MŰEGYETEM 1782

BUDAPESTI MŰSZAKI ÉS GAZDASÁGTUDOMÁNYI EGYETEM
Általánosés Felsőgeodézia Tanszék

Gergő Gelencsér (NWJJ20)

and

Mónika Fanczal (EXCAVA)
Civil Engineer B.Sc. students

Wi-Fi Based Indoor Positioning System

Budapest University of Technology and Economics (BUTE)
Department of Geodesy and Surveying

Supervisor:

Dr. Zoltán Siki assistant professor

Köszönetnyilvánítás

Szeretnénk megragadni a lehetőséget, hogy köszönetet mondjunk mindazoknak, akik lehetővé tették e dolgozat megvalósítását. Elsőként konzulensünknek, Dr. Siki Zoltánnak szeretnénk megköszönni segítő munkáját, és azt, hogy tanácsaival és meglátásaival mindig rendelkezésünkre állt munkánk során.

Köszönettel tartozunk az Általános- és Felsőgeodézia Tanszéknek, hogy megtiszteltek bizalmukkal, és méréseink elkészítéséhez a megfelelő mérőeszközöket a rendelkezésünkre bocsátották.

Abstract

In this study, the Wi-Fi Signal Strength's role in positioning methods was applied. A measurement method was designed and measurements were carried out at Budapest University of Technology and Economics. A fingerprinting model was built, which was used for indoor-positioning.

In accordance with this model, a community map was created: positioning, measuring and searching functions were implemented. This community map allows positioning as well as indoors and outdoors.

Contents

List of Figures

List of Tables

Definitions

Access Point (AP)[1]: is a device that allows wireless devices to connect to a wired network using Wi-Fi, or related standards. The AP usually connects to a router (via a wired network) as a standalone device, but it can also be an integral component of the router itself.

Access Pointname: used to identify cells which are part of the same virtual network. Defines a group of cells connected via repeaters or infrastructure, where the user may roam transparently. Some AP can disable the network name broadcasting, which is hidden.

Access Point Address (SSID, 00:12:D9:42:9A:B0)[2]: this address is the cell identity of the Access Point, as reported by wireless scanning.

Received Signal Strength Indicator[3]: (RSSI - how strong the received signal is). May be arbitrary units or dBm. RSSI is usually used on a 0-100 scale. That scale shows the loss of the signal from the AP, thus the closer to 0 the better. The rule of thumb is that RSSI is reliable around -50 to -60 if you have more loss than that you may see slowed connectivity.

POI (Point of Interest): is a specific point location that someone may find useful or interesting.

dBm[4]: is an abbreviation for the power ratio in decibels (dB) of the measured power referenced to one milliwatt (mW).

Web Map Service (WMS)[5]: is a standard protocol for serving georeferenced map images over the Internet that are generated by a map server using data from a GIS database.

Web Feature Service Interface Standard (WFS)[6]: provides an interface allowing requests for geographical features across the web using platform-independent calls. The basic Web Feature Service allows querying and retrieval of features. A transactional Web Feature Service (WFS-T) allows creation, deletion, and updating of features.

[1]Jean Tourrilhes: *http://linux.die.net/man/8/iwconfig* , Linux Man Page
[2]Jean Tourrilhes: *http://linux.die.net/man/8/iwconfig* , Linux Man Page
[3]http://lists.shmoo.com/pipermail/hostap/2006-December/014832.html
[4]Bigelow, Stephen: *Understanding Telephone Electronics*. Newnes. p. 16. ISBN 978-0750671750.
[5]"Web Map Service". Open Geospatial Consortium. Retrieved 2009-03-23.
[6]Panagiotis (Peter) A. Vretanos: *Geography Markup Language (GML) simple features profile.* OGC® 06-049r1.

1. Introduction

Motivation

To create a community map is complex. The goal of this study is to demonstrate some methods and difficulties to know better how such map-system works.

The main task is to explore a method for indoor-positioning, which can be used for navigation within a building. As we know, GNSS infrastructure used to provide location information, but it does not work indoors. The main part of this study tries to show how this system works by using Wi-Fi infrastructure.

Wi-Fi-based indoor localization has gained interest recently, because buildings are getting equipped with Wi-Fi Access Points (APs) for connectivity. Using these APs as location indicators removes the need for additional infrastructure. The other half of the study is about mapping: If the user's position is known, the map can be used for several purposes, like route planning or finding given POIs (Point of Interest). Specifically, this study intends to demonstrate two approaches:

- Indoor-positioning using Wi-Fi signals and corresponding coordinates
- Creating a community map from the campus which can be used for stacks of community activity and capable of creating position everywhere, especially indoors

Such aim can be reached, but as far as we know, indoor location-positioning is generally much more challenging, and it has to bridge lots of technological difficulties.

Background and problem statement

There are several existing technologies used for positioning. GNSS is one of the most common tracking solutions in the world. Unfortunately, buildings are enclosed by walls and they could not be accurately tracked by a GNSS system. Because of these issues, the GNSS solution was no longer considered as a viable option.

In the recent years, many research showed that indoor positioning systems based on Wi-Fi fingerprints can achieve a high positioning accuracy. However, the main barrier of broad adoption is the labor-intensive process of collecting labeled fingerprints. This work proposes an approach of a Wi-Fi fingerprint model. The study demonstrates through experiments that by constructing a Wi-Fi fingerprint database one can achieve a high positioning accuracy.

Wi-Fi localization using RSSI (Received Signal Strength Indicator) readings was considered as a potential solution. This was ultimately determined to be the most realistic solution for successful Wi-Fi based localization.

2. Indoor-positioning

Opportunities to use Indoor-positioning

Lots of possibilities are available to apply indoor-positioning. Each method has limitations and needs to be further developed. Smart phones with different sensors (e.g. accelerometer, gyroscope, etc.) are one of the most necessary devices to determine location- they give cost-effective methods.

Acceleration sensors

It is possible to use the mobile's sensors to determine position. The sensors can measure only relative displacements. In our case the required accuracy should be in the range of 5-10 meters and with the help of acceleration sensors one cannot achieve this goal.

Radio technologies (Wi-Fi, Bluetooth)

The radio has Signal Strength and position which is known. If these Signals can be measured, the position can be determined. Conducting distances and angles from the measurements is possible.

Creating a Fingerprinting model is also possible to describe position. The radio field can be affected by the walls, the objects within the building or by the moving people inside as well. Because of these factors, the accuracy will be fluctuating. To build a suitable accuracy is the most important aim in this task.

Non-radio technologies (Inertial, magnetic systems)

Magnetic positioning is based on the iron inside buildings that create local variations in the Earth's magnetic field. Each smart phone can sense and record these magnetic variations to map inside the building. Inertial measures generally cover the differentials of motion; hence the location gets determined with integrating and thus requires integration constants to provide results.

To achieve the best result, these technologies should be combined.

Function of Wi-Fi infrastructure

Fingerprinting

Wi-Fi fingerprinting creates a radio map of a given area based on the RSSI (Received Signal Strength Indicator) data from several access points and generates a probability distribution of RSSI values for a given (x,y) location. Live RSSI values are then compared to the fingerprint to find the closest match and generate the predicted (x,y) location.

Wi-Fi is widely used and integrated in various electronic devices. To develop a Wi-Fi-based location, one can take advantage of this. Building indoor-positioning system is rather complicated: The movement or maybe quantity of people can decrease precise positioning. This study tries to pay attention to every factor, but there are many unknown effects, which may disturb positioning and decrease the accuracy.

Advantages and disadvantages of Fingerprinting

The measured field which assumed to be permanent, it does not have disturbing factor. To know the position of the Access Point and the spreading of the Signal Strength is not necessary. The implemented measure is cost-effective and reliable.

The measured field is changing, whenever the AP position is changed. This causes systematic error in the position. New APs can be used only as new RSSI measurements. To follow the change in the Wi-Fi positions and new APs, the measurements must be repeated regularly.

Goals and Requirements

The main goal is to construct a measurement technology, which makes it possible to record and process Wi-Fi fields. Assurance metrics are expected in this work: Wi-Fi Signal's stability and the accuracy of the coordinates. It is also needed to identify usability: how unequivocal the received position is.

The system should be consistent: we want to get obvious position from the processed data on an interface which is available for each user. This interface should be user-friendly. The usage of GIS software shall not be necessary, and each data need to be available and usable on a server.

First experiment

The measurements were performed at the Budapest University of Technology and Economics, particularly in the building 'K'. For easier management, the first experiment took place in the laboratory of the department, which could be extended for larger area maybe for the whole building. Fig 1 shows the elaboration of the method which carried out at several levels. The structure of this measurement works according to the followings:

- Leica TCRA 1103 total station in robotic mode. It follows the moving prism and can automatically calculate the coordinates. Beside the moving prism a laptop was used to collect Wi-Fi signals strength.
- Ulyxes TCL API to track the prism. This software runs on the laptop beside the instrument. It controls the total station and collects coordinates and timestamps.
- Network data can be received from the Wi-Fi adapter of the Rover PC Linux by a bash script (data: mac addresses, SSID codes and Signal Strengths, timestamps; Appendix: 1.1. Linux Bash Script to collect Wi-Fi data, 1.2. AWK Script to analyze Wi-Fi data)

An AWK script was used to post process the raw observation data and create csv file. The moving device (PC) and the prism must be close to each other. It is necessary to mention that the clock on each device must be synchronized. The timestamps are used to connect the location of the prism to the Wi-Fi signal strengths. During the measurement the room was empty, thus the Wi-Fi signal were strong and clear.

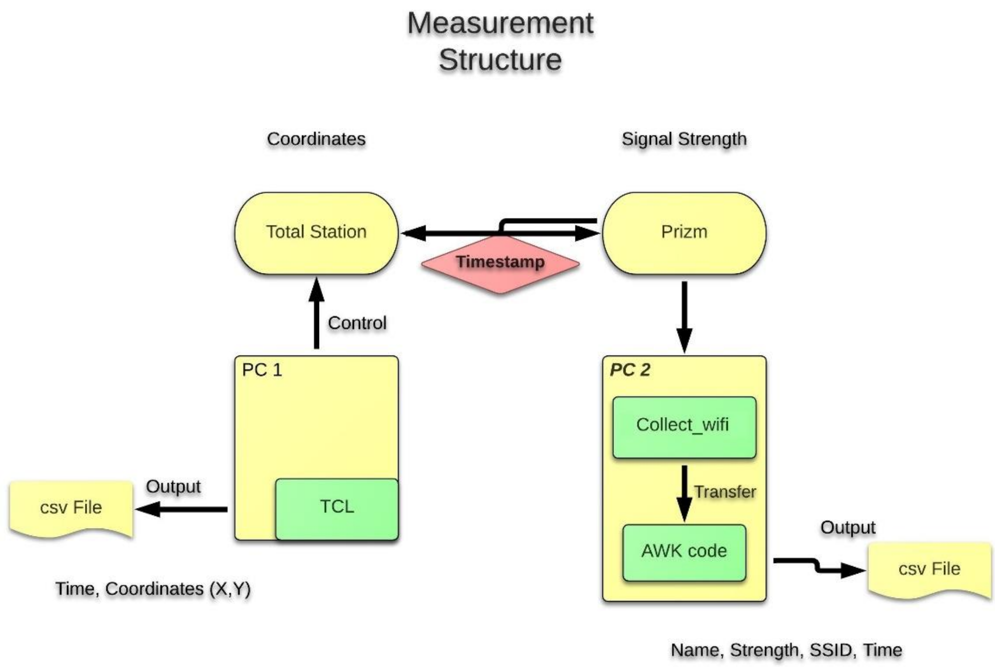


Figure 1: Measurement Structure of Fingerprinting

The Wi-Fi signals were sorted by importance. These signals were recorded continuously. The coordinates and the Signal Strengths were saved to a relational database (PostgreSQL).

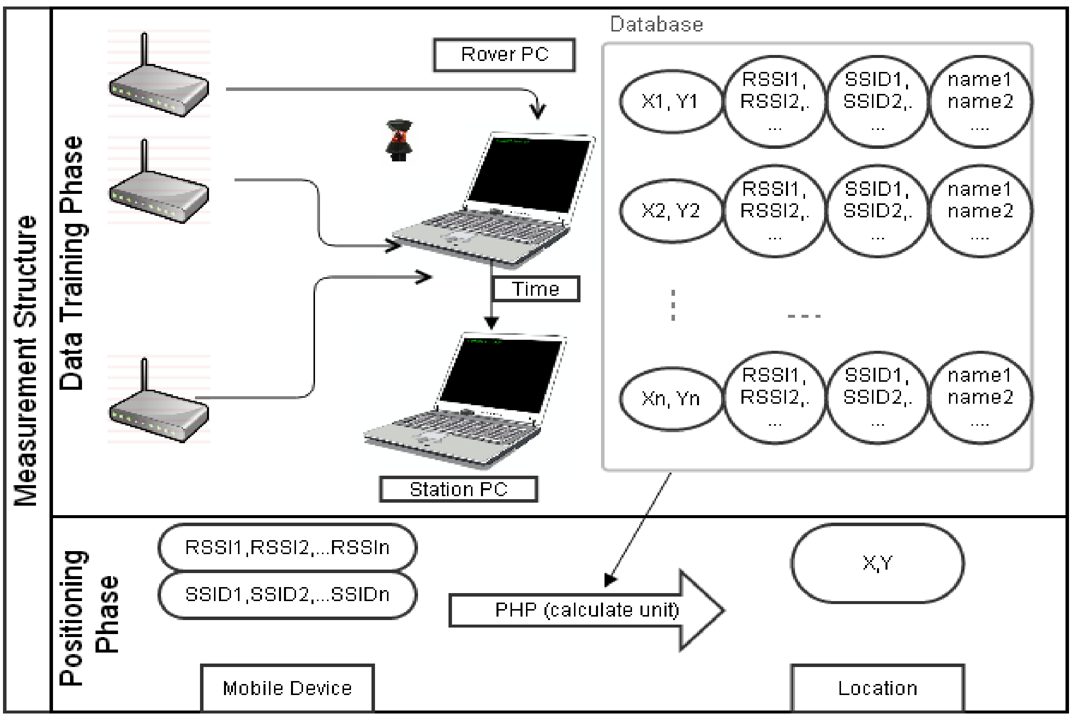


Figure 2: Measurement structure of Fingerprinting

Results

A strongly interpolated surface around the measurement points was determined. Fig 2 shows that the interpolation was rough for this measurement and it shall be smoother. The measured points were scattered.

An extensive measurement needed for more accurate results. The measurement technology was proven to be functional.

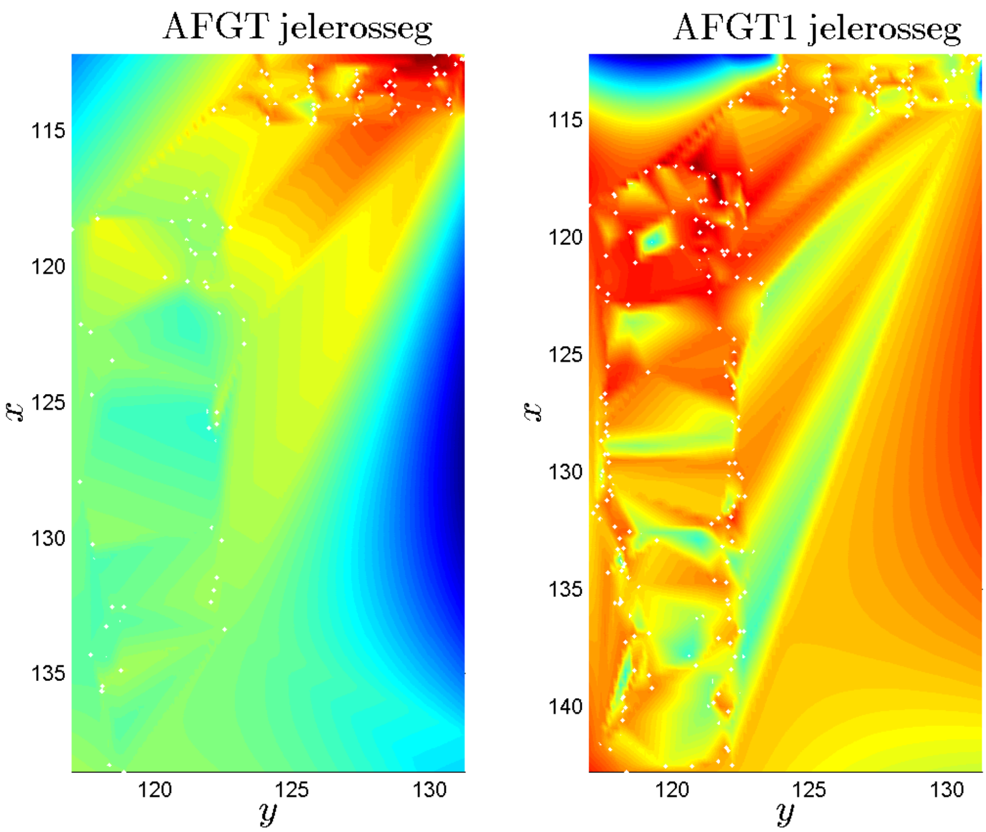


Figure 3: AFGT and AFGT1 Signal Strengths

Static experiment

Static experiment was made in order to define the stability and the standard deviation of Signal Strengths. These values help us to get real position with appropriate accuracy. The definition of the fluctuations can be measured in static mode.

This means measuring only in one position for a few minutes (20-30 minutes). The measurement was completed in relatively traffic-free time. The Signal Strengths were collected by a laptop with a bash script.

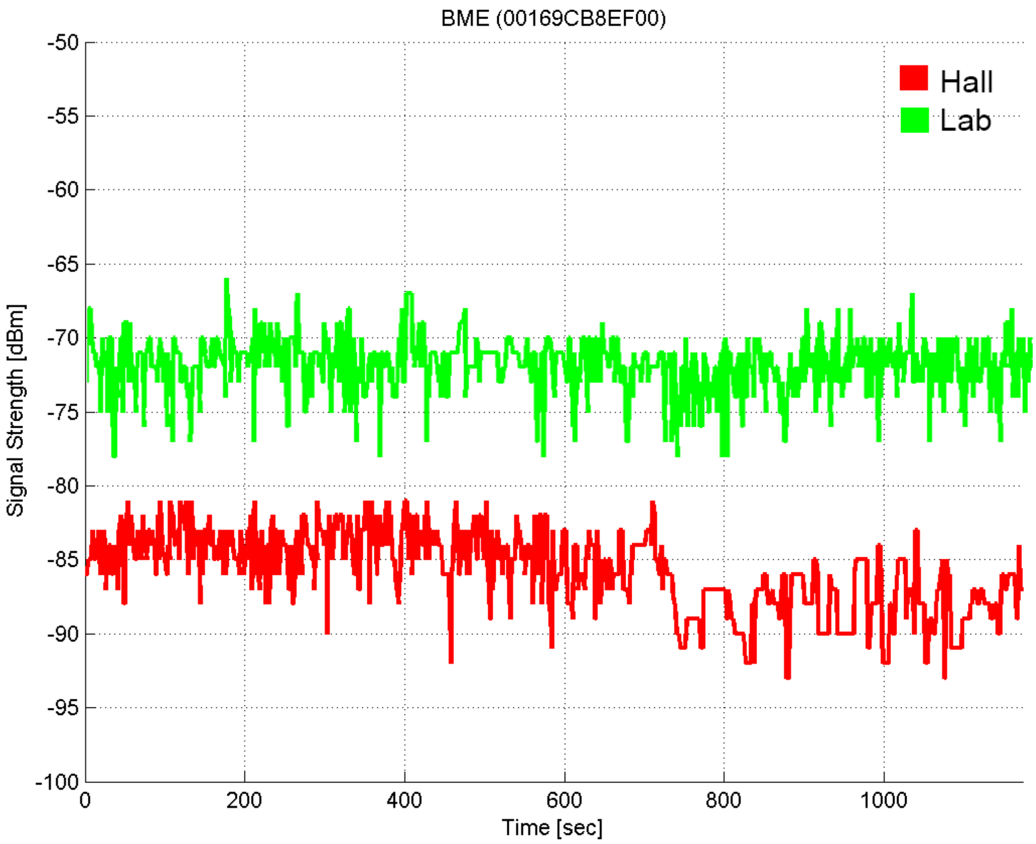


Figure 4: BME network Signal Strength fluctuation

Fig 4 shows the fluctuation of BME network's signal strength, which was measured in the hall and at the same Access Point from the lab. Based on the measurements, the fluctuation was in the range of +/-2.5 dBm. This means the position can be defined from Wi-Fi field because of the minimal starting accuracy. In the first few minutes the Signal was fluctuating.

The reason of the fluctuation is not obvious, because as far as we know, the Access Points has 12 channels: the router changes channels periodically. The measurement from the lab had two parts: For 15 minutes without traffic and for 15 minutes with crowd of people. During the measurement the effect of the presence of people on the signal was tested.

Results

The average deviation was shown to be about 2.6-2.3 dBm from static measurement. It suggests the usage of +/- 3.0 dBm tolerance value in the Positioning Phase. The movement of people did not affect the fluctuation substantially.

Name of APs detected	SSID of APs detected	Standard deviation of APs
AFGT3	B0:48:7A:E1:8F:6E	2.382
BME	00:12:D9:42:9A:B0	3.549
BME	00:12:D9:4A:2C:70	2.376
BME	00:16:9C:B8:EF:00	2.673
BME	54:78:1A:88:58:70	1.494
eduroam	54:78:1A:88:58:71	1.495
eduroam	00:16:9C:B8:EF:01	2.595
visitor	00:12:D9:42:9A:B1	3.513
vkkt	00:18:39:A2:52:B8	1.009
vkkt_labor	00:1D:7E:C6:A6:B3	4.855
Average deviation		**2.594**

Table 1: Average deviation of Signal Strengths in the hall

Name of APs detected	SSID of APs detected	Standard deviation of APs
AFGT	00:21:91:96:35:66	2.799
AFGT3	B0:48:7A:E1:8F:6E	5.107
AFGT4	B0:48:7A:E1:8F:D6	2.966
BME	00:12:D9:42:9B:00	1.479
BME	00:16:9C:B8:EF:00	1.922
BME	54:78:1A:88:58:70	4.386
BME	00:12:D9:42:9B:00	1.479
eduroam	00:12:D9:42:9B:01	1.547
eduroam	00:16:9C:B8:EF:01	1.950
eduroam	54:78:1A:88:58:71	3.796
eduroam	00:12:D9:42:9B:01	1.547
epitesz-DLA	90:F6:52:2A:08:38	1.214
epszerk	00:1C:10:67:65:8B	2.494
epszerk	C0:4A:00:39:47:E8	1.660
epszerk	C0:4A:00:39:48:2A	1.729
fotonet	00:14:BF:BA:F2:09	2.110
fotonet2	C8:3A:35:4C:A6:98	2.612
Magas2	00:1D:7E:2C:7E:D0	2.055
rajzi2	00:25:86:D5:B0:84	1.565
vkkt	00:18:39:A2:52:B8	2.120
vkkt	00:18:F8:FB:C6:34	1.956
vkkt	00:18:F8:FB:E3:80	2.031
vkkt_labor	00:1D:7E:C6:A6:B3	2.020
Average deviation		**2.284**

Table 2: Average deviation of Signal Strengths in the lab

Second experiment

This measurement was performed on the ground floor, on the corridors of building K, for a larger area. We used the same method as in the first experiment.

Stand Position	Number of coordinates detected	Number of SSIDs detected	Number of Access Points detected	Point Loss	Number of unique Wi-Fi detected	The Area of the Test Site
Hall	1 372	183	105	4,4 %	139 704	3 754 m²
Labor	451	93	73	22 %	39 057	425 m²

Table 3: Measure analyzing from the experiments

The csv data from the 2 laptops were loaded to PostgreSQL Database. These data were connected by the timestamps. The first method of the connection based on the correlation. The table confirms that this quantity of data needs too much time to connect (2.5 hours in case of the Labor. In the Hall it would take days to do so.)

To get a clear and obvious connection, the Wi-Fi based measurements (within 1 seconds and with the same SSID) were averaged. Consequently, every coordinate has 1 SSID measurement. Compare to the correlation query, there is only 4.4 % point of loss. (Appendix: 1.3. SQL Script to import coordinates and Wi-Fi data to relational database)

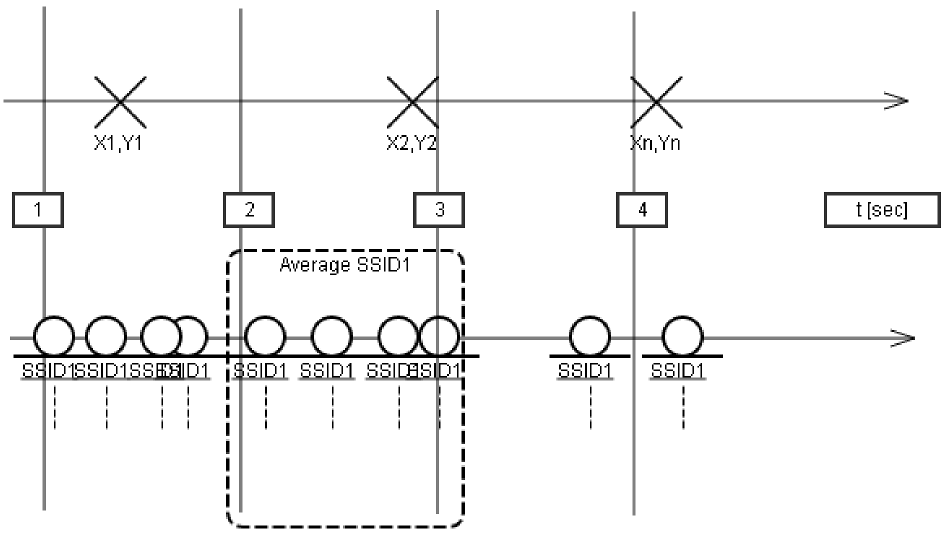


Figure 5: The role of averaging measurement data

Solutions to create position

Grid interpolation method with raster algebra

A Grid was interpolated from the measurements for each Wi-Fi using GDAL. It is a translator library for raster and vector geospatial data formats that is released under an X/MIT style Open Source license by the Open Source Geospatial Foundation. GDAL uses IDW (Inverse Distance Weight) algorithm to create grid from scattered data. For every grid node the resulting value Z will be calculated using formula:

$$Z = \frac{\sum_{i=1}^{n} \frac{z_i}{r_i^p}}{\sum_{i=1}^{n} \frac{1}{r_i^p}}$$

where:

- z_i is a known value at point i,
- r is a distance from the grid node to point i,
- p is a weighting power,
- n is a number of points in search ellipse[7].

In this method the weighting factor *w* is

$$w = \frac{1}{r^p}$$

All of the measurements must be used, only the moving Signal Strengths can be removed (like Wi-Fi hotspots). Grid points can be selected between deviations +/- 3 dBm strength Wi-Fi per Wi-Fi.

Based on the connected database, buffer zones were created around the measured points. Each buffer zone has 3 m radius. Using every SSID, union was made to the buffer zones. In respect of each buffer zone, the whole grid needs to be masked. It helps to determine real measurement results and at the same time it helps avoiding extrapolated results. (Appendix: **1.4. SQL** Script **to** create **buffer zone**).

In this way, the masked grid was reloaded to the database as raster.

Every raster contains one Strength-Grid concerning one SSID.

To process raster data, GIS solution is the most evident method. In the Positioning Phase the mobile device can sense the SSID and the Strengths which is 'visible'. A PHP script(Appendix: 1.7. PHP script for the positioning query) was made to sort the appropriate Strengths from raster in the database. The appropriate raster (compared to the detected Signal Strength)marks the pixels with 1 if the pixels are within the +/- 3 dBm tolerance limit. In other cases it is 0.This tolerance limit was defined by the Wi-Fi fluctuations. (Appendix: 1.5. Typical Signal Strength fluctuations in the hall, 1.6. Typical Signal Strength fluctuations in the lab)

This function runs through each raster of the Wi-Fi APs. The received binary raster will be multiplied. The result is a grid which has all of the detected SSID section with 0/1.

[7]Search ellipse: Search window in gridding algorithms specified in the form of rotated ellipse.

A preliminary image about the potential position can be described from this raster. In lucky cases this position appears in one closed polygon. In other cases, more polygons can appear. At this case the largest polygon will be used referring to the law of probability.

To get a position, a centroid needs to be calculated for every polygon. This is a reliable approximation to determine location indoors. The accuracy is characterized by the size of the polygon. Ultimately the community map (which discussed later) gets back a position in JSON-format string to display.

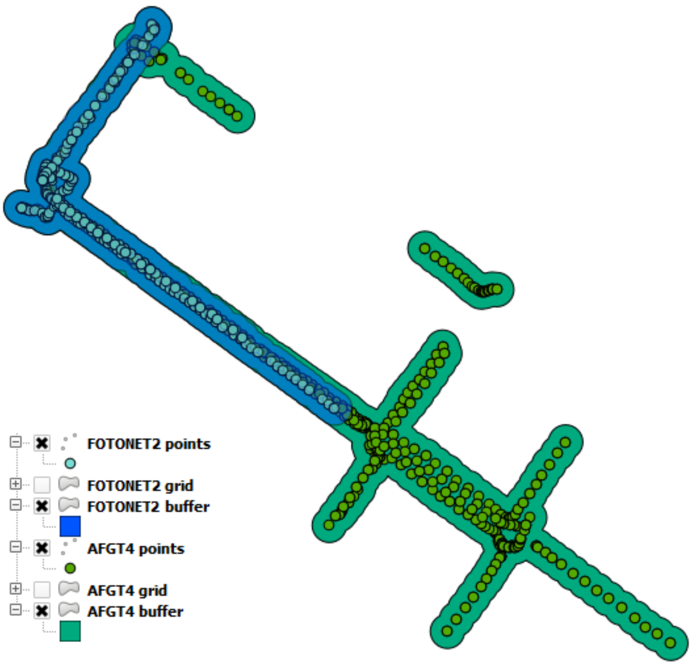


Figure 6: AFGT4 and Fotonet2 points with buffer zones

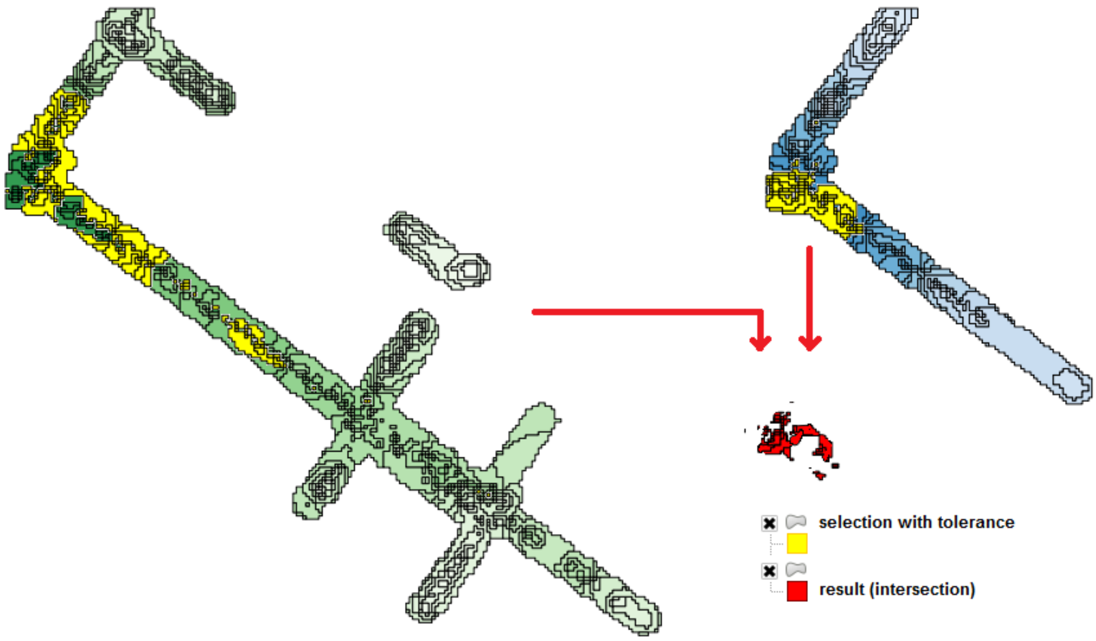


Figure 7: Interpolated grids and it's intersection

Intersection from scattered polygons

Buffer zones with the radius 0.5 m were generated to the scattered points. The Positioning Phase, the incoming Signals were corrected +/- 3 dBm tolerances in the same way as Grid Interpolation method. The processing PHP script selected the suitable polygons based on these values. The selected polygons were intersected twos. The last polygon's centroid gives the position. The accuracy of the measurement structure was specified by the area of each polygon.

Compared with the Grid Interpolation method, this solution relies only on the measured data.

Mathematic solution / Nearest Neighbor

This method simply calculates the minimum distances between the live RSSI readings and each reference point fingerprint in the n-dimensional space. Two versions of the Nearest Neighbor can be used: unconstrained search-space and constrained search-space.

Unconstrained search-space looks at the entire fingerprint map to find the closest match. Constrained search-space only searches within a given distance from a previously predicted location. In this task the Access Points' position was unknown. This method cannot be used for positioning.

$$d_E(P,Q) = \sqrt{[\sum_{i=1}^{n}(p_i - q_i)]^2}$$

Results

Test measurements in Positioning Phase: measuring with mobile device on generally known coordinates makes possible to investigate the values.

Tolerance [dBm]	Method	Buffer zone [m]	Error Y [m]	Error X [m]	Error of vector length [m]	Accuracy from area [m]
6	raster		6.630	-3.610	7.549	0.324
10	raster		12.925	-9.307	15.920	3.619
6	polygon		6.635	-3.605	7.551	0.324
10	polygon		12.497	-8.919	15.353	3.943
3	scatter	1.0	12.918	-9.173	15.843	0.948
3	scatter	0.5	12.558	-8.960	15.427	0.289
3	scatter	0.3	12.558	-8.960	15.427	0.050
6	scatter	0.5	11.109	-7.711	13.523	0.798
10	scatter	0.5	7.344	-4.858	8.805	1.871

Table 4: Comparsion of results

With more perceived Wi-Fi it's more likely not to be formed intersections between the polygons. In this case to get results, the tolerance limit needs to be increased: the results will be distorted. There is still room for improvement.

The farthest points are loaded with the highest error, this error still needs improvement. Furthermore, one shall avoid in the future that the intersection of other common Wi-Fi zones might end up with empty set.

3. Community map

Requirements and definition

The most important aim is to create a map server which works with the client (browser interface)and doesn't need any GIS software on the client side. The system should be able to search and visualize rooms in the building and POI (Point of Interest), route planning and positioning.

The system has more facilities to determine position: Using GPS infrastructure as well as Wi-Fi infrastructure. The user can add some new information to the map which would result a complete and detailed system.

Installation of the infrastructure

OSgeo4w

OSgeo4w is a binary distribution of a broad set of open source geospatial software. To establish server environment on a Windows box the OSGeo4w software package is suitable. OSGeo4w contains the Apache Web Server (Version 2.2), PHP programming language (Version 5.2), MapServer (Version 6.4) as well as QGIS open source desktop GIS software (Version 2.2).

PostgreSQL, PostGIS

After creating the development of the server environment the preparation of a database will take place. PostgreSQL (Version 9.3.) is a powerful, open source object-relational database system which can store all data used in this project.

PostGIS extends the functionality of PostgreSQL to the GIS area. It is a project which adds support for geographic objects in PostgreSQL, allowing it to be used as a spatial database for geographic information systems.

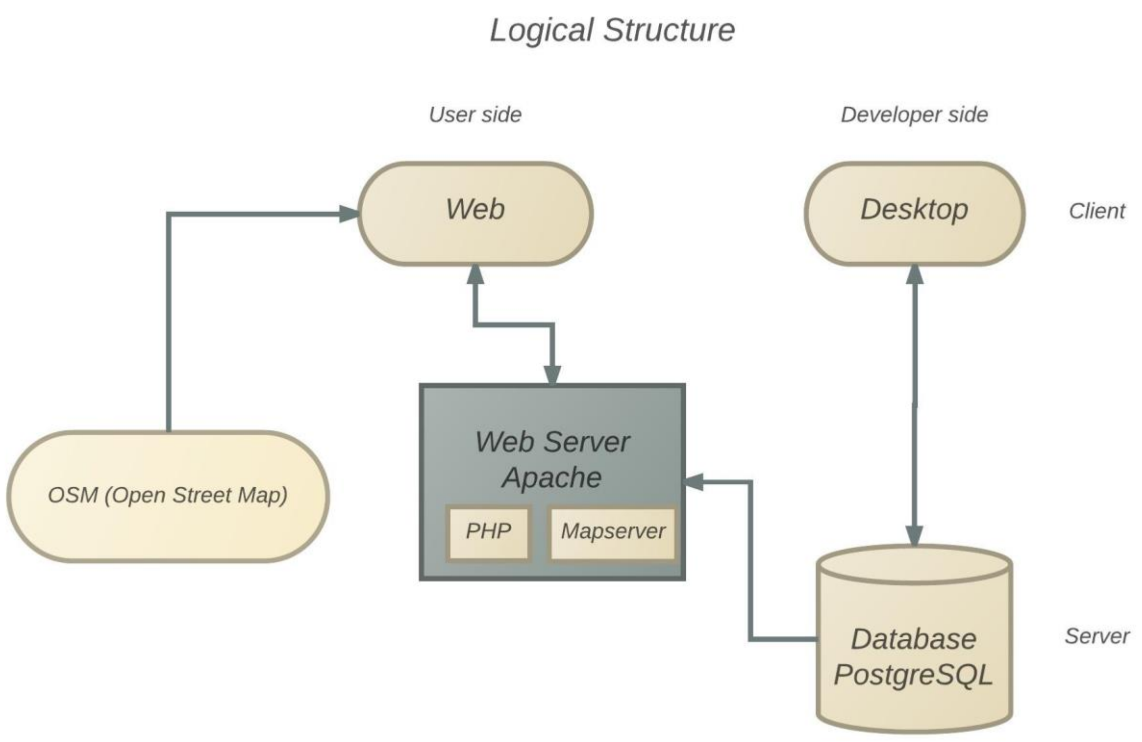


Figure 8: Logical Structure of the Community map

Fig 6 presents how this map system works. The logical structure has two principal sections: the first part (developer side) includes a database which apply Data and edit each maps. This intent can be reached by QGIS (Open Source Geographic Information System) which is user friendly and supports numerous vector, raster, and database formats and serves GIS analysis functions.

The second part of this map structure (user side) has a web-site which is the main goal of this task. These components only work together. The Database (PostGIS and PostgreSQL) contact with both (developer and user) sides. While editing in QGIS the various geometry and descriptive data were saved into the Data Base. The data can be recovered via a web interface.

Open layers

Web-based, map-display client which visualizes, moves, zoom etc., different map formats from different data Source in browser interface. Open layers is a JavaScript library which is similar to Google Maps.

This library is open source and can be customizable easily. To display the community map and implement the features this client was used for.

Creating the Database

Object catalog

A wide variety of objects can be used for creating a community map. There are surfaces like buildings, rooms, laboratories or parking. There are also lines that can be corridors. The point represents lots of object, especially basic points could be useful in this study.

Map resources

Most of the maps which used for this study are vector-based[8]. Furthermore raster-based maps[9] were used which have georeferenced in QGIS.

Map Server

If the user opens the web-site, the request will be served by an Apache Web Server on the server side. It has a connection to the Web Map Server which converts geometric data to pictures. This represents how the user gets information: geometric data is in picture format. The descriptive data are served by the Web Feature Server.

WMS allows the use of data from different servers. It enables the creation of a network of Map Servers, from which clients can build their own customized maps.

[8] *Csábi Zoltán: BME-South (2012), Kovács Géza: BME-North (2012), Perstic Tímea: BME-North (2009).*
[9] *Building 'K', www.sc.bme.hu/content/13/BME_K_epulet.pdf*

Queries

Creating the Map Server needs Map Server configuration file (*.map, in this study bme.map). (Appendix: 2.1. Map Server configuration file).

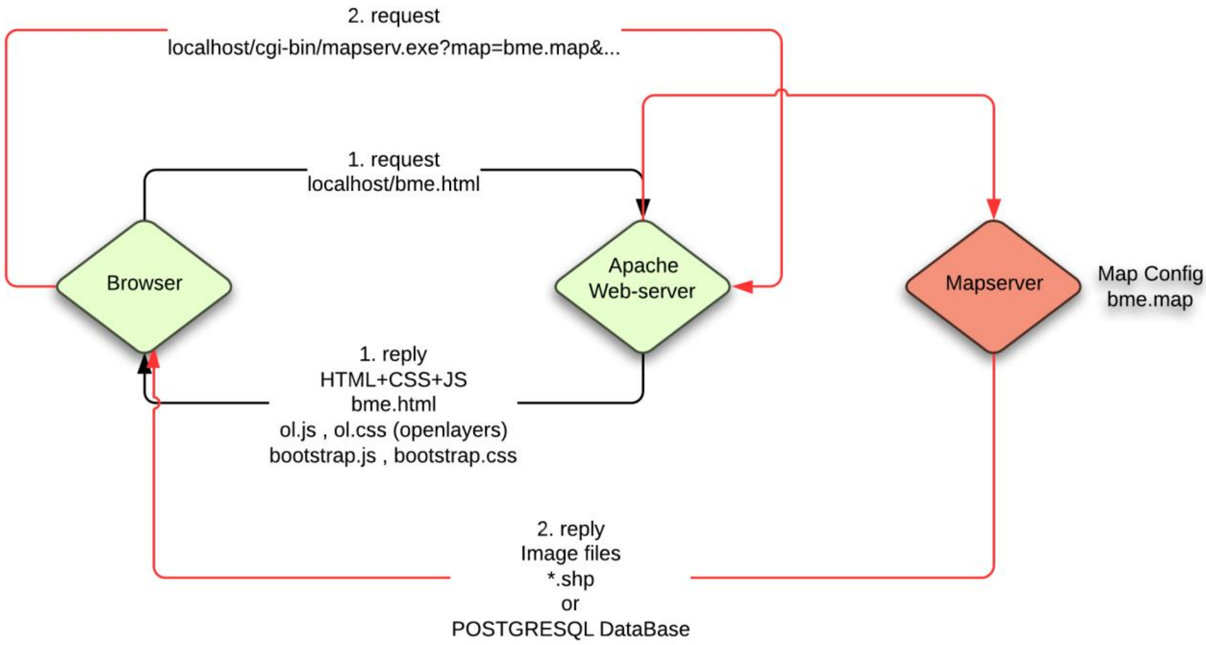
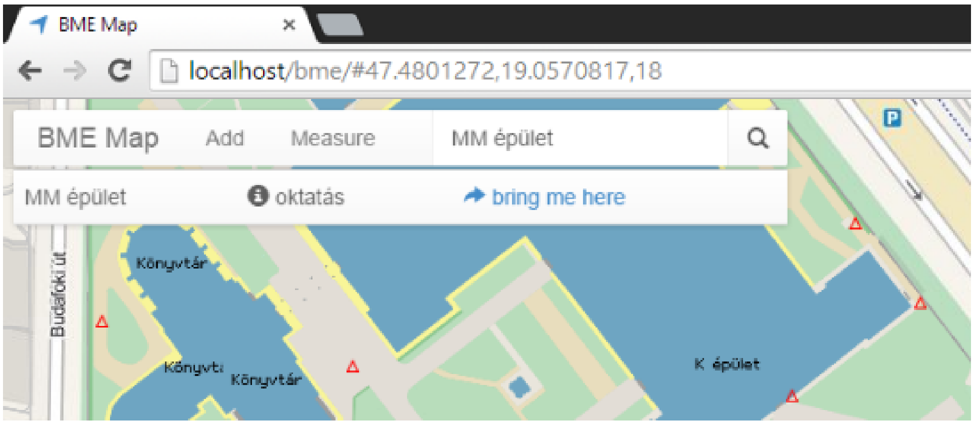


Figure 9: Queries in WMS

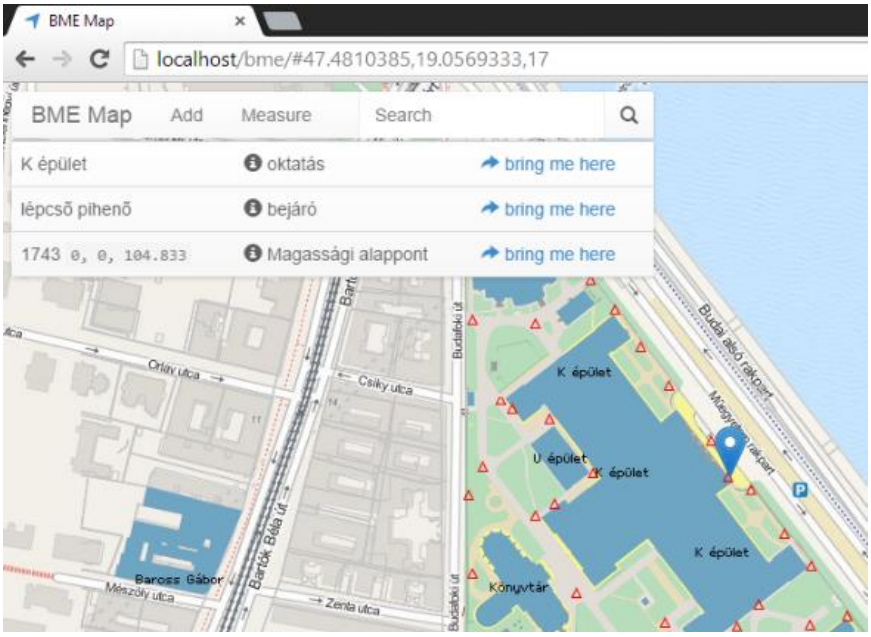
Implementation of features

The features was prepared by html file with style sheets (bootstrap) and java scripts (jQuery and bootstrap, Open Layers (Appendix: 2.2. Open layers initialization and configuration)). The PHP script performs the processing of data. The implemented features look as follows:

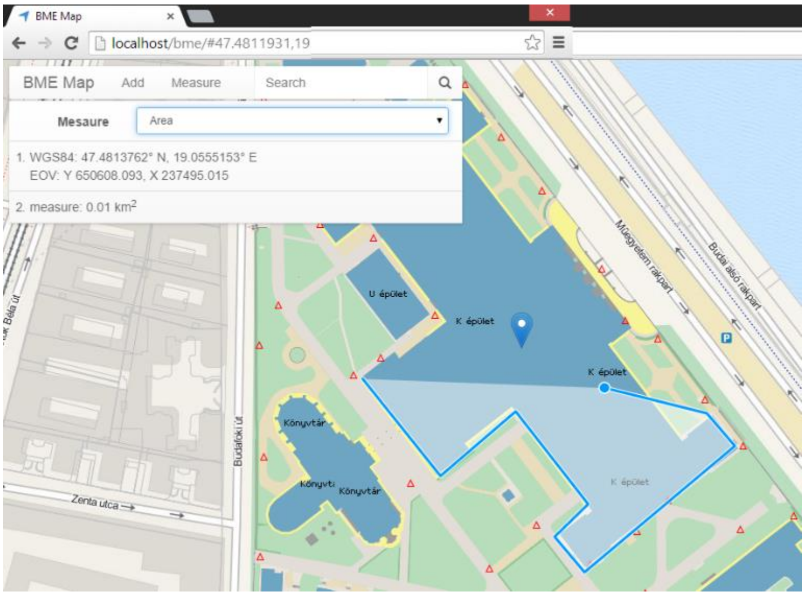
- Search: PHP script receives the text to find. This script opens a database connection and searches for the text in the database table and returns the coordinates of the found object. The point will be marked on the map.



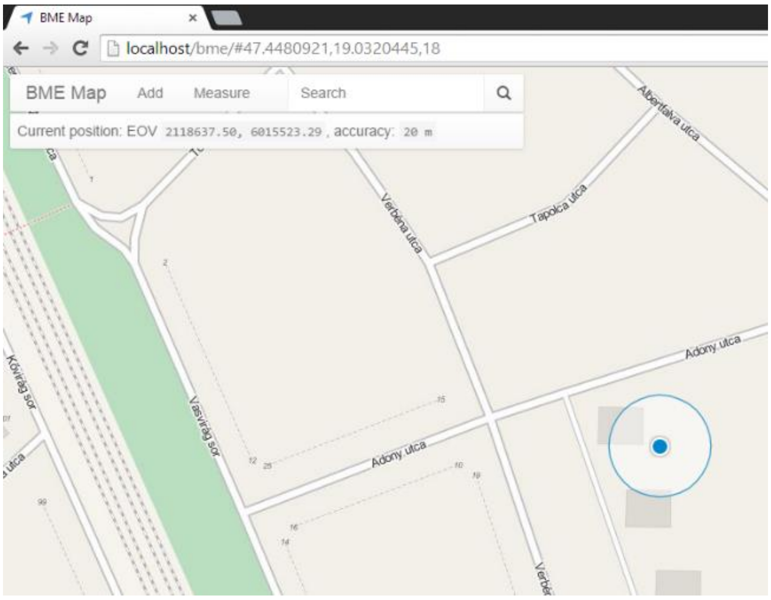
- Information query: clicking on the map information about the nearby objects can be list. The Map Server receive the coordinates of the clicking and gives back a template file including information. This method is the 'get feature info mode' of the WMS server.



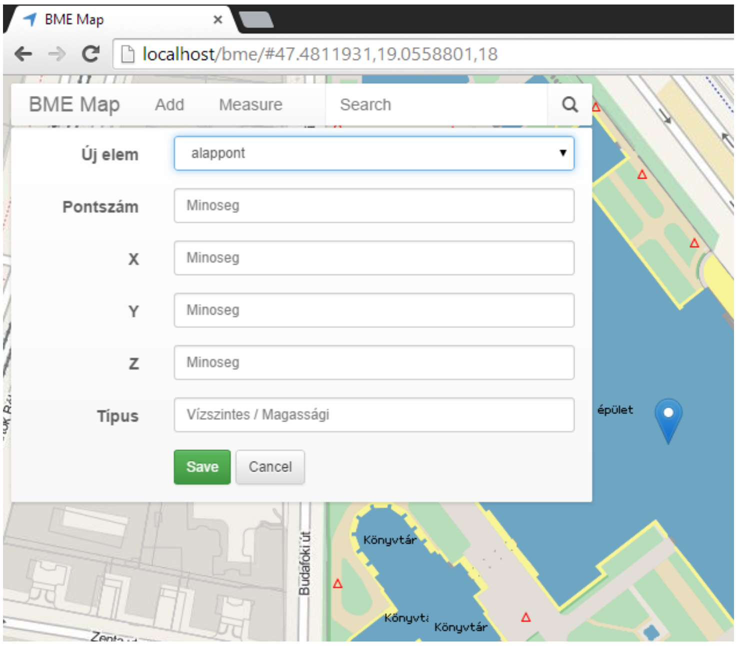
- Measure: works with built-in function which makes possible to measure distances, areas and coordinates.



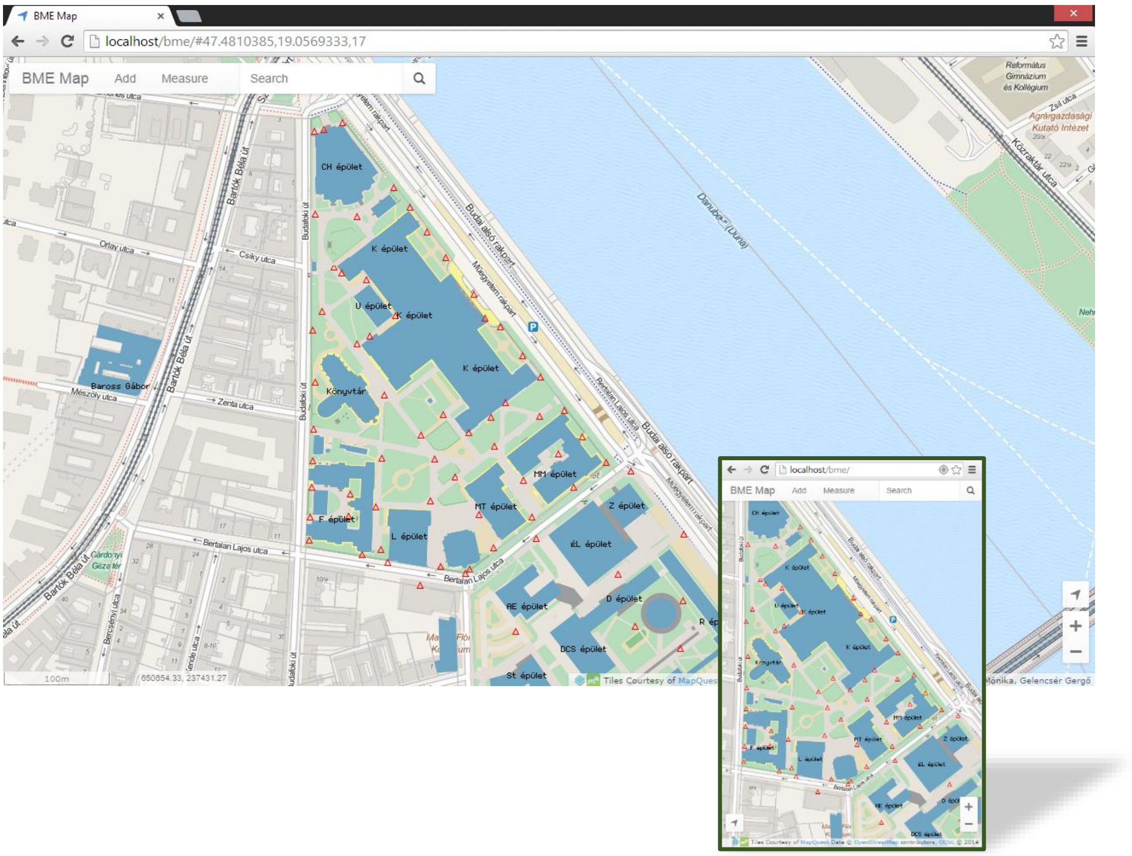
- Positioning: based on the positioning system of the browser interface. The geolocation can implemented by a mobile device (with GPS or Wi-Fi infrastructure) or by Personal Computer (with Internet connection)



- Community feature: it is also possible to record data on the map. PHP script records the form elements into the database.



- The map system is user-friendly through mobile apps as well as on web interface:



4. Summary

The aim of this study was to explore a method for indoor-positioning, which can be used for navigation within a building. Several possibilities were available to produce indoor-positioning software: Creating Fingerprinting model by using radio technologies, using acceleration sensors or using non-radio technologies, which include inertial and magnetic systems.

In our study, the Wi-Fi Signal Strength's role in positioning methods was applied. First of all, a measuring method was created, which was proven to be applicable and reliable. Thereafter, static experiment was made to calculate the stability and the standard deviation of Signal Strengths. After this, a fingerprinting model was built, which was used for indoor-positioning. This model was created with the help of Grid Interpolation and with Intersection from scattered polygons as well. Each method (based on RSSI) was proven to be effective. In accordance with this model, we created a community map: positioning, measuring and searching functions were implemented. This community map allows positioning such as indoors and outdoors.

The main goal of this study was to reach appropriate accuracy: in our case, this meant +/- 10 meters. The average accuracy was approximately +/-18 meters. This approach can be used for indoor positioning; although one might improve the accuracy by performing refined the measurements.

In case of future studies, our Fingerprinting model could be improved: it is possible to combine positioning methods, to try using Voronoy cells, or just improve our algorithms.

New investigations could complete this study: To follow the change in the Wi-Fi positions and new APs, the measurements must be repeated regularly and it could give a more complete picture about this method.

References

[1] Carr, Joseph: *RF Components and Circuits*. Newnes. pp. 45–46. ISBN 978-0750648448, 2002.

[2] Eduardo Navarro, Benjamin Peuker, Michael Quan, Dr. Christopher Clark, Dr. Jennifer Jipson: *Wi-Fi Localization Using RSSI Fingerprinting.* California Polytechnic State University 1 Grand Avenue; San Luis Obispo, CA 93405; USA

[3] GDAL - Geospatial Data Abstraction Library, *http://gdal.org/gdal_utilities.html*

[4] Geospatial World \ August 2014.

[5] Hui Liu, Student Member, IEEE, Houshang Darabi, Member, IEEE, Pat Banerjee, and Jing Liu*: Survey of Wireless Indoor Positioning Techniques and Systems*. IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS—PART C: APPLICATIONS AND REVIEWS, VOL. 37, NO. 6, NOVEMBER 2007.

[6] Jean Tourrilhes: *http://linux.die.net/man/8/iwconfig ,* Linux Man Page

[7] Joo-Yub Lee, Cheal-Hwan Yoon, Hyunjae Park, and Jungmin So Dept. of Computer Engineering, Hallym University, Gangwon-do, Korea: *Analysis of Location Estimation Algorithms for Wifi Fingerprint-based Indoor Localization*. ASTL Vol. 19, pp. 89 - 92, 2013.

[8] Map Server documentation: *http://mapserver.org/documentation.html*

[9] Open layers API documentation: *http://openlayers.org/en/v3.0.0/apidoc/*

[10] Panagiotis (Peter) A. Vretanos: *Geography Markup Language (GML) simple features profile Copyright.*2006

[11] Post GIS documentation:*http://postgis.net/docs/manual-2.1/*

[12] "Web Map Service". Open Geospatial Consortium. Retrieved 2009-03-23.

Appendix

1.1. Linux Bash Script to collect Wi-Fi data

```
#!/bin/bash
echo ""
echo "*** BME MAP - COLLECT WIFI TOOL ***"
echo "This program requires gawk parser."
echo "to install: sudo apt-get install gawk"
echo ""
i=1
user=`whoami`
if [ -z $1 ]
then
  echo "Usage: $0 output.csv"
elif [ $user = "root" ]
then
  while [ 1 -gt 0 ]
  do
    # iwlist command lists wifi data on linux
    # gawk parser collect this, and write in to output file ($1 parameter)
    iwlist wlan0 scan | gawk -f collect_wifi.awk >> $1
    i=$(( $i + 1 ))
    echo $i
    sleep 1
  done
else
  echo $user
  echo "Run as root!"
fi
```

1.2. AWK Script to analyze Wi-Fi data

```
BEGIN { FS="[ =:/]";
  name = "";
  strength = -1000;
  address = "000000000000";
  timestamp = strftime("%H:%M:%S");
}
{ line = $0;

  # trim white spaces
  gsub(/^[ \t]+/, "", line);
  gsub(/[ \t]+$/, "", line);

  # replace neighboring spaces with single space
  gsub(/ +/, " ", line);

  n = split(line, fields);
  if (match(line, /Address: /)) {
    address = "";
    for (i = n - 5; i <= n; i++) {
      address = address fields[i];
    }
  }
  if (match(line, /^Quality=/)) {
    strength = fields[6];
  }
  if (match(line, /^ESSID:/)) {
    # remove quotes
    name = fields[2];
    printf "%s;%s;%d;%s\n", name, toupper(address), strength, timestamp;
    name = "";
    strength = -1000;
    address = "000000000000";
  }
}
```

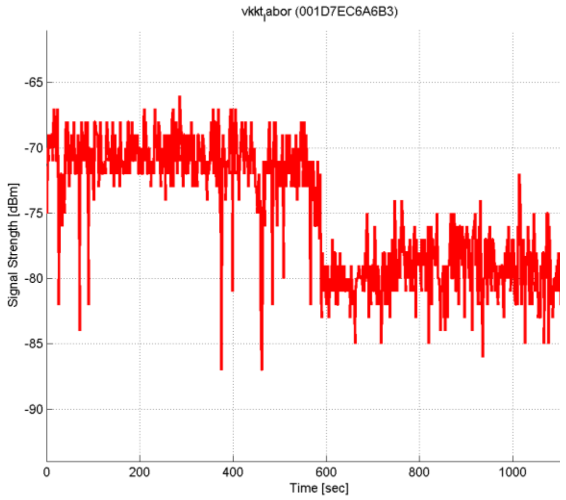
1.3. SQL Script to import coordinates and Wi-Fi data to relational database

```
-- create tables
DROP TABLE IF EXISTS :pointtable;
CREATE TABLE :pointtable( y double precision, x double precision, timestamp integer
NOT NULL );
DROP TABLE IF EXISTS :wifitable;
CREATE TABLE :wifitable( name character varying, address character(12), strength
integer, timestamp integer NOT NULL );
-- import csv files
COPY :pointtable(y, x, timestamp)
FROM :pointpath
WITH DELIMITER ';'
CSV HEADER;
COPY :wifitable(name, address, strength, timestamp)
FROM :wifipath
WITH DELIMITER ';'
CSV HEADER;
-- create indexes
CREATE INDEX ON :pointtable( timestamp );
CREATE INDEX ON :wifitable( timestamp );
CREATE INDEX ON :wifitable( address );
-- remove empty and duplicated records in same sec
DROP VIEW IF EXISTS wifi;
CREATE VIEW wifi AS
SELECT timestamp, name, address, avg(strength) AS strength
FROM :wifitable
WHERE address <> '000000000000'
GROUP BY timestamp, address, name
ORDER BY timestamp, address;
-- join tables
DROP TABLE IF EXISTS :resulttable;
SELECT p.y, p.x, w.address, w.name, w.strength
INTO :resulttable
FROM :pointtable AS p, :wifitable AS w
WHERE p.timestamp = w.timestamp;
-- or correlated: abs(p.timestamp - w.timestamp) = (SELECT min(abs( p.timestamp -
w2.timestamp )) FROM :wifitable AS w2);
-- create geometry
ALTER TABLE :resulttable ADD COLUMN gid serial PRIMARY KEY;
ALTER TABLE :resulttable ADD COLUMN geom geometry(POINT,23700);
UPDATE :resulttable SET geom = ST_SetSRID(ST_MakePoint(x,y),23700);
CREATE INDEX ON :resulttable USING GIST(geom);
-- statistics - point loss
DROP VIEW IF EXISTS wifi_chk_all;
CREATE OR REPLACE VIEW wifi_chk_all AS
SELECT count(*) AS all_num
FROM :pointtable p;
DROP VIEW IF EXISTS wifi_chk_join;
CREATE OR REPLACE VIEW wifi_chk_join AS
SELECT count(*) AS join_num
FROM :pointtable p, ( SELECT DISTINCT wifi."timestamp" FROM wifi) w
WHERE p."timestamp" = w."timestamp";
SELECT round((1-(j.join_num::float/a.all_num))::numeric,4) AS loss FROM
wifi_chk_join AS j, wifi_chk_all AS a;
-- statistics - number of distinct APs
SELECT DISTINCT address FROM wifi;
```

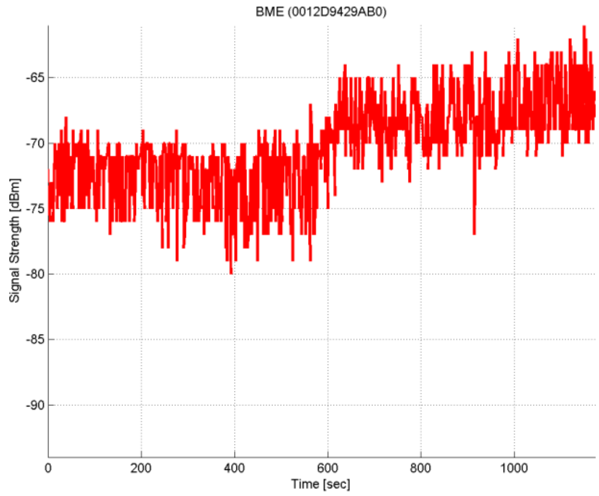
1.4. SQL Script to create buffer zone

```
-- create buffer zones
CREATE TABLE :buffertable AS
SELECT address, ST_Union(ST_Buffer(geom, 3.0)) AS geom
FROM :sourcetable
GROUP BY address;
ALTER TABLE :buffertable ADD COLUMN gid serial PRIMARY KEY;
CREATE INDEX ON :buffertable USING GIST(geom);

-- create polygon table
DROP TABLE IF EXISTS :resulttable;
CREATE TABLE :resulttable (
  address character(12),
  strength double precision,
  geom geometry(Polygon,4237)
);

-- vektorize rasters

INSERT INTO :resulttable (
  SELECT :raster AS address, val AS strength, geom
  FROM (
       SELECT (ST_DumpAsPolygons(ST_MapAlgebra(rast, NULL, 'ceil([rast])'::text,
0::double precision))).*
       FROM :rastertable
  ) AS q
);

ALTER TABLE :resulttable ADD COLUMN gid serial PRIMARY KEY;
CREATE INDEX ON :resulttable USING GIST(geom);
```

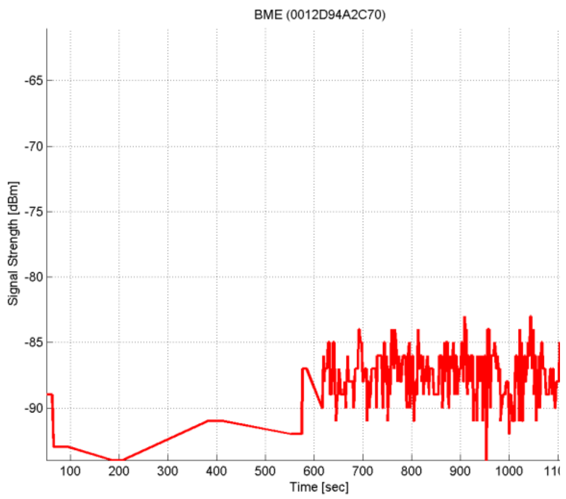
1.5. Typical Signal Strength fluctuations in the hall



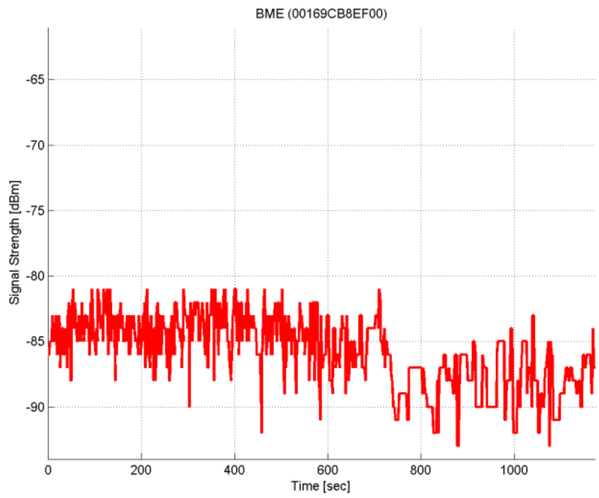
Vkkt_labor (00:1D:7E:C6:A6:B3) Signal Strength fluctuation



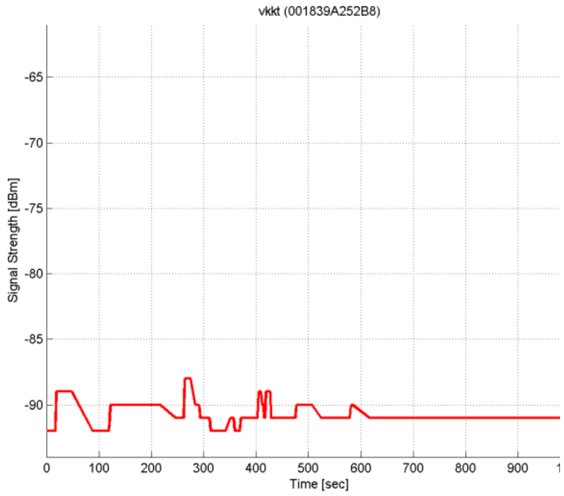
BME (00:12:D9:42:9A:B0) Signal Strength fluctuation



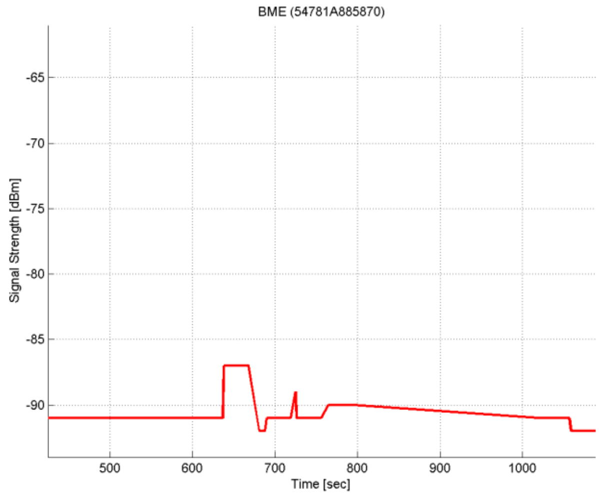
BME (00:12:D9:4A:2C:70) Signal Strength fluctuation



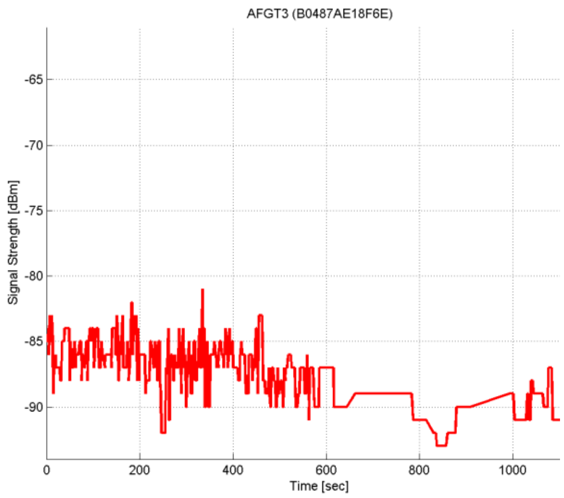
BME (00:16:9C:B8:EF:00) Signal Strength fluctuation



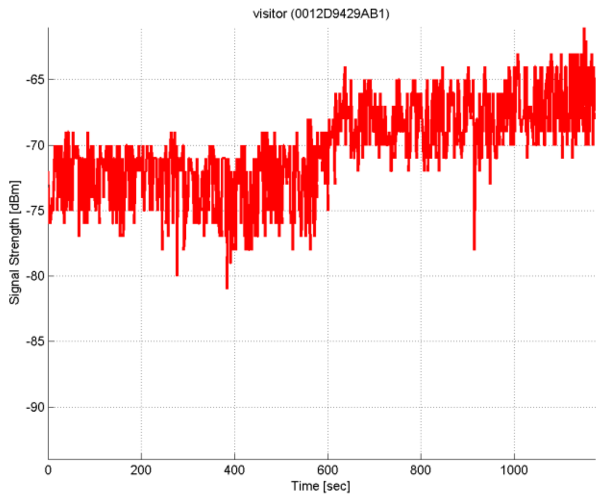
VKKT (00:18:39:A2:52:B8) Singal Strenth fluctuation



BME (00:12:D9:4A:2C:70) Signal Strength fluctuation

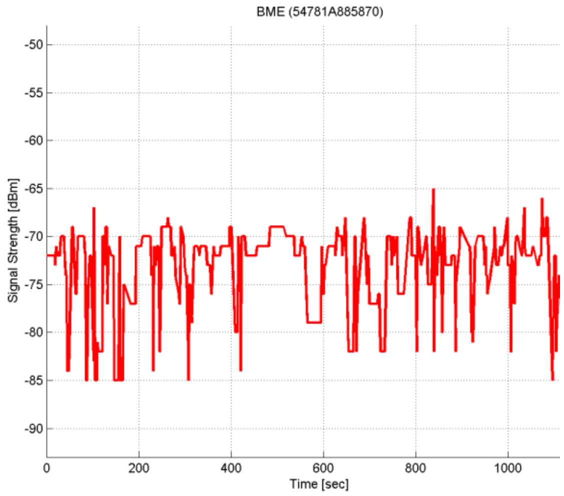


AFGT3 (B0:48:7A:E1:8F:6E) Signal Strength fluctuation

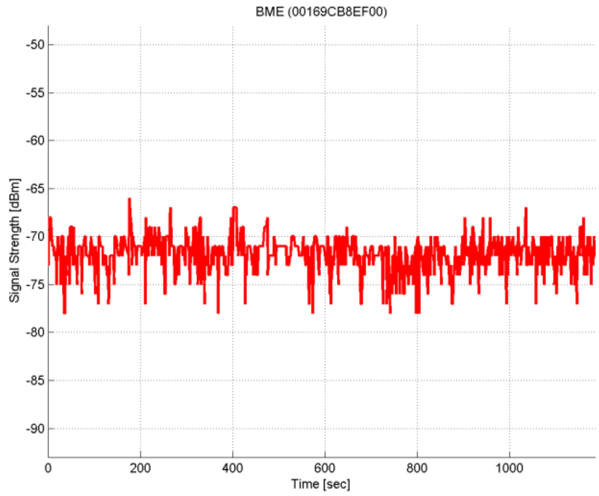


Visitor (00:12:D9:42:9A:B1) Signal Strength fluctuation

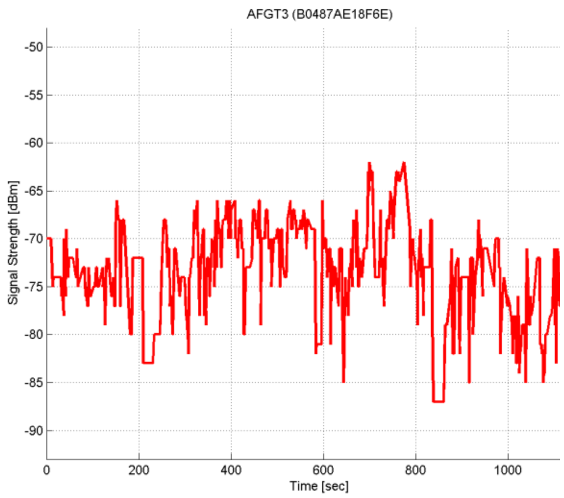
1.6. Typical Signal Strength fluctuations in the lab



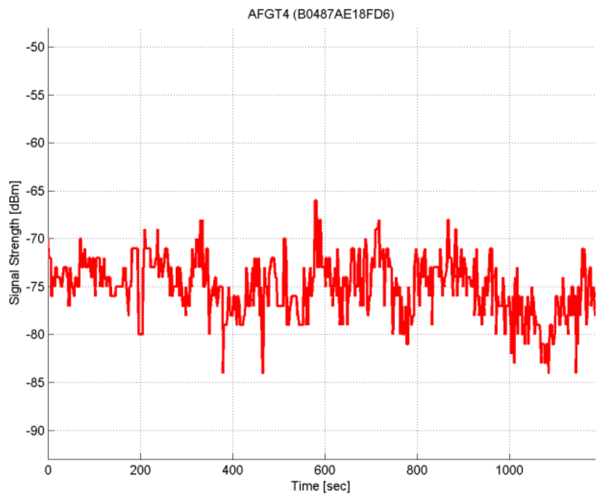
BME (54:78:1A:88:58:70) Signal Strength fluctuation



BME (00:16:9C:B8:EF:00) Signal Strength fluctuation



AFGT3 (B0:48:7A:E1:8F:6E) Signal Strength fluctuation



AFGT4 (B0:48:7A:E1:8F:D6) Signal Strength fluctuation

1.7. PHP script for the positioning query

```
<?php
include('db.php'); // database connection

// INPUT ------------------------------
if (!(isset($_GET['mode']) and !empty($_GET['mode'])))
  $_GET['mode'] = 'raster';
if (!(isset($_GET['address']) and isset($_GET['strength']))
and !empty($_GET['address']) and !empty($_GET['strength']))) {
print("Usage: wifi.php?address=0011223344AA,... &strength=-62,... &mode=
[raster|image|polygon|scatter] optional &tolerance=3 optional");
exit();
}
$address = explode(',', $_GET['address']);
$strength = explode(',', $_GET['strength']);
$tolerance = (isset($_GET['tolerance']) ? $_GET['tolerance'] : 3); // dBm

// DO THINGS --------------------------
switch ($_GET['mode'])
{
case 'raster': // grid method with raster algebra
case 'image':  // same method, but the output is GeoTiff image

$selects = array();
    $intersection = array();
for ($i = 0; $i < count($address); $i++)
{
$intersection[] = 'R'.$i.'.rast';
$selects[] = "(SELECT ST_MapAlgebraExpr(rast, NULL, 'CASE WHEN ( [rast.val] <
".(intval($strength[$i])+$tolerance)." AND [rast.val] > ".(intval($strength[$i])-
$tolerance)." ) THEN 1 ELSE 0 END'::text, 0::double precision) AS rast FROM
wifi_0910_".strtoupper($address[$i])." WHERE rid = 1) AS R".$i;
}
while (is_array($intersection)) {
$max = count($intersection);
if ($max > 1) {
$intersection[0] = '(ST_MapAlgebra('.$intersection[0].', 1,
ST_Resample('.$intersection[1].', R0.rast), 1, \'[rast1]*[rast2]\'::text, NULL,
\'UNION\'::text))::raster';
unset($intersection[1]);
$intersection = array_values($intersection);
}
else
$intersection = ''.$intersection[0].'';
}
$selects = implode(', ', $selects);
$sql_image = 'SELECT ST_AsTiff(Q.the_rast) FROM (SELECT '.$intersection.' AS
the_rast FROM '.$selects.') AS Q';
$sql = 'SELECT ST_X(ST_Centroid(geom)) AS x, ST_Y(ST_Centroid(geom)) AS y,
ST_Area(geom) as area FROM (SELECT (ST_DumpAsPolygons(Q.the_rast)).* FROM (SELECT
'.$intersection.' AS the_rast FROM '.$selects.') AS Q) AS Q2 WHERE val = 1 ORDER BY
area DESC LIMIT 1';

break;

case 'polygon': // polygon intersection with predefined polygons from grid
case 'scatter': // scatter polygon intersection from point buffer zones
default:

$intersection = array();
$selects = array();
```

```
if ($_GET['mode']=='polygon') {
$table = 'wifi_polygons_0910';
        $geom = 'geom';
}
else {
$table = 'wifi_joined_0910';
$geom = 'ST_Buffer(geom, 0.5)';
}
for ($i = 0; $i < count($address); $i++)
{
$intersection[] = 'q'.$i.'.geom';
$selects[] = '(SELECT ST_Union('.$geom.') AS geom FROM '.$table.' WHERE strength <=
\''.(intval($strength[$i])+$tolerance).'\' AND strength >=
\''.(intval($strength[$i])-$tolerance).'\' AND address =
\''.strtoupper($address[$i]).'\') AS q'.$i;
      }
while (is_array($intersection)) {
if (count($intersection) > 1) {
$intersection[0] = 'ST_Intersection('.$intersection[0].','.$intersection[1].')';
unset($intersection[1]);
$intersection = array_values($intersection);
}
else
$intersection = ''.$intersection[0].'';
}
$selects = implode(', ', $selects);
$sql = 'SELECT ST_X(ST_Centroid(the_geom)) AS x, ST_Y(ST_Centroid(the_geom)) AS y,
ST_Area(the_geom) AS area FROM (SELECT '.$intersection.' AS the_geom FROM
'.$selects.') AS Q ORDER BY Area DESC LIMIT 1';

break;
}

// OUTPUT -----------------------------
if ($_GET['mode'] == 'image')
{
header('Content-Type: image/tiff');
$sql = $_DB->prepare($sql_image);
$sql->execute();
$data=$sql->fetchAll();
$data=fgets($data[0][0]);
$data=substr($data,2);
print(pack('H*', $data));
exit;
}
else {
header('Content-Type: application/json');
$sql = $_DB->prepare($sql);
$sql->execute();
$result = $sql->fetch();
print('{"y" = "'.($result['x']).'", "x" = "'.($result['y']).'", accuracy =
"'.(sqrt($result['area']/Pi())).'"}');
}

?>
```

2.1. Map Server configuration file

```
MAP # Start of map file --------------

  NAME "BME Map"
  EXTENT 650240 236000 650260 237780
#HD72/EOV
  SHAPEPATH "data"
  SIZE 800 600
  IMAGETYPE PNG24
  FONTSET "fonts/fonts.list"

  PROJECTION # -----------------------
    "init=epsg:23700" #HD72/EOV
  END # ------------------------------

  WEB # ------------------------------
    #ERROR "http://localhost/bme/"
    METADATA
      WFS_TITLE "BME Map"
      WFS_ABSTRACT "BME Map WFS
server"
      WFS_ONLINERESOURCE
"http://localhost/bme/map?"
      WFS_SRS "epsg:23700 epsg:4326
epsg:3857"

      WMS_TITLE "BME Map"
      WMS_ABSTRACT "BME Map WMS
server"
      WMS_ONLINERESOURCE
"http://localhost/bme/map?"
      WMS_SRS "epsg:23700 epsg:4326
epsg:3857"

      WMS_FEATURE_INFO_MIME_TYPE
"text/html"
      WMS_ENABLE_REQUEST "*"
      WMS_ATTRIBUTION_TITLE "(c) 2014
Fanczal Mónika, Gelencsér Gergő"
    END
  END # ------------------------------

  SYMBOL # ---------------------------
  NAME "triangle"
  TYPE vector
    POINTS
      0 4
      2 0
      4 4
      0 4
    END
  END # ------------------------------

  LAYER # ----------------------------
    NAME "alapterkep"
    DATA "alapterkep"
    STATUS OFF
    TYPE LINE
    TEMPLATE
"templates/alapterkep.html"
    METADATA
      WMS_TITLE "alapterkep"
      WFS_TITLE "alapterkep"
      'gml_featureid' 'ID'
      'gml_include_items' 'all'
    END
    CLASS
      STYLE
        COLOR 200 200 200
      END
    END
  END # ------------------------------

  LAYER # ----------------------------
    NAME "epuletek"
    DATA "epuletek"
    STATUS OFF
    TYPE POLYGON
    TEMPLATE "templates/epuletek.html"
    TOLERANCE 3
    DUMP TRUE
    METADATA
      WMS_TITLE "epuletek"
      WFS_TITLE "epuletek"
      'gml_featureid' 'ID'
      'gml_include_items' 'all'
    END
    LABELITEM "MINOSITES"
    CLASS
      STYLE
        COLOR 110 165 194
      END
      LABEL
        FONT "volter"
        TYPE TRUETYPE
       ENCODING UTF-8
        POSITION CC
        SIZE 7
       BUFFER 2
        MINFEATURESIZE 40
        MINDISTANCE 200
        COLOR 0 0 0
      END
    END
  END # ------------------------------

  LAYER # ----------------------------
    NAME "epitmenyek"
    DATA "epitmenyek"
    STATUS OFF
    TYPE POLYGON
    TEMPLATE
"templates/epitmenyek.html"
    TOLERANCE 3
    METADATA
      WMS_TITLE "epitmenyek"
      WFS_TITLE "epitmenyek"
      'gml_featureid' 'ID'
      'gml_include_items' 'all'
```

```
    END
    CLASS
      STYLE
        COLOR 157 157 157
      END
    END
  END # ------------------------------

  LAYER # ----------------------------
    NAME "burkolatok"
    DATA "burkolatok"
    STATUS OFF
    TYPE POLYGON
    TEMPLATE
"templates/burkolatok.html"
    TOLERANCE 3
    METADATA
      WMS_TITLE "burkolatok"
      WFS_TITLE "burkolatok"
      'gml_featureid' 'ID'
      'gml_include_items' 'all'
    END
    CLASSITEM "MINOSITES"
    CLASS
      EXPRESSION /zöld|díszkert|gyep/
      STYLE
        COLOR 184 221 187
      END
    END
    CLASS
      EXPRESSION
/sövény|virágoskert|tankert/
      STYLE
        COLOR 140 200 145
      END
    END
    CLASS
      EXPRESSION /^aszfalt.*$/
      STYLE
        COLOR 226 221 214
      END
    END
    CLASS
      EXPRESSION
/^beton.*$|^viacolor.*$|^térkő.*$|^lap
.*$/
      STYLE
        COLOR 233 229 220
      END
    END
    CLASS
      EXPRESSION
/^keramit.*$|^márvány.*$/
      STYLE
        COLOR 248 245 152
      END
    END
    CLASS
      EXPRESSION /./ #föld, murva,
zúzott kő
      STYLE
        COLOR 232 221 189
      END
    END
  END # ------------------------------

  LAYER # ----------------------------
    NAME "alappontok"
    DATA "alappontok"
    STATUS OFF
    TYPE POINT
    TEMPLATE
"templates/alappontok.html"
    TOLERANCE 3
    METADATA
      WMS_TITLE "alappontok"
      WFS_TITLE "alappontok"
      'gml_featureid' 'ID'
      'gml_include_items' 'all'
    END
    CLASS
      STYLE
        SYMBOL "triangle"
       SIZE 7
       COLOR 255 0 0
      END
    END
  END # ------------------------------

  # utcabutorok, novenyek, emlekmuvek

END # of Map File --------------------
```

2.2. Open layers initialization and configuration

```
// ------------ PROJECTION DEFINITIONS ---------------

// EOV DEFINITION
proj4.defs('EPSG:23700',"+title=Hungarian EOV EPSG:23700 +proj=somerc
+lat_0=47.14439372222222 +lon_0=19.04857177777778 +x_0=650000 +y_0=200000
+ellps=GRS67 +datum=HD72 +towgs84=57.01,-69.97,-9.29 +units=m +no_defs");


// --------------- VIEW DEFINITIONS ------------------

// DEFAULT PARAMETERS
var Center = [19.056944, 47.481111];
var Zoom = 17;

// GET PARAMETERS FROM URL (HELPS SHARING AND SAVING)
if (location.hash.length <= 25 && location.hash.length >= 24) {
  params = location.hash.substr(1).split(",");
  Center = [ parseFloat(params[1]), parseFloat(params[0]) ];
  Zoom = parseInt(params[2]);
}

// DEFINE THE GLOBAL VIEW
var view = new ol.View({
  center: ol.proj.transform(Center, 'EPSG:4326', 'EPSG:3857'),
  zoom: Zoom
});


// -------------- LAYER DEFINITIONS ------------------

// OPENSTREETMAP LAYER
var OSMLayer = new ol.layer.Tile({
  source: new ol.source.OSM()
});

// MAPQUEST LAYER
var MapQuestLayer = new ol.layer.Tile({
  source: new ol.source.MapQuest({layer: 'osm'})
});

// WMS LAYER - OWN MAPSERVER
var WMSLayer = new ol.layer.Tile({
  source: wmsSource = new ol.source.TileWMS({
    url: 'map?',
    params: {
      'LAYERS': 'alapterkep,epuletek,epitmenyek,burkolatok,alappontok',
        'TILED': true
    },
    serverType: 'mapserver',
    attributions: [new ol.Attribution({
      html: '&copy; 2014 Fanczal Mónika, Gelencsér Gergő'
    })]
  })
});


// -------------- INITIALIZE OPENLAYERS --------------

var map = new ol.Map({
  target: 'map',                      // map container <div id="map">
  view: view,                         // view definition
  layers: [MapQuestLayer, WMSLayer],  // layer definitions
  controls: ol.control.defaults({
      attributionOptions: ({         // attribution control
```

```
      collapsible: false
    })
  }).extend([
    new ol.control.ScaleLine({          // scale line
      units: 'metric'
    }),
      new ol.control.MousePosition({  // coordinate box
             projection: ol.proj.get('EPSG:23700'),
             coordinateFormat: ol.coordinate.createStringXY(2)
      })
  ])
});


// -------------- MARKER DEFINITIONS ----------------

var markerBalloon = new ol.Overlay({
  positioning: 'bottom-center',
  element: document.getElementById('marker_balloon'),
  stopEvent: false
});
map.addOverlay(markerBalloon);

var markerCircle = new ol.Overlay({
  positioning: 'center-center',
  element: document.getElementById('marker_circle'),
  stopEvent: false
});
map.addOverlay(markerCircle);


// -------------- EVENT LISTENERS --------------------

// MAP - PAN EVENT UPDATES THE URL
map.on('moveend', function(evt) {
  Center = ol.coordinate.format(
    ol.proj.transform(view.getCenter(), 'EPSG:3857', 'EPSG:4326'),
    '{y},{x},', 7
  );
  Zoom = view.getZoom();
  location.hash = Center + Zoom;
});

// MAP - CLICK EVENT - GetFeatureInfo
map.on('click', getFeatureInfo);
function getFeatureInfo(evt) {
  var coordinate = evt.coordinate;
  markerBalloon.setPosition(coordinate);

  var url = wmsSource.getGetFeatureInfoUrl(
         evt.coordinate,
         view.getResolution(),
         view.getProjection(),
         {'INFO_FORMAT': 'text/html'}
  );
  getAjaxContent(
         url,
         '<table class="table table-hover">',
         '</table>',
         '<tr><td colspan="3">No matches found</td></tr>'
   );
}

// NAVIGATION LINKS - CLICK EVENT
$('a[data-target="navbar-content"]').click(function(evt) {
  evt.preventDefault();
  if ($(this).attr('href').length) {
```

```
        getAjaxContent('php/'+$(this).attr('href').replace("#", "")+'.php');
    }
});

// SEARCH BUTTON - PRESS AND SUBMIT EVENT
$('#searchbtn').on('click', getSearchResult);
$('#search').on('submit', getSearchResult);
function getSearchResult(evt) {
    evt.preventDefault();
    var query = $('#search input').val();
    if (query.length >= 3) {
        getAjaxContent(
            'php/search.php?q='+query,
            '<table class="table table-hover">',
            '</table>',
            '<tr><td colspan="3">No matches found</td></tr>'
        );
    }
}

// BRING ME HERE BUTTON - CLICK EVENT
$(document).on('click', 'a[href="#gohere"]', function(evt){
    evt.preventDefault();
    var y = $(this).data('y');
    var x = $(this).data('x');
    if (typeof y != 'undefined' && typeof x != 'undefined') {
        var position = ol.proj.transform([x, y], 'EPSG:23700', 'EPSG:3857');
        view.setCenter(position);
        markerBalloon.setPosition(position);
    }
});


// ----------------- FUNCTIONS -----------------------

// AJAX LOADER FUNCTION
function getAjaxContent(url, before, after, defTxt) {
    before = (typeof before !== 'undefined') ? before : '';
    after = (typeof after !== 'undefined') ? after : '';
    defTxt = (typeof defTxt !== 'undefined') ? defTxt : '';

    $('#navbar-content').hide().html('<div class="content-row"><img
src="imgs/loader.gif" alt="" /> Requesting informations...</div>');

    $.ajax({
        url: url,
        cache: false
    })
    .done(function( html ) {
        if (html.length <= 1) html = defTxt;
        $('#navbar-content').html(before + html + after);
    })
    .fail(function( jqXHR, textStatus ) {
        $('#navbar-content').html(before + 'Request failed: ' + textStatus + after);
    });

    $('#navbar-content').show();
}


// ------------- END OF bme.js FILE -----------------
```