(25.3950, 4.3501)	(1.2228, -25.7359)

Table B.1: Way Point Coordinates of Testbed 2

Appendix C

Original Assignment

Master Thesis Task Assignment of: Ka Kei Yeung

Smartphone app for Fused Indoor Localization

Advisors Prof. Dr. Giustiniano and Dr. Lenders
Supervisor Prof. Dr. Plattner
Start Date 5th of November, 2014
End Date 4th of May, 2015

1 Background

Localization using the Time-of-Flight (ToF) of RF signals is today probably the most popular technique to track moving objects. The most prominent usage of ToF for localization purposes is the Global Positioning System (GPS) which exploits differences in signal propagation times between different satellites to provide location services to more than one billion of mobile devices on earth. Also, radar systems rely on the propagation time of RF signals to localize hundreds of thousands of aircrafts per day.

While the ToF technique has been highly successful in these application domains, its success for WiFi-based indoor tracking has been relatively modest so far. The research community has instead focused more intensively on different approaches such as the signal strength [1, 2, 3, 4, 5], the angle of arrival [6, 7, 8]. Previous efforts to make use of the ToF for ranging using WiFi signals have reported relatively inaccurate results both in static [9, 10, 11, 12, 13] and tracking conditions [14], or [15].

In this thesis, we aim to enhance the accuracy of ToF positioning by combining it with data of inertial sensors such as accelerometer and gyroscopes, largely available in off-the-shelf smartphones [16, 17, 18]. The closer to our project is [15], which however required sophisticated APs. Requirements in the hardware hinder the adoption of these techniques for wide-spread and low-cost deployments in the wild. Thus, we make simplistic assumptions on the AP hardware.

The goal of this project is to move ToF positioning from infrastructure-centric system to device-centric system. In our expectation, a smartphone should be capable of geo-locating itself and the infrastructure should help the smartphone to locate itself when available. Improvement in the accuracy and performance of the positioning system shall be expected when GPS (in outdoor) or WiFi ToF (in indoor) are available for location support.

2 Motivation

In our recent research [19], we have demonstrated that ToF based indoor tracking running exclusively on *commercial off-the-shelf* (COTS) WiFi hardware is feasible and can achieve a localization accuracy that is competitive with other approaches. In April 2014, at the IEEE/ACM IPSN 2014 indoor localization competition in Berlin which involved 22 independent approaches from different research teams, we were able to achieve a 3.47m average positioning error reaching the 13th rank.

While two-way ranging has proven to work, it has two main drawbacks. First, two-way ranging produces a lot of traffic since every access point needs to inject his own packets to determine the ToF to the targets. When localizing multiple targets simultaneously, the overhead can get quite significant. We expect that the fusion of ToF with inertial sensor will have benefit both in terms of accuracy and traffic overhead.

In the first stage, the smartphone will compute the velocity, which will be then used by ToF system in [19] to determine the device's position by means of Kalman filter [15]. In the second stage, the inertial sensors will further compute the steps and the walking direction in real time using the embedded sensors in the smartphone, which in turn will determine the position of the device [17, 18]. Since inertial positioning is affected by dead-reckoning problem, the positioning will be corrected again by the ToF WiFi, this time using sensor fusion algorithms.

3 Tasks

The tasks of this thesis to reach a grade of 5.0 are described in what follows:

- Study the current WiFi based indoor localization system developed by your advisors [19] and the literature on localization using smartphone sensors [16, 17, 18].
- Implement a smartphone app running on Android devices. The first step will be to gather data with the internal inertial sensors and implement appropriate smoothing filters of the noisy sensed data.
- Report the velocity estimated by the smartphone to the ToF system [19]. The server will receive the data of the mobile's velocity and will compute the device's position.
- In the next phase, the student will use sensor data such as the one provided by accelerometer and gyroscope, and implement classical algorithms for positioning using sensor fusion available in the literature and propose potential improvements.
- Integrate the app with the ToF system to request updates of the position just when strictly needed. The integration may consist of querying the server for a new position fix to avoid the dead-reckoning problem. Delays in the communication network may be expected, and should be taken into account in the solution.
- Evaluate through experiments the accuracy and the communication overhead of the ToF+sensors localization system and compare it to the performance of i) the existing two-way ranging system and ii) a positioning system solely based on smartphone sensors. Evaluate the trade-off between the higher accuracies of the estimated distance when frequent queries are done to the WiFi positioning system, and the overhead for the network throughput. For this use the testbed at Armasuisse.
- – Optional task – Introduce a map for visualizing all the positions (for administrator) or smartphone's position (client localization) within the smartphone's app.

- – Optional task – If possible, consider energy consumption of the smartphone for the design [16]. Energy consumption of the smartphone should be carefully characterized in order to avoid draining the battery. In a real ToF positioning system, the decision about infrastructure-centric versus device-centric should be taken dynamically based on system constraints (energy consumption, network usage, positioning accuracy requested by the application, etc).

Higher grades can be reached if the work quality goes beyond the expectation above. A considerable independent contribution from the student would lead to a grade of 5.5. Work that would lead to a scientific paper may be considered for a grade of 6.0.

4 Deliverables

- At the end of the second week, a detailed time schedule of the thesis must be given and discussed with the main advisors.
- At the end of the second month, a short discussion of 15 minutes with the supervisor and the advisors will take place. The student has to talk about the major aspects of the ongoing work using slides.
- At the end of month four, another meeting with the supervisor will take place. At this point, the student should already have a preliminary version of the written report or at least a table of content to hand in to the supervisor. This preliminary version should be brought along to the short discussion.
- At the end of the thesis, a presentation of 15 minutes must be given at armasuisse and at ETH (in English) during a CSG group meeting. The presentations should give an overview as well as the most important details of the work.
- The final report should be written in English but may be written in German. It must contain a summary written in both English and German, the assignment and the time schedule. Its structure should include an introduction, an analysis of related work, and a complete documentation of all used hardware/software tools. Exceptionally, if the work results in a publication, it may be considered to present the publication as final report. Four written copies of the final report must be delivered to the main advisor along with CD that includes developments undergone during the thesis.

References

- [1] P. Bahl and V. Padmanabhan, "RADAR: an in-building RF-based user location and tracking system," in *Proc. IEEE INFOCOM'00*, Mar. 2000.
- [2] M. Youssef and A. Agrawala, "The horus wlan location determination system," ser. *MobiSys '05*. New York, NY, USA: ACM, 2005, pp. 205–218. [Online]. Available: <http://doi.acm.org/10.1145/1067170.1067193>
- [3] H. Lim, L.-C. Kung, J. Hou, and H. Luo, "Zero-configuration, robust indoor localization: Theory and experimentation," in *INFOCOM 2006. 25th IEEE International Conference on Computer Communications. Proceedings*, April 2006, pp. 1–12.

- [4] A. Goswami, L. E. Ortiz, and S. R. Das, “Wigem: A learning-based approach for indoor localization,” ser. CoNEXT ’11. New York, NY, USA: ACM, 2011, pp. 3:1–3:12. [Online]. Available: <http://doi.acm.org/10.1145/2079296.2079299>
- [5] K. Chintalapudi, A. Padmanabha Iyer, and V. N. Padmanabhan, “Indoor localization without the pain,” in *In Proc. of ACM MobiCom ’10*, 2010, pp. 173–184. [Online]. Available: <http://doi.acm.org/10.1145/1859995.1860016>
- [6] J. Xiong and K. Jamieson, “Arraytrack: A fine-grained indoor location system,” in *Presented as part of the 10th USENIX Symposium on Networked Systems Design and Implementation (NSDI 13)*. Lombard, IL: USENIX, 2013, pp. 71–84. [Online]. Available: <https://www.usenix.org/conference/nsdi13/technical-sessions/presentation/xiong>
- [7] S. Sen, J. Lee, K.-H. Kim, and P. Congdon, “Avoiding multipath to revive inbuilding wifi localization,” in *Proceeding of the 11th Annual International Conference on Mobile Systems, Applications, and Services*, ser. MobiSys ’13. New York, NY, USA: ACM, 2013, pp. 249–262. [Online]. Available: <http://doi.acm.org/10.1145/2462456.2464463>
- [8] J. Gjengset, J. Xiong, G. McPhillips, and K. Jamieson, “Enabling phased array signal processing on commodity wifi access points,” ser. Mobicom ’14. New York, NY, USA: ACM, 2014.
- [9] X. Li, K. Pahlavan, M. Latva-aho, and M. Ylianttila, “Comparison of indoor geolocation methods in dsss and ofdm wireless lan systems,” in *IEEE Fall VTC 2000*, vol. 6, 2000, pp. 3015–3020 vol.6.
- [10] D. McCrady, L. Doyle, H. Forstrom, T. Dempsey, and M. Martorana, “Mobile ranging using low-accuracy clocks,” *Microwave Theory and Techniques, IEEE Transactions on*, vol. 48, no. 6, pp. 951–958, 2000.
- [11] A. Günther and C. Hoene, “Measuring round trip times to determine the distance between wlan nodes,” ser. NETWORKING’05. Berlin, Heidelberg: Springer-Verlag, 2005, pp. 768–779. [Online]. Available: http://dx.doi.org/10.1007/11422778_62
- [12] M. Ciurana, F. Barcelo-Arroyo, and F. Izquierdo, “A ranging system with IEEE 802.11 data frames,” in *Radio and Wireless Symposium, 2007 IEEE*, 2007, pp. 133–136.
- [13] S. A. Golden and S. S. Bateman, “Sensor measurements for Wi-Fi location with emphasis on time-of-arrival ranging,” *IEEE Trans. Mobile Comput.*, vol. 6, no. 10, pp. 1185–1198, Oct. 2007. [Online]. Available: <http://portal.acm.org/citation.cfm?id=1313052.1313251>
- [14] D. Giustiniano and S. Mangold, “CAESAR: carrier sense-based ranging in off-the-shelf 802.11 wireless LAN,” in *Proceedings of the Seventh Conference on emerging Networking EXperiments and Technologies*. ACM, 2011, p. 10.
- [15] A. T. Mariakakis, S. Sen, J. Lee, and K.-H. Kim, “Sail: Single access point-based indoor localization,” in *Proceedings of the 12th Annual International Conference on Mobile Systems, Applications, and Services*, ser. MobiSys ’14. New York, NY, USA: ACM, 2014, pp. 315–328. [Online]. Available: <http://doi.acm.org/10.1145/2594368.2594393>
- [16] J. Paek, J. Kim, and R. Govindan, “Energy-efficient rate-adaptive gps-based positioning for smartphones,” in *Proceedings of the 8th international conference on Mobile systems, applications, and services*. ACM, 2010, pp. 299–314.

- [17] X. Zhu, Q. Li, and G. Chen, “Apt: Accurate outdoor pedestrian tracking with smartphones,” in *INFOCOM, 2013 Proceedings IEEE*. IEEE, 2013, pp. 2508–2516.
- [18] A. Rai, K. K. Chintalapudi, V. N. Padmanabhan, and R. Sen, “Zee: zero-effort crowdsourcing for indoor localization,” in *Proceedings of the 18th annual international conference on Mobile computing and networking*. ACM, 2012, pp. 293–304.
- [19] M. Rea, A. Marcaletti, A. Fakhreddine, D. Giustiniano, and V. Lenders, “Towards commoditized time-of-flight based mobile device tracking,” *Submitted to JSAC, special issue on location-awareness for radio and networks*, 2014.

Bibliography

- [1] Paramvir Bahl and Venkata N Padmanabhan. Radar: An in-building rf-based user location and tracking system. In *INFOCOM 2000. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, volume 2, pages 775–784. Ieee, 2000.
- [2] Moustafa Youssef and Ashok Agrawala. The horus wlan location determination system. In *Proceedings of the 3rd International Conference on Mobile Systems, Applications, and Services*, MobiSys '05, pages 205–218, New York, NY, USA, 2005. ACM.
- [3] Hyuk Lim, Lu-Chuan Kung, J.C. Hou, and Haiyun Luo. Zero-configuration, robust indoor localization: Theory and experimentation. In *INFOCOM 2006. 25th IEEE International Conference on Computer Communications. Proceedings*, pages 1–12, April 2006.
- [4] Abhishek Goswami, Luis E. Ortiz, and Samir R. Das. Wigem: A learning-based approach for indoor localization. In *Proceedings of the Seventh Conference on Emerging Networking EXperiments and Technologies*, CoNEXT '11, pages 3:1–3:12, New York, NY, USA, 2011. ACM.
- [5] Krishna Chintalapudi, Anand Padmanabha Iyer, and Venkata N. Padmanabhan. Indoor localization without the pain. In *Proceedings of the Sixteenth Annual International Conference on Mobile Computing and Networking*, MobiCom '10, pages 173–184, New York, NY, USA, 2010. ACM.
- [6] Jie Xiong and Kyle Jamieson. Arraytrack: A fine-grained indoor location system. In *Presented as part of the 10th USENIX Symposium on Networked Systems Design and Implementation (NSDI 13)*, pages 71–84, Lombard, IL, 2013. USENIX.
- [7] Souvik Sen, Jeongkeun Lee, Kyu-Han Kim, and Paul Congdon. Avoiding multipath to revive inbuilding wifi localization. In *Proceeding of the 11th Annual International Conference on Mobile Systems, Applications, and Services*, MobiSys '13, pages 249–262, New York, NY, USA, 2013. ACM.
- [8] Jon Gjengset, Jie Xiong, Graeme McPhillips, and Kyle Jamieson. Phaser: Enabling phased array signal processing on commodity wifi access points. In *Proceedings of the 20th Annual International Conference on Mobile Computing and Networking*, MobiCom '14, pages 153–164, New York, NY, USA, 2014. ACM.
- [9] Xinrong Li, K. Pahlavan, M. Latva-aho, and M. Ylianttila. Comparison of indoor geolocation methods in dsss and ofdm wireless lan systems. In *Vehicular Technology Conference, 2000. IEEE-VTS Fall VTC 2000. 52nd*, volume 6, pages 3015–3020 vol.6, 2000.
- [10] D.D. McCrady, L. Doyle, H. Forstrom, T. Dempsey, and M. Martorana. Mobile ranging using low-accuracy clocks. *Microwave Theory and Techniques, IEEE Transactions on*, 48(6):951–958, Jun 2000.
- [11] André Günther and Christian Hoene. Measuring round trip times to determine the distance between wlan nodes. In Raouf Boutaba, Kevin Almeroth, Ramon Puigjaner, Sherman Shen, and JamesP. Black, editors, *NETWORKING 2005. Networking Technologies, Services, and Protocols; Performance of Computer and Communication Networks; Mobile and Wireless Communications Systems*, volume 3462 of *Lecture Notes in Computer Science*, pages 768–779. Springer Berlin Heidelberg, 2005.

- [12] M. Ciurana, F. Barcelo-Arroyo, and F. Izquierdo. A ranging system with ieee 802.11 data frames. In *Radio and Wireless Symposium, 2007 IEEE*, pages 133–136, Jan 2007.
- [13] Stuart A. Golden and Steve S. Bateman. Sensor measurements for wi-fi location with emphasis on time-of-arrival ranging. *IEEE Transactions on Mobile Computing*, 6(10):1185–1198, October 2007.
- [14] Domenico Giustiniano and Stefan Mangold. Caesar: Carrier sense-based ranging in off-the-shelf 802.11 wireless lan. In *Proceedings of the Seventh Conference on Emerging Networking EXperiments and Technologies, CoNEXT '11*, pages 10:1–10:12, New York, NY, USA, 2011. ACM.
- [15] Alex T. Mariakakis, Souvik Sen, Jeongkeun Lee, and Kyu-Han Kim. Sail: Single access point-based indoor localization. In *Proceedings of the 12th Annual International Conference on Mobile Systems, Applications, and Services, MobiSys '14*, pages 315–328, New York, NY, USA, 2014. ACM.
- [16] Jeongyeup Paek, Joongheon Kim, and Ramesh Govindan. Energy-efficient rate-adaptive gps-based positioning for smartphones. In *Proceedings of the 8th International Conference on Mobile Systems, Applications, and Services, MobiSys '10*, pages 299–314, New York, NY, USA, 2010. ACM.
- [17] Xiaojun Zhu, Qun Li, and Guihai Chen. Apt: Accurate outdoor pedestrian tracking with smartphones. In *INFOCOM, 2013 Proceedings IEEE*, pages 2508–2516. IEEE, 2013.
- [18] Anshul Rai, Krishna Kant Chintalapudi, Venkat Padmanabhan, and Rijurekha Sen. Zee : Zero-effort crowdsourcing for indoor localization. In *Mobicom*, August 2012.
- [19] Andreas Marcaletti, Maurizio Rea, Domenico Giustiniano, Vincent Lenders, and Aymen Fakhreddine. Filtering noisy 802.11 time-of-flight ranging measurements. In *Proceedings of the 10th ACM International on Conference on Emerging Networking Experiments and Technologies, CoNEXT '14*, pages 13–20, New York, NY, USA, 2014. ACM.
- [20] National Centers for Entironmental Information. Magnetic Field Calculator.
- [21] Dries Vandermeulen, Charles Vercauteren, and Maarten Weyn. Indoor localization using a magnetic flux density map of a building. In *AMBIENT 2013, The Third International Conference on Ambient Computing, Applications, Services and Technologies*, pages 42–49, 2013.
- [22] Android Inc. Android API 4.4, New Sensor Types.
- [23] Richard B Langley. Dilution of precision. *GPS world*, 10(5):52–59, 1999.
- [24] Jason Zhang, Kefei Zhang, Ron Grenfell, and Rod Deakin. *Journal of Geodesy*, 80(2):104–110, 2006.
- [25] Takashi Watanabe, Hiroki Saito, Eri Koike, and Kazuki Nitta. A preliminary test of measurement of joint angles and stride length with wireless inertial sensors for wearable gait evaluation system. *Intell. Neuroscience*, 2011:6:1–6:12, January 2011.