Jussi Parviainen

Studies on Sensor Aided Positioning and Context Awareness

Thesis for the degree of Doctor of Science in Technology to be presented with due permission for public examination and criticism in Tietotalo Building, Auditorium TC133, at Tampere University of Technology, on the 27th of September 2016, at 12 noon.
Abstract

This thesis studies Global Navigation Satellite Systems (GNSS) in combination with sensor systems that can be used for positioning and obtaining richer context information. When a GNSS is integrated with sensors, such as accelerometers, gyroscopes and barometric altimeters, valuable information can be produced for several applications; for example availability or/and performance of the navigation system can be increased. In addition to position technologies, GNSS devices are integrated more often with different types of technologies to fulfil several needs, e.g., different types of context recognition. The most common integrated devices are accelerometer, gyroscope, and magnetometer but also other sensors could be used.

More specifically, this thesis presents sensor aided positioning with two satellite signals with altitude assistance. The method uses both pseudorange and Doppler measurements. The system is required to be stationary during the process and a source of altitude information, e.g., a MEMS barometer, is needed in addition to a basic GNSS receiver. Authentic pseudorange and Doppler measurements with simulated altitude were used to test the algorithm. Results showed that normally the accuracy of couple of kilometers is acquired. Thesis also studies on what kind of errors barometric altimeter might encounter especially in personal positioning. The results show that barometers in differential mode provide highly accurate altitude solution (within tens of centimeters), but local disturbances in pressure need to be acknowledged in the application design. For example, heating, ventilating, and air conditioning in a car can have effect of few meters. Thus this could cause problems if the barometer is used as a altimeter for under meter-level positioning or navigation.

We also explore methods for sensor aided GNSS systems for context recognition. First, the activity and environment recognition from mobile phone sensor and radio receiver data is investigated. The aim is in activity (e.g., walking, running, or driving a vehicle) and environment (e.g., street, home, or restaurant) detection.
The thesis introduces an algorithm for user specific adaptation of the context model parameters using the feedback from the user, which can provide a confidence measure about the correctness of a classification. A real-life data collection campaign validate the proposed method. In addition, the thesis presents a concept for automated crash detection to motorcycles. In this concept, three different inertial measurement units are attached to the motorist’s helmet, torso of the motorist, and to the rear of the motor cycle. A maximum a posteriori classifier is trained to classify the crash and normal driving. Crash dummy tests were done by throwing the dummy from different altitudes to simulate the effect of crash to the motorist and real data is collected by driving the motorcycle. Preliminary results proved the potential of the proposed method could be applicable in real situations. In all the proposed systems in this thesis, knowledge of the context can help the positioning system, but also positioning system can help in determining the context.
Preface

The work for this dissertation was carried at the Department of Pervasive Computing, Tampere University of Technology (TUT). Also, I acknowledge the financial support from Ulla Tuomisen säätiö, Jenny ja Antti Wihurin rahasto and Nokia Foundation.

I would like to thank my supervisors Prof. Jarmo Takala and Dr. Jussi Collin for their guidance and comments throughout my research work. In addition, I thank Dr. Valerie Renaudin and Dr. Paul Groves for their constructive comments for the thesis manuscript. In addition, I want to also thank all the colleagues at TUT I have been working with. Especially, I would like to thank Dr. Martti Kirkko-Jaakkola, Dr. Helena Leppäkoski, Dr. Pavel Davidson, Mr. Jayaprasad Bojja and Mr. Arto Perttula.

Last, I would also like thank my family on my path through all the education systems. Especially, my thank goes to my wife Anniina and our little handful trio at home; Julius, Joonatan and Aini.

Tampere, 3.August 2016

Jussi Parviainen
Contents

Abstract i
Preface iii
Abbreviations vii
Symbols ix
List of Publications xiii

1 Introduction 1
  1.1 Objectives and Scope of Research . . . . . . . . . . . . . . . . . . . 3
  1.2 Main Contributions . . . . . . . . . . . . . . . . . . . . . . . . . . 3
  1.3 Author’s Contributions . . . . . . . . . . . . . . . . . . . . . . . . 4

2 Background 7
  2.1 Coordinate Frames . . . . . . . . . . . . . . . . . . . . . . . . . . 7
  2.2 Global Navigation Satellite Systems . . . . . . . . . . . . . . . . . 8
  2.3 Sensors . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 11
  2.4 Dead Reckoning . . . . . . . . . . . . . . . . . . . . . . . . . . . . 14
  2.5 Kalman Filtering . . . . . . . . . . . . . . . . . . . . . . . . . . . 16
  2.6 Pattern Recognition . . . . . . . . . . . . . . . . . . . . . . . . . 16
  2.7 Classifiers and Learning . . . . . . . . . . . . . . . . . . . . . . . 18

3 Positioning with GNSS and Sensors 23
  3.1 Altitude Aiding . . . . . . . . . . . . . . . . . . . . . . . . . . . . 23
  3.2 Sub Four Satellite Positioning . . . . . . . . . . . . . . . . . . . . 26

4 Context Recognition and Positioning 31
  4.1 Mobile Context Awareness . . . . . . . . . . . . . . . . . . . . . . 32
### Abbreviations

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AGNSS</td>
<td>Assisted Global Navigation Satellite System</td>
</tr>
<tr>
<td>ANN</td>
<td>Artificial Neural Network</td>
</tr>
<tr>
<td>C/N₀</td>
<td>Carrier power to Noise density ratio</td>
</tr>
<tr>
<td>CPU</td>
<td>Central Processing Unit</td>
</tr>
<tr>
<td>DOF</td>
<td>Degrees of Freedom</td>
</tr>
<tr>
<td>DOT</td>
<td>Department of Transportation</td>
</tr>
<tr>
<td>DT</td>
<td>Decision Tree</td>
</tr>
<tr>
<td>eCall</td>
<td>Emergency call</td>
</tr>
<tr>
<td>ECEF</td>
<td>Earth Centre Earth Fixed</td>
</tr>
<tr>
<td>ECI</td>
<td>Earth Centered Inertial</td>
</tr>
<tr>
<td>EKF</td>
<td>Extended Kalman filter</td>
</tr>
<tr>
<td>ENU</td>
<td>East-North-Up coordinate system</td>
</tr>
<tr>
<td>GNSS</td>
<td>Global Navigation Satellite System</td>
</tr>
<tr>
<td>GPS</td>
<td>Global Positioning System</td>
</tr>
<tr>
<td>GSM</td>
<td>Global System for Mobile communications</td>
</tr>
<tr>
<td>HVAC</td>
<td>Heating, Ventilating, and Air Conditioning</td>
</tr>
<tr>
<td>IMU</td>
<td>Inertial Measurement Unit</td>
</tr>
<tr>
<td>INS</td>
<td>Inertial Navigation System</td>
</tr>
<tr>
<td>Abbreviation</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>------------------------------</td>
</tr>
<tr>
<td>KF</td>
<td>Kalman filter</td>
</tr>
<tr>
<td>k-NN</td>
<td>k Nearest Neighbors</td>
</tr>
<tr>
<td>LBS</td>
<td>Location Based Services</td>
</tr>
<tr>
<td>LS-SVM</td>
<td>Least Square-Support Vector Machines</td>
</tr>
<tr>
<td>MAP</td>
<td>Maximum a Posteriori</td>
</tr>
<tr>
<td>MEMS</td>
<td>Micro Electro Mechanical System</td>
</tr>
<tr>
<td>NED</td>
<td>North East Down</td>
</tr>
<tr>
<td>OBD-II</td>
<td>On-Board Diagnostics</td>
</tr>
<tr>
<td>PDR</td>
<td>Pedestrian Dead Reckoning</td>
</tr>
<tr>
<td>PSAP</td>
<td>Public Service Answering Point</td>
</tr>
<tr>
<td>SMS</td>
<td>Short Message Service</td>
</tr>
<tr>
<td>SV</td>
<td>Space Vehicle</td>
</tr>
<tr>
<td>SVM</td>
<td>Support Vector Machine</td>
</tr>
<tr>
<td>TOA</td>
<td>Time Of Arrival</td>
</tr>
<tr>
<td>TCP/IP</td>
<td>Transmission Control Protocol / Internet Protocol</td>
</tr>
<tr>
<td>WLAN</td>
<td>Wireless Local Area Network</td>
</tr>
</tbody>
</table>
## Symbols

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\delta_t u(t)$</td>
<td>Additive clock biases of the receiver at time $t$</td>
</tr>
<tr>
<td>$\delta t^{(k)}(t)$</td>
<td>Additive clock biases of the satellite $k$ at time $t$</td>
</tr>
<tr>
<td>$\epsilon^{(k)}(t)$</td>
<td>Unmodelled errors and multipath of pseudorange measurement of the satellite $k$ at time $t$</td>
</tr>
<tr>
<td>$\epsilon_a(t)$</td>
<td>Unmodelled errors of accelerometer triad measurement at time $t$</td>
</tr>
<tr>
<td>$\epsilon_g(t)$</td>
<td>Unmodelled errors of gyroscope triad measurement at time $t$</td>
</tr>
<tr>
<td>$\epsilon_D$</td>
<td>Unmodelled errors of Doppler measurement the satellite $k$ at time $t$</td>
</tr>
<tr>
<td>$\zeta_i(x)$</td>
<td>Discriminant function of class $i$ with state $x$</td>
</tr>
<tr>
<td>$\kappa(0)$</td>
<td>Sea level pressure</td>
</tr>
<tr>
<td>$\kappa(h)$</td>
<td>Pressure at altitude $h$</td>
</tr>
<tr>
<td>$\mu$</td>
<td>Mean</td>
</tr>
<tr>
<td>$\xi$</td>
<td>Ionospheric, tropospheric and unmodelled errors of pseudorange</td>
</tr>
<tr>
<td>$\prod$</td>
<td>Product operator</td>
</tr>
<tr>
<td>$\rho$</td>
<td>Pseudorange</td>
</tr>
<tr>
<td>$\rho^{(k)}$</td>
<td>Pseudorange to satellite $k$</td>
</tr>
<tr>
<td>$\rho^{(k)}(t)$</td>
<td>Pseudorange to satellite $k$ at time $t$</td>
</tr>
<tr>
<td>$\dot{\rho}^{(k)}$</td>
<td>Pseudorange rate to satellite $k$</td>
</tr>
<tr>
<td>$\sum$</td>
<td>Sum operator</td>
</tr>
</tbody>
</table>
\[\Sigma\] Covariance

\[\Psi(t)\] Heading at time \(t\)

\[\Psi(t_0)\] Initial heading at time \(t_0\)

\[\omega_h(t)\] Measurement of a horizontal gyroscope at time \(t\)

\[\omega_m(t)\] Measurement of a gyroscope triad at time \(t\)

\[\omega_t(t)\] True angular rate at time \(t\)

\[\mathbf{a}_m^B(t)\] Measurement of a triad accelerometers in body frame at time \(t\)

\[a_{\text{norm}}\] Norm of the accelerometer triad

\[\mathbf{a}_t^B(t)\] True acceleration of triad of accelerometers in body frame at time \(t\)

\[a_t(t)\] True acceleration of an accelerometer at time \(t\)

\[a_x\] x-component of the accelerometer triad

\[a_y\] y-component of the accelerometer triad

\[a_z\] z-component of the accelerometer triad

\[b\] Clock bias caused by receiver clock

\[\mathbf{b}_a(t)\] Biases of accelerometer triad

\[\mathbf{b}_g(t)\] Biases of gyroscope triad

\[\dot{b}\] Clock drift caused by receiver clock

\[C_i\] Class \(i\)

\[c\] Velocity of light

\[D\] Coordinate transformation matrix

\[\mathbf{f}_k\] State transition function at time \(k\)

\[g\] Magnitude of gravitational acceleration

\[\mathbf{g}^B\] gravity vector converted to a body frame

\[h\] Altitude
<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$h_k$</td>
<td>Measurement model function at time $k$</td>
</tr>
<tr>
<td>$I^{(k)}(t)$</td>
<td>Ionospheric propagation delay of the satellite $k$ at time $t$</td>
</tr>
<tr>
<td>$k_T$</td>
<td>Temperature lapse rate</td>
</tr>
<tr>
<td>$n$</td>
<td>Number of steps</td>
</tr>
<tr>
<td>$P(C_i</td>
<td>x)$</td>
</tr>
<tr>
<td>$P(C_i)$</td>
<td>Prior probability of class $C_i$</td>
</tr>
<tr>
<td>$p(x</td>
<td>C_i)$</td>
</tr>
<tr>
<td>$p(x)$</td>
<td>Probability density function of state $x$</td>
</tr>
<tr>
<td>$q_k$</td>
<td>Process noise at time $k$</td>
</tr>
<tr>
<td>$r$</td>
<td>Travelled distance</td>
</tr>
<tr>
<td>$r_k$</td>
<td>Measurement noise at time $k$</td>
</tr>
<tr>
<td>$R$</td>
<td>Universal gas constant</td>
</tr>
<tr>
<td>$S$</td>
<td>Step length</td>
</tr>
<tr>
<td>$S_a$</td>
<td>Scale factor and cross-coupling matrix of accelerometer triad</td>
</tr>
<tr>
<td>$S_g$</td>
<td>Gyroscope scale factor</td>
</tr>
<tr>
<td>$s^{(k)}(t)$</td>
<td>Position of the satellite $k$ at time $t$</td>
</tr>
<tr>
<td>$s^{(k)}$</td>
<td>Position of the satellite $k$</td>
</tr>
<tr>
<td>$s_x$</td>
<td>$x$-component of the position of the satellite $k$</td>
</tr>
<tr>
<td>$s_y$</td>
<td>$y$-component of the position of the satellite $k$</td>
</tr>
<tr>
<td>$s_z$</td>
<td>$z$-component of the position of the satellite $k$</td>
</tr>
<tr>
<td>$T^{(k)}(t)$</td>
<td>Tropospheric propagation delay of the satellite $k$ at time $t$</td>
</tr>
<tr>
<td>$T(0)$</td>
<td>Sea level temperature</td>
</tr>
<tr>
<td>$u(t)$</td>
<td>Position of the receiver (or user) at time $t$</td>
</tr>
<tr>
<td>$u$</td>
<td>Position of the receiver (or user)</td>
</tr>
<tr>
<td>Symbol</td>
<td>Description</td>
</tr>
<tr>
<td>--------</td>
<td>-------------</td>
</tr>
<tr>
<td>$u_x$</td>
<td>x-component of the position of the receiver (or user)</td>
</tr>
<tr>
<td>$u_y$</td>
<td>y-component of the position of the receiver (or user)</td>
</tr>
<tr>
<td>$u_z$</td>
<td>z-component of the position of the receiver (or user)</td>
</tr>
<tr>
<td>$\mathbf{v}^{(k)}$</td>
<td>Relative velocity between the receiver and satellite $k$</td>
</tr>
<tr>
<td>$\mathbf{w}_i$</td>
<td>Weight vector for class $i$</td>
</tr>
<tr>
<td>$w_{i0}$</td>
<td>Bias for class $i$</td>
</tr>
<tr>
<td>$\mathbf{x}_k$</td>
<td>State vector at time $k$</td>
</tr>
<tr>
<td>$x_i$</td>
<td>$i$th element of vector $\mathbf{x}$</td>
</tr>
<tr>
<td>$\mathbf{y}_k$</td>
<td>Measurement vector at time $k$</td>
</tr>
</tbody>
</table>
List of Publications

This thesis consists of an introductory part and the following publications referred as I-IV


The publications are reproduced here with kind permissions from the publishers.
1 Introduction

Recently applications related to positioning have spread to many areas of our digital life. The navigating devices based on satellite positioning are getting more and more popular. These Global Navigation Satellite Systems (GNSS) will indeed operate well when the receiver has a free view of the satellites. However, the need for positioning has not been limited only to open roads where an unrestricted view to the satellites has often been guaranteed. Apart from these open areas, positioning should also be possible in tunnels, indoors, and in other regions where the satellite signals are otherwise unavailable. In addition, the availability of positioning is extremely important to rescue vehicles for which a defective or faulty position can lead even to fatal situations. Systems which are aware of location information could also assist consumers, e.g., a motorist could be notified of free places in a nearby parking lot where the device then would guide him/her.

When satellite signals are blocked, a secondary system is needed to produce the navigation solution. There are devices, which do not require external signals, e.g., Inertial Navigation Systems (INS) consisting of three accelerometers and three gyroscopes integrated with a navigation computing unit. This kind of navigation system produces a 3-dimensional position. However, an INS unit does not offer an absolute position but the navigation takes place relatively with respect to the initial location. There is also a problem that the position error accumulates, in other words the position error increases in time whereas the variance of the error of the satellite positioning remains constant provided that the view of the satellites remains unrestricted. Navigation grade INS units are still extremely expensive; the prices exceed 100,000 euros, whereas one can get a satellite navigation receiver with a few euros. By combining the satellite and inertial navigation, lower quality INS components can be used. Then the system needs to rely on the pure INS solution only when the satellite signals are blocked and the errors do not usually have time to accumulate too large. However, when the satellite signals are available, it will be possible to use the advantages of both
methods. For example, satellite navigation unit can compensate the offset errors, which are present especially in lower quality inertial sensors. In general, the errors in inertial navigation are usually in form of low frequency variation whereas the disturbance of the satellite positioning is a high frequency noise.

Currently gyroscopes are the most expensive components in INS units, thus there has been several studies where only one gyro is used instead of three. However, in such a case, the acceleration measurement cannot be separated from the gravity component, which can cause large position errors. Yet another sensor, which can be used for positioning, is a barometric pressure sensor, also known as a barometer. In theory, a barometer can measure absolute height information. However, in practice there are large errors due to the variation of the local air pressure. Because of this, a reference barometer or some other method, for instance GNSS calibration, is needed to indicate the local pressure, which changes over the time. By using this differential barometry information, the height component of the satellite positioning can be assisted, which can contain more than 10 meters of vertical position error. The meter level accuracy of the height can also help, among others, navigation at multilevel road interchanges and indoor positioning in high buildings.

In addition to positioning technologies, GNSS devices are integrated more often with different types of technologies to fulfil several needs, e.g., different types of context recognition, where the information of the used sensors is used to determine some physical context, e.g., the user is running or walking or even that the user is in a meeting or shopping. The most common integrated devices are accelerometer, gyroscope, magnetometer and barometer but also other sensors could be used, such as proximity and light sensors. Many of the modern smartphones already contain these sensors and design of context based applications for modern smartphones has boomed in the past years. For application designers the smartphone environment is very different from the earlier personal computer world. Many of the applications are now running continuously and the environment and activity of the user changes many times a day. In addition, the applications are often required to provide only the most relevant information or services to the user. In the present paradigm, applications require the user to provide this information, such as ‘I am in a meeting’. Clearly it would be more convenient if this context information could be inferred automatically. For this reason, context recognition using built-in sensors of modern mobile phones is an active research topic. The research problem is far from trivial, as the sensors provide only indirect information; there is no sensor for directly detecting a context, e.g., ‘meetings’, or motion mode, e.g.,
'jogging'. Furthermore, the required accuracy for the recognition is relatively high as misclassifications can lead to very annoying user experiences.

1.1 Objectives and Scope of Research

When GNSS is integrated with sensors, such as accelerometers, gyroscopes, and barometric altimeters valuable information can be produced for multiple applications. When the satellite signals in a positioning system are restricted, sensor aided system can increase availability or/and performance of the system. In addition to positioning, other valuable information can be produced, such as the motion mode for the user.

The main research question in this thesis is how sensors and GNSS can be combined to increase availability of positioning systems and what contextual information can be extracted from such a system. More specifically, how we can use a limited number of satellite signals with altitude aiding, what kind errors will affect on barometric altimeter estimates and what kind of contextual information can be acquired from mobile devices and inertial measurement units. In this thesis, we are mainly interested in the applications that can infer the activity and environment of the user, i.e. they will answer the questions what you are doing and what kind of environment you are in. More specifically, in this topic, the questions are how user specific adaptation can be obtained based on feedback of the user. In addition, we are exploring how inertial sensors should be distributed in order to detect the crash of the motorcycle. The focus in this thesis is to exploit micro-electro-mechanical system (MEMS) or equivalent low cost sensors.

1.2 Main Contributions

The main contributions of the thesis are algorithm development for GNSS aided sensor systems as follows:

- Two satellite positioning algorithm with height assistance. This algorithm gives coarse position of the device, which allows the rest of the visible GNSS satellites to be acquired faster. Algorithm combines pseudorange and Doppler positioning with an external altitude measurement.

- The error sources that may occur when using barometric altimeter for personal positioning. We show that different disturbances, such as opening a window or driving in a tunnel, affect to the pressure reading and thus
have effect on altitude estimation. To separate these error sources from the varying atmospheric pressure due to different weather conditions, reference barometer at a known altitude is used. The results show that barometers in differential mode provide highly accurate altitude solution (within tens of centimeters).

- Methods for context recognition with sensors and GNSS system. The proposed methods detect different activity modes (e.g., walking, running or driving a vehicle), environments (e.g., office, restaurant, street), and finally a crash detection. Furthermore, an adaptation algorithm for the user specific data is proposed based on the feedback from the user. User gives feedback whether the classification result given by the algorithm was correct or incorrect.

1.3 Author’s Contributions

The work presented in this thesis has been reported in publications I–IV and none of these publications has been used in another academic thesis. The author has been the first author in the publications II–IV and was the second in publication II. The co-authors of the publications have seen this description of the contributions related to the publications and they agree with the description.

The results of sensor aided navigation are published in publications I and II. The author implemented the two satellite positioning algorithm proposed in publication I, which combines pseudorange and Doppler positioning with altitude aiding. The author wrote also the first versions of the paper. Publication I is a journal paper and is fully reviewed.

In publication II, the author performed the data analysis and arranged the data collection campaign. In addition, the author made a major contribution to the writing of the manuscript. Publication II is a conference paper and is accepted based on abstract.

Results on sensor aided context recognition have already been published in two publications, III and IV. In publication III, the author developed a context aware algorithm with adaptivity for mobile phones. The author also had a large role in organising the data collection and was the main contributor in analyzing the collected data. Publication III is a journal paper and is fully reviewed.

In publication IV, the author developed the algorithm for crash detection for motor cycles. The author made a major contribution to the writing of the manuscript.
Publication IV is a conference paper and is fully reviewed.
2 Background

The theory and background behind the methods used in this theses are introduced in this chapter. First, the coordinate frames and the satellite positioning systems are introduced with their common positioning algorithms. After this the sensors are introduced continuing with an introduction of sensor based positioning and pattern recognition.

2.1 Coordinate Frames

In order to navigate or position with any system, we need to have an unambiguous frame, thus we need a well defined coordinate system. For a coordinate frame we need an origin and a set of axes with respect to which the motion or position can be described. An inertial coordinate frame is a frame that does not accelerate or rotate respect to rest of the universe [41]. For navigation on the Earth we use Earth centered inertial (ECI) frame. Its origin is at the center of the Earth but the axes do not spin along with the Earth but keep their direction with respect to the stars. In ECI, z axis is aligned with the Earth rotation axis in the direction of the North pole and x and y axes are in the equatorial plane. Naturally, ECI frame is not a good frame to express a navigation solution on the Earth as the coordinate frame axes do not rotate with the Earth. An Earth fixed earth centered frame (ECEF) suits better for that purpose, as the axes are fixed to rotating Earth. It has the same origin and z axis as in ECI, but x and y axes are pointing to zero degree and 90 degree longitude, respectively.

A Local tangential-plane frame or local frame is a frame where coordinates are determined by a tangent plane of the Earth model ellipsoid fitted to a reference point. This is usually modelled as a East-North-Up (ENU) or North-East-Down system (NED). For example in ENU system, the Up axis is pointing perpendicular to the tangent plane, towards sky, and the North and East axes are pointing to North and East, respectively.
Chapter 2. Background

Fig. 2.1. Visualization of ECI or ECEF (in ECEF Earth rotates with the axes in ECI not), local (ENU), body (subindex u in coordinates) and device (subindex d in coordinates) coordinate frames

Body frame (or user frame) can be defined by the vehicle (in land, air, water), pedestrian or any object, where the origin is conventionally located at the same reference point of the (moving) object. In aviation convention, X axis is usually aligned with (typical) forward velocity vector, Z is aligned with local gravity vector (up) and Y is cross product of X and Z axis vectors. However, there are no standards for land vehicles or pedestrians and, therefore, various conventions are used.

In addition, sometimes there is need to also define a device frame which can be inferred as a type of body frame. In this case, the sensors are fixed in directions which do not coincide with the body frame axes. The orientation between the body and device is often relevant to navigation. One example of this is a pedestrian who is carrying a navigation device in free orientation in hand. This orientation is independent of the velocity of the pedestrian and there is no preferred direction of motion in the device frame. These most relevant frames for navigation are illustrated in Fig. 2.1.

2.2 Global Navigation Satellite Systems

Currently two GNSS systems exist, which have full global coverage, namely the Global Positioning System (GPS) operated by United States and the Russian GLONASS. In addition, there exists European Galileo and Chinese Beidou, which have been under construction during the work done in this thesis.

All GNSS systems can be divided to three segments: space, control, and user segments. The space segment consists of satellites, also known as space vehicles (SV), orbiting the Earth. For example, the GPS system has an almost circular
orbit with a radius of approximately 26560 km, which makes the period of the GPS satellites 11 hours 58 minutes. The in orbit or the planned number of satellites for each GNSS is around 30 depending on the system.

The control segment is the controller unit of GNSS. It consists of ground based receivers and transmitters, which are communicating with the satellites. The tasks of the control segment are to maintain and predict satellite orbits, monitor and maintain satellite health, update navigation messages and adjust the onboard high-precision clocks to maintain accuracy.

The user segment consists of the positioning devices that process the signals transmitted by GNSS satellites. These devices will then provide location and time information for the user. Devices may range from low cost hand held receivers to high-end survey and mapping devices.

2.2.1 Conventional GNSS Positioning

The GNSS uses carrier frequencies in the L-band, onto which ranging code signals are modulated. Determination of the user (i.e. receiver) position is made with these ranging signals and the navigation messages transmitted by the satellites. Traditional GNSS is a one way positioning system, i.e. the user does not need to send any signals, nor is even able to communicate with the satellites. Positioning with these modern GNSS systems is based on the concept of time of arrival (TOA) ranging [53].

The distance between satellite and receiver can be measured using the encoded message transmitted by the satellite. The message includes the time of transmission and current satellite position. As GNSS satellites contain nanosecond level atomic clocks, the time of transmission is very accurate [53]. However, GNSS receivers usually contain less accurate crystal clocks and thus the time of reception is biased. Biased range measurements are called pseudoranges. Pseudorange \( \rho^{(k)}(t) \) at time \( t \) can be written as

\[
\rho^{(k)}(t) = \| s^{(k)}(t) - u(t) \| + c \left( \delta t_u(t) - \delta t^{(k)}(t) \right) + I^{(k)}(t) + T^{(k)}(t) + \epsilon^{(k)}(t),
\]

where \( k = 1, 2, \ldots, N \) is the unique number of the satellite and \( s(t)^{(k)} \) and \( u(t) \) are the Cartesian positions of the satellite and the receiver in a fixed frame (e.g., ECI or ECEF), respectively. Parentheses around variables are used to distinguish a superscript index from an exponent. The symbol \( c \) is used to denote the speed of light, \( \delta t_u(t) \) and \( \delta t^{(k)}(t) \) are the additive clock biases of the receiver and satellite,
respectively, \( I^{(k)}(t) \) and \( T^{(k)}(t) \) are the ionospheric and tropospheric propagation delays, and \( \epsilon^{(k)}(t) \) encompasses unmodelled errors and multipath [82].

To simplify more, if we combine the ionospheric and tropospheric errors with the unmodelled errors (to term \( \xi \)) and omit time domain indexing, a pseudorange of the satellite \( k \), \( \rho^{(k)} \), is written as [82]

\[
\rho^{(k)} = \left\| s^{(k)} - u \right\| + c(\delta t_u - \delta t^{(k)}) + \xi. \tag{2.2}
\]

In addition, if the ECEF frame is used, the Sagnac effect due to Earth rotation has to be compensated. During propagation time of the satellite signal, a user clock on the surface of the Earth will experience a finite rotation with respect to an ECI coordinate system. There exist multiple ways to do Sagnac correction, but conventionally, correction is applied after the rough position is calculated [53]. Thus, we do not present Sagnac error terms in this thesis. As the user can correct the satellite clock error, \( \delta t^{(k)} \) in equation (2.2), by using parameters transmitted in the navigation message and when the distance error caused by user clock bias is replaced with \( b \) and the norm in the (2.2) is expanded, corrected pseudorange \( \rho_{cor} \) is obtained as

\[
\rho_{cor}^{(k)} = \sqrt{(s_x^{(k)} - u_x)^2 + (s_y^{(k)} - u_y)^2 + (s_z^{(k)} - u_z)^2} + b + \xi. \tag{2.3}
\]

In (2.3), the unknowns are user position in ECEF frame \( u_x, u_y, u_z \) and bias \( b \). Generally solving these four unknowns requires four pseudoranges and then the receiver position and clock bias can be solved either in closed-form, iteratively based on linearization, or by using Kalman filtering. A review of these different methods can be found, e.g. from [53, 58].

### 2.2.2 Doppler Positioning

In addition to pseudoranges, Doppler shift based measurements can be used for positioning. Indeed before the era of modern TOA based GNSS systems there were systems, such as US Transit and Russian Tsikada, which were purely based on Doppler shift based methods [41, 53]. The Doppler positioning equations can be expressed as a dot product

\[
\dot{\rho}^{(k)} = v^{(k)} \cdot \frac{s^{(k)} - u}{\| s^{(k)} - u \|} + \hat{b} + \epsilon_D^{(k)}, \tag{2.4}
\]

where \( \dot{\rho}^{(k)} \) is the measured pseudorange rate (in units of velocity) to satellite \( k \), \( v^{(k)} \) denotes the relative velocity between the receiver and satellite \( k \), and \( \hat{b} \) is
the receiver clock drift (i.e., time derivative of the receiver clock bias $b$) [66]. Term $\epsilon_D^{(k)}$ represents other unmodelled error sources. Unlike pure pseudorange positioning, Doppler positioning does not require the receiver clock bias to be solved, but another unknown, the receiver clock drift $\dot{b}$, is introduced. In addition, in reality there exists also a satellite clock drift. However, as the satellite ephemeris contains the information about the satellite clock errors, satellite clock drift is here neglected.

The relative velocity $v^{(k)}$ in (2.4) consists of the velocity of satellite $k$ with respect to the velocity of the receiver. The former can be obtained using the satellite ephemeris data [98], but the latter is unknown, and typically assumed to be zero. If the receiver, however, is not stationary, three more equations would be needed to account for its velocity. In personal positioning, the requirement of stationarity can be estimated using MEMS accelerometers very accurately [26].

2.3 Sensors

GNSS alone cannot provide reliable, everywhere available positioning service for all situations. To go beyond capabilities of standalone GNSS, the receivers are often combined with additional sensors. In this section, we will introduce the main sensors investigated in this thesis.

2.3.1 Accelerometer

A simple accelerometer can be modelled as a spring mass, where the pickoff measures the position of the proof mass respect to case. This is illustrated in Fig.2.2. However, during free fall the accelerometer shows zero acceleration. Thus, an accelerometer does not directly measure a component of acceleration, but a component called specific force [104], which includes only non-gravitational acceleration. This means that if the sensitive axis of the accelerometer is placed vertically and the sensor is at rest, the output of the accelerometer is equal to negative magnitude of gravity. Adding in the gravitational acceleration $g$ enables the use of accelerometer as a velocity or position source.

As Micro Electro Mechanical System (MEMS) technology allows manufacturing of very small sensors, it is not too costly to equip a personal device with a triad of accelerometers. Especially, almost all the current smart phones have it. Probably the most common application that uses accelerometers is the re-orientation of the screen in a horizontal or vertical position.
An error model of the measurements of a triad of accelerometers $a_m^B$ in the body frame can be written as follows

$$a_m^B(t) = S_a \left( a_t^B(t) - D_L^B g_L \right) + b_a(t) + \epsilon_a(t), \quad (2.5)$$

where $a_t^B$ is true acceleration, $S_a$ is scale factor and cross-coupling matrix, $D_L^B$ is coordinate transformation matrix from local frame to body frame, $g_L$ is gravity vector in local frame (i.e. $g = \begin{bmatrix} 0 & 0 & -g \end{bmatrix}^T$), $b_a(t)$ is accelerometer bias term, and $\epsilon_a(t)$ is containing random noise and unmodelled systematic errors [40]. The diagonal terms of scale factor and cross-coupling matrix $S_a$ model the linearly proportional scale factor errors, whereas off diagonal terms model the cross coupling errors due to misalignment of the sensitive axes of the sensor with respect to the orthogonal axes. Furthermore, the bias term $b_a(t)$ is a threshold error, which is in the most cases the dominant term in inertial sensors [41]. Reader may find more detailed information about $S_a$ and $b_a(t)$ from [40, 41, 116].

After the gravity component have been compensated and the transformation to the proper frame, e.g. ENU, ECEF, or ECI frame, is done, the velocity and travelled distance can be calculated using numerical integration. The final equation for solving the velocity and position depends on the selected frame. For example, the Coriolis force needs to be taken into account. However, due to the large error terms in current MEMS accelerometers, in many applications it is not feasible to calculate travelled distance using the integration as the errors accumulate over time. Instead, other methods such as pedestrian dead reckoning are introduced, where the steps of a pedestrian are calculated based on accelerometer signal [69, 78]. In addition to positioning, the accelerometer has shown to be very powerful sensor in motion recognition [110]. This is partly discussed in Chapter 4.
2.3. Sensors

2.3.2 Gyroscope

As the accelerometer was the first mass commercialized inertial sensor, gyroscopes have followed their path [18]. Gyroscope is an inertial sensor that measures angular rotation with respect to inertial space about its input axis [48]. Usually the rate is expressed in units of rad/s. Three gyroscopes with linearly independent measurement axes are needed to observe orientation of freely rotating object. As accelerometers, conventionally the error model for gyroscope triad measurement \( \omega_m \) is

\[
\omega_m(t) = S_g \omega_t(t) + b_g(t) + \epsilon_g(t),
\]

(2.6)

where \( \omega_t \) is true angular rate, \( S_g \) is scale factor and cross coupling matrix, \( b \) is bias term and \( \epsilon_g(t) \) contains random noise and unmodelled errors. In theory, the heading \( \Psi(t) \), i.e. the angle in respect of north direction, at time \( t \) can be calculated using single integral horizontal plane gyro measurements \( \omega_h(t) \)

\[
\Psi(t) = \Psi(t_0) + \int_{t_0}^{t} \omega_h(t)dt,
\]

(2.7)

where \( t_0 \) is the initial time epoch, \( t \) is the current time epoch, and \( \Psi(t_0) \) is initial heading. However, the gyro measurement is almost never in horizontal plane. Thus, in most cases, 3-dimensional attitude solution must be calculated. Attitude can be updated using coordinate transformation matrices or calculating quaternion. More detailed information can be found e.g. from [41, 116]. In uncalibrated MEMS gyroscope, usually the main source of heading error is the bias term, which is affected sources such as temperature changes and linear acceleration. In addition, even in constant circumstances there is temporal variation in the bias term. When gyroscopes are used for navigation purposes, sensor is always calibrated before or during the use. Even though bias estimation and calibration has been done for decades, it is still an active research topic [27, 55].

2.3.3 Barometric Pressure Sensor

It is commonly known that air pressure varies depending on the altitude [12, 119]: the higher the altitude, the lower the pressure. Advances in MEMS technology have made it possible to integrate low-cost and small-size barometric altimeters to mobile devices. However, air pressure measurements do not give absolute altitude information: the normal air pressure at sea level is approximately 1013 hPa, but it varies locally depending on weather conditions. Near the surface of the Earth, a
pressure change of one hPa corresponds approximately to an altitude difference of 8 meters, provided that weather conditions are not changing [46]. More accurately, the altitude $h$ can be related to the ambient pressure $p(h)$ using the International Standard Atmosphere model [119] given as

$$h = \frac{T(0)}{k_T} \left( 1 - \frac{\kappa(h)}{\kappa(0)} \right) \frac{k_T R}{g}, \tag{2.8}$$

where $R$ is the universal gas constant, $k_T$ is the temperature lapse rate, $g$ is the gravitational acceleration, and $T(0)$ and $\kappa(0)$ are the sea level temperature and pressure, respectively. There are also more complex derivations of pressure altitude, e.g. in [16]. More detailed information about barometric formulae can be found in [12].

### 2.3.4 Other Sensors

In addition to above mentioned sensors, GNSS systems can be combined with many other sensors that are not in the scope of this thesis. For example, in land vehicle navigation different types of odometers can be used, e.g., wheel based sensors and Doppler radars [70, 87]. Also one option is to use a magnetometer, i.e. digital compass, to produce heading of the measurement device [99]. In addition, as the current mobile phones contain multiple number of different sensors, these can be combined with GNSS or they can work alone to produce information about the user’s state. Some of these sensors are WLAN, bluetooth, light, and proximity sensors.

### 2.4 Dead Reckoning

In Dead Reckoning (DR) measured displacement is added to the previous position in order to compute the current position. Obviously, the initial position needs to
be obtained from another source. Position change can be measured using many different methods, e.g. using odometers, ground speed radars, inertial sensors, or accelerometer based pedometers [78, 88]. One drawback of the dead reckoning method is that the position error increases during time as errors accumulate over time as illustrated in Fig. 2.3.

An inertial navigation system (INS) is complete 3-dimensional dead reckoning system which is implemented using inertial measurement unit (IMU) containing accelerometers and gyros. Typically, an IMU contains three orthogonal accelerometers and gyros, which provide a full 6 degrees of freedom (DOF) INS. There have been two types of IMUs, gimbaled IMU where a stabilized platform where gimbals are used to retain a orientation respect to inertial space, and strapdown IMU where sensors are attached to device without such mechanization [106]. However, currently gimbaled IMUs are no longer made and low cost MEMS sensors have always been using strapdown method.

As a form of dead reckoning, inertial navigation also suffers from error accumulation. Full determination of INS error is a complex problem. The accelerometer bias heavily affects the position (distance traveled) error and the gyro bias affects to both orientation and position error. For example, constant velocity error of $0.01 \text{ m/s}^2$ or constant $1 \text{ mrad}$ attitude error will cause almost $50 \text{ m}$ of error in position in 100 seconds. Thus, without very high accuracy sensors the error in navigation solution will grow rapidly in a short time [40]. More detailed description of INS and its performance can be found in numerous works, e.g. from [36, 41, 104, 105, 116].

One form of DR is pedestrian dead reckoning, PDR, where the travelled distance is calculated using the kinematics of human gait [85]. Traditionally this is done by detecting the user’s steps, estimating their length, and determining the heading direction [44, 63, 78]. A method is to estimate the steps using norm of the accelerometer triad measurement as follows

$$a_{\text{norm}} = \sqrt{a_x^2 + a_y^2 + a_z^2} \quad (2.9)$$

and the heading $\Psi$ is estimated using gyros and/or digital compass i.e. magnetometer. By detecting the cyclic pattern of the pedestrian signal, the travelled distance $r$ can be calculated using number of steps $n$ and estimated steps length $S$ simply as

$$r = n S. \quad (2.10)$$

Because PDR is a relative position method, the initial position of the user must also be known by some means. Usually this is done by integrating PDR with
GNSS outdoors and WLAN-based positioning indoors where the initial or reference position can be acquired [23, 43].

2.5 Kalman Filtering

A common way to combine data from different sensors, especially for navigation purposes, is to use Kalman filter (KF) [19, 51]. Kalman filtering can be seen as a special case of Bayesian filtering where the conditional posterior distribution can be computed analytically. The algorithm works in a recursive manner using two steps. In the first step the KF predicts the estimates of current states $x$ through state transition function $f$ using the system model

$$x_{k+1} = f_k(x_k, q_k), \quad (2.11)$$

where $q$ is a random variable denoting process noise. In the second step, the states can be updated using measurement model

$$y_{k+1} = h_k(x_{k+1}, r_k). \quad (2.12)$$

Here $h$ is a measurement model function and $y$ denotes the current measurements and $r$ denotes measurement noise. In the KF, both transition and measurement functions, $f$ and $h$, respectively, are assumed to be linear. In addition, noise terms $q$ and $r$ are assumed to be additive zero mean Gaussian white noises. However, in many applications, the assumption of the linearity does not apply. In an extended Kalman filter (EKF), the system and measurement model functions are linearized using first order Taylor series, which allows us also work with nonlinear $h$ and $f$ functions (when those can be linearized). However, whereas KF provides statistically optimal estimate for linear systems with no time-correlated noise, due to approximation the EKF will only give approximations of these.

2.6 Pattern Recognition

In addition to positioning technologies, GNSS devices are integrated more often with different types of technologies to fulfil several needs. One of these is context recognition, where the information of the sensors is used to determine some physical context, e.g., the user is running or walking or even that the user is in a meeting or shopping. Conventionally, context recognition is done using a pattern recognition methods. In pattern recognition we try to extract different features from the sensed data, e.g., variances, means, maximum value, and then try to
classify the data based on the extracted features. For example, data of the mobile device including sensors, such as accelerometer, gyroscope, WiFi and GNSS, could be used for obtaining the different features. After processing those features in some smart algorithm the result could be for example, that the user is walking in an office.

A basic pattern recognition system can be divided into five different components; sensing, segmentation, feature extraction, classification and post-processing [30] as shown in Fig. 2.4.

- **Sensing.** The first part of the pattern recognition system is always the sensing element. It is some kind of transducer, e.g., camera, microphone, or accelerometer. The specifications such as sampling frequency, sensitivity, bandwidth, etc. of the transducer may limit it’s use in some specific problems. For example, accelerometer sampling frequency should be high enough, if we want to detect very quick movements.

- **Segmentation.** After the sensing element the data needs to be segmented. This means that data is divided into a certain time domain windows, from which the features then can be calculated. How the segmentation is done depends on the application; different tactics are used if the application is camera vision, speech recognition or motion mode estimation. For, example if we want to detect that the user is walking from the accelerometer data, the length of window could be chosen so that it correlates approximately...
with the stepping rate of normal human walking motion (i.e. around 1.8 Hz [64]).

- **Feature Extraction.** Feature extraction is also a key element of a pattern recognition system. The boundary between it and classification is not always clear. Ideally, feature extraction should yield features that would make the classification task trivial. The idea is to extract features that would have similar values from samples from the same category, e.g. the user is walking, but different values for samples from other categories, e.g. the user is running or sitting. For example, if we want to classify between the running and walking categories, generally, the variance from the body mounted accelerometer has higher values during running and lower values during walking. In some cases, we want the features to be invariant to some specific transformations. For example, in motion mode recognition, usually the features should be invariant of rotation, i.e. the attitude of the sensing element should not have an effect on the features.

- **Classification.** The task of a classifier is to take the features given by the feature extraction component and give them a category label. However, it is not always straightforward to select unanimously one label, thus many classifiers assign probabilities (or values proportional to probabilities) to each category. Usually the most probable category (class) is given as a output, however, the probabilities of the other categories might contain useful information, e.g. if we want to make the classifier user specific.

- **Post Processing.** Once the classifier has made decision about the category, there usually follows an action. For example, if a classifier running on a mobile phone shows a classification result of ‘walking’, the step counting application might be opened automatically.

### 2.7 Classifiers and Learning

An important task in designing any pattern recognition system is the selection of the classifier. Features cannot always be extracted in a such a manner that the classification can be done in a trivial manner [30]. This section gives an introduction to the classification and learning methods which are relevant to the thesis.
2.7. Classifiers and Learning

2.7.1 Bayesian Classifiers

In the publications III and IV, the main approach used to classify different types of context (or classes) is Bayesian decision theory. In a Bayesian classification task with $N$ classes, $C_1, C_2, \ldots, C_N$ with an unknown pattern, presented by a feature vector $\mathbf{x}$, the idea is to form $N$ conditional probabilities also called a posteriori probabilities $P(C_i|\mathbf{x})$. They present a probability that an unknown pattern $\mathbf{x}$ will belong to the respective class $C_i$. Thus, the selected class will then be computed based on the maximum value of these $N$ conditional probabilities or the equivalent maximum of a suitably defined function of them. This kind of classification is called maximum a posteriori (MAP) classification [30].

In MAP classification, the probabilities $P(C_i|\mathbf{x})$ for the classes can be obtained with Bayes’ rule as follows

$$P(C_i|\mathbf{x}) = \frac{p(\mathbf{x}|C_i)P(C_i)}{p(\mathbf{x})},$$

(2.13)

where $p(\mathbf{x}|C_i)$ is a class conditional probability density function (PDF), $P(C_i)$ is the prior probability of class $C_i$, and $p(\mathbf{x})$ is PDF of $\mathbf{x}$, i.e. $p(\mathbf{x}) = \sum_{i=1}^{N} p(\mathbf{x}|C_i)P(C_i)$. The prior probabilities represent how probable each class is before we have any other information. If we have some historical data from each class, we can use that to estimate the prior probabilities. As the denominator in Bayes formula in (2.13) is the same for all the classes, a conventional approach is just to maximize the numerator term. In addition, when the class conditional distribution is Gaussian, we can write

$$p(\mathbf{x}|C_i) = \frac{1}{(2\pi)^{d/2} |\Sigma_i|^{1/2}} e^{\frac{1}{2} (\mathbf{x}-\mu_i)^T \Sigma_i^{-1} (\mathbf{x}-\mu_i)},$$

(2.14)

with covariance matrix $\Sigma_i$ and mean $\mu_i$. A common approach is to maximize the natural logarithm of this numerator term as it will simplify the calculations. Thus we get the classification result by selecting the class $i$ that gives the largest value of discriminant function of class $i$ with feature vector $\mathbf{x}$, $\zeta_i(\mathbf{x})$, defined as

$$\zeta_i(\mathbf{x}) = -\mathbf{x}^T \frac{1}{2} \Sigma_i^{-1} \mathbf{x} + \Sigma_i^{-1} \mu_i \mathbf{x} - \frac{1}{2} \mu_i^T \Sigma_i^{-1} \mu_i - \frac{1}{2} \ln \Sigma_i + \ln P(C_i).$$

(2.15)

When Gaussian class conditional distributions are used, forming a classifier is straightforward. However, when class conditional distributions are not Gaussian, calculating the $p(\mathbf{x}|C_i)$ might be very complex or even impossible. In a naïve Bayes method, we assume that features are conditionally independent given the class. However, this assumption is not always valid, e.g. different features extracted from
the accelerometer can be highly correlated. Nevertheless, when this assumption is valid, we can write

\[ p(x|C_i) = \prod_{i=1}^{N} p(x_i|C_i). \] (2.16)

To evaluate (2.16) continuous distributions can be used directly or if \( x \) is composed of discrete variables, \( p(x_i|C_i) \) can be estimated as the frequency that \( x_i \) takes on particular value for particular class [23]. In addition, even if \( x \) is comprised of continuous values, one common way is to discretize them, e.g., using a histogram or kernel estimator.

### 2.7.2 Other Classifiers

Besides the Bayesian classifier, there exist many different methods which can offer more accurate results in some applications. Some of those are introduced here briefly.

#### 2.7.2.1 Support Vector Machines

Support vector machines rely on the principle of linear classifier, illustrated in Fig. 2.5, where the discriminant function is

\[ \zeta_i(x) = w_i^T x + w_{i0}, \] (2.17)

where \( w_i \) is the weight vector and \( w_{i0} \) is the bias or threshold weight [30]. The weight and threshold values are usually obtained using predefined training data and some learning algorithm. However, in the SVM the data is mapped to a higher dimensional feature space which is usually much higher than the original feature space. With appropriate nonlinear mapping two classes are always separable from each other with an appropriate hyperplane. Indeed, the the basic SVM is directly applicable only to two class tasks. Nevertheless, classifiers such as one-versus-all or one-versus-one have been developed. If we assume that we have \( N \) different classes then, in the one-versus-all approach \( N \) different classifiers are trained. For a class \( C_i \) we will assume the labels to be positive and the rest of the classes to be negative. Then the class is selected based on the classifier which classifies the test data with the greatest margin, i.e. the distance of features from the separating hyperplane is largest. In the one-versus-one classification, we need to make classifiers for the each different pair of classes, i.e. in total of \( \frac{N(N-1)}{2} \) classifiers. Final classification is then performed, e.g., by calculating the maximum number of wins from each pair.
The main drawback of SVM is high computational burden due to required constrained optimization programming. Thus, there have been developed more lightweight solutions, called least squares support vector machine LS-SVM. The key difference between SVM and LS-SVM is that inequality constraints are replaced by equality constraints and LS-SVM uses a quadratic error criterion [59, 120].

### 2.7.2.2 Minimum-distance classifier

Minimum-distance classifier is a special case of a linear classifier, where the sample feature vector is assigned to the class associated with the nearest point. Euclidean distance can be used as distance measure [121].

### 2.7.2.3 k-Nearest Neighbor

In k-Nearest Neighbor (k-NN) classifier, a sample feature vector is assigned to the class that is the most frequently represented among the $k$-nearest samples in the training set. The selection of the number of neighbors $k$ is a very important issue that can affect the decision made by the k-NN classifier. However, there does not exist a pre-defined rule for the selection of the value of $k$ [7, 57, 121].

### 2.7.2.4 Decision Trees

A decision tree (DT) is an intuitive classifier where the algorithm is produced through splitting the data with a sequence of questions. The tree starts with a root node and splitting can be done based on the answers, such as *yes/no* or *true/false*, based on the questions in the each node. Splitting can also be
multidimensional; then we select value or property from the given set of values.
More detailed information about decision trees can be found, e.g., from [30].

2.7.2.5 Dynamic Time Warping

Dynamic time warping is an algorithm which measures similarity between two sequences that may vary in time or speed. The sequences are warped non-linearly in the time dimension to determine a measure of their similarity independent of certain nonlinear variations in the time dimension [7, 84].

2.7.2.6 Artificial neural networks

Artificial neural networks (ANNs) consist of an input layer, one or more hidden layers to extract more meaningful features, and a single output layer. Each of those are composed of a number of units called neurons. The model of each neuron includes a smooth nonlinearity, called the activation function [7].

2.7.3 Learning and Adaptation

Typically, classification is based on collected data that have been used to train the classifier. More advanced classifiers are able also to learn during the process, e.g., adapt more to an individual user’s behavior. In supervised learning, the classifier parameters are calculated from a labeled training data set. For example in the Gaussian MAP case, means \( \mu_i \) and covariances \( \Sigma_i \) are calculated using a training data set.

Unsupervised learning is carried out with the aid of natural groupings (i.e. clusters) of the data. Certainly, the definition of natural groups varies depending on the features used and the application. In addition, different algorithms may end up clustering the results differently. One method to adapt the classifier, e.g. to adapt to the user’s individual behavior, is to compute a tentative category label for the present input feature and then use that target category to improve the classification. Another method is to use reinforcement learning where we only tell to the classifier whether the result was right or wrong, but not the actual class label [30].
3 Positioning with GNSS and Sensors

This chapter presents methods for positioning using sensors and GNSS. The methods explored are altitude aiding using barometer and positioning with two satellites. In traditional navigation sensor fusion where inertial sensors and GNSS device is used, inertial sensors provide continuous solution with high sampling rate sensors which is corrected by more sparse GNSS measurements. Fusion is usually done using extended Kalman filter which provides the corrections to the inertial sensors. In this case, the system is more GNSS aided sensor positioning system. However, in the mobile devices, the energy consumption is usually one of the key factors and the Kalman filter combining GNSS and inertial sensors cannot always run continuously. In addition, when for example, barometric altimeter and GNSS devices are used, the main means of obtaining the position is GNSS where as barometer is not able to provide 2D of 3D position. Thus in this case, we can talk of sensor aided GNSS positioning systems. In this chapter, the emphasis is on these kind of systems. Selection of useful sensors and the integration method depends highly on the application and its requirements. For example vehicles with totally autonomous control have higher requirements for position solution accuracy than consumer grade navigators [86]. In addition, autonomous vehicles are designed for navigation and may have large number high-end sensors with tight integrity control whereas consumer navigation devices cannot be designed for one specific purpose.

3.1 Altitude Aiding

As was discussed in Section 2.3.3 barometric pressure can be converted to altitude readings and, therefore, barometric altimeters can be used for altitude aiding in many applications. However, because of temporal variations in the ambient
air pressure, barometric altimeters must be calibrated frequently. For example, the mean air pressure at a certain location can vary by up to 25 hPa between winter and summer, which would correspond to an altitude estimate error of 200 meters [92]; in a shorter time scale, the altitude estimate of a stationary barometer can change by dozens of meters in 24 hours [54]. On the other hand, in the in very short term, barometers can be very precise, being able to detect altitude differences even of centimeters [114].

Before the era of satellite navigation, the use of barometers to detect the altitude of an aircraft was known for many decades. In the beginning of the satellite navigation era, one of the main advantages using barometer assistance, was to improve GPS accuracy which was poor at that time due to selective availability (SA) [9]. One of the first papers discussing barometers in differential mode, i.e. using two barometers in different locations, where altitude of other is known, was presented by McLellean at al. [77] in 1994 where a positioning system using differential barometry with GPS for seismic survey operations was presented. To provide accurate height information using barometers, pressure data obtained by a number of base barometers at points with known height were used to continuously model local ambient atmospheric changes. The paper also showed results where the system was implemented in a helicopter. They report that better than 2 meter height accuracy at 2 $\sigma$ in mountainous terrain.

Positioning systems including a barometer have frequently been used in aviation. Height differences in aviation are usually very large and the absolute aircraft’s altitude is not as crucial as the height difference between different aircrafts. Thus, the pressure values of standard mean sea level are usually selected for calibration of barometers [41] in aircrafts. This ensures that the height difference between two aircraft is known regardless of weather conditions. In addition, barometers in the aircraft are mounted so that they measure only the outdoor pressure, i.e., the pressure inside the aircraft does not have any effect.

However, the smaller size and lower cost of current mobile devices has also enabled their use in personal positioning. They can be found, e.g., in mobile phones, tablets, and sport watches. In personal positioning applications, the error sources differ from those in aviation as the user may carry the device including the barometer in various pressurized environments, e.g., indoors and vehicles. In some applications, height measurements with absolute values are needed, in some others, relative measurements are sufficient. For example, a common application is floor detection in indoor positioning [24]. In this case, the absolute altitude is needed. However, absolute altitude information does not necessarily need to
3.1. Altitude Aiding

come from a barometric altimeter, but can be a result of a fusion algorithm combining, e.g., building plans, WLAN and barometer. For example, Sagawa et al. proposed in 1998 [101] a system for classifying human moving patterns using a barometer and accelerometer. The paper illustrated how different floors of a building can be identified using a barometric altimeter. The main objective in using the barometer in the paper was to infer movements in the vertical direction, e.g. walking upstairs or downstairs or taking an elevator. In addition, in [28] it was shown that a barometer can indeed be used for floor detection in pedestrian dead-reckoning-based indoor navigation and a concept for indoor navigation including barometric aiding was presented in [95]. Reference [83] discusses the possibility of improving positioning accuracy in multistory buildings with wireless sensors networks. The paper shows that the pressure data from mobile devices and base stations in local positioning systems can turn 2D positioning of an object into 3D localization. Moreover, these additional barometric data reduce the number of ranging operations and thus may save radio bandwidth. However, possible barometric error sources were not taken into account in these publications.

In personal navigation devices, a barometer is not only helpful indoors, but can also improve accuracy outdoors, especially when there are challenging conditions for GNSS, e.g., the user is walking in an urban area with very tall buildings. For example, [54] proposed a method to improve assisted-GNSS (AGNSS) performance in terms of vertical positioning accuracy and reliability by using barometric altimeter measurements for the GNSS positioning solution. An AGNSS receiver was integrated with a MEMS air pressure sensor. The reference pressure information was transmitted via a TCP/IP network in order to cope with local changes in pressure. Recently many researchers have combined both GNSS and low-cost MEMS IMU with a barometer in order to conduct seamless navigation indoors and outdoors.

Barometric altimeters also help in vehicle navigation. Especially, when a barometer is integrated with GNSS/INS it can reduce error growth in the vertical channel [127]. Nevertheless, besides pure navigation applications, a barometer can help in road grade estimation [89], i.e. estimating the road tilt based on the altitude change. There also exist applications for different types of sports, such as skiing and roller skating [129], medical applications, such as fall detection [22]. For example, the authors of [15] proposed that the measured pressure altitude might help improving accelerometry-based fall detection systems. In addition, differential barometers in medical application for high precision altitude sensing have been introduced in [17].
There has been little research on the errors of a barometric altimeter. Mainly the error sources are limited to temporal variations in the barometric data. For example, [13] showed that typical barometric errors like thermal divergences and fluctuations in the pressure altitude due to changing weather conditions can be compensated by topographic map information and a barometric error Kalman filter. In the Kalman filter, authors compare the current user position with the horizontal positions in a topographic database. In addition, there has been research on the modeling of the barometric noise properties in short intervals [100]. In contrast to stochastic errors, [71] found out that there was a latency of several seconds when barometric altimeter of mobile phones were tested in a fast elevator.

As discussed above, there has not been research in the prior art on how the error sources in a barometric altimeter affect personal positioning indoors and in vehicle use. Thus, in this thesis, in publication II, MEMS barometer data were collected in several scenarios where different disturbances affect the pressure reading and thus have an effect on altitude estimation. To separate error sources, a reference barometer at a known altitude was used. The results show that barometers in differential mode provide a highly accurate altitude solution, within tens of centimeters, but local disturbances in pressure need to be taken into account in the application design. Publication II shows that heating, ventilating, and air conditioning (HVAC) of the car have a large effect when the barometer is used as an altimeter for accurate meter level positioning or navigation.

### 3.2 Sub Four Satellite Positioning

Even though more satellite navigation systems are emerging, there exists situations when only a few satellite signals are available. For example, in urban canyons, line of sight visibility to the navigation satellites is masked by tall buildings. Thus, it could be useful to be able to calculate position by using only few navigation satellites. As already discussed in Section 2.2.1, at least four satellites are needed to form a position solution for conventional GNSS positioning. However, there are methods that allow positioning with less than four satellites. Sub four satellite methods can be pure GNSS based or they can use some external information such as altitude aiding or approximate position from other sources. The methods can use either only pseudorange based measurements or both pseudorange and Doppler measurements. In this thesis, we try to use a minimal number of GNSS signals with altitude aiding. Even if we use both pseudorange and Doppler equations
with altitude aiding, producing position only by using one satellite would still need some additional aiding information. Thus, the emphasis in this thesis is in two satellite positioning. Nevertheless, it should be noted that only a single satellite can sometimes bring useful information. For example, if the position of the receiver is known, clock bias and clock drift can be calculated.

The idea of resolving the user position by measuring the time and frequency of arrival of a signal from two sources is well known in the literature and sometimes referred to as ‘dual-satellite geolocation’ [25, 90, 118]. For example, Chestnut [25] mentioned in 1982 a method of determining the location of a terrestrial radio emitter using two receivers at known locations; the idea was not, however, discussed in more detail. Nowadays, with GNSS being the most widely used means of radiolocation, the same method is still valid: a GNSS receiver measures both the time of arrival (pseudorange) and frequency of arrival (Doppler shift) of the satellite signals. For the emitter positioning problem, Ho and Chan [45] present in 1997 a closed-form algebraic method using a spherical-Earth coordinate system with an iterative correction scheme to adapt to the ellipsoidal model, whereas Mason [75, 76] has derived direct algebraic algorithms for an ellipsoidal Earth. All these solutions assume that emitter is on the surface of the Earth.

Above solutions can also be applied in GNSS positioning. However, compared to normal four or above satellite position solution, it must be assumed that receiver is not moving or otherwise the solution is inaccurate due to bias introduced in Doppler equations [122]. Unlike in the common four satellite pseudorange positioning, in two satellite pseudorange/Doppler positioning there exists two distinct locations that will fulfill the measurements. When the altitude constraint is used, two satellite solution produces both solutions to be on the surface on the Earth [26]. Thus, the final solution needs to be sought with other means. Closed-form algebraic methods [45, 75, 76] are capable of finding both the solution candidates. Pattison and Chou [90] consider the scenario where a reference receiver is located near the user and propose a solution based on linearization of the measured time difference of arrival (TDOA) and frequency difference of arrival (FDOA) at the location of the reference receiver. Moreover, they present a differential calibration technique for error mitigation. This uses an analogy to the practice of differential GPS. By applying differential corrections they can correct the inaccurately estimated altitude constraint. However, they do not address the ambiguity of the solution. Furthermore, in personal positioning, a reference receiver is not necessarily available for calibration, especially in real-time applications.
There are also methods which use only pseudorange. For example, [72, 113] present a pseudorange-only two-satellite positioning algorithm whose key idea is to use measurements from two consecutive epochs with the receiver standing still. Accuracy of the position solution varies between tens of meters to kilometer level. As in the all two satellite solutions, the geometry of satellites plays important role in the accuracy of the position solution. The methods rely on the assumption that the receiver clock bias remains constant between the two epochs; however, this is a rather strong assumption particularly with low-cost equipment, where also the clock drift plays important role.

As discussed, closed-form algebraic methods [45, 75, 76] are capable of finding both the solution candidates, as opposed to the iterative approach proposed in [90] where each solution must be sought separately. On the other hand, these closed-form algorithms are rather complex, requiring, e.g., solving the roots of a high-order (6th or 8th order) polynomial with complicated coefficients or finding 35 eigenvectors for a generalized eigenvalue problem. In addition, iterative method in [90] needs differential corrections in order to estimate the drift terms.

As the previous methods do not explicitly solve the receiver clock drift, which may cause large errors in low-cost devices, in this thesis we propose a method that solves explicitly receiver clock errors instead of dealing with differenced pseudorange and Doppler measurements. This increases the computational complexity of the system slightly compared to [90] as there are now five nonlinear equations instead of three, but computing a four-satellite position fix using pseudoranges is not significantly less complex than the proposed approach. Thus, solving the system of five equations is not a large computational burden for a modern GNSS receiver, and on the other hand, it is more straightforward to implement than closed-form solvers.

In this thesis in publication I, a method is proposed for two satellite positioning which can be implemented on low-cost devices, as it can also implicitly calculate the clock drift. Given simultaneous pseudorange and Doppler measurements from two satellites and an altitude constraint, positioning equations for the five unknowns (3-dimensional position, clock bias and clock drift) can be solved. In addition, the sensitivity to errors of altitude constraint can be estimated.

In the proposed method, the assisting altitude information obtained from low-cost MEMS sensors or digital elevation data base is integrated into a GNSS receiver. The method to solve the position problem with five unknowns, proposed in this thesis, in publication I, is iterative Gauss–Newton method, which can be used
for pure pseudorange based positioning as well; the algorithm does not require complex closed-form algorithms to be implemented in the receiver.
4 Context Recognition and Positioning

Besides positioning, sensor and GNSS systems can be used in other applications. One very active field is context recognition where the information from the sensors is used to determine some physical context [50], e.g., the user is running or walking or even that the user is in a meeting or shopping. For navigation algorithms, knowing the context information can be also very useful [42, 128]. For example, navigation algorithms should adapt according to the motion mode; there are usually different algorithms for walking and vehicle use or whether the user is indoor or outdoor. In addition, in mobile device based pedestrian dead reckoning, the step detection algorithms should adapt to how the device is carried. Also, one application that is discussed in this thesis in Section 4.2 is vehicle crash detection.

One of the first papers introducing context awareness was published by Weiser in 1991, [123], where he discussed about ubiquitous computer and how they would vanish into the background. Modern mobile devices with wireless networks and cloud based data processing has made his thinking partly true and context aware computing is now an active research area.

As context awareness is still quite a young field of computer science, the terms context and context awareness or context aware systems do not have strict definitions yet and their interpretation varies depending on the application. One definition of context is “information that can be used to characterise the situation of an entity” by Dey in 2001, [29], which is currently the most commonly used definition [73]. Continuing, the context awareness can be defined as “a state of the device that is aware of circumstances and adapts itself and responds suitably” [60].

31
4.1 Mobile Context Awareness

A large driver of the recent progress in context awareness research is the increased use of personal mobile devices. Thus, there has arisen a definition of *mobile context awareness*, where the sensing, computing and reaction happens in the device itself. Thus it is both a sensing platform and a computing unit. As the mobile devices are battery powered, the energy constraints of these portable handsets make the task of sensing very challenging. Usually the main challenge is to accurately recognize the required context with a minimum number of sensors and a low sensing frequency [47]. From the beginning of the first smartphones (e.g. Nokia N95 in 2006 and iPhone in 2007), the research has expanded rapidly. Sensing with mobile phones is nowadays an active research area. Almost every high end mobile device has currently 3 accelerometers, 3 gyros, and 3 magnetometers.

Based on the user involvement in the sensing process, the context aware applications can be divided to *participatory sensing*, where the user is actively participating in the sensing process, and *opportunistic sensing*, where the user remains passive and is not required to participate. Nevertheless, opportunistic sensing can be perceived to be more user friendly as it does not annoy the user with additional questions or calibration processes and thus more recent progress has been made towards solving the key challenges in opportunistic sensing systems [47]. However, the end applications define what kind of sensing system can be used and opportunistic sensing is not always possible. An example application of participatory sensing is as a health and wellness monitoring where individuals can self-monitor to observe and adjust their medication, physical activity, nutrition, and interactions [34]. In this kind of applications, the user feedback is essential. In contrast, for example in motion mode detection for navigation purposes, e.g. is user walking or driving, opportunistic algorithms are preferred as extra questions will only annoy the users. In participatory sensing, complex operations can be supported with the of the user, which can significantly reduce the sensing, calibration, and classification challenges when compared to an opportunistic approach.

Many of the first context aware systems developed in mobile devices were only using location based services (LBS) and thus being location aware. However, location awareness is one element of context awareness which has pioneered context aware services [23]. Recent advances in sensor technology in modern smart phones sensors, such as accelerometers, gyroscopes, light sensors, and multiple radio sensors in combination in growing processing power has enabled their wider use for mobile context research and development [73]. Some of the current mobile
devices even have co-processors, such as the Apple M7 [1], which are specifically designed to collect sensor data from integrated sensors in order to offload the collecting and processing of sensor data from the main central processing unit (CPU).

The existing literature on this topic includes many interesting examples of what can be done with the sensors installed in mobile devices [49, 65]. In addition, closely related applications are context recognition done with wearable sensors, i.e. the sensors are integrated in a jacket or worn in a wrist or on a shoe. Examples of different sensed activities and environments are presented in Table 4.1.

Currently, smart phones are integrated with a rich set of sensors, such as accelerometer, gyroscope, GPS, microphone, camera, proximity and light sensors, Wi-Fi, and Bluetooth interfaces [49]. However, as the majority of those were not present in the mobile devices in the beginning of 21st century, the first context aware systems relied on the radio sensors such as bluetooth and GSM (Global System for Mobile communications) signal strength [8, 31, 62]. For example, Eagle and Pentland [31] introduced in 2006 a system using Bluetooth-enabled phones to infer social interactions between phone users and Anderson et al. [8] developed in

<table>
<thead>
<tr>
<th>Context class</th>
<th>Activity or environment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Locomotion [52, 57, 80, 91, 107]</td>
<td>Walking, running, sitting, standing, still, lying</td>
</tr>
<tr>
<td>Mode of transport [52, 97]</td>
<td>Cycling, traveling with a vehicle, riding a bus, driving</td>
</tr>
<tr>
<td>Exercise [32, 57]</td>
<td>Outdoor bicycling, soccer playing, biking on a fitness bike, rowing</td>
</tr>
<tr>
<td>Daily activities [32, 91]</td>
<td>Shopping, using computer, sleeping, going to work, going back home, being home, working, lunch, dinner, breakfast, in a conversation, attending a meeting, phoning</td>
</tr>
<tr>
<td>Health related activities [15, 117]</td>
<td>Falls, rehabilitation activities, following routines</td>
</tr>
<tr>
<td>Environmental context [14, 42, 80]</td>
<td>Indoor, outdoor</td>
</tr>
</tbody>
</table>
2007 a prototype application that tracks the daily exercise activities of the user.

The microphone can also be used for context recognition purposes by collecting audio during daily events and detecting contexts like being in a conversation, road, bus, restaurant and a noisy environment [33, 79]. In addition, the camera can be used to obtain context information. An example application, EyePhone [81], uses the camera to detect activities, such as tracking the eye movements of a user to start the applications on the phone.

GNSS is a powerful sensor for tracking the location of the mobile device, but it can also bring useful features to context recognition algorithms. For example, the speed information from GNSS can be used to recognize transportation modes [74, 97]. In addition, the availability of GPS signals can determine whether the device is indoors or outdoors [102, 103]. Or more specifically, as was shown in the work by Groves et al. [42], GNSS carrier to noise (C/N₀) measurements can be used to distinguish indoor from outdoor environments.

Accelerometers are some of the most commonly used sensors for mobile context awareness. They are also some of the most effective sensors when we are trying to infer the activity of the mobile user [49]. Although it was integrated in the mobile phone with the objective of enhancing the user experience by changing the orientation of the display according to the orientation of the phone held by the user [49, 65], it can be used for more complex sensing such as is the user walking, sitting or running [37, 52, 97]. For example, Reddy et al. [97] presented a classification system that uses a mobile phone with a built-in GPS receiver and an accelerometer. The system recognizes the following transportation modes: stationary, walking, running, biking, and motorized transport.

In addition, gyroscopes and magnetometers are useful sensors for context recognition [91, 107]. For example, Shoaib et al. [107] used a smartphone’s gyroscopes and magnetometers with commonly used accelerometers in activity recognition. The paper indicated that the results for the magnetometers are not encouraging because they cause over-fitting in training classifiers due to their dependence on direction. In addition, paper also states that in most cases, gyroscopes not only improve the recognition accuracy in combination with an accelerometer, but they also achieve a reasonable performance when used alone. However, even though magnetometers were not reported to be useful in action recognition they can bring useful information about, for example, the carrying mode of the phone [111]. A barometer which is mainly integrated in some high end mobile devices can be used, e.g., in fall detection [15]. Proximity and light sensors are also sensors that
are included in the mobile devices. For context recognition purposes, they can be used together with other sensors to infer more accurate activity information. For instance, the light sensor may provide information about the environment of the user, such as being in a dark environment [49]. In addition, the paper [80] reports an indoor or outdoor classifier which is implemented using external temperature, light, and humidity sensors.

The algorithms used in context classification originate from statistical machine learning techniques. However, a trendy algorithm in machine learning research may not exhibit a superior performance in the field of context recognition, especially when we are using mobile phone platforms with limited resources and limited processing power and the limited battery capacity [49]. For example, Anderson et al. [8] developed in 2007 a prototype application that tracks the daily exercise activities of people, using an Artificial Neural Network (ANN) to analyze GSM cell signal strength and visibility to estimate movement of a user. In addition, decision trees are used when working on a multitude of sensors with limited computing power. For example, in the User-Programmable Context-Aware Services (UPCASE) project [102, 103], multiple sensors, such as accelerometers, light, sound, humidity, temperature and GPS sensors, were connected to a mobile phone via Bluetooth. Decision trees were used to obtain the activity (walking, running, standing or lying) of the user. The system also detected whether the device is indoors or outdoors by using information on the availability of GPS signals.

As discussed in Section 2.7, the features in context recognition should ideally be selected so that classification is trivial. Thus, the feature selection and extraction must be done carefully. For example, Könönen et al. described several classification and automatic feature selection algorithms which are compared in the problem of context recognition [57]. The paper studied how sophisticated and complex classification methods compared with a simple method that can easily be implemented in mobile devices. The paper showed that even a simple linear classification algorithm, namely the k-NN, can achieve almost the same level of accuracy as more complex algorithm, namely SVM, if the features calculated from raw data are selected in a suitable way. In addition, they reported superior performance for the Minimum-distance classifier compared to other classifiers from the view point of computational load and power consumption of a smart phone.

In addition, Altun et al. presented a study that classified human activities using body-worn inertial and magnetic sensors [7]. The authors performed a comparative study of different methods, including, Bayesian classifier, DT, k-NN, DTW, SVM, and ANN. The results in the paper indicated that in general, Bayesian classifier
gives the highest correct classification rate with a relatively small computational cost.

A related work in the mobile context awareness within the scope of this thesis using a foot mounted IMU with GPS to detect the activity and environment of the user was presented by Bancroft et al. in [10]. This paper presented an algorithm that classifies environment: indoor or outdoor and activity: stationary, crawling, walking, running, cycling, or moving in vehicle, and vertical movement: level, up or down in an elevator and up or down on stairs. The paper used a naïve Bayes algorithm to infer different activities and environments. As the measurement device was mounted on the foot the results are not directly comparable to those systems implemented purely with mobile device sensors.

In [91], Pei at al. use location and motion tracking to build a context aware system. In the system, positioning technologies and smartphone sensors are used to capture human movements in natural environments and use the movement information to study human behavior. They are inferring different activities such as sitting, standing, walking at different speeds, and turning through different angles. Features from accelerometers, gyroscopes and magnetometers were used and Least Square-Support Vector Machines (LS-SVM) were utilized as classifiers. In the paper, they also infer different contexts using location history, such as fetching papers from a printer or being in a coffee break.

As is shown in the literature (e.g. [7, 10]) the Bayes classifiers can achieve good accuracy in mobile context recognition with a low computational cost. However, in many cases the features obtained from the mobile devices may be user dependent. For example, people tend to have different walking styles [61, 67], which will produce varying feature vectors and thus it will affect to the final sensing result of the activity recognition algorithm. In addition, when radio signals are used for environmental detection, there might be device dependent differences in the levels of the measured signals [68]. This will also have effect on the final decision of environment context algorithms.

As the previous work lacks a smooth user adaptation mechanism, in this thesis in publication III, activity and environment recognition from mobile phone sensor and radio receiver data is developed. By environment recognition we mean the automatic recognition of user surroundings, such as whether the user is in a meeting, at the office, or inside a vehicle. An example of the environment feature space is given in Fig. 4.1. More specifically, our novel contribution in publication III is that we introduce an algorithm for user specific adaptation of
the context model parameters from binary feedback. This is done by moving the mean parameter and scaling the covariance of the Bayesian maximum a posteriori classifier based on user feedback indication whether the classification result was correct or not. Furthermore, the adaptation method is able to provide a confidence measure about the correctness of the classification for the host application.

4.2 Automatic Crash Detection

An application that can be categorized as a context aware system is the European eCall initiative where in the case of a crash, a vehicle automatically calls to the nearest emergency center and sends some data including exact crash location [4]. The eCall architecture uses the GSM cellular network to communicate between the vehicle involved in an incident and the Public Service Answering Point (PSAP). The rationale for this is that when a traffic accident takes place, correct assistance can be sent to injured passengers as soon as possible. For example, it has been shown that reducing the accident response time by 1 min correlates to a six percent difference in the number of lives saved [35].

In addition, the US Department of Transportation (DOT) envisions a future wherein systems will automatically detect and characterize incidents using wireless networked communications with equipment installed in the vehicles and personal communication devices of the travellers. The Connected Vehicle Program launched by the DOT aims to achieve the required sensor technology, and infrastructure for the US transportation system [39].
The majority of the academic literature discusses crash detection methods for cars, e.g., [20, 93, 94, 108, 109]. Crash detection algorithms are usually based on acceleration or a dedicated crush sensor [124]. Then, the severity of the crash can be estimated using the magnitude of acceleration or crush peak. Naturally, a larger peak may imply a more severe crash.

Commercial automated emergency call systems for accident detection, such as BMW’s Automatic Crash Notification System, General Motor’s OnStar [5], Peugeot-Citroën’s Appel d’Urgence [21], WirelessCar [6] and Connexis [2], notify emergency responders immediately by utilizing built-in cellular radios and detect car accidents with in-vehicle sensors, such as accelerometers and airbag deployment monitors. They also use a GPS receiver for notifying the position of the vehicle [96].

Besides commercial products, there have been scientific projects where devices from third parties are installed on vehicles’ On-Board Diagnostics (OBD-II) bus to conduct automatic crash detection and handle communication. An example of this kind of system is the e-NOTIFY project [38], where the automatic crash detection and communication is done by installing an external onBoard unit on the cars. A data acquisition unit reads the data from the sensors installed in the vehicle and passes the information to the OnBoard communication unit which has access to the wireless infrastructure via a GSM channel. In order to get position information, the OnBoard unit communicates also with an external GPS unit.

Another system using OBD-II in combination with a mobile phone is presented in [126]. The proposed method uses an Android based application that monitors the vehicle through an OBD-II interface, enabling it to detect accidents. The method uses the acceleration experienced by the passengers in case of a frontal collision, which is used together with airbag triggers to detect accidents. If the acceleration exceeds a predetermined threshold then the application will send an e-mail or SMS to pre-defined destinations, such as an emergency center, immediately followed by an automatic phone call.

Whereas the systems proposed in [38] and [126] are based on the built-in sensors of the vehicle, a pure smart phone based system is presented in [115, 125]. The proposed system, Wreck Watch, is an automatic accident detection system for smartphones, which includes a client developed for Google Android and a server. Accidents are inferred using smartphone accelerometers and acoustic data. If needed, a warning message is passed to the Wreck Watch server where data is analyzed and a message is passed to the emergency responders. Nevertheless, the
false alarm rate can be high as the measurements are obtained only from the smartphone sensors. For example, if the phone fell down, a false message could be generated. In addition, if the smartphone gets damaged due to the accident there would not be any message passed as the system uses the smartphone’s data connectivity.

A crash notification system for the cars that was implemented on handheld and aftermarket devices and is compliant with the future European eCall standard was presented in [93]. The system consists of a crash detector, which is implemented using an accelerometer and which can be connected over a wired or wireless link to an eCall box. This box provides the notification to the eCall Public Service Answering Point. The paper proposes that a sensor detecting the crash could be installed below the seat on a motorbike or in the helmet of the motorist.

An automatic accident reporting apparatus for a two-wheel vehicle is presented in [112]. It includes an accident detecting module and an accident informing module. The accident is inferred using a built-in inclination sensor and the speedometer, instead of using a conventionally-used acceleration sensor to measure the crash pulse. A simple decision tree type of system is then used to detect the accidents. When the inclination remains higher than a predetermined level throughout a specific length of time and the speed history of the vehicle meets predefined thresholds, an accident is reported. Instead of built-in sensors, [11] proposes a crash detection system comprising a portable device including the sensors and mobile phone. The measurement device is installed in a vehicle and it communicates with a mobile phone or any connection module. The measurement devices are comprised of inertial sensors which are then connected to the connection module.

A conventional crash pulse detection is problematic from the perspective of motorcyclists. If the crash is alerted using a vehicle mounted accelerometer, the severity of the crash to the motorcyclist is not necessarily known. For example, when a motorist falls from a motorcycle during the crash, only the motorcycle or the motorist may undergo high energy crash. In the prior art [3, 11, 56, 93, 112], the crash detection is carried out with the aid of a single sensor or only sensors attached to the motor cycle. Thus, in this thesis in publication IV, a concept for automated crash detection for motorcycles is proposed. In this concept, three different inertial measurement units are attached to the helmet of the motorist, to the torso of the motorist, and to the rear of the motor cycle. A maximum a posteriori classifier is trained to classify a crash and normal driving, using data from crash dummy throwing tests. In addition, the proposed approach also detects the free fall state (i.e., when the motorist is not touching the ground), which may
give valuable information about the severity of the crash.
5 Conclusions

In this chapter, conclusions regarding the work presented in this thesis are drawn. We first summarize the methods presented in this thesis, then discuss them and finish with suggestions for future work.

5.1 Summary of Contributions

In publication I, a method for computing a position fix using two GNSS satellites and altitude aiding is presented. To compute a position solution, both pseudorange and Doppler measurements were needed from both the satellites. The system is required to be stationary during the process and a source of altitude information, e.g., a MEMS barometer, is needed in addition to a basic GPS receiver, and it can be replaced by an estimate of the altitude if the method is used for purposes where the less accurate positioning solution is needed, e.g., as a position estimate for further satellite acquisition. Authentic pseudorange and Doppler measurements with simulated altitude were used to test the proposed method. The results showed that normally an accuracy of about 2 kilometers is achieved. However, it should be noted that the noise in a Doppler measurement can induce an error thousands of times larger than an typical magnitude noise component in a pseudorange measurement. In addition, the altitude measurement error may also cause a significant horizontal position bias. The geometry of the satellites has also a drastic impact on the accuracy of the position solution. However even an inaccurate initial position solution can be very important, e.g., in the case of emergency calls, if it can be obtained quickly, provided that the inaccuracy is acknowledged and can be accepted.

In publication II, MEMS barometer data is collected in several scenarios where different disturbances, such as opening the window or driving in a tunnel, affect the pressure reading and thus have an effect on altitude estimation. To separate these error sources from the varying atmospheric pressure due to different weather
conditions, a reference barometer at a known altitude is used. The results show that barometers in differential mode provide a highly accurate altitude solution (within tens of centimeters). However, local disturbances in pressure can still cause degradation in accuracy in altitude estimation. The paper shows, for example, that heating, ventilating, and air conditioning (HVAC) of a car can have effect of few meters. Thus this could cause problems if the barometer is used as an altimeter for under meter-level positioning or navigation.

In publication III, activity and environment recognition from mobile phone sensor and radio receiver data is investigated. The purpose of activity recognition algorithms was to output information on the activity of the user. In particular, we investigated algorithms which classified the physical activity of the user, such as walking, running, driving a car, riding a bicycle, or being still. By environment recognition we meant the automatic recognition of user surroundings, such as the user being in a meeting, at the office, or inside a vehicle. The publication introduced an algorithm for user specific adaptation of the context model parameters using the feedback from the user. Furthermore, the adaptation method can provide a confidence measure about the correctness of the classification for the host application. A real-life data collection campaign was arranged to test the methods proposed in the publication. The result showed that for the activities the estimated confidence measure is consistent. However, for the environments the confidence measure was slightly optimistic.

Publication IV presents a concept for automated crash detection for motorcycles. In this concept, three different inertial measurement units are attached to head of the motorist, torso of the motorist, and to the rear of the motor cycle. A maximum a posteriori classifier is trained to classify the crash and normal driving. Crash dummy tests were done by throwing a dummy from different heights to simulate the effect of crash on the motorist and real data was collected by driving the motorcycle. Preliminary results showed that the classifier could be applicable in real situations.

5.2 Discussion

This thesis showed that GNSS-aided sensor systems can be used to give more information of the user’s context, e.g., what user is doing, in addition to positioning. The role of context is very often bidirectional - knowing the context can help the positioning system, but also the positioning system can help in determining the context.
The results in the thesis show that a low-cost MEMS barometer can be used as an aiding source for multiple applications when the local disturbances are taken into account. In addition, different disturbance models are needed for different applications. Thus, it would also be helpful to know the context where the application is used. For example, there are very different error sources indoors and outdoors or if the user is driving a vehicle. In an indoor environment, ventilation systems may cause meter level differences even between rooms which are next to each other and vehicle HVAC may also cause pressure difference of many meters compared to an outdoor environment. If the environmental context is known, it is possible to acknowledge the types of possible error sources and thus it could even be possible to estimate or model different error sources more accurately.

One possible aiding source in the proposed two satellite positioning algorithm in this thesis is a barometric altimeter. Thus, knowing the context could be important, if other altitude aiding sources, such as terrain based databases are not available. In addition, the context recognition can be used to detect whether the receiver is not moving, which was an assumption in the proposed two satellite algorithm. This kind of activity context information can be obtained using a simple pattern recognition algorithm that uses data from a triad of accelerometers.

Vice versa, satellite based positioning information is often very helpful in context aware systems. Many of the first context aware applications used only the location information [23]. It is still some of the most important information in many systems. For example, in the proposed crash detection system for motorcycles, besides reporting the location, the velocity history provided by a GNSS positioning system can bring crucial information to the system. As the number of sensors has increased dramatically in mobile devices in recent years, the possibilities for different types of context aware applications have also increased.

In general, it can be concluded that in many positioning and navigation applications knowing the context can improve accuracy and reliability of the system. The sensors are becoming cheaper every year and even though the number of GNSS satellites will increase, sensors are needed to carry on positioning when GNSS signals are not available and a large number of different types of sensors are needed for rigorous context recognition applications.

5.3 Future Work

A sensor that had minimal attention in this thesis was the magnetometer. In the prior art, it has shown capabilities for both positioning and context recognition
and it is integrated in to many devices. Thus, as future work, magnetometer data could be used in the proposed context recognition algorithms.

In addition, in order to effectively validate the proposed adaptive algorithm for context recognition presented in the publication IV, more data can be collected from many different users. Also data for the proposed crash detection algorithm from real crash tests could be acquired in order to rigorously validate the proposed method.
Bibliography


Errata for Publications

Publication I

- In page 6, the sentence “the Doppler measurement could have been biased by a half wavelength” should be “Carrier phase measurement could have been biased by a half wavelength, causing error in the Doppler measurement”.

- In page 7, “the number of available geometries” should be “the number of possible two satellite combination geometries”.

Publication II

- In the tests described in Section IV, barometer was attached inside the car on the floor of passenger side. Tests were conducted in Tampere, Finland, in February 2008.

- In eqn. (3), $\cos^{-1}$ means arc cosine and $\Delta s$ is travelled distance, not speed as it was described in the text.

- In last paragraph of Section VI, “resampled to 1Hz rate” should be “resampled to 1Hz rate using moving average filter”.

Publication III

- In page 20754, the sentence “the required accuracy for the recognition is relatively high” should be “accuracy for the recognition is wanted to be as high as possible”.

- In pages 20760, 20761 and 20762, $(k - by - n)$, $(r - by - r)$ and $(q - by - 1)$ can be also expressed as $(k \times n)$, $(r \times r)$ and $(q \times 1)$, respectively.
- In page 20762, \( q \) is the length of feature vector.

- In page 20763, in equation (9) \( F \) is Chi-square cumulative distribution function and its degree of freedom parameter and value are evaluated recursively. \( F \) is not independent of parameter \( k \).

- In page 20764, the distribution parameters to be modified are mean \( \mu \) and scale \( s \).

- In page 20764 (and in the rest of the paper), correct term for equation (13) is minus log likelihood not likelihood. In addition, there should be parentheses around \( s \Sigma \), i.e. equation (13) should be \( L_i = (\mathbf{z} - \mu_i)^T(s_i \Sigma_i)^{-1}(\mathbf{z} - \mu_i) \).

- In page 20764, set A presents the yes answers of which likelihood (minus log likelihood) values are greater than the point where Chi-square distribution with similar parameters (mean and covariance) has a value of 0.95. Similarly, set B presents the no answers of which likelihood (minus log likelihood) values are smaller than the point where Chi-square distribution with similar parameters (mean and covariance) has a value of 0.95.

- In page 20766, in equation (19), \( \sup \) means supremum of the set of distances and \( n \) is the sample size.

- In page 20766, caption for Figure 3 should be “The 95 % ellipses of the simulated distributions”

- In page 20768, the results in Table 4 are using features from GPS, WLAN and Bluetooth.

- In page 20769, in Figures 6 and 7, comp. means compressed.

- In page 20774, \( Bin \) is binomial distribution with two parameters \( n \) and \( p \); the number of successes in a sequence \( n \) of independent yes or no experiments, each of which yields success with probability \( p \).

**Publication IV**

- In the second page, \((q - by - 1)\) can be also expressed as \((q \times 1)\).

- In the second page, by “null hypothesis” we mean the data where the accidents do not occur.
• In TABLE I, measurement 2. and 3. mean that dummy was first standing and then fell on his back or frontally.

• In Figure 4, the caption should be “Estimated states (peak, zerosG, drive, static) in a dummy crash test (upper one) and accelerometer norm (bottom one) during the same test”.

• Data shown in Figure 4. belongs to measurement 1. given in TABLE I.

• In Figure 5, y axis is samples at 1000 Hz.
Publications
Abstract—This article proposes a way of improving the time to first fix (TTFF) in GNSS positioning by combining pseudorange and Doppler positioning with external altitude information. This way, the position of a stationary receiver can be resolved using two satellites only. Test results obtained using authentic satellite data demonstrate that the method is sensitive to Doppler measurement errors but can improve the TTFF significantly if the sky view is constrained as, e.g., in urban canyons.

Index Terms—Doppler positioning, GNSS, time to first fix (TTFF), urban canyon positioning

I. INTRODUCTION

In Global Navigation Satellite System (GNSS) positioning it can take a long time until the receiver has found four satellites, which is the minimum when using traditional positioning algorithms, in the acquisition phase if the sky view is significantly blocked and no previous almanac or ephemeris data are available. Particularly in personal navigation, the time to first fix (TTFF) is often regarded as an essential performance measure. Hence, TTFF could be reduced if less than four satellites would suffice. Moreover, an initial position fix using less satellites, even if fairly inaccurate, would help in finding more satellites to be acquired.

Resolving the position of a GNSS receiver using two satellites and altitude information is theoretically possible [1]. The major obstacles when implementing two-satellite positioning methods are the requirement of immobility and the need of altitude information. In practice, this means that additional sensors are required. Sensors based on Microelectromechanical System (MEMS) technology can be integrated into handheld GNSS receivers and used as a source of the required aiding information. In this paper, using an iterative method [2] for resolving the position of a stationary GNSS receiver based on pseudorange and Doppler shift measurements along with external altitude information is proposed. To obtain the necessary aiding information, integrating a MEMS accelerometer triad and a MEMS barometric altimeter into the receiver is proposed. Such a setup is capable of verifying its stationarity and measuring the altitude, and therefore, can meet the requirements for two-satellite positioning [3]. As an alternative for the altimeter, digital elevation model (DEM) information can be used instead; if the position estimate of the two-satellite method is only used for acquiring other satellites, the receiver can be assumed to be located at, e.g., sea level, which eliminates the need of external height information.

Since one of the measurements used is the altitude, the measurement equations are usually fulfilled at two distinct points at that height. Hence, a means of choosing the correct solution and excluding the false one must be available. In this article, satellite visibility information is used as a discrimination criterion: if any of the two satellites is not visible at a solution candidate point, the candidate is discarded. If the correct solution can be efficiently chosen, the proposed positioning method is capable of providing a significant improvement in the TTFF in adverse signal conditions where it is hard and time-consuming to find satellites. However, the precision of the two-satellite fix is significantly coarser than that of a traditional four-satellite position solution; therefore, after computing the two-satellite fix, more satellites should be acquired to improve the position estimate.

This article is organized as follows. First, prior art is reviewed in Section II. Then, hardware requirements concerning the necessary aiding information are discussed in Section III. Section IV introduces combined pseudorange and Doppler positioning and an iterative algorithm for solving that problem, and addresses the uniqueness of the solution. The sensitivity of the algorithm to different measurement errors is discussed and error estimates are derived in Section V. In Section VI, authentic Global Positioning System (GPS) data are used to show that the method works with low-cost equipment and the presented sensitivity analysis is valid in practice, and finally, Section VII summarizes the article.

II. RELATED WORK

The idea of resolving the user position by measuring the time and frequency of arrival from two sources is well known in the literature and sometimes referred to as ‘dual-satellite geolocation’. For example, Chestnut [4] mentioned the essentially same idea for determining the location of a terrestrial radio emitter using two receivers at known locations; the idea was not, however, discussed in more detail. Nowadays, with GNSS being the major means of radio-location, the same method is still valid: a GNSS receiver measures both the time of arrival (pseudorange) and frequency of arrival (Doppler shift) of the satellite signals.

For the emitter positioning problem, Ho and Chan [1] present a closed-form algebraic method using a spherical-Earth coordinate system with an iterative correction scheme to adapt to the ellipsoidal model, whereas Mason [5], [6] has
derived direct algebraic algorithms for an ellipsoidal Earth. These methods are capable of finding all solution candidates, as opposed to the iterative approach proposed in this paper where each solution must be sought separately. On the other hand, these closed-form algorithms are rather complex, requiring, e.g., solving for the roots of a high-order (6th or 8th order) polynomial with complicated coefficients or finding 35 eigenvectors for a generalized eigenvalue problem.

Pattison and Chou [7] consider the scenario where a reference receiver is located near the user and propose a solution method based on linearization at the location of the reference receiver. Moreover, they present a differential calibration technique for error mitigation. In personal positioning, a reference receiver is not necessarily available for calibration, especially in real-time applications. Furthermore, the paper does not address the ambiguity of the solution.

More recently, Li and Wu [8] proposed a pseudorange-only two-satellite positioning algorithm whose key idea is to use measurements from two consecutive epochs with the receiver standing still. The method relies on the assumption that the receiver clock bias remains constant between the two epochs; however, this is a rather strong assumption particularly with low-cost equipment. Another pseudorange-only two-satellite algorithm was proposed by Tan et al. [9], but their method outputs a line on which the receiver is estimated to lie, instead of a single point estimate. In order to work, the method of [9] requires prior information on the receiver location.

Unlike the previous papers on the topic, the proposed method is targeted to personal positioning requiring implementation with low-cost devices. In the proposed method, the necessary assisting information is obtained from low-cost MEMS sensors which can be easily integrated into a GNSS receiver. In addition, the method presented in this paper is iterative and based on the Gauss–Newton method which can be used for pure pseudorange based positioning as well; thus, the algorithm does not require complex closed-form algorithms to be implemented in the receiver. The proposed method also solves explicitly for receiver clock errors instead of dealing with differenced pseudorange and Doppler measurements. This increases the computational complexity of the system slightly as there are now five nonlinear equations instead of three, but computing a four-satellite position fix using pseudoranges is not significantly less complex than the proposed approach. Thus, solving the system of five equations is not a large computational burden for a modern GNSS receiver, and on the other hand, is more straightforward to implement than closed-form solvers.

III. REQUIRED AIDING INFORMATION

The positioning method described in this paper relies on aiding information for GNSS. Some possible sources of the necessary assisting data are discussed in this section. For altitude information, barometers and Digital Elevation Models (DEM) are considered; for ensuring the stationarity of a mobile receiver, a means of motion detection is also needed. The altitude aiding can be replaced by always assuming that the receiver is located on a certain altitude (e.g., sea level), but this will degrade the positioning accuracy if the assumption does not hold.

A. Motion Detection

A fundamental assumption when using Doppler measurements for positioning is that the velocity of the receiver is known; in practice it is assumed that the receiver is stationary. In personal positioning, this assumption is somewhat problematic if there is no way to verify the immobility. In principle it is possible to determine the velocity of the receiver using GNSS Doppler measurements, but to the best of the knowledge of the authors, no velocity determination method using two satellites only exists. In contrast, using a MEMS accelerometer triad is one potential means: although, in principle, an accelerometer cannot distinguish between rest and constant velocity, a moving body is subject to vibrations that increase the variance of accelerometer measurements [10]. Thus, it can be estimated if the body is moving or not. This information can be used to determine whether the proposed method is applicable in the present situation or not. In Inertial Navigation System implementations, it is a common strategy to try to detect stationary periods in order to mitigate the effect of error accumulation with low-cost inertial measurement units; this process is known as the zero-velocity update.

B. Digital Elevation Models

Another way to obtain altitude information is to utilize a digital elevation model. DEMs are discussed more profoundly in, e.g., [11], [12], but basically, a DEM contains a topographical representation of the surface of the Earth; for example, the GTOPO30 DEM [13] has a horizontal resolution of about one kilometer. As a drawback, a DEM does not know if the receiver is located on the top of a building or some other man-made object; on the other hand, no calibration is needed as opposed to barometers. DEM information can be obtained, e.g., online if network assistance is available. If a DEM is used, the 2-dimensional position must be first obtained using a rough altitude constraint (e.g., assume the user is located at sea level) and then iteratively corrected using the altitude from the DEM in the current 2-dimensional location estimate.

C. Barometer

It is commonly known that air pressure varies depending on the altitude: the higher in altitude one goes, the lower the pressure. Advances in MEMS technology have made it possible to integrate low-cost and small-size barometric altimeters to positioning units. However, air pressure measurements do not give absolute altitude information: the normal air pressure at sea level is approximately 1013 hPa, but it varies locally depending on weather conditions. Near the surface of the Earth, a pressure change of one hPa corresponds approximately to an altitude difference of 8 meters, provided that weather conditions are not changing [14]. More accurately, the altitude $h$ can be related to the ambient pressure $p(h)$ using the International Standard Atmosphere model [15] as

$$h = \frac{T(0)}{k} \left(1 - \frac{p(h)}{p(0)}\right)^{\frac{kR}{\gamma}}$$

(1)
where $R$ is the universal gas constant, $k$ is the temperature lapse rate, $g$ is the gravitational acceleration, and $T(0)$ and $p(0)$ are the sea level temperature and pressure, respectively.

Because of temporal variations in the ambient air pressure, barometric altimeters must be calibrated frequently. For example, the mean air pressure at a certain location can vary by up to 25 hPa between winter and summer, which would correspond to an altitude estimate error of 200 meters [16]; in a shorter time scale, the altitude estimate of a stationary barometer can change by dozens of meters in 24 hours [17]. On the other hand, barometers can be differentially very precise, being able to detect altitude differences even of centimeters [18]. For a more detailed discussion on barometric aiding in personal positioning, see e.g. [17].

The performance of a low-cost barometric altimeter is illustrated in Figure 1 where a MEMS barometer [19] has been used. The solid line shows the barometer data and the dashed line represents a reference altitude measured by a dual-frequency differential GPS (DGPS) receiver capable of a centimeter-level positioning accuracy. The data were measured using normal passenger car with the barometer mounted inside the vehicle. At the beginning of the ride, the barometer was calibrated using altitude information obtained from the DGPS receiver. It can be seen from the figure that the corresponding sea-level air pressure did not change remarkably during the test—otherwise, the barometer measurements would start to drift farther from the DGPS estimates. The DGPS solution, however, is less noisy than the barometer-derived altitude estimate.

IV. COMPUTING THE POSITION SOLUTION

In conventional pseudorange positioning, the position solution is obtained utilizing four or more pseudorange measurements

$$\rho_i = \|s_i - u\| + b + e_i \tag{2}$$

where $\rho_i$ is the measured pseudorange (in units of distance) to satellite $i$, $b$ is the receiver clock bias, $u$ and $s_i$ are the 3-dimensional positions of the receiver and satellite $i$, respectively, and $e_i$ contains all errors such as multipath and noise. Four or more satellites are needed to solve for four unknowns, i.e., the 3-dimensional position and the clock bias. In addition to the pseudorange measurement, most receivers report precise measurements of the pseudorange rate, i.e., Doppler shift. Moreover, incorporating altitude information from a barometric altimeter enables constructing enough positioning equations using only two satellites.

A. Doppler Positioning

The Doppler positioning equations used in this paper are studied in [20]. The Doppler shift can be modeled as the dot product

$$\dot{\rho}_i = v_i \cdot \frac{s_i - u}{\|s_i - u\|} + \dot{b} + e_i \tag{3}$$

where $\dot{\rho}_i$ is the measured pseudorange rate (in units of velocity) to satellite $i$, $v_i$ denotes the relative velocity between the receiver and satellite $i$, and $\dot{b}$ is the receiver clock drift (i.e., time derivative of the receiver clock bias $b$). $e_i$ represents other error sources. Unlike pure pseudorange positioning, Doppler positioning does not require the receiver clock bias to be solved for, but another unknown, the receiver clock drift $\dot{b}$, is introduced. The range rate $\dot{\rho}_i$ is equal to the Doppler shift scaled by the signal wavelength, and is referred to as the Doppler measurement in this paper.

The relative velocity $v_i$ in (3) consists of two parts: the velocity of satellite $i$ and the velocity of the receiver. The former can be obtained using the satellite ephemeris data [21], but the latter is unknown, and typically assumed to be zero. If the receiver, however, is not stationary, three more equations would be needed to account for its velocity. In personal positioning, the requirement of stationarity is fairly strict, but MEMS accelerometers can be used to detect such cases very accurately [3].

B. Combined Doppler and Pseudorange Positioning

Given simultaneous pseudorange and Doppler measurements from two satellites and an altitude constraint, five positioning equations for the five unknowns can be formulated as

$$f(x) = \begin{bmatrix} \frac{\|s_i - u\| + b - \rho_i}{\|s_i - u\|} - b - \dot{\rho}_i \\ \frac{\|s_j - u\| + b - \rho_j}{\|s_j - u\|} - b - \dot{\rho}_j \\ v_i \cdot \frac{s_i - u}{\|s_i - u\|} + \dot{b} - \dot{\rho}_i \\ v_j \cdot \frac{s_j - u}{\|s_j - u\|} + \dot{b} - \dot{\rho}_j \\ \left(\frac{u_t}{\tau_a + h}\right)^2 + \left(\frac{u_y}{\tau_b + h}\right)^2 + \left(\frac{u_z}{\tau_b + h}\right)^2 - 1 \end{bmatrix} = 0 \tag{4}$$

where the unknown state vector $x$ comprises five unknowns:

$$x = \begin{bmatrix} u \\ b \\ \dot{b} \end{bmatrix}^T \text{ with } u = [u_x \ u_y \ u_z]^T.$$  The first two equations in (4) correspond to pseudorange measurements (2), the next two are Doppler positioning equations (3), and the last one is derived from the World Geodetic System ‘84 (WGS84) ellipsoid augmented by altitude $h$. The symbols $\tau_a$ and $\tau_b$ refer to the semi-major and semi-minor axis of the Earth, respectively, as defined in the WGS84 model.

This system of equations (4) can be solved, e.g., by using an iterative algorithm to minimize $\|f(x)\|^2$. In this paper, the Gauss–Newton method [2] is used. It utilizes the Taylor series expansion to approximate the original nonlinear equation with a linear model from which the model’s parameters are straightforward to estimate.
First, an initial state estimate $\hat{x}_0$ is chosen for the iteration. At each iteration cycle, the estimate is refined by a correction

$$\delta\hat{x}_k = -J^{-1}_{f(\hat{x}_k)} f(\hat{x}_k)$$  \hspace{1cm} (5)$$

where $J_{f(\hat{x}_k)}$ denotes the Jacobian matrix of $f$ computed at $\hat{x}_k$, explicitly expressed as

$$J_{f(\hat{x}_k)} = \begin{bmatrix} \frac{s_i - \hat{u}_k}{\|s_i - \hat{u}_k\|} & 1 & 0 \\ \frac{s_j - \hat{u}_k}{\|s_j - \hat{u}_k\|} & 1 & 0 \\ \frac{s_k - \hat{u}_k}{\|s_k - \hat{u}_k\|} & \frac{V_i}{\|s_i - \hat{u}_k\|} & \frac{V_j}{\|s_j - \hat{u}_k\|} & \frac{V_k}{\|s_k - \hat{u}_k\|} \end{bmatrix}^T$$

For a detailed derivation of the gradient of Doppler shift (3), see [20]. The next iteration estimate is obtained by adding the computed correction to the current state estimate:

$$\hat{x}_{k+1} = \hat{x}_k + \delta\hat{x}_k.$$  \hspace{1cm} (7)$$

Each new measurement epoch gives additional pseudorange and Doppler measurements from the same satellites. This enables using more than five equations to compute a position solution by using pseudorange and Doppler measurements from each additional epoch. In that case, the correction (5) must be computed using, e.g., least-squares estimation as the Jacobian matrix is no longer square and thus is not invertible. However, if pseudoranges from more than one epoch are used, the change in the clock bias $b$ caused by the drift $\dot{b}$ between the epochs must be compensated for. It is also possible to take multiple epochs into account by means of filtering; however, time series computations are beyond the scope of this article.

Iteration steps are taken until a stopping criterion is met. Stopping criteria may be based on, e.g., the norm of the correction $\delta\hat{x}_k$ or the maximum number of iteration steps to be taken. It should be noted that the choice of the initial guess affects the final position solution; simulation results indicate that this system of equations always has two possible solutions on the Earth, which is verified in [3], [5]. However, in some cases it is possible to exclude one of the solutions by examining the visibility areas of the satellites.

**C. Conditioning the Jacobian Matrix**

Depending on the scaling of the measurements and the clock errors $b$ and $\dot{b}$, the Jacobian matrix $J_{f(x)}$ can be badly conditioned. Consider the row vectors formed by the first three elements of each row of the Jacobian in the case where pseudorange is measured in meters and Doppler shift in meters per second. The rows corresponding to the pseudorange measurements have unit Euclidean norm; in contrast, the vectors corresponding to the Doppler measurements are not normalized. Knowing that

$$\|x \times y\| = \|x\| \|y\| \sin \angle(x, y) \leq \|x\| \|y\|,$$  \hspace{1cm} (8)$$

the norm is bounded from above by

$$\begin{bmatrix} s_i - \hat{u}_k \\ s_j - \hat{u}_k \\ s_k - \hat{u}_k \end{bmatrix} \left( \begin{bmatrix} \|s_i - \hat{u}_k\| & \|s_j - \hat{u}_k\| & \|s_k - \hat{u}_k\| \end{bmatrix} \begin{bmatrix} V_i \\ V_j \\ V_k \end{bmatrix} \right) \leq \begin{bmatrix} \|s_i - \hat{u}_k\| \\ \|s_j - \hat{u}_k\| \\ \|s_k - \hat{u}_k\| \end{bmatrix} \leq \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}.$$  \hspace{1cm} (9)$$

Since GPS satellites move at approximately 4 km/s at an altitude of 20 000 km, the norm of the Doppler measurement derivative vector (excluding the clock drift) is with this scaling at maximum approximately 0.0002 Hz; however, the last element of the line is 1, which causes the Doppler rows to be close to linear dependency. The row corresponding to the barometer measurement also has a small norm (approximately $10^{-7}$) but does not have ones in either of the last two columns.

The bad scaling of $J_{f(x)}$ suggests that the matrix is badly conditioned, reflecting that the solution of (5) will be very sensitive to measurement errors. Indeed, numerical experiments with authentic GPS data have shown that the condition number of $J_{f(x)}$ tends to be in the order of $10^5$ or larger. In traditional pseudorange positioning (using four satellites), the condition number is typically in the order of 10. Therefore, the two-satellite solution can be significantly more error-prone if implemented on devices with a low numerical precision.

However, the Jacobian can be conditioned by choosing different units for the measurements and clock errors such that all entries of the Jacobian can be expected to lie in the same columns.

Iteration steps are taken until a stopping criterion is met. Stopping criteria may be based on, e.g., the norm of the correction $\delta\hat{x}_k$ or the maximum number of iteration steps to be taken. It should be noted that the choice of the initial guess affects the final position solution; simulation results indicate that this system of equations always has two possible solutions on the Earth, which is verified in [3], [5]. However, in some cases it is possible to exclude one of the solutions by examining the visibility areas of the satellites.

**D. Selecting the Correct Solution**

As discussed above, the system of equations (4) has two possible solutions. An example of the situation is illustrated in Figure 2. Two pseudorange measurements define a hyperboloid; the intersection of this hyperboloid and the altitude constraint ellipsoid is close to a circle. Finally, the Doppler constraint surface intersects this curve at two points [3].

One potential way to test whether a solution candidate is admissible is to check the visibility of the satellites at the obtained solution. When satellite $i$ is above zero elevation at receiver location $u$, the following holds [22]:

$$s_i \cdot u \geq u \cdot u.$$  \hspace{1cm} (10)$$

Usually the solution candidates are located nearly on opposite sides of the globe. In this case picking the right solution is trivial, as (10) is not satisfied for both satellites at both candidates. This is the case in Figure 2, although the false
solution is fairly close to the boundary of the visibility region. However, in some cases both satellites are visible at both solutions. In this case, the solution can be chosen based on some additional information (e.g., knowledge on the drifting characteristics of the receiver clock, although cheap oscillators tend to behave unpredictably; receivers integrated to cellular phones may take advantage of knowing the location of the serving cell), or, if the solution is only used to help satellite phones, it may be feasible to use both solutions. In this case, the solution can be chosen based on some additional information (e.g., knowledge on the drifting characteristics of the receiver clock, although cheap oscillators tend to behave unpredictably; receivers integrated to cellular phones may take advantage of knowing the location of the serving cell), or, if the solution is only used to help satellite acquisition, simply by guessing.

Because of the possibility that both solutions can be visible to each of the satellites, it is important to find both solutions on the Earth surface. A good iteration starting point with satellites $i$ and $j$ visible is the mean of the unit vectors pointing from the center of the Earth to the satellites multiplied by the semi-major axis of Earth:

$$\hat{u}_0 = \frac{s_i}{\|s_i\|} + \frac{s_j}{\|s_j\|} \frac{r_a}{2}. \quad (11)$$

To find the other solution, an obvious candidate for the initial position is $-\hat{u}_0$. If it does not give a different solution, the vectors parallel and opposite to the Cartesian basis vectors can be tried (totaling six different vectors). If two distinct solutions have still not been found, initial positions can be tried (totaling six different vectors). If two distinct solutions have still not been found, initial positions can be tried (totaling six different vectors). If two distinct solutions have still not been found, initial positions can be tried (totaling six different vectors). If two distinct solutions have still not been found, initial positions can be tried (totaling six different vectors). If two distinct solutions have still not been found, initial positions can be tried (totaling six different vectors).

V. ERROR ANALYSIS

In conventional pseudorange positioning, the dilution of precision (DOP) is a quantity describing the goodness of the geometry of satellite set independently of the measurement precision; it is also possible to compute DOP values for Doppler positioning [20]. However, the same derivations cannot be used for combined pseudorange and Doppler positioning because the measurement errors cannot be assumed to be identically distributed, which is the key assumption that enables separating satellite geometry and measurement precision in DOP analysis. Furthermore, since Doppler shift is essentially the time derivative of pseudorange, the measurements cannot be assumed to be mutually independent either.

Given the measurement error covariance matrix it is possible to estimate the covariance of the state estimation errors based on the Jacobian of the measurements. However, since the fifth equation in (4) is implicit in terms of the altitude $h$, the corresponding Jacobian (6) cannot be used for this purpose directly. Therefore, following the analysis in [23, Section IV], we linearize (4) with respect to both the state variables and observations to get

$$\frac{\partial f}{\partial x} \delta x + \frac{\partial f}{\partial y} \delta y = 0 \quad (12)$$

where the vectors $\delta x$ and $\delta y$ contain the state and measurement perturbations, respectively. The partial derivatives with respect to $x$ correspond to the Jacobian (6), and the partials with respect to the measurements need to be determined. This is straightforward because in the case of explicit equations, the partials equal $-1$, and for the altitude constraint we obtain

$$\frac{\partial f_5}{\partial h} = -2 \frac{u_x^2}{(r_a + h)^3} - 2 \frac{u_y^2}{(r_a + h)^3} - 2 \frac{u_z^2}{(r_b + h)^3}; \quad (13)$$

all off diagonals of this derivative matrix are zero. Substituting this information into (12) and solving for $\delta x$ yields

$$\delta x = -J_{f(x)}^{-1} \left[ -I_{4 \times 4} \begin{array}{cc} -2 & 0 \\ 0 & -2 \end{array} \right] \delta y \quad (14)$$

where the entries of the matrix $\partial f/\partial y$ have been expressed explicitly. This enables estimating the covariance of the state vector errors as

$$\text{cov} \delta x \approx J_{f(x)}^{-1} \frac{\partial f}{\partial y} \text{cov}(\delta y) \left( J_{f(x)}^{-1} \frac{\partial f}{\partial y} \right)^T. \quad (15)$$

The same result can be derived by applying the Theorem of Implicit Functions [24, Section III.1.5] to the implicit altitude equation to obtain the partial derivatives of the explicit altitude function $h(x)$.

The different scaling of the rows of the Jacobian causes the sensitivity of the solution to measurement errors to depend on the type of the measurement. Let us partition the original Jacobian matrix as

$$J_{f(x)} = \begin{bmatrix} s_i r_i^T & 1 & 0 \\ s_j r_j^T & 1 & 0 \\ s_i c_i^T & 0 & 1 \\ s_j c_j^T & 0 & 1 \\ s_h h^T & 0 & 0 \end{bmatrix} \quad (16)$$

where $s_r$, $s_D$, and $s_h$ denote the optional conditioning coefficients discussed in Section IV-C, and write the measurement vector as $\tilde{y} = y + \delta y$ where $y$ is the ideal measurement and $\delta y$ is the measurement error. Now consider the problem $J_{f(x)} \delta x = \delta y$. First let $\delta y = [0 \ 0 \ s_D \epsilon_i \ 0 \ 0]^T$. Substituting this to (16) gives

$$\begin{bmatrix} s_D c_i^T \delta u + \delta b &= s_D \epsilon_i \\ s_D c_j^T \delta u + \delta b &= 0 \end{bmatrix}, \quad (17)$$

and subtracting the second equation from the first one yields $s_D (c_i - c_j)^T \delta u = s_D \epsilon_i$. Now, a lower bound for the error in the differential position correction can be obtained using the Cauchy–Schwarz and triangle inequalities:

$$\|\delta u\| \geq \frac{|\epsilon_i|}{\|c_i\| + \|c_j\|}. \quad (18)$$
Knowing, by (9), that $\|c\| \approx 2 \cdot 10^{-4}$, we obtain

$$\|\delta u\| \geq 2500 |\epsilon_i|$$  \hspace{1cm} (19)

for the Doppler measurement error. It can be seen that this bound is invariant to the conditioning of the Jacobian.

By letting $\delta y = [s_c \epsilon_i \ 0 \ 0 \ 0]^T$ and eliminating the clock bias instead of the drift, an analogous result for the pseudorange measurements can be derived:

$$\|\delta u\| \geq \frac{|c_i|}{2}. \hspace{1cm} (20)$$

This bound would be an equality if the satellites were located at opposite azimuth angles in the horizontal plane, i.e., at $0^\circ$ elevation. However, such a geometry both seldom appears in practice and is usually discarded because of a cutoff elevation of, e.g., $10^\circ$. Nevertheless, it is evident that pseudorange equations are significantly less sensitive to measurement errors than Doppler equations. Although Doppler shift can be measured more precisely than the code phase—the noise standard deviation in Doppler measurements is in cm/s range using low-cost receivers whereas pseudorange noise tends to be in decimeter to meter order—the precision difference is not as large as the difference in the coefficients in (19) and (20). Also in the literature, Doppler positioning has been observed to be more sensitive to errors than pseudorange positioning [20]. For an altitude measurement biased by $\delta h$, trivially $\|\delta u\| \geq \delta h$.

It should also be noted that these derivations assume the current position estimate to be close to the correct position (i.e., geometry errors in the Jacobian were not considered). As the algorithm presented in this paper is iterative and the starting point may be far from the correct solution, it is difficult to analyze the errors rigorously. The validity of the analysis presented above is assessed in the next section.

VI. TEST RESULTS

Although the GLONASS constellation is getting close to full operational capability, GPS is currently the only fully operational GNSS and was used in our experiments. Authentic pseudorange and Doppler observations were used in all tests. However, the altitude estimate was not measured but “simulated” by choosing it manually to approximately match the true location. The receiver was a priori known to be stationary during all measurements, and thus, motion detection was neither necessary nor attempted. All measurements were done in Tampere, Finland, roughly at 61$^\circ$ N, 23$^\circ$ E, 160 m in latitude, longitude, and altitude, respectively, with a commercial U-blox EVK-5T GPS receiver [25], except for the results presented in Section VI-C which are based on data originating from various locations around the world. First, the effect of this method on the TTFF is examined, followed by experimental sensitivity testing. Since (4) has two possible solutions on the Earth surface, the solution closer to the true receiver location was chosen for the analysis in these tests. Finally, a separate test is done on choosing the correct solution of the two candidates.

A. Impact on Time to First Fix

To test the improvement in TTFF, the receiver antenna was taken to a balcony where the sky view is significantly blocked by the ceiling and surrounding walls, thus constituting a multipath-prone environment. Then, the receiver was issued a cold start (to erase all previous information on the receiver location and almanac etc.) and data logging was started. The receiver achieved a TTFF of 385.6 seconds when computing the position based on pseudoranges only without aiding information. However, 24 seconds after the cold start, the receiver had decoded the clock correction and ephemeris parameters and resolved the carrier phase half-cycle ambiguity for two satellites. Resolving the half-cycle ambiguity is essential for this method with receivers implementing the carrier tracking loop as a Costas loop, as is the case with the GPS receiver used in our experiments, because otherwise, the Doppler measurement could be biased by a half wavelength [26]. Given an altitude constraint, sufficient information is available for computing a position fix; the altitude was simulated for each test run as $h = 170$ m plus zero-mean Gaussian noise with standard deviation 5 m. In our experiments, the barometer noise level was less than five meters which is illustrated in the example in Fig. 1.

The first 300 position fixes and the horizontal 95% confidence ellipse derived from the estimated solution covariance (15) are shown in Figure 3. It can be seen that the position estimates lie within a couple of kilometers from the true location and their horizontal coordinates are highly correlated, which is most probably due to the effect of Doppler measurement errors. The predicted confidence ellipse reflects this correlation fairly well, but it can be seen that it is not oriented exactly parallel to the set of solution points. A reason for this discrepancy can be that the confidence ellipse was computed by naively neglecting the correlations between the different measurements; a more accurate model of the measurement covariance would probably yield a better confidence estimate. In an urban environment, as was the case in this test, a precision of two kilometers is inadequate for direct personal positioning or navigation. Nevertheless, the improvement in TTFF was dramatic, and in many cases even a rough position estimate is better than none, e.g., for finding more satellites to be tracked.

![Fig. 3. 300 consecutive position fixes obtained using two satellites and the predicted 95% confidence ellipse.](image-url)
B. Error Sensitivity

The error sensitivity of the proposed method was tested by means of perturbation analysis. A set of data, measured with the antenna located on the roof of a building, with 11 visible satellites as shown in Figure 4 was taken, and for one epoch, position solutions using all possible two-satellite geometries (no cutoff elevation was chosen) were computed in four cases:

1) No artificial biases introduced
2) One pseudorange biased by 0.05 m
3) One Doppler measurement biased by 0.05 m/s
4) The altitude measurement biased by 0.05 m

The horizontal differences of the solutions 2)–4) with respect to case 1) were computed, and the results for the satellite pairs with the highest, lowest, and median position errors are listed in Table I. It can be seen that all pseudorange and Doppler caused errors obey the lower bounds (19) and (20), but in the worst case, an error of 0.05 m/s in a Doppler measurement can bias the position solution by 5.5 kilometers. As it can be seen from the Table I, the pseudorange-induced error for the satellite pair 12&23 is close to equality with bound (20); Figure 4 shows that the geometry of this satellite pair is close to the equality case of the bound described in Sec. V. The horizontal position error caused by an altitude bias seems to be in the same order as the altitude bias itself, but in some cases the resulting position error can be significantly worse.

The artificial biases in the different measurement types were chosen to be equal in order to directly demonstrate the different sensitivity to different measurement errors. In fact, the chosen error magnitude is significantly lower than the pseudorange noise level but, on the other hand, larger than the expected Doppler measurement noise. However, such an error can easily be induced in the Doppler measurement by receiver motion: a receiver velocity of 0.05 m/s directly towards the satellite would cause such a bias, which can be directly seen from (3). Consequently, it is vitally important that the receiver remains static during the measurement. Since an altitude bias of 0.05 m is practically negligible, the altitude perturbation test was repeated with \( \delta h = 8 \) m, i.e., simulating an air pressure measurement error of 1 hPa. The resulting errors were 8/0.05 times those obtained using the 0.05 m bias: the smallest horizontal position error was 3.6 m and the largest exceeded 190 m.

C. Solution Discriminability

Since the method presented in this paper has two solution candidates, it is important that both are found and the wrong one can be excluded; otherwise, the method would not be reliable. The efficiency of finding the two solutions and discrimination using satellite visibility information (10) is examined in this section.

The discrimination test was done using data logged on 16 Dec 2009 at five IGS stations [27] located at different latitudes (see Table II). Measurements made at 2:00, 8:00, 14:00, and 20:00 (in GPS time) were considered. At these stations, both solution candidates for all visible two-satellite geometries at each epoch were computed and the visibility discrimination test was applied. This yielded a total of 813 test cases.

The discrimination results are shown in Table III. The discrimination results were categorized as ‘correct’ (the right solution accepted and the false one excluded), ‘incorrect’ (the wrong solution accepted and the correct one discarded), or ‘none’ (both solutions accepted or both solutions rejected). An incorrect discrimination did not occur at any test case, which is vitally important. It can be seen that the discrimination method worked in more than 60% of the cases in all location and time combinations except one where the number of available geometries \( n \) was only 10. The total percentage of correct decisions is 75.5% in the 813 cases.

The last column of Table III shows the mean number of iteration starting points needed to find both solutions (the starting points were generated as explained in Sec. IV-D); in most of the cases, the starting point according to (11) and its opposite converged to the distinct solutions. There were, however, cases where more starting points were needed, but the distinct solutions were always found without the need of randomly generating initial positions. The table also shows that the positioning errors are approximately 1.6 kilometers or less; however, a gross error of over 6 km was also encountered.

### Table I

<table>
<thead>
<tr>
<th>PRNs</th>
<th>Resulting position error</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>( e = 0.05 ) m</td>
</tr>
<tr>
<td>12&amp;23</td>
<td>0.026</td>
</tr>
<tr>
<td>16&amp;21</td>
<td>0.33</td>
</tr>
<tr>
<td>13&amp;29</td>
<td>0.045</td>
</tr>
<tr>
<td>29&amp;21</td>
<td>0.071</td>
</tr>
<tr>
<td>16&amp;29</td>
<td>0.051</td>
</tr>
<tr>
<td>2&amp;31</td>
<td>0.038</td>
</tr>
</tbody>
</table>

### Table II

<table>
<thead>
<tr>
<th>Station ID</th>
<th>Location</th>
<th>Latitude</th>
</tr>
</thead>
<tbody>
<tr>
<td>alrt</td>
<td>Ellesmere Island, Canada</td>
<td>82 N</td>
</tr>
<tr>
<td>vill</td>
<td>Villafranca, Spain</td>
<td>40 N</td>
</tr>
<tr>
<td>glps</td>
<td>Puerto Ayora, Ecuador</td>
<td>1 S</td>
</tr>
<tr>
<td>chat</td>
<td>Waitangi, New Zealand</td>
<td>44 S</td>
</tr>
<tr>
<td>mcm4</td>
<td>Ross Island, Antarctica</td>
<td>78 S</td>
</tr>
</tbody>
</table>
Table III
RESULTS OF THE DISCRIMINATION TEST.

<table>
<thead>
<tr>
<th>Station</th>
<th>Time</th>
<th>( n )</th>
<th>Decision (%)</th>
<th>Min. separation [m]</th>
<th>Max. error [m]</th>
<th>Avg # of runs</th>
</tr>
</thead>
<tbody>
<tr>
<td>alrt</td>
<td>02:00</td>
<td>55</td>
<td>Correct</td>
<td>0</td>
<td>146 000</td>
<td>785</td>
</tr>
<tr>
<td></td>
<td>08:00</td>
<td>36</td>
<td>Correct</td>
<td>0</td>
<td>776 000</td>
<td>618</td>
</tr>
<tr>
<td></td>
<td>14:00</td>
<td>55</td>
<td>Correct</td>
<td>0</td>
<td>51 900</td>
<td>1130</td>
</tr>
<tr>
<td></td>
<td>20:00</td>
<td>45</td>
<td>Correct</td>
<td>0</td>
<td>1 250 000</td>
<td>235</td>
</tr>
<tr>
<td>chat</td>
<td>02:00</td>
<td>45</td>
<td>Correct</td>
<td>0</td>
<td>1 750 000</td>
<td>373</td>
</tr>
<tr>
<td></td>
<td>08:00</td>
<td>21</td>
<td>Correct</td>
<td>0</td>
<td>2 040 000</td>
<td>1390</td>
</tr>
<tr>
<td></td>
<td>14:00</td>
<td>21</td>
<td>Correct</td>
<td>0</td>
<td>719 000</td>
<td>262</td>
</tr>
<tr>
<td></td>
<td>20:00</td>
<td>55</td>
<td>Correct</td>
<td>0</td>
<td>2 230 000</td>
<td>1270</td>
</tr>
<tr>
<td>glps</td>
<td>02:00</td>
<td>36</td>
<td>Correct</td>
<td>0</td>
<td>154 000</td>
<td>1150</td>
</tr>
<tr>
<td></td>
<td>08:00</td>
<td>45</td>
<td>Correct</td>
<td>0</td>
<td>69 700</td>
<td>604</td>
</tr>
<tr>
<td></td>
<td>14:00</td>
<td>45</td>
<td>Correct</td>
<td>0</td>
<td>91 200</td>
<td>1190</td>
</tr>
<tr>
<td></td>
<td>20:00</td>
<td>36</td>
<td>Correct</td>
<td>0</td>
<td>393 000</td>
<td>1550</td>
</tr>
<tr>
<td>mcm4</td>
<td>02:00</td>
<td>36</td>
<td>Correct</td>
<td>0</td>
<td>235 000</td>
<td>695</td>
</tr>
<tr>
<td></td>
<td>08:00</td>
<td>55</td>
<td>Correct</td>
<td>0</td>
<td>657 000</td>
<td>1210</td>
</tr>
<tr>
<td></td>
<td>14:00</td>
<td>36</td>
<td>Correct</td>
<td>0</td>
<td>912 000</td>
<td>390</td>
</tr>
<tr>
<td></td>
<td>20:00</td>
<td>55</td>
<td>Correct</td>
<td>0</td>
<td>33 900</td>
<td>1440</td>
</tr>
<tr>
<td>vill</td>
<td>02:00</td>
<td>45</td>
<td>Correct</td>
<td>0</td>
<td>155 000</td>
<td>6440</td>
</tr>
<tr>
<td></td>
<td>08:00</td>
<td>10</td>
<td>Correct</td>
<td>0</td>
<td>989 000</td>
<td>1080</td>
</tr>
<tr>
<td></td>
<td>14:00</td>
<td>45</td>
<td>Correct</td>
<td>0</td>
<td>1 300 000</td>
<td>1220</td>
</tr>
<tr>
<td></td>
<td>20:00</td>
<td>36</td>
<td>Correct</td>
<td>0</td>
<td>101 000</td>
<td>1520</td>
</tr>
</tbody>
</table>

Table I shows, such an error can result from a centimeter-level Doppler bias in the case of an unfavorable satellite geometry. It should be noted that the data were not screened beforehand for gross measurement errors or bad satellite geometries.

In all test cases, the two solutions were separated by more than 33 kilometers. Since the largest encountered positioning error was 6.5 km, the solution separation was significantly larger than the solution biases, and there was thus no danger of selecting the wrong solution because of positioning errors. If the solution candidates were spaced by 6.5 km only, it would be likely that the visibility test would fail and the method would not yield a solution unless some other method of discrimination is used. If the receiver is integrated into a cellular phone, solutions separated by more than 30 kilometers are probably discriminable when the location of the serving cell is known.

VII. CONCLUSIONS

In this paper, a method for computing a position fix using two GNSS satellites and altitude aiding was presented. To compute a position solution, both pseudorange and Doppler measurements were needed from the two satellites; a source of altitude information, e.g., a MEMS barometer, is the only extra hardware requirement in addition to a basic GPS receiver, and it can be replaced by a guess of the altitude if the method is used for purposes where the positioning accuracy is irrelevant, e.g., as a position estimate for further satellite acquisition. Nevertheless, since the applicability of the method is not limited to that use case only, it is justified to use an external altitude estimate to improve the accuracy of the position estimate.

The method assumes that the receiver is stationary, and this assumption may be verified using further aiding information (from, e.g., motion sensors); it should be noted that stationarity is only required for one measurement epoch (typically 1 s). Noise in a Doppler measurement can induce an error thousands of times larger than an equal magnitude noise component in a pseudorange measurement would cause. The altitude measurement error may also in some cases cause a significant (horizontal) position bias. The geometry of the satellites also has a drastic impact on the accuracy of the position solution. However, even an inaccurate initial position solution can be very important, e.g., in the case of emergency calls, if it can be obtained quickly (provided that the inaccuracy is acknowledged and can be accepted). It was demonstrated that the method can provide a significant decrease in the TTFF, particularly if the sky view is partly blocked and acquiring four satellites takes a long time.

For a receiver integrated into a cellular phone, it is also possible to decrease the TTFF by Assisted GNSS (AGNSS) techniques where the satellite navigation data are downloaded from the cellular network. In terms of positioning accuracy, AGNSS is better than the proposed method, but on the other hand, there are situations where the two-satellite approach has advantages over AGNSS. Firstly, if the user is abroad, he or she may be reluctant to use AGNSS because of the potentially high roaming data transfer costs required for obtaining the assistance data. Secondly, if the user is located in an urban canyon or other place where the sky view is mostly blocked, it may be that four satellites cannot be successfully acquired even when assistance data is available.

As the vertical component is fixed by the external altitude information, noise in Doppler and pseudorange measurements induces horizontal error on the solution. The most significant error source is the Doppler measurement error, possibly causing very large horizontal positioning biases. As the system of equations (5) is not overdetermined, standard error detection procedures such as, e.g., Receiver Autonomous Integrity Monitoring are not applicable, which may prevent using the method in safety-critical applications; on the other hand, such applications are not usually stationary either.

In most cases the system has two solutions that satisfy
the altitude constraint, and thus, the solution is ambiguous. In some cases, the false solution can be excluded based on the visibility regions of the satellites, but this cannot be generalized to all satellite geometries; according to the test results, in more than 75% of the cases, the correct solution can be selected based on the satellite visibilities. Depending on the application, other assistance for selecting the correction solution may be available. On the other hand, if the position solution is only used for further satellite acquisition, the false solution need not be explicitly excluded: whichever solution helps in satellite acquisition is the right one.

As future work, a closed-form algorithm—for example, as proposed by [5]—could be implemented and compared with the algorithm presented in this article in terms of, e.g., numerical stability, computational complexity, and efficiency in finding both possible solutions.

REFERENCES


Differential Barometry in Personal Navigation

J. Parviainen, J. Kantola, J. Collin
Tampere University of Technology
P.O.Box 553, 33101 Tampere
Finland

Abstract- In many personal navigation applications accurate altitude information is required. Finding a correct floor in a tall building or precise guidance in multi-layer intersection is not possible with two-dimensional navigation system. Satellite navigation systems provide altitude information, but accuracy is not necessarily sufficient for this kind of situations. Barometers can be used to improve the accuracy and availability of the altitude solution. The objective of this study was to find relevant error sources of barometer based altitude in the context of personal navigation. MEMS barometer data were collected in several scenarios where different disturbances affect the pressure reading. To separate error sources, reference barometer at known altitude was used. The results show that barometers in differential mode provide highly accurate altitude solution, but local disturbances in pressure need to be taken into account in the application design.

I. INTRODUCTION

As location based services become increasingly popular, accurate 3D position must be provided to the service application. In many situations, the navigation solution from a satellite-based system is not available, or accuracy of the solution is wanting. The lack of visible satellites, and the bad quality of observations is a common problem in urban canyons and even more severe in indoor navigation. In vertical direction, the positioning availability and accuracy can be improved by adding a barometer to the system. Current Micro-Electro-Mechanical Systems (MEMS) technology enables barometers to be integrated with satellite navigation units. Barometers have their own difficulties as changes in ambient air pressure are not necessarily due to changes in altitude. Thus, the pressure sensor must be constantly calibrated, and preferably a reference barometer is needed.

The possibility to aid GPS with a barometer was well known at the beginning of the nineties, e.g. [1], [2], and [3]. However, because of the size and the cost it was not feasible to use them as a part of personal navigation. The idea of using differential barometry with GPS to seismic measurements was presented in [4]. In the publication it was noticed that in the forested areas positioning accuracy can be increased when the results were obtained by post processing the data. The use of barometer to obtain solution with less than four GPS satellites was represented in [5]. It was found that when the result is fed in to the Kalman filter, the acceptable accuracy can be achieved with less than four satellites.

In [6], the accuracy of GNSS positioning with MEMS pressure sensor was studied. Reference barometer information was obtained via auxiliary TCP/IP server connection. The pressure altitude was used with the GPS pseudoranges in the least square iteration process. In the paper evaluation was carried out using both indoor and outdoor measurements. Horizontal accuracy was not increased but in vertical direction the accuracy was noticeably improved. The objective of [7] was to find working settings for data delivery between the reference and measuring barometer. Also the accuracy of GPS/barometer positioning was discussed with and without the use of reference barometer. The reference data delivery was done using internet (TCP/IP).

Recent studies in the use of MEMS barometer to increase the accuracy of GNSS/INS vertical channel are presented in [8], [9] and [10]. These papers state that MEMS barometers can be integrated successfully with all kinds of sensors.

In this paper, the aim is to estimate the magnitude of different disturbances in barometer measurements and their effects on height solution. This analysis contains the influences of change of speed, ventilation, car fan, and distance to the reference barometer. This research was done for the purposes of various applications, such as estimating the profile of the road in the direction of movement and guiding the user to a correct floor in a tall building. Such applications require meter-level accuracy, and thus even small disturbance sources need to be studied. The research was carried out using low-power MEMS barometers. These sensors are small and can easily be integrated with a GNSS receiver. Used pressure sensors are able to sense altitude changes of even a few centimeters. The measurements were carried out with a ‘rover’ barometer, while another barometer, ‘base’, was kept at a constant altitude and used as a reference. The results were obtained by post processing these measurements using the base barometer data and GPS/DGPS solutions.

II. PRESSURE ALTITUDE

As it is commonly known, pressure measured by barometers can be converted to altitude information. A popularly used term for this height is pressure altitude. However, barometric altimeters give no absolute height information without the proper knowledge of local sea level pressure. The normal air pressure on the sea level is approximately 101300 Pa (1013 hPa), but it varies locally depending on the weather. If the weather conditions are not changing, near the Earth’s surface a difference of 100 Pa in pressure equals approximately 8 meters in altitude difference [11]; as the altitude increases the pressure decreases.
Altitude can be modeled more accurately by the International Standard Atmosphere (ISA) given in [13]. The equation for altitude $h$ is

$$ h = \frac{T_0}{R} \left( 1 - \frac{p(0)p_R}{p_0} \right), $$

where $R$ is the gas constant, $k$ is the temperature lapse rate, and $T_0$ and $p_0$ are temperature and pressure, respectively, at the sea level. If the altitude is known and the sea level pressure is to be solved, the equation can be written in

$$ p_0 = \frac{p(h)}{\left( \frac{Rk}{T_0} \right)^{\frac{1}{k}}} . $$

This equation is useful especially when reference barometers are used.

### III. MEASUREMENT DEVICES

The barometric tests were carried out using SCP1000 MEMS barometers manufactured by VTI Technologies. In the car tests the reference route was measured using Novatel L1/L2 DGPS equipment. To get consumer grade GPS position solution, Holux Sirf III receivers were used.

The VTI’s pressure sensor has four different operation modes: high resolution mode, which uses an output refresh rate of typically 1.8 Hz, and a high speed measurement mode, that uses data refresh rate of approximately 9 Hz. Typical noise resolution of the sensor is 3 Pa in the high speed measurement mode and 1.5 Pa in the high resolution mode. There is also available an ultra low power measurement mode and a low power measurement mode with an external trigger [13]. However, in our measurements only high resolution and high speed measurement modes were used. The diameter and height of the SCP1000 pressure sensor are 6.1 mm and 1.7 mm, respectively. Thus, such the pressure sensor can be easily integrated to a mobile device.

A custom program was made to log data from the barometer demo kit via USB interface to the PC. In all barometers there is a bias term, which should be compensated before use as an independent pressure sensor. Fig. 1 depicts pressures measured by three uncalibrated SCP1000 pressure sensors next to each other. The visible bias term between each sensor must be taken into account in high-accuracy differential use.

### IV. ERROR SOURCES IN VEHICULAR ENVIRONMENT

To assess barometer error sources in car navigation, a passenger car was equipped with GPS receivers and barometer. Fig. 2 illustrates the differences in altitude solutions. Barometer measurements were corrected with the reference barometer. In the very beginning the car was stationary, which can be seen as almost straight line in the barometer measurement. However, both GPS-based results wander. This is caused by a building near the vehicle. As the result shows, the pressure altitude from the barometers can be used to provide accurate altitude
At the start of measurement in Fig. 2, the height component is almost a constant as the car is not moving. Nevertheless, the change of the car inside fan can change the altitude component. Fig. 3 presents the changes of fan blower settings when the car was stationary. The setting 0 means that blower is off and setting 4 means the highest blowing rate. Initially the blower was in setting 2. Then the setting was changed to 0, back to 2, to 4 and finally to 0. The change points are marked in Fig. 3.

The pressure altitude varies over six meters depending on a blower setting. As could be predicted, the higher fan blowing rate caused lower the altitude solution. The same kinds of results were obtained also when the car was in motion on a relatively flat road. Fig. 4 depicts such a test. It can be seen that the errors caused by blower correlate with the results in Fig. 3, but separation of fan setting and real changes in altitude become naturally more difficult.

During testing it was also noticed that opening the car window causes an error from some tens of centimeters to a few meters in the altitude measurement. This error is dependent on the speed of the car and, naturally, how open the window is. It was also noticed that when the other disturbances were minimized, pure speed, acceleration and jerk had no measurable error to pressure.

A test route included a 180 meter long tunnel. Fig. 5 presents the measurement with 9 Hz sampling rate. It was noticed that the variance of the pressure measurement increased during the tunnel. However, if the measurements are averaged during the tunnel there were no or only a very little bias. Thus, when the measurement is resampled to 1 Hz the effect of tunnel nearly disappears.

V. REFERENCE BAROMETER ERROR SOURCES

To obtain altitude accurately with a barometer a reference barometer is needed. The altitude of the reference barometer must be known or at least the altitude difference between measurement and reference barometers. Some tests were carried out to compare pressures in different locations. It was noticed that pressure inside the large building did not differ drastically.

Most significant errors were noticed when the barometer was moved from inside to outside. Fig. 6 presents a pressure measurement where at the beginning both barometers are inside the house. Then after five minutes the other is moved outside. As can be seen from the figure, pressure starts to grow increasingly compared with the one left inside. This error was partly caused by the temperature change between indoors and outdoors; the temperature was +22 degrees Celsius indoors and -3 outdoors. In general it was noticed that if ventilation systems are kept constant inside, the pressure has only some bias term when compared with outdoor pressure. Thus, in principle, it is also possible to use a reference barometer inside, even if measurements are made outdoors. However, preferable configuration is to keep both base and rover in the same environment. However, if the barometer is used as reference
during the temperature change, also the sensitivity of temperature to barometer output in a constant pressure need to be considered.

An important feature in barometric measurements is the distance to the reference barometer. Fig. 7 presents the pressure measurements of two stationary barometers, approximately 10 kilometers apart from each other. Both sensors were indoors and the measurement lasted seven hours. The figure shows that the maximum pressure difference between these two was 0.1 hPa. This converts approximately to 80 cm error. In general it was noticed that commonly the error caused by reference barometer with 10 km distance was less than one meter.

VI. TILT OF A MOVING VEHICLE

In general, inertial aiding of radio navigation systems requires a full six degree of freedom inertial measurement system (IMU). However, as gyroscopes are relatively expensive for personal navigation systems, aiding systems with reduced degrees of freedom have recently gained interest [14], [15]. In the absence of horizontal gyros, road grade need to be estimated. Altitude information provided by barometer can be used for this purpose. In this case, speed of the car is also needed. The following equation can be used to form the tilt angle

$$\alpha = \cos^{-1}\left(\frac{\Delta h}{\Delta \theta}\right),$$

where $\Delta h$ is the difference between two consecutive barometric altitude readings and $\Delta \theta$ is the average speed during the same time interval. Differentiator is needed in this process, and results are heavily affected by wideband noise of the barometer. Filtering of noise introduces a delay, and speed estimation via accelerometer data yields a positive-feedback loop [16]. Thus, measurement noise of barometer should be very small. From this viewpoint, new low-noise barometers are especially interesting. Fig. 8 presents the results from one measurement drive. The uppermost part presents the speed measurement by the Doppler radar. In the middle plot there is the barometric altitude measurement, which was corrected with the base barometer, with 5 km maximum distance to the rover barometer. Both data were resampled to 1 Hz rate and then filtered with a median filter with a window size 15. The lowest curve represents the road tilt that was calculated using (3). Comprehensive testing of the accuracy of this method would require high-quality IMU, and this is left for future research. As presented in the earlier sections, the barometric altitude solution is prone to all ventilation changes inside the car. Thus, for example, opening the car window could cause severe problems.

VII. INDOOR MEASUREMENTS

Barometers were also tested inside a five floor building to estimate their accuracy as floor detectors. The main errors in indoor measurements were caused by the different ventilation
in different rooms in building. However, no drastic changes in the air pressure were noticed. The highest errors were observed when the barometer was moved from a room to another, which had different ventilation conditions. An example of indoor processed measurement is presented in Fig. 9. In this test the barometer was kept in the hand while walking around the building and taking an elevator. Different floor levels can be clearly seen. Base barometer is required in this task.

It was observer that opening a door could temporary induce a meter level error. Fig. 10 presents an example data. Probably this was because of a slight temperature differences in the rooms, which caused a small draft when the door was opened. Also, the air flow caused by the swinging door increases the error.

VIII. CONCLUSIONS

In this paper different error sources in differential barometry were studied. The results showed that in vehicular environment accurate barometric altitude reading is notably affected by changes in ventilation. This study shows that the inclusion of low-cost and small MEMS barometers in personal navigation units with proper reference information can provide very useful information for personal navigation applications. However, as accurate pressure sensing does not directly translate to accurate vertical information, the changes in the local air pressure need to be taken into account when designing the system.

REFERENCES

Publication III

Adaptive Activity and Environment Recognition for Mobile Phones

Jussi Parviainen 1,*, Jayaprasad Bojja 1, Jussi Collin 1, Jussi Leppänen 2 and Antti Eronen 2

1 Department of Pervasive Computing, Tampere University of Technology, FI-33101 Tampere, Finland; E-Mails: jayaprasad.bojja@tut.fi (J.B.); jussi.collin@tut.fi (J.C.)
2 Nokia Technologies, FI-33721 Tampere, Finland; E-Mails: jussi.ar.leppanen@nokia.com (J.L.); antti.eronen@nokia.com (A.E.)

* Author to whom correspondence should be addressed; E-Mail: jussi.parviainen@tut.fi; Tel.: +358-3-3115-11.

External Editor: Gianluca Paravati

Received: 10 September 2014; in revised form: 16 October 2014 / Accepted: 20 October 2014 / Published: 3 November 2014

Abstract: In this paper, an adaptive activity and environment recognition algorithm running on a mobile phone is presented. The algorithm makes inferences based on sensor and radio receiver data provided by the phone. A wide set of features that can be extracted from these data sources were investigated, and a Bayesian maximum a posteriori classifier was used for classifying between several user activities and environments. The accuracy of the method was evaluated on a dataset collected in a real-life trial. In addition, comparison to other state-of-the-art classifiers, namely support vector machines and decision trees, was performed. To make the system adaptive for individual user characteristics, an adaptation algorithm for context model parameters was designed. Moreover, a confidence measure for the classification correctness was designed. The proposed adaptation algorithm and confidence measure were evaluated on a second dataset obtained from another real-life trial, where the users were requested to provide binary feedback on the classification correctness. The results show that the proposed adaptation algorithm is effective at improving the classification accuracy.

Keywords: mobile sensing; classifier design and evaluation; activity recognition; environment recognition; Bayes classifier; adaptation; pervasive computing
1. Introduction

The design of novel applications for modern smartphones has boomed in the past few years. This is partly due to easy access to information sources previously unavailable, such as the Global Positioning System (GPS) and motion sensors. For application designers, the smartphone environment is very different from the earlier personal computer world. Applications are running continuously, and the environment and activity of the user changes many times a day. Context information is often required to provide only the most relevant information or services to the user. In the present paradigm, the application requires the user to provide this information, such as ‘I am in a meeting’. Undoubtedly, it would be more convenient if this context information could be inferred automatically. For this reason, context recognition using the built-in sensors of modern mobile phones is an active research topic. The research problem is far from trivial, as the sensors provide only indirect information; there is no sensor for directly detecting ‘meetings’ or ‘jogging’. Furthermore, the required accuracy for the recognition is relatively high, and misclassifications can lead to very annoying user experiences. In this paper, we introduce a framework for automatic activity and environment classification that can be easily implemented and evaluated on any modern smartphone.

In this paper, we consider activity and environment recognition from mobile phone sensor and radio receiver data. The goal of activity recognition algorithms is to output information on the activity of the user. In particular, we consider algorithms that try to classify the physical activity of a user, such as walking, running, driving a car, riding a bicycle or being still. By environment recognition, we mean the automatic recognition of user surroundings, such as whether the user is in a meeting, at the office or inside a vehicle.

Existing literature on this topic includes many interesting examples of what can be done with sensor systems carried by the user. Altun et al. present a study of classifying human activities using body-worn inertial and magnetic sensors [1]. The authors perform a comparative study of different methods. The results in the paper indicate that, in general, Bayesian decision making results in the highest correct classification rate with a relatively small computational cost. Gu et al. describe a body sensor network for activity recognition for multiple users [2]. Könönen et al. describe several classification and automatic feature selection algorithms, which are compared in the problem of context recognition [3]. Reddy et al. present a classification system that uses a mobile phone with a built-in GPS receiver and an accelerometer [4]. The system recognizes the following transportation modes: whether an individual is stationary, walking, running, biking or in motorized transport. In the UPCASEproject [5,6], multiple sensors, such as accelerometers, light, sound, humidity, temperature and GPS sensors, were connected to a mobile phone via Bluetooth. Decision trees were used to obtain the activity (walking, running, standing or lying) of the user. The system also detects whether the device is indoors or outdoors by using information on the availability of GPS signals. After recognition, the context of the user can be published on social networks, such as Twitter or Facebook. The project CenceMe [7] also describes an activity recognizer where the results can be sent to social networks. The works in [8,9] present activity-based pattern matching in different mobile environments. Bancroft et al. use a foot-mounted inertial measurement unit (IMU) with a GPS to detect activity and the environment of the user [10]. In [11], Pei et al. use location and motion tracking to build a context-aware system. Susi et al. present
in [12] motion mode recognition for pedestrian dead reckoning. A smartphone-based lightweight hierarchical activity recognition framework is presented in Han et al. [13], where the recognition of 15 activities is done using the accelerometer, gyroscope, proximity sensor and GPS modules.

Our novel contribution to prior work is that we introduce an algorithm for user-specific adaptation of the context model parameters from user-friendly binary feedback. Furthermore, the introduced adaptation method is able to provide a confidence measure about the correctness of a classification for the host application.

This paper is organized as follows. In Section 2, we present the data used for training and testing the proposed classifier algorithms. Section 3 describes the features that were extracted from the collected data. Section 4 presents the feature compression algorithm used. An introduction to our classifier is presented in Section 5. In Section 6, we introduce our adaptation method. Section 7 presents the results from multiple classifiers and the results obtained with the proposed adaptation method. Finally, we conclude the paper in Section 8.

2. Data Collection

To be able to train and evaluate the classification system, a comprehensive annotated data set was collected. We chose Nokia N900 mobile phones for data collection, the main reason being easy access to sensor data via application programming interfaces and the open source nature of the Maemo platform. Two data collection campaigns were arranged. The first campaign, with 21 users, was arranged to collect a basic training set with full annotations, providing enough data for the design and validation of the feature extraction and pattern recognition algorithms. During the first campaign, the users were requested to manually input their current activity and environment from the mobile phone user interface (UI). The second campaign, with 10 users, was arranged for testing the adaptation of algorithm parameters for each individual user. The campaign was done with context recognition software running and with only binary yes, no input, indicating whether or not a recognition result was correct or not, required from the user. It should be noted that the first campaign required much more effort from the test persons since the users were providing full annotation compared to the binary yes/no answers in the second campaign.

2.1. The First Data Collection Campaign

In the first data collection campaign, the users were requested to carry the phone along with them as they normally would during three to four weeks. Our software would then ask the users to annotate their environment (e.g., outdoors, street), activity (e.g., walking) and the location of the phone (e.g., pocket) at set intervals. Once the annotation was input the phone sensors would then record data for two minutes. The default time interval between recordings was 20 min, but could be set by the user to be from a minimum of 10 min to a maximum of 60 min. The annotations were tagged to the recorded sensor data by a file name convention. The user interface used by the users to provide the annotations is depicted in Figure 1. There was a fixed set of predefined environments, activities, and phone locations available but the users could also add their own if they thought none of them matched their own situation.
Figure 1. Data collection program.

(a) Selecting the higher level or global environment
(b) Selecting a more detailed environment

c) Selecting the physical activity
(d) Selecting the mobile phone location

After the user provided an annotation for the context, the sensors on the device were activated and a short period of data collection was performed. The used sensors and statistics of the obtained data are presented in Tables 1 and 2. The collected sensor data is described in more detail below.

Table 1. Data trial statistics.

<table>
<thead>
<tr>
<th>Total Number of Users</th>
<th>21</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of men</td>
<td>15</td>
</tr>
<tr>
<td>Number of women</td>
<td>6</td>
</tr>
<tr>
<td>Median age</td>
<td>31</td>
</tr>
<tr>
<td>Mean age</td>
<td>32.3</td>
</tr>
<tr>
<td>Total number of annotations</td>
<td>3478</td>
</tr>
<tr>
<td>Number of unique annotations</td>
<td>620</td>
</tr>
</tbody>
</table>
Table 2. Collected data.

<table>
<thead>
<tr>
<th>Sensor</th>
<th>Sampling Rate</th>
<th>Sample Length</th>
</tr>
</thead>
<tbody>
<tr>
<td>3-axis Accelerometer</td>
<td>100 Hz</td>
<td>1 min</td>
</tr>
<tr>
<td>GPS</td>
<td>1 Hz</td>
<td>2 min</td>
</tr>
<tr>
<td>WLAN</td>
<td>$\frac{1}{12}$ Hz</td>
<td>1 min</td>
</tr>
<tr>
<td>Bluetooth</td>
<td>recorded once</td>
<td>snapshot</td>
</tr>
<tr>
<td>GSM/3G Cell information</td>
<td>1 Hz</td>
<td>1 min</td>
</tr>
<tr>
<td>Audio</td>
<td>16 kHz</td>
<td>1 min</td>
</tr>
</tbody>
</table>

- **Accelerometer**: The sampling rate of the 3-axis accelerometer was approximately 100 Hz. Data was collected for one minute.

- **GPS**: Three minutes of GPS data was collected with a 1 Hz sampling rate. The recorded data included position, velocity, satellite elevation and azimuth angles and signal to noise ratio.

- **WLAN**: WLAN signal strengths and WLAN station identifiers (IDs) and Media Access Control (MAC) addresses were recorded. Samples were collected for one minute and the sampling interval was approximately 12 s. Thus, five scans were performed during one minute.

- **Bluetooth**: A single scan of the Bluetooth environment was performed, and the names and addresses of the visible Bluetooth devices were logged.

- **GSM/3G Cell information**: Global System for Mobile Communications (GSM) or 3rd generation mobile communications (3G) location area codes (LAC) and cell IDs were recorded.

- **Audio**: One minute of audio with a single mono channel with 16-bit resolution at 16 kHz sampling rate was recorded.

### 2.2. Data Campaign for Adaptation

The second campaign was done using Nokia N900 devices wherein the developed classification algorithm for activity (classes: running, bicycling, walking, standing, table, vehicles) and environment (classes: office, nature, street/road, home, restaurant/pub/cafe) was running. Algorithms are described in detail in the Section 5.

The user feedback was requested by displaying a green button for "yes" and a red button for "no" right after the classification result was available. The user interface of the program is depicted in Figure 2. If the user did not provide an answer, the data was not used for adaptation. For this data collection campaign, ten users who did not participate the first campaign were selected for two week trial. In total, there were 2674 classification results with the version that asks the feedback.
3. Feature Extraction

This section presents the features extracted from the mobile phone sensor and radio receiver data. Multiple features were implemented and used, as the redundancy can be removed using a compression algorithm presented in Section 4.

3.1. GPS Features

The following features are extracted from the GPS receiver.

- The median carrier to noise ratio value of one minute of GPS data. The carrier to noise ratio is an estimate of the received power most GPS/GNSS (Global Navigation Satellite System) receivers provide for each tracked satellite. A low carrier to noise ratio indicates that there are obstacles such as building walls in the satellite-receiver path. The carrier to noise values are much lower indoors than outdoors [14].

- The minimum elevation angle value from the satellites that are used for getting the GPS position fixes. The elevation angle is the angle between the local horizontal plane and the user-satellite vector. In urban canyons satellites with a small elevation angle are rarely tracked.

- The maximum speed value from one minute of GPS data. Preferably a Doppler-based speed estimate computed by the receiver, but can also be derived from two consecutive position fixes. High speed values indicate that the device is in a moving vehicle.

- The best horizontal accuracy value of the GPS position fixes, i.e., the smallest horizontal accuracy value that the GPS device outputs. Most receivers provide this kind of value, the best accuracy is obtained in rural areas with a clear sky view.

- Time to first fix (TTFF), i.e., the time passed from the time of power-up to the instant when the first location measurement is obtained.
3.2. WLAN Features

The features extracted from the WLAN receiver are

- The number of unique MAC addresses.
- The number of unique station names. Usually in public areas one observes more MAC addresses than station names due to having several access points connected to the same WLAN network, whereas in private places such as homes one usually observes less MAC addresses per station name.
- The average signal strength on a scale from one to five.
- The average signal strength given on a decibel scale.
- The standard deviation of the signal strength on a scale from 1 to 5. A high standard deviation usually indicates that user is on the move.
- The standard deviation of the signal strength on a decibel scale.
- The maximum signal strength on a decibel scale. A high signal strength usually indicates that the user is indoors.
- The maximum signal strength on a scale from 1 to 5.

3.3. Bluetooth Features

A single feature related to the Bluetooth radio environment is used. The feature is the number of visible Bluetooth devices. A high number of Bluetooth devices indicates that the user is in a public place.

3.4. Cellular Network Environment Related Features

The cellular base stations to which the mobile phone connects during a minute are logged. From the logged data, the following statistics are used as features:

- The number of unique Cell IDs (base transceiver station identifiers).
- The number of unique location area codes (LAC).
- The number of Cell ID changes per minute.
- The number of LAC changes per minute.
- The standard deviation of the strength of the signal to the transceiver station the mobile phone is connected.
3.5. Accelerometer Features

To remove the effect of orientation of the mobile phone, the Euclidean norm of the three dimensional accelerometer signal is used. The following features are extracted from the accelerometer signal norm:

- The variance of acceleration.
- The value of the maximum spectrum magnitude peak.
- The bin index of the spectrum magnitude maximum value.
- The difference between the maximum and minimum spectrum magnitude value.
- The number of zero crossings.

3.6. Audio Features

For the audio features, likelihoods given by a global model trained for the audio data are used. The likelihood features are obtained as follows. First, during an offline training phase, 13 mel-frequency cepstral coefficients (MFCCs) along with their first- and second-order derivatives are calculated from audio training data [15]. The MFCCs are calculated from audio data recorded at 16 kHz sampling rate using a window of 30 ms and a frame-skip of 40 ms. After this, a single Gaussian mixture model (GMM) with 32 component densities is trained. Once the above offline training phase is complete, the likelihood features can be calculated for input audio data. This is done by, first, calculating MFCCs for the input audio data and then calculating the mixture likelihoods of the GMM for the MFCC feature vectors. The mixture likelihoods for the whole audio clip are then output as the likelihood features. Thus, for each input audio clip, we obtain a vector of 32 likelihood values. By using likelihood features we can represent a whole audio clip with a single feature vector instead of a sequence of MFCC vectors. An alternative way to represent an audio clip with a single feature vector would be to, for example, use the mean of the calculated MFCCs. Also, a covariance matrix could be calculated. However, information is lost when using only the mean of the MFCCs and for shorter audio clips, there might not be enough data to reliably calculate the variances or a covariance matrix.

4. Feature Compression

Given that a smartphone includes a multitude of relevant sensors for the classification task, the number of features is inevitably very high. To avoid numerical problems and make the adaptation more straightforward it is necessary to ignore redundant and irrelevant features. In order to reduce the dimensionality of the feature space, we use a method based on a transformation with two stages [16,17].

First, all the training features are concatenated in a $k \times n$ matrix $G$, where $k$ is the number of features and $n$ the number of feature vectors. In the following, by covariance matrix $\Sigma$ we mean the sample covariance matrix obtained from the training data $G$.

The averaged within-class covariance matrix is defined as

$$ S_W = \frac{1}{C} \sum_{j=1}^{C} \Sigma_j $$  \hspace{1cm} (1)
where $C$ is the number of classes (in this work we have multiple sets of classes $C$, e.g., the activities and environments have separate training data sets and classifiers. Nevertheless, for notational simplicity here we only use a single set of classes $C$). and $\Sigma_j$ is the covariance matrix of class $j$. The required transformation matrix $R_1$ is obtained using the eigenvectors (in the matrix $U$) and the rank of $S_W (r)$:

$$R_1 = \Lambda_r^{-\frac{1}{2}} U_r^T$$

(2)

where the diagonal matrix $\Lambda$ contains the $r$ eigenvalues of $S_W$, and thus $R_1S_WR_1^T = I$, an $r \times r$ identity matrix. This transformation scales the raw features conveniently. The dimension reduction removes redundant features, for example those that are constant all the time.

The next phase is to compress the class mean information. The between-class covariance matrix is defined as

$$S_B = \frac{1}{C} \sum_{j=1}^{C} (\mu_j - \mu)(\mu_j - \mu)^T$$

(3)

where $\mu$ is the mean of the within-class means $\mu_j$. $S_B$ describes how far apart the class means are from each other. In the second phase, we seek for a transformation $R_2 = V_{\nu}^T$ that diagonalizes $S_B$:

$$V_{\nu}^T (R_1S_BR_1^T)V_{\nu} = \tilde{\Lambda}_{\nu}$$

(4)

where $\nu$ is the rank of $R_1S_BR_1^T$ and the matrix $V_{\nu} = [v_1, \ldots, v_{\nu}]$ consists of $\nu$ eigenvectors of $R_1S_BR_1^T$ (i.e., eigenvectors of $S_B$ after the first transformation $R_1$). Similarly, $\tilde{\Lambda}_{\nu}$ is a diagonal matrix containing $\nu$ eigenvalues.

There are at most $C - 1$ non-zero eigenvalues (after the transformation the number of features is reduced at maximum to $C - 1$) and after the transformation $R_2R_1$ the first feature vector can be considered as the most important for the classification purposes. This geometric interpretation of linear discriminant analysis (LDA) is very useful in displaying the data, and it is straightforward to implement. The matrix $R = R_2R_1$ can be computed off-line, and the software in the mobile phone only needs to perform a matrix-vector multiplication for the raw features. In addition, some features clearly do not follow a Gaussian distribution (number of WLAN APs, for example), but after the feature compression we can quite safely work with Gaussian models.

5. Classification

For the classification task, we use a Bayesian maximum a posteriori classifier (MAP). For comparison, the classification results are also calculated using support vector machines (SVM) and decision trees (DT). Nevertheless, the probabilistic Bayesian MAP method has the following advantages:

(1) If the features follow Gaussian distribution the classifier is straightforward to implement. Furthermore, the output is optimal in a probabilistic sense, and it is possible to derive a proper confidence measure for the classification result.

(2) Adaptation can be performed by changing the parameters of the class distribution.
The Bayesian approach allows a straightforward implementation of recursive filters. For example, data from multiple users can be combined recursively from the same geolocation.

With the underlying Gaussian assumption, the pattern recognition problem simplifies to a discrimination between \( p \) multivariate normal populations. First, a training data set with known states is collected to obtain

\[
z_j \sim N(\mu_j, \Sigma_j)
\]

(5)

the distribution of the observed \( q - b \times y - 1 \)-vector \( z \) given that the observation comes from the class \( j \).

At this stage, the mean vector \( \mu_j(q \times 1) \) and the covariance matrix \( \Sigma_j(q \times q) \) are here assumed to be perfectly known (very representative training data set). We also assume that for all classes \( \Sigma_j > 0 \).

If Equation (5) holds, the density function of \( z_j \) is

\[
f_{z_j}(z; \mu_j, \Sigma_j) = \frac{1}{(2\pi)^{q/2}\sqrt{|\Sigma_j|}} \exp\left[-\frac{(z - \mu_j)^T \Sigma_j^{-1}(z - \mu_j)}{2}\right]
\]

(6)

A new observed \( z = z_x \) can then be classified by maximizing Equation (6) over all the classes \( j = 1...p \). To assign a probability to the classification result, we need to use unconditional prior probabilities \( P(C = j) \), and assume that all the possible classes are included. Then the probabilities for all the classes are obtained from the Bayes’ rule:

\[
P(C = j|z_x) = \frac{f_{z_j}(z_x)P(C = j)}{f_z(z_x)}
\]

(7)

where \( f_z \) is the unconditional density function for the observation \( z \). However, Equation (7) may not work well in practical classification problems as the Gaussian assumption does not necessarily hold. In addition, the class parameters \( \mu_j, \Sigma_j \) are only estimates obtained based on incomplete training data and the list of classes is not complete – we do not have data from all the possible activities or environments. For these reasons it is necessary to modify this basic model.

5.1. Combining Independent Classifiers

The information provided by all the \( p \) likelihoods \( f(z_x; \mu_j, \Sigma_j) \) of the classes \( j = 1...p \) becomes interesting when combining different classifiers, such as an audio-based classifier and radio receiver based classifiers. The motivation for this approach is that often there are specialized classifiers which operate on certain type of sensor data (e.g., audio or accelerometer). Different sensors have very different data rates and feature statistics, and often it is impossible to concatenate raw features to the same feature vectors to perform feature level fusion. Instead, if we have a classifier, the output of this classifier can be fed to another classifier as a feature vector. This makes it possible to roughly match the data rates of different sensors, as in the case of a radio receiver producing a set of data every one minute combined with the output of the audio classifier at the same rate. In addition, in this case the application software does not have access to the classifiers, but only obtains the result containing the likelihoods for all the possible classes, and the decision itself.

Thus it is convenient to use these likelihoods as features for another classifier that is running on a higher level in the software. By doing this we would not need to know any details of the original
classifier. First we need to investigate how the \( f_{z_j}(z) \) itself is distributed, considering \( f \) as a deterministic function of a random variable \( z \). First, assume that the new observation \( z = z_e \) comes from the population class \( j = 1 \). To simplify the computation we take the natural logarithm of the likelihood, yielding a constant term (for each class) \( \ln \frac{1}{\sqrt{2\pi\Sigma_1}} \) plus the term

\[
-\frac{(z - \mu_1)^T \Sigma_1^{-1} (z - \mu_1)}{2}
\]  

(8)

As the actual class is correct (i.e., \( z \) and \( \mu_1 \) belong to the same class), by Theorem A.85 in [18], the numerator in Equation (8) is chi-square distributed with \( q \) degrees of freedom.

Then, we need to study the distribution of incorrect class (i.e., classifier parameters are wrong). The distributions of likelihoods for the incorrect classes is not as straightforward. We will have a quadratic form \( (z + a)^T C (z + a) \), where the vector \( a \) and the matrix \( C \) have nothing to do with the parameters of the distribution of \( z \). From [19] we find that

\[
P\{ (z + a)^T C (z + a) \leq t \} = \sum_{k=0}^{\infty} c_k F(\cdot, t)
\]

(9)

where \( F \) is a chi-square cumulative distribution function (CDF). This formula can be used to obtain a numerical approximation of the CDF for the likelihoods in the case of incorrect class. The normally distributed features produce chi-squared likelihoods when the class model is correct. We also know the shape of the distribution (Equation (9)) in the case of an incorrect class. Thus, if we leave the assumption of normally distributed features aside, we can train a new classifier using directly the likelihoods of the classifier running on a lower level. Feature compression is clearly necessary when feature vectors with a non-normal degenerate distribution are added to the classifier. We used this approach to combine an existing audio-based environment classifier with the radio-based classifier developed in this study. In our existing audio classifier we could only touch the outputs of the classifier (i.e., in this case likelihoods). In addition, these likelihoods were not from the same classes that were interested in the final classification. Thus, classical ensemble methods [20] were not valid in this case.

6. Adaptation

It is clear that one single global model for activity or environment classification will not perform well in the case of mobile phones. There are individual differences that affect the activity monitoring between users (walking pace, step impact, bicycling speed, etc.). The same applies for the environment classification, for example, offices tend to have different WLAN coverage and there are differences in the audio and radio environment between cities. For that reason, adaptation via user feedback is needed. To reduce the amount of work the user has to do it is preferable to ask only binary information from the user: given the global distribution model the software outputs the classification result and prompts the user to provide a yes or no answer depending on whether the classification result was correct or not. This user input is then used to adapt the distribution parameters. A similar idea is also proposed in [3], but we will add the likelihood distribution models for the adaptation process.
6.1. Adaptation Algorithm

Using likelihood distributions we can derive a new optimization criteria: the yes answer likelihoods should be chi-square distributed with \( q \) degrees of freedom after the adaptation (this was the definition of numerator in Equation (8)). In addition, to reduce the amount of incorrect classification results, the no answer likelihoods should be far away (towards \(+\infty\)) from this distribution.

In this paper we use the following optimization criterion: modify the distribution parameters so that the function

\[
f = \frac{|A|}{N(\text{yes})} + \frac{|B|}{N(\text{no})}
\]

is minimized. It should be noted that it is possible to use other suitable adaptation functions, for example, the one proposed in [3]. Function in the paper was chosen somewhat heuristically. In our function symbols \( N(\text{yes}) \) and \( N(\text{no}) \) denote the total numbers of yes and no answers and \(|A|\) and \(|B|\) denote the number of items in the set \( A \) and \( B \), respectively. The sets are defined as

\[
A = \{L_i(\text{yes})|L_i(\text{yes}) > \chi_{95}\}
\]

and

\[
B = \{L_i(\text{no})|L_i(\text{no}) < \chi_{95}\}
\]

Here \( i \) is the index of the current class and \( L_i(\text{yes}) \) and \( L_i(\text{no}) \) are the set of likelihood values corresponding to observations with a yes-label and a no-label, respectively, and \( \chi_{95} \) is the point where the cumulative Chi-square distribution and has a value of 0.95. The likelihood values are

\[
L_i = (z - \mu_i)^T s_i \Sigma_i^{-1} (z - \mu_i)
\]

The parameters to be adapted are the scales \( s_i \) and the class means \( \mu_i \). After the adaptation, the new covariance will be \( s_i \Sigma_i \).

The motivation for the function Equation (10) is that it resembles the Neyman-Pearson type false positive, false negative terminology [21,22]. Minimization of the function \( f \) can be thought of as minimizing the false positive and false negative errors. However, the actual classification method is not taken into account here.

In essence, the adaptation algorithm attempts to minimize the number of samples with a yes-label that do not fit the model distribution (in Equation (6)), meaning that they fall outside the 95% threshold. Furthermore, the algorithm attempts to minimize the number of samples with no-label that fit the model distribution well, meaning that they fall inside the 95% threshold. To minimize the function \( f \) in Equation (10), i.e., to find

\[
\arg\min_{s_i \in \mathbb{R}^+, \mu \in \mathbb{R}^N} f
\]

We used the Matlab function fminsearch, which finds the minimum of the function using the simplex search method [23]. Functions \( f \) are implemented separately for activity and environment. The parameters to be adapted are the scale \( s \), so that the new covariance will be \( s \Sigma \), and the class means \( \mu \). For fminsearch the scale is represented as \( d = s - 1 \), so that the adaptation starts from the zero-vector. The result of fminsearch then tells how the parameters need to be adapted. If the adaptation is successful, we
should obtain better classification results in the future. Furthermore, the likelihood distribution should
approximately follow the model we have imposed - regardless of the original assumption of a Gaussian
distribution.

The information contained in the sample distribution (obtained using the Matlab function ksdensity,
for example) of yes-likelihoods and no-likelihoods is very useful for classifier diagnostics. For example,
if the real feature distribution is actually multimodal, the yes-likelihood distribution should not look like
a Chi-squared distribution.

6.2. Simulated Example

To illustrate our adaptation algorithm, we present a simulated example in this section. In this
simulation we assume that we have trained a MAP classifier having parameters \( \hat{\mu} \) and \( \hat{\Sigma} \). However,
the true parameters for the individual user would be \( \mu \) and \( \Sigma \) and there exists a nearby untrained class
with parameters \( \mu_{other} \) and \( \Sigma_{other} \). The numerical values for the distribution parameters are given in
Table 3 and the data is illustrated in Figure 3. Using Equation (9) the probability density function for the
yes-likelihoods \( (p_{L|yes}) \),

\[
s_1 = (z - \hat{\mu})^T \hat{\Sigma}^{-1} (z - \hat{\mu})
\]

\[
z \sim N(\mu, \Sigma)
\]

and the probability density function for the no-likelihoods \( (p_{L|no}) \),

\[
s_2 = (z - \hat{\mu})^T \hat{\Sigma}^{-1} (z - \hat{\mu})
\]

\[
z \sim N(\mu_{other}, \Sigma_{other})
\]

can be derived \([24–27]\). The CDF for the random variable \( s_i \) is given by \( P\{s_i \leq t\} = \sum_{k=0}^{\infty} c_k F(\cdot,t) \).
The algorithm for the coefficients \( (c_0 \ldots c_N) \) can be found from \([19]\). The likelihood distributions are
shown in Figure 4 (top). The aim of the adaptation process (using Equation (10)) is to push the “no”
distribution \( (p_{L|no}) \) to the right side of the \( \chi_{95} \) threshold and the “yes” distribution \( (p_{L|yes}) \) to the left side.
Assuming that the adaptation is perfect, the estimated parameters \( \hat{\mu}, \hat{\Sigma} \) equal the true parameters \( \mu, \Sigma \).
Thus, the probability density function for the yes-likelihoods \( (p_{L|yes}) \) becomes

\[
s_3 = (z - \mu)^T \Sigma^{-1} (z - \mu)
\]

\[
z \sim N(\mu, \Sigma)
\]

and the probability density function for the no-likelihoods \( (p_{L|no}) \) become,

\[
s_4 = (z - \mu)^T \Sigma^{-1} (z - \mu)
\]

\[
z \sim N(\mu_{other}, \Sigma_{other})
\]

which are shown in Figure 4 (bottom). The chosen threshold is somewhat arbitrary, but it should be noted
that in a real situation the parameters \( \mu_{other} \) and \( \Sigma_{other} \) are not known, so it is difficult to avoid heuristics.
However, the distribution for \( s_3 \) Equation (18) is known in the case of perfect adaptation. This provides
means to find a stopping criterion for the adaptation. For example, when a sample distribution function
$S_n(x)$ of yes-likelihoods is sufficiently close to the optimal $\chi^2$ distribution function the adaptation process can be stopped. The measure for closeness can be, for example, Kolmogorov’s $D_n$,

$$D_n = \sup_x |S_n(x) - F(x)|$$

(19)

where $F$ is the known $\chi^2$ cumulative distribution function with $q$ degrees of freedom.

**Figure 3.** Simulated distributions.

**Table 3.** Parameters used in the simulation.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\mu$</td>
<td>( \begin{bmatrix} 4 \ 6 \end{bmatrix} )</td>
</tr>
<tr>
<td>$\Sigma$</td>
<td>( \begin{bmatrix} 0.8 &amp; 0.5 \ 0.5 &amp; 0.7 \end{bmatrix} )</td>
</tr>
<tr>
<td>$\hat{\mu}$</td>
<td>( \begin{bmatrix} 1 \ 2 \end{bmatrix} )</td>
</tr>
<tr>
<td>$\hat{\Sigma}$</td>
<td>( \begin{bmatrix} 0.4 &amp; -0.2 \ -0.2 &amp; 1.2 \end{bmatrix} )</td>
</tr>
<tr>
<td>$\mu_{other}$</td>
<td>( \begin{bmatrix} 0 \ 0 \end{bmatrix} )</td>
</tr>
<tr>
<td>$\Sigma_{other}$</td>
<td>( \begin{bmatrix} 2 &amp; -0.6 \ -0.6 &amp; 1 \end{bmatrix} )</td>
</tr>
</tbody>
</table>
6.3. A Confidence Measure

A fundamental requirement for sensor fusion is to obtain some kind of confidence measure for the decisions made. With this kind of information sensors can be be weighted based on the accuracy, or more sensors may be turned on if, for example, the uncertainty is high. We already described a likelihood based method that can be extended to the adaptation case easily: to obtain a confidence measure for the classifications after the adaptation, we will try to predict the probability of the user answering yes, given the current observed features and the adaptation data. To do this we use the Bayes’ theorem,

$$P_{yes|L} = \frac{p_{L|yes}P_{yes}}{p_L}$$  \hspace{1cm} (20)

where $P_{yes|L}$ is the probability we are looking for, that is, the probability of the user answering yes given the likelihood value of the class that was selected. $p_{L|yes}$ is the (empirical) conditional density of the yes-likelihood. $P_{yes}$ is the probability of successful detection after the adaptation. $p_L$ is the weighted combination of the yes and no likelihoods. The no likelihood distribution enables us to normalize the posterior probability, which would be very difficult without the adaptation process [28]. It should be noted that in the case of the user answering no, we do not know the actual correct class. This is a drawback of the simple binary feedback method, but our method of predicting the probability of the yes answers does not require the knowledge of the true class in the case of a negative answer. To avoid heavy computations on the mobile phone, the required distribution estimates can be computed on the server side. The necessary information to be transmitted to the server would be the $L_i(yes)$ and $L_i(no)$ values, and the server would return the new classifier parameters along with $p_{L|yes}, P_{yes}$ and $p_L$.

7. Results and Discussion

We implemented and tested three different classifiers: an indoor/outdoor classifier, an environment classifier, and an activity classifier. In addition to the maximum a posteriori (MAP) classification method
presented in Section 5 we present results using decision trees (DT) and support vector machines (SVM) for comparison. For those purposes MatLab’s ClassificationTree class with default options and SVM functions with linear kernel function were used. In the case of SVM, one-versus-others classifiers were implemented and the class was selected based on the classifier which classified the test data with the greatest margin. Evaluation of the accuracy of the all algorithms was first done by using leave-one-user-out cross validation, where at each round all the data from a user is held out and the system is trained with the data from the remaining users. This way we can ensure that the system is not overfitting to the individual characteristics of any user. As the training data was collected in opportunistic way, the class imbalance problem [29] may occur. Thus, equal priors were given for all classes in MAP classifier as we cannot prefer any class over another.

In the second phase, to evaluate our adaptation method presented in Section 6 the second data collection campaign was arranged and the method for classifier adaptation was evaluated.

7.1. Indoor/outdoor Classification

A study to evaluate the accuracy of a simple \{indoor, outdoor\} classifier was performed. In this task GPS signals are very relevant as the low power levels prevent satellite signal tracking in most buildings. The binary classification induces some problems in defining the boundaries. We defined driving a passenger car or travelling in a train as outdoor environment, the main reason being that radio receivers are less affected by vehicles than buildings.

The confusion matrix for the MAP classifier is presented in Table 4.

<table>
<thead>
<tr>
<th></th>
<th>In</th>
<th>Out</th>
</tr>
</thead>
<tbody>
<tr>
<td>In</td>
<td>82</td>
<td>18</td>
</tr>
<tr>
<td>Out</td>
<td>18</td>
<td>82</td>
</tr>
</tbody>
</table>

Table 4. Confusion matrix of the classes indoor and outdoor using MAP classifier. The values denote percentages.

Especially good features for differentiating between these classes were the WLAN signal strengths, which are usually higher indoors than outdoors. By using only the WLAN signal strength we could obtain classification accuracies of 78% and 82% for indoor and outdoor, respectively. Nevertheless, the results may become very different if the data would be collected only from areas where the WLAN station density is low.

By using DT and SVM we could achieve higher classification rates for the indoor and outdoor class. Confusion matrices, where all the features from GPS, WLAN and Bluetooth were used, are presented in Tables 5 and 6 for DT and SVM, respectively. This shows that the decision tree gives the highest classification rate with these classes. However, to test our adaptation method we continued to use MAP algorithm also for these classes.
Table 5. Confusion matrix of the classes *indoor* and *outdoor* using decision trees. The values denote percentages.

<table>
<thead>
<tr>
<th>true \ decision</th>
<th>In</th>
<th>out</th>
</tr>
</thead>
<tbody>
<tr>
<td>In</td>
<td>96</td>
<td>4</td>
</tr>
<tr>
<td>Out</td>
<td>6</td>
<td>94</td>
</tr>
</tbody>
</table>

Table 6. Confusion matrix of the classes *indoor* and *outdoor* using SVM. The values denote percentages.

<table>
<thead>
<tr>
<th>true \ decision</th>
<th>In</th>
<th>out</th>
</tr>
</thead>
<tbody>
<tr>
<td>In</td>
<td>86</td>
<td>14</td>
</tr>
<tr>
<td>Out</td>
<td>6</td>
<td>94</td>
</tr>
</tbody>
</table>

7.2. Environment Classifier

For environment classification we selected the set of {restaurant/pub/cafe, office, home, street/road, nature} as the categories to be recognized. The list includes environments that would be useful for many mobile applications, but it is short enough for reasonable accuracy analysis. In environment classification all the features from the GPS, WLAN and Bluetooth were selected. In addition, the likelihoods output by the audio classifier were used as features. Features from the accelerometer were not used in this case as they did not improve the recognition accuracy.

Figure 5 shows the scatter plot of the first two features after the compression method presented in Section 4 was done. Some of the most important features were audio features and GPS TTFF and maximum WLAN signal strength. The confusion matrix of leave-one-user-out cross-validation method is presented in Table 7.

Figure 5. Two main features after the feature compression.

DT and SVM results for the classes *office, nature, street/road, home* and *restaurant/pub/cafe* are presented in Table 8. As the same classes
Class average classification accuracy for MAP classifier is 70%, for DT 68% and for SVM 63%. Thus, there are no great differences between these classifiers. It should be noted that default parameters from MatLab’s `ClassificationTree class` were used for DT and SVM, in addition, the class imbalance problem was not taken into account for these classifiers, thus by taking these aspects into account, accuracies may slightly increase. However, the main idea in the paper was not to show whether the Bayesian classifier can beat DT or SVM, but to show that our Bayesian classifier is not dramatically inferior to state of art methods and thus it is feasible to propose adaptation algorithm for this Bayesian approach. Comprehensive comparative between classifiers is thus left out.

### Table 7. Confusion matrix of five environment classes office, nature, street/road, home and restaurant/pub/cafe using MAP classifier. The values denote percentages.

<table>
<thead>
<tr>
<th>true \ decision</th>
<th>Off.</th>
<th>Nat.</th>
<th>Str.</th>
<th>Home</th>
<th>Res.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Off.</td>
<td>75</td>
<td>0</td>
<td>2</td>
<td>7</td>
<td>17</td>
</tr>
<tr>
<td>Nat.</td>
<td>0</td>
<td>67</td>
<td>21</td>
<td>5</td>
<td>8</td>
</tr>
<tr>
<td>Str.</td>
<td>1</td>
<td>9</td>
<td>72</td>
<td>6</td>
<td>12</td>
</tr>
<tr>
<td>Home</td>
<td>3</td>
<td>5</td>
<td>15</td>
<td>62</td>
<td>15</td>
</tr>
<tr>
<td>Res.</td>
<td>7</td>
<td>0</td>
<td>21</td>
<td>0</td>
<td>73</td>
</tr>
</tbody>
</table>

### Table 8. Confusion matrix of five environment classes office, nature, street/road, home and restaurant/pub/cafe using the decision tree classifier. The values denote percentages.

<table>
<thead>
<tr>
<th>true \ decision</th>
<th>Off.</th>
<th>Nat.</th>
<th>Str.</th>
<th>Home</th>
<th>Res.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Off.</td>
<td>70</td>
<td>0</td>
<td>1</td>
<td>20</td>
<td>10</td>
</tr>
<tr>
<td>Nat.</td>
<td>2</td>
<td>65</td>
<td>20</td>
<td>6</td>
<td>7</td>
</tr>
<tr>
<td>Str.</td>
<td>2</td>
<td>12</td>
<td>62</td>
<td>11</td>
<td>13</td>
</tr>
<tr>
<td>Home</td>
<td>4</td>
<td>2</td>
<td>9</td>
<td>77</td>
<td>8</td>
</tr>
<tr>
<td>Res.</td>
<td>2</td>
<td>1</td>
<td>26</td>
<td>4</td>
<td>68</td>
</tr>
</tbody>
</table>

### Table 9. Confusion matrix of five environment classes office, nature, street/road, home and restaurant/pub/cafe using the support vector machine classifier. The values denote percentages.

<table>
<thead>
<tr>
<th>true \ decision</th>
<th>Off.</th>
<th>Nat.</th>
<th>Str.</th>
<th>Home</th>
<th>Res.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Off.</td>
<td>87</td>
<td>0</td>
<td>0</td>
<td>5</td>
<td>8</td>
</tr>
<tr>
<td>Nat.</td>
<td>2</td>
<td>36</td>
<td>26</td>
<td>23</td>
<td>13</td>
</tr>
<tr>
<td>Str.</td>
<td>3</td>
<td>13</td>
<td>56</td>
<td>8</td>
<td>20</td>
</tr>
<tr>
<td>Home</td>
<td>2</td>
<td>5</td>
<td>5</td>
<td>73</td>
<td>16</td>
</tr>
<tr>
<td>Res.</td>
<td>18</td>
<td>3</td>
<td>8</td>
<td>7</td>
<td>65</td>
</tr>
</tbody>
</table>

7.3. Activity Classifier

The following classes were included in the activity recognition:
• table; phone is placed on a non-moving surface
• standing; phone is in the hand or pocket, but the user is not walking or running
• walking; the user is walking
• running; the user is running
• bicycling; the user is bicycling
• vehicles; the user is driving or traveling with a motorized vehicle

It is quite clear that the GPS speed would be a very important feature for this kind of classification. However, reliable speed information is not available all the time, so the activity classifier was implemented with two different modes, depending on the availability of the GPS [30].

In this kind of activity classifier, the classes table, standing, and vehicles may overlap in some cases. Reason for this is that the variance of the accelerometer data norm, which is usually the most dominant feature in the activity classifier, can be similar in some cases. For example, a car stopped in traffic lights or a person standing still can be confused as the class on table. Nevertheless, if we also use features from the GPS (GPS speed) and causal information based on, e.g., markov chain modelling of the classification outputs ([31]), the separation of a car stopped in traffic lights and standing can usually be done.

The original data set did not contain enough data for the results with the activity classifier when the GPS speed is available. Thus, data were generated where the standard deviations (std) and means of the GPS speeds for each activity were chosen manually. This made it impossible to perform the decision tree classification but enabled us to calculate the adaptation results also for the activity classifier. The values selected are shown in Table 10. It should be noted that the values are naturally changing depending on the user. However, our adaptation algorithm can scale these values for the individual use.

Table 10. The means and standard deviations of the GPS speeds that were used in the activity classifier.

<table>
<thead>
<tr>
<th></th>
<th>run</th>
<th>bicyc</th>
<th>walk</th>
<th>stand</th>
<th>table</th>
<th>vehic</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean(km/h)</td>
<td>7</td>
<td>15</td>
<td>4</td>
<td>0.5</td>
<td>0.5</td>
<td>60</td>
</tr>
<tr>
<td>Std(km/h)</td>
<td>2</td>
<td>5</td>
<td>2</td>
<td>0.5</td>
<td>0.5</td>
<td>30</td>
</tr>
</tbody>
</table>

7.4. Adaptation Results

For the test persons, carrying the phones and answering yes or no to the recognition results does not require a big effort, but the results show that the information obtained is very valuable in both adaptation and evaluation of the classifier performance. As an example, a scatter plot of the two most important transformed features of the data from the class street/road is shown in Figure 6. In the upper panel, the original 95 percent confidence ellipse (The ellipse containing 95% of the mass of the Gaussian probability density function when using the estimated covariance calculated from the training data) of the street/road training data is depicted using a dashed line, and the ellipse after the adaptation in solid line. The data with yes-tag is plotted with small circles, and the data with no-tag with small crosses.
The bottom panel of Figure 6 is the same for the features three and four. Figure 7 shows the estimated probability density functions \( p(y) \) obtained using Matlab’s `ksdensity` function. The figure shows the yes-tag likelihoods (dash-dotted) and no-tag likelihoods (solid line) before and after the adaptation. The dashed curve is the Chi squared probability density function which would be the optimal distribution for the yes-tag likelihoods. This means that if our model would be correct and our assumption of the Gaussian distributed features would be true, the dash-dotted curve (“yes” distribution) should follow the dashed curve (chi2pdf). One can clearly see that there is not much difference between the yes- and no-likelihoods before adaptation (upper plot in Figure 7). This means that we cannot assign a confidence measure based on these likelihoods alone. The reason is that the original training data was not sufficient to obtain accurate class parameters or the adaptation data is collected in an environment having different features compared to the training data. After the adaptation is completed, the bottom plot in Figure 7 shows a clear improvement: the no-likelihoods are visibly separated from the yes-likelihoods.

**Figure 6.** Class street/road features.

**Figure 7.** Class street/road probability distribution functions.
After the adaptation round is complete, a convenient way to estimate the accuracy of the classifiers is to use the prior probabilities, \( P(\text{user answers yes}) \), using Equation (20). To evaluate accuracy of the proposed adaptation method the data from the very first data collection campaign was used (Section 2.1). Ten users with the most data were selected and the classifier was trained leaving one user out of the training data set at the time. Data from this user was then divided to half, other half containing test set for classifier and other half was used to create virtual yes and no answers using. This enabled us to know also the ground truth of the no answers (which was not available in the second data collection campaign presented in Section 2.2). Table 11 presents the classification data of these ten users before the adaptation was applied. Classifier precision and recall are 50% and 67%, respectively. The results after the adaptation are presented in Table 12. In this case precision is 68% and recall 70%. Naturally, the no answers were more valuable information for the adaptation process. It was noticed that already couple of no answers could improve the class parameters.

### Table 11. Confusion matrix of five environment classes office, nature, street/road, home and restaurant/pub/cafe before adaptation.

<table>
<thead>
<tr>
<th>true \ decision</th>
<th>Off.</th>
<th>Nat.</th>
<th>Str.</th>
<th>Home</th>
<th>Res.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Off.</td>
<td>76</td>
<td>0</td>
<td>2</td>
<td>11</td>
<td>11</td>
</tr>
<tr>
<td>Nat.</td>
<td>0</td>
<td>55</td>
<td>35</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>Str.</td>
<td>3</td>
<td>8</td>
<td>78</td>
<td>0</td>
<td>11</td>
</tr>
<tr>
<td>Home</td>
<td>3</td>
<td>1</td>
<td>17</td>
<td>65</td>
<td>14</td>
</tr>
<tr>
<td>Res.</td>
<td>11</td>
<td>0</td>
<td>30</td>
<td>0</td>
<td>59</td>
</tr>
</tbody>
</table>

### Table 12. Confusion matrix of five environment classes office, nature, street/road, home and restaurant/pub/cafe after adaptation.

<table>
<thead>
<tr>
<th>true \ decision</th>
<th>Off.</th>
<th>Nat.</th>
<th>Str.</th>
<th>Home</th>
<th>Res.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Off.</td>
<td>81</td>
<td>0</td>
<td>1</td>
<td>6</td>
<td>13</td>
</tr>
<tr>
<td>Nat.</td>
<td>0</td>
<td>60</td>
<td>20</td>
<td>5</td>
<td>15</td>
</tr>
<tr>
<td>Str.</td>
<td>3</td>
<td>5</td>
<td>73</td>
<td>5</td>
<td>14</td>
</tr>
<tr>
<td>Home</td>
<td>3</td>
<td>1</td>
<td>9</td>
<td>74</td>
<td>13</td>
</tr>
<tr>
<td>Res.</td>
<td>11</td>
<td>0</td>
<td>25</td>
<td>0</td>
<td>64</td>
</tr>
</tbody>
</table>

### 7.5. Confidence Measure Results

To show that the estimated probabilities of yes-answers actually resemble the true probabilities, a rough visualization is made for the activities in Figure 8 and for the environments in Figure 9. In the figures, all the answers from all users are sorted by the estimated probability, and the results are plotted using a circle if the user answer was yes and with a cross in the case of a no answer. To see if the prediction of probability is approximately working, we can then zoom into some probability level and see if the distribution between the circle (yes) values agrees with the y-axis number. The lower box of Figure 8 shows an example where the predicted probability level of our model is 0.35 and the actual amount of yes answers in the data is 32%. Respectively, the higher box depicts a situation
where our model predicts a proportion of yes answers to be 0.88, whereas the actual proportion of yes labeled samples in the data is 87%. This approximation shows that the estimation is working very well for activities, however, for environments (Figure 9) there are more differences in predicted probability compared to the estimate calculated using yes and no answers.

**Figure 8.** Predicted yes-probability for each sample in the sorted activities data. The small bar plots tell how many yes and no answers there are inside the two small rectangural areas.

**Figure 9.** Predicted yes-probability for each sample in the sorted environment data. The small bar plots tell how many yes and no answers there are inside the two small rectangural areas.

To analyze the result more in detail, we also show the limits $t_l, t_u$ around the mean

$$P(t_l \leq \theta < t_u) = 0.95,$$

with $\theta \sim Bin(N(\text{yes}) + N(\text{no}), p_m)$ for each bin to see how the samples would deviate with perfectly known $p_m$. The limits are shown in Figure 10 for the activities and Figure 11 for the environments,
where the data is divided into 10 boxes having the middle points $p_m = 0.05 \ldots 0.95$. The height of each bin is the number of yes answers satisfying $|p_m - P_{yes}| < 0.05$. This result shows that for the activities the estimated confidence measure is consistent. However, for the environments the confidence measure is slightly optimistic.

**Figure 10.** The number of samples (yes answers) at each bin centered at $p_m$ and 95% limits from Binomial distribution (activities).

**Figure 11.** The number of samples (yes answers) at each bin centered at $p_m$ and 95% limits from Binomial distribution (environments).

8. Conclusions

We have presented an activity and environment recognition implementation suitable for modern mobile phones. Two data collection campaigns were organized, the first for the background training data and the second for testing the adaptation algorithms. The initial design goal of having an adaptive system that can provide confidence measures along with the classification results was met by developing a Bayesian approach that utilizes binary user feedback.

The results show that some important environments and activities can be recognized with reasonable accuracy, but individual adaptation is very likely needed for applications requiring context information. In addition, the adaptation is needed for providing a proper confidence measure for the classification
result. Furthermore, the paper showed that the confidence measure is consistent with the selected set of activities classes. For the selected environments, the confidence measure was slightly optimistic.

Our implementation enables an individual adaptation in a mobile device. To avoid heavy computations on the mobile device, the required distribution estimates can be computed on the server side. The only necessary information to be transmitted to the server would be the set of likelihood values corresponding to observations with a yes-label and a no-label.

Author Contributions

All authors in the group designed the experiments. Jussi Parviainen and Jussi Collin were the main authors who analyzed the data. Jayaprasad Bojja performed the implementation on the mobile phone. Jussi Parviainen was the main writer of the paper, and all of the other authors also edited the paper.

Conflicts of Interest

The authors declare no conflict of interest.

References


© 2014 by the authors; licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution license (http://creativecommons.org/licenses/by/4.0/).
Publication IV


In reference to IEEE copyrighted material which is used with permission in this thesis, the IEEE does not endorse any of Tampere University of Technology’s products or services. Internal or personal use of this material is permitted. If interested in reprinting/republishing IEEE copyrighted material for advertising or promotional purposes or for creating new collective works for resale or redistribution, please go to http://www.ieee.org/publications_standards/publications/rights/rights_link.html to learn how to obtain a License from RightsLink.
Automatic Crash Detection for Motor Cycles

Jussi Parviainen∗, Jussi Collin∗, Timo Pihlström∗, Jarno Takala∗, Kari Hanski† and Aki Lumiaho†
∗Department of Pervasive Computing
Tampere University of Technology, Finland
Email: firstname.lastname@tut.fi
† Ramboll Finland
Tampere, Finland
Email: firstname.lastname@ramboll.fi

Abstract—In this paper we present a concept for automated crash detection to motorcycles. In this concept, three different inertial measurement units are attached to head of the motorist, torso of the motorist and to the rear of the motor cycle. Crash dummy tests are done by throwing the dummy with different altitudes to simulate the effect of crash to the motorist and real data is collected by driving the motorcycle. A maximum a posteriori classifier is trained to classify the crash and normal driving. The implemented prototype system shows promising results for automatic crash detection.

Index Terms—Context awareness, Pattern recognition, Wearable computers, Automotive applications, Vehicle safety.

I. INTRODUCTION

Safety in traffic is taken into consideration more and more nowadays. Nevertheless, accidents still occur and help is usually needed as fast as possible. There has risen a discussion of automatic accident detection methods which would trigger emergency call (eCall) automatically. An example of this is a European eCall initiative1 where in case of a crash, a vehicle automatically calls to the nearest emergency center and sends some data including exact crash location.

In order to trigger eCall, the system needs to detect accidents. The majority of the literature discusses crash detection methods for cars, e.g. [1]–[4]. Crash detection algorithms are usually based on acceleration or crush measurement. For two wheel vehicles, motorcycles, the crash detection algorithms are not thoroughly explored. In the prior art [5]–[9] the motor cycle crash detection is done using the sensors attached to the motorcycle. However, the sensors in the motorcycle may not know the severity of the crash, unlike in the cars, in the case of crash, the motorist and motorcycle may fall away from each other. In addition to triggering eCall, there have been research over motorist safety jackets, which could inflate the airbag during the crash [10], [11].

In this paper, we propose to use additional sensors to improve crash detection and situational awareness. These wearable sensors [12]–[14] are integrated to the motorist suite. Namely, one attached in torso and other one in the helmet (Fig 1). Small wireless GPS synchronized micro-electromechanical MEMS inertial measurement units (IMU) were build in order to collect data. Data from normal motor cycle driving was collected and the crash situation was simulated by dropping the dummy from different altitudes on the floor. Maximum a posteriori (MAP) classifier was build in order to infer between the crash pulse and normal driving.

The paper continues in Section II with a discussion of crash pulse detection and the presentation of our classification algorithm. Next, our measurement devices and training data with the selected features are presented in Section III. In Section IV the results from the collected data are illustrated and Section V presents real time implementation of the classification system. Finally we conclude the paper in Section VI.

II. CRASH DETECTION

It is not straightforward to detect whether the crash has occurred. A known way is to follow the speed information and inclination of the motor cycle [15] or to explore the accelerations of the motor cycle and the motorist. If conventional crash pulse detection is used, where the crash is alerted using vehicle mounted accelerometer, the severity of the crash to the motorcyclist is not necessarily known. For example, when motorist falls from the motorcycle during the crash, only the motorcycle or the motorist may undergo high energy crash. Thus, it would be feasible to distribute the sensors between the motorcycle and the motorist. One option for the sensor locations was given in Fig 1. If the high crash pulse was occurred in the helmet sensor, the probability of the more severe accident is higher. This could help authorities to put more priority to more severe accidents.

In our work we take the pattern classification approach and aim to recognize the following events:

- Static Motorist is not moving
- Move The motorist is driving a motorcycle or e.g. walking
- zeroG The motorist/unit is in free fall (this might happen before crash). A long free fall may implicate severe crash.
- Peak There is high peak in the acceleration data, this usually means crash.

In order to detect when the crash has occurred we build an MAP classifier.

### A. Classification

For the classification task, we use a supervised Bayesian maximum a posteriori classifier (MAP). A supervised classification process starts by collecting a training data set with known states. This set is used to obtain

\[
\mathbf{z}_j \sim N(\mu_j, \Sigma_j),
\]

the distribution of the observed \( q - by - 1 \)-vector \( \mathbf{z} \) given that the observation comes from the class \( j \). At this stage, the mean vector \( \mu_j (q \times 1) \) and the covariance matrix \( \Sigma_j (q \times q) \) are here assumed to be perfectly known (learned from a training set). We also assume that for all classes \( \Sigma_j > 0 \) (i.e. \( \Sigma_j \) is positive definite). If (1) holds, the density function of \( \mathbf{z}_j \) is

\[
f_{\mathbf{z}_j}(\mathbf{z}; \mu_j, \Sigma_j) = \frac{1}{(2\pi)^{q/2} \sqrt{\det(\Sigma_j)}} \exp\left[-\frac{\|\mathbf{z} - \mu_j\|^2}{2\det(\Sigma_j)}\right].
\]

A new observed \( \mathbf{z} = \mathbf{z}_x \) can then be classified by maximizing (2) over all the classes \( j = 1 \ldots p \). To assign a probability to the classification result, we need to use unconditional prior probabilities \( P(C = j) \), and assume that all the possible classes are included. Then the probabilities for all the classes are obtained from the Bayes’ rule:

\[
P(C = j | \mathbf{z}_x) = \frac{f_{\mathbf{z}_x}(\mathbf{z}_x) P(C = j)}{f_{\mathbf{z}_x}(\mathbf{z}_x)},
\]

where \( f_{\mathbf{z}_x} \) is the unconditional density function for the observation \( \mathbf{z} \). Prior probability can be used to control the false alarms or false negatives.

### III. Measurement Device

In order to log and process the data a small wireless GPS synchronized micro-electromechanical MEMS inertial measurement units (IMU) were build. The device was designed for use as a wearable element and therefore, small size and wireless operation is required. Also, the unit should have enough processing power so it can operate the functions of an inertial measurement unit and user-defined algorithms. The device is based on an ARM Cortex M3 microcontroller operating at 72 MHz with 512 KB of program memory and 64 KB of data memory. It provides USB and micro SD card connectivity for data transfer and logging capabilities.

The combined accelerometer and magnetometer is ST Microelectronics LSM303DLHC2 which operates over a wide acceleration sensitivity range of \(+16\) g. The sensor outputs measurements with over 1 kHz rate which is necessary in highly dynamic crash situations. The gyroscope is ST Microelectronics L3GD20 with user-defined sensitivity range from 250 dps to 2000 dps. Both of the sensors are connected to the microcontroller with an I²C bus. Also, they are readily available, inexpensive, and have a small footprint.

Unit contains an u-blox UC530 GPS module whose main purpose is to provide time synchronization for sensor measurements, but it can also be used to track the unit position and speed. The other radio module is Microchip RN41 Bluetooth module which provides wireless connectivity. The Bluetooth protocol was chosen because of its wide use in cellular phones and tablets.

All of the aforementioned functionality is implemented on a single four-layered PCB and packaged inside a 50 mm x 50 mm x 20 mm plastic enclosure alongside a 400 mAh lithium-polymer battery.

### A. Training data Collection

Normal motor cycle driving in traffic including cobblestone road was collected for null hypothesis (approximately 11 minutes). It is costly to arrange real motor cycle crash tests, however, crash pulses were simulated by throwing the crash dummy from two different altitudes 1.7 m and 3 m. Details of the crash dummy tests are given in the Table I. It should be noted that the lateral speed of the dummy was very small during throwing tests. In real life crashes speed would be greatly larger and the forces sensed by the sensors are naturally larger. However, the main difference is in the size of the crash pulse (norm of the acceleration during the crash) and the greater pulse is usually easier to recognize.

In addition, data collection was performed in a motocross track. Data included high jumps as illustrated in Fig. 2. It should be noted that the driving environment is very different in motocross track compared to normal driving in traffic. For example, if the same algorithm is used in both cases, false crash detections may occur more often in motocross track.

### B. Accelerometer features

Before we can use the classification algorithm as it was described in Section II-A, the feature extraction should be done. It is known that excessive number of features can be

| Measurement 1 | Dummy is dropped from 1.70 meters |
| Measurement 2 | Dummy falls from his back |
| Measurement 3 | Dummy falls from his back |
| Measurement 4 | Dummy is dropped from 1.70 meters |
| Measurement 5 | Dummy is dropped from 3 meters (front first) |
| Measurement 6 | Dummy is dropped from 3 meters (back first) |

\[3\text{http://www.st.com/web/catalog/sense_power/FM89/SC1449/PF251940} \]

\[4\text{http://www.st.com/web/catalog/sense_power/FM89/SC1288/PF252443} \]

\[5\text{http://www.u-blox.com/ensetPositioning-antennas/gnss-antenna-modules/uc530.html} \]

\[6\text{http://www.rovingnetworks.com/products/RN41} \]
extracted from accelerometer data [16] and those are also very feasibly for activity recognition [17], [18]. In this work we also concentrate on accelerometer feature based recognition.

If data from each axis of 3D accelerometer is used separately we can get information about the attitude and the inclination of the motor cycle [15]. However, it is possible to infer the states given in Section II only by using the Euclidean norm of the three dimensional accelerometer signal. The following features (Fn) are extracted from the accelerometer signal norm:

- F1: Variance
- F1: Maximum value
- F3: Mean
- F4: The number of zero crossings

In addition if we want to improve especially the detection accuracy of free fall state, we can also add features which are calculated from the full 3D accelerometer data. For example, we derived empirically two features that can be used for zeroG detection even if the motor is running:

- F5: The absolute value of mean of mean of each accelerometer axis
  \[
  F5 = \left| \frac{1}{n} \sum_{i=1}^{n} a_i^x + \frac{1}{n} \sum_{i=1}^{n} a_i^y + \frac{1}{n} \sum_{i=1}^{n} a_i^z \right| / 3 \tag{4}
  \]

- F6: The variance of mean of each accelerometer axis
  \[
  F6 = \text{var} \left( \frac{1}{n} \sum_{i=1}^{n} a_i^x, \frac{1}{n} \sum_{i=1}^{n} a_i^y, \frac{1}{n} \sum_{i=1}^{n} a_i^z \right) \tag{5}
  \]

Here the symbols \(a_i^x, a_i^y\) and \(a_i^z\) are the \(i\)th components of \(x\), \(y\) and \(z\) direction accelerometer readings, respectively, where the total number of samples in sliding window is \(n\) and \(\text{var}(\cdot)\) is variance operator. In this work we chose some what heuristically the length of the sliding window to be around 50 ms. It should be noted that features F5 and F6 are orientation depended. Thus, these features should be trained in many different orientations. One possible approach is to multiply the data with direction cosine matrices (DCM) having different orientations and make replicas of the training data having different orientations.

As we are using data from three different units it is possible to combine features as a large feature vector (FV) from all devices
\[
FV = [F_{1\text{unit1}} \cdots F_{6\text{unit1}} F_{1\text{unit2}} \cdots F_{6\text{unit2}} F_{1\text{unit3}} \cdots F_{6\text{unit3}}] \tag{6}
\]
and then train a classifier. Another option is to train all three units separately and then combine the output of classifiers.

In addition to classifier presented in Section II-A, it would be feasible to use time filtering, for example Hidden Markov models, as it was done in [2] for the vehicle crash detection.

### IV. RESULTS

In order to visualize the data more clearly the features compression method based on Karhunen-Loève transformation given in [19, pp. 331-334] was implemented. The method is highly similar to linear discriminant analysis (LDA), which is widely used in feature compression. After the transformation the number of features is reduced to \(C - 1\), where \(C\) is the number of classes. These transformed features are linear combinations of the original features. Three best linear combinations of training data features are shown in Fig. 3 which were produced after this compression method. These features were then used to train the classifier as it was described in Section II-A.

Fig. 4 illustrates the states of one crash dummy test with the classified states (above) and norm of the acceleration (below). Classifier can successfully find all the states during the dummy test. First the dummy is in move state as the dummy was hold by two persons. Next in the data we can see zeroG state where the dummy is in free fall and after this the peak state implicates the crash on the floor. Finally dummy immobile in static state.

The accelerometer data in the motorcycle tests was just used to find parameters for the normal driving phase when the crash peaks should not occur. As was mentioned the real motorcycle crash situations were not performed. Fig. 5 shows
the raw data the cobblestone road drive. In the beginning of the figure motorcycle was stopped in the traffic lights, after that it continued driving where we can see great increase in variance. We can see directly from the figure that the different locations of the accelerometer senses different amount of vibrations during driving in the bumpy road. Thus, right amount of the training data from the each location is needed in order to get good training parameters for the classifiers; false crash alarms should not occur in any normal driving environments.

V. Real Time Implementation

The feature extraction and MAP classifier was implemented on the measurement device with the help of FreeRTOS \(^7\) tasks and queues. Two of the tasks are responsible for the operations of an inertial measurement unit. Namely, reading the sensors, calibrating the measurements and adding them into a buffer structure. The inter-task and outside world communications are based on a message system with one central communications task. All inter-task messaging and movement of the sensor data is done using thread-safe queues.

The classifier algorithm has its own task where it collects 50 accelerometer measurements into a buffer. The inertial measurement unit tasks provide these readings at a 1 kHz rate. In addition, the result of the algorithm is sent over Bluetooth using the communications task functions.

For the visualization of the algorithm data and automatic detection of a selected states an Android phone is used. The application opens a Bluetooth connection to each of the measurement units and sends a Start measurements message. Afterwards, the algorithm tasks of each device start receiving accelerometer measurements and send the results to the Android phone. The results are shown in a graph on the phone screen, this is illustrated in Fig. 6. Each green dot in the screen means one classification result which are updated every 50 ms. In the figure it is shown a case when the device was taken into the hand was then thrown up and caught back to hand. States are in same order as in 4 above plot.

VI. Conclusion

In this paper we presented a new concept for automated method for crash detection to motorcycles. In contrast to conventional systems, where the sensors are attached to the motor cycle, we propose a system with three different inertial measurement units were attached to head, torso of the motorist and to the rear of the motor cycle. Crash dummy tests were done by throwing the dummy with different altitude to simulate the effect of crash to the motorist and real data is collected by driving the motorcycle. A MAP classifier was trained to make decision between the crash and normal driving. When the crash occurs automated system can send automated emergency call.

\(^7\)http://www.freertos.org
and the position of the motorist. In addition to these, data containing information of the severity of the incident could be sent.

In order to really validate our classifier the real data with motor cycle crashes would be needed. However, these preliminary results show that the classifier could be applicable in real situations. In future work we will investigate how to optimally combine data from several spatially distributed inertial measurement units.

REFERENCES


