RICE UNIVERSITY

Location Estimation Through Inexact Machine Learning Approach

by

Juan Jose Gonzalez Espana

A Thesis Submitted
in Partial Fulfillment of the
Requirements for the Degree

Master of Science

Approved, Thesis Committee:

Krishna Palem, Chair
Professor of Electrical and Computer
Engineering and Computer Science

Behnaam Aazhang
J.S. Abercrombie Professor of Electrical
and Computer Engineering

Anshumali Shrivastava
Assistant Professor of Computer Science

Houston, Texas

October, 2018

ABSTRACT

Location Estimation Through Inexact Machine Learning Approach

by

Juan Jose Gonzalez Espana

Location estimation has become a field of increasing interest in recent years. The main reason is the multiple applications that can be enabled based on this technology. Fields such as entertainment, health care, tourism and advertisement are some of the areas where a plethora of applications can be implemented. In outdoors this problem is solved, for most of the cases, with Global Navigation Systems (GNSS). However, in indoors is a current topic of interest that has been addressed from different perspectives with different technologies. Nonetheless, there is no technology that is as established as GNSS is for outdoors. One promising approach is Inertial Measurement Units (IMU) which are low cost and widely accessible in multiple SmartDevices such SmartPhones, SmartWatches, WristBands, among others. Two of the main difficulties that hinder the wide adoption of this technology are the error accumulation between estimations and the scarce availability of the Ground Truth data to train and test the models. In this work both challenges are addressed by two methods, one which corrects the error by using the structure of the map where the user is located and the other method improves the Ground Truth data provided by GNSS measurements. Energy consumption is reduced by a factor $27x$ when compared with GPS and the accuracy of the labels is improved by 26% on average.

Contents

Illustrations

Tables

Chapter 1

Introduction

Real Time Locating Systems (RTLSs) and Indoor Positioning Systems (IPSs) is expected to provide market opportunities on the order of 10 billion yearly in 2024. [1]. Many technologies have been proposed to provide solutions in this field. However, they are partial because they can be implemented under specific conditions or environments [2, 3]. For example, Global Navigation Systems (GNSS) are successfully implemented in outdoors [4]. Nonetheless, in indoors they fail because the GNSS signal is highly attenuated by the walls of the building [3]. Other approaches require modifying the environment in order to install enabling technologies, which represents an additional cost and raises privacy concerns because the user location is constantly known by a third party [3].

Inertial Measurement Units (IMU) provides a solution which is not intrusive and not require the installation of additional technologies but suffer from high error because the cumulative error nature of the models. This is not only a problem of IMU, other technologies that estimates a variable without compensation from the environment, tends to accumulate error over time, for example every clock suffers an error accumulation due to the timing jitter, which requires a correction depending on the accuracy of the process [5]. Similarly, in position estimation depending on the application, the accuracy and the periodicity of the error correction is defined.

In this master thesis the ideas from Inexactness [6–9], Inertial Measurements Units and Machine learning are combined to predict the position of the user in a map. The correction of the error accumulation is based on the constraints defined by the map which is an option that doesn't require additional technologies to correct the error avoiding the reliance in exogenous information which translates in a more private position estimation [3, 10–12].

Also it's proposed a method which automate the process of obtaining Ground Truth data without requiring the user walking/running in predefine maps. In this way is expected that the data collection will be more accessible to a wider populations based on the few requirements to collect data: using a Smartphone while the person is walking/running outdoors.

1.1 Contributions

Location estimation based on Inertial Measurement Units (IMU) suffers from two main drawbacks: 1. Error accumulation between estimations 2. Reliable and widely accessible method to obtain Ground Truth data to train and test the model. The first problem is addressed in this work using map information of the environment where the user is located. For the second case it's proposed a method which combines GPS labels based on the similitude of IMU measurements. To the best of our knowledge is the first time this kind of algorithm is proposed to train models based on IMU sensors.

1.2 Organization

The rest of this thesis is organized as follows. Chapter 2 provides Background which includes some alternatives to solve the current problem. Then, Chapter 3 describes the proposed method. Afterward, Chapter 4 presents our results. Chapter 5, illustrates some possible applications of the research in this thesis. Finally, Chapter 6 presents the Conclusions.

Chapter 2

Background

2.1 Position Estimation

2.1.1 Global Navigation Satellite Systems

Human navigation encompass traveling from an origin to a destination through the use of knowledge of the environment under exploration and the ability of the person to locate themselves in the environment. This knowledge comes from previous experiences of the person or interaction with other persons [13]. In the cases that the knowledge is absent or incomplete is necessary to use some tools to find the location of the person in a map and the correspondent progress from the origin to the goal. Nowadays, one of the more widely used technologies are based on Global Navigation Satellite Systems (GNSS). The GNSS, initially considered for military uses and made public since 1983 [13], requires that the user is in Line of Sight of four different satellites and have a GNSS receiver to obtain the longitude, latitude and altitude of their current location. Howbeit these values by themselves are not enough for a human-being to locate themselves, it's also necessary a map in a proper format to match these coordinates [13]. With the surge of mobile devices the availability of GNSS receivers and those maps are not constraints, which has enabled its widely used for different applications. Global Positioning System (GPS) from USA and GLObal NAvigation Satellite System (GLONASS) from Rusia, are the most common operational GNSS. Other GNSS alternatives are the European GALILEO system, the Chinese BeiDou

system, the Indian Regional Navigational Satellite System and in the future Japanese Quasi-Zenith Satellite System [14]. As it was mentioned GNSS are widely adopted for position estimation. Though the performance of GNSS is not always reliable and depending of static and dynamic conditions it can be affected.

Some of the most common problems in GNSS are multipaths and non-line-of-sight (NLOS) propagation. This is more common in urban canyons where flat surfaces reflect the GNSS signal creating multipath delays leading to a distance estimation longer than the real. If the effects of Multipath are combined with NLOS, the distance estimation from the GNSS becomes more inaccurate, even above $100m$ [14]. These problems are even worst for the GNSS sensors embedded in cellphones which are usually low-cost and single-frequency [15]. Some techniques are used to identify the NLOS satellites and do the correspondent corrections but this requires more than four visible satellites which usually is not the case [16]]. This is the reason why it has been proposed the combination of different GNSS systems which will lead to more accurate estimation [13, 17]. However, for the time being most of the smartphones rely only on one of those Satellite Systems and the errors mentioned here are solved using the information of other sensors available in the Smart device. For example, for devices that uses GPS it is called *Assisted GPS* [18]. In this case the celullar network (Celltower-RSS) is used to acquire additional information to reduce the error in the position calculations. On good multi-path conditions the accuracy could increase to be in the range between 3 and 5m. [16]. Besides the accuracy problems the energy consumption of GNSS is considerably high, which hampers the implementation of applications with high dependence on it [19]. Because of the high energy consumption, some approaches combined GPS with other technologies available in Smartphones

which could maintain similar accuracy with less frequent use of GPS. [15, 20, 21]

In general, it can be said that most of the time GPS solves the problem of outdoor navigation [4, 12]. However, for indoor navigation the GPS signals are tremendously attenuated and therefore is infeasible to use it for location detection and the associated navigation [4]. Then, while GPS remains king in outdoors, or GPS combined with other technologies, in indoors is necessary to find other solutions, which are non intrusive and accurate enough for the intended application [4].

2.1.2 Triangulation

In a similar line of thought than GNSS, the triangulation systems uses the known position of at least three points to identify the location of the user. For outdoors this can be achieved by GNSS, Cell-tower positioning [22], WLANS [23] and for indoors Radio Frequency Identifier Description (RFID) tags [24], ultrasound [25] and Infrared (IR) systems [26].As with GNSS, also these technologies suffer from inaccuracies due to the non-line-of-sight or multipaths.

2.1.3 Encoded Sensing

In this case a device broadcast a specific location and the user with a proper reader locate themselves on a map. Some examples of this technology are provided below. [2]

- **Radio Frequency Identifier Description (RFID)** passive or active RFID are installed on the environment and the location can be obtained using a RFID reader. The former has a larger range at the expense of requiring maintenance, as replacement of batteries is required. Other difference between the two technologies is the information they could store: the active until 128kb and the passive 128b. [2]

- **Barcodes** Similar to the previous case, barcodes are located in fixed locations. A barcode reader is necessary, which can be found in many smartphone devices. The main difficulty is that constantly reading the codes reduces the fluency of navigation in the environment.

- **Infrared** IR transmitters broadcast their ID which is associated to the location of their placement. As it is common in IR transmitters their transmission angle is very narrow which makes difficult to the user to locate them.

- **Bluetooth** Multiple Bluetooth beacons are used and depending on the user's device ability to communicate to the beacons the location can be estimated. This is different with the triangulation case because the position is not estimated in a discrete location but in a cell based method. [27]

In the technologies mentioned above the main constraint is that the environment must be modified in order to locate the user. Even more, in some case the user should wear special devices or act in specific manner which thwart their wide implementation. Following are presented some alternative which has taken significant interest in the research community. [2]

2.1.4 Pattern Recognition

In this case a initial stage of training takes place to identify features on the environment that will enable a system to locate the user when they are navigating the environment. As it can be seen, this technique cannot be used for non-previous seen environments. Some examples are provided in the next lines.

- **Computer Vision** A camera or the camera of handheld device is used to capture images from the location of the user. Then, image matching is done

between a database and the query image. The metadata of the location of the database matched image will be use to return the position and orientation of the user. [2] Usually this approach has a high requirements of storage and computing time. However, approaches such as [28] reduces the computing time and storage requirement by a clever use of hashes. Still the fluency of the user's navigation is reduced because of the constant need of taking pictures to find their location.

- **Signal Distribution** In this case is necessary a mapping of the environment of interest. In this mapping different signals in specific points are recorded and a matching is done when the user is navigating this same environment. Depending on the matching the location is identified. An example is the use of WLANs access points, where the dependence of signal strength and distance is foster to identify the position of the user. [2]. Other example is found creating a electromagnetic map of the environment and use a magnetometer to identify the position based on a previous mapping. [29]

2.1.5 Dead Reckoning

From an initial position provided by the user or recognized with the support of any of the previous methods, a sequence of estimations of the position are performed. These estimations are done based on the measurements from accelerometer, gyroscope or magnetometer. However, these estimates are not perfectly accurate and the error tends to accumulate over time because of the recursive approach to estimate distance. In this case also training is performed but it is mainly dependent on the user and not the specific environment [2, 3].

Correction in Dead Reckoning

The nature of Dead Reckoning leads to a cumulative error that increases with time. Which makes necessary to apply correction if the accuracy of the estimations wants to be maintained within a specific boundary [30]. Some of the common techniques for this purpose are presented below:

- Map Matching: in indoors the user movement is constrained by the shape and size of the building. This assumption has been used to correct the distance and angle estimation based on the Dead Reckoning. This is done by matching the prediction with the map of the building and choosing the most probable paths or adjusting the estimation. [10–12].

- Combination with Other Methods: the methods exposed in Sections 2.1.4, 2.1.3 and 2.1.2 don't accumulate the error between predictions which makes them useful to correct the error accumulation in Dead Reckoning. In this case Dead Reckoning will inherit the difficulties associated with those methods used in the correction process [2].

In this research Dead Reckoning was addressed and in the following section is described in detail the technology that is commonly used to implement it.

2.2 Inertial Measurement Unit

The Inertial Measurement Units (IMUs) has taken relevance in recent years mainly because of their low cost and availability. This low cost feature is a result of the improvement of the Micro-Machined Electromechanical Systems (MEMS) [31].This is also the reason that many of the wearable Smart-Devices has one or multiple IMUs. As a consequence, many applications have emerged to leverage the use of this

technology. In particular, location estimation covers many of the current efforts. Two of their main hurdles are the cumulative error, as is the nature of the Dead Reckoning approach (See Section 2.1.5), and obtaining reliable Ground Truth labels.
Following is explained each of the most common IMU found on SmartDevices.

2.2.1 Gryroscope

A gyroscope is defined as a device able to measure the angular speed of an object. There are different types of gyroscope, the main types are mechanical, optical and MEMS. The last one is the cheapest and smallest among the three and it is the one commonly used in Smart Devices [31]. The four different sources of error which affects MEMS gyroscopes are:

- **Constant bias:** it is the offset of the output from the true value. It's easily corrected by taking a long term average on the gyro's measurements when it's in steady state. For correction this value is subtracted from the new measurements. If this noise is not corrected it's effect increase linearly over time ($\epsilon \cdot t$)

- **White Noise:** corresponds to the thermo-mechanical noise. This noise is pervasive to all the frequency spectrum and it's difficult to eliminate it's effects completely. It can be represented by a sequence of Random Variables with zero mean and variance σ^2. It's effect over time is defined by a random variable with zero mean and variance equal to $\delta t \cdot n\sigma^2$, where δt is the time between successive samples. Specifications about this noise are provided by the manufacturer.

- **Temperature effects:** which affects the electronics of the device producing a variation in the bias.

- **Bias stability:** as a result of the electronic and other components which are

subject to random flickering. It has a $1/f$ power spectral density. It's a low frequency phenomenon which may change between different periods of time.

2.2.2 Accelerometer

Some of the most popular types of accelerometer are mechanical, solid state and MEMS. As in Section 2.2.1, the most widely adopted accelerometer is the MEMS. Additionally, it's the one in the devices focus of this research. Its mode of operation encompass two main approaches: mass displacement and resonance frequency variation. On the first one, the second law of Newton for constant mass ($F = m \cdot a$) is applied to obtain the acceleration of a supported mass. On the second case, the acceleration produces a change on tension of a beam which produces a change in the resonance frequency of the beam. This swift in frequency is measure to obtain the acceleration [31].

Many of the noises that affect gyroscopes, mentioned in Section 2.2.1, are similar to the ones which affect the MEMS acceloremeter. They are explained below:

- **Constant Bias:** in this case the no correction of this effect produces an error which increases quadratically with time. The main difficulty for its correction is the need of a system without acceleration of any kind which is not possible because of the gravity. One alternative is knowing the exact orientation of the device to make the correspondent corrections due to gravity.

- **White Noise:** as mentioned for the case of the gyro this noise is thermo-mechanical noise. The characteristics are very similar except for the variance of its random variable which now is $\frac{1}{3} \cdot \delta t \cdot t^3 \cdot \sigma^2$

- **Temperature effects:** again for this case the effect of temperature in the

electronics affects the noise associated with the bias.

- **Bias stability:** the uncertainty in distance in this case is proportionally to $t^{5/2}$.

These errors become significant with time for the methods based on the double integration to obtain displacement [31].

2.2.3 Magnetometer

The magnetometer is a electronic compass where the earth's magnetic field is measured in micro Teslas μT. The main drawback of this technology is that is affected by the ferrous and magnetized materials of the environment of implementation which are called as the Soft Iron and Hard Iron effects. Additionally, their performance depends on the actual latitude and longitude where the measurements are taken. On SmartPhones are available in three orthogonal axes which are used to estimate the Magnetic North. [32] While they are more inaccurate for angle estimation than the gyroscopes, their energy consumption is lower [33] which make them an attractive option to give a coarse estimation of the angle. Also they can be combined with other sensors to obtain a better estimation. [34]

2.3 Machine Learning

Machine Learning or Statistical Learning considers a machine which learns the underlying model of a phenomenon by the viewing of examples or training data. When the machine receives feedback about how close is to the model, we talked about *Supervised Learning*. If the machine doesn't receive feedback we talked about *Unsupervised Learning* [35]. This work is mainly focused with Supervised Learning and below are

explained some methods.

2.3.1 Regression

Consider the data pairs of the form x_i, y_i which are points in a $(p+1)$-dimensional space and they are related through the model:

$$y_i = f(x_i) + \epsilon \tag{2.1}$$

where usually x_i belongs to a p-dimensional space and ϵ is noise. [35] The process to find the function f is described as *Function Approximation* or *Regression*. In next lines some methods to achieve this goal are presented.

Nearest Neighbor Regression

If there is little knowledge about the shape of the function in 2.1 Nearest Neighbor Regression can be used. [36] This method relies on the assumption that points which are close should have a similar mean value if the mean function is smooth. Based on this the query data can be approximated by the label of it's nearest neighbor in a historical dataset.

This dataset is commonly referred as the training dataset, which is described as

$(x_1, y_1), ..., (x_i, y_i)...(x_N, y_N)$

where (x_i, y_i) is a sample, x_i is the feature vector i, y_i is the label correspondent to that feature vector and N is the number of samples. Then, the query feature vector x_q is predicted to have the label of the nearest neighbor from the training dataset, which can be written as:

$\hat{y} = arg_{y_i} min D(x_i, x_q)$

Where D is a similarity function. This similarity can be a distance metric using a specific $\ell - norm$ value depending on the characteristics of the data. [37].

Linear Regression (Ridge)

Linear Regression assumes that the underlying model in 2.1 is linear. Based on this a linear model is obtained minimizing the Mean Square error function. [38] Mathematically the model is:

$$Y = XB + \epsilon \tag{2.2}$$

where Y, X, B and ϵ are the labels, the matrix of feature vectors, the parameters of the model $B = \beta_1, \beta_2, \beta_3 ... \beta_p$ and the error, respectively.

Model 2.2 will lead to a successful model estimation if $X'X$ is nearly an unit matrix otherwise the residual sum of squares will be unsatisfactory with a high probability. This is the reason why in Ridge Regression [38] to address this B is estimated as:

$B = \underset{B=\beta_1,\beta_2,\beta_3 ... \beta_p}{\operatorname{argmin}} L(B)$

Where

$L(B) = \sum_{i=1}^{N}(B^T x - y_i)^2 + \mu \sum_{i=1}^{p} \beta_i$

Which is similar to 2.2 with the difference of the Ridge Regression constraint correspondent to the the second term.

Random Forest Regression

For the cases where 2.1 cannot be explained by a linear model other approaches such as Regression Tree can be used [39]. Particularly Regression Trees are powerful because they partition the dataset on subsets based on the features, which create smaller regions were less complex models can be fitted. These small regions are called *leaf-node*. While creating very small subdivisions could lead to overfit the data, Regression Trees provides other characteristic which make them successful in prediction: pruning. Pruning in this case is done to minimize the validation variance on the output variable.

On the leaf-node the classical approach obtains a model which is a constant estimate $\hat{y}$ which is the result of:

$\hat{y} = \sum_{i=1}^{c} y_i^j / c$

where $(x_1^j, y_1^j), (x_2^j, y_2^j)...(x_c^j, y_c^j)$ are the samples in node j and c is the number of samples in that node. For further details in this method please see [39].

2.3.2 Classification

In classification still the problem mentioned in Expression 2.1 is valid. However, in this case the y_i sample is restricted to take a discrete number of values which are not necessarily *quantitative* but also *qualitative*. Random Forest and Near Neighbor also can be used for classification and the details are not provided because they are very similar to which was mentioned in Section 2.3.1. One additional powerful method is Support Vector Machine and it's presented below.

Support Vector Machine

Support Vector Machines (SVM) relies on the linear separability of training data on feature space. [40] While many type of data are not linearly separable, SVM uses a kernel K which maps the data into a higher feature space where the data is linearly separable. [41]

From the training data the hyperplane that is chosen is the one that maximizes the margin in the feature space. The model that is obtained is used to classify new data based on which side of the hyperplane is the sample. Further details can be found in [40].

2.4 Inexactness

If different position estimation techniques mentioned in Section 2.1 are correctly combined, the accuracy may increase significantly. Nonetheless, the energy consumption of the system will increase accordingly. Furthermore, for some cases the frugality of the navigation will become significantly affected [2]. For example, if GNSS signal are sampled very often, the energy consumption will increase significantly while the accuracy won't see a significant improvement [20, 21]. Then, identifying the sweet spot where energy and accuracy are in a proper balance for the specific application is an important task for applications dependent on location. This trade off that can be seen vastly addressed in [6–9] shows the importance of allocating energy resources in a smart manner which will lead to a significant reduction in energy consumption while maintaining the accuracy within useful range.

Chapter 3

Methodology

As it was mentioned in Section 2.1.5 location estimation in humans based on IMU sensors requires a model and Ground Truth labels to train and test the model. There are two general approaches for the location estimation: parametric and biomechanical. In the biomechanical is commonly assumed that the device is in the Center of Mass of the user and models the legs as a pendulum. The parametric uses features from the signal to estimate the parameters of the correspondent model [42]. In the current work a parametric approach is considered because the flexibility that provides fixing the device in locations different than the Center of Mass of the user [42]. Additionally, Machine Learning will be used to obtain and tune the model because its ability to automate efficiently the process of obtaining and tunning the respective models [43]. In Section 3.1 is described the Ground Truth chosen to tune the parameters of the model and in Section 3.2 will be explained the Machine Learning model.

3.1 Ground Truth Generation

3.1.1 Types of Ground Truth

The model trained by Machine Learning will be in the best scenario as accurate as the Ground Truth data [44,45]. For this purpose the scenario where the data is taken is a controlled environment with the combination of multiple location measuring devices [3, 42, 46, 47] to obtain highly accurate labels to train and test their models. In

this work this type of Ground Truth data is called *Lab Maps* because the conditions to take the data are under the control of the researcher. However, obtaining highly accurate Ground Truth labels in a massive scale adds significant cost and complexity to the data recollection process [48]. One alternative for IMU is GNSS which is easily accessible from many Smart Devices making easy to collect massive amounts of data without requiring the user to do additional activities besides their daily routines or having additional devices. This type of Ground Truth data is called in this research as *GNSS map*. In this research we chose GPS because it's the more easily accessible GNSS in our work environment. However, before using raw GPS data to train a model, it's necessary to understand the error behavior of this technology. Particularly, how the error behaves in function of the time between measurements. In [49] is described the error for GPS measurements taken every 24 hours. Nonetheless, in this research the sampling frequency of the GPS signals is much higher and therefore it's necessary to see how it's affected the accuracy as the sampling frequency changes. For this purpose the Relative Standard Deviation (RSD) of the GPS error in distance estimation could gives insight in how precise and repeatable are GPS distance measurements . In Figure 3.1 is presented the results for different sampling periods, where it can be seen that in general shorter periods are associated with less precise GPS measurements. While it can be said that it's better to choose a longer sampling period to improve the accuracy of the GPS measurements is also true that in that case the inaccuracy will increase also as a consequence of the difference between the actual walking path of the user and the straight line assumed in GPS calculations [18,50]. To study in more detail the behavior of GPS error, Section 3.1.2 described the probable Error Probability Distribution that GPS error follows.

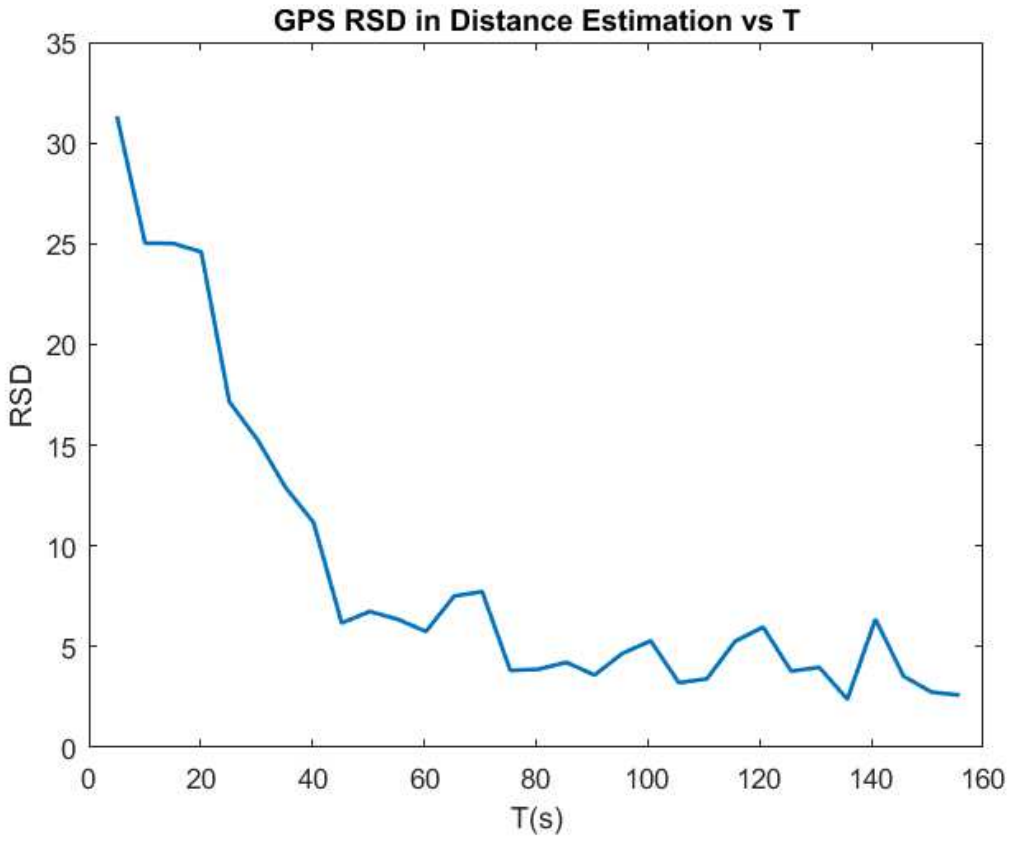

Figure 3.1 : Relative Standard Deviation in Distance Estimation from GPS when the measurements of two GPS similar receivers in the same location are compared. The sampling period changes 5 to 150 seconds

3.1.2 GPS Error Probability Distribution

In this section is characterized the behavior of GPS error. The procedure of the experiment performed to characterize the GPS error is described as follows:

1. Fixed two GPS Receivers (in our case two iPhones6S see Table 4.1) on the two sides of the waist of an user.
2. Let the user walk/run for a time t_R.
3. While the user is walking/running, in both Receivers measure and store the GPS coordinates at a sampling frequency F_s.
4. Define one of the Receivers as the Baseline and the other as the Comparison.

After this procedure the measurements are divided in k_i intervals of size Δt_i, where $i = 1 : N$. The horizontal (X_i) and vertical (Y_i) distances in each interval corresponds to a sample. Then we have four distances, for the Baseline Receiver (X_i^B and Y_i^B) and two for the Comparison Receiver (X_i^C and Y_i^C). Two measures of error are define, the horizontal ($\epsilon_X = X_i^B - X_i^C$) and vertical ($\epsilon_Y = Y_i^B - Y_i^C$) errors.

From the literature for some specific conditions it seems that the error follows a Gaussian Distribution [49]. However, for the specific conditions of this research is necessary to prove this. To prove that the GPS error, ϵ_X and ϵ_Y, follows a Gaussian Distribution the two most powerful normality tests are used, i.e. the Shapiro-Wilk (SW) test and the Anderson-Darling (AD) test [51]. Additionally, also the normal Quantile-Quantile plot (Q-Q plot) were obtained and the results are presented for X and Y in Figures 3.2, respectively. For the first case the null and the alternative hypothesis are:

H_0 : The distribution is normal

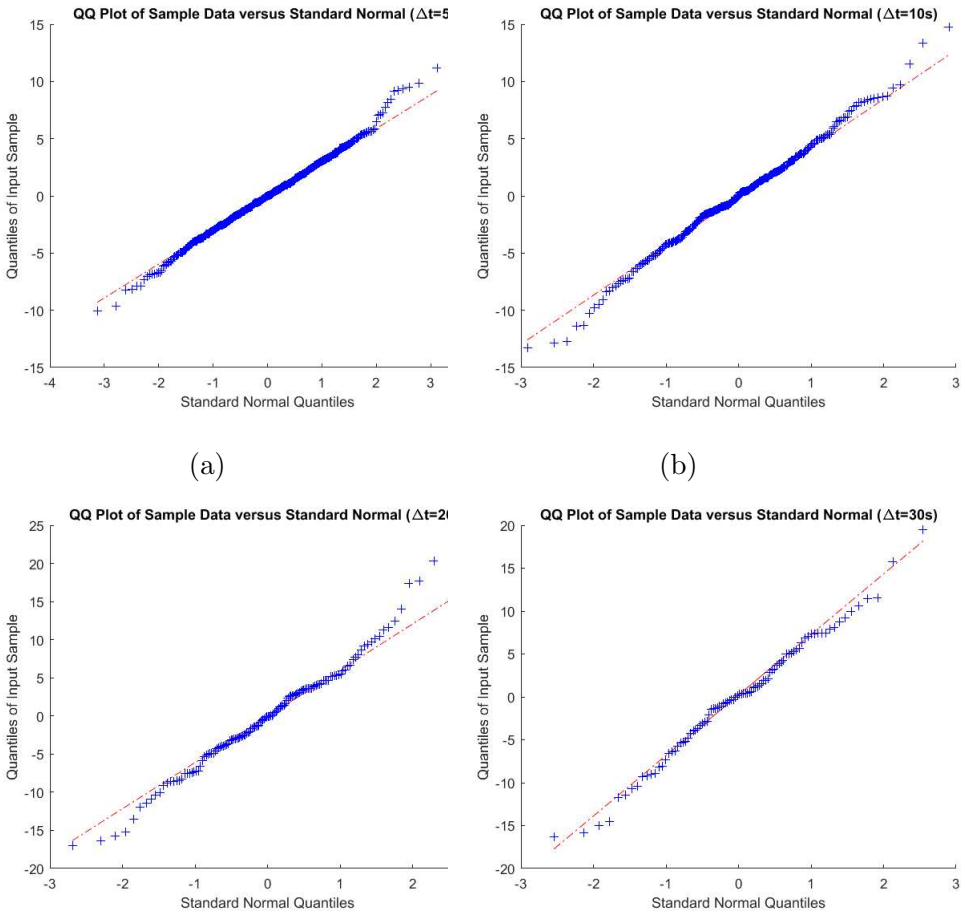

Figure 3.2 : Normality test: Quantile-Quantile plot for horizontal error in GPS measurements for time intervals of (a) 5 (b) 10 (c) 20 and (d) 30 seconds.

H_1 : The distribution is not normal

The results for these two tests are presented in Tables 3.1 and 3.2 for the Horizontal and Vertical error, respectively.

We can conclude from the tables and the plots that the error follows a Gaussian Distribution.

There are multiples approaches to improve GPS measurements in the literature [14–16, 18, 20]. However, in the current application is not intended to improve the

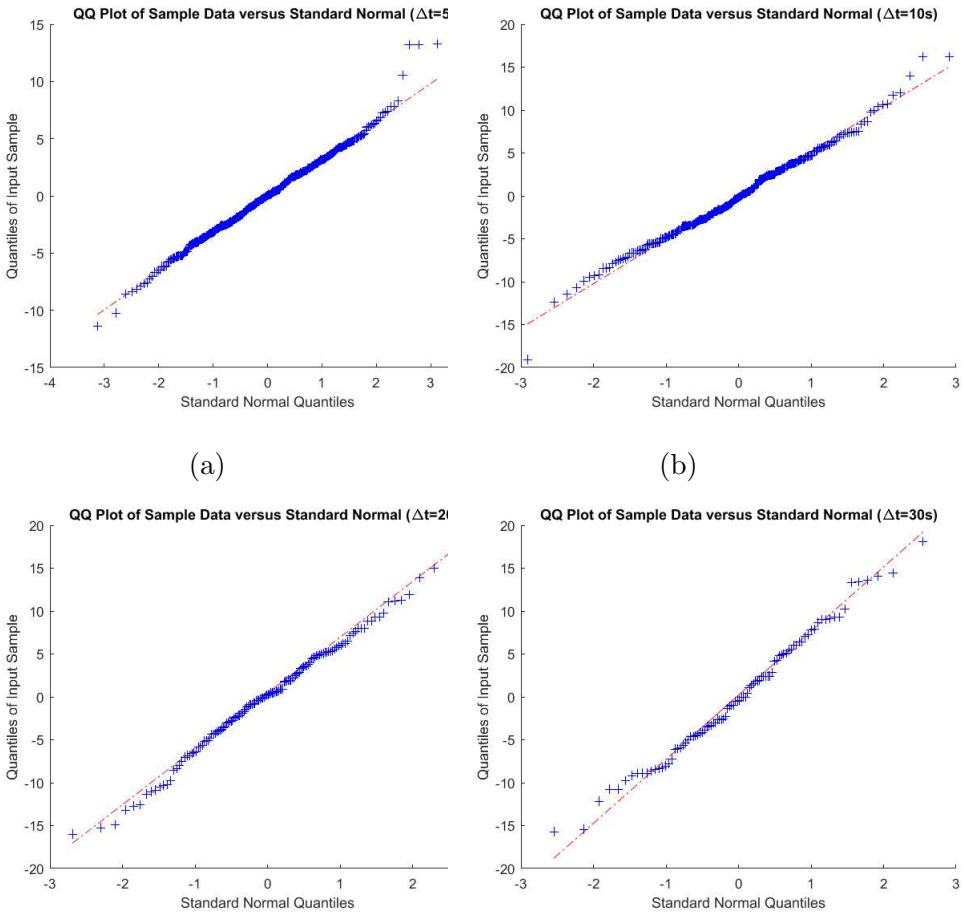

Figure 3.3 : Normality test: Quantile-Quantile plot for vertical error in GPS measurements for time intervals of (a) 5 (b) 10 (c) 20 and (d) 30 seconds

localization based on GPS but the estimated distance or angle between points which is a simpler task. For this purpose next section presents, to the best of my knowledge, a novel method for improving the distances and angle estimation from GPS.

3.1.3 Label Improving

In the current application, the distance between two points is below $100m$ which allow us to use the equirectangular approximation without a significant effect on the

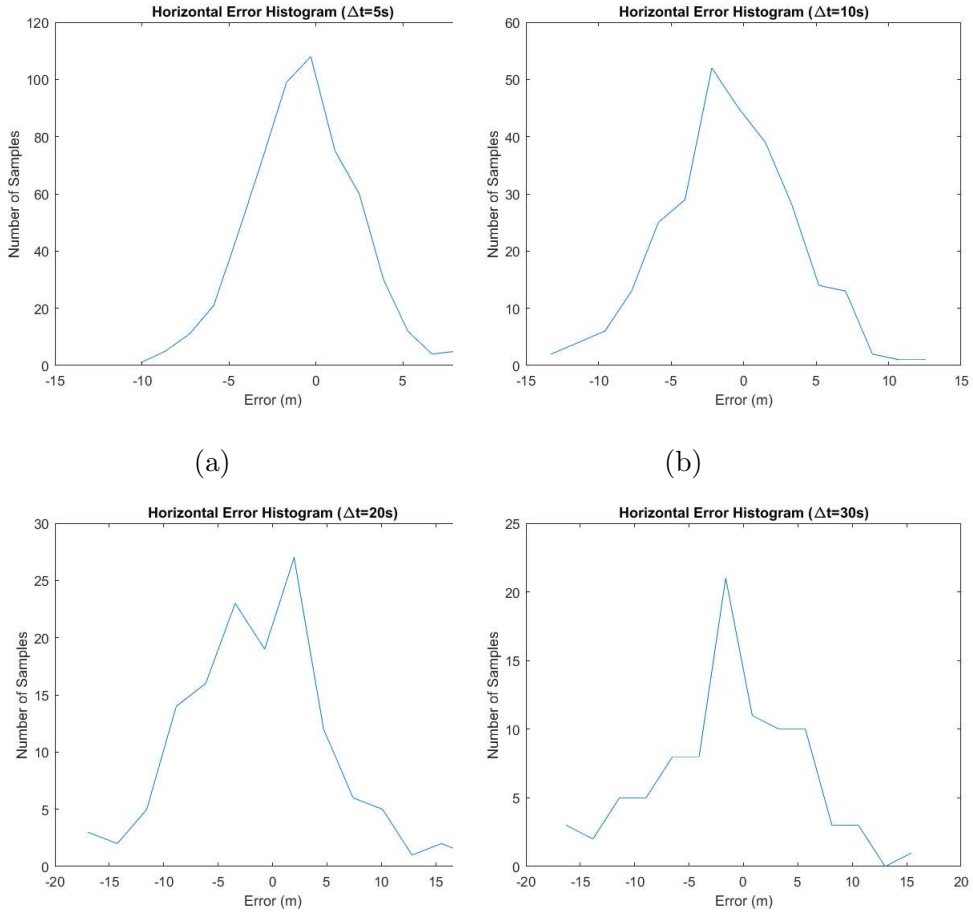

Figure 3.4 : Horizontal Error Histogram from GPS measurements for time intervals of (a) 5 (b) 10 (c) 20 and (d) 30 seconds

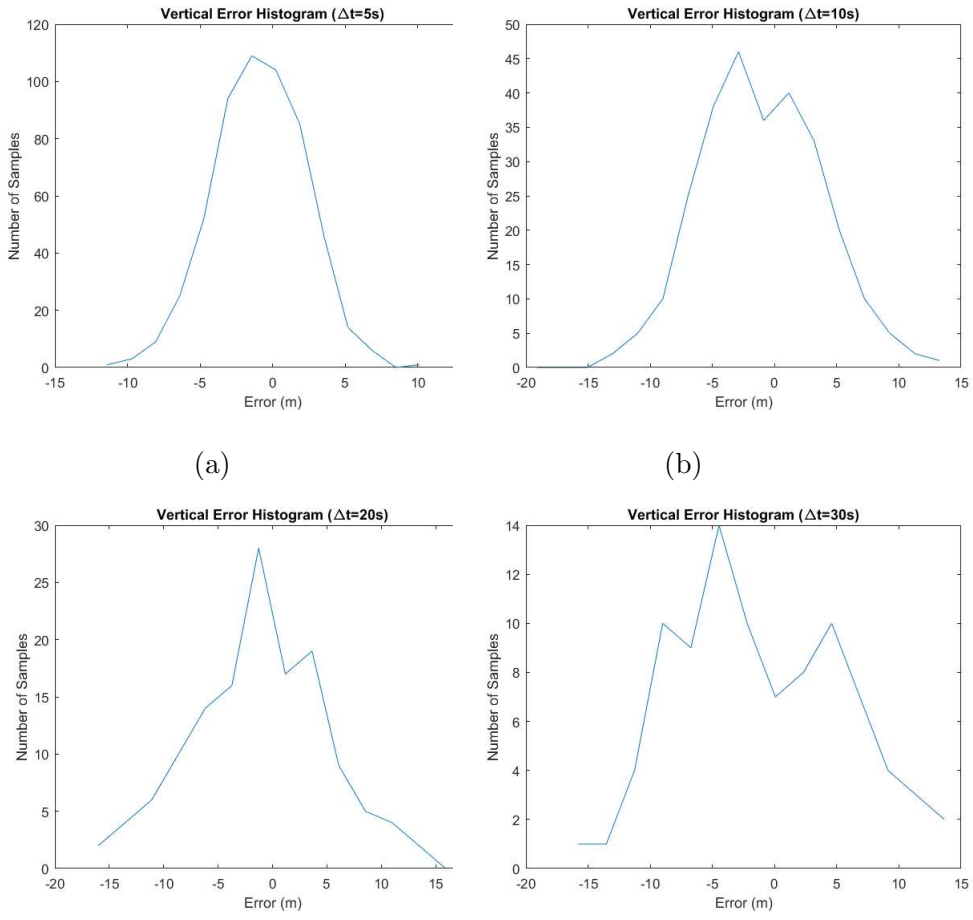

Figure 3.5 : Vertical Error Histogram from GPS measurements for time intervals of (a) 5 (b) 10 (c) 20 and (d) 30 seconds

Table 3.1 : Normality Tests for horizontal error on GPS measurements for time intervals of 5, 10, 20 and 30 seconds. The hypothesis are 0 : The distribution is normal, 1 : The distribution is not normal

$\Delta t(s)$	**Horizontal GPS error**	
	Shapiro-Wilk	**Anderson-Darling**
5	0	0
10	0	0
20	0	0
30	0	0

error [52], which was also confirmed by experiments. Based on this we define two points: i and $i-1$ with the following definitions:

- Horizontal and vertical distances obtain from GPS defined as X_i^{gps} and Y_i^{gps}, respectively.
- GPS coordinates for points i and $i-1$: $(lat_i^{gps}, lon_i^{gps}, alt_i^{gps})$ and $(lat_{i-1}^{gps}, lon_{i-1}^{gps}, alt_{i-1}^{gps})$.
- $\Delta lat_i^{gps} = lat_i^{gps} - lat_{i-1}^{gps}$ and $\Delta lon_i^{gps} = lon_i^{gps} - lon_{i-1}^{gps}$.
- Ground truth coordinates for points i and $i-1$: $(lat_i^R, lon_i^R, alt_i^R)$, $(lat_{i-1}^R, lon_{i-1}^R, alt_{i-1}^R)$
- Changes in latitude and longitude Δs: $\Delta lat_i^R = lat_i^R - lat_{i-1}^R$ and $\Delta lon_i^R = lon_i^R - lon_{i-1}^R$.

Additionally, we define the error in longitude and latitude for GPS estimation as:

$$\epsilon_i^{lon} = lon_i^{gps} - lon_i^R \tag{3.1}$$

Table 3.2 : Normality Tests for vertical error on GPS measurements for time intervals of 5, 10, 20 and 30 seconds. The hypothesis are 0 : The distribution is normal, 1 : The distribution is not normal

$\Delta t(s)$	**Vertical GPS error**	
	Shapiro-Wilk	**Anderson-Darling**
5	0	0
10	0	0
20	0	0
30	0	0

$$\epsilon_i^{lat} = lat_i^{gps} - lat_i^R \tag{3.2}$$

And the change in this error for multiple samples as:

$$\Delta\epsilon_i^{lat} = (\epsilon_i^{lat} - \epsilon_{i-1}^{lat}) \tag{3.3}$$

$$\Delta\epsilon_i^{lon} = (\epsilon_i^{lon} - \epsilon_{i-1}^{lon}) \tag{3.4}$$

Based on these definitions and using the equirectangular approximation [52] we have:

$$X_i^{gps} = C_1 cos(lat_i^{gps})\Delta lon_i^{gps} \tag{3.5}$$

$$Y_i^{gps} = C_1 \Delta lat_i^{gps} \tag{3.6}$$

$$d_i^{gps} = ((X_i^{gps})^2 + (Y_i^{gps})^2)^{\frac{1}{2}} \tag{3.7}$$

Where C_1 is the radius of the Earth. Similarly for the real coordinates we have:

$$X_i^R = C_1 cos(lat_i^R)\Delta lon_i^R \tag{3.8}$$

$$Y_i^R = C_1 \Delta lat_i^R \tag{3.9}$$

$$d_i^R = ((X_i^R)^2 + (Y_i^R)^2)^{\frac{1}{2}} \tag{3.10}$$

Using 3.1 and 3.4 we can rewrite 3.5 as:

$$\begin{aligned} X_i^{gps} = C_1 cos(lat_i^{gps})(lon_i^{gps} - lon_{i-1}^{gps}) = C_1 cos(lat_i^{gps})(lon_i^R + \epsilon_i^{lon} - lon_{i-1}^R - \epsilon_{i-1}^{lon}) \\ = C_1 cos(lat_i^{gps})(\Delta lon_i^R + \Delta\epsilon_i^{lon}) = C_2(\Delta lon_i^R + \Delta\epsilon_i^{lon}) \end{aligned} \tag{3.11}$$

where

$$C_2 = C_1 cos(lat_i^{gps}) \tag{3.12}$$

C2 is not constant in general but for regions where the samples were taken it remains constant with a maximum relative difference of less than 0.012%.

Similarly, for Y_i^{gps} we have:

$$Y_i^{gps} = C_1(lat_i^{gps} - lat_{i-1}^{gps}) = C_1(lat_i^R + \epsilon_i^{lat} - lat_{i-1}^R - \epsilon_{i-1}^{lat}) = C_1(\Delta lat_i^R + \Delta\epsilon_i^{lat}) \tag{3.13}$$

Replacing 3.11 and 3.13 into 3.7 we obtain:

$$\begin{aligned} (d_i^{gps})^2 = ((X_i^{gps})^2 + (Y_i^{gps})^2) = C_2^2(\Delta lon_i^R + \Delta\epsilon_i^{lon})^2 + C_1^2(\Delta lat_i^R + \Delta\epsilon_i^{lat})^2 \\ = (d_i^R)^2 + \delta_i \end{aligned} \tag{3.14}$$

where δ_i is defined as:

$$\begin{aligned} \delta_i = C_2^2(\Delta lon_i^R)^2 - C_1^2 cos^2(lat_i^R)(\Delta lon_i^R)^2 + C_2^2(\Delta\epsilon_i^{lon})^2 + C_1^2(\Delta\epsilon_i^{lat})^2 \\ +2C_1^2\Delta\epsilon_i^{lat}\Delta lat_i^R + 2C_2^2\Delta\epsilon_i^{lon}\Delta lon_i^R \end{aligned} \tag{3.15}$$

If we consider $N+1$ samples of the same X_i^R, Y_i^R distances, then we have N equations as 3.15. It is important to highlight that (Δlon_i^R) and (Δlat_i^R) are the same for all N based on their definition. The expected value of δ is:

$$\begin{aligned} E(\delta) = E((C_2^2 - C_1^2 cos^2(lat_i^R))(\Delta lon_i^R)^2) + E(C_2^2(\Delta\epsilon_i^{lon})^2) \\ +E(C_1^2(\Delta\epsilon_i^{lat})^2) + E(2C_1^2\Delta\epsilon_i^{lat}\Delta lat_i^R) + E(2C_2^2\Delta\epsilon_i^{lon}\Delta lon_i^R) \\ = C_3 E((\Delta lon_i^R)^2) + C_2^2 E((\Delta\epsilon_i^{lon})^2) + C_1^2 E((\Delta\epsilon_i^{lat})^2) \\ +2C_1^2 E(\Delta\epsilon_i^{lat}\Delta lat_i^R) + 2C_2^2 E(\Delta\epsilon_i^{lon}\Delta lon_i^R) \end{aligned} \tag{3.16}$$

Where $C_3 = (C_2^2 - C_1^2 cos^2(lat_i^R))$ is assumed to be constant for the same reason C_2 was explained to be constant. From experiments mentioned in Section 3.1.2 it was recognized that the horizontal and vertical GPS error follows a Gaussian distribution. Therefore, we can obtain the following result in the second term of 3.16:

$$\begin{aligned} E((\Delta\epsilon_i{}^{lon})^2)\Big|_{i=1:N} = E((\epsilon_i^{lon} - \epsilon_{i-1}^{lon})^2)\Big|_{i=1:N} = [E((\epsilon_i^{lon})^2) + E((\epsilon_{i-1}^{lon})^2) \\ -2E(\epsilon_i^{lon} * \epsilon_{i-1}^{lon})]\Big|_{i=1:N} \approx [2E((\epsilon_i^{lon})^2) - 2E(\epsilon_i^{lon} * \epsilon_i^{lon})]\Big|_{i=1:N} \approx 0 \end{aligned} \tag{3.17}$$

Additionally, if we consider the definition that the N samples have the same X_i^R, Y_i^R distances, we can say that Δlon_i^R is the same for all samples. Therefore, we obtain from the fifth term in 3.16 the following:

$$E(\Delta\epsilon_i^{lon}\Delta lon_i^R))\Big|_{i=1:N} = \Delta lon_i^R E(\Delta\epsilon_i^{lon}))\Big|_{i=1:N} \approx 0 \tag{3.18}$$

A similar procedure with the correspondent terms of the latitude (terms 3 and 4) will lead to also a value of approximately 0. Therefore, we can rewrite 3.16 as:

$$E(\delta) = C_3 E((\Delta lon_i^R)^2) \tag{3.19}$$

Obtaining finally:

$$E((d_i^{gps})^2) = E((d_i^R)^2) + C_3 E((\Delta lon_i^R)^2) = (d_i^R)^2 + C_3(\Delta lon_i^R)^2 \tag{3.20}$$

Where it was used the definition that the N samples have the same X_i^R, Y_i^R distances. This lead us to the following conclusion:

- With the previous procedure, the expected value for GPS distance is only different from the real by a constant value $C_3(\Delta lon_i^R)^2 = (C_2^2 - C_1^2 cos^2(lat_i^R))(\Delta lon_i^R)^2$
- For the conditions assumed here the GPS overestimates the actual distance which is consistent with results in Chapter 4.

From previous analysis we can conclude that if samples with GPS labels correspondent to the same real distance are combined, then the expected value will be closer to the real value than each sample individually. Therefore, it's necessary to identify which GPS samples correspond to the same real distance. To identify those samples we can measure the speed or acceleration directly or have a variable which is directly proportional to them, then we grouped together samples of similar acceleration (speed) and combine them as described previously. In next Section is explained the procedure to achieve this.

3.2 Machine Learning Model

To obtain the position of a person is necessary to know their distance and angle traveled from an origin of coordinates. As it was mentioned in section 2.2 accelerometer and gyroscope are widely used for this purpose. In this case they are going to be used separately for angle and distance because if they are combined the number of features considered for each class is higher, which increases the amount of data required to train the model [45] making more difficult a wide adoption by the final user.
In the next lines the Machine Learning Model is going to be explained with the two goals in mind: Improve the GPS labels and Train the model.

3.2.1 Unsupervised Learning

As it was mentioned in Section 3.1.2 the labels correspondent to GPS position estimation has a high RSD for small time intervals which provides not reliable information for the training process [48]. Therefore, in this stage it is supposed that the labels are not available for the samples and the samples are grouped based on their similarity. Considering [37] results, which recommends to be $0 < \ell < 1$, the similarity metric

is chosen as the $\ell - norm$ between the samples with $\ell - norm$ being the one that produces best results in the range $0 < \ell < 10$. Once the groups are formed the labels are combined according to Section 3.1.3. In the Algorithm 1 is described in detail the process of this stage.

Algorithm 1 Grouping of Training Samples

Input: Training samples Xtr_0 and their labels Ytr_0

Output: Samples with more accurate labels

▷ Initialization Step

Take k samples at random from Xtr_0 and define them as $subtrt$

Define the remaining $n - k$ samples in Xtr_0 as Xtr

▷ Near Neighbor

Define the $\ell - norm$ metric for Near Neighbor

for i in 1 to $(n - k)$ **do**

    Find the Near Neighborg of Xtr(i) in $subtrt$

for i in 1 to k **do**

    Define $A = \begin{bmatrix} subtrt(i) \\ neighbors_i \end{bmatrix}$ where $neighbors_i$ are the neighbors of $subtrt(i)$ in Xtr.

    $subtrt(i) = mean(A)$

While it's known that Near Neighbor fails for high dimensions, the data in this research doesn't consider high dimensions which is going to be explained in detail in Sections 3.2.2 and 4.2. Additionally, the $\ell - norm$ was chosen using the results in [37] combined with Cross-Validation.

3.2.2 Feature Extraction

If the samples are used in raw the number of predictors will be higher than if the samples are mapped into a proper space. Additionally, the time the model takes to detect obvious trends will be higher and more important trends may be not discovered. Furthermore, the model is more probable to fit noise instead of useful patterns. [43, 53, 54] Based on this some features in the time domain and frequency domain where explored and chosen based on Cross-Validation. The characteristics of the dataset and the details of the procedure are given in Section 4.2.

3.2.3 Number of Samples

An heuristic to choose the number of training samples is to have at least $10p$ samples per class, where p is the number of features [45, 55, 56]. If angle and distance are not estimated by separate models the number of samples would increment significantly. For example for a model with four features in the accelerometer and two from the gyroscope the number of samples needed is 60 samples per class. Which is not significant if only two possible locations are consider. Though, this is not usually the case. If not enough samples are considered the risk is that the model will tend to overfit the data making it not useful for prediction [56].

3.2.4 Supervised Learning

In the supervised learning the Machine Learning (ML) approach is classification for the angle estimation and classification and regression for the distance estimation. The methods used are divided in two:

- ML model for the angle estimation: Support Vector Machine, Nearest Neighbor, Random Forest.

- ML model for the distance estimation: Random Forest Regression, Nearest Neighbors Regression, Ridge Regression and numerical integration.

These methods were described in Section 2.3.

3.2.5 Time Interval

The size of the time interval is recognized to affect the performance of the Models using IMU sensors for position estimation [31]. Therefore, Cross-Validation was used to estimate the best size of the time window for distance and angle model.

Chapter 4

Results

In this Chapter is presented the data collected and the results of the Methods proposed in Chapter 3.

4.1 Data Collection

As it was mentioned in Chapter 3 Machine Learning requires Ground Truth data to train and test the model. In this Section is described the data obtained from the two possible approaches mentioned in Section 3.1.1, i.e. *Lab Maps* and *GNSS maps*.

4.1.1 GNSS Labeled Data

The GNSS receivers used were Smartphones. Two types of smartphones were used, the iPhone 6S from Apple and the BLU Advance 5.0 from BLU with the Technical Specifications detailed in Table 4.1. These devices were fixed in four different locations over the user: wrist, waist, pocket and arm. These locations were chosen based on usual locations where users have their Smartdevice [42] and to identify the dependence of the model with the location of the Smartphone. From each measurement ten variables were recorded: 3-axis accelerometer, 3-axis gyroscope, 3-GPS coordinates and the timestamp.

Two main approaches were considered in this case:

Single device per experiment

For each of the Smartphones and for each of the different locations on the body of the user, multiple experiments were performed. It was identified in these experiments that the GPS labels obtained when the user has the device on the waist are more accurate. This is because in other parts of the body the GPS receiver is more occluded by the body which is consistent with what is found in the literature [18,50] Therefore, new experiments were performed as it is described below.

Two devices or more per experiment

Taking into account that the more accurate results were obtained with a Smartphone fixed at the waist new experiments were performed with two devices, one fixed at the waist and the other fixed at one of the other three possible body locations. With these experiments it was recognized that the GPS labels obtained from the receiver on the waist can be used to train the model for the wrist, arm or pocket cases. These models will be more accurate for training and testing the model than if the GPS labels obtained from those body locations are used. Then, for the focus of this research the GPS labels obtained from the waist will be used in the cases where GPS labels are mentioned.

4.1.2 Lab Maps

As it was described in Section 3.1.1 *Lab maps* can be used to have a highly accurate Ground Truth data. Two different fields were chosen: the Academic Quad and the Twilight Epiphany Skyspace Square from Rice University as is shown in Figures 4.2a and 4.2b, respectively. For each of them a Random Walk was defined [31] and the data was taken with iPhone6s on four different users. The dataset collected has thirteen

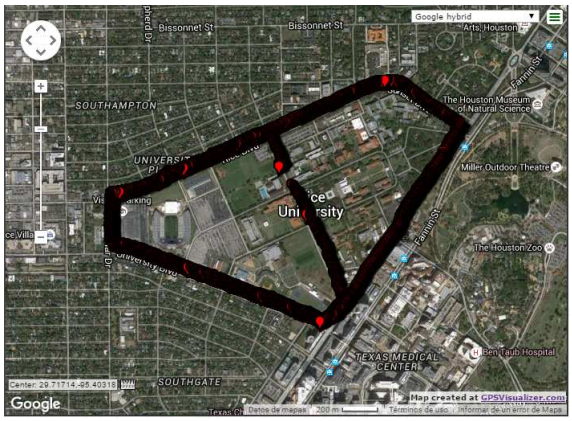

Figure 4.1 : Outer Loop of Rice University Campus where GNSS Labeled Data was taken

components, 3-axis gyroscope, 3-axis magnetometer, 3-axis accelerometer, location based on GPS coordinates (latitude, longitude and altitude) and the timestamps. Additionally, every time a intersection is encountered the time is recorded. The sampling frequency is approximately 51Hz. Data was collected by four users who walked for more than two hours. The types of turns included left, right, U and no-turns. For this case, we do not rely on GPS-based location labels. The ground truth was the "actual" distance of the grid provided by matching the actual map with the timestamps of the check points. The dataset contains 100 turns over 45 minutes for first, second and third day.

4.2 Feature Extraction

The data from the lab maps (See Section 4.1.2) was used to tune some parameters of the model such as the features which provides a better estimation. The algorithm to perform this parameter selection was based on Cross-Validation an it's explained in Algorithm 2 where $f_1, ...f_p$ are different features and their correspondent combina-

Table 4.1 : Technical Specifications of the devices used in this research

Feature	**Device**	
	iPhone6S	**BLU Advance 5.0**
Memory	32 GB — 2 GB RAM	4GB — 768MB RAM
Processor	Apple A9 Dual-core 1.84 GHz Twister	1.3 GHz Quadcore
Operating System	iOS 9	Android 5.1 Lollipop
Battery	Li-Ion 1715 mAh	Li-Ion 1800 mAh
Sensors	Accelerometer, gyro, proximity, compass, barometer	Accelerometer, gyro, proximity, compass

tions.

For the accelerometer and gyroscope measurements this procedure was followed and the features which provides the best result were chosen. The features that were explored on frequency domain were Spectral Centroid, Spectral Spread, Spectral Skewness, Spectral Kurtosis, Spectral Slope, Spectral Band Ratio and in time domain were Mean, Standard Deviation, Maximum, Minimum and Zero Crossing Rate (ZCR). Among these the Spectral Energy Distribution, proposed in this work, gave good prediction results for distance estimation based on accelerometer and it's explained in Section 4.2.1. It's important to highlight that the models were designed in a way that the number of features is increased only if adding more features improves the accuracy prediction in at least 1%. This is because, as it was explained in Section 3.2.3, increasing the number of features not only increases the complexity of the model but also the size of the training data in order to have reliable results.

Algorithm 2 Feature selection

Input: Training samples (Xtr_0, Ytr_0)

Output: Best features for training the model

▷ Initialization Step

Divide the samples in Xtr_0 into $k - folds$

for f_j in f_1 to f_p **do**

    Map samples to feature space f_j

    for i in 1 to k **do**

        Define fold i as testing and the other $k - 1$ folds as training

        Train the model

        Find the validation error and store it in $error_{v0}(i, j)$

    Find the average error as $error_v(j) = mean(error_{v0}(:, j))$

Choose from $f_1, ...f_p$ the f_b which provides the lowest error.

(a) Rice's Academic Quad

(b) Rice's Twilight Epiphany Skyspace Square

Figure 4.2 : Fields were the data was taken considering random walks of 100 turns.

Besides obtaining the type of features to extract from each sample also some parameters of the model are tuned in this stage, such as the size of the time window and the number of bands to divide the frequency spectrum.

4.2.1 Features from Accelerometer

To obtain the Spectral Energy Distribution the algorithm is described in Algorithm 3. In this Algorithm is mentioned a bandpass filter with cutoff frequencies $0.1Hz$ and $15Hz$ which is used based on the results of [57] where it's mentioned that 99% of the frequency components of gait are below $15Hz$. Therefore, filtering frequencies above

that value will reduce the noise. Additionally, the DC components are also filter out with the cutoff frequency of $0.1Hz$.

Algorithm 3 Spectral Energy Distribution Transformation

1: **Input:** Samples X
2: **Output:** The samples mapped to feature space
3: ▷ Initialization Step
4: Take the Fast Fourier Transform of X and define it as X_f
5: Applied to X_f a bandpass filter with cut-off frequencies as f_l and f_u.
6: Define m bands of frequencies between f_l and f_u.
7: Find the energy in each band
8: Normalize the energy in each band with respect to the maximum energy

The best results were obtained with the features of Spectral Energy Distribution combined with Mean and the Standard Deviation in the time domain. In Table 4.2 the results are provided for four different users in the range of ages $[22, 32]$ for time windows of size $15s$, heights in the range $[1.55m, 1.85m]$.

4.2.2 Features from Gyroscope

For the case of the gyroscope the features which gave the best results were the Zero Crossing Rate combined with the Energy in each time interval. In Table 4.3 the results for different features can be seen.

4.3 Localization with Lab Maps

In order to estimate the position from IMU measurements the problem was divided in two steps: 1. estimate the distance travel by the user and 2. estimate the turn

Table 4.2 : Feature Selection Results for Accelerometer. Error is given in meters for each Feature-User combination. *Spectral represents the best result among Spectral Centroid, Spectral Spread, Spectral Skewness, Spectral Kurtosis, Spectral Slope. **Best combination refers to the best result obtained from combining different features

Feature	**User**			
	1	**2**	**3**	**4**
Energy	6.234m	8.862m	8.879m	6.843m
Mean	6.316m	8.909m	8.192m	6.800m
σ	6.082m	8.921m	6.354m	8.906m
Min/Max	6.163m	7.469m	6.512m	6.216m
ZCR	6.087m	6.723m	6.035m	5.987m
Spectral*	5.691m	5.848m	6.257m	5.862m
Spectral Energy Distribution	5.117m	5.605m	5.323m	5.190m
Best Combination**	5.118m	5.624m	5.125m	5.123m

given by the user in every intersection. In next paragraphs is explained the model obtained for these cases.

4.3.1 Distance Estimation

As it was shown previously the frequency domain provides the best results for distance estimation. To confirm the usefulness of frequency domain for distance estimation, different Machine Learning Methods were tried and the results are presented in Table 4.4. Also in this table can be seen the results obtained with Numerical Integration. These results corresponds to training the distance on the GPS data and testing on

Table 4.3 : Feature Selection Results for Gyroscope. Error is given in percentage of correct turn estimation for each Feature-User combination. *Spectral represents the best result among Spectral Centroid, Spectral Spread, Spectral Skewness, Spectral Kurtosis, Spectral Slope. **Best combination refers to the best result obtained from combining the two best features results

Feature	**User**			
	1	**2**	**3**	**4**
Energy	98.0%	92.6%	90.6%	89.7 %
Mean	98.0%	91.6%	89.6%	88.2%
σ	70.2%	68.2%	69.2%	68.7%
Min/Max	77.1%	69.2%	66.1%	68.2%
ZCR	75.1%	69.7%	66.7%	68.2%
Spectral*	71.7%	70.7%	72.7%	68.2%
Spectral Energy Distribution	61.2%	58.2%	59.2%	58.2%
Best Combination**	98.2%	93.7%	91.6%	91.2%

Lab Maps data.

4.3.2 Turn Estimation

As it was mentioned, from Table 4.3 it can be seen that time domain provides the best results for turn estimation. To confirm this the accuracy of the models in angle estimation is compare in Table 4.5 with the step of feature extraction.

It can be seen that for Gyroscope the features based on the time domain provide a better prediction than the frequency domain. The possible explanation for this result is that the short duration of turns doesn't provide significant patterns in the

Table 4.4 : Distance Estimation results for Lab Maps labels training on GPS labels.

Method	Percentage Error Rate (Error in Meters)	
	Frequency Domain	Time Domain
Nearest Neighbors	26.34% (8.18m)	67.68%(20.77m)
Ridge Regression	27.08% (8.39m)	71.23%(22.01m)
Random Forest Regression	23.5% (7.2m)	66.32%(20.46m)
Numerical Integration	NA	70% (22m)

Table 4.5 : The results for Turning detection and Turning classification using Lab Maps labels

Method	Turn Detection Accuracy		Turn Classification Accuracy	
	Frequency	Time	Frequency	Time
NN	52%	54%	33%	31%
RF	67%	83%	65%	81%
SVM	72%	**95%**	69%	**90%**

frequency spectrum while in time domain their differences are more significant.

4.3.3 Position Estimation

The error in distance described in Table 4.4 is on average and if the error cumulative properties of Dead Reckoning are considered, then the error is going to increase significantly until the prediction becomes useless. If no-correction is performed the average error in location estimation is 28.72m with a maximum error of 99m. Therefore, it's necessary to apply a correction to the prediction as it was described in Section 2.1.5. In this case was chosen the map information because it doesn't depend on other devices and it's easily integrated in the model [10–12].

The map correction was integrated in the model as it's described in Algorithm 4.

This algorithm was tested on 200 paths in Figure 4.2a and 200 paths in Figure 4.2b. The paths range from 16 to 78 meters.

Result. For the random paths, all the final edges were identified successfully. The high accuracy obtained in this is considered to be based on the almost perfect turn estimation obtained in 4.5. For the location estimation the average error is $4.3 meters$ with the best error being $0.05 meter$ and the worst case error being $11.5 meters$.

4.3.4 Energy Consumption and Accuracy of GPS

As it was described in Section 2.1.5 error correction can be performed with other technologies, such as GPS. Being GPS widely used technology and the standard for outdoors, the current results it's compared with it in terms of energy consumption and error. If 15 seconds time windows are considered our algorithm consumes only $0.44J$ while GPS requires $11.85J$. Therefore, the energy consumption is reduced by a factor of $27x$. The energy consumed by the sensors is $0.271311J$ and the rest is

Algorithm 4 Map integration in the location estimation

1: **Input:** Trained Model, testing samples $X_t s$, map information

2: **Output:** Location of the user

3:

4: Set the initial position of the person. This can be provided by the user or obtained from other source such as GPS.

5: Keep a running overlapping window of Δt time. Use the trained turn estimation model (See Table 4.5) and identify the turns in $X_t s$.

6: Estimate the distance between turns using Random Forest (See Table 4.4). Accumulate the distance in 15seconds non-overlapping windows.

7: Use the map information to correct the turns estimation. Based on the distance estimated, determine if there is a turn in a range defined by the expected distance error. If there is no intersection in the vicinity, then that turn is skipped (turn estimate correction). If a turn is found, then the distance is corrected in a way that matches the distances of the map (distance estimate correction)

correspondent to the proposed method. In terms of accuracy the GPS error for time intervals of 15seconds is $8.7m$ while for our method is $4.3m$, as it was mentioned before. [58]

4.4 Localization with GNSS

While 4.3 meters could be good enough for outdoors, in indoors one room from another can have a difference smaller than this value which makes it not useful in some scenarios. Additionally, the requirement of knowing the map of the location beforehand which is not always true for indoor navigation. In this section this problematic is addressed.

If we consider the position estimation from GNSS perspective we can see that one of its more significant advantages is that its estimation does not depend on the error of previous estimations. Contrary, Dead Reckoning methods suffer from this drawback. If GNSS is considered from the Dead Reckoning perspective is similar to having a method which takes its previous estimation and predicts a new estimation which is corrected by a specific factor and guarantees that the error is always below μ_{error} with high probability. This behavior is desired to be obtained in Dead Reckoning where the error increases with time without a bound [3]. In this section is not searched to obtain this bound but to decrease the rate at which the error is accumulated.

4.4.1 Distance Estimation

In Section 3.1.2 was described the error behavior on distance estimation based on GPS measurements. In Section 3.1.3 is proposed a method to improve the GPS labels. In this Section is explored how that method improves the performance of the estimation of the IMU method for distance estimation using GPS labels and testing on Lab Map

labels.

The algorithm to define the groups and the $\ell - norm$ which produces best results is defined in Algorithm 5. With this process was identified to be $\ell = 0.4$ which produces the best grouping. One the trained model is obtained is tested on the labels obtained from the Lab Maps. This model was trained and tested on user 1, but for comparison purposes was also tested on the Lab Maps data obtained from users 2, 3 and 4. In Table 4.6 it can be seen the results for training in user 1 the model and testing in any of the four different users.

Algorithm 5 Algorithm for implementing GPS distance labels improvement

1: **Input:** Samples correspondent to Accelerometer measurements ($X_t r0$) and GPS Labels correspondent to angle and distance)($Y_t r0$)
2: **Output:** Model with more accurate GPS distance labels
3: ▷ Initialization Step
4: Define $\ell - norm$ and $\#groups$ as the parameters to tune in the model.
5: Define the range for $\ell - norm$ and for $\#groups$ to iterate.
6: ▷ Tunning Parameters of the Model
7: Define the error between the position estimated from GPS and the one estimated from the model as the metric to choose the best values for the parameters.
8: Apply Cross-Validation to choose the best values for $\ell - norm$ and for $\#groups$ according to the defined metric in the previous step.
9: Find the label in each group as it was defined in Expression 3.20
10: Return the model

This result was not expected because in the literature is commonly found that the models are trained in a user and test in the same user [3, 42, 46, 47]. However, in this

Table 4.6 : Distance Estimation results training on GPS data of User 1 and testing on the other 3 users. The error is given in percentage and meters. It's compared the performance of the method proposed in Section 3.1.3 to improve the accuracy of GPS labels. The method is referred here as LC which stands for Label Correction

Method	Percentage Error (Error in Meters)	
	Without LC	With LC
User 1	26.974% (8.362m)	19.731%(6.117m)
User 2	30.045% (9.314m)	17.965%(5.569m)
User 3	22.231% (6.892m)	17.437%(5.406m)
User 4	25.577% (7.929m)	21.311% (6.607m)

case training in a different user and testing in another still provides decent results for distance estimation. It means that the data from different users can be combined in a clever way to obtain a general model and reduce the training size require from each new user. It's suspected that this result is because the frequency spectrum is sparse and the model is based in walking patterns that are common to different users.

4.4.2 Angle Estimation from GPS

The angle can be obtained from GPS measurements based on the equirectangular approximation mentioned in Section 3.1.3. If a model based on Gyroscope measurements is trained on this GPS labels the main difficulty is the simultaneity of the events. While Gyroscope will reflect the turn event almost immediately, the GPS receiver will reflect the turn with a delay which depends on many factors of the environment or the GPS receiver. This delay can be understood based on Figure 3.1 where the GPS labels are more accurate if they are sampled with low sampling fre-

quencies. Which means that GPS provides an accurate location in the long term but inaccurate over short terms. However, the turn events are of a short duration and when the GPS does the correspondent correction the angle will be associated probably to a no-turn event which makes difficult to combine the labels in a correct way.

An approach similar to the one in Expression 3.20 was followed but the results were worst for combining the labels that without combining them. It's considered that an additional preprocessing can be done to improve GPS labels for angle estimation. One option is defining a group of Gyroscope samples which are associated to a GPS turn and then give a higher probability based on specific features. This will be explored in detail in future work.

The accuracy of angle estimation described in Section 4.3.2 is no longer valid for angles obtained from GPS measurements because in this case there is a wide spectrum of values which is the result of inaccuracies in estimation from GPS measurements. Then, it's necessary to define another metric of comparison. For this purpose the angle estimated by the model is compared to the one obtained from GPS measurements in every time interval. For this case on average the error is around 1.29°. However, this value should be considered with caution because is the result of intervals of over-estimation and others of under-estimation, which means that there is compensation between time intervals. Then, this value is only meaningful in the position estimation based on combining the angle and distance models.

4.4.3 Location Estimation from GPS

The Location Estimation model is obtained by combining the distance estimation model with the angle estimation model. The location in this case is defined as the

Table 4.7 : Average Location Estimation error in distance and angle without correction using GPS for training the distance and angle models.

User	Location Error	
	Average Distance error (m)	Average Angle error (°)
User 1	7.763m	2.137°
User 2	9.326m	1.042°
User 3	8.234m	1.273°
User 4	9.976m	2.609°

position of the user in a $X-Y$ plane. The error is defined as the average error between the estimated and the real location of the user from the perspective of the final point of the user. In Table 4.7 the results for the different users is provided. The error is not good enough for indoor applications. However, it's necessary to consider than in this case was not applied any kind in correction. Additionally, the error can be reduced if the angle labels are improved in the training process but as it was mentioned this is part of future work.

Chapter 5

Industrial Applications

Location Based Services (LBS) have multiple applications in fields such as Marketing, Health Care, Pet Care, Tourism, Education, among others [59–61]. The current research is important for those areas because the energy consumption, as it was shown in Section 4.3.4, is much lower than GPS and also because is not prohibitive for indoors. Particularly, the low energy consumption enable not only the implementation in common Smart Devices (Smartphones, Smartwatches, WristBands, among others) but also to design low-cost, low-size devices to identify the position of the user [31]. Below it's explained how some of these fields could be impacted by the research of this thesis.

Marketing

Knowing the location of an user in indoors and their correspondent interests, ads can be provided of nearby stores which matches user interests [59, 60]. In this case the location privacy can be maintained because the location estimation is done in the Smart Device instead of sending the information to the cloud. [28]

Health care

In this field some patients need a special care. They need to avoid outdoors or some indoors locations in a hospital. Making important that a care specialist receives alarms if a patient is in a unsafe location or is lost. In this way emergency services

can be provided in a timely manner. [62]

Entertainment

Recommendation systems based on the location of the user and their interests can be used to make the experience of a tourist more enjoyable. [63] For example, the visits to museums can be tailored to the user interests or providing additional information based on their location creating a more interactive experience. [64]

Chapter 6

Conclusions

Location Estimation based on IMU sensors is a popular approach which faces two main difficulties in the literature: 1. Error accumulation with each estimation 2. Accurate Ground Truth Labels to train and test the model.

The error accumulation was corrected using information from the map and results for time intervals of $15s$ show that error is on average around $4.3m$ with an energy consumption of $0.44J$ while GPS error is $8.7m$ and energy consumption is $11.85J$.

For the Ground Truth labels it was identified that GPS can be used to train the model if a correspondent correction method is applied on the labels. A method is proposed for this purpose which is based on the error profile of GPS receivers available in iPhone 6S. An error reduction from $9.314m$ to $5.569m$ was achieved for distance estimation when GPS labels are used as the Ground Truth for training. However, for angle estimation it's necessary to combine the proposed method with other approach because the turn event is not simultaneously reflected on GPS and Gyroscope measurements.

Also experiments were performed to identify the best features for distance and angle estimation from IMU. It was recognized that the frequency domain it's more useful for the distance estimation whereas the time domain provides better results for angle estimation. On the experiments the age range of the participants was $[22-32]$ years. Other experiments are important to verify that the features that were here identified to work best for a specific population are valid for other age ranges. However, in case

that the specific features are not valid, the training process for those cases will find the correspondent features that provide the best results for the user.

Bibliography

[1] P. Harrop and R. Das, *Mobile Phone Indoor Positioning Systems (IPS) and Real Time Locating Systems (RTLS) 2014-2024: Forecasts, players, opportunities.* Cambridge, U.K.: IDTechEx, 2014.

[2] N. Fallah, I. Apostolopoulos, K. Bekris, and E. Folmer, "Indoor human navigation systems: A survey," *Interacting with Computers*, vol. 25, no. 1, pp. 21–33, 2013.

[3] R. Harle, "A survey of indoor inertial positioning systems for pedestrians," *IEEE Communications Surveys Tutorials*, vol. 15, pp. 1281–1293, Third 2013.

[4] D. Dardari, P. Closas, and P. M. Djurić, "Indoor tracking: Theory, methods, and technologies," *IEEE Transactions on Vehicular Technology*, vol. 64, pp. 1263–1278, April 2015.

[5] J. Skaf and S. Boyd, "Analysis and synthesis of state-feedback controllers with timing jitter," *IEEE Transactions on Automatic Control*, vol. 54, pp. 652–657, March 2009.

[6] K. Palem and A. Lingamneni, "Ten years of building broken chips: The physics and engineering of inexact computing," *ACM Trans. Embed. Comput. Syst.*, vol. 12, pp. 87:1–87:23, May 2013.

[7] K. V. Palem, "Inexactness and a future of computing," *Philosophical Transac-*

tions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences, vol. 372, no. 2018, 2014.

[8] K. V. Palem, *Computational Proof as Experiment: Probabilistic Algorithms from a Thermodynamic Perspective*, pp. 524–547. Springer Berlin Heidelberg, 2003.

[9] K. V. Palem, "Energy aware computing through probabilistic switching: A study of limits," *IEEE Transactions on Computers*, vol. 54, pp. 1123–1137, 2005.

[10] K. Abdulrahim, C. Hide, T. Moore, and C. Hill, "Using constraints for shoe mounted indoor pedestrian navigation," *Journal of Navigation*, vol. 65, no. 1, p. 15–28, 2012.

[11] K. Nakamura, Y. Aono, and Y. Tadokoro, "A walking navigation system for the blind," *Systems and Computers in Japan*, vol. 28, no. 13, pp. 36–45.

[12] S. Koide and M. Kato, "3-d human navigation system considering various transition preferences," in *2005 IEEE International Conference on Systems, Man and Cybernetics*, vol. 1, pp. 859–864 Vol. 1, Oct 2005.

[13] J. L. Awange and J. B. Kyalo Kiema, *Modernization of GNSS*, pp. 47–54. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013.

[14] S. Miura, L. T. Hsu, F. Chen, and S. Kamijo, "Gps error correction with pseudorange evaluation using three-dimensional maps," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, pp. 3104–3115, Dec 2015.

[15] S. Hwang and D. Yu, "Gps localization improvement of smartphones using built-in sensors," 2012.

[16] D. Yoon, C. Kee, J. Seo, and B. Park, "Position accuracy improvement by implementing the dgnss-cp algorithm in smartphones," *Sensors*, vol. 16, no. 6, 2016.

[17] L. Zhao, S. Ye, and J. Song, "Handling the satellite inter-frequency biases in triple-frequency observations," *Advances in Space Research*, vol. 59, no. 8, pp. 2048 – 2057, 2017.

[18] M. Bierlaire, J. Chen, and J. Newman, "A probabilistic map matching method for smartphone gps data," *Transportation Research Part C: Emerging Technologies*, vol. 26, pp. 78 – 98, 2013.

[19] R. K. Balan, Y. Lee, T. K. Wee, and A. Misra, "The challenge of continuous mobile context sensing," in *2014 Sixth International Conference on Communication Systems and Networks (COMSNETS)*, pp. 1–8, Jan 2014.

[20] 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*, MobiSys '10, (New York, NY, USA), pp. 299–314, ACM, 2010.

[21] F. Ben Abdesslem, A. Phillips, and T. Henderson, *Less is more: energy-efficient mobile sensing with SenseLess*, pp. 61–62. United States: ACM, 8 2009. Workshop held as part of ACM SIGCOMM 2009.

[22] M. Arikawa, S. Konomi, and K. Ohnishi, "Navitime: Supporting pedestrian navigation in the real world," *IEEE Pervasive Computing*, vol. 6, pp. 21–29, July 2007.

[23] M. T. Günther Retscher, "Navio-a navigation and guidance service for pedestrians," *J. GPS*, vol. 3, no. 1, pp. 208–217, 2004.

[24] T. Amemiya, J. Yamashita, K. Hirota, and M. Hirose, "Virtual leading blocks for the deaf-blind: a real-time way-finder by verbal-nonverbal hybrid interface and high-density rfid tag space," in *IEEE Virtual Reality 2004*, pp. 165–287, March 2004.

[25] N. B. Priyantha, A. Chakraborty, and H. Balakrishnan, "The Cricket Location-Support System," in *6th ACM MOBICOM*, (Boston, MA), August 2000.

[26] J. Baus, A. Krüger, and W. Wahlster, "A resource-adaptive mobile navigation system," in *Proceedings of the 7th International Conference on Intelligent User Interfaces*, IUI '02, (New York, NY, USA), pp. 15–22, ACM, 2002.

[27] S. S. Chawathe, "Low-latency indoor localization using bluetooth beacons," in *2009 12th International IEEE Conference on Intelligent Transportation Systems*, pp. 1–7, Oct 2009.

[28] Y. Moon, S. Noh, D. Park, C. Luo, A. Shrivastava, S. Hong, and K. Palem, "Capsule: A camera-based positioning system using learning," in *2016 29th IEEE International System-on-Chip Conference (SOCC)*, pp. 235–240, Sept 2016.

[29] H. Xie, T. Gu, X. Tao, H. Ye, and J. Lv, "Maloc: A practical magnetic fingerprinting approach to indoor localization using smartphones," in *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, UbiComp '14, (New York, NY, USA), pp. 243–253, ACM, 2014.

[30] J. M. Krisp, *Progress in Location-Based Services*. Springer Publishing Company, Incorporated, 2013.

[31] O. J. Woodman, C. O. J. Woodman, and O. J. Woodman, "An introduction to inertial navigation," 2007.

[32] S. A. Hoseini-Tabatabaei, A. Gluhak, and R. Tafazolli, "A survey on smartphone-based systems for opportunistic user context recognition," *ACM Comput. Surv.*, vol. 45, pp. 27:1–27:51, July 2013.

[33] K. Katevas, H. Haddadi, and L. Tokarchuk, "Sensingkit: Evaluating the sensor power consumption in ios devices," in *2016 12th International Conference on Intelligent Environments (IE)*, pp. 222–225, Sept 2016.

[34] O. Woodman and R. Harle, "Pedestrian localisation for indoor environments," in *Proceedings of the 10th International Conference on Ubiquitous Computing*, UbiComp '08, (New York, NY, USA), pp. 114–123, ACM, 2008.

[35] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning*. Springer Series in Statistics, New York, NY, USA: Springer New York Inc., 2001.

[36] N. S. Altman, "An introduction to kernel and nearest-neighbor nonparametric regression," *The American Statistician*, vol. 46, no. 3, pp. 175–185, 1992.

[37] C. C. Aggarwal, A. Hinneburg, and D. A. Keim, "On the surprising behavior of distance metrics in high dimensional space," in *Database Theory — ICDT 2001* (J. Van den Bussche and V. Vianu, eds.), (Berlin, Heidelberg), pp. 420–434, Springer Berlin Heidelberg, 2001.

[38] A. E. Hoerl and R. W. Kennard, "Ridge regression: Biased estimation for nonorthogonal problems," *Technometrics*, vol. 12, no. 1, pp. 55–67, 1970.

[39] M. Xu, P. Watanachaturaporn, P. K. Varshney, and M. K. Arora, "Decision tree regression for soft classification of remote sensing data," *Remote Sensing of Environment*, vol. 97, no. 3, pp. 322 – 336, 2005.

[40] S. Tong and D. Koller, "Support vector machine active learning with applications to text classification," *J. Mach. Learn. Res.*, vol. 2, pp. 45–66, Mar. 2002.

[41] J. Shawe-taylor and N. Cristianini, "Further results on the margin distribution," in *In Proc. 12th Annu. Conf. on Comput. Learning Theory*, pp. 278–285, ACM Press, 1999.

[42] V. Renaudin, M. Susi, and G. Lachapelle, "Step length estimation using handheld inertial sensors," *Sensors*, vol. 12, no. 7, pp. 8507–8525, 2012.

[43] I. H. Witten and E. Frank, *Data Mining: Practical Machine Learning Tools and Techniques, Second Edition (Morgan Kaufmann Series in Data Management Systems)*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2005.

[44] G. M. Foody and A. Mathur, "Toward intelligent training of supervised image classifications: directing training data acquisition for svm classification," *Remote Sensing of Environment*, vol. 93, no. 1, pp. 107 – 117, 2004.

[45] G. M. Foody, A. Mathur, C. Sanchez-Hernandez, and D. S. Boyd, "Training set size requirements for the classification of a specific class," *Remote Sensing of Environment*, vol. 104, no. 1, pp. 1 – 14, 2006.

[46] L. H. Chen, E. H. K. Wu, M. H. Jin, and G. H. Chen, "Intelligent fusion of wi-fi and inertial sensor-based positioning systems for indoor pedestrian navigation," *IEEE Sensors Journal*, vol. 14, pp. 4034–4042, Nov 2014.

[47] Z. Chen, H. Zou, H. Jiang, Q. Zhu, Y. C. Soh, and L. Xie, "Fusion of wifi, smartphone sensors and landmarks using the kalman filter for indoor localization," *Sensors*, vol. 15, no. 1, pp. 715–732, 2015.

[48] V. C. Raykar, S. Yu, L. H. Zhao, A. Jerebko, C. Florin, G. H. Valadez, L. Bogoni, and L. Moy, "Supervised learning from multiple experts: Whom to trust when everyone lies a bit," in *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, (New York, NY, USA), pp. 889–896, ACM, 2009.

[49] W. J. H. T. Center, "Global positioning system standard positioning service performance analysis report," tech. rep., 2014.

[50] J. Chen and M. Bierlaire, "Probabilistic multimodal map matching with rich smartphone data," *Journal of Intelligent Transportation Systems*, vol. 19, no. 2, pp. 134–148, 2015.

[51] N. M. Razali and Y. B. Wah, "Power comparisons of shapiro-wilk, kolmogorov-smirnov, lilliefors and anderson-darling tests," 2011.

[52] J. P. Snyder, *Flattening the earth : two thousand years of map projections / John P. Snyder*. University of Chicago Press Chicago, 1993.

[53] D. Pyle, *Data Preparation for Data Mining*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1st ed., 1999.

[54] S. J. Preece*, J. Y. Goulermas, L. P. J. Kenney, and D. Howard, "A comparison of feature extraction methods for the classification of dynamic activities from accelerometer data," *IEEE Transactions on Biomedical Engineering*, vol. 56, pp. 871–879, March 2009.

[55] T. G. V. Niel, T. R. McVicar, and B. Datt, "On the relationship between training sample size and data dimensionality: Monte carlo analysis of broadband multi-

temporal classification," *Remote Sensing of Environment*, vol. 98, no. 4, pp. 468 – 480, 2005.

[56] J. Piper, "Variability and bias in experimentally measured classifier error rates," *Pattern Recognition Letters*, vol. 13, no. 10, pp. 685 – 692, 1992.

[57] E. K. Antonsson and R. W. Mann, "The frequency content of gait," *Journal of Biomechanics*, vol. 18, no. 1, pp. 39 – 47, 1985.

[58] E. J. J. Gonzalez, C. Luo, A. Shrivastava, K. Palem, Y. Moon, S. Noh, D. Park, and S. Hong, "Location detection for navigation using imus with a map through coarse-grained machine learning," in *Design, Automation Test in Europe Conference Exhibition (DATE), 2017*, pp. 500–505, March 2017.

[59] S. Dhar and U. Varshney, "Challenges and business models for mobile location-based services and advertising," *Commun. ACM*, vol. 54, pp. 121–128, May 2011.

[60] Z. Farid, R. Nordin, and M. Ismail, "Recent advances in wireless indoor localization techniques and system," *Journal of Computer Networks and Communications*, vol. 2013, 2013.

[61] A. Kupper, *Location-based Services: Fundamentals and Operation*. John Wiley & Sons, 2005.

[62] M. N. K. Boulos, A. Rocha, A. Martins, M. E. Vicente, A. Bolz, R. Feld, I. Tchoudovski, M. Braecklein, J. Nelson, G. Ó Laighin, C. Sdogati, F. Cesaroni, M. Antomarini, A. Jobes, and M. Kinirons, "Caalyx: a new generation of location-based services in healthcare," *International Journal of Health Geographics*, vol. 6, p. 9, Mar 2007.

[63] A. Zipf, "User-adaptive maps for location-based services (lbs) for tourism," 2001.

[64] M. Koühne and J. Sieck, "Location-based services with ibeacon technology," in *2014 2nd International Conference on Artificial Intelligence, Modelling and Simulation*, pp. 315–321, Nov 2014.