ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Distributed Computing Group

Semester Thesis:

Semantic Understanding of Location and Movement Information collected by a Mobile Application

Author: Manuel Baumann
manubaum@ee.ethz.ch

Advisor: Michael Kuhn
Fabio Magagna

Tutor: Prof. Dr. Roger Wattenhofer

Date: January 14, 2009

Abstract

The past few years mobile devices emerged to a very important and ubiquitous communication medium. They are able to handle email, multimedia contents, websites, etc. The same time the idea of the *Web 2.0* came up. The spirit behind the *Web 2.0* is that everybody is invited to share his/her knowledge[1] with the rest of the world. One logical consequence is the melt down of these two technologies. Social platforms on mobile devices will be the killer application of mobile devices. We aim to build a framework for the location and movement awareness of these devices. This adds further value to our mobile devices, because we could gain lots of information about a user's situation. This thesis focuses on processing real-time logged data consisting of WLAN occurrences, GSM cells and GPS data for finding out how a user moves and where he/she is located. All the computations are done on a server. We compute five characteristical metrics which then are used for movement estimation by three different approaches. We try to classify the data in 4 different kinds of movement: standing, walking, moving by car and moving by train. To evaluate the performance of the algorithms, their classifications are tested against a hand kept journal. The best performing movement estimation algorithm shows a classification accuracy of 85 % and locations are recognized 87% of the time.

[1] In form of videos or media in general, social networking, wikis and so on.

Contents

1 Introduction

The success of mobile devices in the past few years and the shift of the internet towards a platform-like information medium changed our way of communication. Many people provide personal information about their daily life on social plattforms. It seems to be obvious that this kind of information exchange will be done more and more using mobile devices.

The future services and applications on the mobile phone will be on one hand similar as on the personal computer; but must be adapted to the special features of mobile phones to minimize the limitations and maximize the advantages of mobile phones. Some of the limitations of mobile phones are the relatively small screen and input interface; whereas some of the advantages of mobile phones are to provide the opportunity for users to be "always online" and "traceable". Note that the latter advantage may also create privacy concerns. Nevertheless, all of the advantages of mobile phone open new opportunities in the design of service applications via mobile phone. Among the many possible service applications, some experts argue that the location-based services (LBS) will become the "killer application" in the near future. Indeed, the market size for LBS has been growing in an exponential rate. The market size is estimated to be 447 Mio US$ in Asia, 622 Mio US$ in Europe and 1.3 Bio US$ in USA 2010. [9]

This semester thesis aims to extend a framework for location based services. These services can and will improve the way we socialize today. They add further value to our ubiquitous mobile devices, because socializing can be improved by the knowledge about our current location. Much of the daily social information for example where we are, where we are going and even who we are meeting can be retrieved automatically. A multitude of possible applications could be built on top of the framework.

The basis of our work is a project called Abakabar (Indonesian for "how are you?") and was founded by Fabio Magagna in a former semester thesis [1]. It does not aim to provide a kind of website for mobile socializing but only the core application which consists of a mobile software and a web service which is accessed through websites by API calls. This makes this framework interesting for other developers. From the former project everything but the software on the mobile phone was reimplemented for the sake of the programmers deeper understanding of the system.

The main topic of this thesis is movement recognition. We defined four classes of movement we want to distinguish: *standing*, *walking*, *moving by car* and *moving by train*. This is done by analyzing the occurrences of WLANs, GSM cells and, if available, GPS coordinates. To reach this, five metrics were introduced. The first two are measures for a user's speed gained by analyzing WLAN and GSM occurrences. These measures are justified by a simple mathematical model. The other three measures are retrieved from GPS coordinates. The most intuitive measure is the user's speed. The other two variables are the distances to railroads and streets. These five values, later called feature-vectors, are then processed by three different algorithms for movement estimation.

The thesis is structured as follows: First we describe the former project called Abakabar. Then follows a short overview of the basics of how the mobile software works and then we proceed to the later used definitions. After showing how the data is visualized, the location estimation part is explained. The main part deals with movement estimation. At the end of the thesis, we evaluate and compare the performance of the three algorithms.

2 Related Work

There are dozens of applications for mobile phones which try to improve sozializing. They use different approaches for doing that. Some of them rely only on the WLAN or Bluetooth devices they see to distinguish which user are close-by (e.g. Geode, iCloseBy,... [11]). Other act as improved interface to a social-network. They are able to capture the user's activity like sitting, walking, meeting friends, etc. (CenceMe [11]). One uses a built in accelerometer to distinguish between the actions sitting, walking and running.

We do not cover the analysis of patterns in a user's daily life. So there is no statement whether the user is on the way to work, or visiting a friend. Nevertheless, it is a very interesting topic and was analyzed in another paper. [4]

A mobile phone manufacturer looks at this topic form another aspect. They look at the mobile devices as the world's most distributed and pervasive sensing instrument. Also because there are more and more built-in sensors. [10]

3 Project Setup

3.1 Mobile Application

As this thesis is based on an other semester-thesis written by Fabio Magagna [1], this part shows the premises of my thesis. The system consists of a mobile application which logs occurrences of wireless access-points, the CGIs (Cell Global Identifier) from GSM antennas and, if available, coordinates measured by GPS in 10 second intervals. This data is then transferred to a server. The application has two modes of operation. The first one submits the new data every 10 seconds (online modus) using UMTS while the other mode stores all data and only transmits the data if requested by the user (offline mode). This mode is very suitable for long time logging and allows cheap uploading through WLAN. The application is written in C and runs on Symbian based mobile phones.

3.2 Application Overview

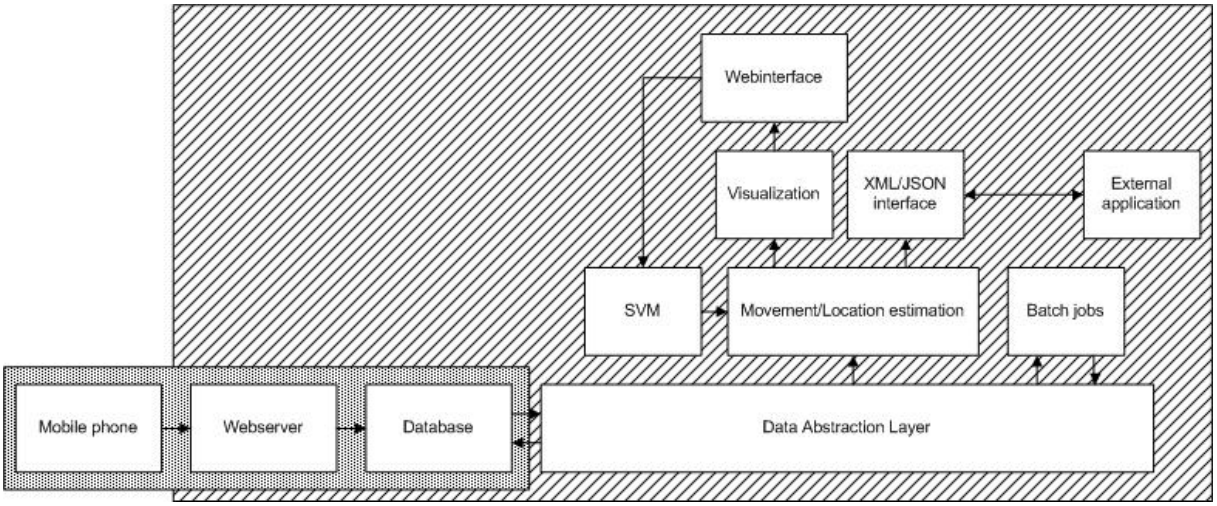

Figure 1: Project setup

Figure 1 shows an abstract overview of the system. The dotted area depicts the work of Fabio Magagna while the striped part shows the substance of this thesis. The arrows stand for a certain data flow. The software described in [1] on the *mobile phone* logs all occurrences of WLANs, GSM cells and GPS coordinates. This data is sent over a usual HTTP request to the *web server*, which preprocesses the data and stores it in a *database*. The whole software is written in Java. The basis of further data processing is the *data abstraction layer*. It provides abstract access to the data by representing it in form of objects and lists of objects. The *batch jobs* part consists of algorithms for data-mining. These algorithms update graphs for location approximation. As this data

is not mandatory for the live part of the system, this data structures are only updated daily. The *Movement/Location estimation* part is the core of the application. There are two interfaces attached to the core. The *Visualization* part provides visual inspection of the data gathered for a certain user and during a certain interval which can be displayed in any web browser. It is used to implement and verify the software. The *SVM* provided by libsvm [2] provides a second approach for movement estimation. The *XML/JSON interface* allows access to the data in realtime for external applications.

3.3 Log Format

The produced log files are in a differential format. This means, for example, only the change of a GSM-cell is reported. If there was no change at all, a ping is reported back to the server. This is needed to detect whether the user is still online.

```
Data: occurrences of WLAN, actual GSM-Cell, GPS if available
Result: differential log
Set WLAN, WLAN_prev;
Cell GSM, GSM_prev;
while true do
    WLAN = getWLANinRange();
    GSM = getGSMinRange();
    coordinates = getGPSCoordinates();
    if WLAN=WLAN_prev then
        addToLog(WLAN)
    if GSM!=GSM_prev then
        addToLog(GSM)
    if coordinates.speed ≥ 0.2 then
        addToLog(coordinates)
    if WLAN==WLAN_prev and GSM==GSM_prev and coordinates==null then
        addToLog("Ping")
    if mode==islive then
        sendLog()
    else
        storeLogToFile()
    WLAN_prev = WLAN;
    GSM_prev = GSM;
    wait_seconds(10);
```

Procedure `logger`

Each log entry consists of a timestamp, the userid (in this case "testID"), a logtype identifier (**P**ing, **W**LAN,**G**PS, **C**ell) and the log specific data for example GPS coordinates,

...
02/11/2008 22:14:31 testID C 110957 7500 228 03
02/11/2008 22:14:32 testID W 00:0f:b5:d4:7d:65 00:6a:11:0a:33:9c 00:14:6c:f5:8d:64
02/11/2008 22:14:42 testID P
02/11/2008 22:14:52 testID G +8°33'29".267 +47°22'36".206
...

Figure 2: Summary of a log file

MAC addresses of WLANs or CGI of cells. At a certain point in time, there could be several log entries.

4 Data Sources

In this part the three data sources of the data logger are discussed in a short manner. There is no rigorous discussion about the technology itself since these issues are covered in other literature.

4.1 Wireless Local Area Network

4.1.1 Identification

Each WLAN has a globally unique identifier called MAC address[2]. The usual notation is in form of 6 hex values in the range 0x00-0xff, spread by a colon. Fortunately, it does not matter whether a WLAN base station has some access restrictions. The WLAN is detected, as long as it broadcasts beacons. A WLAN occurrence is always in conjunction with the detection of a MAC address.

4.1.2 Notation

In order to describe a set of MAC addresses by their occurrences, the following notation is introduced. The time instance t_n denotes the start of an interval of $10s$. Therefore it holds $t_n - t_{n-1} = 10sec \quad \forall n$.

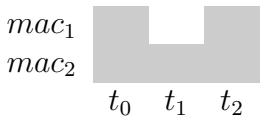

Figure 3: Notation

$$A = \begin{pmatrix} 1 & 0 & 1 \\ 1 & 1 & 1 \end{pmatrix}$$

Figure 3 shows two different MAC addresses on the ordinate. If a certain MAC address was seen at a time instant, the square is filled in gray. In our example the mac_1 was seen at the time instances t_0 and t_2 and mac_2 at the time instances t_0, t_1 and t_2. This notation was also used to implement the algorithms for motion detection. A occurrences matrix with the same design was introduced. Every element in the matrix A is either 0 or 1, where 0 denotes the absence and 1 the occurrence of a WLAN.

[2]There are devices which allow changing their MAC address.

4.1.3 Availability

With the emergence of broadband internet access, the private use of WLAN became very popular. Therefore many households have their own WLAN base station which replaces the router and switches. This makes it very suitable for detecting a user's position. The density of public WLAN access points is very dependent on the location. In a city there are many more public access points than in rural areas.

4.1.4 Precision

As the range of a WLAN access point heavily depends on the surroundings, it is not possible to make a general statement about it. But in comparison to GSM, WLAN is much more fine-meshed and therefore allows a more precise location estimation. Measurements in Figure 13.1 show that 80 % of the logged WLANs have a range of 60 - 240 meters. The range is the maximum distance between two GPS coordinates where the same WLAN was seen.

4.1.5 Remarks

A WLAN provides very precious information concerning the location detection of a user because the range is limited and can directly map the user to his/her location.

4.2 GSM

4.2.1 Identification

Every GSM antenna is identified by the CGI (Cell Global Identity). The CGI consists of four parts:

- MCC: Mobile Country Code (MCC = 228 for Switzerland)
- MNC: Mobile Network Code (identifies the network operator)
- LAC: Location Area Code
- CI: Cell identifier

More information about the CGI is found in the semester thesis by Fabio Magagna [1, p. 17].

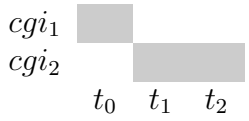

Figure 4: Notation

4.2.2 Notation

The notation of GSM occurrences is quite similar to the one of WLAN occurrences.

$$A = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \end{pmatrix}$$

As discussed later, the main difference is that during every time instance only one GSM cell or CGI is visible. Therefore the matrix $A_{n \times m}$ has to hold the condition:

$$\sum_{i=1}^{n} A_{i,m} \leq 1 \qquad \forall m$$

4.2.3 Availability

Nowadays, GSM is available almost everywhere. There are only some situations, for example when travelling in a train passing through a long tunnel, where the connection could be lost. But in general, also in very rural places, a GSM signal is available. The most important difference to WLAN is the fact, that at some time instance, at most one GSM cell[3] is available trough the API of the phone.

4.2.4 Precision

The maximal dimension of a GSM cell is about 30 km [5, p. 24]. This implies, that the GSM information provides only little information about the users location, but only allows for rough estimation. In places, where the net is very dense, the cell is often changed due to load balancing and signal improvement.

4.2.5 Remarks

The knowledge about the cell that the user is in has, compared to WLAN or GPS information, the least worth because it is very unprecise in the sense that the seen cell could be very different at the same location.

[3] A mobile phone keeps track of multiple antennas internally for performing a cell handover.

4.3 GPS

The global positioning system provides very precise information about the location and the speed of an object. A GPS device receives the exact position and timestamp from multiple satellites (at least four) which enables it to calculate the exact position of an object.

4.3.1 Identification

Each point of the earth's surface can be characterized by its longitude λ and latitude φ. The longitude describes the point's position east or west in respect to the zero meridian. As there is no natural zero boundary for the longitude (as the equator is for latitude), the zero meridian was defined in the 19th century through the Greenwich observatory (see Figure 5). The latitude is defined as the north or south distance with respect to the equator. To avoid negative numbers, the notation consists of the magnitude of the two angles, subsequently with N for north and S for south for the latitude and E for east and W for west for the longitude. GPS delivers very accurate and precious data. For reasonable statements, this data needs to be converted. The distance between two GPS coordinates (λ_1, φ_1) and (λ_2, φ_2) is given by:

$$d = r_{earth} \arccos\left(\cos\frac{2\pi(\lambda_1-\lambda_2)}{360^\circ}\cos\frac{2\pi\varphi_1}{360^\circ}\cos\frac{2\pi\varphi_2}{360^\circ} + \sin\frac{2\pi\varphi_1}{360^\circ}\sin\frac{2\pi\varphi_2}{360^\circ}\right)$$

As two GPS coordinates are mostly very close by, this equation can confidently be linearized.

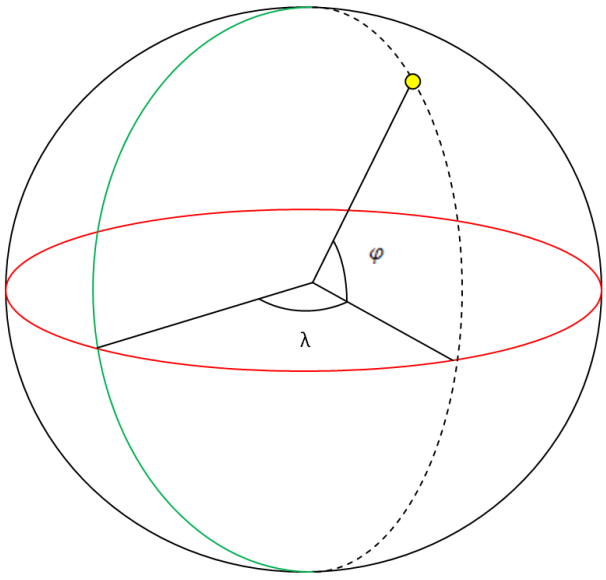

Figure 5: equator (red circle), zero meridian (green circle)

4.3.2 Availability

In general, GPS is freely available worldwide. But in this context, the term availability better covers the fact, that the GPS reception is mostly only possible outdoors. There are rare situations where a GPS signal can be received indoors. But in general, the reception is ideal within a clear line of sight to the sky.

4.3.3 Precision

GPS has a accuracy of a few meters [3]. Our practical experience however, shows that this accuracy is achieved only in the best case. It mainly depends on the speed of the users movement and the actual number of satellites receiving information. Sometimes the effective location is up to 100 m away from the GPS measurement.

4.3.4 Remarks

A GPS measuring point is in this context the most valuable data. It provides precise knowledge about the user's location without further knowledge or data processing. Having several of them allows for a very accurate distance and speed calculation.

4.4 Notation

4.4.1 Definition of the Measuring Point

A *measuring point* is defined by one piece of information consisting of a WLAN MAC address, a GSM cell CGI or coordinates from GPS. It always has a timestamp and a correlating userid, which maps the information to a specific user.

4.4.2 Definition of Estimation

Both location and movement estimation are done by analyzing an interval of 5 minutes. Logging every 10 seconds, we potentially have 30 time instances with one or more measuring points. This is a key parameter of the system. Making the interval too long, decreases the resolution of the estimation. Reducing the interval makes an estimation of movement hard.

5 Data Visualization

5.1 Web Frontend

To visualize the measured data, a web frontend was implemented. It has many advantages over images, because Javascript adds a lot of features for direct interaction. Figure 6 shows a screenshot of the site. The red, green and blue dots denote the occurrences of some measuring points. For convenience, they are ordered by their first occurrence. All of these points provide additional information when one hovers over them with the mouse. The abscissa represents the time evolving in 1 minute steps indicated by the distance between two black bars. For each time instance (interval of 10 seconds) the ordinate depicts the measuring point seen at this time. The points among the time line depict the location and movement estimations.

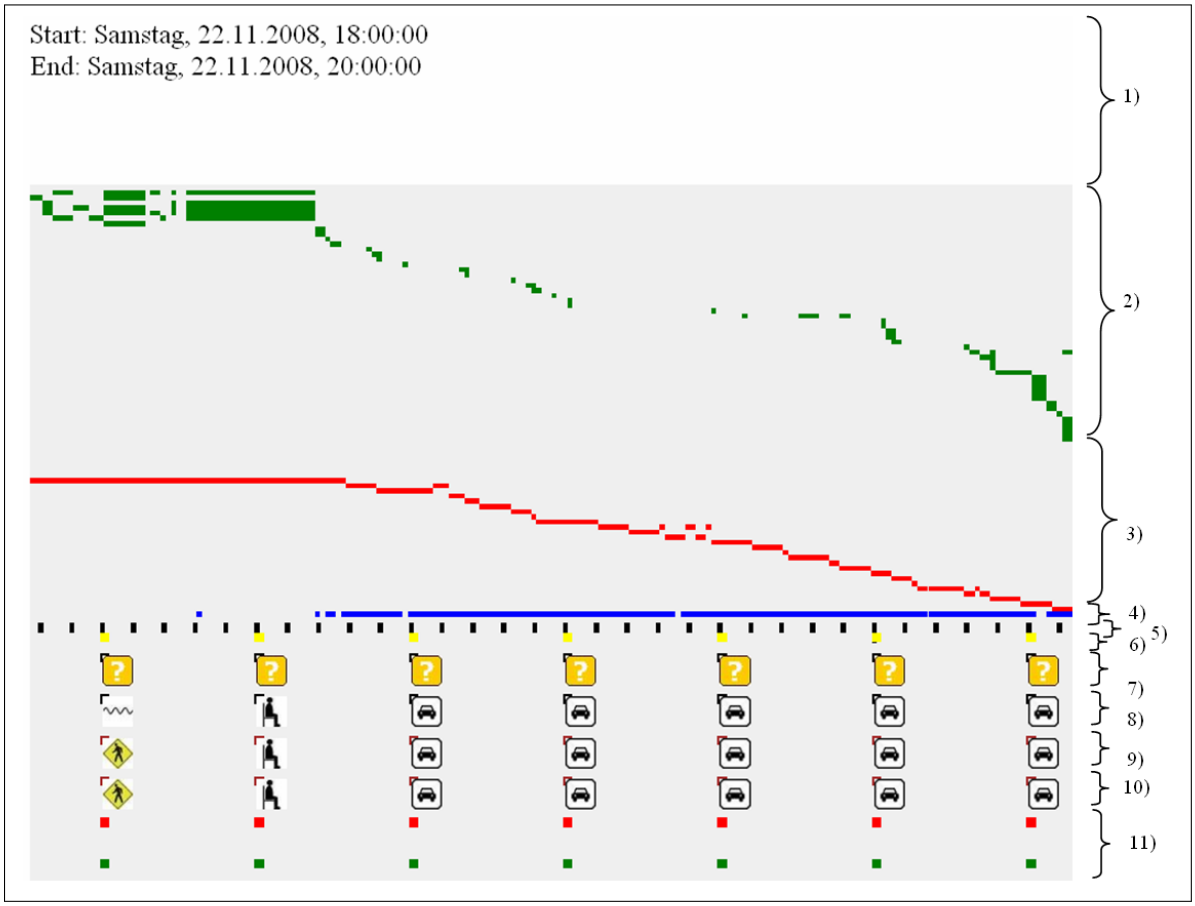

Figure 6: data visualization in a web browser

The header of the diagram *1)* shows the weekday and the exact start and end of the visualized data. The green dots *2)* depict WLAN occurrences, where the red ones *3)* show the seen GSM cell for each time instance . The blue dots *4)* indicate, that there are

available GPS measurements. As mentioned above, the black bars *5)* indicate the time in one minute steps. The yellow dot *6)* provides information about the location when hovered by the mouse. *7)* denotes the movement type (if any), which was found in the journal. *8),9),10)* are the suggestions from three different movement detection algorithms (the symbols denote classes), while *11)* provides functionality for training the SVM.

5.2 Manual Data Collection

For later use it is very important to have reference data depicted by some time intervals where a location and/or movement is/are defined. This data is important for the later evaluation of the algorithm's performances. For training the classifier the web frontend in Figure 6 is used. Such a training-vector only consist of a timestamp and the kind of movement (standing, walking, moving by train, moving by car). The journal used for the performance analysis always consist of a timestamp, a location and/or a movement type. This data is collected using a simple website.

6 Data Gathering

This part discusses some data structures, which could be derived from the log data in order to approximate the users location or movement. Because many of these algorithms take a lot of time to be computed, they are computed in a daily batch mode.

6.1 GSM,WLAN Positions

When GPS coordinates are available the locations of WLAN and GSM can be estimated. If there are multiple coordinates available for a cell or WLAN, the known coordinates are averaged. The estimation accuracy grows with the number of measuring points. This rough position estimation could then be used for determining the user's environment.

6.2 WLAN Connectivity Graph

The WLAN connectivity graph is defined by $G_W = (V, E)$. V consists of the set of all different MAC addresses ever seen for any user. Two vertices are connected if a time instance exists, where both WLAN were seen. This graph can then be used to approximate distances between nodes in a hop-like manner. It can be built in spacial and temporal domains. Figure 7 shows a simple example. The degree of a node then directly relates to the density of WLANs in a certain area.

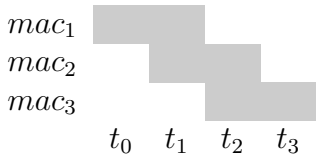

Figure 7: Sample occurrences

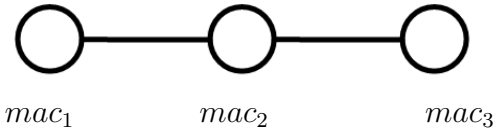

Figure 8: Resulting Graph from occurrences from Figure 7

An example of this graph can be found in Appendix 13.2.

6.3 WLAN Position Approximation

There is another possibility to approximate a WLAN's location. For this method, a WLAN-vertex in the connectivity graph G_W needs to be connected to two nodes with GPS

information. This approximation is rather imprecise but allows a rough approximation of the WLANs location. Like before, the coordinates of the two outer WLANs are just averaged.

7 Location Estimation

The preceding Abakabar application already had a location estimation feature. It allowed adding personal places, which then were recognized. But the system was not able to capture structures for example having different rooms in one building.

7.1 Definition of Location

In our context, a location is a hierarchically defined and categorized place of a certain dimension. Each location has a parent location. The root of this tree is the world. All locations are arranged in categories. These categories are sorted by size and are defined by:

- Country
- Region
- Village
- Area
- Building
- Room

The definition of *country* is self-explanatory. A *region* is given by a division of a country. In Switzerland, this division is given by the cantons. A *village* is the subdivision of a region. This subdivision is defined by different postal codes. The category was introduced to represent big areas for example a railway station, a stadium and so forth. A *building* is a site with a well defined postal address. A building allows the embedding of a *Room*. Figure 9 shows an example. This tree is held in the database. The three upper parts

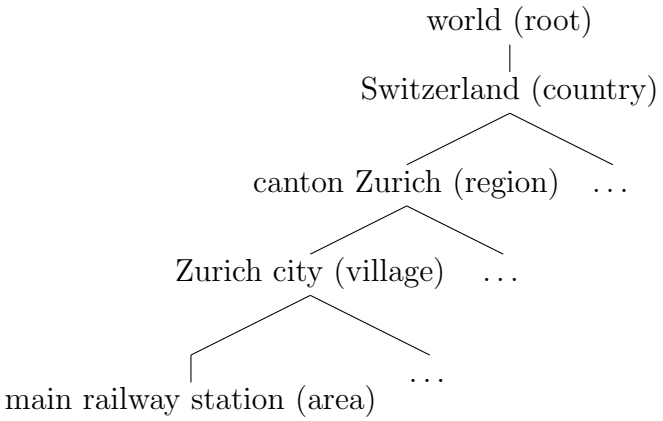

Figure 9: Example hierarchy for Zurich main rail station

(*Country*,*Region*, *Village*) are up to now hand maintained. By using third party providers, the handling of the tree could be automated by coming up with suggestions for locations.

7.2 Location Acquisition

The locations are learned from the user's input. The user can decide if a location is public (for example a railway station) or if it is private (for example his/her accomodation). New places can be added online over the mobile device. If a place is tagged as location, all the concerning log information is stored as characteristical for that place. For places with dimensions larger than the range of a WLAN, the data of the WLAN connectivity graph (see section 6.2) is taken into account. This means that WLANs which are 1 hop (for buildings) or 2 hops (for areas) away are taken into the set of characteristical WLANs. Up to now only the leaves of the tree are taken into account for location estimation.

7.3 Implementation

The recognition of a place can be done by preloading all the users characteristical WLAN, GSM and GPS information. Then for each location, a metric is calculated by:

$$m(location_n) = \frac{n_{WLAN,hits}a_{WLAN} + n_{GSM,hits}a_{GSM} + n_{GPS,hits}a_{GPS}}{n_{WLAN,characteristic}a_{WLAN} + n_{GSM,characteristic}a_{GSM} + n_{GPS,characteristic}a_{GPS}}$$

where $n_{X,hits}$ are the numbers of found characteristical information, $n_{X,characteristic}$ the characteristical information for a given place and a_X are constants for weighing the importance of the information. It always holds that $a_{GPS} > a_{WLAN} > a_{GSM}$, because GPS coordinates bear much more information than information about a GSM cell. This measure is always $\in [0, 1]$. The location with the highest metric is then chosen as the estimated location.

8 Movement Estimation

One main goal of the project is to make statements about the user's movement. As far as possible, these statements should not base upon third party imported data[4]. The possible kinds of movements (later called classes) were refined to:

- standing ($c = 1$)
- walking ($c = 2$)
- moving by car ($c = 3$)
- moving by train ($c = 4$)

Each of these classes has its characteristics. Standing for example, can be detected very easily just by looking for a WLAN that appeared constantly over a given time. This also means that statements about standing are pretty reliable, because there is no ambiguity. Walking or moving in general is more complicated, because different kinds of movement lead to similar patterns. Having GPS data is very helpful because the exact speed and place where somebody is going is known.

8.1 Movement Measurements by GPS

This is the only part of the project which uses data from a third party provider to make a statement about the user's movement. It is further important that the following algorithm is only applicable if the exact position of a user is known, which means that GPS is available over a certain period. The idea is to check the users position against a map to determine whether the user moves by train or by car. The data is provided by openstreetmap [7], which is a free-to-use map provider and the information is available for the whole world. For performance reasons, only the data of the Swiss railsystem was used.

8.1.1 Map Data Format

Both streets and railroads are given by subsequent points in form of GPS coordinates. This means that every subsequent pair of points forms a vector.

8.1.2 Distance to Railroads and Streets

The algorithm has to return the distances to the next road and the next rail. The definition of this distance is illustrated in Figure 10.

[4]E.g. maps or coordinates-to-address mappings.

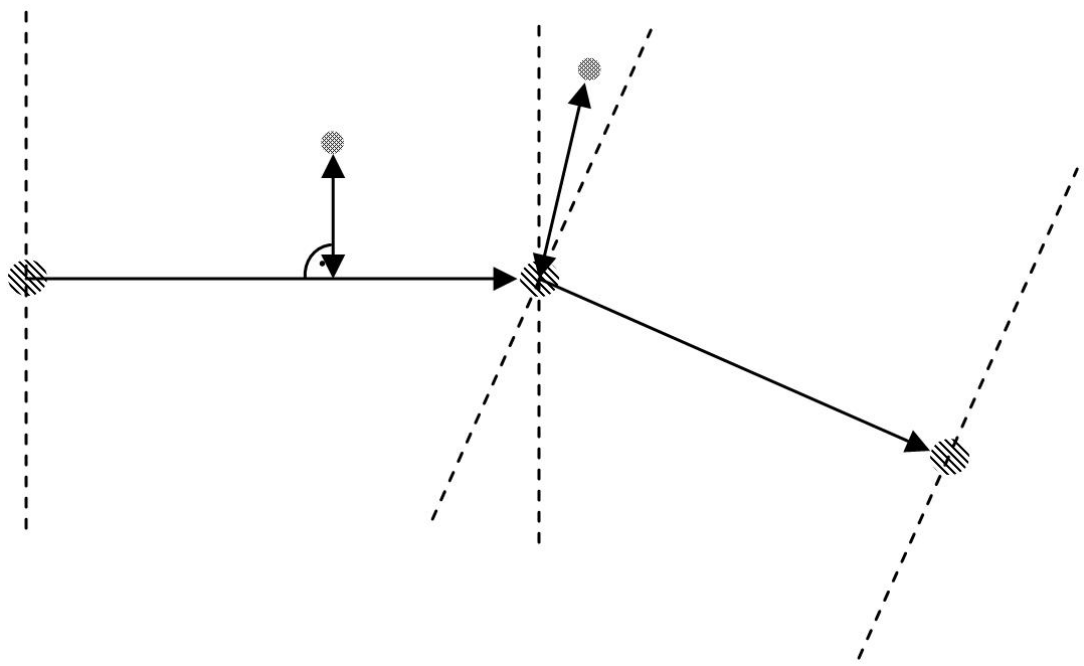

Figure 10: The striped spots are the known locations of a railroad or street, where the dotted ones identify GPS measurements.

If the point lies within the perpendicular boundaries of two railroad/street-points, the distance is defined by the perpendicular distance. If this point lies outside these boundaries, the distance is given by the shortest gap to a railroad/street-point. Figure 11 shows the distance to railsroad/street in details.

The vector pointing from (x_1, y_1) to (x_2, y_2) is given by

$$\vec{r}_{12} = \begin{pmatrix} x_2 - x_1 \\ y_2 - y_1 \end{pmatrix}$$

The vector from (x_1, y_1) to (x_0, y_0) is represented as

$$\vec{r}_1 = \begin{pmatrix} x_0 - x_1 \\ y_0 - y_1 \end{pmatrix}$$

Building the vector $\vec{p}$ which is perpendicular to $\vec{r}_{12}$ can be retrieved by rotating $\vec{r}_{12}$ by 90 degrees:

$$\vec{p} = \begin{pmatrix} \cos\alpha & -\sin\alpha \\ \sin\alpha & \cos\alpha \end{pmatrix} \vec{r}_{12} = \begin{pmatrix} y_1 - y_2 \\ x_2 - x_1 \end{pmatrix}$$

for $\alpha = \frac{\pi}{2}$. The distance d is now given by the length of the projection of $\vec{r}_1$ onto $\vec{p}$:

$$d_{rail,road} = \frac{\vec{r}_1 \cdot \vec{p}}{|\vec{p}|} = \frac{|(x_0 - x1)(y_1 - y_2) + (y_0 - y_1)(x_2 - x_1)|}{\sqrt{(x_2 - x_1)^2 + (y_1 - y_2)^2}}$$

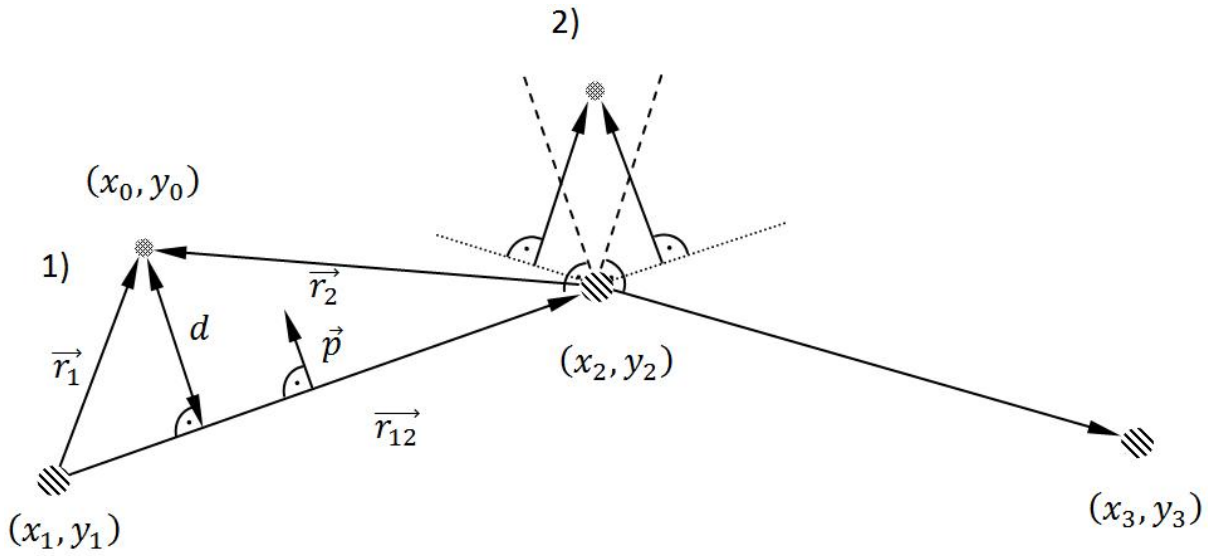

Figure 11: Detailed view

To detect if the point is lying inside the defined perpendicular boundaries the point has to fulfill the following inequation:

$$sgn(\vec{r}_{12} \cdot \vec{r}_1) \neq sgn(\vec{r}_{12} \cdot \vec{r}_2)$$

This means, that the scalar products onto the vector $\vec{r}_{12}$ have different signs. If this inequation is not fulfilled, one just take the distance to the nearest railroad/street point.

8.1.3 Speed Estimation

Having n logs in a given time interval with GPS coordinates, the average speed can be calculated as:

$$v_{GPS} = \frac{1}{n-1} \sum_{i=0}^{n-2} \frac{d(\lambda_{i+1}, \lambda_i, \varphi_{i+1}, \varphi_i)}{\Delta t(i+1, i)}$$

8.2 Movement Metrics by WLAN

This part describes an approach for speed estimation using only WLAN occurrences. The main problem is that there is no a-priori knowledge about the WLAN densities.

8.2.1 Problem Description

For simplicity it is assumed that the range of a WLAN forms a circle with the radius r. As illustrated in Figure 12, one can imagine a person following one of the arrows in the picture. Since the distance to the WLAN access point is unknown, it is not distinguishable

which arrow (1 or 2) the user is following. The only thing that is measurable is the time duration during which the WLAN was seen.

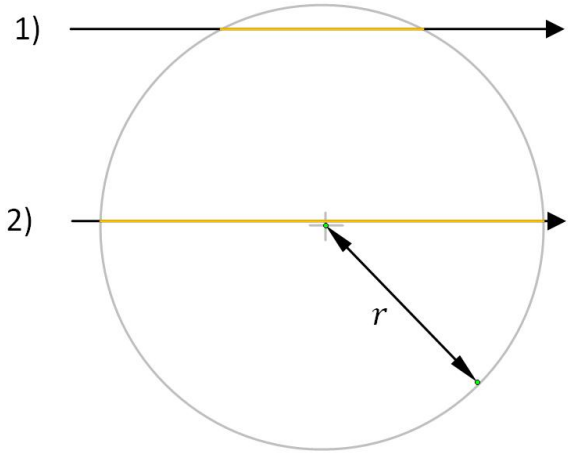

Figure 12: WLAN access point indicated by the cross

8.2.2 Approximation

Introducing the random variable $D \sim \mathcal{U}(0, r)$ which expresses the perpendicular distance to the WLAN passing through the range, and defining the random variable L as the length of seeing the WLAN crossing the range and is given by:

$$\begin{aligned} \left(\frac{L}{2}\right)^2 + D^2 &= r^2 \\ L &= 2\sqrt{r^2 - D^2} \\ E[L] &= 2E[\sqrt{r^2 - D^2}] = \int_0^r \frac{2}{r}\sqrt{r^2 - x^2}dx = \frac{r\pi}{2} \end{aligned}$$

This leads to the conclusion that in average a WLAN is seen in $\frac{r\pi}{2}$ of its length. So, at least in average, the speed can be stated as:

$$v_{approx} = \frac{r\pi}{2t_{WLAN}} \tag{1}$$

where t_{WLAN} can be extracted directly out of the logs, and r is a statistically revealed constant. As seen later, it suffices to take t_{WLAN} into account for movement recognition. The assumption that the reception area of a WLAN is a circle is very naive. Even in free space the directional characteristic of the antenna would not be spheric. But to keep the calculations simple, the circle is always the easiest choice. Also the choice of the random variable D is held as simple as possible.

8.2.3 Practical Implementation

There are two facts which are worthwhile emphasizing. If the received signal of a WLAN is very low, the SNR could be just around the threshold which leads to discontinuous patterns, as illustrated in Figure 13. Therefore the occurrences matrix is filtered before further processing takes place. The filter needs low-pass characteristics in order to filter out fast changes of the signal.

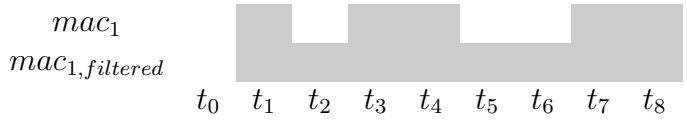

Figure 13: Noisy and filtered occurrences

A further effect has to be taken into account. The quantity of WLANs seen in a certain interval play no role. This makes sense by recalling the initial assumption: The algorithm should not depend on any prior knowledge of the area we are moving in. It is clear that the quantity of WLANs clearly has no impact on the user's speed. The other fact is that there are many WLANs which have exactly the same pattern in the diagram. This is due to the fact that professional access points are able to provide several SSID with different MAC addresses in one box. Then the pattern does not provide more information than if there was only one access point.

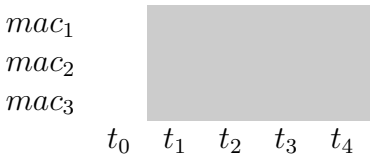

Figure 14: Redundant occurrences

The occurrences of the WLAN mac_1, mac_2, mac_3 in Figure 14 contain no more information than the degenerated occurrence in figure 15.

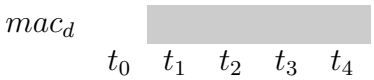

Figure 15: Degenerated occurrence (after redundancy elimination)

8.2.4 Summary

The algorithm for estimating the speed using WLAN occurrences can now be summarized to the following steps:
The WLAN occurrences of the time interval, of which the approximation is applied, are

loaded (sorted by first time of occurrence) into the occurrences matrix A. Then, for each row of the matrix, the low-pass filter is applied to suppress unwanted noise. After that lines with the same patterns are filtered out. The last part averages the maximum occurrence of each WLAN with a unique pattern. This algorithm returns the metric for the time interval $[t - 300, t]$:

$$t_{WLAN}(t)$$

8.2.5 Remarks

This algorithm works best, when many WLANs are available. Because then, the initial assumption that the distance to any station is uniformly distributed, fits best and the averaging becomes more meaningful. In the other extreme, when there are just a few WLANs, the algorithm's variance increases and may return small $t_{WLAN}(t)$ which implies very fast movement.

8.3 Movement Metrics by GSM

The GSM related logs carry similar information to that of the WLAN related logs, just in another scale. The problem here is that there are very discontinuous patterns showing up without movement. As mentioned above, the behavior of the patterns is dependent on the density of cells. But since GSM is available almost everywhere in Switzerland [5], the information could be very useful to determine fast movement such as when traviellng by train or car. Therefore the very same algorithm as in WLAN movement detection was applied. The part which removes identical patterns in occurrences could be omitted.
This algorithm returns the metric for the time interval $[t - 300, t]$:

$$t_{GSM}(t)$$

$t_{GSM}(t)$ represents the average duration of occurrences in a certain time interval.

8.4 Movement Suggestion

As discussed before, there are 5 measures to make a suggestion about a user's movement. They are given by:

d_{rail}	the distance to the next railroads
d_{street}	the distance to the next streets
v_{GPS}	the speed retrieved by GPS coordinates
t_{GSM}	the average duration of occurrences of GSM cells
t_{WLAN}	the average duration of occurrences of WLANs

Now 3 algorithms are proposed to estimate the users movement.

8.4.1 Greedy Algorithm

The first approach is kept very easy. The idea is to find some defined patterns in the logs and make a suggestion as soon as such a pattern is found. This means that, firstly the GPS coordinates are consulted, and only if they bear to little information, further information as WLAN is taken into account. This algorithm is given in pseudocode in the next figure.

```
Data: occurrences of WLAN, actual GSM-Cell, GPS if available
Result: movement suggestion
if GPS.isAvailable then
    if GPS.speed < 2 then
        return walking
    if abs(GPS.distanceto_road - GPS.distanceto_rail) < 30 then
        return vehicle
    if GPS.distanceto_road < GPS.distanceto_rail then
        return car
    else
        return train
if WLAN.MovementMetric > threshold then
    return moving
if WLAN.detectStanding then
    return standing
return unknown;
```

Procedure `greedy`

The *threshold* is chosen to be 160 seconds. Using equation 1, this leads to a speed of about 1 to $2\frac{m}{s}$.

8.4.2 Support Vector Machine Approach

A support vector machine is a classifier. It can be trained by a set of training objects with given classes and given feature vectors. These objects can be formalized by the set:

$$\mathcal{D} = \{(\vec{x}_i, c_i) | x_i \in \mathbb{R}, c_i \in \mathbb{N}\}_{i=1}^{n}$$

where c_i denotes the classes to which the so called feature vector $\vec{x}_i$ belongs. This data set is called the training-set of the support vector machine. In this particular example, this data set consists of the features explained in Section 8.4 and the classes mentioned in the beginning of Chapter 8. The actual processing was done by a well known library for support vector classification called libsvm [2]. The library allows one to create a model file, which represents the classification in the $\mathbb{R}^n$ space, which is learned from the training-vectors. It then allows the classification of a feature-vector. The SVM then returns the probability by which a feature-vector belongs to a certain class. In this case, the algorithm returns the most likely kind of movement.

8.4.3 k-Nearest Neighbor Algorithm

This algorithm also takes a feature vector as input and returns a classification. Having the given point $\vec{p} \in \mathbb{R}^n$ which is needed to be classified, the k nearest points $x_1, ..., x_k$ to $\vec{p}$ are sought. In our case this is done using the Euclidian distance. In particular for $k = 1$ we are looking for the closest point:

$$\vec{x}_{nearest} = \min_{\vec{x}_i \in \mathcal{D}} |\vec{x} - \vec{p}|_2 \qquad k = 1$$

As these found points could belong to different classes, only the majority is considered. This implies k to be odd-numbered. For improving this algorithm, k could be chosen dynamically dependent on the density of features in the region of $\vec{p}$.

9 Performance Analysis

To make clear statements about the algorithm's performance, a test bench was needed. To do that a journal had to be kept by hand. This data is then fed into the database for later evaluation of the three algorithms. An entry in the journal consist of the time and of the place or the movement. The time span for which the diary is available is then iterated and compared against the movement estimations of the three algorithms and the location estimation.

9.1 Location Estimation Analysis

The location estimation lead to a accuracy of about 87%. This means that 87% of the time the estimation matched the given location in the journal. As the test bench was put into context, where most of the locations were tagged this value is not a surprise. This value shows that if user depending locations are tagged, they are recognized.

9.2 Movement Estimation Analysis

9.2.1 Notation

Later used percentages called precision and recall are defined by:

$$Precision = \frac{tp}{tp + fp} \qquad \in [0, 1]$$

$$Recall = \frac{tp}{tp + fn} \qquad \in [0, 1]$$

The number tp (true-positive) depicts the items correctly labeled as belonging to the class (the journal entry corresponds to the estimation of the algorith). fp (false-positive) is the number wronlgy classified as belonging to a class (altough it does not belong to) and fn (false-negative) is the count of items not labeled as belonging to a class altough it does belong to. [8]

The lines of the following figures depict the classes of movement given by the journal and the colums indicate the estimation/classification of the algorithm. For example the value 20 indicates that the greedy algorithm returned *standing* 20 times while the journal indicated *walking*.

To retrieve an overall performance meassurement of the classificator's accuracy the f-measure is calculated and is given by:

$$F = 2\frac{precision \cdot recall}{precision + recall}$$

The later used recognition rate is the rate between the number of correct (the gray dyed entries) and the number of all classifications.

9.2.2 Greedy Algorithm

	unknown	standing	walking	car	train	precision
standing	2	244	16	1	0	93%
walking	14	20	32	0	0	48%
car	0	0	2	9	0	82%
train	1	1	2	1	8	62%
recall		92%	62%	82%	100%	

Figure 16: performance of the Greedy Algorithm

average precision	71 %
average recall	84 %
F-measure	77 %
recognition rate	82 %

Figure 17: overall performance of the Greedy Algorithm

The greedy algorithm shows good performance detecting *standing*, *moving by train* and *moving by car*. Especially the last two classes are detected very reliable because the algorithm never will propose these two classes without having some GPS informations. Also in estimating *standing* it shows high precision and recall. This is not a surprise because standing is the easiest recognizable class. The main weakness of the algorithm is found in estimating the class *walking*. This comes from the fact that the decision between *walking* and *standing* is mostly[5] done by analyzing the WLAN occurrences. The reason for clasify *standing* wrongly as *walking* comes from the fact that at some locations certain WLANs are visible very sporadicly.
The reason why the greedy algorithms performs better than the other two algorithms destinguishing between *train* and *car* is due the fact that the greedy algorithm has a return value *vehicle*. If this estmation shows up the software decides for the given movement in the journal.

9.2.3 Support Vector Machine Classification

As expected, also the support vector machine approach shows good performance in detecting *standing*. The reason for this is the fact that *standing* is specified only by one feature-vector[6] which is very different (in a Eucledian distance manner) to feature-vectors of other classes.
The classification of *walking* is in comparison to the greedy algorithm better because the SVM approach does not depend on a given threshold and also takes GSM information into account.

[5]Because no GPS data is available.
[6]All WLAN and GSM cells were see all the time and the GPS data indicates no movement

	standing	walking	car	train	precision
standing	226	35	2	0	86%
walking	5	60	0	1	91%
car	0	1	8	2	73%
train	0	3	4	6	46%
recall	98%	61%	57%	67%	

Figure 18: performance of the Support Vector Machine Classification

average precision	75 %
average recall	71 %
F-measure	72 %
recognition rate	84 %

Figure 19: overall performance of the Support Vector Machine Classification

The classification between the two motored classes is worse than the one of the greedy algorithm. In comparison to the greedy algorithm, the SVM approach does not know the class *vehicle* and therefore always has to decide between *train* and *car*. This classification is better for rural areas where the density of railroads/streets is low. In the area of a city, where the most of the meassurements were made, the decision is harder because railroads and streets are often very close-by.

9.2.4 k Nearest Neighborhood Classification

	standing	walking	car	train	recall
standing	221	41	1	0	84%
walking	5	58	0	3	88%
car	0	1	7	3	64%
train	0	2	4	7	54%
precision	98%	57%	58%	54%	

Figure 20: performance of the k Nearest Neighborhood Classification

The k Nearest Neighborhood algorithm shows very same characteristics as the SVM approach. It would inicate a principal flaw of the SVM approach if these two aolgorithms do not show the more or less same performance. The major advantage of the SVM and k Nearest Neighborhood over the greedy algorithm is the fact that they are able to learn from further data for improving their performance.

average precision	72 %
average recall	67 %
F-measure	69 %
recognition rate	82 %

Figure 21: overall performance of the k Nearest Neighborhood Classification

10 Conclusion

The main question of this thesis was whether it is possible to make statements about a user's movement just by analyzing occurences of WLAN, GSM cells and GPS coordinates. However the more interesting question was, whether it is possible to make an estimation without GPS. This is important for estimating a user's movement e.g. in a building where no GPS is available or when she/he moves outdoors and no GPS localization is possible[7].

Unfortunately, not all the ideas of the task description could be realized due to the time frame given by the semester thesis. We refined the problem statement to the recognition of four classes of movement and location estimation. To improve the system, further data sources for movement estimation could be taken into account for example schedules of public transport.

There is not enough data to make general statement of the algoritms' performances. This follows from the fact that most of the data is concerning a city, where lots of WLANs and GSM cells are available. However the results show that a movement detection by GSM, WLAN and GPS is feasible with good accuracy. Even if there is no GPS data available, the classification is done reliable. Also the location estimation can be done with a high accuracy. We think that the classification of movement could be improved to provide a reliability up to 95%.

We are convinced that the idea behind this project, to provide location-based services as a framework which is open for developers of location-based services, is promising and will lead to further reasearch and commercial applications. The idea of taking the user's location and/or movement into context will be the futures approach of plenty mobile applications.

[7]This is often the case because the GPS receiver of the mobile devices are not able to acquire a user's position as fast as expected.

11 Future Work

11.1 Massive Data Gathering

To improve the proposed algorithms for estimating the locations of WLANs and GSM cells, it is very important to have as much information as possible. Up to now, there are only a bit more than 2 weeks of logs concerning only one user. The available data is sufficient to implement algorithms but not to show the functionality of the system in a large scale.

11.2 Mobile Application

Today's mobile devices provide more information about their environment than used in this project. There is a tendency to equip the mobile devices with more and more sensors, e.g. accelerometers. This enables the detection of movement from another aspect. For example walking up stairs, using an elevator or similar moevements. Another approach is using the microphone to "hear" whether the user is for example moving in a train or car. These features are better implemented directly onto the mobile device but could surely improve the accuracy of our estimations.

11.3 Server Application

In order to provide much more information about a user's location or movement, several free web services could be taken into account to improve this service. As an example, there is a commercial product which provides a coordinates-to-address mapping. This could be used to provide proposals to the user's location for easier tagging of a place. To improve the movement detection, one could try to directly map a train ride to the actual timetable of the trains. This could then lead to further estimations, where a user will be in future, because the system learned from the past, that the user left a train at a certain station at a certain time and weekday.

11.4 Improvement of the Classification Algorithms

The support vector machine and the k Nearest Neighborhood Algorithms could be extended with more features. Further features could be the maximal and minimal velocity over a certain period of time to distinguish more precisely between walking and driving a car that is stuck in traffic.
A further approach is to take into account more than five minutes of a user's movement. For example a period of 20-30 minutes. This allows a better detection of a fast moving user by considering the GSM cells.

12 Acknowledgments

I would like to thank my advisor, Michael Kuhn, whose knowledge and patience added considerably to my semester thesis. A special thanks goes to Fabio Magagna, who supported me during the thesis and shared his experience with me.

References

[1] Semester Thesis of Fabio Magagna

[2] LibSVM – A library for Support Vector Machines

[3] Erreichbare Genauigkeit bei GPS-Positionsbestimungen

[4] Eigenbehaviors: Identifying Structure in Routine

[5] Ad Hoc and Sensor Networks, Chapter 14, Mobility

[6] Magizne 'explore', p. 18

[7] Openstreetmap

[8] Precision and recall - Wikipedia, the free encyclopedia

[9] Berg Insight. Strategic analysis of the European mobile LBS market. Berg Insight BRG1352742, 2006

[10] Sensing the World with Mobile Devices

[11] Apple - iPhone - App Store und Programme für das iPhone

13 Appendix

13.1 WLAN Range Histogram

The following figure depicts the ranges, WLAN have been seen. These measurements followed from WLAN occurrences, where also GPS was available. The majority (over 80%) of the ranges lie between 60 and 240 meters.

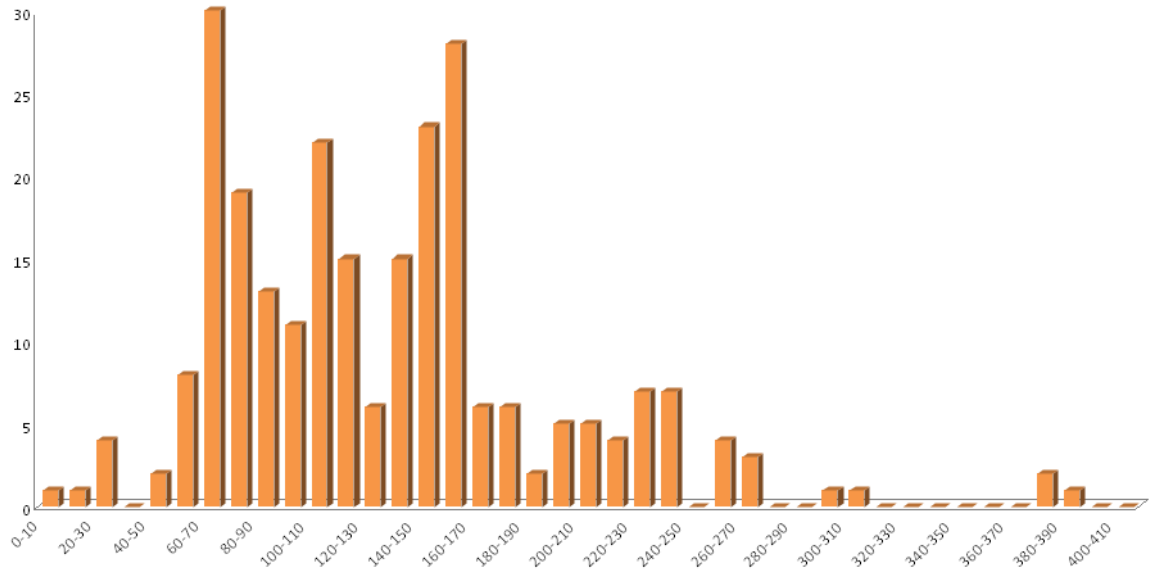

Figure 22: WLAN range histogram

13.2 WLAN Connectivity Graph Example

The next figure shows an abstract of the WLAN connectivity graph. It depicts a walk starting in the lab *ETZ G69*, leaving the building trough the *Foyer*, walking trough the ETH mainbuilding *HG*, passing the *Central* and Zurich main station (*HB*) and walking to *Ende Bahnhofstrasse*. The main information we gained is that the graph does not stay connected in practice while moving between buildings.

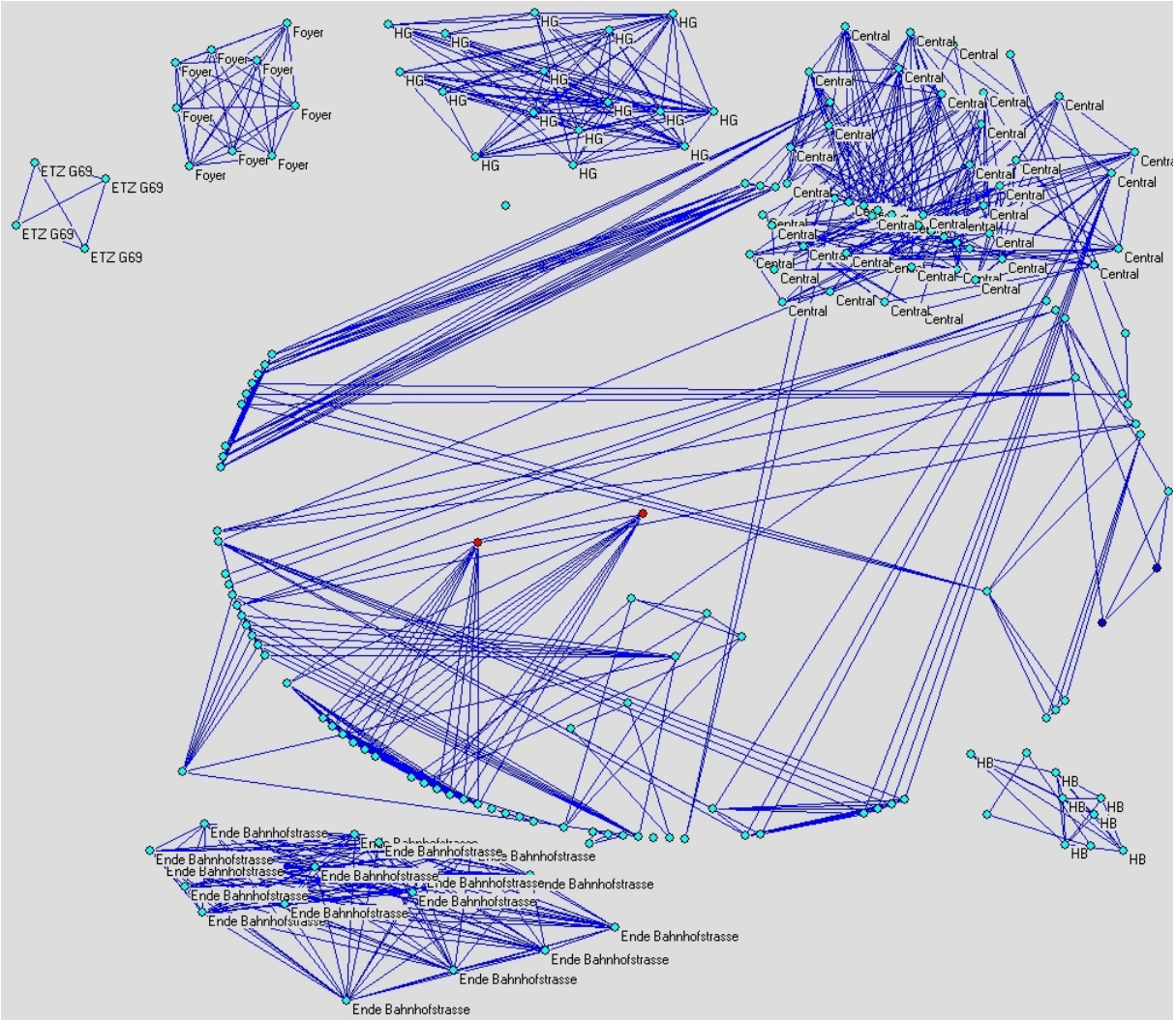

Figure 23: Abstract of the WLAN Connectivity Graph