Efficient Positioning Methods and Location-Based Classification in the IP Multimedia Subsystem

Dissertation

zur Erlangung des Grades eines
Doktor-Ingenieurs
der Fakultät für Elektrotechnik und Informationstechnik
an der Ruhr-Universität Bochum

von
Yuheng He
aus Shanghai

Bochum 2013
Contents

Abbreviations 3

1. Introduction 7
   1.1. IP Multimedia Subsystem ................................. 7
   1.2. Location Based Services ................................. 8
   1.3. Positioning ................................................. 9
   1.4. Contribution and Related Work .......................... 10
      1.4.1. Positioning .......................................... 10
      1.4.2. Location Based Services ............................ 12
      1.4.3. IP Multimedia Subsystem ............................ 13
   1.5. Structure of the Thesis .................................. 14

2. Technical Background 17
   2.1. GPS and DGPS Positioning Methods ..................... 17
      2.1.1. GPS Positioning ...................................... 17
      2.1.1.1. Observation Equation .............................. 18
      2.1.1.2. Applying Least-Squares Method for Position Esti-
               mation .............................................. 21
      2.1.2. Differential GPS Positioning ........................ 22
      2.1.3. Recursive DGPS ....................................... 26
   2.2. Location Based Services (LBS) .......................... 27
      2.2.1. Classifying LBS .................................... 28
      2.2.2. Service Categories and Characteristics .......... 29
      2.2.3. Updating Database .................................. 30
      2.2.4. Implementation for LBS ............................ 30
   2.3. IP Multimedia Subsystem (IMS) .......................... 31
      2.3.1. IMS Architectural Requirements .................... 32
      2.3.1.1. IP Multimedia Sessions .......................... 32
      2.3.1.2. Ensuring Quality of Service for IMS ............ 32
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.3.1.3. Secure Communication</td>
<td>32</td>
</tr>
<tr>
<td>2.3.1.4. IMS Layer Design and Stack Basics</td>
<td>33</td>
</tr>
<tr>
<td>2.3.1.5. Transport Layer</td>
<td>35</td>
</tr>
<tr>
<td>2.3.1.6. Session Control Layer/IMS Layer</td>
<td>35</td>
</tr>
<tr>
<td>2.3.1.7. Application Layer/Service Layer</td>
<td>35</td>
</tr>
<tr>
<td>2.3.2. IMS Core Network</td>
<td>35</td>
</tr>
<tr>
<td>2.3.2.1. Call Session Control Function</td>
<td>35</td>
</tr>
<tr>
<td>2.3.2.2. Home Subscriber Server</td>
<td>36</td>
</tr>
<tr>
<td>2.3.3. Access Network</td>
<td>37</td>
</tr>
<tr>
<td>2.3.4. Other Elements in IMS Network</td>
<td>37</td>
</tr>
<tr>
<td>2.3.5. IMS Network Protocols</td>
<td>38</td>
</tr>
<tr>
<td>3. Scalable Positioning Methods</td>
<td>39</td>
</tr>
<tr>
<td>3.1. Approximate Solution of LS Problems</td>
<td>42</td>
</tr>
<tr>
<td>3.1.1. Givens Rotation</td>
<td>42</td>
</tr>
<tr>
<td>3.1.2. CORDIC-Based Approximate Rotation</td>
<td>43</td>
</tr>
<tr>
<td>3.1.3. CORDIC-Based Approximate Rotation</td>
<td>43</td>
</tr>
<tr>
<td>3.1.4. QR Decomposition (QRD)</td>
<td>45</td>
</tr>
<tr>
<td>3.1.5. Approximate QRD</td>
<td>46</td>
</tr>
<tr>
<td>3.1.6. Applying the Approximate QRD to GPS/DGPS</td>
<td>46</td>
</tr>
<tr>
<td>3.1.7. Applying the Approximate QRD to Recursive DGPS</td>
<td>47</td>
</tr>
<tr>
<td>3.2. Experimental Results</td>
<td>48</td>
</tr>
<tr>
<td>3.2.1. GPS Data</td>
<td>49</td>
</tr>
<tr>
<td>3.2.1.1. Rinex Data</td>
<td>49</td>
</tr>
<tr>
<td>3.2.1.2. Raw Data</td>
<td>49</td>
</tr>
<tr>
<td>3.2.2. Positioning Results</td>
<td>50</td>
</tr>
<tr>
<td>3.2.2.1. Experimental Results for GPS</td>
<td>50</td>
</tr>
<tr>
<td>3.2.2.2. Experimental Results for DGPS</td>
<td>58</td>
</tr>
<tr>
<td>3.2.2.3. Experimental Results for Recursive DGPS</td>
<td>59</td>
</tr>
<tr>
<td>3.3. Conclusion</td>
<td>59</td>
</tr>
<tr>
<td>4. Learning Geographic Regions Using Location Based Services in IMS</td>
<td>61</td>
</tr>
<tr>
<td>4.1. IMS-for-Vehicle Architecture</td>
<td>62</td>
</tr>
<tr>
<td>4.1.1. Local Network Access</td>
<td>62</td>
</tr>
<tr>
<td>4.1.2. Global Network Access</td>
<td>63</td>
</tr>
<tr>
<td>4.2. Classification and Filter Functions</td>
<td>63</td>
</tr>
<tr>
<td>4.2.1. TF-CAS-LF</td>
<td>64</td>
</tr>
</tbody>
</table>
## Contents

4.2.1. Type Filter ............................................. 65  
4.2.1.2. Classification ....................................... 66  
4.2.1.3. Location Filter ....................................... 69  
4.2.2. LF-CAS-TF ............................................. 69  
4.3. Design and Architecture of the Presented Concept ................. 70  
4.3.1. IMS Location Server (IMLS) ................................ 70  
4.3.2. Group List Manage Server (GLMS) .......................... 71  
4.3.3. Classification Application Server (CAS) ..................... 72  
4.4. Implementation and Experimental Results ......................... 72  
4.4.1. Analyzing the performance of the classification methods .... 73  
4.4.2. Classification .......................................... 76  
4.5. Conclusion .............................................. 77  

5. Implementation on IMS Platform .................................. 87  
5.1. Internal Package Transmission Delay for Registration and Session Setup ........................................... 88  
5.1.1. Call Setup Description ..................................... 88  
5.1.1.1. Traditional Network (T-Net) ............................. 89  
5.1.1.2. IMS-Net .............................................. 90  
5.1.2. Implementation Environment ................................ 92  
5.1.2.1. General Setup Environment ............................. 92  
5.1.2.2. T-Net Environment ..................................... 93  
5.1.2.3. IMS-Net Environment ................................... 93  
5.1.2.4. Measurement Method .................................... 94  
5.1.3. Test Results ............................................ 94  
5.1.3.1. VoIP Registration ...................................... 95  
5.1.3.2. VoIP Session ......................................... 96  
5.2. Establish Application Servers in IMS .......................... 98  
5.2.1. Communication Establishment and Possibilities ............... 98  
5.2.1.1. Communication Establishment ............................ 98  
5.2.1.2. Communication Possibilities ............................ 100  
5.2.2. Experimental Results ..................................... 102  
5.2.2.1. Time Delay of Data Exchange between Clients (C-C) in T-Net and IMS-Net ........................................ 103  
5.2.2.2. Time Delay of Data Exchange from Clients to Servers (C-S) in T-Net and IMS-Net ........................................ 103
5.2.2.3. Time Delay of Data Exchange from Servers to Clients
(S-C ) in T-Net and IMS-Net . . . . . . . . . . . . 105
5.2.2.4. Comparing the Packet Flow for all Cases in IMS-Net 105
5.3. Conclusion . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 107

6. Conclusion and Outlook 109

A. Basic Linear Algebra 113
A.1. Vectors and Matrices . . . . . . . . . . . . . . . . . . . . . . . . . . 113
A.2. Orthogonal Bases . . . . . . . . . . . . . . . . . . . . . . . . . . . . 113
A.3. The Least Squares Problem . . . . . . . . . . . . . . . . . . . . . 114
A.4. Solving the Least Squares Problem . . . . . . . . . . . . . . . . . 115
A.4.1. Normal Equations . . . . . . . . . . . . . . . . . . . . . . . . 115
A.4.2. QR Decomposition . . . . . . . . . . . . . . . . . . . . . . . . 115

Zusammenfassung 137

Publications 139

Curriculum Vitae 141

Acknowledgments 143
Summary

Positioning Methods and Location Based Services (LBS) in the IP Multimedia Subsystem (IMS) are the topics of this thesis. First GPS/DGPS positioning methods with scalable positioning accuracies are presented. Classification methods are used for learning geographic regions based on the location information and the states of the users gathered in a database. Furthermore, the implementation of a Classification Application Server (CAS) in the IMS Network is presented, where by the time delays of the servers in the IMS Network are also discussed.

GPS-based techniques are commonly used position estimation methods. The efficient implementation of GPS and DGPS positioning methods using COordinate Rotation DIgital Computer (CORDIC)-based approximate rotations for solving the occurring Least Squares (LS) problems is discussed. By changing the number of optimal CORDIC angles (approximation accuracy), different accuracies of the positioning results can be realized. The accuracy of the positioning results is compared for varying approximation accuracies using Rinex files from a high-end GPS receiver and raw GPS data from a low-end GPS receiver. The experimental results show that coarse approximations are sufficient for obtaining the positioning accuracies required by different Location Based Services (LBS). The presented positioning methods with scalable accuracies allow cost- and power-efficient software/hardware implementations for both GPS and DGPS positioning.

Assuming information in local network (cellular network) can be freely exchanged with global IP network (IMS) the position estimates as well as other states of the users are gathered in a database. Using this database statistical classification methods are applied to learn geographic regions. Two cases are distinguished:

- Learning the geographic regions in which certain events happen. Depending on the information provided by the users, they are divided into different user groups (state classes) using Type Filters. Then, classification methods are applied to the position information to determine the geographic regions of the different classes.
• Learning events that happen inside certain geographic regions. The observed area is divided into different geographic regions (location classes) using Location Filters. Then, classification methods are applied to determine patterns of behavior in these regions. Both classifications are done in a CAS established in the IMS. The results of the classifications can be used to establish services for this region or for other regions over IMS.

The network provider needs a network server which can offer a high data rate and low latency communication processing for multimedia services. Therefore, the delays of the IMS Network are of interest. The registration and session setup delay through the servers of the IMS Network (packet-based) and the Traditional Network (circuit-based) are compared without considering the end-to-end packet transmission over the wired or wireless link. Using Voice over IP (VoIP) as an example, the measurements show that compared to the Traditional Network IMS offers shorter session setup delays. This is beneficial for real-time applications. Although the registration time in IMS is longer than in the Traditional Network, IMS demands less call session setup delay. This means that in a multimedia communication IMS can offer advanced services with decreased session setup delay. This is a benefit not only to the IMS clients but also to the IMS Network provider.

As a platform for IP-based applications, IMS provides the possibility to develop applications fast and easy. Different servers can be implemented in IMS to offer different services. The architecture for integrating a CAS in IMS is presented. A CAS uses the information in the database to determine geographic regions directly in the IMS Network. A CAS is shared between users in the local cellular network and the global IMS Network. The presented architecture using Client-to-Server (C-S) and Server-to-Client (S-C) communications (session based communication) has lower time delay for packet transmission than Client-to-Client (C-C) communication (message based communication). Furthermore, it is shown that compared to the Traditional Network, the IMS Network can offer better data transmission services (after session setup) for CAS and other real-time applications.
## Abbreviations

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>3GPP</td>
<td>3rd Generation Partnership Project</td>
</tr>
<tr>
<td>AAA</td>
<td>Accounting, Authentication, Authorization</td>
</tr>
<tr>
<td>ADSL</td>
<td>Asymmetric Digital Subscriber Line</td>
</tr>
<tr>
<td>AGPS</td>
<td>Assisted GPS</td>
</tr>
<tr>
<td>AKA</td>
<td>Authentication and Key Agreement</td>
</tr>
<tr>
<td>AS</td>
<td>Application Server</td>
</tr>
<tr>
<td>BGCF</td>
<td>Breakout Gateway Control Function</td>
</tr>
<tr>
<td>CAN</td>
<td>Carrier Access Networks</td>
</tr>
<tr>
<td>CAS</td>
<td>Classification Application Server</td>
</tr>
<tr>
<td>Cell-ID</td>
<td>Cell sector IDentification</td>
</tr>
<tr>
<td>C-C</td>
<td>Client-to-Client</td>
</tr>
<tr>
<td>C-S</td>
<td>Client-to-Server</td>
</tr>
<tr>
<td>CORDIC</td>
<td>COordinate Rotation DIgital Computer</td>
</tr>
<tr>
<td>CSCF</td>
<td>Call Session Control Function</td>
</tr>
<tr>
<td>CSV</td>
<td>Comma Separated Volumes</td>
</tr>
<tr>
<td>DGPS</td>
<td>Differential GPS</td>
</tr>
<tr>
<td>E-OTD</td>
<td>Enhanced Observation Time Difference</td>
</tr>
<tr>
<td>GLMS</td>
<td>Group List Manage Server</td>
</tr>
<tr>
<td>GPS</td>
<td>Global Positioning System</td>
</tr>
<tr>
<td>GPRS</td>
<td>General Packet Radio Service</td>
</tr>
<tr>
<td>HSS</td>
<td>Home Subscriber Server</td>
</tr>
<tr>
<td>HTTP</td>
<td>HyperText Transfer Protocol</td>
</tr>
<tr>
<td>IANA</td>
<td>Internet Assigned Numbers Authority</td>
</tr>
<tr>
<td>I-CSCF</td>
<td>Interrogating-Call Session Control Function</td>
</tr>
<tr>
<td>IETF</td>
<td>Internet Engineering Task Force</td>
</tr>
<tr>
<td>IMS</td>
<td>IP Multimedia Subsystem</td>
</tr>
<tr>
<td>IMSLS</td>
<td>IMS Location Server</td>
</tr>
<tr>
<td>IMS-Net</td>
<td>IMS Network</td>
</tr>
<tr>
<td>IP</td>
<td>Internet Protocol</td>
</tr>
<tr>
<td>Abbreviation</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>-------------</td>
</tr>
<tr>
<td>ISDN</td>
<td>Integrated Services Digital Network</td>
</tr>
<tr>
<td>ISO</td>
<td>International Organization for Standardization</td>
</tr>
<tr>
<td>ISP</td>
<td>Internet Service Provider</td>
</tr>
<tr>
<td>LBS</td>
<td>Location Based Services</td>
</tr>
<tr>
<td>LF</td>
<td>Location Filter</td>
</tr>
<tr>
<td>LS</td>
<td>Least Squares</td>
</tr>
<tr>
<td>LTE</td>
<td>Long Term Evolution</td>
</tr>
<tr>
<td>MGCF</td>
<td>Media Gateway Control Function</td>
</tr>
<tr>
<td>MGW</td>
<td>Media Gateway</td>
</tr>
<tr>
<td>MRF</td>
<td>Media Resource Function</td>
</tr>
<tr>
<td>NGN</td>
<td>Next Generation Networks</td>
</tr>
<tr>
<td>MPE</td>
<td>Mean Positioning Error</td>
</tr>
<tr>
<td>OSA</td>
<td>Open Service Access</td>
</tr>
<tr>
<td>OSI</td>
<td>Open Systems Interconnection</td>
</tr>
<tr>
<td>OTDOA</td>
<td>Observed Time Difference of Arrival</td>
</tr>
<tr>
<td>P-CSCF</td>
<td>Proxy-Call Session Control Function</td>
</tr>
<tr>
<td>PE</td>
<td>Positioning Errors</td>
</tr>
<tr>
<td>PDA</td>
<td>Personal Digital Assistants</td>
</tr>
<tr>
<td>PDE</td>
<td>Position Determination Equipment</td>
</tr>
<tr>
<td>PDF</td>
<td>Policy Decision Function</td>
</tr>
<tr>
<td>POI</td>
<td>Points of Interest</td>
</tr>
<tr>
<td>PS</td>
<td>Positioning Server</td>
</tr>
<tr>
<td>PSTN</td>
<td>Public Switched Telephone Network</td>
</tr>
<tr>
<td>QoS</td>
<td>Quality of Service</td>
</tr>
<tr>
<td>QRD</td>
<td>QR Decomposition</td>
</tr>
<tr>
<td>RDS</td>
<td>Radio Data System</td>
</tr>
<tr>
<td>RFC</td>
<td>Request For Comments</td>
</tr>
<tr>
<td>RTP</td>
<td>Real-time Transport Protocol</td>
</tr>
<tr>
<td>SBC</td>
<td>Session Border Controller</td>
</tr>
<tr>
<td>S-C</td>
<td>Server-to-Client</td>
</tr>
<tr>
<td>S-CSCF</td>
<td>Serving-Call Session Control Function</td>
</tr>
<tr>
<td>SDPE</td>
<td>Standard Deviation of the Positioning Errors</td>
</tr>
<tr>
<td>SER</td>
<td>SIP Express Router</td>
</tr>
<tr>
<td>SIP</td>
<td>Session Initiation Protocol</td>
</tr>
<tr>
<td>SIP-AS</td>
<td>SIP Application Servers</td>
</tr>
<tr>
<td>SSF</td>
<td>Service Switching Function</td>
</tr>
<tr>
<td>Acronym</td>
<td>Full Form</td>
</tr>
<tr>
<td>---------</td>
<td>-----------</td>
</tr>
<tr>
<td>SVN</td>
<td>SubVersioN</td>
</tr>
<tr>
<td>TAS</td>
<td>Telephony Application Server</td>
</tr>
<tr>
<td>TCP</td>
<td>Transport Control Protocol</td>
</tr>
<tr>
<td>TDM</td>
<td>Time-division Multiplexing</td>
</tr>
<tr>
<td>TF</td>
<td>Type Filter</td>
</tr>
<tr>
<td>T-Net</td>
<td>Traditional Network</td>
</tr>
<tr>
<td>UDP</td>
<td>User Datagram Protocol</td>
</tr>
<tr>
<td>UE</td>
<td>User Equipment</td>
</tr>
<tr>
<td>UMTS</td>
<td>Universal Mobile Telecommunications System</td>
</tr>
<tr>
<td>URI</td>
<td>Uniform Resource Identifier</td>
</tr>
<tr>
<td>VoIP</td>
<td>Voice over IP</td>
</tr>
<tr>
<td>V2N</td>
<td>Vehicle to Network</td>
</tr>
<tr>
<td>WLAN</td>
<td>Wireless Local Area Network</td>
</tr>
</tbody>
</table>
1. Introduction

1.1. IP Multimedia Subsystem

The 3rd Generation Partnership Project 3GPP defines an all IP-based wireless network as compared to the historically disparate voice, data, signaling, and control network elements. Therefore, an all IP-based architecture which integrates IP with wireless technologies, has been proposed [PNKM06]. The IP Multimedia Subsystem (IMS) [VWF07][Fra08] is an architectural framework for delivering IP-based multimedia services to users across wireless and wireline terminals [UKCGT07]. Fig. 1.1 shows the global structure of IMS. The main layers of the IMS-based architecture are Transport Layer, IMS Layer and Application Layer. More details can be found in section 2.3. IMS uses both packet switched and circuit switched domains, so it offers network access independently of the different wireless communication standards (GPRS, UMTS, LTE...) [XZZ07][EMG05][LCHW05]. As an open platform, IMS offers network controlled multimedia services [Gou07] and provides new person-to-person multimedia communication services [GM07][LL07]. Due to its standard architecture, the IMS allows a faster deployment of new services and eases service provisioning [MSKK06][KG07][Pav07]. As part of Next Generation Networks (NGN) which are supposed to be all IP-based networks, IMS is taken by 3GPP to reduce operational cost and to provide converged services to its customers [CMVE06].

With the increasing computing power as well as the evolving high speed wireless networking technology (e.g. 3GPP Long Term Evolution (LTE) [LZZ+06][CY07] providing significantly higher data rates, reduced control and user plane latency), many new applications become possible for different wireless services. Wireless services are no longer limited to just voice communication but comprise many other applications such as Internet, multimedia services or vehicle-to-network communication. For these applications, IMS offers a lot of services, like presence, messaging, quality of service, charging and security [BLMR07][YC08][KGSC07].
1.2. Location Based Services

A lot of applications (single executable applications) and services (the applications serving other applications) in IMS can be supported by Location Based Services (LBS) [MSKK06, PM07] as shown in Fig. 1.1. LBS are wireless ‘mobile content’ services which are used to provide location-specific information to mobile users moving from location to location [SV04, Kue05]. LBS are used in military or government industries, emergency services and the commercial sector. A lot of applications of LBS are specified for daily life, in order to meet the requirements of various business and consumers, e.g. taxi companies, police, airports, emergency and tracking [KH07].

Vehicles are also on the edge of becoming one of the biggest mobile users, as they demand more and more wireless applications [Var04]. There are a lot of
ideas and applications for vehicles from different manufactures or groups in the market. For example, getting the fastest route, finding the nearest gas station or car services. Almost all of these applications can be supported by LBS. Currently, GPS techniques [DR01], network positioning methods [DMS98] as well as other positioning methods [HHGC08] can be utilized for location estimation for LBS. Each positioning method has its own positioning accuracy. The required level of positioning accuracy depends very much on the services.

Usually location information is obtained in the user equipment, e.g., sensors in case of inertial navigation or GPS [GWA07]. However, LBS do not simply mean that they only provide mobile users the coordinate position information, e.g., telling the users where they are or which routes should they follow. For example, in many circumstances drivers know their routes, for example, driving everyday from the home to the office and the way back, or visiting their parents in other cities on the weekend. Therefore, navigation is used just as one part of location based services in everyday life. There are more and more cars on the road, which leads to bad traffic scenario every day. How to change the route at the right times? How to control the speed on roads? These kind of questions appear more and more frequently and can not be solved in cars locally. Thus, LBS are employed for large-scale services helping drivers from the outside of the cars. One of the most significant questions is how to utilize the location information in the network to help controlling or improving the traffic conditions.

1.3. Positioning

Widely expanding requirements of LBS [MSKK06][LNES10] demand different positioning accuracies, different platforms and technologies. For example, searching a nearest gas station while driving requires low positioning accuracy of around 100m; and looking for an ATM while walking in a city needs high positioning accuracy of around 10m [CY07][BDP04]. Therefore it is difficult to meet all expectations of the users with low overall implementation costs, especially in case of the Global Positioning System (GPS) [DF03][SK06][EKR88]. Particularly in computational and energy resources limited devices like smart-phones, the development of LBS-oriented applications becomes more and more dynamic and strategically important. Thus, efficient implementations are required.

Using the base stations of the cellular network as reference receivers, Assisted GPS (AGPS) [DR01][Sch04] and Differential GPS (DGPS) [MSB06][Sch07] can be
used. DGPS is usually used, when a high positioning accuracy is required.

Besides the GPS and GPS-based techniques (DGPS, AGPS) [AS09] [CH10], network positioning methods (Cell sector IDentification (Cell-ID)), Enhanced Observation Time Difference (E-OTD), Observed Time Difference of Arrival (OTDOA)) [DMS98] [KPKV07] as well as other positioning methods [CH94] are all commonly used location estimation methods for LBS [JSL03]. Each positioning method has its own positioning accuracy and requires its own software/hardware implementation. The properties of these positioning methods are summarized in Tab. 1.1.

<table>
<thead>
<tr>
<th></th>
<th>Accuracy</th>
<th>Roaming</th>
<th>Implementation cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cell-ID</td>
<td>Low</td>
<td>Excellent</td>
<td>Negligible</td>
</tr>
<tr>
<td></td>
<td>(200m – 20km)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>E-OTD</td>
<td>Average</td>
<td>Poor</td>
<td>Medium</td>
</tr>
<tr>
<td></td>
<td>(100m – 500m)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>OTDOA</td>
<td>Average</td>
<td>Poor</td>
<td>Medium</td>
</tr>
<tr>
<td></td>
<td>(40m – 150m)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>GPS</td>
<td>high</td>
<td>Excellent</td>
<td>High</td>
</tr>
<tr>
<td></td>
<td>(10m – 30m)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DGPS</td>
<td>Very high</td>
<td>Excellent</td>
<td>High</td>
</tr>
<tr>
<td></td>
<td>(10cm – 15m)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 1.1.: The performance of different positioning methods [JSL03]

Radio network positioning methods [Zha02] [VEBC09] require low implementation costs, e.g. concerning computational complexity and power consumption, but have low positioning accuracies. On the contrary, GPS and DGPS provide high and very high positioning accuracies, but their implementation costs are comparably high [SSCW04].

1.4. Contribution and Related Work

1.4.1. Positioning

Positioning Systems are well established in many applications and find increasing usage in mobile communication systems with exponential growth in smart-phones. Especially smart-phones based on so called Open Operating Systems generate a
large growth in applications using LBS. Providing LBS in portable mobile devices has been defined as one of the basic tasks for network providers. However, the current portable mobile devices have fairly restricted computational and energy resources. These limitations dictate the need for efficient main memory spatial processing algorithms and intelligent user interfaces as well as efficient implementations of the positioning system. However, the widely varying requirements of LBS demand different positioning accuracies, such that different platforms, technologies and positioning methods are used. Here, the efficient implementation of a unified GPS-based positioning method providing scalable positioning accuracy is presented.

The efficient implementation is presented for Global Positioning System (GPS) and Differential GPS (DGPS) using approximations of the required Least Squares (LS) solvers. The commonly used algorithm for position computation is triangulation using a non-linear LS method. For GPS, the non-linear system of equations is converted into an iterative procedure, which requires the solution of a linear LS problem in each iteration. In case of DGPS, the problem can be directly solved by a linear LS problem. In both cases the occurring linear LS problems are solved by the QR Decomposition (QRD) using COordinate Rotati

One important application of the proposed method is to adapt GPS positioning accuracy to network positioning accuracy. If the required network information is not available, network positioning can be replaced by approximate GPS. This achieves lower computational complexity and lower power consumption than using conventional GPS. Another important application of the proposed method is to adapt DGPS positioning accuracy to GPS positioning accuracy, i.e. GPS accuracy is obtained by approximate DGPS. Thus, if a connection to the network is available (e.g. a cellular network or Radio Data System (RDS) can be taken as network connection), GPS can be replaced by approximate DGPS. Again, this achieves a significant reduction in computational complexity and power consumption compared to using GPS.

By changing the number of optimal CORDIC angles (approximation accuracy), different accuracies of the positioning results can be obtained. The implementa-
tion enables the adaptation to different accuracy requirements of LBS with only one positioning method. It can be realized on a single platform and promises cost-efficient solutions for LBS applications. The proposed method enables a dynamically in-the-field scalable positioning accuracy and scalable power consumption, such that only the required accuracy for the use-case is delivered and only the minimal effort is used to generate this accuracy. Therefore, the battery life time can be significantly prolonged for LBS usage.

1.4.2. Location Based Services

IMS is all IP-based and the information of the users in the IMS Network can be exchanged freely. The information can be collected in IMS databases, which are accessible by the application servers in IMS. Furthermore, position information is provided for LBS, such that the position of the users is also available in the database. Using the database classification methods can be established. The classification is exemplarily discussed for a vehicular environment.

When the position information of vehicles is pushed frequently to a location server located in the network, the traffic observer and controller can obtain traffic information automatically and persistently all the time. Thus, an IMS Location Server (IMSLS) [HB09a][HB09b][HB10] is established on the IMS Network platform and the on-road vehicle location information is pushed to it. Self-positioning method, which estimate locations directly in the device, e.g. GPS, can be utilized for location estimation in LBS. However, GPS location information is only available for a part of the vehicles of the traffic. For the remaining part of the traffic, network positioning [DMS98] is available, since the mobile phone of the driver is usually powered on all the time.

In this thesis, a Classification Application Server (CAS) [McL04] is implemented in the IMS to figure out traffic situations in different geographic regions [HB09b], e.g. the weather condition, the speed of cars, traffic jams, etc. After that the information stored in IMS can be shared with all the vehicles who are IMS subscribers. This kind of service can also be used to yield traffic scenario mirrors for the drivers who are in the local region or to get traffic forecasts for the drivers who are going to enter this region. The up-to-date information learned in geographic regions supporting can be used to establish services for this region or for other regions over IMS. The presented concept, which is exemplarily applied to a vehicular scenario in this thesis, can also be applied to other scenarios with location-based events.
1.4. Contribution and Related Work

1.4.3. IP Multimedia Subsystem

IMS offers the possibility to combine services provided by different sources, e.g. positioning, LBS, classification, sensor information. An important topic is to investigate the session setup delays, which is directly related to the expected quality of services. This thesis investigated the session setup delays for VoIP service, since for VoIP applications there are a lot of available server- and client-softwares. The results can be, of course, applied to other services like LBS, for which the required software still needs to be developed.

One of the main application of IMS is that IMS can combine ongoing voice sessions with multimedia elements (sharing video while talking, for example) or enrich shared applications with voice communication (for instance, talking while playing a multiplayer game). This has resulted in an increasing popularity of VoIP communications, which have been deployed rapidly in the wireline segment recently and are expected to have a similar impact on wireless networks. The ubiquity of IP makes IMS a convenient platform for launching voice traffic [MM02]. There is a great deal of interest to improve the effectiveness of Voice over IP (VoIP) services over wireless [FNMS08]QVY+08, because VoIP offers significant cost savings while providing more flexible and advanced features. The proliferation of VoIP calls has also introduced significant implications on the security and privacy aspects of phone calls [Cho07][BLG07][CWJ06]. VoIP quality has been studied a lot [LZ06][SLPP06][JCG04], but session setup time for VoIP has received relatively less attention. The session setup time has a direct impact on the users’ satisfaction, so the session setup delay should be optimized. In [FCP06], SIP session setup delay was analyzed and it was proposed to optimize it by using an adaptive retransmission. The end-to-end time delay measurements done by [VW06] show how IMS supports different connections of User-Endpoints over various wireless networks. However, the time delay for end-to-end performance depends not only on network server, but also on other processes as e.g. signal processing in clients, transmission delay over wired or wireless links. Therefore, the time delay of the signal flow in the server can not be clearly observed, and the signal flow delay on the network server was only roughly estimated by [VW06] (an average of 14 ms) from end-to-end delay measurements. Of course, this is very small such that the clients might not notice it and for this reason it was not in the focus of [VW06]. However, due to today’s numerous applications in the network and the resulting complexity, the load of the server is an important issue. The network provider needs a network server which can offer a high date rate and low latency
communication processing for multimedia services.

This thesis presents that the IMS server has shorter call session setup delay compared to Traditional Network. And it shows that compared to the current system IMS can offer faster data transmission services to real-time applications which benefits not only the IMS clients but also the IMS Network provider. The work focuses on the registration and call session setup delay through the servers of both networks, without considering the end-to-end packet transmission over the wired or wireless link. The results show that registration time in IMS is longer than in Traditional Network because of the special structure of IMS core, but it demands shorter call session setup delay. This means that in a keep-alive-communication, such as real-time streaming applications for voice, video, or location over IP, IMS can offer advanced services with decreased session setup delay.

Additionally, this thesis presents an architecture for integrating an Classification Application Server (CAS) for learning geographic regions in IMS. CAS located in the IMS uses the information in the IMS database (e.g. position information) to figure out geographic regions directly in the IMS Network. CAS is shared between users in the local network (cellular network) and global IMS Network. This architecture realizes centralized network controlling applications with fast data transmission. Furthermore, it is shown that compared to the Traditional Network, IMS Network can offer better data transmission services for CAS and other real-time applications. Client-to-server (C-S) and server-to-client (S-C) communications (session based communication) have shorter delay for packet transmission than client-to-client (C-C) communication (message based communication). The designed architecture using C-S and S-C communication in the IMS Network can be applied to realize driver support and active safety services.

1.5. Structure of the Thesis

Chapter 2 explains the basics of the IP Multimedia Subsystem (IMS), Location Based Services (LBS) as well as positioning methods, including the introduction to the architecture of IMS, the fundamentals of LBS, the GPS and DGPS positioning methods as well as the positioning algorithms.

Chapter 3 describes the novel implementation of GPS/DGPS as well as of recursive DGPS methods. A QRD using CORDIC-based approximate rotations solves the required LS problems. The trade-off between accuracy of the positioning results and accuracy of the approximations is investigated using Rinex files and raw
GPS data. Some results of this chapter are presented in [HMB10].

Chapter 4 shows the classification methods for learning geographic regions using LBS in IMS. This chapter presents the integration of vehicle-to-network communication into the architecture of IMS and describes filter functions and classification method. The design and the implementation of the concept is shown and experimental results are discussed. Some results of this chapter are partly presented in [HB09b].

Chapter 5 presents the implementation of an application server on the IMS platform and studies the delays of the packet transmission in the IMS Network. First, we compare the time delay measurement of VoIP registration and call session setup in the Traditional Network and in the IMS Network including the description of the implementation environment and the discussion of the measurement results. Then, the integration of CAS in IMS is shown and how the communication in IMS is established after session setup is presented. We analyze the packet flow for C-C, C-S and S-C to compare the time delay of the data-exchange in the Traditional Network and in the IMS Network. Some results of this chapter are presented in [HB10] and [HVB10].

Chapter 6 draws conclusion and presents possible future work based on the results of this thesis.
1. Introduction
2. Technical Background

The technical background related to this thesis is summarized in this chapter.

2.1. GPS and DGPS Positioning Methods

GPS and DGPS are widely used positioning methods based on satellite communication [MYP12].

2.1.1. GPS Positioning

The GPS positioning [Goe07] [Can07] procedure includes three tasks: acquisition, tracking and positioning [Bor06] [LL99]. The acquisition searches for satellites and gives rough estimates of signal parameters, e.g. pseudoranges and Doppler shifts. Then, these estimated results are refined and tracked, as the signal properties change over time. After tracking, the navigation data can be extracted and pseudoranges can be computed. The final task of the receiver is to compute the user position.

The GPS satellites’ arrangement ensures that every point on our planet is in contact with at least six satellites at all times. Fig. 2.1 gives a schematic view of GPS. Each satellite $k$ continuously broadcasts a digital radio signal that includes its position $(X^k, Y^k, Z^k)$ and its time $t^k$. Onboard atomic clocks ensure an accurate time to a billionth of a second. The radio signal of a satellite propagates with $c = 3 \times 10^8 \text{ m/s}$ in free space. GPS receivers measure the time delay $\tau^k$ of the signal from each satellite $k$ to the receiver, so $\tau^k = t - t^k$, where $t$ is the time of the receiver. The measurement of $t$ in the receiver is not very accurate (as compared to the satellite time $t^k$). Furthermore the speed of the radio signal from the satellites is smaller than $c$ because of ionosphere and troposphere. Therefore, the measured distance from the satellite to the receiver $P^k = \tau^k \cdot c$ is an estimate of the geometrical range $\rho^k$ and is called ‘Pseudorange’ [Bor06] [Tor84]. In Fig. 2.1 the geometrical ranges $\rho^k$ (dashed lines) and the measured pseudoranges $P^k$ (solid
lines) are shown for the satellites \((k = 1, 2, \cdots, 6)\). The receiver simultaneously collects these measurements from at least four satellites and processes them to solve for position and time measurement error.

### 2.1.1.1. Observation Equation

The geometrical ranges \(\rho^k\) and the pseudoranges \(P^k\) define balls with satellite \(k\) as the center of the balls. Finding the intersection of the balls determines the position of the receiver (see Fig. 2.2 for a two-dimensional example). The basic observation equation for the pseudorange \(P^k\) is

\[
P^k = \rho^k + c(\Delta t - \Delta t^k) + T^k + \ell^k + e^k.
\]  

(2.1)

\(\rho^k\) is the geometrical range between satellite \(k\) and receiver \(r\), which can be computed as:

\[
\rho^k = \sqrt{(X^k - X)^2 + (Y^k - Y)^2 + (Z^k - Z)^2}
\]  

(2.2)

where:

---

**Figure 2.1.** Schematic view of GPS with observed pseudoranges \(P^k\) (solid lines) and geometrical ranges \(\rho^k\) (dashed lines)
2.1. GPS and DGPS Positioning Methods

- $\mathbf{x}_r = (X, Y, Z)$ is the position of receiver $r$ [Dan03].

- $\Delta t$ denotes the receiver clock offset.

- $\Delta t_k$ is the satellite clock offset.

- From the ephemerides [MB94] [Bor06], which also include information on the satellite clock offset $\Delta t_k$, the position of the satellite $\mathbf{x}_s^k = (X^k, Y^k, Z^k)$ can be computed.

- $T^k$ is the tropospheric error and $\ell^k$ is the ionospheric error. These two errors are computed from the a-priori models, whose coefficients are part of the broadcast ephemerides.

- $e^k$ is the observation error.

Therefore (2.1) contains four unknowns: $X, Y, Z$ and $\Delta t$. The error terms [Wor10] [SHS05] are minimized by solving a non-linear LS problem, using the non-linear equation (2.1) of $m$ satellites [Pul10].

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{figure2_2.png}
\caption{Definition of observed pseudorange $P^k$ (solid lines) and geometrical range $\rho^k$ (dashed lines)}
\end{figure}
Starting from an initial position for the receiver \((X_1, Y_1, Z_1)\), the position estimate is improved iteratively. Let \(itr\) be the number of iterations \((itr = 1, 2, \cdots, itr_{max})\). The increments \(\Delta X_{itr}, \Delta Y_{itr}, \Delta Z_{itr}\) update the receiver coordinates as follows:

\[
\begin{align*}
X_{itr+1} &= X_{itr} + \Delta X_{itr}, \\
Y_{itr+1} &= Y_{itr} + \Delta Y_{itr}, \\
Z_{itr+1} &= Z_{itr} + \Delta Z_{itr}.
\end{align*}
\] (2.3)

The center of the Earth \((0, 0, 0)\) can be chosen as the initial point, if no a-priori-information is available.

For the following derivations, we assume three satellites determining three balls with radii given by the pseudoranges. The intersection of two of the balls is a circle and then the intersection of this circle with the third ball yields two intersection points. Only one of the two points is on the surface of the earth (which is a fourth ball). The radii of the three balls are around 22000 km (pseudoranges) while for the earth it is about 6400 km. Therefore, the initialization \((0, 0, 0)\) is actually close to the minimum we are looking for. We obtain one unambiguous solution for the actually non-linear problem.

The Taylor series expansion of \(f(X_{itr} + \Delta X_{itr}, Y_{itr} + \Delta Y_{itr}, Z_{itr} + \Delta Z_{itr})\) is

\[
f(X_{itr+1}, Y_{itr+1}, Z_{itr+1}) = f(X_{itr}, Y_{itr}, Z_{itr}) + \frac{\partial f(X_{itr}, Y_{itr}, Z_{itr})}{\partial X_{itr}} \Delta X_{itr} + \frac{\partial f(X_{itr}, Y_{itr}, Z_{itr})}{\partial Y_{itr}} \Delta Y_{itr} + \frac{\partial f(X_{itr}, Y_{itr}, Z_{itr})}{\partial Z_{itr}} \Delta Z_{itr}.
\] (2.4)

Here, it includes only first-order terms. The partial derivatives of \((2.4)\) are

\[
\begin{align*}
\frac{\partial f(X_{itr}, Y_{itr}, Z_{itr})}{\partial X_{itr}} &= -\frac{X^k - X_{itr}}{\rho^k_{itr}}, \\
\frac{\partial f(X_{itr}, Y_{itr}, Z_{itr})}{\partial Y_{itr}} &= -\frac{Y^k - Y_{itr}}{\rho^k_{itr}}, \\
\frac{\partial f(X_{itr}, Y_{itr}, Z_{itr})}{\partial Z_{itr}} &= -\frac{Z^k - Z_{itr}}{\rho^k_{itr}}.
\end{align*}
\]

Let \(\rho^k_{itr} = \sqrt{(X^k - X_{itr})^2 + (Y^k - Y_{itr})^2 + (Z^k - Z_{itr})^2}\) be the range computed from the satellite position \((X^k, Y^k, Z^k)\) to the approximate receiver position
2.1. GPS and DGPS Positioning Methods

(X_{itr}, Y_{itr}, Z_{itr}), and applying the Taylor expansion to the estimated current position (X_{itr}, Y_{itr}, Z_{itr}) in (2.2), the first-order linearized observation equation is obtained \[Bor06\] \[HMB10\]:

\[
P^k_{itr} = \rho^k_{itr} - \frac{X^k - X_{itr}}{\rho^k_{itr}} \Delta X_{itr} - \frac{Y^k - Y_{itr}}{\rho^k_{itr}} \Delta Y_{itr}
- \frac{Z^k - Z_{itr}}{\rho^k_{itr}} \Delta Z_{itr} + c(\Delta t_{itr} - \Delta t^k) + T^k_{itr} + \ell^k_{itr} + e^k_{itr},
\]

(2.5)

where \(\rho^k_{itr}\) and \(\Delta t_{itr}\) are the estimated geometric range and the estimated clock error for iteration \(itr\) at the receiver, respectively. \(T^k_{itr}\) and \(\ell^k_{itr}\) are the tropospheric and ionospheric errors estimated by the a priori models in each iteration \(itr\).

2.1.1.2. Applying Least-Squares Method for Position Estimation

Equation (2.5) can be rewritten as

\[
\begin{bmatrix}
- \frac{X^k - X_{itr}}{\rho^k_{itr}} & - \frac{Y^k - Y_{itr}}{\rho^k_{itr}} & - \frac{Z^k - Z_{itr}}{\rho^k_{itr}} & 1
\end{bmatrix}
\begin{bmatrix}
\Delta X_{itr} \\
\Delta Y_{itr} \\
\Delta Z_{itr} \\
c\Delta t_{itr}
\end{bmatrix}
= P^k - \rho^k_{itr} + c\Delta t^k - T^k_{itr} - \ell^k_{itr} - e^k.
\]

(2.6)

\(m \geq 4\) satellites are required to form a system of linear equations (usually \(m \geq 6\) satellites are available). Let \(b_{itr} = [b^1_{itr}, b^2_{itr}, \ldots, b^m_{itr}]^T\) and \(||\cdot||_2\) be the 2-norm, then the LS problem is stated as:

\[
\min_{\Delta x_{itr}} ||A_{itr}\Delta x_{itr} - b_{itr}||_2^2,
\]

(2.7)

where

\[
A_{itr} =
\begin{bmatrix}
- \frac{X^1 - X_{itr}}{\rho^1_{itr}} & - \frac{Y^1 - Y_{itr}}{\rho^1_{itr}} & - \frac{Z^1 - Z_{itr}}{\rho^1_{itr}} & 1 \\
- \frac{X^2 - X_{itr}}{\rho^2_{itr}} & - \frac{Y^2 - Y_{itr}}{\rho^2_{itr}} & - \frac{Z^2 - Z_{itr}}{\rho^2_{itr}} & 1 \\
- \frac{X^3 - X_{itr}}{\rho^3_{itr}} & - \frac{Y^3 - Y_{itr}}{\rho^3_{itr}} & - \frac{Z^3 - Z_{itr}}{\rho^3_{itr}} & 1 \\
\vdots & \vdots & \vdots & \vdots \\
- \frac{X^m - X_{itr}}{\rho^m_{itr}} & - \frac{Y^m - Y_{itr}}{\rho^m_{itr}} & - \frac{Z^m - Z_{itr}}{\rho^m_{itr}} & 1
\end{bmatrix},
\]

and

\[
b_{itr} =
\begin{bmatrix}
P^1 - \rho^1_{itr} + c\Delta t^1 - T^1_{itr} - \ell^1_{itr} - e^1 \\
P^2 - \rho^2_{itr} + c\Delta t^2 - T^2_{itr} - \ell^2_{itr} - e^2 \\
P^3 - \rho^3_{itr} + c\Delta t^3 - T^3_{itr} - \ell^3_{itr} - e^3 \\
\vdots \\
P^m - \rho^m_{itr} + c\Delta t^m - T^m_{itr} - \ell^m_{itr} - e^m
\end{bmatrix}.
\]
2. Technical Background

2.1.2. Differential GPS Positioning

The DGPS method increases the accuracy of GPS by simultaneously taking GPS observation at two locations (see Fig. 2.3), which have identical geometric dilution of precision [EKR88] [HTJ90]. Then, the correction of pseudoranges at the unknown receiver position is done by using the measured-pseudorange-errors at the known position of the reference station. Mobile radio base stations with known coordinates can be used as reference stations. Usually DGPS is applied for obtaining very accurate position estimates. However, if GPS position accuracy is sufficient, approximate DGPS solutions can also be used to simplify the implementation of the receiver (hardware or software) and reduce the computational complexity as detailed below.

The algorithms used for GPS positioning can also be applied for DGPS [HMB10].

Figure 2.3.: Differential GPS positioning

and $x_{itr} = [\Delta X_{itr} \; \Delta Y_{itr} \; \Delta Z_{itr} \; c\Delta t_{itr}]^T$. The solution $\Delta X_{itr}$, $\Delta Y_{itr}$, $\Delta Z_{itr}$ has to be added to the previous approximate receiver position to get the next approximate position (see (2.3)). These iterations continue until the solution $\Delta X_{itr}$, $\Delta Y_{itr}$, $\Delta Z_{itr}$ reaches the required precision.

The basic linear algebra concerning the solution of LS problems is described in Appendix A. In the following, the QRD by Givens rotations is used.
However, it is also possible to obtain a linear LS algorithm for DGPS. Compared to [CP03], a slightly different derivation of the LS problem for solving DGPS is presented. The geometric constellation for DGPS is given as follows (without restriction of generality, the 2D-constellation is illustrated in Fig. 2.4).

- $(0,0,0)$ is the origin of the coordinate system (e.g. the center of the Earth)
- $\mathbf{x}_f$ is the vector from the origin of the coordinate system to receiver $f$ (reference station with known position)
- $\mathbf{x}_r$ is the vector from the origin of the coordinate system to receiver $r$ (with unknown position)
- $\mathbf{x}_k^s$ is the vector from the origin of the coordinate system to satellite $k$

![Figure 2.4: Geometry for DGPS](image)
- $\mathbf{x}_{sf}^k$ is the vector from receiver $f$ to satellite $k$
- $\mathbf{x}_{sr}^k$ is the vector from receiver $r$ to satellite $k$
- $\mathbf{x} = \mathbf{x}_r - \mathbf{x}_f$ is the vector from receiver $f$ to receiver $r$, so if $\mathbf{x}_f$ and $\mathbf{x}$ are known, $\mathbf{x}_r$ can be obtained by $\mathbf{x}_r = \mathbf{x}_f + \mathbf{x}$. $\mathbf{x}$ can also be obtained by $\mathbf{x} = \mathbf{x}_{sf}^k - \mathbf{x}_{sr}^k$.
- $\mathbf{a}^k$ is the unit vector from the center of $\mathbf{x}$ to satellite $k$.

\[
\mathbf{a}^k = \frac{\mathbf{x}_{sf}^k + \mathbf{x}_{sr}^k}{\|\mathbf{x}_{sf}^k + \mathbf{x}_{sr}^k\|_2} \quad \text{and} \quad \|\mathbf{a}^k\|_2 = 1. \quad (2.8)
\]

From Fig. 2.4, the following equations can be obtained:

\[
\cos(\varphi_1) = \frac{(\mathbf{a}^k)^T \cdot \mathbf{x}}{\|\mathbf{a}^k\|_2 \|\mathbf{x}\|_2} = \frac{(\mathbf{a}^k)^T \cdot \mathbf{x}}{\|\mathbf{x}_{sf}^k - \mathbf{x}_{sr}^k\|_2}, \quad (2.9)
\]

\[
\cos(\varphi_2) = \frac{(\mathbf{x}_{sf}^k + \mathbf{x}_{sr}^k)^T \cdot \mathbf{x}}{\|\mathbf{x}_{sf}^k + \mathbf{x}_{sr}^k\|_2 \|\mathbf{x}\|_2} = \frac{(\mathbf{x}_{sf}^k + \mathbf{x}_{sr}^k)^T (\mathbf{x}_{sf}^k - \mathbf{x}_{sr}^k)}{\|\mathbf{x}_{sf}^k + \mathbf{x}_{sr}^k\|_2 \|\mathbf{x}_{sf}^k - \mathbf{x}_{sr}^k\|_2}. \quad (2.10)
\]

Because $\cos(\varphi_1) = \cos(\varphi_2)$, (2.9) is equal to (2.10), i.e.:

\[
\frac{(\mathbf{a}^k)^T \cdot \mathbf{x}}{\|\mathbf{x}_{sf}^k - \mathbf{x}_{sr}^k\|_2} = \frac{(\mathbf{x}_{sf}^k + \mathbf{x}_{sr}^k)^T (\mathbf{x}_{sf}^k - \mathbf{x}_{sr}^k)}{\|\mathbf{x}_{sf}^k + \mathbf{x}_{sr}^k\|_2 \|\mathbf{x}_{sf}^k - \mathbf{x}_{sr}^k\|_2}. \quad (2.11)
\]

Therefore,

\[
(\mathbf{a}^k)^T \cdot \mathbf{x} = \frac{(\mathbf{x}_{sf}^k + \mathbf{x}_{sr}^k)^T (\mathbf{x}_{sf}^k - \mathbf{x}_{sr}^k)}{\|\mathbf{x}_{sf}^k + \mathbf{x}_{sr}^k\|_2^2} = \frac{(\|\mathbf{x}_{sf}^k\|_2 + \|\mathbf{x}_{sr}^k\|_2) (\|\mathbf{x}_{sf}^k\|_2 - \|\mathbf{x}_{sr}^k\|_2)}{\|\mathbf{x}_{sf}^k + \mathbf{x}_{sr}^k\|_2^2}. \quad (2.12)
\]

With

\[
g = \frac{\|\mathbf{x}_{sf}^k\|_2 + \|\mathbf{x}_{sr}^k\|_2}{\|\mathbf{x}_{sf}^k + \mathbf{x}_{sr}^k\|_2^2} \geq 1 \quad (2.13)
\]
(2.12) can be written as:

\[
(a^k)^T \cdot x = g \cdot (\|x^k_{sf}\|_2 - \|x^k_{sr}\|_2)
\] (2.14)

g can also be estimated as (see the additional gray part of Fig. 2.4):

\[
g = \frac{\|x^k_{sf} + x^k_{sr}\|_2}{\|x^k_{sf} + x^k_{sr}\|_2}
\]

\[
\leq \frac{\|x^k_{sf} + x^k_{sr}\|_2 + \|x^k_{sf} + x^k_{sr}\|_2 - \|x^k_{sf} + x^k_{sr}\|_2}{\|x^k_{sf} + x^k_{sr}\|_2}
\]

\[
\leq \frac{\|x^k_{sf} + x^k_{sr}\|_2 + \|x^k_{sf} + x^k_{sr}\|_2}{\|x^k_{sf} + x^k_{sr}\|_2} = 1 + \|x^k_{sf} + x^k_{sr}\|_2.
\] (2.15)

With (2.13) and (2.15), we obtain

\[
1 \leq g \leq 1 + \|x\|_2 \|x^k_{sf} + x^k_{sr}\|_2.
\] (2.16)

Since the altitude of a GPS satellite is about 20200 km above the sea level, and \(\|x^k_{sf} + x^k_{sr}\|_2\) is about 2 times the altitude of a satellite, it holds that \(\|x^k_{sf} + x^k_{sr}\|_2 \approx 40400\) km. Centimeter-level DGPS systems are restricted to distances from the reference station to the receiver in the order of 10 km [Wan03]. These distances can also be assumed for base stations, which are used as reference receivers. Since \(\|x^k_{sf} + x^k_{sr}\|_2 \approx 10^{40400}\) = 0.0002, with (2.16) one obtains \(g \leq 1 + \|x\|_2 \|x^k_{sf} + x^k_{sr}\|_2 \approx 1.0002\). Therefore, with \(\rho^k_{sf} = \|x^k_{sf}\|_2\) and \(\rho^k_{sr} = \|x^k_{sr}\|_2\) (see (2.2)), one obtains for (2.14):

\[
(a^k)^T \cdot x \approx \|x^k_{sf}\|_2 - \|x^k_{sr}\|_2 = \rho^k_{sf} - \rho^k_{sr}.
\] (2.17)

As explained in Sec. 2.1.1.1, in practice the exact geometrical ranges \(\rho^k_{sf}\) and \(\rho^k_{sr}\) are not known. Only the pseudoranges \(P^k_{sf}\) and \(P^k_{sr}\) (see (2.1)) are available. Comprising all the error terms of (2.1) together in \(v^k\), (2.17) can be rewritten as

\[
(a^k)^T \cdot x = P^k_{sf} - P^k_{sr} + v^k.
\] (2.18)

For \(m\) satellites, we get

\[
(a^1)^T \cdot x = P^1_{sf} - P^1_{sr} + v^1
\]

\[
(a^2)^T \cdot x = P^2_{sf} - P^2_{sr} + v^2
\]

\[
\vdots
\]

\[
(a^m)^T \cdot x = P^m_{sf} - P^m_{sr} + v^m.
\] (2.19)
With

\[
A = \begin{bmatrix}
(a_1)^T \\
(a_2)^T \\
\vdots \\
(a_m)^T
\end{bmatrix}, \quad
v = \begin{bmatrix}
v_1 \\
v_2 \\
\vdots \\
v_m
\end{bmatrix},
\]

(2.20)

and

\[
b = \begin{bmatrix}
P_{sf}^1 - P_{sr}^1 \\
P_{sf}^2 - P_{sr}^2 \\
\vdots \\
P_{sf}^m - P_{sr}^m
\end{bmatrix},
\]

(2.21)

(2.19) can be written as:

\[
A \cdot x = b + v.
\]

(2.22)

Solving this system of linear equation for \( x \), i.e.

\[
\min_x \|Ax - b\|_2^2
\]

(2.23)

yields \( x \). Using this solution yields the position of the receiver \( x_r = x_f + x \). Obviously, DGPS yields the position by solving the linear LS problem in (2.23). Compared to the GPS solution, the DGPS solution is obtained by solving one linear LS problem, i.e. no iterations are required.

### 2.1.3. Recursive DGPS

Since \( a^k \) depends on the baseline \( x \) (see Sec. 2.1.2), the system matrix \( A = A(x) \) depends on \( x \). Let \( \min_{x_{t-1}} \|A_{t-1}x_{t-1} - b_{t-1}\|_2^2 \) and \( \min_{x_t} \|A_tx_t - b_t\|_2^2 \) be two subsequent LS problems for DGPS positioning at two subsequent time steps \( t - 1 \) and \( t \). As two subsequent solutions \( x_{t-1} = [x_{t-1} y_{t-1} z_{t-1}] \) and \( x_t = [x_t y_t z_t] \) are spatially close to each other, the system matrices \( A_{t-1} \) and \( A_t \) show small difference. Therefore, once the result for \( A_{t-1} \) is computed, the computations can be reused for solving the problem for \( A_t \). This is particularly advantageous for our presented methods as shown in Sec. 3.1.7.
2.2. Location Based Services (LBS)

Location Based Services (LBS) [SV04] denote applications which integrate geographic location with the general notion of services. LBS are wireless 'mobile content' services which are used to provide location-specific information to mobile users moving from location to location. Examples for applications of LBS are car navigation systems, emergency services, person tracking or tourist tour planning.

LBS have a long tradition. The global positioning system (GPS) (satellite infrastructure) has been used to provide the positioning of people and objects since 1976s. GPS positioning was initially used for military purposes, but since 1980s this system’s positioning data has been made freely available. Since then many industries have tried to enhance their products and services by taking up the opportunity to access position data through GPS. The LBS market is growing quickly worldwide with the continued rise of GPS-equipped cell phones and in-car GPS and navigation devices. Location based technology is being used to create a wide variety of new location-aware applications [Kue05]. Combined with notification services in LBS, these applications can automatically alert users who are close to a preselected destination. For example, based on the phone’s fixed address, traditional emergency 911 services can automatically deliver a caller’s position to the appropriate public safety entity. Using GPS and other technologies, advanced emergency 911 can detect the caller’s mobile phone position.

In traditional positioning systems, location information is obtained by GPS positioning with a device and the help of a satellite system. Currently, mobile network operators also focus other types of localization technology and new market opportunities. Besides GPS technique [DR01], network positioning methods [DMS98] as well as the positioning estimation by triangulation [HHGC08] can be utilized for location estimation for LBS [Goz06] [MLVEC09]. Many of the data services are location enhanced and they provide enhanced LBS solutions for people to be able to find one another and stay in touch with their friends and family, or to get directions to Points of Interest (POI), e.g., shops or restaurants. Therefore, operators need to invest in new technologies to develop the data related business, especially in mobile messaging.

Most of the applications of LBS are a part of everyday life. They run on computers, personal digital assistants or mobile devices. Because of various possible applications, the basic requirements of LBS are numerous. It is a complex task to provide users with value-added location information. As described in [SV04], the
major challenges to consider are the personalization of services, the ubiquity of services to the mobile user, and the combination of services with the transmission of context, such as time, location and possibly other dimensions like the user profile. The user profile includes not only the user-related data (name and address), but also preferences set by the user or inferred by the system. A description of promising approaches for information delivery and exchange among many users who are geographically grouped can be found in chapter 4.

LBS applications use information from several content databases, which are Digital Road Databases (the road network with digital road maps), Point of Interest information (business and landmark information) and Dynamic Data (such as traffic and weather reports). Dynamic Data gets more and more attentions by LBS applications. Those who drive to work daily are aware that the optimal route from home to office may change during the rush hours. Bad weather, road construction, accidents or traffic jams can all affect a road’s traffic situation [Kue05]. Daily traffic conditions cannot be coded into a map database a priori. Location information is required to work with dynamic data to change road features and to supplement or override existing map information. Emergency 911 and fleet management depend on LBS engines with dynamic data capabilities and are time-critical applications, because they have to react almost instantaneously to changing conditions.

LBS provide and deliver information to the user in a highly selective manner depending on users’ past, present, or future location and other context information. However, LBS are often even more generally defined as an value-added service offered in a wireless environment that exploits mobile terminal location position information of user groups. Group finder system (see chapter 4) is one of these LBS applications, where correlations between a number of moving users must be established. Another one is tracking applications where a number of moving objects must be tracked simultaneously and the state of individuals or groups of objects must be observed.

2.2.1. Classifying LBS

Location services and LBS are different. Location services provide geographic location information about mobile terminals, e.g. cell phones, Personal Digital Assistants (PDAs) or with sensors tagged on moving objects. LBS refer to the information services using this location information about a mobile terminal to offer customized information content to the mobile user or to third parties, i.e. other mobile terminals or static users.
LBS can be classified into two categories: person-oriented LBS and device-oriented LBS \cite{SV04}.

- Person-oriented LBS consist of all of those applications where a service is user-based. The focus of the application is to position a person or using the position of a person to enhance a service. The service is controlled by the person located.

- Device-oriented LBS applications are applications without the control from the user. They might be the position of a person, but they do not need to. Instead of a person, an object (e.g. a car) or a group (e.g. people/cars in a certain region) could also be located. In device-oriented applications, the person or object located can usually not control the service (e.g. car tracking for theft recovery).

In addition to this first classification, the services are also being distinguished by push and pull services:

- Push services imply that the user receives information without having to request it. The information may be sent to the user with prior consent or without prior consent. In a push-based application service, based on the occurrence of an event or the changing of a condition, the information is pushed to mobile terminals automatically.

- Pull services mean that a user actively utilizes an application to ‘pull’ information from the network. This information can be location-based. In a pull-based application service, requests are initiated by the mobile terminal, i.e. the user.

### 2.2.2. Service Categories and Characteristics

There are many service categories of LBS, and each service category has its own characteristic \cite{Kue05}. The mostly used services are Infotainment Services, Tracking Services, Selective Information Dissemination Services, Location-based Games and Emergency Support Services. These services depend totally on the available positioning technology.
2.2.3. Updating Database

A well-functioning value chain is important for the delivery of LBS in practice. The value chain begins with content providers. They provide different kinds of contents, which can be geo-referenced to a content integrator. Traffic condition data with information about accidents and congestion, weather data and the information about hotel rooms as well as the information about the current locations of a group of service users are the content. Special techniques are required to get the content from tracking continuous processes, e.g. positions of moving mobile users, since their present, the past and the future positions of a large group of mobile users. Therefore, the content must be continuously updated. For example, low-accuracy tracking is sufficient for up-to-date weather information. In contrast, higher accuracy is needed, if we want to use navigation. Different representations of a user’s movement result in different rates of update [SV04].

2.2.4. Implementation for LBS

LBS consists of a positioning layer and an application layer. The positioning layer calculates the position of a mobile device or user. Position Determination Equipment (PDE) (Cell-ID, E-OTD or GPS) and geo-spatial data in a geographic information system are used. The location information would be managed and sent directly to an application by the positioning layer. The application layer is often and confusingly referred to as a ‘client’ in the LBS industry. It comprises all of those services that request location data to be integrated into provided services. Many network operators have put a middleware layer between positioning and application layer due to the increasing number of LBS applications. Since PDE is very deep in the network of a mobile operator, it results in a complex and lengthy hookup of each individual new data service. Because a middleware layer is connected to the network and controls all location services added in the future, it can significantly reduce the complexity of service integration. Additionally, a middleware layer allows users to manage location access rights of third-party applications on the downstream (pull services), and it systematically makes location information anonymous on the upstream (push services). Thus, the middleware layer takes over a similar role as an anonymizing proxy does on the Internet [Kue05].

In LBS, there is a spatial database server which retrieves location information and manages the geo-spatial data. Spatial databases play central role in modern location-based applications [Kue05]. LBS need a cohesiveness between disparate
components and conformance with open standards to achieve their full potential. Therefore, it is necessary to develop a standard open platforms for LBS.

IMS is supposed to be taken as an open platform for integrating the applications of LBS, see section 2.3. Therefore, the infrastructure required for the deployment of the ideas presented in this thesis to practical application will be available in next generation networks.

2.3. IP Multimedia Subsystem (IMS)

Fixed and mobile networks have been growing quickly in the last 20 years. In the fixed network, voice and video communication are most important for the traditional Public Switched Telephone Network (PSTN) [CML07] and the Integrated Services Digital Network (ISDN). Faster and cheaper Internet connection such as Asymmetric Digital Subscriber Line (ADSL) is becoming more and more important to users. These types of Internet connections enable always-on connectivity, which is important for real-time communication like online gaming, Voice over IP (VoIP) or chatting applications.

The third generation (3G) of mobile communication also enables faster data rates and various multimedia services for mobile devices. Fixed and mobile networks should work together. The mobile devices have large, high-precision displays and cameras. Furthermore, in these devices a lot of resources for applications are built. The applications are peer-to-peer entities and the information exchange is done not only with the user interface, but also by sharing, e.g. shared browsing, shared whiteboard, shared game experience, shared two-way radio session (Push to talk Over Cellular). To establish a peer-to-peer connection over Internet Protocol (IP) is the main task of mobile network communication.

IMS [PNKM06][VWF07][Fra08] is an architectural framework for delivering Internet Protocol (IP) multimedia services. There are various standards for transmission in mobile radio communication, e.g. GPRS, UMTS, LTE, WLAN. Each standard demands its own protocol and service. If various standards are needed at the same time, a lot of customized protocol stacks are required and architecture cost is increased. IMS based on IP-based Services can solve this problem and enables different standards to communicate to each other [AB07].

IMS makes IP-based services between the terminals possible. In order to ease the integration with the Internet, Internet Engineering Task Force (IETF) protocols are used by IMS [SWSC07]. According to the 3GPP, IMS provides the access
of multimedia and voice applications from wireless and wireline terminals.

2.3.1. IMS Architectural Requirements

Some basic requirements are needed to create the IMS architecture [PNKM06].

2.3.1.1. IP Multimedia Sessions

Existing communication networks, which use circuit switched bearers, offer voice, video and messaging type of services. The IMS improves communication by offering enriched communication means. During a single communication session, IMS users can mix and match different IP-based services in any way they want. Users can combine voice, video, content sharing and presence as part of their communication. They can also add or drop services when they want [RLCS07]. For example, a voice session can be started as a session and later a game component can be added to the same session.

2.3.1.2. Ensuring Quality of Service for IMS

On the public Internet, since packets arrive out of order and some packets are lost or discarded, delays are high and variable. IMS should avoid this. IMS together with the underlying access and transport networks provide end-to-end Quality of Service (QoS). The QoS of the user equipment (UE) is determined during a Session Initiation Protocol (SIP) for session setup or session modification procedure [SKMDM07]. The UE can require the following parameters: a) Media type, direction of traffic; b) Media type bit rate, packet size, packet transport frequency; c) Usage of Real-time Transport Protocol (RTP) payload for media types; d) Bandwidth adaptation. UEs reserve suitable resources from the access network, after the parameters are negotiated at the application level.

2.3.1.3. Secure Communication

In every telecommunication system, security is a basic requirement. Specified authentication and authorization mechanisms are desired by the IMS. Moreover, between the UE and the IMS Network, the integrity and optional confidentiality of the SIP messages is provided. The IMS provides at least a similar level of security as the corresponding GPRS and circuit-switched networks: for example, the IMS
ensures that before users start using services, they are authenticated and users can ask for privacy during a session.

2.3.1.4. IMS Layer Design and Stack Basics

A number of main elements compose the architecture of an IMS:

- **UE** is the endpoint. It is a part of the IMS architecture and stay with the user.

- The **Access network** is a large part of the IMS architecture. The overall network can be communicated by access network.

- The **Core network** is a major element within the IMS architecture. All the core functionality is provided by the core network.

- The **Application layer** contains the web portal and the application servers providing the end user with service and enhanced service controls.

![Figure 2.5.: OSI and IMS layered Architecture](Image)
Concerning the communication between systems the layered Open Systems Interconnection (OSI) model (see Fig. 2.5) is widely used [IMS10]. OSI is used in communications systems and the communication channel is divided by OSI into various levels of tasks. Each layer has its own services which is different from the services in the other layers. However, some services overlap the layers above layer 5 (sometimes above layer 4). IMS has its own definitions for these layers. Fig. 2.6 shows the IMS layered architecture. There are three main IMS layers, which correspond to OSI-model layers 4 and above. These are Transport Layer which equates to OSI layer 4, Session Control/IMS Layer which equates to OSI layer 5, and Application/Service Layer which equates to OSI layers 6 and 7 [IMS10].

**Figure 2.6:** IMS Network Architecture Overview
2.3.1.5. **Transport Layer**

SIP signalling initiation and termination as well as session setup are done by this layer. It provides bearer services which include the conversion from analogue or digital formats to packets. All media processing facilities are contained in this layer, including media gateways. Many media-related services such as conferencing, playing announcements and speech recognition can be provided by these facilities.

2.3.1.6. **Session Control Layer/IMS Layer**

The Call Session Control Function (CSCF) is in this layer. It provides session control for terminals and applications using the IMS Network. By communicating with the transport and endpoint layer, the CSCF guarantees QoS. Home Subscriber Server (HSS) maintaining the user profiles, such as their registration details or preferences, is also located in the Session Control Layer. The Media Gateway Control is another element in this layer.

2.3.1.7. **Application Layer/Service Layer**

The Application Server Layer controls the services required by the end users. Because of the flexible design of the IMS architecture and SIP signalling, different telephony and non-telephony servers can be supported at the same time, e.g. a Telephony Application Server (TAS), IP Multimedia - Service Switching Function (IM-SSF), Non-Telephony Application Server, Open Service Access - Gateway, etc [IMS10].

2.3.2. **IMS Core Network**

IMS core network is the most important part of the network.

2.3.2.1. **Call Session Control Function**

The Call Session Control Function (CSCF) is the heart of the IMS Core Network in the IMS layer. It is used to process SIP signalling [Eri07] [CKK10]. It provides session control for terminals and applications using the IMS Network and is responsible for interacting with Home Subscriber Server (HSS) [Eri07].
Proxy Call Session Control Function (P-CSCF), Interrogating Call Session Control Function (I-CSCF) and Serving Call Session Control Function (S-CSCF) are three different kinds of Call Session Control Functions. Each SCSF has its own special tasks. During registration and session establishment, all these three functions play a role and form the SIP routing machinery.

- **P-CSCF** is a SIP proxy which is the entry point to IMS from any access network. All SIP signalling traffic from the UE is sent to the P-CSCF. The P-CSCF sends all terminating SIP signalling from the network to the UE. P-CSCF supports compression of the SIP signalling and provides secure transmission of the SIP signalling. There are four tasks assigned for the P-CSCF: SIP compression, IPSec security association, interaction with Policy Decision Function (PDF) and emergency session detection [PNKM06].

- **I-CSCF** is a contact point within an operator’s network for all connections to a subscriber of that network operator. I-CSCF is the first point of contact in the home network. Together with HSS, it can find the S-CSCF where the user is registered or selecting a new S-CSCF for an unregistered user. There are four tasks assigned for I-CSCF: Obtaining the name of the next hop (either S-CSCF or application server) from the HSS, assigning an S-CSCF based on received capabilities from the HSS, routing incoming requests further to an assigned S-CSCF or the application server, providing Topology Hiding Inter-network Gateway functionality [PNKM06].

- **S-CSCF** is the central node for the provision of the SIP signalling. It is also responsible for routing and translation, maintenance of sessions, interaction with other services and charging. S-CSCF uses Diameter (AAA protocol, see section [2.3.4]) interfaces to the HSS to assist in user authentication. S-CSCF is the focal point of the IMS, as it is responsible for handling registration processes, making routing decisions and maintaining session states as well as storing the service profiles.

### 2.3.2.2. Home Subscriber Server

Except the control of IMS Core functions, there are other important elements in the IMS core, e.g. the **Home Subscriber Server** (HSS), a database for user-related information such as service profile, name of S-CSCF, user authentication...
and service authorization data. HSS keeps all the user information and is available for the access to user data. The main data stored in the HSS includes user identities, registration information, access parameters and service-triggering information. User-specific requirements for S-CSCF capabilities can be provided by HSS. I-CSCF uses this information to select the most suitable S-CSCF for a user.

### 2.3.3. Access Network

IMS services can be provided over any networks with IP connectivity (e.g., GPRS, WLAN, UMTS, LTE). The elements associated with communication from the core network to external networks and users have made up the IMS access network. Various forms of IP Carrier Access Networks (IP-CAN), which provides the IP connectivity as well as mobility, can access the IMS Network. Control plane signalling and media transfer is sent through the IP-CAN to the IMS core network by the IMS terminal.

### 2.3.4. Other Elements in IMS Network

Various protocols are used in IMS [Eri07] (see Fig. 2.6):

- SIP Application Servers (SIP-AS) execute all the applications and services in IMS, and guarantee the update and upgrade of the application for all users.
- Media Resource Function (MRF) provides media services in the home network. Functionality is implemented by MRF to manage as well as process media streams.
- Breakout Gateway Control Function (BGCF) is a logical entity in the IMS Network. It routes the session to MGCF.
- Media Gateway Control Function (MGCF) controls the media resources used when traffic needs to flow between networks.
- Media Gateway (MGW) provides the interworking of media flows between different networks
- Session Border Controller (SBC) are IP to IP gateways at the boarder between an operator’s IMS Network and other networks. It ensures Security and QoS for real time IP streams [YC08]. Not only the signal traffic but also media traffic can be controlled by SBC.
2.3.5. IMS Network Protocols

- Session Initiation Protocol (SIP) is a main signalling protocol used in IMS Networks for establishing, modifying and terminating multimedia session. It is widely used for VoIP, instant messaging, presence and more.
- Diameter is the Accounting, Authentication, Authorization (AAA) protocol for IMS.
- H.248 is a control protocol used between media resources and media control functions.
3. Scalable Positioning Methods

In this chapter, we present novel implementations for GPS and DGPS positioning methods with scalable positioning accuracies, which provide a broad range of accuracies from average to very high and which significantly reduce the software and hardware implementation costs of GPS/DGPS positioning. Instead of optimizing GPS/DGPS, approximated GPS/DGPS is investigated in this thesis. The key of the work is to enable the trade-off between positioning accuracies and computational efforts. Compared to [MSB06], which compares the positioning accuracy between GPS and low-cost DGPS, the presented method in this thesis allows to compute positioning results with various accuracies by low-end or high-end GPS/DGPS receivers.

As already mentioned in the introduction, important applications of our methods are:

- adapting GPS positioning accuracy to network positioning accuracy. When the required network information is not available, network positioning can be replaced by approximate GPS. This achieves lower computational complexity and lower power consumption than using conventional GPS (note that, in this case the GPS accuracy is not required).

- adapting DGPS positioning accuracy to GPS positioning accuracy. GPS accuracy can be obtained by approximate DGPS. Thus, when a connection to the network is available (e.g. cellular network or Radio Data System (RDS)), GPS accuracy can be achieved by approximate DGPS with reduced software and hardware implementation costs.

As shown in Fig. 3.1, the proposed methods enable a dynamically scalable on-demand positioning accuracy, such that only the required accuracy for the use-case is delivered. The computational complexity is adapted to the required accuracy; hence low/high positioning accuracies should be obtained by low/high computational efforts. By this we can significantly prolong the battery life time of mobile equipments, e.g., for remote child tracking with scalable positioning accuracies.
In order to accomplish the positioning, an initial set of pseudoranges, which are the measured distances from satellites to the GPS receiver, is needed. For GPS non-linear LS [Pull10] is the most common method to determine the receiver’s position from the pseudoranges [Bor06] [Tor84]. As shown in Sec. 2.1, linearization is done to convert the non-linear algorithm into a linear iterative algorithm, which requires the solution of an over-determined system of linear equations in each iteration step $itr$ ($itr = 1, 2, \ldots, itr_{max}$), i.e., a linear LS problem is the core of each iteration step. The linear LS method is also used for DGPS positioning. In this case, the same linear LS method may be used, but no iterations are required ($itr_{max} = 1$), which significantly reduces the computational complexity. In both cases, the QR Decomposition (QRD) can be used to solve the linear LS problems [GL96] [AAM09].

To realize scalable positioning methods, we use approximate versions of the QRD to solve the linear LS problems in all cases. Instead of annihilating the lower diagonal matrix elements during the QRD, CORDIC-based approximate rotations are applied [Goe94] [GPS93], which only diminish these matrix elements. By
choosing the accuracy of the approximation (based on the required positioning accuracy of the use case), i.e. by choosing \( itg \) (\( itg = 1, 2, \ldots, itg_{\text{max}} \)), which is the number of optimal CORDIC angles per rotation, the computational efforts for solving the LS problems can be adjusted. An approximate solution to the LS problem is obtained and the accuracy depends on \( itg \). Note that in all cases, for \( itg_{\text{max}} = w \), where \( w \) is the word length, we obtain the exact solution of the LS problem \([\text{Goe94}]\). The approximate GPS/DGPS method converges to the exact solution with increasing \( itg \).

The presented method is particularly suited for recursive positioning, where the positioning problem must be solved for two spatially close positions (two subsequent time steps). In this case, only small rotation angles are required such that only a very small number of optimal CORDIC angles (small \( itg \)) is sufficient to obtain accurate positioning results.

Figure 3.2 shows the basic structure of our GPS/DGPS positioning methods:

- For GPS, approximate solutions of the linear LS problems are obtained in each iteration and the trade-off between \( itr \) and \( itg \) in terms of positioning accuracy and complexity is compared.

- For DGPS, where \( itr_{\text{max}} = 1 \), approximate solutions for the (single) linear LS problem are obtained and the positioning accuracies for varying \( itg \) are compared.

These comparisons are done using Rinex files from a high-end GPS receiver and raw GPS data from a low-end GPS receiver. The experimental results show that coarse
approximations are sufficient for obtaining required position estimates. Thus, the
presented methods reduce the computational complexity and the required power
consumption significantly. They are also highly suited for hardware implementa-
tion [DM93] [HD93].

3.1. Approximate Solution of LS Problems

3.1.1. Givens Rotation

Given a vector \( z = [a \ b]^T \), a Givens rotation \( G(\phi) \) [GL96] annihilates the second
component of \( z \):

\[
\begin{bmatrix}
a' \\
0
\end{bmatrix} = G(\phi) \begin{bmatrix}
a \\
b
\end{bmatrix}, \text{ where } G(\phi) = \begin{bmatrix}
\cos(\phi) & \sin(\phi) \\
-\sin(\phi) & \cos(\phi)
\end{bmatrix}
\] (3.1)

with \( \phi = \arctan(b/a) \). The numerical stability of Givens rotations is due to their
orthogonality, i.e. \( G^T(\phi)G(\phi) = I \).

3.1.2. COordinate Rotation DIgital Computer (CORDIC)

The COordinate Rotation DIgital Computer (CORDIC) algorithm [Del89a] [EL90]
computes the Givens rotation by using a sequence of \( w \) micro-rotations, where \( w \)
is the word length:

\[
G(\phi) = S \cdot \tilde{J}(\phi) = \cos(\phi) \begin{bmatrix}
1 & \tan(\phi) \\
-\tan(\phi) & 1
\end{bmatrix}
\] (3.2)

\[
\approx \prod_{\ell=0}^{w-1} \frac{1}{\sqrt{1 + 2^{-2\ell}}} \prod_{\ell=0}^{w-1} \begin{bmatrix}
1 & \mu_\ell \cdot 2^{-\ell} \\
-\mu_\ell \cdot 2^{-\ell} & 1
\end{bmatrix} \tilde{J}(\phi)
\]

where \( \mu_\ell \in \{+1, -1\} \). Applying \( \tilde{J}(\phi) \) to \( z \), only shift-add operations are required.
The CORDIC angles \( \phi(\ell) = \arctan(2^{-\ell}) \), \( \ell \in \{0, 1, 2, \ldots, w - 1\} \) (see Tab. 3.1),
form a discrete basis for the angle \( \phi \) [Zar04]:

\[
\phi \approx \sum_{\ell=1}^{w} \mu_\ell \phi(\ell).
\] (3.3)
Since for a given \( w \), the factor \( S = \prod_{\ell=0}^{w-1} \sqrt{1 + 2^{-2\ell}} = \text{const} \), the multiplication by the binary representation of \( S \) can also be realized by a sequence of shift-add operations. Therefore, requiring only shift-add operations CORDIC is highly suited for hardware implementations.

### 3.1.3. CORDIC-Based Approximate Rotation

Instead of annihilating \( b \) (\( b' = 0 \)), a CORDIC-based approximate rotation diminishes \( b \) to \( |b'| = |d||b| \) with \( 0 \leq |d| < 1 \) [Goe94] [GPS93]. The CORDIC-based approximate Givens rotations \( \tilde{G}(\phi(\ell)) \) is computed by determining the shift value \( \ell \), \( \ell \in \mathcal{I} = \{0, 1, 2, \ldots, w-1\} \) corresponding to the CORDIC angle \( \phi(\ell) = \arctan(2^{-\ell}) \), which is closest to the exact rotation angle \( \phi \). The CORDIC-based approximate rotation

\[
\tilde{G}(\phi(\ell)) = \frac{1}{\sqrt{1 + 2^{-2\ell}}} \begin{bmatrix} 1 & \mu_\ell \cdot 2^{-\ell} \\ -\mu_\ell \cdot 2^{-\ell} & 1 \end{bmatrix} = \tilde{S}(\ell) \cdot \tilde{J}(\phi(\ell)) \tag{3.4}
\]

is only one single step of the CORDIC sequence, i.e. one micro-rotation. \( \tilde{G}(\phi(\ell)) \) is also orthogonal, i.e. \( \tilde{G}(\phi(\ell))^T \tilde{G}(\phi(\ell)) = I \).

The optimal CORDIC-based approximate rotation given by the optimal angle \( \phi(\ell) \) is defined by the CORDIC angle \( \phi(i) \) (\( i \in \mathcal{I} \)) which is closest to the exact rotation angle \( \phi \), i.e.

\[
|\phi(\ell) - \phi| = \min_{i \in \mathcal{I}} |\phi(i) - \phi|. \tag{3.5}
\]

For example, in Fig. 3.3 for \( \phi = 19^\circ \), \( \ell = 2 \) is used, since \( \phi(2) = 14.036^\circ \) is the CORDIC angle closest to \( \phi \), i.e. \( \tilde{G}(\phi(\ell)) \) is applied. This optimal \( \phi(\ell) \) can

<table>
<thead>
<tr>
<th>( \ell )</th>
<th>( \phi(\ell) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>45(^\circ)</td>
</tr>
<tr>
<td>1</td>
<td>26.565(^\circ)</td>
</tr>
<tr>
<td>2</td>
<td>14.036(^\circ)</td>
</tr>
<tr>
<td>3</td>
<td>7.125(^\circ)</td>
</tr>
<tr>
<td>4</td>
<td>3.5763(^\circ)</td>
</tr>
<tr>
<td>\vdots</td>
<td>\vdots</td>
</tr>
</tbody>
</table>
be determined following the ideas presented in [GH94, GPS93]. Let \( \exp(a) \) and \( \text{man}(a) \) \((0.5 \leq \text{man}(a) < 1)\) denote the exponent and the mantissa of a binary floating point number \( a \), respectively. Then

1. according to \( \tan(\phi) = y/x \approx 2^{-\ell} \) an estimate for \( \ell \) can be obtained by
   \[
   \ell_e = \exp(y) - \exp(x); \tag{3.6}
   \]

2. since \( \text{man}(y)/\text{man}(x) \in [0.5, 2[ \), one obtains
   \[
   \ell \in \mathcal{I} = \{\ell_e - 1, \ell_e, \ell_e + 1\}; \tag{3.7}
   \]

3. by computing \( [\hat{x}, \hat{y}]^T = \tilde{\mathbf{J}}(\phi(\ell)) \cdot [x, y]^T \) with \( \ell \in \mathcal{I} \) and checking the sign of \( \hat{y} \), the optimal value for \( \ell \) can be identified [GH94].

This procedure guarantees \( |d| \leq 1/3 \).

The main difference to the original CORDIC is that only one specific rotation angle \( \phi(\ell) \) is chosen (the original CORDIC uses all \( \phi(i) \) for \( i = 0, 1, \ldots, w \)).
3.1. Approximate Solution of LS Problems

Therefore, contrary to the original CORDIC algorithm (constant scaling factor) the scaling factor
\[ S(\ell) = \frac{1}{\sqrt{1 + 2^{-2\ell}}} \] (3.8)
depends on the value of \( \ell \). The different \( S(\ell) \) can for example be implemented in a lookup table.

Another way for an easy scaling factor compensation is to change the approximate rotation slightly. Using a double rotation with \( \ell = \ell + 1 \) avoids the square root in the scaling factor [Del89b]. Therefore, the unscaled approximate rotation \( \tilde{J}(\phi(\ell)) \) is given by
\[
\tilde{J}(\phi(\ell)) = \tilde{J}(\phi(\ell)) \cdot \tilde{J}(\phi(\ell)) = \begin{bmatrix}
1 - 2^{-2\ell} & \mu 2^{-\ell+1} \\
-\mu 2^{-\ell+1} & 1 - 2^{-2\ell}
\end{bmatrix}.
\] (3.9)

The scaled rotation is \( G_d(\phi(\ell)) = \tilde{S}(\ell) \tilde{J}(\phi(\ell)) \), where \( \tilde{S}(\ell) = 1/(1 + 2^{-2\ell}) \) can recursively be computed by shift–and–add operations as follows [GPS93]:
\[
\tilde{S}(\ell) = (1 - 2^{-2\ell}) \prod_{i=1}^{c} (1 + 2^{-2^{i+1}+\ell}) \text{ with } c = \log_2 \left[ \frac{w}{2\ell} \right].
\] (3.10)

The number of shift and add operations for scaling decreases for increasing values of \( \ell \) (small angles), e.g., for \( \ell > w/2 \) no scaling is required at all. This procedure slightly changes the original approximation and yields \( |d| < 0.51 \).

Depending on the required approximation accuracy, CORDIC-based approximate rotation can be applied \( \text{itg} \) times. For example (Fig. 3.3): applying the optimal angle \( \phi(2) = 14.036^\circ \) to the vector with \( \phi = 19^\circ \), in the first step (\( \text{itg} = 1 \)) \( \phi = 19^\circ - 14.036^\circ = 4.964^\circ \) remains. Then, in the next step (\( \text{itg} = 2 \)) the optimal angle is \( \phi(4) = 3.5763^\circ \), such that now \( \phi = 4.964^\circ - 3.5763^\circ = 1.3877^\circ \) remains, and so on.

3.1.4. QR Decomposition (QRD)

The QR decomposition of a \( m \times n \) matrix \( A = Q \cdot R \) can be computed by applying a sequence of Givens rotations \( G(\phi_{ij}) \) (\( \phi_{ij} = \arctan(a_{ij}/a_{jj}) \)) to the matrix, such that the matrix entries below the main diagonal of \( A \) are annihilated [GL96]. The resulting matrix is upper triangular and denoted by \( R \). The product of all required Givens rotations forms the orthogonal matrix
\[
Q^T = \prod_{j=1}^{n} \prod_{i=j+1}^{m} G(\phi_{ij}),
\] (3.11)
such that
\[
\min_x \|Ax - b\|_2^2 = \min_x \|Rx - Q^Tb\|_2^2
\] (3.12)
and the solution \(x\) can be computed by back substitution.

3.1.5. Approximate QRD

Instead of annihilating the elements in the lower triangular of matrix during the course of the QRD, each matrix element is only reduced by an approximate rotation using \(itg\) CORDIC angles. Since the matrix elements below the main diagonal are no longer annihilated, the QRD using CORDIC–based approximate rotations yields approximate solutions of the LS problem. The accuracy of the approximation of the rotations, which is directly related to the accuracy of the LS solution, is determined by \(itg\).

3.1.6. Applying the Approximate QRD to GPS/DGPS

In this thesis, approximate QRD is applied to GPS and DGPS positioning for solving the LS problems in (2.7) and (2.23) respectively.

- GPS positioning: the LS problem in each iteration \(itr\) is solved approximately, where the approximation accuracy is defined by \(itg\).

- DGPS positioning: only one linear LS problem is required and this linear LS problem is solved approximately, where the approximation accuracy is defined by \(itg\).

Very few steps, i.e. \(itg \ll w\), of the CORDIC–based approximate rotation are sufficient to obtain similar results as using exact rotations. The micro-rotations (shift-add operations) are implemented with word length \(w\) (usually \(w = 32\text{bits}\)). The approximation concerns only the number \(itg\). While the conventional implementation needs \(w\) micro-rotations per exact rotation, the presented approximate implementation only performs \(itg\) micro-rotations. This yields a reduction in computational complexity by roughly \(itg/w\) (we will show in the following section that in some cases even \(itg = 2\) gives reasonable results). Furthermore, as this method only requires shift and add operations, it is very well suited for hardware implementation. The silicon design studies undertaken for \(itg = 3\) (see table 2 of
show that compared to the conventional CORDIC algorithm, approximate rotations require roughly 40% of the total computation time for a similar silicon area cost. Furthermore, they require only half of the power of the conventional CORDIC algorithm, since the clock frequency of the approximate rotations is roughly 50% of the conventional CORDIC.

Note that, the approximate methods are numerically as stable as the conventional positioning algorithms, since the CORDIC angles yield a discrete basis for the exact rotation angle and the approximate rotations are orthogonal (see also [PGS95]). Applying the presented approximate methods to GPS/DGPS positioning, with increasing itg the positioning results converge to the positioning results using the conventional method.

3.1.7. Applying the Approximate QRD to Recursive DGPS

The presented idea is particularly suited for recursive positioning methods, e.g. for a recursive version of the presented DGPS algorithm (see section 2.1.3). The QRD using CORDIC-based approximate rotations is used to solve \( \min_{x_{t-1}} \| A_{t-1} x_{t-1} - b_{t-1} \|_2^2 \), i.e. \( A_{t-1} = Q_{t-1} R_{t-1} \) and \( x_{t-1} \) are computed. \( Q_{t-1}^T \) is a product of rotations (parametrization of \( Q_{t-1} \) by \( G(\phi_{ij}) \), see (3.11)) and each rotation is represented by a sequence of \( \mu_\ell \) (parametrization of \( G(\phi) \) by the sequence \( \mu_\ell \), see (3.4)). Therefore, only the parametrization of the orthogonal matrix \( Q_{t-1} \) is stored. In the next step, we compute \( A_{tr} = Q_{t-1}^T A_{t} \) and \( b_{tr} = Q_{t-1}^T b_{t} \) before solving

\[
\min_{x_t} \| A_{tr} x_t - b_{tr} \|_2^2. \tag{3.13}
\]

\( A_{tr} \) is an approximation to the upper triangular matrix \( R_t \), i.e. \( A_{tr} \approx R_t = Q_t^T A_t \).

The presented QRD using CORDIC-based approximate rotations is used to solve (3.13). Since the lower triangular elements of \( A_{tr} \) are small, here only small rotation angles are required, which is particularly advantageous for CORDIC-based approximate rotations. For a small rotation angle (\( \phi \)), a small number of optimal CORDIC angles (small itg) results in accurate rotations (see the example of Fig. 3.3). On the contrary, the conventional CORDIC implementation, which executes the whole sequence (\( \ell = (0, 1, 2, \ldots, w - 1) \)) of CORDIC angles, does not benefit from small angles. A small angle is increased in the first step (\( \ell = 0 \)) before it will be reduced again in the later steps (\( \ell = (1, 2, \ldots, w - 1) \)).

To clarify the improved performance of recursive DGPS in comparison to DGPS, one data set of GPS data for recursive DGPS is taken (see chapter 3.2) in the
following example. QRD using CORDIC based approximate rotation is used for recursive DGPS with \( itg = 2 \). The approximate orthogonal matrix \( Q_{t-1} \) is known, so we obtain: \( A_{tr} = Q_{t-1}^T \cdot A_t \), where

\[
A_t = \begin{bmatrix}
0.58468 & 0.75504 & 0.29677 \\
0.21308 & -0.69241 & 0.68932 \\
0.74916 & -0.42511 & 0.50797 \\
0.51218 & 0.27159 & 0.81480 \\
0.01941 & -0.85896 & 0.51168
\end{bmatrix}
\]

and

\[
A_{tr} = \begin{bmatrix}
1.1005 & 0.089264 & 1.0251 \\
2.6578 \cdot 10^{-4} & -1.4261 & 0.54577 \\
1.9641 \cdot 10^{-4} & 8.908 \cdot 10^{-4} & -0.63108 \\
9.636 \cdot 10^{-4} & 8.544 \cdot 10^{-4} & 2.1986 \cdot 10^{-3} \\
1.6348 \cdot 10^{-5} & -9.309 \cdot 10^{-4} & 1.0812 \cdot 10^{-5}
\end{bmatrix}
\]

\( \approx R_t \)

Obviously, the lower triangular elements of \( A_{tr} \) are significantly smaller than the lower triangular elements of \( A_t \), such that small rotation angles, e.g. \( \phi_{21} = \arctan(\frac{2.6578 \cdot 10^{-4}}{1.1005}) = 0.0138^\circ \), are sufficient to solve (3.13).

### 3.2. Experimental Results

We now analyze the accuracy of the positioning results for GPS using the non-linear LS method and for DGPS using the linear LS method. These results are compared with the results obtained by the conventional method using the Matlab function for QRD. For GPS, the accuracy of the positioning results is compared for varying \( itr \) and \( itg \). For DGPS, the accuracy of the positioning results for varying \( itg \) is compared. The Positioning Errors (PE) are measured by the distances of the estimated positions \( \mathbf{x}_{est} \) and the exact position \( \mathbf{x}_{ex} \). For each measurement \( i = 1, 2, \cdots n \), we have \( PE_i = \| \mathbf{x}_{est,i} - \mathbf{x}_{ex} \|_2 \). We compare the Mean Positioning Error (MPE) and the Standard Deviation of the PE (SDPE) for \( n \) measurements, which are

\[
MPE = \frac{1}{n} \sum_{i=1}^{n} (PE_i) \quad \text{and} \quad SDPE = \sqrt{\frac{1}{n-1} \sum_{i=1}^{n} (PE_i - MPE)^2},
\]

respectively.
3.2. Experimental Results

3.2.1. GPS Data

Rinex data and raw GPS data are both used for GPS/DGPS positioning. Using the Rinex data allows to demonstrate that the presented methods work with high accuracy. Using the raw GPS data shows the performance of the methods using low-end receivers. In both cases we also demonstrate that with low approximation accuracy (low computational efforts) GPS can be used to obtain network position accuracy and DGPS can be used to obtain GPS position accuracy.

3.2.1.1. Rinex Data

Our evaluation uses the Rinex files [Gur07] from the high-end GPS receiver of the Danish GPS center [Aal10]. These data include observation and navigation files of the satellites. Each Rinex file contains 2880 data sets for a whole day (one data set is gained every 30 seconds). Every data set at the measurement point includes information of \( m = 7 \) visible satellites. 100 Rinex files are used in our experiment, i.e. there are totally \( n = 288000 \) data sets with which we solved the positioning problem. Position calculation is done using all presented positioning methods. For GPS positioning, the center of the Earth \((0, 0, 0)\) is chosen as the initial point. The calculated positions are compared with the exact position (known coordinates from the Danish GPS Center [Aal10]). In the case of DGPS positioning using Rinex data, the distance between the reference station and the receiver is 0.5km. Since Rinex files for the reference station were not available, we simulated the pseudorange measurement of the reference station by adding an additional AWGN error with variance of \( \sigma^2 = (2.8)^2 \) (chosen according to the experimental results in Tab. 3.4) to the pseudorange measurements obtained from the Rinex data.

3.2.1.2. Raw Data

A SiGe GN3S Sampler v2 is used to capture the raw GPS data, which are low level signal data (raw intermediate frequency samples) and processed by the SiGe radio front end [Pro10]. The obtained raw GPS data at the measurement points include information of \( m = 7 \) visible satellites and the respective pseudoranges. \( n = 78 \) raw GPS data records are gained every 20 seconds. Position calculation is done for all positioning methods. The calculated positions are compared with the exact positions (known coordinates at the measurement points from land surveying office
3.2.2. Positioning Results

3.2.2.1. Experimental Results for GPS

For GPS positioning, approximate solutions of the LS problem in each iteration are obtained and the trade-off between \( \text{itr} \) and \( \text{itg} \) in terms of positioning accuracy is analyzed using Matlab. These results are also compared with the ones from the conventional method. As reference, we show the result from the conventional method in the 1st row of all tables (i.e. 'exact'). The MPE of position estimates using Rinex data for varying number of iterations \( \text{itr} \) (see Sec. 2.1.1.1) and varying approximation accuracies \( \text{itg} \) (see Sec. 3.1) is shown in Tab. 3.2, while MPE of GPS position estimates with raw GPS data is presented in Tab. 3.3. The results show that by increasing \( \text{itg} \), the accuracy of the positioning results converge towards the exact result.

By increasing \( \text{itg} \), the accuracy of the positioning results increases too. Note that if \( \text{itr} \) is sufficiently large, e.g. \( \text{itr} \geq 3 \), only three optimal CORDIC angles \( \text{itg} = 3 \) are required to obtain network positioning accuracy of around 300m [JSL03]. And if \( \text{itr} \geq 4 \), the typical GPS accuracy can be achieved with only two optimal CORDIC angles \( \text{itg} = 2 \), i.e. around 10m using Rinex data and around 100m using raw data. Therefore, very coarse approximations are sufficient for obtaining the required positioning accuracies.

The SDPE of the position estimates using Rinex data is shown in Tab. 3.4. A low standard deviation around 2.8m is obtained after a few iterations. The SDPE of GPS positioning results using raw GPS data is presented in Tab. 3.5. Comparing Tab. 3.3 and 3.5 with Tab. 3.2 and 3.4, the estimated positions and the standard deviations are much worse for the raw data, which reflects the quality of the the low-end GPS receiver used in these experiments.

By increasing \( \text{itg} \), the accuracy of the exact QRD methods is achieved. In order to demonstrate more clearly how the results are obtained, we show the computed position estimates using Rinex data for \( \text{itr} = 4 \) and \( \text{itg} = 2 \) in Fig. 3.4 as well as for \( \text{itr} = 4 \) and \( \text{itg} = 6 \) in Fig. 3.5. All the positioning results are considered as accuracy of position, i.e. we show the position estimates subtracted by the exact position. Fig. 3.4(b) is an enlarged part of Fig. 3.4(a), while Fig. 3.5(b) is an enlarged part of Fig. 3.5(a).
### Table 3.2: MPE of positioning results (in meter) by GPS using Rinex data

<table>
<thead>
<tr>
<th>↓itr\↑itg→</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>exact</td>
<td>43420</td>
<td>48.993</td>
<td>7.106</td>
<td>7.106</td>
<td>7.106</td>
</tr>
<tr>
<td>1</td>
<td>274890</td>
<td>43134</td>
<td>7031.5</td>
<td>1249.2</td>
<td>232.84</td>
</tr>
<tr>
<td>2</td>
<td>154220</td>
<td>330.15</td>
<td>14.181</td>
<td>8.401</td>
<td>8.086</td>
</tr>
<tr>
<td>3</td>
<td>45875</td>
<td>55.851</td>
<td>10.945</td>
<td>8.082</td>
<td>7.712</td>
</tr>
<tr>
<td>4</td>
<td>43883</td>
<td>53.060</td>
<td>7.106</td>
<td>7.106</td>
<td>7.106</td>
</tr>
<tr>
<td>5</td>
<td>43561</td>
<td>50.749</td>
<td>7.106</td>
<td>7.106</td>
<td>7.106</td>
</tr>
<tr>
<td>6</td>
<td>43448</td>
<td>49.898</td>
<td>7.106</td>
<td>7.106</td>
<td>7.106</td>
</tr>
<tr>
<td>7</td>
<td>43436</td>
<td>48.999</td>
<td>7.106</td>
<td>7.106</td>
<td>7.106</td>
</tr>
<tr>
<td>8</td>
<td>43431</td>
<td>48.998</td>
<td>7.106</td>
<td>7.106</td>
<td>7.106</td>
</tr>
<tr>
<td>9</td>
<td>43420</td>
<td>48.995</td>
<td>7.106</td>
<td>7.106</td>
<td>7.106</td>
</tr>
<tr>
<td>10</td>
<td>43420</td>
<td>48.993</td>
<td>7.106</td>
<td>7.106</td>
<td>7.106</td>
</tr>
<tr>
<td>11</td>
<td>43420</td>
<td>48.993</td>
<td>7.106</td>
<td>7.106</td>
<td>7.106</td>
</tr>
<tr>
<td>12</td>
<td>43420</td>
<td>48.993</td>
<td>7.106</td>
<td>7.106</td>
<td>7.106</td>
</tr>
<tr>
<td>13</td>
<td>43420</td>
<td>48.993</td>
<td>7.106</td>
<td>7.106</td>
<td>7.106</td>
</tr>
</tbody>
</table>

### Table 3.3: MPE of positioning results (in meter) by GPS using raw GPS data

<table>
<thead>
<tr>
<th>↓itr\↑itg→</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>exact</td>
<td>54222</td>
<td>251.892</td>
<td>62.171</td>
<td>62.171</td>
<td>62.171</td>
</tr>
<tr>
<td>1</td>
<td>127250</td>
<td>14085.4</td>
<td>4767.1</td>
<td>973.53</td>
<td>175.28</td>
</tr>
<tr>
<td>2</td>
<td>101650</td>
<td>92724</td>
<td>102.28</td>
<td>64.610</td>
<td>63.538</td>
</tr>
<tr>
<td>3</td>
<td>60144</td>
<td>1297.27</td>
<td>71.384</td>
<td>62.279</td>
<td>62.246</td>
</tr>
<tr>
<td>4</td>
<td>54281</td>
<td>257.262</td>
<td>62.227</td>
<td>62.184</td>
<td>62.171</td>
</tr>
<tr>
<td>5</td>
<td>54237</td>
<td>248.383</td>
<td>62.171</td>
<td>62.171</td>
<td>62.171</td>
</tr>
<tr>
<td>6</td>
<td>54225</td>
<td>242.059</td>
<td>62.171</td>
<td>62.171</td>
<td>62.171</td>
</tr>
<tr>
<td>7</td>
<td>54222</td>
<td>241.914</td>
<td>62.171</td>
<td>62.171</td>
<td>62.171</td>
</tr>
<tr>
<td>8</td>
<td>54222</td>
<td>241.910</td>
<td>62.171</td>
<td>62.171</td>
<td>62.171</td>
</tr>
<tr>
<td>9</td>
<td>54222</td>
<td>231.899</td>
<td>62.171</td>
<td>62.171</td>
<td>62.171</td>
</tr>
<tr>
<td>10</td>
<td>54222</td>
<td>231.892</td>
<td>62.171</td>
<td>62.171</td>
<td>62.171</td>
</tr>
<tr>
<td>11</td>
<td>54222</td>
<td>231.892</td>
<td>62.171</td>
<td>62.171</td>
<td>62.171</td>
</tr>
<tr>
<td>12</td>
<td>54222</td>
<td>231.892</td>
<td>62.171</td>
<td>62.171</td>
<td>62.171</td>
</tr>
<tr>
<td>13</td>
<td>54222</td>
<td>231.892</td>
<td>62.171</td>
<td>62.171</td>
<td>62.171</td>
</tr>
</tbody>
</table>
Table 3.4: SDPE of positioning results (in meter) by GPS using Rinex data

<table>
<thead>
<tr>
<th>itg</th>
<th>itr</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>exact</td>
<td>8655.2</td>
<td>19.502</td>
<td>2.813</td>
<td>2.813</td>
<td>2.813</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>244910</td>
<td>52095</td>
<td>11680</td>
<td>2790.4</td>
<td>717.23</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>40627</td>
<td>1367.6</td>
<td>60.670</td>
<td>4.567</td>
<td>2.723</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>11686</td>
<td>334.73</td>
<td>4.207</td>
<td>2.818</td>
<td>2.814</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>8686.7</td>
<td>58.369</td>
<td>2.816</td>
<td>2.813</td>
<td>2.813</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>8678.4</td>
<td>23.65</td>
<td>2.814</td>
<td>2.813</td>
<td>2.813</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>8655.5</td>
<td>19.528</td>
<td>2.813</td>
<td>2.813</td>
<td>2.813</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>8655.3</td>
<td>19.514</td>
<td>2.813</td>
<td>2.813</td>
<td>2.813</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>8655.2</td>
<td>19.506</td>
<td>2.813</td>
<td>2.813</td>
<td>2.813</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>8655.2</td>
<td>19.502</td>
<td>2.813</td>
<td>2.813</td>
<td>2.813</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>8655.2</td>
<td>19.502</td>
<td>2.813</td>
<td>2.813</td>
<td>2.813</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>8655.2</td>
<td>19.502</td>
<td>2.813</td>
<td>2.813</td>
<td>2.813</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>8655.2</td>
<td>19.502</td>
<td>2.813</td>
<td>2.813</td>
<td>2.813</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>8655.2</td>
<td>19.502</td>
<td>2.813</td>
<td>2.813</td>
<td>2.813</td>
<td></td>
</tr>
</tbody>
</table>

Table 3.5: SDPE of positioning results (in meter) by GPS using raw GPS data

<table>
<thead>
<tr>
<th>itg</th>
<th>itr</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>exact</td>
<td>88.677</td>
<td>41.579</td>
<td>41.547</td>
<td>41.547</td>
<td>41.547</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>1371.8</td>
<td>891.38</td>
<td>89.568</td>
<td>45.489</td>
<td>43.082</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>42976</td>
<td>951.97</td>
<td>45.888</td>
<td>41.816</td>
<td>41.791</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>1087.7</td>
<td>495.04</td>
<td>41.701</td>
<td>41.632</td>
<td>41.619</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>992.00</td>
<td>82.101</td>
<td>41.553</td>
<td>41.550</td>
<td>41.552</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>199.85</td>
<td>46.451</td>
<td>41.547</td>
<td>41.547</td>
<td>41.547</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>91.940</td>
<td>41.675</td>
<td>41.547</td>
<td>41.547</td>
<td>41.547</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>88.875</td>
<td>41.673</td>
<td>41.547</td>
<td>41.547</td>
<td>41.547</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>88.865</td>
<td>41.588</td>
<td>41.547</td>
<td>41.547</td>
<td>41.547</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>88.708</td>
<td>41.582</td>
<td>41.547</td>
<td>41.547</td>
<td>41.547</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>88.694</td>
<td>41.578</td>
<td>41.547</td>
<td>41.547</td>
<td>41.547</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>88.677</td>
<td>41.579</td>
<td>41.547</td>
<td>41.547</td>
<td>41.547</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>88.677</td>
<td>41.579</td>
<td>41.547</td>
<td>41.547</td>
<td>41.547</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>88.677</td>
<td>41.579</td>
<td>41.547</td>
<td>41.547</td>
<td>41.547</td>
<td></td>
</tr>
</tbody>
</table>
In these figures, circles present individual positioning results from 100 GPS data records with exact QRD; a bold plus presents the mean value of the positioning results from exact QRD, which corresponds to the exact solution in Tab. 3.2 for

Figure 3.4: (a) GPS positioning results and (b) enlarged positioning results (in meter) using Rinex data for \(itr = 4\) and \(itg = 2\)
$itr = 4$. Stars present the positioning results from our approach using approximate QRD (in Fig. 3.4 ($itg = 2$) and in Fig. 3.5 ($itg = 6$)); a bold star presents the mean value of the positioning results from our approach, which corresponds to

Figure 3.5.: (a) GPS positioning results and (b) enlarged positioning results (in meter) using Rinex data for $itr = 4$ and $itg = 6$
the respective values (for $itr = 4$ and $itg = 2$ resp. for $itr = 4$ and $itg = 6$) in Tab. 3.2. Correspondingly, for raw GPS data, the position estimates of 78

**Figure 3.6.** (a) GPS positioning results and (b) enlarged positioning results (in meter) using raw GPS data for $itr = 4$ and $itg = 2$
measurements using approximate QRD with \( itg = 2 \) and \( itg = 6 \) are shown in Fig. 3.6 and Fig. 3.7 respectively.

**Figure 3.7:** (a) GPS positioning results and (b) enlarged positioning results (in meter) using raw GPS data for \( itr = 4 \) and \( itg = 6 \)
Table 3.6: MPE and SDPE of positioning results (in meter) by DGPS and recursive DGPS (for 0.5 km distance between 2 receivers) using Rinex data

<table>
<thead>
<tr>
<th>itg</th>
<th>MPE</th>
<th>SDPE</th>
<th>MPE-Rec</th>
<th>SDPE-Rec</th>
</tr>
</thead>
<tbody>
<tr>
<td>exact</td>
<td>0.140</td>
<td>1.001</td>
<td>0.140</td>
<td>1.001</td>
</tr>
<tr>
<td>1</td>
<td>162170</td>
<td>170750</td>
<td>475.450</td>
<td>5530.7</td>
</tr>
<tr>
<td>2</td>
<td>34234</td>
<td>32444</td>
<td>118.77</td>
<td>1322.3</td>
</tr>
<tr>
<td>3</td>
<td>1796.3</td>
<td>9889.1</td>
<td>41.782</td>
<td>358.75</td>
</tr>
<tr>
<td>4</td>
<td>119.44</td>
<td>2075.8</td>
<td>10.104</td>
<td>96.388</td>
</tr>
<tr>
<td>5</td>
<td>2.635</td>
<td>395.84</td>
<td>1.951</td>
<td>20.143</td>
</tr>
<tr>
<td>6</td>
<td>1.108</td>
<td>72.598</td>
<td>0.645</td>
<td>6.859</td>
</tr>
<tr>
<td>7</td>
<td>0.766</td>
<td>13.51</td>
<td>0.194</td>
<td>1.478</td>
</tr>
<tr>
<td>8</td>
<td>0.172</td>
<td>2.703</td>
<td>0.153</td>
<td>1.042</td>
</tr>
<tr>
<td>9</td>
<td>0.142</td>
<td>1.150</td>
<td>0.153</td>
<td>1.042</td>
</tr>
<tr>
<td>10</td>
<td>0.141</td>
<td>1.024</td>
<td>0.144</td>
<td>0.999</td>
</tr>
<tr>
<td>11</td>
<td>0.140</td>
<td>1.001</td>
<td>0.140</td>
<td>1.001</td>
</tr>
<tr>
<td>12</td>
<td>0.140</td>
<td>0.999</td>
<td>0.140</td>
<td>1.001</td>
</tr>
<tr>
<td>13</td>
<td>0.140</td>
<td>1.001</td>
<td>0.140</td>
<td>1.001</td>
</tr>
</tbody>
</table>

Note that when the accuracy of approximate QRD increases, i.e, itg increases, the positioning results become more and more similar to the results of exact QRD. The stars in Fig. 3.5 and Fig. 3.7 (itg = 6) are closer to the circles than the stars in Fig. 3.4 and Fig. 3.6 (itg = 2). Thus, the mean value of the positioning results for itg = 6 is almost the same as from exact QRD method. Furthermore, in Fig. 3.4(b) and Fig. 3.6(b) even the positioning results for itg = 2 (stars) are already close to the exact solutions (circles). The difference between the mean value of the positioning result from our approach and the exact QRD is only 7 m using Rinex data and 40 m using raw GPS data, therefore itg = 2 is sufficient to obtain network positioning accuracy.

The same behavior can be seen for all the following experiments. Please note that with increasing itg the LS solution by approximate QRD obviously converges to the exact LS solution. Hence, we will not show the detailed results given in Fig. 3.4 - Fig. 3.5 using Rinex data and Fig. 3.6 - Fig. 3.7 using raw data for the experimental results of DGPS again.
3.2.2. Experimental Results for DGPS

For DGPS with approximate QRD (see Sec. 2.1.2) we calculated the position with varying approximation accuracies \( \text{itg} \). These results are compared with the ones from the conventional method. As reference, we show the result from the conventional method in the 1st row of all the tables (i.e. 'exact'). The 2nd and the 3rd columns of Tab. 3.6 and Tab. 3.7 present the MPE and SDPE of DGPS position estimates using Rinex data and raw GPS data, respectively.

With increasing \( \text{itg} \) in the approximate QRD, the accuracy of positioning results also increases and converges to the result of exact QRD. The MPE in Tab. 3.6 and Tab. 3.7 shows that the accuracy of the exact QRD is achieved, if \( \text{itg} \geq 11 \). Using Rinex data, if \( \text{itg} \geq 6 \), very high positioning accuracy (around 1m) can be reached. In order to get the GPS-level positioning accuracy, i.e. around 10m, only \( \text{itg} = 5 \) is required using Rinex data and \( \text{itg} = 8 \) is required with raw GPS data. The results show that even for low-end receivers the accuracy of DGPS is about 10m, which is the accuracy we usually expect from GPS receivers. This shows that coarse approximations are sufficient in DGPS positioning to get GPS positioning.
accuracy.

3.2.2.3. Experimental Results for Recursive DGPS

For recursive DGPS method (see Sec. 3.1.7), we calculated the position with varying approximation accuracies \( itg \). The 4th (MPE-Rec) and the 5th (SDPE-Rec) column of Tab. 3.6 and Tab. 3.7 present the MPE and SDPE of recursive DGPS using Rinex data and raw GPS data respectively.

The recursive DGPS positioning requires a even smaller \( itg \) to achieve the same accuracy as DGPS. For example, in order to achieve GPS-level positioning accuracy (around 10m), recursive DGPS requires only \( itg = 4 \) CORDIC rotations (using Rinex data) and \( itg = 3 \) (using raw data), while DGPS requires \( itg = 5 \) (using Rinex data) and \( itg = 8 \) (using raw data). The results show that very small rotation angles are sufficient to achieve a high approximation accuracy for recursive DGPS. Therefore, the presented method is particularly suited for recursive DGPS.

3.3. Conclusion

The presented positioning methods provide a trade-off between positioning accuracies and computational complexities. QR Decomposition (QRD) using COordinate Rotation DIgital Computer (CORDIC) based approximate rotations is applied to the positioning algorithms for GPS (linear LS problem solved in each iteration) and DGPS (one linear LS problem) as well as recursive DGPS. To suit the different use cases of LBS, it is possible to choose the positioning accuracy by choosing the numbers of required iterations and the number of optimal CORDIC-angles. Even with small number of CORDIC-angles positioning accuracies sufficient for many LBS applications can be achieved. Therefore, a significant reduction in computational complexity and power consumption can be obtained depending on the use cases. Because this method only requires shift and add operations, it is very well suited for hardware implementation.
4. Learning Geographic Regions Using Location Based Services in IMS

Once the positioning is done by the users using a positioning method (e.g. GPS, DGPS), we assume that the position estimates are available in the network for providing LBS.

This chapter presents how to apply classification [Seb04] [McL04] [WD94] for learning geographic regions using LBS in IMS. The information in local network (cellular network) can be freely exchanged with global IP network (IMS) and the information can be gathered in a database. Positioning methods provide location information for the data sets. Statistical classification methods are applied to the data sets in the database. Two cases are distinguished:

- Depending on the information provided by the users, they are divided into different groups. This process is defined as Type Filter (TF) in IMS. A classification application server (CAS) uses discriminant analysis or a local-nearest neighbor algorithm to perform the learning of the geographic regions. The learned geographic regions of the groups can be used to define Location Filters (LF). This case is denoted as 'TF-CAS-LF'.

- The observed area (predefined) is divided into different geographic regions, i.e. the regions are known and fixed. This process is defined as Location Filter (LF) in IMS. The CAS uses discriminant analysis or a local-nearest neighbor algorithm to distinguish patterns of behavior inside these regions of the groups. The learned behavior can be used to define Type Filter (TF). This case is denoted as 'LF-CAS-TF'.

These cases can be used to establish services (e.g. warning services for the vehicles) in this region or other regions over IMS Network. Therefore, the presented concept
4. Learning Geographic Regions Using Location Based Services in IMS

4.1. IMS-for-Vehicle Architecture

Fig. 4.1 shows the architecture of the IMS Network for vehicles. The vehicles are equipped with a lot of sensors, mobile communication devices (GPRS, UMTS, LTE...) as well as positioning devices (GPS, Assisted-GPS, network positioning). It is assumed that the vehicles are already registered as IMS clients, so that the vehicles in the cellular local network can access the global IP network over IMS.

4.1.1. Local Network Access

Vehicles are all connected to the cellular local network (through base stations) and are deployed to collect and transmit real-time traffic and environmental data, e.g. driving habits, roadway congestion and sensor states. The positioning devices of the vehicles provide the current positions of the vehicles. With the help of all

Figure 4.1.: Vehicle-to-IMS architecture
4.2. Classification and Filter Functions

these devices, a data set can be obtained, which includes all the information from
the vehicle (e.g. sensor data) and from mobile communication devices (e.g. Cell
ID, GPS position). The exemplary data set in Fig. 4.1 contains the information
about the User ID, the time stamp, the user’s Cell ID, the position of the user
and states of user’s car (e.g. driving speed, fog lamp on/off, rain warning sensor
on/off). Using the data sets of all the users, a database can be established.

4.1.2. Global Network Access

The continuous communication between local network and IMS Network is achieved
by the Session Initial Protocol (SIP), a main signaling protocol used in IMS Net-
work for establishing, modifying and terminating multimedia session (see section
2.3). This kind of communication provides an interface to retrieve user local in-
formation frequently. It also enables the gathering of all the information in the
database. Furthermore, it makes up-dated information available to other IMS
clients who are in different local networks, i.e. once geographic regions are learned,
this information can be broadcasted to other users in the vicinity.

4.2. Classification and Filter Functions

In order to learn geographic regions, multivariate observations of the given sensor
data from the cars in the observed area are studied [Seb04]. The statistical analysis
of many variables effectively refers to the study of vectors of correlated random
variables, e.g. \( n \) random vectors \( s_1, s_2, \ldots, s_n \), each of dimension \( d \). These vectors
arise from taking measurements or characteristics on \( d \) variables for each of \( n \)
objects. The variables \( s_{ij} \) within each \( s_i = [s_{i1}, s_{i2}, \ldots, s_{id}] \) are generally correlated.
The variables may be quantitative (discrete or continuous) or qualitative (ordered
or unordered).

For example, let \( S \) be the collection of \( n \) data sets \( s_i = [s_{i1}, s_{i2}, \ldots, s_{id}] \), i.e. \( d \)
variables are obtained for each of the \( n \) objects. It is convenient to express the
database in matrix \( S \) with random variables \( s_{ij} \), namely,

\[
S = \begin{bmatrix}
s_{11} & s_{12} & \cdots & s_{1d} \\
s_{21} & s_{22} & \cdots & s_{2d} \\
\vdots & \vdots & \ddots & \vdots \\
s_{n1} & s_{n2} & \cdots & s_{nd}
\end{bmatrix}.
\] (4.1)
4. Learning Geographic Regions Using Location Based Services in IMS

![Diagram](image)

**Figure 4.2.**: Learning geographic regions by applying filters and classification to the database in IMS

Given such a database in IMS, Fig. 4.2 shows the overall procedure for learning geographic regions by applying filters and classification. Subsection 4.2.1 describes the case TF-CAS-LF (see left part of Fig. 4.2). Subsection 4.2.2 describe the case LF-CAS-TF (see right part of Fig. 4.2).

### 4.2.1. TF-CAS-LF

An example of a data set \(s_i\) of the database \(S\) is given as in Tab. 4.1. The following \(d = 6\) variables are recorded:

- User-ID: \(s_{i1}\);
- Date-Time: \(s_{i2}\) (the time stamp of the data collected by the user);
- Cell ID: \(s_{i3}\) (the Cell ID of the user);
- Location: \(s_{i4} = [x_i, y_i]\) (the position of the user);
4.2. Classification and Filter Functions

<table>
<thead>
<tr>
<th>User ID</th>
<th>Date ID</th>
<th>Cell ID</th>
<th>Location ID</th>
<th>Fog Lamp</th>
<th>Rain Sensor</th>
</tr>
</thead>
<tbody>
<tr>
<td>$s_{i1}$</td>
<td>$s_{i2}$</td>
<td>$s_{i3}$</td>
<td>$s_{i4}$</td>
<td>$s_{i5}$</td>
<td>$s_{i6}$</td>
</tr>
</tbody>
</table>

Table 4.1.: Data sets in IMS database $S$

- FogLamp: $s_{i5} \in \{ 'f', 'n' \}$, where $s_{i5} = 'f'$ if the fog lamp is turned on and $s_{i5} = 'n'$ otherwise;

- RainSensor: $s_{i6} \in \{ 'r', 'n' \}$, where $s_{i6} = 'r'$ if the rain warning sensor is turned on and $s_{i6} = 'n'$ otherwise.

Here $s_{i1}, s_{i2}, s_{i3}$ and $s_{i4}$ are discrete quantitative variables, while $s_{i5}$ and $s_{i6}$ are unordered qualitative variables.

Because in this thesis vehicles are taken as IMS clients, it is assumed that only the data sets from vehicles are analyzed. Furthermore, only the data sets for a certain period of time are taken. Using the resulting database classification methods are applied.

4.2.1.1. Type Filter

Based on the different information in the variables, the data sets can be divided into $K$ groups. This procedure is known as Type Filter (TF).

In our example, the data sets of car-sensor states are divided into three states, which are 'fog', 'rain' and 'none', denoted as 'f', 'r' and 'n' in the following. On the basis of $n$ data sets, the positions of cars are assigned to three states (fog/rain/none).

The well-defined clusters are obtained by a type filter which is realized by accessing only a part of the whole data set:

$$g_1 = \{ s_{i4} | s_{i5} = 'f' \}$$
$$g_2 = \{ s_{i4} | s_{i6} = 'r' \}$$
$$g_3 = \{ s_{i4} | s_{i5} = 'n' \land s_{i6} = 'n' \}$$

(4.2)

where $g_1$ is the group in which all fog lamps are on; $g_2$ is the group in which all rain sensors are on; and in group $g_3$, neither the fog lamps nor the rain sensors are on. Note that both fog and rain hardly happen at the same time, so it will not be considered here.
In the following example there are \( n = 7 \) users and \( K = 3 \) groups. \( n_k \) is the number of elements in group \( k \) (\( k = 1, 2, \ldots, K \)). The information of the database required here is a subset of the database \( S \) (the 4th, 5th and 6th column of \( S \)) and is presented in matrix \( S_s \):

\[
S_s = \begin{bmatrix}
    s_{i4} & s_{i5} & s_{i6} \\
    [x_1 \ y_1] & n & n \\
    [x_2 \ y_2] & f & n \\
    [x_3 \ y_3] & n & r \\
    [x_4 \ y_4] & n & r \\
    [x_5 \ y_5] & f & n \\
    [x_6 \ y_6] & n & n \\
    [x_7 \ y_7] & f & n
\end{bmatrix}
\] (4.3)

Now the group vectors \( g_1, g_2 \) and \( g_3 \) can be determined by applying (4.2) to matrix \( S_s \), and we obtain:

\[
g_1 = \begin{bmatrix}
    [x_2 \ y_2] \\
    [x_5 \ y_5] \\
    [x_7 \ y_7]
\end{bmatrix}, \quad i_1 = \{2, 5, 7\}
\]

\[
g_2 = \begin{bmatrix}
    [x_3 \ y_3] \\
    [x_4 \ y_4]
\end{bmatrix}, \quad i_2 = \{3, 4\}
\] (4.4)

\[
g_3 = \begin{bmatrix}
    [x_1 \ y_1] \\
    [x_6 \ y_6]
\end{bmatrix}, \quad i_3 = \{1, 6\}
\]

with the corresponding index sets \( i_k \).

## 4.2.1.2. Classification

Classification can be done by discriminant analysis or many other methods using the following input data:

- the group vectors \( g_1, g_2 \) and \( g_3 \) for training;
- the sample data which could originate from the possible positions in the considered region.

In the first classification approach, a quadratic discrimination function [Seb04] is used to separate different swarms of points in the region. In the above example,
there are totally \( n = 7 \) users and \( K = 3 \) groups \( g_1, g_2 \) and \( g_3 \). There are \( n_1 = 3 \), \( n_2 = 2 \) and \( n_3 = 2 \) users in these groups respectively. Let \([x_1 \ y_1]\ldots[x_{n_k} \ y_{n_k}]\) be the subset of the training data belonging to group \( k \). For every group, the sample mean value of the training data:

\[
[x_k \ y_k] = \frac{1}{n_k} \sum_{i \in i_k} [x_i \ y_i]
\]  

(4.5)

and the sample covariance matrix of the training data

\[
S_k = \frac{1}{n_k - 1} \sum_{i \in i_k} ([x_i \ y_i] - [x_k \ y_k])^T ([x_i \ y_i] - [x_k \ y_k])
\]  

(4.6)

are computed. With \(|S_k|\) being the determinant of \(S_k\), the following parameters for each group can be computed:

\[
H_k = \frac{1}{2} ([x_k \ y_k] S_k^{-1} [x_k \ y_k]^T + \log(|S_k|)),
\]  

(4.7)

\[
r_k = S_k^{-1} [x_k \ y_k]^T,
\]  

(4.8)

\[
Q_k = -\frac{1}{2} S_k^{-1}.
\]  

(4.9)

Let \( k \) and \( \ell \) be two different groups in the region. Now with \( v = [x_s \ y_s]^T \), the sample data in the considered region (local map), the quadratic discriminant function between group \( k \) and \( \ell \) can be computed as follows:

\[
f_{k,\ell}(v) = (H_k - H_\ell) + v^T [r_k - r_\ell] + v^T [Q_k - Q_\ell] v.
\]  

(4.10)

The positions \([x_s \ y_s]^T\) for which \( f_{k,\ell}(x_s, y_s) = 0 \) determine the border between groups \( k \) and \( \ell \). Once the discriminant analysis for all possible pairs of groups \((k, \ell)\) is done, the geographic regions are successfully learned (see experimental results in Sec. 4.4) and a location filter can be defined.

Local-nearest neighbor algorithm is another method to classify the elements in the geographic regions. Each element in the region is compared with its neighborhood using the Euclidean distance. The neighborhood is pre-defined by the area inside a circle with radius \( r \) around each analyzed element. Depending on the observed region, the \( r \) can be chosen in different values. If the observed region is big, \( r \) is also big; and if the observed region is small, then \( r \) is small.

For each observed element \( i \) with position \([x_i \ y_i]\) \((i = 1, 2, \ldots n)\), we only look at all elements \( j \) with the position \([x_j \ y_j]\) \((j = 1, 2, \ldots n)\) in the circle of radius \( r \) around \( i \), i.e. \( S_{si} \) is now defined for each \( i \) as follows:

\[
S_{si} = \{[[x_j \ y_j] \ s_j1 \ s_j2] | (x_j - x_i)^2 + (y_j - y_i)^2 \leq r^2\}.
\]  

(4.11)
Therefore, $S_{si}$ looks like $S_s$ in (4.3), but only contains the elements inside the circle around the observed element $i$. Now we can define the groups from $S_{si}$ as in (4.4) resulting in $g_{s1}$, $g_{s2}$ and $g_{s3}$.

$n_{ik}$ is the number of elements in group $g_{ik}$. For each group the sum of the distances $dis_{ik}$ from the analyzed element $[x_i \ y_i]$ to all the other elements $[x_j \ y_j]$ of group $g_{ik}$ are calculated.

\[
\text{dis}_{ik} = \sum_{j \in g_{ik}} (([x_i \ y_i] - [x_j \ y_j])^T([x_i \ y_i] - [x_j \ y_j]))^{\frac{1}{2}}.
\]  (4.12)

The maximum value of the number of elements in the groups is:

\[
mx_i = \max(n_{i1} \ n_{i2} \cdots n_{iK}),
\]  (4.13)

and how often this maximum value occurs is given by:

\[
\text{num}_{mx_i} = |\{n_{ik}|n_{ik} = mx_i\}|,
\]  (4.14)

where $|\cdot|$ denotes the cardinality of a set.

Depending on $num_{mx_i}$ and $dis_{ik}$, the type of the analyzed element can be decided:

- If $num_{mx_i} = 1$, i.e. one group dominates the region and there is only one maximum value of the number of elements in groups, we will look for $n_{ik} = mx_i$ and change the state of $i$ to the state of group $g_{ik}$;

- If $1 < num_{mx_i} < K$, i.e. many groups have the maximum value of the number of elements in groups at the same time, $dis_{ik}$ will additionally be taken into account. The state of $i$ is then changed to the state of the group with the smallest $dis_{ik}$ among these groups;

- If $num_{mx_i} = K$, i.e. all the groups are equally distributed in the region and the maximum value of the elements in each group is same, we will keep the state of $i$.

Additionally, if the observed region is big, i.e., if $r$ is defined with a big radius, there will be a large number of elements in each group. In this case, it is worthwhile to allow for some margins (like 5% or 10%) to make the algorithm more robust.
4.2.1.3. Location Filter

The learned region of each group can be used to define a Location Filter (LF). Depending on different LFs in various regions, IMS sends different warnings to the users who are in these areas (see Fig. 4.3). Cars in the fog or rain area, which have not turned on their fog lamp/windshield wiper (without light symbol in Fig. 4.3), will get a warning from IMS. The warning (fog/rain-warning) service from the IMS Network will also be demanded by users who are going to enter that LF-region for safe-driving support. This service helps the IMS clients to notice special features of the geographic surrounding immediately, so that they can have enough time to prepare to upcoming situations.

4.2.2. LF-CAS-TF

In this approach, first LFs are used to define regions, so the position of the region is known and fixed. Each element in the region has its own state, which is taken as training data. Classification method is applied to these training data to figure out
the state of the LF-region (group class). Depending on the different group states in the various LF-regions, TFs can adjust the states of the groups.

This kind of concept can be applied to traffic speed control, in order to keep a smooth traffic flow on the highway [Ker04]. Since the positions of the highways are all known, the whole highway can be divided into several sections by using LFs, see rectangles forming LF shown in Fig. 4.3. The state (speed) of each car in every section is used as training data in the classification method, which finds out the group state (group speeds/class) in this region. Then, TFs will define the speed of the cars which is going to enter this region.

For example, in Fig. 4.3 assuming the traffic situation in LF3 is still smooth and the speed of the cars in this region is fast. The group speed in LF2-region in front of LF3 is slowed down. The LF1-region has traffic jam, so the speed in this region is very slow. Therefore, the traffic density in LF1 should be reduced. In order to do this, IMS can warn the cars in LF2 and LF3, which are behind LF1 and going to enter LF1, and ask them to slow down their speed, i.e. using TF to change the users type/state, in order to help ease the traffic jam situation in front of them. Otherwise, the cars in LF2 and LF3 should change the route, i.e. keeping the state and changing the group.

4.3. Design and Architecture of the Presented Concept

The realization of the proposed concept in IMS Network is shown in this section. It is based on the service-orientated architecture of IMS. Figure 4.4 shows the architecture of the concept. The data sets of the users from all the regions who have access to the IMS are collected in a master user database in IMS. Referring to the problem of learning geographic regions by using LBS in IMS, a CAS is additionally implemented in IMS.

4.3.1. IMS Location Server (IMSLS)

IMSLS [MSKK06] is designed as a SIP Application Server located in the IMS Application Layer. Considering the job priority of the different network layers, the positioning job is not done in IMSLS, but in Positioning Server (PS) (in the transport layer) [HB09a] or in User Equipment (UE) (in the physical layer). Therefore, IMSLS is like a location service client which just obtains the location informa-
4.3. Design and Architecture of the Presented Concept

Figure 4.4.: Architecture of applying classification with LBS in IMS to learn geographic regions

Two positioning methods are distinguished. If user 1 receives satellite signals, the location of this user can be obtained by GPS. The positioning will be done directly in the user equipment (UE1). The position information will be delivered to Call Session Control Function (CSCF) and then to the IMSLS through SIP. If user 2 does not receive satellite signals but has connection to base stations, the positioning will be done by PS in the network, which is composed of Position Calculation Server (PCS) and Assisted Positioning Server (APS) [HB09a]. The position information will be send directly to IMSLS. After successful positioning, the users’ Cell ID as well as the users’ location will be delivered to IMSLS with time stamp and is available for the database.

4.3.2. Group List Manage Server (GLMS)

This server allows users to store service-specific data in the service provider network. These data can be created, modified and deleted at will by the user. Data could be anything that a user needs to complete a service. This enables data to
be shared and accessed by the services that need them [PNKM06]. Here GLMS contains a IMS user group member list, e.g. the User ID and the sensor state with time stamp. All the information is sent to GLMS by Hypertext Transfer Protocol (HTTP) [KGL07].

### 4.3.3. Classification Application Server (CAS)

For the 'TF-CAS-LF' case, CAS determines the geographic regions of the different groups using the position information carried by the users. After receiving a SIP request for a group session, the CAS contacts the GLMS to retrieve the IMS member list. The session setup is explained in chapter 5. The server checks the state of each member, e.g. checking whether their fog lamps or rain sensors are on/off. According to the state of the users, the members will be divided into groups. Then the CAS retrieves the location information from the IMSLS for each member of a group. After obtaining the positions of each member of every user group, which form the training data for classification, the geographic region can be learned.

For the 'LF-CAS-TF' case, CAS together with IMSLS defines the location of the LF-regions. After contacting GLMS, CAS can determine the group in the region depending on the states of users (training data). Once the states of the geographic regions are learned, TF are defined, which can be used to adjust the states of the groups who are approaching these LF-regions.

### 4.4. Implementation and Experimental Results

A desktop PC equipped with an Intel® Core™ 2 Quad Q6600 CPU with 2.4 GHz, 4 GB of memory and a gigabit Ethernet controller is used. Besides the software specific to the individual tests, this computer runs a name server for resolution of the IMS domain and a MySQL database. Fraunhofer’s OpenIMSCore [Fra08] is implemented as IMS server. The server machine runs all of the Call Session Control Functions (CSCFs) [PNKM06] in parallel. Clients for all scenarios are laptops running pjsua [PJS06].

In the experiment, applying the classification methods for learning the weather conditions in the region is chosen, so that IMS can offer warning services to the drivers in and approaching this region. The experiments are done by running several clients on the laptops, which send their data (location, sensor states, etc.)
4.4. Implementation and Experimental Results

Figure 4.5.: Data sets of one user in IMS
to the database. The information (data sets) of the users is simulated and is transmitted to IMS database via SIP. Figure 4.5 shows the data sets of one user in IMS database. Except for the last column Car-Informed which is explained in section 4.4.2, the structure of the data sets is the same as in Tab. 4.1.

4.4.1. Analyzing the performance of the classification methods

In order to evaluate the performances of the discriminant analysis and the local-nearest neighbor algorithm, a ground truth scenario is constructed as follows: there are \( n = 150 \) users and each of the three regions contains 50 users, i.e. \( n_1 = 50, n_2 = 50 \) and \( n_3 = 50. \) \( e \) errors are added to the ground truth, where \( e \) is 0%, 8%, 20% and 50% of \( n. \) The errors are equally distributed. For example, for 8% errors there are 4% errors for each error type. We use \( P = 10 \) runs \( p = \{ 1, 2, \ldots P \} \) with independent error distributions for each \( e. \) The experimental results for one specific \( p \) are shown in in Fig. 4.6 - Fig. 4.9 for each \( e. \) The respective locations and states of the users are presented in these figures. Red crosses indicate that the fog lamps of these cars are on; blue triangles indicate that the rain sensors of these cars are on; and black points indicate that the car devices are off. The \( X \) and \( Y \) coordinates are scaled by 30m, i.e. 100 indicates 3km on the maps.

Now, the discriminant analysis and the local-nearest neighbor algorithm are applied to learn the geographic regions. The results are compared to the ground truth. Let \( CFD_p \) be the confusion matrix obtained after the classification. We
evaluate the mean of the confusion matrices:

\[
MCF = \frac{1}{P} \sum_{p=1}^{P} CFD_p
\]  

(4.15)

and the variance of the confusion matrices:

\[
VCF = \frac{1}{P - 1} \sum_{p=1}^{P} (CFD_p - MCF)^2.
\]  

(4.16)

The mean of the confusion matrices \(MCF\) (see Tab. 4.3 - Tab. 4.6) and their variance \(VCF\) (see Tab. 4.7 - Tab. 4.9) are shown in order to compare the performances of the used classification methods. The first column of all the tables shows the three groups of sensors’ states pre-defined by a ground truth with errors \(e(\text{‘\textquoteLEFT}gt\textquoteRIGHT\text{’})\) and saved as training data sets in IMS. The first row of these tables (‘cl’) shows the classified sensors’ states of the cars in the observed region. The values of the off-diagonal elements in the confusion matrices indicate the errors in the observed regions. The experimental results show that both classification methods can learn the geographic regions.

Using the discriminant analysis method (Fig. 4.6(a) - Fig. 4.9(a)), TFs divide the positions into different groups (fog, rain, none). Then, the CAS can determine the respective geographic regions. The fog region is inside the pink ellipse, while the rain region is inside the green ellipse. From now on, the geographic regions are learned and LFs can be generated.

Using the local-nearest neighbor algorithm (Fig. 4.6(b) - Fig. 4.9(b)), TFs divide the states of the cars in the defined region into different groups (fog, rain, none). The neighborhood is defined by radius \(r = 3\)km. Then, the CAS can determine the state of the sensor of each car individually depending on the states of its neighbors. A mismatch between the desired true state and the recommendation derived from the classification is indicated by circles. The red circle is a warning signal indicating that the car is currently in the fog area, but the current state of its sensor does not match the estimated geographic situation saved in IMS (the fog lamp should be turned on). The blue circle is a warning signal indicating that the car is currently in the rain area, but the current state of its sensor does not match the estimated geographic situation saved in IMS (the windshield wiper should be turned on). The black circle is a warning signal which indicates the car is currently in normal area without rain or fog and the fog lamp or windshield wiper should be turned off. These warning signals are the results of the local-nearest neighbor classification.
With the discriminant analysis, a simple function directly representing the geographic region can be obtained. That means that the regions are explicitly defined. The states of the elements can be easily evaluated depending on the explicitly defined regions (inside or outside of the ellipses). Comparing the mean of the confusion matrices of the discriminant analysis to the mean of the confusion matrices of the local-nearest neighbor algorithm (Tab. 4.3 - 4.6), the discriminant analysis exhibits bigger errors than the local-nearest neighbor algorithm. This shows that the discriminant analysis learns the region not as precise as the local-nearest neighbor algorithm. Table 4.7 - 4.9 show that the variance of the discriminant analysis is smaller than the variance of the local-nearest neighbor algorithm. This means that once the geographic region is learned, the pre-defined data saved in the IMS Network can also be used for the changed scenarios. The information update, i.e. redefinition of the situation of the regions, needs not to be done very often. This benefits the large-range traffic situation control.

With the local-nearest neighbor algorithm, no functions explicitly define the regions. This method approximates the regions locally and all computation is deferred until classification. Each element must be compared with other elements in its neighborhood individually. From Tab. 4.3 - 4.6 it is obvious that the off-diagonal elements of the confusion matrices of the local-nearest neighbor algorithm are much smaller than those of the discriminant analysis. The local-nearest neighbor algorithm learns the region more precisely than the discriminant analysis. Table 4.7 - 4.9 show that the variance of the local-nearest neighbor algorithm is bigger than the variance of the discriminant analysis. This means that once the geographic region has changed, the pre-defined data saved in the IMS Network should be learned again. The database should be updated more often than for the discriminant analysis. This method benefits the small-range urban area traffic control, where the precise region situation should be figured out and the traffic scenario changes frequently.

An overview of the properties of both methods is shown in Tab. 4.2.

<table>
<thead>
<tr>
<th></th>
<th>accuracy</th>
<th>sensitivity</th>
<th>update</th>
</tr>
</thead>
<tbody>
<tr>
<td>discriminant analysis</td>
<td>low</td>
<td>low</td>
<td>infrequently</td>
</tr>
<tr>
<td>local-nearest neighbor</td>
<td>high</td>
<td>high</td>
<td>frequently</td>
</tr>
</tbody>
</table>

Table 4.2: Comparison of the characteristics of the discriminant analysis and the local-nearest neighbor algorithm
4.4.2. Classification

After analyzing the performance of the two classification methods using a ground truth with errors scenario, we look at these two methods for an instantaneous scenario at time $t$. We simulated $n = 119$ users in the region. $n_1 = 36$ of them are red crosses (fog); $n_2 = 60$ of them are blue triangles (rain) and $n_3 = 23$ are black points (none). The region is learned and the data is saved in the data bank in IMS as the ground truth at time $t$. After that at time $t + \Delta t$, some new users are coming into the observed region, while some older user have left the region. These users are classified. Table 4.10 and Tab. 4.11 present the classification results using the discriminant analysis method and the local-nearest neighbor algorithm in percentage at time $t$ and $t + \Delta t$ respectively. The first column of both tables (‘eref’) shows the estimated sensors’ states of the cars saved as training data sets in the data bank in IMS in the observed region. The first row of these tables (‘cl’) shows the classified sensors’ states of the cars in the observed region. After the classification, the warning service from IMS can be realized. The warning server in IMS sends different warnings to the users in the different areas, see the last column Car-Informed in Fig. 4.5. A warning signal 'Warning-Fog’/’Warning-Rain’ is sent to the users who are in or approaching the fog/rain area, in order to remind the drivers to turn on the fog lamps or the windshield wiper. Otherwise the warning service is in IDLE state.

The respective locations and states of the cars are visualized in Fig. 4.10 and Fig. 4.11. The discriminant analysis (see Fig. 4.10(a) and Fig. 4.11(a)) and the local-nearest neighbor algorithm (see Fig. 4.10(b) and Fig. 4.11(b)) are used to classify the users’ states.

In both cases, due to the movement of the cars and the change of the weather situation, the data sets of the geographic region in the IMS server should be redefined in certain time periods. The time periods are adaptive and can be determined by observing the off-diagonal elements of the confusion matrices based on the actual data as shown in Tab. 4.11. Therefore, at time $t + n\Delta t$ the geographic region should be learned again, if the off-diagonal elements of the actual confusion matrix are bigger than a predefined percentage depending on the use cases. And the data sets saved in the data bank should also be updated, i.e. new data sets should be uploaded and saved in IMS data bank at time $t + n\Delta t$. 
4.5. Conclusion

This chapter presents classification methods to determine geographic regions using LBS in IMS. Establishing a CAS service in IMS and combining it with the GLMS achieves TF, which divides all IMS clients into groups due to different variables in users’ data sets. CAS using classification methods, which are discriminant analysis or local-nearest neighbor algorithm, together with IMSLS determine the position of the user group depending on the location of each member in every group. CAS can also be used together with IMSLS to define the location of the groups. After contacting GLMS, CAS can determine the class of the groups depending on the states of users. Because the states of the users change from time to time, the learned region must be updated from time to time. The time period of the update can be determined by observing the confusion matrix of the classification method. The proposed method can be used to realize up-to-date driver support. After geographic regions are learned by LBS in IMS, active safety traffic conditions are achieved via warning services from IMS.
4. Learning Geographic Regions Using Location Based Services in IMS

(a) Discriminant analysis

<table>
<thead>
<tr>
<th></th>
<th>f-cl</th>
<th>r-cl</th>
<th>n-cl</th>
</tr>
</thead>
<tbody>
<tr>
<td>f-gtwe</td>
<td>54%</td>
<td>44%</td>
<td>2%</td>
</tr>
<tr>
<td>r-gtwe</td>
<td>14%</td>
<td>80%</td>
<td>6%</td>
</tr>
<tr>
<td>n-gtwe</td>
<td>12%</td>
<td>0%</td>
<td>88%</td>
</tr>
</tbody>
</table>

(b) Local-nearest neighbor

<table>
<thead>
<tr>
<th></th>
<th>f-cl</th>
<th>r-cl</th>
<th>n-cl</th>
</tr>
</thead>
<tbody>
<tr>
<td>f-gtwe</td>
<td>92%</td>
<td>4%</td>
<td>4%</td>
</tr>
<tr>
<td>r-gtwe</td>
<td>0%</td>
<td>100%</td>
<td>0%</td>
</tr>
<tr>
<td>n-gtwe</td>
<td>4%</td>
<td>6%</td>
<td>90%</td>
</tr>
</tbody>
</table>

Table 4.3.: The confusion matrix ($P = 1$ run) in percentage with 0% errors by using (a) the discriminant analysis method and (b) the local-nearest neighbor algorithm for three classes fog(f), rain (r) and none (n)

(a) Discriminant analysis

<table>
<thead>
<tr>
<th></th>
<th>f-cl</th>
<th>r-cl</th>
<th>n-cl</th>
</tr>
</thead>
<tbody>
<tr>
<td>f-gtwe</td>
<td>49%</td>
<td>40%</td>
<td>11%</td>
</tr>
<tr>
<td>r-gtwe</td>
<td>20%</td>
<td>72%</td>
<td>8%</td>
</tr>
<tr>
<td>n-gtwe</td>
<td>9%</td>
<td>3%</td>
<td>88%</td>
</tr>
</tbody>
</table>

(b) Local-nearest neighbor

<table>
<thead>
<tr>
<th></th>
<th>f-cl</th>
<th>r-cl</th>
<th>n-cl</th>
</tr>
</thead>
<tbody>
<tr>
<td>f-gtwe</td>
<td>90%</td>
<td>4%</td>
<td>6%</td>
</tr>
<tr>
<td>r-gtwe</td>
<td>2%</td>
<td>96%</td>
<td>2%</td>
</tr>
<tr>
<td>n-gtwe</td>
<td>4%</td>
<td>7%</td>
<td>89%</td>
</tr>
</tbody>
</table>

Table 4.4.: Mean of the confusion matrices ($P = 10$ runs) in percentage with 8% errors by using (a) the discriminant analysis method and (b) the local-nearest neighbor algorithm for three classes fog(f), rain (r) and none (n)

(a) Discriminant analysis

<table>
<thead>
<tr>
<th></th>
<th>f-cl</th>
<th>r-cl</th>
<th>n-cl</th>
</tr>
</thead>
<tbody>
<tr>
<td>f-gtwe</td>
<td>44%</td>
<td>38%</td>
<td>18%</td>
</tr>
<tr>
<td>r-gtwe</td>
<td>28%</td>
<td>62%</td>
<td>10%</td>
</tr>
<tr>
<td>n-gtwe</td>
<td>16%</td>
<td>8%</td>
<td>76%</td>
</tr>
</tbody>
</table>

(b) Local-nearest neighbor

<table>
<thead>
<tr>
<th></th>
<th>f-cl</th>
<th>r-cl</th>
<th>n-cl</th>
</tr>
</thead>
<tbody>
<tr>
<td>f-gtwe</td>
<td>86%</td>
<td>5%</td>
<td>9%</td>
</tr>
<tr>
<td>r-gtwe</td>
<td>4%</td>
<td>91%</td>
<td>5%</td>
</tr>
<tr>
<td>n-gtwe</td>
<td>7%</td>
<td>10%</td>
<td>83%</td>
</tr>
</tbody>
</table>

Table 4.5.: Mean of the confusion matrices ($P = 10$ runs) in percentage with 20% errors using (a) the discriminant analysis method and (b) the local-nearest neighbor algorithm for three classes fog(f), rain (r) and none (n)
4.5. Conclusion

(a) Discriminant analysis

<table>
<thead>
<tr>
<th></th>
<th>f-cl</th>
<th>r-cl</th>
<th>n-cl</th>
</tr>
</thead>
<tbody>
<tr>
<td>f-gtwe</td>
<td>44%</td>
<td>38%</td>
<td>18%</td>
</tr>
<tr>
<td>r-gtwe</td>
<td>33%</td>
<td>49%</td>
<td>18%</td>
</tr>
<tr>
<td>n-gtwe</td>
<td>20%</td>
<td>20%</td>
<td>60%</td>
</tr>
</tbody>
</table>

(b) Local-nearest neighbor

<table>
<thead>
<tr>
<th></th>
<th>f-cl</th>
<th>r-cl</th>
<th>n-cl</th>
</tr>
</thead>
<tbody>
<tr>
<td>f-gtwe</td>
<td>71%</td>
<td>13%</td>
<td>16%</td>
</tr>
<tr>
<td>r-gtwe</td>
<td>17%</td>
<td>68%</td>
<td>15%</td>
</tr>
<tr>
<td>n-gtwe</td>
<td>23%</td>
<td>16%</td>
<td>61%</td>
</tr>
</tbody>
</table>

Table 4.6.: Mean of the confusion matrices ($P = 10$ runs) in percentage with 50% errors by using [a] the discriminant analysis method and [b] the local-nearest neighbor algorithm for three classes fog(f), rain (r) and none (n)

(a) Discriminant analysis

<table>
<thead>
<tr>
<th></th>
<th>f-cl</th>
<th>r-cl</th>
<th>n-cl</th>
</tr>
</thead>
<tbody>
<tr>
<td>f-gtwe</td>
<td>2.2E-4</td>
<td>1.5E-4</td>
<td>4.3E-4</td>
</tr>
<tr>
<td>r-gtwe</td>
<td>1.6E-3</td>
<td>1.8E-3</td>
<td>1.4E-4</td>
</tr>
<tr>
<td>n-gtwe</td>
<td>4.8E-4</td>
<td>1.8E-4</td>
<td>5.1E-4</td>
</tr>
</tbody>
</table>

(b) Local-nearest neighbor

<table>
<thead>
<tr>
<th></th>
<th>f-cl</th>
<th>r-cl</th>
<th>n-cl</th>
</tr>
</thead>
<tbody>
<tr>
<td>f-gtwe</td>
<td>1.7E-1</td>
<td>1.3E-1</td>
<td>3.3E-3</td>
</tr>
<tr>
<td>r-gtwe</td>
<td>3.4E-2</td>
<td>5.7E-2</td>
<td>3.1E-3</td>
</tr>
<tr>
<td>n-gtwe</td>
<td>2.4E-3</td>
<td>1.8E-3</td>
<td>5.5E-4</td>
</tr>
</tbody>
</table>

Table 4.7.: Variance of the confusion matrix ($P = 10$ runs) in percentage with 8% errors by using [a] the discriminant analysis method and [b] the local-nearest neighbor algorithm for three classes fog(f), rain (r) and none (n)

(a) Discriminant analysis

<table>
<thead>
<tr>
<th></th>
<th>f-cl</th>
<th>r-cl</th>
<th>n-cl</th>
</tr>
</thead>
<tbody>
<tr>
<td>f-gtwe</td>
<td>3.3E-4</td>
<td>3.2E-4</td>
<td>8.9E-4</td>
</tr>
<tr>
<td>r-gtwe</td>
<td>2.0E-3</td>
<td>1.3E-3</td>
<td>3.8E-4</td>
</tr>
<tr>
<td>n-gtwe</td>
<td>6.5E-4</td>
<td>5.4E-4</td>
<td>1.5E-3</td>
</tr>
</tbody>
</table>

(b) Local-nearest neighbor

<table>
<thead>
<tr>
<th></th>
<th>f-cl</th>
<th>r-cl</th>
<th>n-cl</th>
</tr>
</thead>
<tbody>
<tr>
<td>f-gtwe</td>
<td>1.8E-1</td>
<td>1.1E-1</td>
<td>7.4E-3</td>
</tr>
<tr>
<td>r-gtwe</td>
<td>5.7E-2</td>
<td>8.4E-2</td>
<td>2.9E-3</td>
</tr>
<tr>
<td>n-gtwe</td>
<td>9.7E-3</td>
<td>9.6E-4</td>
<td>5.8E-3</td>
</tr>
</tbody>
</table>

Table 4.8.: Variance of the confusions matrices ($P = 10$ runs) in percentage with 20% errors by using [a] the discriminant analysis method and [b] the local-nearest neighbor algorithm for three classes fog(f), rain (r) and none (n)
80 4. Learning Geographic Regions Using Location Based Services in IMS

(a) Discriminant analysis

<table>
<thead>
<tr>
<th></th>
<th>f-cl</th>
<th>r-cl</th>
<th>n-cl</th>
</tr>
</thead>
<tbody>
<tr>
<td>f-gtwe</td>
<td>2.4E-4</td>
<td>5.5E-5</td>
<td>1.7E-4</td>
</tr>
<tr>
<td>r-gtwe</td>
<td>3.6E-4</td>
<td>2.6E-4</td>
<td>6.4E-4</td>
</tr>
<tr>
<td>n-gtwe</td>
<td>4.3E-4</td>
<td>2.3E-4</td>
<td>4.6E-4</td>
</tr>
</tbody>
</table>

(b) Local-nearest neighbor

<table>
<thead>
<tr>
<th></th>
<th>f-cl</th>
<th>r-cl</th>
<th>n-cl</th>
</tr>
</thead>
<tbody>
<tr>
<td>f-gtwe</td>
<td>7.2E-2</td>
<td>6.5E-2</td>
<td>1.4E-3</td>
</tr>
<tr>
<td>r-gtwe</td>
<td>2.6E-2</td>
<td>3.9E-2</td>
<td>2.4E-3</td>
</tr>
<tr>
<td>n-gtwe</td>
<td>2.1E-3</td>
<td>4.2E-3</td>
<td>3.9E-3</td>
</tr>
</tbody>
</table>

Table 4.9.: Variance of the confusion matrices ($P = 10$ runs) in percentage with 50% errors by using (a) the discriminant analysis method and (b) the local-nearest neighbor algorithm for three classes fog (f), rain (r) and none (n)

(a) Discriminant analysis

<table>
<thead>
<tr>
<th></th>
<th>f-cl</th>
<th>r-cl</th>
<th>n-cl</th>
</tr>
</thead>
<tbody>
<tr>
<td>f-eref</td>
<td>67%</td>
<td>29%</td>
<td>4%</td>
</tr>
<tr>
<td>r-eref</td>
<td>14%</td>
<td>86%</td>
<td>0%</td>
</tr>
<tr>
<td>n-eref</td>
<td>14%</td>
<td>14%</td>
<td>72%</td>
</tr>
</tbody>
</table>

(b) Local-nearest neighbor

<table>
<thead>
<tr>
<th></th>
<th>f-cl</th>
<th>r-cl</th>
<th>n-cl</th>
</tr>
</thead>
<tbody>
<tr>
<td>f-eref</td>
<td>89%</td>
<td>8%</td>
<td>3%</td>
</tr>
<tr>
<td>r-eref</td>
<td>8%</td>
<td>90%</td>
<td>2%</td>
</tr>
<tr>
<td>n-eref</td>
<td>13%</td>
<td>4%</td>
<td>83%</td>
</tr>
</tbody>
</table>

Table 4.10.: Confusions matrix in percentage for an instantaneous scenario at time $t$ by using (a) the discriminant analysis method and (b) the local-nearest neighbor algorithm for three classes fog (f), rain (r) and none (n)

(a) Discriminant analysis

<table>
<thead>
<tr>
<th></th>
<th>f-cl</th>
<th>r-cl</th>
<th>n-cl</th>
</tr>
</thead>
<tbody>
<tr>
<td>f-eref</td>
<td>63%</td>
<td>29%</td>
<td>8%</td>
</tr>
<tr>
<td>r-eref</td>
<td>14%</td>
<td>83%</td>
<td>3%</td>
</tr>
<tr>
<td>n-eref</td>
<td>14%</td>
<td>17%</td>
<td>69%</td>
</tr>
</tbody>
</table>

(b) Local-nearest neighbor

<table>
<thead>
<tr>
<th></th>
<th>f-cl</th>
<th>r-cl</th>
<th>n-cl</th>
</tr>
</thead>
<tbody>
<tr>
<td>f-eref</td>
<td>87%</td>
<td>10%</td>
<td>3%</td>
</tr>
<tr>
<td>r-eref</td>
<td>10%</td>
<td>88%</td>
<td>2%</td>
</tr>
<tr>
<td>n-eref</td>
<td>15%</td>
<td>15%</td>
<td>70%</td>
</tr>
</tbody>
</table>

Table 4.11.: Confusions matrix in percentage for an instantaneous scenario at time $t + \Delta t$ by using (a) the discriminant analysis method and (b) the local-nearest neighbor algorithm for three classes fog (f), rain (r) and none (n)
4.5. Conclusion

Figure 4.6: Learned geographic regions using (a) discriminant analysis and (b) local-nearest neighbor algorithm with the ground truth and no errors.
Learning the regions using discriminant analysis with e=8%

Learning the regions using local-nearest neighbor algorithm with e=8%

Figure 4.7: Learned geographic regions using (a) discriminant analysis and (b) local-nearest neighbor algorithm with 8% errors added to the ground truth
Figure 4.8: Learned geographic regions using (a) discriminant analysis and (b) local-nearest neighbor algorithm with 20% errors added to the ground truth.
Figure 4.9: Learned geographic regions using (a) discriminant analysis and (b) local-nearest neighbor algorithm with 50% errors added to the ground truth.
Learning the regions using discriminant analysis at time t

(a) Discriminant analysis

Learning the regions using local-nearest neighbor algorithm at time t

(b) Local-nearest neighbor algorithm

Figure 4.10: Learned geographic regions using (a) discriminant analysis and (b) local-nearest neighbor algorithm for an instantaneous scenario at time t.
Figure 4.11.: Learned geographic regions using (a) discriminant analysis and (b) local-nearest neighbor algorithm for an instantaneous scenario at time $t + \Delta t$. 
5. Implementation on IMS Platform

This chapter focuses on how the IMS server decreases the session setup delay compared to the Traditional Network (T-Net), which uses the OSI layered model described in Sec. 2.3. Voice over IP (VoIP) is used to measure the session setup delay in both networks, since the required software for the experiments is readily available and the presented results can also be applied to other services (e.g., LBS). The time consumption of SIP message transmission flow for the registration and call session setup of VoIP over the Traditional Network (T-Net) and the IMS Network (IMS-Net) are analyzed under the same measurement conditions. The time delay caused by the infrastructure is analyzed, which is independent of the signal propagation methods, the user’s device or the data links. In contrast to other work [VW06], here no matter what kind of network access is used, only internal processing time of the server is measured. Because the SIP message transmission flow from user equipment to proxy for T-Net and IMS-Net are the same, this part of the time delay can be neglected, and only the time delay of the SIP message transmission in the internal server is measured. The result shows that the registration of IMS-Net is longer than that of T-Net, but IMS-Net registration is refreshed with an interval of one week while for T-Net refresh interval is 10 to 30 minutes. This decreases the registration traffic in the IMS Network. The session setup delay of IMS-Net with an average 2.3ms is better than that of T-Net with an average of 3.5ms (see Sec. 5.1.3). Therefore, IMS provides fast and efficient service creation and delivery. Short time data-exchange can be achieved through the IMS Network, so that this can benefit real-time streaming applications for online games, video and VoIP. Additionally, the implementation of an application server, the Classification Application Server (CAS), in the IMS is done. CAS is shared between the local network and the IMS global network. The experimental results show that compared to the T-Net, IMS-Net can offer faster data transmission services for CAS and other real-time applications. Client-to-server (C-S) and server-to-client (S-C) communications (session based communication) have lower time delay for packet transmission than client-to-client (C-C) communica-
tion (message based communication). Therefore, the designed architecture using C-S and S-C communication in IMS Network, which provides network centralized up-to-date information, can be applied to realize driver support and active safety services.

5. Implementation on IMS Platform

5.1. Internal Package Transmission Delay for Registration and Session Setup

The time consumption of SIP message transmission flow for the registration and call session setup of VoIP over T-Net and IMS-Net are analyzed under the same measurement conditions.

5.1.1. Call Setup Description

Call setup time of Voice over IP in the Traditional Network (T-Net) and the IMS Network (IMS-Net) are compared. In each case, delays for registration and call session setup are studied. In this thesis, only the signaling part for initialization and termination of SIP calls were studied and not the IMS specific signaling for QoS nor the quality of the media streams (183 Session Description Protocol, PRACK, RTP etc). This has already been studied in [FCP06]. The results show that the session setup performances are affected by various underlying protocols and the error correction mechanisms could improve the performance of VoIP session setup time by correcting the SIP messages and avoiding retransmissions on the wireless link. Figure 5.1 shows an overview of the test scenario. To provide comparability between a traditional and an IMS-based VoIP system this work conducts all measurements on the same server machine (thus on the same hardware and operating system). Because the focus of the research in this thesis is on the impact of the transaction delay on different network servers, only the core VoIP systems as shown in Fig. 5.1 were observed. Transmission delays induced by the network link from the UEs to the VoIP core as well as processing delays of the UE hardware, which can be found in [MJP07] [YKSE08] [NTW07], are not considered.
5.1.1. Traditional Network (T-Net)

In T-Net SIP is used to establish and terminate a communication session between two entities. SIP uses IP as the network layer protocol, and supports both TCP and UDP as transport layer protocol. In VoIP scenarios UDP is more commonly applied for real-time communication.

- Register: Trying to keep the examination of the VoIP protocols close to a real-world application, the SIP proxy was configured to require registration and authentication. Figure 5.2 shows the T-Net registration process. On the first REGISTER request the client is prompted to authenticate itself. The
registration is successfully done after the second REGISTER gets the reply from SIP-Server with ”200 OK”. As the scope of this thesis is the delay caused by the infrastructure, and not clients or data links, only internal processing time of the server is measured. The total delay therefore is calculated as 
\[ t_{\text{total}} = t_1 + t_2. \]

- Call: Figure 5.3 displays the T-Net session setup process. Due to non-persistent storage of authentication information in T-Net, a client has to re-authenticate the proxy before placing each call. This is achieved via the answer “407 - Proxy Authentication required” to the first INVITE message. To keep the measurements comparable to the IMS, a \( t_A \) represents the delay for generating the “407” response and \( t_B \) represents the delay between the acknowledgment from the client to forwarding the second INVITE. These two delays are added as \( t_1 \). It is comparable to the \( t_1 \) in IMS session setup (see Fig. 5.5). Thus, we can define \( t_1 \) for T-VoIP is \( t_1 = t_A + t_B \) and \( t_{\text{total}} = t_1 + t_2 + t_3 + t_4 + t_5 + t_6. \)

5.1.1.2. IMS-Net

The IP Multimedia Subsystem is also SIP-based. In fact the SIP message flow in IMS is very similar to T-Net if the IMS core is considered as one entity. This allows the comparison of the performance of IMS-Net and T-Net

- Register: As Fig. 5.4 shows, the registration delays are measured between the times when a packet enters the IMS Core and when it leaves the IMS core. The packets are transmitted from P-CSCF via I-CSCF and then to S-CSCF. Compared with T-Net registration (see Fig. 5.2), it is easy to notice that IMS-Net needs more registration time than T-Net. The total delay for an IMS-Net registration operation is calculated (analogous to T-Net) as \( t_{\text{total}} = t_1 + t_2. \) The travel time of packets between UE and P-CSCF as well as the processing time in the UE are not considered in the measurement, since they are outside the scope of the IMS core. The function of the SUBSCRIBE and NOTIFY methods shown in Fig. 5.4 is to inform the P-CSCF about the client’s registration information. As the UE is already aware of it’s successful registration, these methods were not considered as a part of the core registration process and are not included in the measurements.
5.1. Internal Package Transmission Delay for Registration and Session Setup

Call: As mentioned before, only internal SIP message flow delay of IMS Network server is measured to express network-induced delays. The individual sub-delays form the total IMS session delay are shown in Fig. 5.5. Therefore, we obtain $t_{total} = t_1 + t_2 + t_3 + t_4 + t_5 + t_6$. $t_1$ is comparable to the $t_A + t_B$ in T-Net session setup (see Fig. 5.3). Because authentication information is stored persistently in IMS core, the client does not need to re-authenticate before initiating a session. Note that because the experimental setup incorporates only one IMS server realm and I-CSCF is only responsible for inter-domain routing, it is not a part of the routes in this example.
5.1.2. Implementation Environment

5.1.2.1. General Setup Environment

All tests were conducted on a desktop-class machine equipped with an Intel® Core™ 2 Quad Q6600 CPU with 2.4 GHz, 4 GB of memory and a Gigabit-Ethernet controller. The operating system is OpenSuSE 10.3. Besides the software specific to the individual tests, this computer runs a name server for resolution of the IMS and SIP domains and a MySQL database. Clients for all scenarios are laptops running pjsua, a reference implementation of PJSIP, an open source SIP library [PJS06]. Due to its command line interface, it can be easily scripted for autonomous interaction with the SIP / IMS Network.
5.1.2.2. T-Net Environment

Analysis of T-Net performance was done by using the SER, the SIP Express Router, in version 2.0.0 Release Candidate 1. SER is a very flexible open-source SIP-Server, which can be configured to fulfill a variety of purposes. As it was used as a basis for the OpenIMS Core [Fra08], the SER is chosen for the T-Net measurements too. This allows an even better view on the impact of IMS on the session setup and registration delays. SER was configured to act as a SIP registrar using a configuration file provided by the SER buildsystem. User data is stored persistently in the MySQL database.

5.1.2.3. IMS-Net Environment

Performance testing of IMS was done using the Fraunhofer’s reference implementation of an IMS core called OpenIMSCore [Fra08]. This thesis used SVN revision 621 during the tests. The server machine runs all of the CSCFs in parallel. The P-CSCF is configured to listen on the interface with a public IP address, while the I-CSCF and the S-CSCF listen on private IP addresses assigned to virtual
instances of the network interface. For IMS Registration AKAv1-MD5 is used as authentication algorithm [NAT02].

5.1.2.4. Measurement Method

As stated before, two test cases are defined, for both T-Net and IMS-Net: Register and call operations. The focus of the measurements lies on the internal processing of the analyzed system, and not on network parameters. Thus, the measurements take place on the network interface of the server machine, analyzing the exact time when a packet enters or leaves the system. All SIP packets using UDP in the system are captured by tshark (the command line equivalent of wireshark), and stored in a text file for later analysis using the specialized tool, the SIPanalyzer.

The SIPanalyzer was developed in [HVB10] to automatically analyze SIP sessions from tshark trace files by identifying a packet and joining them in the sessions they belong to. These sessions are analyzed and discarded if they do not follow the packet flow depicted in section 5.1.1. Based on this data the delays as introduced in section 5.1.1 are calculated and exported in Comma Separated Volumes (CSV) for further processing with Matlab or standard office software.

The SIP traffic for the tests was generated by one or more pjsua clients (depending on the scenario for registration or for session) running on different computers. These clients were automated by piping commands to them from Linux shell scripts.

5.1.3. Test Results

Table 5.1 and Fig. 5.6 show the comparison of registration delays between T-Net and IMS-Net while Tab. 5.2 and Fig. 5.7 show the comparison of session setup delays between them. Frequency distributions of the session setup delays of T-Net and IMS-Net for one connection between 2 clients is shown in Fig. 5.8. Frequency distributions for higher number of clients (in this case 500) and various numbers

<table>
<thead>
<tr>
<th>Registration delay (ms)</th>
<th>T-Net</th>
<th>IMS-Net</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean</td>
<td>2.651</td>
<td>47.745</td>
</tr>
<tr>
<td>Minimum</td>
<td>0.408</td>
<td>33.365</td>
</tr>
<tr>
<td>Maximum</td>
<td>70.219</td>
<td>102.845</td>
</tr>
</tbody>
</table>

Table 5.1.: Registration delays
of connections are shown in Fig. 5.9

5.1.3.1. VoIP Registration

Due to the IMS core structure, the average time elapsed for IMS-Net registration is relatively long compared to T-Net (see Fig. 5.6 and Tab. 5.1). But for T-Net, it is commended that registration should be refreshed at an interval of 10 to 30 minutes, while in IMS the suggested refresh time may increase to one week. With such a long interval, the IMS registration traffic in an IMS Network with normal user behavior will drop compared to T-Net [VW06].

![Comparison of registration delays](image)

**Figure 5.6.** Comparison of registration delays

<table>
<thead>
<tr>
<th>Session setup delay (ms)</th>
<th>T-Net</th>
<th>IMS-Net</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean</td>
<td>3.587</td>
<td>2.316</td>
</tr>
<tr>
<td>Minimum</td>
<td>1.424</td>
<td>1.642</td>
</tr>
<tr>
<td>Maximum</td>
<td>9.356</td>
<td>9.570</td>
</tr>
</tbody>
</table>

**Table 5.2.** Session setup delays
5.1.3.2. VoIP Session

The results in Tab. 5.2 and Fig. 5.7 show that IMS-Net needs less call session setup time (with an average of 2.316ms) than T-Net (with an average for 3.587ms). Especially, the frequency distribution of session setup shows that IMS-Net (see lower part of Fig. 5.8) has a better session setup delay than T-Net (see upper part of Fig. 5.8).

In IMS-Net session setup is mostly (230 of 300 measurements) done in 2.8ms while for T-Net there are only 20 of 300 measurements done in this time interval. Because authentication information is stored persistently in IMS core, the client does not need to be re-authenticated before initiating a session, which is imperatively demanded by T-Net. This accelerates the call setup process of IMS-Net.

In order to study session setup delays when the servers are loaded with many clients and connections, tests with 500 clients and various number of connections per second was done: Session setup delays are evaluated for the server with different load situations. The server is loaded with 1, 10, 50, 100, 200 - call(s)/second respectively, and the corresponding session setup delay for every load situation is observed. Each setup delay is measured 50 times. Note the measurements are done without exceeding the capacity of the servers, i.e. the server overload situation is

![Figure 5.7: Comparison of session setup delays](image-url)
5.1. Internal Package Transmission Delay for Registration and Session Setup

**Figure 5.8.** Frequency distribution of session setup delays between two clients for T-VoIP (top) and IMS-VoIP (bottom)

**Figure 5.9.** Frequency distribution of session setup delays in T-Net (left) and IMS-Net (right) with many clients
not considered here. The results of T-Net and IMS-Net are shown in Fig. 5.9. The results show that with more clients and connections, session setup time of IMS-Net stays almost the same compared to the network platform loaded with only one connection with 2 clients. Figure 5.9 also shows that IMS-Net has a shorter call setup delay with a lower variance than T-Net.

5.2. Establish Application Servers in IMS

Time delay for registration and session setup of T-Net and IMS-Net is compared in Sec. 5.1. In this section, the comparison of the time delay of the data transmission after Session setup in the IMS Network and in the Traditional Network is analyzed. Additionally, the implementation of an application server, the Classification Application Server (CAS) in the IMS, is done. In chapter 4, an architecture for integrating a CAS for learning geographic regions in Next Generation Networks (NGN) was described (see Fig. 4.4) and the application of classification methods was accomplished. A CAS is established to make use of the location information of the IMSLS for classifying different traffic situations based on the states of the users. This section shows the architecture of this concept and presents how to integrate CAS in IMS Network as well as the transmission time in IMS-Net. CAS is located in the IMS Network, so the classification of the geographic region is done in IMS. This accelerates the process of learning geographic region. CAS is shared between the local network and the IMS global network. Therefore, network centralized up-to-date driver support and active safety services can be provided. The results show that IMS provides faster and more efficient services than Traditional Network.

5.2.1. Communication Establishment and Possibilities

5.2.1.1. Communication Establishment

As shown in chapter 4, the CAS can be used for two different filter functions (Type Filter and Location Filter). So if CAS contacts other servers in IMS-Net, the required tasks of Learning Geographic Regions can be achieved. For example:

1) Determining LF by TF (TF-CAS-LF): After receiving a SIP request for a group session, the CAS contacts the GLMS to retrieve the IMS member list. Then, the CAS retrieves the location information from the IMSLS for each
5.2. Establish Application Servers in IMS

2) Determining TF by LF (LF-CAS-TF): The CAS together with IMSLS defines the location of the user-regions. After contacting GLMS, CAS can adjust the types in the user-regions.

In Fig. 5.10 user1 utilize GPS method (self-positioning method) and has an UE with GPRS class 10. He wants to get the position information of user2. User2 has an UE with LTE and uses networking positioning method (remote-positioning) [DMS98].

User1 sends out a communication invitation over CSCF to CAS by SIP. After receiving a SIP request for a group session, the CAS contacts GLMS to get the group member list and then the server checks the status of each member of this group to see whether the invited users are available in this group. Afterwards

---

**Figure 5.10.** Communication establishment diagram

member of a group. After obtaining the positions of each member of every user group, which form the training data for classification, the geographic regions can be learned.
CAS server gets the location information from IMSLS for available users and determines the geographic regions of different groups. This realizes a service oriented architecture. All the services are separated on the IMS platform, but they can communicate to each other and share the information at any time. The geographic region information in user2’s local network is detected by CAS and will be sent to user1 through SIP. User1’s invitation will be sent to user2. Although user1’s equipment has GPRS and user2 uses LTE equipment, they can communicate with each other directly because of IP-based multimedia services from IMS platform [PNKM06]. From now on, user1 and user2 can communicate to each other and exchange their geographic region information through the IMS.

This kind of concept is useful for long scaled traffic controlling and traffic data exchange. Assuming that the cars on the road do not want to exchange information or communicate to each other (client-to-client communication), the last step ‘Invite_U2/Send U2,URI’ in Fig. 5.10 can be saved. IMS traffic control center wants to keep the traffic smooth and safe, so all the local traffic information is demanded to detect the traffic situation in all the regions. The local information of the users is sent automatically to IMS server (client-to-server communication). Depending on different traffic scenarios in different regions, IMS traffic control center sends different warning services to inform the drivers on the roads (server-to-client communication). In these cases, a client-to-server/server-to-client instead of client-to-client communication is needed. The architecture of the proposed concept offers this kind of communication.

5.2.1.2. Communication Possibilities

- Client-to-Client (C-C) Communication: Client-to-Client communication is realized by message exchange, see Fig. 5.11. The MESSAGE request is a mechanism for delivering short text messages (similar to the ones popular with today’s mobile phones) between two users. This mechanism is analyzed as an example for peer-to-peer data exchange. The sender’s user equipment generates a SIP request of the MESSAGE type and fills in the text message in the body and the recipient’s URI in the header. The request is then routed through the SIP infrastructure and, upon successful reception, acknowledged by the receiving UE with a 200 OK status message.

- Client-to-Server (C-S) and Server-to-Client (S-C) Communications: Client-to-Server and Server-to-Client communications are session based communi-
5.2. Establish Application Servers in IMS

![Diagram of Client-to-Client (message based) communication]

**Figure 5.11.**: Client-to-Client (message based) communication

![Diagram of Client-to-Server and Server-to-Client (session based) communications]

**Figure 5.12.**: Client-to-Server and Server-to-Client (session based) communications

This kind of communication is provided by an Application Server (AS) [PNK06] (CAS in this thesis), i.e. the information required is available in CAS and must not be obtained from the other client. These two cases are introduced together in Fig. 5.12. C-S and S-C communications are realized in three steps:

1. **Subscription**: A user subscribes to the region state of another user by sending a *SUBSCRIBE* request to the CAS. The CAS will then inform the subscribed user if the state of the region from the desired user
changes. Once the subscription is done, the network will remember the subscribed users (session set up). This step only needs to be done once and is called client and server pairing.

2. Publishing of region states: A user informs the CAS about his local region state via a *PUBLISH* request. With this step, the information of the user’s local region will be pushed to the CAS. All the information in the local regions of the users will be gathered in the CAS. This step is a C-S communication.

3. Notification: Once the local region state of a user changes, the CAS informs all users subscribed to the region state of this user via a *NOTIFY* request. This process is done automatically by CAS without the response from the users. This step is S-C communication.

All the above requests are acknowledged by the receiving party with a *200 OK* response in case of success.

### 5.2.2. Experimental Results

Time measurements of the packet flow in T-Net and IMS-Net are done. Depending on different communication possibilities, three cases for the data transmission are considered, which are client-to-client (C-C), client-to-server (C-S) and server-to-client (S-C). According to these cases, the latency time of the packet flow is compared between T-Net and IMS-Net. 100 measurements for each situation are done in the tests. The histograms and the mean-, maximum- and minimum values of the measurement results are shown.

<table>
<thead>
<tr>
<th>C-C data exchange delay (ms)</th>
<th>T-Net</th>
<th>IMS-Net</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean</td>
<td>0.724</td>
<td>0.561</td>
</tr>
<tr>
<td>Minimum</td>
<td>0.375</td>
<td>0.388</td>
</tr>
<tr>
<td>Maximum</td>
<td>32.127</td>
<td>2.448</td>
</tr>
</tbody>
</table>

*Table 5.3:* Comparison of time delay of packet flow for C-C in the T-Net and IMS-Net
5.2. Establish Application Servers in IMS

5.2.2.1. Time Delay of Data Exchange between Clients (C-C) in T-Net and IMS-Net

In the case of C-C, Tab. 5.3 and Fig. 5.13 show that the average delay for transporting packets between the clients in T-Net (mean value with 0.724 ms) is higher than in IMS-Net (mean value with 0.561 ms). Figure 5.14 shows the frequency distribution (histogram) of T-Net and IMS-Net. The first measurement in T-Net has a peak because of the first session setup. Both networks show good results: 95% of the measurements are done in 0.6 ms in both networks.

5.2.2.2. Time Delay of Data Exchange from Clients to Servers (C-S) in T-Net and IMS-Net

In the case of C-S, Tab. 5.4 and Fig. 5.15 show that the average time delay for packet flow in T-Net (mean value with 1.526 ms) is higher than in IMS-Net (mean value with 0.407 ms). Figure 5.15 also shows that T-Net has more peaks than IMS-Net, which means that T-Net has higher fluctuation than IMS-Net. Figure 5.16 shows that the frequency distribution of IMS-Net is better than that of T-Net: 99% of the measurements are done in 0.6 ms in IMS-Net, while only 21% of the measurements are done in 0.6 ms in T-Net.

![Figure 5.13: Time delay of data exchange for C-C in T-Net and IMS-Net](image-url)
Figure 5.14.: Frequency distribution of C-C delays in T-Net (top) and in IMS-Net (bottom)

<table>
<thead>
<tr>
<th>C-S data exchange delay (ms)</th>
<th>T-Net</th>
<th>IMS-Net</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean</td>
<td>1.526</td>
<td>0.407</td>
</tr>
<tr>
<td>Minimum</td>
<td>0.306</td>
<td>0.370</td>
</tr>
<tr>
<td>Maximum</td>
<td>33.768</td>
<td>0.602</td>
</tr>
</tbody>
</table>

Table 5.4.: Comparison of time delay of packet flow for C-S in the T-Net and IMS-Net
5.2. Establish Application Servers in IMS

5.2.2.3. Time Delay of Data Exchange from Servers to Clients (S-C) in T-Net and IMS-Net

In the case of S-C, Tab. 5.5 and Fig. 5.17 show that the average time delay for packet flow in T-Net (mean value with 0.264 ms) is lower than that in IMS-Net (mean value with 0.350 ms). But Fig. 5.17 also shows that T-Net has more peaks than IMS-Net, which means that T-Net has higher fluctuation than IMS-Net. Figure 5.18 shows again that the frequency distribution of IMS-Net is better than that of T-Net: 98% of the measurements are done in 0.6 ms in IMS-Net, while 78% of the measurements are done in 0.6 ms in T-Net.

The comparisons of the time delay of data exchange for C-C, C-S and S-C between T-Net and IMS-Net show that IMS-Net offers better session setup than T-Net.

5.2.2.4. Comparing the Packet Flow for all Cases in IMS-Net

Putting Tab. 5.3, 5.4 and 5.5 together leads to Tab. 5.6 which shows the time delay of data exchange between C-C, C-S and S-C in IMS-Net. Taking the latency time in C-C as reference, the results show that C-S needs only 73% and S-C communication needs only 62% average data exchange time than C-C communication. It means that C-S and S-C offer better performances than C-C.
Figure 5.16.: Frequency distribution of C-S delays in T-Net (top) and in IMS-Net (bottom)

<table>
<thead>
<tr>
<th>S-C data exchange delay (ms)</th>
<th>T-Net</th>
<th>IMS-Net</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean</td>
<td>0.264</td>
<td>0.350</td>
</tr>
<tr>
<td>Minimum</td>
<td>0.090</td>
<td>0.303</td>
</tr>
<tr>
<td>Maximum</td>
<td>1.151</td>
<td>0.562</td>
</tr>
</tbody>
</table>

Table 5.5.: Comparison of time delay of packet flow for S-C in the T-Net and IMS-Net
5.3. Conclusion

This chapter compares registration and session setup delay in the Traditional Network server (T-Net) and in the IMS Network server (IMS-Net), where VoIP is taken as an example. IMS-Net requires more registration time than T-Net, but with the long de-registration interval (up to one week instead of 10-30 minutes for T-Net) IMS registration traffic in the network drops. IMS Network uses the 3GPP Authentication and Key Agreement (3GPP-AKA) mechanism. This avoids a re-authentication between the clients before each new call begins, therefore session setup time for IMS-Net is shorter than for T-Net.

Furthermore, the architecture for integrating Classification Application Server (CAS) in IMS-Net is also presented. CAS is established in IMS-Net, so learning the geographic regions is done directly in the network. This architecture realizes centralized network controlling applications and enables a low latency data exchange on the IMS platform and between the network and the users. The time
delay of message flows in the T-Net and IMS-Net are also compared. IMS-Net offers a shorter data exchange delay than T-Net.

Due to today’s numerous applications in the network and the resulting complexity, the load of the server is an important issue. The network provider needs a network server which can offer a high date rate and low latency communication processing for multimedia services. Compared to the current system, IMS can offer better data transmission services to real-time applications which benefits not only the IMS clients but also the IMS Network provider.

Figure 5.18: Frequency distribution of S-C delays in T-Net (top) and IMS-Net (bottom)
6. Conclusion and Outlook

This thesis focuses on the use of advanced positioning methods and Location Based Services (LBS) in the IP Multimedia Subsystem (IMS). IMS, specified by the 3rd Generation Partnership Project (3GPP) as part of Next Generation Networks (NGN), is used to reduce operational costs and provides converged services to its customers. First GPS/DGPS positioning methods with scalable positioning accuracies are presented. Using the position information, classification methods are applied to learn geographic regions based on the location information and users’ states gathered in the IMS database. Furthermore, the implementation of a Classification Application Server (CAS) in the IMS Network is discussed, where the time delays of the servers in the IMS Network are also considered. The presented positioning methods are based on the trade-off between positioning accuracy and computational complexity. They can be implemented on an unified platform and provide the required positioning accuracies for different Location Based Services (LBS). QR Decomposition (QRD) using COordinate Rotation DIgital Computer (CORDIC) based approximate rotations is applied to the positioning algorithms for GPS (linear LS problem solved in each iteration) and DGPS (one linear LS problem) as well as recursive DGPS. To suit the different use cases of LBS, it is possible to choose the positioning accuracy by choosing the numbers of required iterations \(itr\) and the number of optimal CORDIC-angles \(itg\). The experiment results show that even for small \(itg\), positioning accuracies can be achieved, which are high enough for many LBS applications. Therefore, a significant reduction in computational complexity and power consumption can be obtained depending on the use cases. Furthermore, because this method only requires shift and add operations, it is very well suited for hardware implementation.

The presented method, which is applied to GPS/DGPS, can also be applied to any other positioning method based on Least Squares computations (e.g. \[Fen99\] \[SA87\] \[Fri87\]) as well as recursive positioning methods \[MW01\] \[Des74\]. The proposed positioning methods can be considered as one very important step towards an autonomous GPS receiver working continuously in the field using very restricted...
power resources. Here, the proposed methods will be very beneficial, because they offer a positioning method with scalable positioning accuracy (approximations) and optimized power consumption for the individual use-cases.

In future work, this idea can be applied to Kalman Filter based recursive GPS algorithms [GWA07] [AK07]. In the square-root version of the Kalman filter [MW01] QRD can be applied, and the required number of optimal CORDIC angles can be investigated to obtain a desired positioning accuracy.

Additionally, the FFTs required for GPS acquisition and tracking tasks [Bor06] can also be implemented using CORDIC processors [Des74] [ZC04]. Therefore, a CORDIC based GPS/DGPS baseband processor for all signal processing tasks (acquisition, tracking, positioning) as well as the measurement noise estimation [AH10] can be designed.

Once the position estimates are available, it is assumed that the position estimates are transferred to a database. Then, classification methods can be deployed to detect geographic regions using LBS in the IP Multimedia Subsystem (IMS). Exchanging the user information (data sets) between the cellular local network and the IMS Network through the IMS kernel sets up the database in the IMS Network. Establishing a Classification Application Server (CAS) in the IMS and combining it with the Group List Manage Server (GLMS) achieves Type Filter (TF), which divides all IMS clients into groups due to different variables in users’ data sets. CAS using discriminant analysis or local-nearest neighbor algorithm together with the IMS Location Server (IMLS) determines the position of the user group depending on the location of each member in every group. After that, the geographic regions are learned by the IMS and Location Filter (LF) can be defined. Depending on LF, different services (e.g. warning service) can be established for the users in varied positions. The role of TF and LF can also be exchanged. CAS together with IMLS defines the location of the regions. After contacting GLMS, CAS can determine the class in the region depending on the states of users (training data). After that, the behaviors of the geographic regions are learned and TF can be defined.

Here, we concentrated on the implementation of the proposed concept in the IMS application layer using only a simple classification example. In future work, real traffic data should be incorporated into the simulations. Also other classification methods can be applied [WSH11] [SLLK12]. One important example which will be investigated in the future is learning of the geographic region of Cell IDs [Abe10], which can be used for handover resource reservation [RKK07]. Because
of the highly irregular shape of these geographic regions, sophisticated classification methods are required and a uniform distribution of the users with the same Cell ID is probably reasonable. Overall, in future mobile communication systems, where location based services gain more and more importance for many applications, classification for geographic regions of events and the deduced services will play an important role.

The comparison of SIP message flow in the Traditional Network server and in the IMS Network server is also done in this thesis. VoIP is used as an example, but the results can also be applied to other services, e.g. LBS. The time delay of registration and session setup for both the Traditional Network and IMS Network are analyzed. Concerning the IMS core, the IMS Network requires more registration time than the Traditional Network, but with the long de-registration interval (up to one week instead of 10-30 minutes for Traditional Network) IMS registration traffic in the network drops. Furthermore, in the IMS the user registration procedure performs mutual authentication between the user equipment and the IMS Network using the 3GPP Authentication and Key Agreement (3GPP-AKA) mechanism. This avoids a re-authentication before each new call begins between the clients, which decreases the call setup time for the IMS Network compared with the Traditional Network. This is important, when the location information is exchanged frequently but not continuously. Moreover, with an increasing number of connections, session setup time for the IMS Network remains more stable than for the Traditional Network.

Furthermore, the architecture for integrating CAS in the IMS Network is presented. Exchanging the location information between local network (cellular network) and global IP network (IMS) through the IMS kernel offers the data resource for CAS. CAS is established in the IMS Network, so learning the geographic regions is done directly in the network. This enables a low latency data exchange on the IMS platform and between the network and the users. The time delay of message flows in the Traditional Network and IMS Network are also compared. The result show again that the IMS Network offers a shorter and more stable data exchange delay with a lower variance than the Traditional Network. Using the CAS for learning geographic regions in the vehicular environment is only one possible application of CAS in the IMS. CAS could also be applied for other classification tasks, e.g. classifying multimedia data in order to optimize multimedia services in the IMS.
A. Basic Linear Algebra

A.1. Vectors and Matrices

The inner product of two column vectors \( \mathbf{v} \) and \( \mathbf{w} \) with \( n \) components is the number

\[
\mathbf{v}^T \cdot \mathbf{w} = \sum_{i=1}^{n} v_i w_i. \tag{A.1}
\]

The inner product \( \mathbf{w}^T \cdot \mathbf{v} \) equals \( \mathbf{v}^T \cdot \mathbf{w} \). The length (norm) of a vector \( \mathbf{v} \) is the square root of \( \mathbf{v}^T \cdot \mathbf{v} \):

\[
\|\mathbf{v}\|_2 = \sqrt{\mathbf{v}^T \cdot \mathbf{v}} = \sqrt{\sum_{i=1}^{n} v_i^2}. \tag{A.2}
\]

The length is always positive, except for the zero vector with \( \|\mathbf{0}\|_2 = 0 \). If \( \mathbf{v} \) and \( \mathbf{w} \) are nonzero vectors, then for the angle \( \theta \) between vector \( \mathbf{v} \) and vector \( \mathbf{w} \) holds

\[
\cos(\theta) = \frac{\mathbf{v}^T \cdot \mathbf{w}}{\|\mathbf{v}\|_2 \|\mathbf{w}\|_2}. \tag{A.3}
\]

and therefore,

\[
|\mathbf{v}^T \cdot \mathbf{w}| \leq \|\mathbf{v}\|_2 \|\mathbf{w}\|_2. \tag{A.4}
\]

The inner product \( \mathbf{v}^T \cdot \mathbf{w} = 0 \) when \( \mathbf{v} \) is orthogonal to \( \mathbf{w} \), i.e. \( \theta = 90^\circ \) [Str94].

A.2. Orthogonal Bases

When the inner products of the vectors \( \mathbf{q}_1, \ldots, \mathbf{q}_n \) are zero, they are orthogonal, i.e. \( \mathbf{q}_i^T \mathbf{q}_j = 0 \), whenever \( i \neq j \). Dividing each vector by its length, the vectors
become orthogonal unit vectors. Then the basis is called orthonormal \[\text{Str94}\].

The vector \(q_1, \ldots, q_n\) are orthonormal, if

\[
q_i^T q_j = \begin{cases} 
0 & \text{when } i \neq j (\text{orthogonal vectors}) \\
1 & \text{when } i = j (\text{unit vectors}: \|q_i\|_2 = 1)
\end{cases}
\]  

(A.5)

A matrix with orthonormal columns is denoted as \(Q\) matrix, i.e. \(Q^TQ = I\). When the matrix is square, \(Q^TQ = I\) means that \(Q^T = Q^{-1}\).

A key property of \(Q\) is that, it leaves vector lengths unchanged:

\[
\|Qx\| = \|x\| \quad \text{for every vector } x.
\]

(A.6)

\(Q\) also preserves inner products and angles:

\[
(Qx)^T (Qy) = x^T Q^T Q y = x^T y.
\]

(A.7)

### A.3. The Least Squares Problem

The LS problem solves the problem of finding a vector \(x \in \mathbb{R}^n\) such that \(Ax = b\), where the data matrix \(A \in \mathbb{R}^{m \times n}\) and the observation vector \(b \in \mathbb{R}^m\) are given and \(m \geq n\). When there are more equations than unknowns \((m > n)\), the system \(Ax = b\) is overdetermined. Since \(b\) is generally not in the column space of \(A\), an overdetermined system has no exact solution. Therefore, we have to minimize \(\|Ax - b\|_p\) for some suitable choice of \(p\).

The function \(f(x) = \|Ax - b\|_p\) is not differentiable in the 1-norm and \(\infty\)-norm, so minimization is complicated. For \(p = 2\), we obtain the least squares (LS) problem \[\text{GL96}\]

\[
\min_{x \in \mathbb{R}^n} \|Ax - b\|_2,
\]

(A.8)

which is more tractable, because

- \(\theta(x) = \frac{1}{2} \|Ax - b\|_2^2\) is a differentiable function of \(x\). The minimizers of \(\theta\) satisfy the gradient equation \(\nabla \theta(x) = 0\).

- The 2-norm is protected under orthogonal transformation. This means that for an orthogonal \(Q\) the equivalent problem of minimizing \(\|(Q^T A)x - (Q^T b)\|_2\) can be solved.
A.4. Solving the Least Squares Problem

A.4.1. Normal Equations

Given \( A \in \mathbb{R}^{m \times n} \) with the property that \( \text{rank}(A) = n \) and \( b \in \mathbb{R}^{m} \), this algorithm computes the solution \( x_\ell \) to the LS problem

\[
\min_x \|Ax - b\|_2.
\]  

(A.9)

First, compute \( C = A^T A \) and \( z = A^T b \), then compute the Cholesky factorization \( C = GG^T \). At last solve \( Gy = z \) and \( G^T x_\ell = y \) by back substitution \([GL96]\).

The normal equation approach relies on standard algorithms: Cholesky factorization, matrix-matrix multiplication and matrix-vector multiplication. It is attractive to compress the \( m \times n \) data matrix \( A \) into the much smaller \( n \times n \) matrix \( C \). However, the accuracy of the computed normal equations solution depends on the square of the condition of \( A \).

A.4.2. QR Decomposition

The full rank LS problem can also be solved by QR Decomposition \([GL96]\). Suppose that an orthogonal matrix \( Q \in \mathbb{R}^{m \times m} \) has been computed such that

\[
Q^T A = R = \begin{bmatrix} R_1 \\ 0 \end{bmatrix} \quad n \quad m - n
\]  

(A.10)

is upper triangular. If

\[
Q^T b = \begin{bmatrix} c \\ z \end{bmatrix} \quad n \quad m - n
\]  

(A.11)

then

\[
\|Ax - b\|_2^2 = \|(Q^T A)x - (Q^T b)\|_2^2 = \|R_1 x - c\|_2^2 + \|z\|_2^2
\]  

(A.12)

for any \( x \in \mathbb{R}^n \). If \( \text{rank}(A) = \text{rank}(R_1) = n \), \( x_\ell \) is defined by the upper triangular system \( R_1 x_\ell = c \) which can be solved by back substitution. Note that the least square error is

\[
e_\ell = \|z\|_2.
\]  

(A.13)
Bibliography


List of Figures

1.1. Global structure of positioning and LBS in IMS ........................................... 8

2.1. Schematic view of GPS with observed pseudoranges $P_k$ (solid lines) and geometrical ranges $\rho_k$ (dashed lines) ................................................................. 18

2.2. Definition of observed pseudorange $P_k$ (solid lines) and geometrical range $\rho_k$ (dashed lines) ......................................................................................... 19

2.3. Differential GPS positioning ........................................................................... 22

2.4. Geometry for DGPS ......................................................................................... 23

2.5. OSI and IMS layered Architecture [IMS10] .................................................... 33

2.6. IMS Network Architecture Overview ............................................................. 34

3.1. Approximate GPS/DGPS methods achieve scalable positioning accuracies and low power consumption by choosing the number of iterations ($itr$) and the number of optimal CORDIC angles per rotation ($itg$) ......................................................................................... 40

3.2. GPS/DGPS positioning algorithm with varying numbers of required iterations ($itr$) and varying approximation accuracies ($itg$) ........................................................................... 41

3.3. Approximate QRD: CORDIC-based approximate Givens rotation ................. 44

3.4. GPS positioning results (in meter) using Rinex data for $itr = 4$ and $itg = 2$ .................................................................................................................. 53

3.5. GPS positioning results (in meter) using Rinex data for $itr = 4$ and $itg = 6$ .................................................................................................................. 54

3.6. GPS positioning results (in meter) using raw GPS data for $itr = 4$ and $itg = 2$ .................................................................................................................. 55

3.7. GPS positioning results (in meter) using raw GPS data for $itr = 4$ and $itg = 6$ .................................................................................................................. 56

4.1. Vehicle-to-IMS architecture ............................................................................. 62

4.2. Learning geographic regions by applying filters and classification to the database in IMS ................................................................................................. 64
<table>
<thead>
<tr>
<th>Figure</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.3</td>
<td>Using Filters for warning service</td>
</tr>
<tr>
<td>4.4</td>
<td>Architecture of applying classification with LBS in IMS to learn</td>
</tr>
<tr>
<td>4.5</td>
<td>Data sets of one user in IMS</td>
</tr>
<tr>
<td>4.6</td>
<td>Learned geographic regions using discriminant analysis and local-</td>
</tr>
<tr>
<td>4.7</td>
<td>learned geographic regions using discriminant analysis and local-</td>
</tr>
<tr>
<td>4.8</td>
<td>learned geographic regions using discriminant analysis and local-</td>
</tr>
<tr>
<td>4.9</td>
<td>learned geographic regions using discriminant analysis and local-</td>
</tr>
<tr>
<td>4.10</td>
<td>Learned geographic regions using discriminant analysis and local-</td>
</tr>
<tr>
<td>4.11</td>
<td>Learned geographic regions using discriminant analysis and local-</td>
</tr>
<tr>
<td>5.1</td>
<td>Traditional Network and IMS Network</td>
</tr>
<tr>
<td>5.2</td>
<td>T-Net registration</td>
</tr>
<tr>
<td>5.3</td>
<td>T-Net session setup</td>
</tr>
<tr>
<td>5.4</td>
<td>IMS registration</td>
</tr>
<tr>
<td>5.5</td>
<td>IMS session setup</td>
</tr>
<tr>
<td>5.6</td>
<td>Comparison of registration delays</td>
</tr>
<tr>
<td>5.7</td>
<td>Comparison of session setup delays</td>
</tr>
<tr>
<td>5.8</td>
<td>Frequency distribution of session setup delays between two clients</td>
</tr>
<tr>
<td>5.9</td>
<td>Frequency distribution of session setup delays in T-Net (left) and</td>
</tr>
<tr>
<td></td>
<td>IMS-Net (right) with many clients</td>
</tr>
<tr>
<td>5.10</td>
<td>Communication establishment diagram</td>
</tr>
<tr>
<td>5.11</td>
<td>Client-to-Client (message based) communication</td>
</tr>
<tr>
<td>5.12</td>
<td>Client-to-Server and Server-to-Client (session based) communications</td>
</tr>
<tr>
<td>5.13</td>
<td>Time delay of data exchange for C-C in T-Net and IMS-Net</td>
</tr>
<tr>
<td>5.14</td>
<td>Frequency distribution of C-C delays in T-Net and in IMS-Net</td>
</tr>
</tbody>
</table>
5.15. Time delay of data exchange for C-S in T-Net and IMS-Net . . . 105
5.16. Frequency distribution of C-S delays in T-Net and in IMS-Net . . 106
5.17. Time delay of data exchange for S-C in T-Net and IMS-Net . . . 107
5.18. Frequency distribution of S-C delays in T-Net and IMS-Net . . . 108
List of Tables

1.1. The performance of different positioning methods \[\text{JSL03}\] . . . . . . 10

3.1. Shift value $\ell$ and the corresponding CORDIC angle $\phi(\ell)$ . . . . . . 43
3.2. MPE of positioning results (in meter) by GPS using Rinex data . . 51
3.3. MPE of positioning results (in meter) by GPS using raw GPS data . 51
3.4. SDPE of positioning results (in meter) by GPS using Rinex data . . 52
3.5. SDPE of positioning results (in meter) by GPS using raw GPS data . 52
3.6. MPE and SDPE of positioning results (in meter) by DGPS and re-
    cursive DGPS (for 0.5km distance between 2 receivers) using Rinex
    data . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 57
3.7. MPE and SDPE of positioning results (in meter) by DGPS and
    recursive DGPS (for 512m distance between 2 receivers) using raw
    GPS data . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 58

4.1. Data sets in IMS database $S$ . . . . . . . . . . . . . . . . . . . . . . . . . . . . 65
4.2. Comparison of the characteristics of the discriminant analysis and
    the local-nearest neighbor algorithm . . . . . . . . . . . . . . . . . . . . . . . 75
4.3. The confusion matrix ($P = 1$ run) in percentage with 0% errors . . 78
4.4. Mean of the confusion matrices ($P = 10$ runs) in percentage with
    8% errors . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 78
4.5. Mean of the confusion matrices ($P = 10$ runs) in percentage with
    20% errors . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 78
4.6. Mean of the confusion matrices ($P = 10$ runs) in percentage with
    50% errors . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 79
4.7. Variance of the confusion matrix ($P = 10$ runs) in percentage with
    8% errors . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 79
4.8. Variance of the confusions matrices ($P = 10$ runs) in percentage
    with 20% errors . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 79
<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.9.</td>
<td>Variance of the confusion matrices ($P = 10$ runs) in percentage with 50%</td>
<td>80</td>
</tr>
<tr>
<td></td>
<td>errors</td>
<td></td>
</tr>
<tr>
<td>4.10.</td>
<td>Confusions matrix in percentage for an instantaneous scenario at time $t$</td>
<td>80</td>
</tr>
<tr>
<td>4.11.</td>
<td>Confusions matrix in percentage for an instantaneous scenario at time $t + \Delta t$</td>
<td>80</td>
</tr>
<tr>
<td>5.1.</td>
<td>Registration delays</td>
<td>94</td>
</tr>
<tr>
<td>5.2.</td>
<td>Session setup delays</td>
<td>95</td>
</tr>
<tr>
<td>5.3.</td>
<td>Comparison of time delay of packet flow for C-C in the T-Net and IMS-Net</td>
<td>102</td>
</tr>
<tr>
<td>5.4.</td>
<td>Comparison of time delay of packet flow for C-S in the T-Net and IMS-Net</td>
<td>104</td>
</tr>
<tr>
<td>5.5.</td>
<td>Comparison of time delay of packet flow for S-C in the T-Net and IMS-Net</td>
<td>106</td>
</tr>
<tr>
<td>5.6.</td>
<td>Data exchange latency between C-C, C-S and S-C in IMS-Net</td>
<td>107</td>
</tr>
</tbody>
</table>
Zusammenfassung


Unter der Annahme, dass die Informationen des lokalen Netzwerks (zelluläres Netz) mit dem globalen IP Netzwerk (IMS) ausgetauscht werden können, werden die geschätzten Positionen sowie andere Zustände der Nutzer in einer Datenbank
gesammelt. Unter Verwendung dieser Datenbank können statistische Klassifikationsverfahren angewandt werden. Hierbei werden zwei Fälle unterschieden:


Die vorgestellten Ergebnisse ermöglichen es in zukünftigen Netzen ortsbezogene Dienste mit skalierbarer Genauigkeit und daher skalierbarem Energieverbrauch kosteneffizient für den jeweiligen Anwendungsfall zur Verfügung zu stellen.
Publications

Peer-Reviewed Journal Publications


Peer-Reviewed Conference Publications


- Yuheng He, Johannes Veerkamp, Attila Bilgic. Analyzing the Internal Processing of IMS-based and traditional VoIP systems, *Proc. World Telecommu-


Curriculum Vitae

Name: Yuheng He
Date of birth: October 6th, 1976
Place of birth: Shanghai, P.R.China

Academic History

10/2002–10/2007 Diplom engineer (Dipl.-Ing.)
Electrical Engineering and Information Technology
Technical University Dortmund
Thesis topic: Direction-of-Arrival and Frequency Estimation
in Communication Systems (Nokia Diploma Thesis Award)

Professional Experience

06/2006–10/2007 Research internship, final year project of diploma work and
student trainee for ‘3GPP’ project
Nokia Research Center, Bochum, Germany

04/2007–09/2007 Student assistant
Electrical Engineering and Information Technology
Technical University Dortmund

01/2008–10/2012 Research engineer
Institute of Integrated Systems
Ruhr-Universität Bochum, Germany

02/2010–10/2012 Research engineer
Institute of Communication Acoustics
Ruhr-Universität Bochum, Germany

Bochum, January 28, 2013
Acknowledgments

This thesis has been written at the Institute of Integrated Systems (IS) and the Institute of Communication Acoustics (IKA) at the Ruhr-Universität Bochum, Germany. The thesis would not have been successful without the excellent supervision and the continuous support of my doctoral advisers. I would like to thank Prof. Dr.-Ing. Attila Bilgic for all the continuous supports and advises. He gave me a fascinating topic, a lot of ideas and discussions as well as many practical helps during my doctor study time. I would also like to thank Prof. Dr.-Ing Rainer Martin for all the enlightening discussions, valuable suggestions as well as the open friendly work environment in IKA. My particular thanks go to Prof. Dr.-Ing. Josef Hausner, Prof. Dr.-Ing. Thomas Musch, Prof. Dr.-Ing Edmund Coesmeier, Prof. Dr.-Ing Juergen Goetze, Prof. Dr.-Ing Steven Ding and Prof. Dr. Adel S. Elmaghraby for their interest in my work. They were always good discussion partners and encouraged me with valuable suggestions. Additionally, I would like to thank Irv Badr from IBM. He gave me valuable suggestions during the eMuCo project. Furthermore, I would like to thank Dr.-Ing. Pierre Mayr, Prof. Dr.-Ing Nils Pohl and Dipl.-Ing. Anil Nagathil for sharing plenty of good ideas and supports. I appreciate the very pleasant work environment between all the members of IS and IKA. Finally, I want to thank my family and my friends for their consistent encouragement and supports.