Önálló laboratórium beszámoló

Dolgozat címe: Földrajzi adatbázis GSM alapú helymeghatározáshoz

Konzulens(ek) neve: Dr. Lukács Gergely és Tihanyi Attila

(Külső cég neve:

címe:…….)

A Hallgató a kitűzött feladatot megfelelő színvonalon és a kiírásnak megfelelően

teljesítette nem teljesítette

_____

Konzulens aláírása

Hallgató neve: ……..

Leadás dátuma: ……..……..

Tartalom

Ábrajegyzék:

Készítette: Vinis Ádám

Kelt: Gyúró, 2009.05.20.

Bevezetés

Az önálló laboratórium projectemen egy népes csapatban dolgoztam, ezért tevékenységemet nem kellett az alapoktól elkezdeni. Rengeteg adat, információ rendelkezésre állt már a témáról, amikor tavasszal csatlakoztam a csapathoz. Most röviden bemutatom a csapat tagjait, és az eddigi eredményeket.
A csapatban több kiváló oktató is részt vett. Dr. Takács György, Dr. Tihanyi Attila, Dr. Lukács Gergely, és Feldhoffer Gergely. A csapat diáksága több felsőbb éves hallgatóból állt, akik már régóta foglalkoztak a témával, és 3 BSC-s hallgató lett újonnan tagja, köztük én is.
Mivel nekünk, újoncoknak az alapoktól kellett kezdenünk, a kutatási munkánk eleje az eddig elért eredmények megismerésével, a technológiai kihívások megismerésével kezdődött.
Tehát a cél egy működő helymeghatározó rendszer kifejlesztése, amely épületen belül és kívül a piacon (jelenleg, vagy a közeljövőben) kapható mobiltelefonok használatával megfelelő pontosságú adatot biztosít a pozíciónkról. Ezzel elkerülhető a GPS készülékek megvásárlása, a mobiltelefon manapság majdnem minden ember zsebében megtalálható, illetve erre a szolgáltatásra további értékes szolgáltatások építhetők. Például a családtagok, barátok helyzetét figyelő, folyamatosan frissítő rendszer, amely a mobiltelefonokat használja a pozíciónk detektálására, ismerőseink pozíciójának egymás közti adatátvitelére.
A technológiai megoldások kutatása során felmerült, hogy a helymeghatározáshoz jól működő, földrajzi koordinátákat jól kezelő adatbázis szükséges és erre a célra legjobb egy földrajzi adatbázis-kezelő rendszer lehet. Egy ilyen adatbázis gyorsan képes nagy mennyiségű adatokat kezelni és feldolgozni a saját földrajzi célokra fejlesztett beépített függvényei segítségével, amellyel egy hagyományos adatbázis-kezelő rendszer nem rendelkezik.

1. Háttér

2.1. Mobil helymeghatározás

A csapat egyik korábbi publikációjából vett bevezető szöveg jól leírja a feladatot:
„A technika fejlődésével, és a lehetséges szolgáltatások bővülésével egyre nő az igény arra, hogy minél pontosabban, gyorsabban, és olcsóbban meg tudjuk határozni saját, vagy esetleg társunk tartózkodási helyét. Erre a műholdas rendszerek mellett a GSM hálózat is alkalmat kínál. A GSM hálózat működési tulajdonságainak kihasználásával szolgáltatás építhető, amely alkalmas helymeghatározásra. A felépített rendszer pontossága városi környezetben a GPS-hez mérhető. Ezzel alkalom nyílik személyi, gyalogos navigációra, akár beltéri környezetben is, ahol a műholdas helymeghatározás nem működik.
A rendszer helymeghatározásra alkalmas GSM hálózat felhasználásával működik, de a szokásos rendszerek hátrányinak kiküszöbölését is megvalósítja. A GPS hálózat üzemeltetési bizonytalansága, és a GPS készülékek magas ára nem okoz gondot, hiszen a kialakított rendszer GPS-hez hasonló pontosságú adatokat GPS felhasználása nélkül biztosítja. GSM szolgáltató közreműködése nélkül igénybe vehető, így biztosan nem tartozik hozzá számla és fizetési kötelezettség. A GSM szolgáltató nem is vesz részt a folyamatban, hiszen a szolgáltatás felépítése ezt nem teszi szükségessé. A helymeghatározás során a GSM hálózat pillanatnyi sugárzási karakterisztikáját

használjuk fel, és az adatok szolgáltató által történő kismértékű megváltoztatása csak némi pontosság romlást eredményez, de nem hiúsítja meg a helymeghatározást.
Az elmúlt pár évben rohamosan nőt a világon a mobiltelefonok száma (több okos telefont adnak el egy órában, mint ahány gyermek születik), és Magyarországon 2008. július végén 11.601.000 mobil előfizető volt (forrás: Nemzeti Hírközlési Hatóság). A mobil készülékek egyre okosabbak, több és több szolgáltatást nyújtanak a telefonálási lehetőségen túlmenően. A szolgáltatóknak növekvő számú mobiltelefon kiszolgálására alkalmas hálózatot kell biztosítania, tehát a forgalom növekedése következtében egyre több, és nagyobb kapacitású bázisállomást kell építeniük. A bázisállomások sűrűségének növekedése azt is jelenti, hogy lakott területen általában sok bázisállomás adókörzetében vagyunk, ami lehetőséget biztosít a helymeghatározásra." (Részlet a Bányai Balázs, Feldhoffer Gergely, Tihanyi Attila által a GSM alapú helymeghatározásról írt cikkből.)

2.2. Földrajzi adatbázisok

2.2.1. Bevezetés

A térinformatika (angolul GIS (Geographic Information Systems)) többféle megközelítése:
A térinformatika tudomány, az informatika egy speciális ága, olyan informatika, amelyben az információ alapjául szolgáló adatok földrajzi helyhez köthetők. A térinformatika által megválaszolandó tipikus kérdések:
Mi van ott? Hol van? Mi történik akkor, ha ...? *(Márkus Béla: Bevezetés a térinformatikába)*
Olyan rendszer, mely olyan adatokat gyűjt, tárol, ellenőriz, integrál, kezel, elemez és megmutat, amelyek térbelileg a Földhöz kötöttek. *(Chorley, 1987)*
Automatizált rendszer, mely térbeli adatokat gyűjt, tárol, visszakeres, elemez és megmutat. *(Clarke, 1990)*
Információs rendszer, amit olyan adatokkal való munkára terveztek, amelyek térbeli vagy földrajzi koordinátákkal vannak összekapcsolva. Más szavakkal a GIS egyrészt egy speciális adatbank rendszer, speciális térbeli vonatkozásokkal rendelkező adatokkal, másrészt egy utasításkészlet, amely ezekkel az adatokkal dolgozni képes. *(Star and Estes, 1990)*
A GIS egyidejűleg teleszkópja, mikroszkópja, számítógépe és xerox-gépe a térbeli adatok elemzésének és szintézisének. *(Abler, 1988)*
A GIS egy megfelelő hardver környezetben működő olyan szoftver együttes, amely eljárásai révén támogatja a területfüggő adatok nyerését, kezelését, manipulálását, analízisét, modellezését és megjelenítését komplex tervezési és működtetési feladatok megoldása érdekében. (*David Rhind**)***

Rövid **bevezetés** arról mi az a **földrajzi információs** rendszer:
A földrajzi információs rendszer egy olyan számítógépes rendszer, melyet földrajzi helyhez kapcsolódó adatok gyűjtésére, tárolására, kezelésére, elemzésére, a levezetett információk megjelenítésére, a földrajzi jelenségek megfigyelésére, modellezésére dolgoztak ki. Nevezik térinformatikai, geoinformációs rendszernek vagy angolul rövidítve GIS-nek (Geographic Information Systems). A GIS egyetlen rendszerbe integrálja a térbeli és a leíró információkat – alkalmas keretet biztosít a földrajzi adatok elemzéséhez.

A geoinformatika rendkívül nagy jelentőséggel bír a természeti erőforrások kutatásában, állapotának figyelésében; a közigazgatásban; a földhasználati- és tájtervezésben; az ökológiai- és gazdasági összefüggések feltárásában, a döntéshozásban; ugyanakkor a közlekedési-, szállítási-, honvédelmi-, piackutatási feladatok megoldásában; a szociológiai-, társadalmi összefüggések vizsgálatában; a település-fejlesztésben és a létesítmény-tervezésben. A geoinformatikában összefonódik a több ezer évre visszatekintő térképészet, a pár száz éves földtudományok és a pár évtizedes múlttal rendelkező számítástechnika.

A hálózatok terjedésével egyre nagyobb hangsúlyt kap az információk elérését, továbbítását szolgáló szerep. Alkalmazási oldalról a GIS egy eszköz a térkép használat, pontosabban a földrajzi adatok használatának fejlesztésére. A GIS lehetőséget ad nagyszámú helyzeti és leíró adat gyors, együttes, integrált áttekintésére és elemzésére. A GIS felépítésében, tartalmában, az alkalmazott hardver és szoftver tekintetében, a felhasználói környezetet illetően nagyon eltérő formákban jelenik meg.

A hagyományos adatbázis-kezelő rendszereket olyan eszközökkel lehet kiterjeszteni, amellyel hatékonyan lehet térbeli adatokat tárolni, és hatékonyan lehet térinformatikai adatokon lekérdezéseket végrehajtani. Ilyen kiterjesztése az Oracle-nek a Spatial modul, a PostgreSQL-nek a PostGIS, ezen túl az ESRI SDE több különböző adatbázis-kezelővel is képes együttműködni.
Az adatbázis-kezelők a térinformatikai adatokat, azaz a geometriai objektumokat több különböző típusba sorolják: pont, törött vonal, polygon, ugyanezen típusokból alkotott halmazok, vagy általános geometriai halmaz. Ezen túl több adatbázis-kezelő több további típust is definiál, például kör, ellipszis vagy spline görbék. Ezen túl általában a 2D modellen túllépve lehetőség nyílik 2,5D adatokat tárolni, amivel magassági értékeket is tárolhatunk az objektumok pontjaihoz.
Azonban a tárolás nem elegendő egy hatékony térinformatikai rendszer megvalósításához, hatékonyan kell a lekérdezéseket is kezelnie az adatbázis-kezelőknek. Ehhez általában a geometriai objektumok befoglaló téglalapja szerint egy R-fa alapú indexet épít fel. Ezen keresőfa segítségével egyes lekérdezéseket gyorsan tudunk végrehajtani.

További lényeges tulajdonsága a térinformatikai adatbázisoknak az, hogy térbeli objektumokat georeferáltan tárolják. Ehhez minden objektumhoz hozzárendelnek egy meghatározott vetületi rendszer azonosítót (SRID). Ennek segítségével képesek arra, hogy különböző vetületi rendszerben lévő objektumokat helyesen kezeljék, vagy a tárolt vetületi rendszertől eltérő rendszerben kaphassuk vissza a lekérdezésünk eredményét.
A geometria típusokat (pont, vonal, poligon) az adatbázis saját formátumba kódoltan tárolja, ami nem olvasható, nem feldolgozható az ember számára közvetlenül. Ezért kiolvasásnál egy függvénnyel alakítjuk vissza a feldolgozásra alkalmas formára az adatokat. Erre a saját reprezentációra azért van szüksége a rendszernek, hogy kisebb helyfoglalással, jobban kereshetően tárolja el az adatokat.
A földrajzi adatbázisok kiválóan alkalmasak térképészeti feladatokra, alakzatok, sokszögek elhelyezkedésének vizsgálatára. Lehetséges például pontok sokszögbe foglalása, sokszögek egymáshoz való viszonyának vizsgálata (metszet,unió, stb.)

2.2.2. PostgreSQL és PostGIS

PostgreSQL (röviden: **PG**) korszerű objektumrelációs adatbázis-kezelő rendszer, amit a Berkeley Egyetem Számítástudományi tanszékének vezetésével fejlesztenek. A program támogatói között olyan híres szervezetek is részt vállalnak, mint az amerikai DARPA, ARO és NSF. A jelenlegi legújabb PG a 8.3-es változat támogatja az SQL92/SQL99 szabványokat, valamint lehetővé teszi az öröklődést, SQL-adattípusok, tárolt eljárások, kioldók (trigger), megszorítások, számlálók használatát, valamint a tranzakció kezelést (BEGIN WORK, COMMIT, ROLLBACK).
A PostgreSQL teljesen ingyenes, szabad szoftver, BSD (Berkeley Software Distribution) licences, nyílt forráskódú.
A PostgreSQL fejlesztőcsoport egy vállalatokból és együttműködő magánszemélyekből álló közösség a fejlesztések összehangolására. A PostgreSQL szoftver kifejlesztése 1986-ban kezdődött a Kaliforniai Egyetemen Berkeley-ben kutatási projektként, és az azóta eltelt 23 évben alakult ki a világ minden részére kiterjedő fejlesztőhálózat, amelynek központi szerverei Kanadában találhatóak.
A PostgreSQL egy szinte minden platformon működő nagyteljesítményű legendásan megbízható és stabil adatbázis szerver alkalmazás.

Néhány fontosabb technikai lehetőség:
- tökéletesen megfelel az ACID szabványnak
- teljesíti az ANSI SQL szabvány kritériumait
- natív programozási felületek ODBC, JDBC, C, C++, PHP, PERL, TCL, ECPG, Python és Ruby programnyelvekhez
- szabályok (rule)
- nézetek (view)
- triggerek
- Unicode-támogatás
- szekvenciák
- öröklődés
- outer join
- al-szelekciók
- procedurális nyelvek (tárolt eljárások)
- funkcionális és részleges indexelés

A PostgreSQL adatbázis-kezelőt széleskörű térinformatikai funkciókészlettel felruházó **PostGIS** kiterjesztés legújabb változata 1.3.6.
A PostGIS nevében a GIS szó utal a térinformatikai rendszer elterjedt angol kifejezésre: Geographic Information Systems. Ez a PostGIS tehát egy összetett adatbázis kiterjesztés a térinformatikai, földrajzi, térbeli adatok gyors, hatékony tárolására, kezelésére, feldolgozására. Ez a kiterjesztés szintén ingyenesen elérhető, mint maga a PostgreSQL. A program alapvetően az OGC SQL alapú adatbázis-kezelők térinformatikai funkciói számára kidolgozott ajánlásának (Simple Features SQL) megfelelően működik, tehát egy szabványos eszközről van szó. (A MySQL 4.1-es verziójában megjelenő térinformatikai funkciók ugyanezt a szabványt alkalmazzák.)

A legfontosabb dolgok, amiket a PostGIS tud:
- térképi adatok tárolása 'geometry' típusú mezőkben

- egy sor függvény és operátor ezen mezők kezelésére (terület/kerület/hossz számítás, halmazműveletek, övezet generálás, topológiai vizsgálatok, generalizálás, stb.)
- GIST típusú térbeli index alkalmazása, hogy nagyobb méretű állományok esetében is hatékonyan lehessen térképi adatok alapján keresni vagy egyéb lekérdezéseket végezni.
- Az 1.1.0 verziótól kezdődően megjelent a topológiai modell támogatása

A PostGIS-es kiegészítéssel rendelkező PostgreSQL adatbázisokat kezelni tudjuk például a QGIS térinformatikai programmal, vagy az UMN Mapserver segítségével internetes térképek hátterét is szolgáltathatják.

Térbeli (több dimenziós) **indexek** haszna:
Térbeli indexet hozhatunk létre egy tábla térképi elemeket tartalmazó oszlopához, meggyorsítva ezzel a lekérdezéseket olyan esetekben, amikor például az adatbázisból azokat az elemeket szeretnénk lekérdezni, melyek egy megadott területre esnek. Ezek a lekérdezések rendkívül gyakoriak, hiszen ilyen feladattal állunk szemben valahányszor ki akarjuk rajzolni egy ablakban egy terület térképének egy részletét. Térbeli index nélkül egy ilyen lekérdezéshez át kellene nézni a tábla valamennyi rekordját, mindet végigvizsgálva, hogy vajon tényleg a kérdéses területre esik-e az objektum. Ez nem valami hatékony megoldás, különösen nagyméretű táblák esetében. A térbeli index segítségével a program egy előszűrést tud végezni, amivel gyorsan kiszórhatja azokat az elemeket, melyek a vizsgált területnek még a közelében sincsenek. A többdimenziós indexelésre többféle módszert fejlesztettek ki. A PostGIS ezek közül a GiST (Generalized Search Tree) algoritmust használja.
A PostgreSQL GIST indexek kezeléséért felelős kódját a 8.1-es sorozattól kezdődően a PostGIS-t fejlesztő Refraction Research támogatásával teljesen újraírták. Ennek köszönhetően egy sokkal hatékonyabban működő kódot kaptak, ami a régebbivel ellentétben már támogatja a soronkénti zárolást a GIST-el indexelt oszlopot tartalmazó táblákban.

Az OGC WKT (Well-Known Text) és WKB (Well-Known Binary) szabvány szerint írja le a geometria típust a PostGIS.
A térképi elemeket tartalmazó oszlopot (geometry type) nem a CREATE TABLE parancs oszloplistájában kell hozzáadni a táblához, hanem az AddGeometryColumn függvény segítségével. Ez azért fontos, mert így létrejönnek a szükséges bejegyzések a geometry_columns táblában, ami egyes PostGIS funkciók és kliensprogramok helyes működése szempontjából fontos. Ha törölni szeretnénk a táblából egy térképi elemet, akkor ezt (az előbbihez hasonló megfontolásból) a DropGeometryColumn függvénnyel tegyük.

Térképi elemet tartalmazó táblák létrehozása:
Egy térképi adatokat tartalmazó tábla létrehozása tehát a következőképpen néz ki:

```
CREATE TABLE vezetekek (ID int4, anyag_kod char(5), atmero_mm int4);
SELECT AddGeomertyColumn('vezetekek','geom',-1,'LINESTRING',2);
```

Az első paranccsal létrehoztuk a 'vezetekek' nevű táblát az attribútum adatok oszlopaival. A második sorban lévő SELECT parancs az AddGeometry függvény meghívásával hozzáad a 'vezetekek' táblához egy 'geom' nevezetű oszlopot, ahova majd a térképi elemek kerülhetnek. A térképi elemek leírásához használt koordinátarendszerhez nem tartozik vetületi leírás, amire a harmadik paraméterben található -1 érték utal. A térképi elemek típusa 'LINESTRING', vagyis vonallánc lesz.

Az utolsó paraméterben lévő 2 a dimenziószámot jelenti. Ha 3 lenne itt, akkor az egyes alakjelző pontok helyzetét három koordinátával kellene megadnunk, vagyis tárolnánk a magasságukat is. A példában viszont csak kétdimenziós, síkbeli koordinátákat használunk az objektumok geometriájának meghatározásakor.

Sorok beszúrása:
Ezt az SQL INSERT INTO parancsával tehetjük meg. Ez eddig egyszerű dolog volna, de hogyan adhatjuk meg a térképi elemeket? A PostGIS-ben erre többféle függvény alkalmazható, ezek egyike a GeomFromText(), melyet a következőképpen használhatunk:

```
 INSERT INTO vezetekek (ID, anyag_kod, atmero_mm, geom) VALUES
 (1, 'PVC', 150, GeomFromText('LINESTRING(102.15 541.32,110.08
597.29,130.07 610.88,135.45 620.32)', -1));
```

Mint az a példából is látszódik, a GeomFromText függvénynek két paramétere van. Az első szöveges adatként tartalmazza a geometria leírását WKT formátumban, a második pedig a koordináták vetületi rendszerének kódját adja meg (SRID).
Ha csak pontokat akarunk megadni, akkor alkalmazhatjuk az egyszerűbb MakePoint() függvényt is:

```
 INSERT INTO meghibasodas (ID, leiras, hely) VALUES
 (1, 'dugulás', MakePoint(140.25,530.64));
```

A térképi elemet tartalmazó oszlop itt most a 'hely' nevet viseli.

Lekérdezések:
Ha fel vannak töltve a tábláink, lehetőségünk nyílik sokféle lekérdezésre. Néhány példa:

```
 SELECT Length(geom) FROM vezetekek WHERE ID=112;
```

A fenti módon kaphatjuk meg a 112-es azonosítójú vezeték hosszát.

```
SELECT telek.hrsz, COUNT(*),
100*area(telek.geom)/SUM(area(epulet.geom))
 FROM telek,epulet
 WHERE contains(telek.geom,epulet.geom)
 GROUP BY telek.hrsz, telek.geom
 ORDER BY 1;
```

Egy listában helyrajzi szám szerint rendezve megkapjuk, hogy az egyes telkeken hány épület áll, és hogy a telek hány százaléka van beépítve. A lekérdezés a contains() függvény miatt csak a GEOS támogatással lefordított PostGIS-ben használható.

Vetületi rendszerek kezelése:
A PostGIS a térképi elemek leírása mellett egy kódot is tárol, ami a vetületi rendszerre vonatkozik. Ez egy szám, aminek az értéke -1, ha a koordinátarendszer nem kapcsolódik semmilyen vetülethez (helyi rendszer), egyébként pedig a vetületi rendszernek a SPATIAL_REF_SYS táblában használt azonosítóját tartalmazza.

2.3. Google Earth és KML

Röviden arról mi is az a Google Earth:
A **Google Föld** (eredetileg: **Google Earth**) egy ingyenes számítógépes program, ami virtuális földgömbként használható. A Föld háromdimenziós modelljére mértékhelyes műholdképek, légi felvételek és térinformatikai adatok vannak vetítve. A programban a Föld minden részéről leolvashatók a földrajzi koordináták, és az adott pont magassága. A programot eredetileg a Keyhole cég fejlesztette ki, majd 2004-ben

megvásárolta a Google. A program Mac OS X Tiger, Linux (2006. június 12-étől), illetve Microsoft Windows 2000 és Windows XP operációs rendszereken fut.
A Google Föld Fizetős verziói 2009-ben megszűnnek, tehát egy teljesen ingyenesen hozzáférhető rendszerről van szó.
A Google Earth azért hasznos számomra, mert az adatbázisban tárolt adatokat lekérve megjeleníthetem őket egy valós térképen, így az összefüggéseket a valósághoz viszonyítva is vizsgálhatom, sokszögeket, vonalakat vagy éppen pontokat helyezhetek el a térképen. Így az elméletben kiszámolt, kitalált helyzeteket, a gyakorlatban is meg tudom vizsgálni.

A Google Earth a **KML** file-okat tudja felhelyezni a saját földgömbjére, ezért én is ilyen KML file-okat készítettem. (KMZ file-okat is kezel, de én azt nem használtam, mert a KML fájl ZIP programmal tömörített formája a KMZ fájl.). A PostGIS tartalmaz egy olyan függvényt ami a lekérdezéseket rögtön a KML formátumnak megfelelően formában adja vissza. Ez nagyban megkönnyítette a dolgomat. A lekérdezések eredményét csak be kellett másolni egy szabványos headerrel és footerrel ellátott üres kml file-ba és máris megjelenthettem azt a földgömbre vetítve a Google Earth-el.

Röviden a **KML** nyelvről:
A **KML** (**K**eyhole **M**arkup **L**anguage) XML-alapú jelölőnyelv térben ábrázolt alakzatok megjelenítésére a Google Earth, Google Maps és a „Google Maps for mobile” programokban.
A KML fájl különféle tulajdonságokat határoz meg, ilyenek a hely, kép, poligon, 3D modell, textúra, leírás, stb. jelölése. A helynek mindenképpen van földrajzi hosszúsági és szélességi koordinátája. A további paraméterek a kamera-nézetet határozzák meg, ezek a dőlésszög, az irány, és a magasság. Mivel egyfajta XML fájlról van szó, ezért itt is különbözőek a kis- és nagybetűk, a paraméterek azonosítóit pontosan kell beírni.
Példa egy egyszerű KML fájlra:

```
<?xml version="1.0" encoding="UTF-8"?>
<kml xmlns="http://earth.google.com/kml/2.0">
<Placemark>
  <description>New York City</description>
  <name>New York City</name>
  <Point>
    <coordinates>-74.006393,40.714172,0</coordinates>
  </Point>
</Placemark>
</kml>
```

Példa egy hosszabb általam készített KML file-ra:

```
<?xml version="1.0" encoding="UTF-8"?>
<kml xmlns="http://www.opengis.net/kml/2.2"
xmlns:gx="http://www.google.com/kml/ext/2.2"
xmlns:kml="http://www.opengis.net/kml/2.2"
xmlns:atom="http://www.w3.org/2005/Atom">
<Document>
   <name>GSM.kml</name>
<Placemark>
   <description></description>
   <name></name>
"<Polygon><outerBoundaryIs><LinearRing><coordinates>19.2176616666667,
47.4384316666667 19.2156283333333,47.4396 19.20874,47.443655
19.20802,47.4441633333333 19.2085066666667,47.4440233333333
```

```
19.2094283333333,47.443525 19.2121666666667,47.4418516666667
19.2176616666667,47.4384316666667</coordinates></LinearRing></outerBo
undaryIs></Polygon>"
  </Placemark>
  </Document>
  </kml>
```

Az első példában egyszerűen csak egy pontot jelölünk ki a térképen és megadjuk a pont címkéjét ami utal rá, hogy ott található New York City.
A második példában egy poligont (sokszöget) rajzolunk ki a térképre aminek a <coordinates> tag után található számpárok lesznek a sarokpontjai, és az első számpár lesz a kezdő és végpontja a sokszögnek.

3. Földrajzi adatbázis GSM helymeghatározáshoz

3.1. Architektúra

Az adatok, amelyek az adatbázis alapját képezték, a korábban és folyamatosan zajló mérésekből származó mérési eredmények voltak. Az adatokat erre a célra készített mérésadatgyűjtő eszközökkel gyűjtöttük. A mérésadatgyűjtő készülék minden mérést egy txt file-ban tárolta és már az adatbázisom építésének kezdetén rengeteg mérésfájl állt rendelkezésre.

A laptopomra telepítettem a PostgreSQL adatbázis-kezelő rendszert és ennek kiterjesztését a PostGIS-t. Ezt én Windows XP operációs rendszerre telepítettem, de természetesen a PostgreSQL-nek, mint a legtöbb ingyenes szoftvernek a Linux az alapvetően javasolt környezete, a legtöbb Linux disztribúción már az O.S. telepítésekor választható opció a PostgreSQL telepítése. Ez a háttért szolgálta az adatbázis építését.

Tervbe van véve, hogy az adatbázisom a későbbiek során átveszi, az eddig is használt MySQL adatbázist, ami jelenleg is fut a kutatócsoport szerverén. Ehhez viszont még az kellene, hogy jobban kiismerjem a PostgreSQL-t és a PostGIS-t, és hogy az adatok feltöltését minél egyszerűbben mindenki számára használhatóan tudjam garantálni. Sajnos az adatok feltöltésének szakszerű módját nem sikerült kidolgozni. Az idő hiánya miatt a konzulensemmel úgy gondoltuk, hogy inkább hagyunk több időt a lekérdezések elkészítésére és így az adatok betöltését egyfajta „kerülőúttal” oldottam meg. Az adatokat lekérdeztem a már meglévő MySQL adatbázisból az adatbázis phpMyAdmin grafikus felületén és az így kapott INSERT INTO (…) SQL parancsokat futtattam az én adatbázisomon. Így a feltöltés ideiglenesen megoldódott, de a későbbiekben tervezem, hogy az adatokat közvetlenül a mérésből származó txt file-okból lehessen feltölteni, egy pl. Java nyelven írt feltöltő programmal, ami JDBC driver segítségével kapcsolódik az adatbázishoz és feltölti a teljes file tartalmát.

A teljes adatbázison keresztüli architektúra tehát így épül fel:

- mérési adatok gyűjtése txt-be
- ennek betöltése az adatbázisba a megfelelő struktúra szerint
- az adatok feldolgozása, szűrése, lekérdezése
- a lekérdezések kiértékelése a megjelenő szöveges információk alapján
- vagy a lekérdezések kml fileba írása, majd megjelenítése a Google Earth-ön és a látottak értékelése.

3.2. Adatbázis séma

Az adatbázis sémáját a már meglévő MySQL adatbázisból vettem át, mivel nem láttam értelmét új séma kialakításának. A meglévő sémát használta már régebb óta a csoport többi tagja.
A séma a következőképpen néz ki részletesen:

GPS tábla:

```
CREATE TABLE gps (
  id integer NOT NULL,
  mid integer NOT NULL,
  lat decimal(9,4) default NULL,
  lng decimal(9,4) default NULL,
  sats smallint default NULL,
  alt decimal(5,1) default NULL,
  PRIMARY KEY  (id)
) WITH (OIDS=FALSE);
ALTER TABLE attributes OWNER TO postgres;
```

id: saját ID – minden gps mérési bejegyzéshez
mid: mesure ID – melyik mérésadatgyűjtő készülékkel készült
lat: latitude, azaz É-D hosszúsági fok
lng: longitude, azaz K-NY szélességi fok
sats: a GPS vevő által látott műholdak száma
alt: altitude, azaz tengerszint feletti magasság

Ehhez a táblához tartozik még egy nagyon fontos bejegyzés amit én raktam hozzá, gyakorlatilag ez a kulcsa az egész földrajzi adatbázis kezelési rendszernek, ez pedig a geometria típus.
Ezt a következőképp lehet hozzáadni a táblához:

```
SELECT AddGeometryColumn('gps', 'gps_geom2', 4326, 'POINT', 2);
```

AddGeometryColumn(): ez egy függvény ami generál egy geometria típusú oszlopot
Paraméterei:
gps: a tábla neve, ahova hozzá akarom adni az új oszlopot
gps_geom2: ez az új oszlop neve
POINT: ez a geometria típusa, lehetne még LINESTRING, POLYGON, ezek többszörös multi változatai (pl.: MULTIPOINT) és GEOMETRYCOLLECTION, amibe többfajta geometriát gyűjthetünk.
2: ez a pont dimenziószáma
4326: ez az SRID (**S**patial **R**eference **ID**entifier) ami azonosítja a használt koordináta rendszert. Ilyen SRID-je van pl. a jelentős földrajzi azonosítási rendszereknek, minden országnak, Magyarországé a 23700 számú SRID. Én azért használom a 4326-osat mert ez a GPS rendszerek hivatalos térbeli referencia azonosítója és ezt használja a Google Earth is, így teljesen kompatibilis lesz az adatbázisom vele.

Most egy kicsit előreszaladnék az időben, mert az adatok betöltése utánra ugrok. Az adatok betöltése után lefuttattam ezt az UPDATE-ot:

```
update gps set gps_geom2 = GeometryFromText('POINT(' || (((lng/100)-
trunc(lng/100))/60*100)+(trunc(lng/100)) || ' ' || (((lat/100)-
trunc(lat/100))/60*100)+(trunc(lat/100)) || ')', 4326);
```

Ez az update a már betöltött lng és lat koordinátákat transzformálja át a geometria oszlopba. A transzformáció azért kell, mert a vevő készülék által rögzített formátum első két számjegye fok, második két számjegye szögperc (0 és 60 között változik) és a többi érték pedig a szögperc tört részei (tizedszögperc, stb.) Ezzel szemben a Google Earth által is elfogadott és helyes tároláshoz a két számjegy a fok, a többi pedig a tizedesjegy, tizedfok, századfok, stb.
Példa: a vevő által generált: 1913.0251 ; 4726.3264
átkonvertálva a helyes, jól kezelhető formára: 19.217085 ; 47.438773
Gyakorlatilag a konvertálás során a tizedesvesszőt arrébb viszem kettővel balra, hogy a fok után következzen és ezután a tizedes részt osztom 60-al és szorzom 100-al így megkapom a helyes törtrészt.

GSM tábla:

```
CREATE TABLE gsm (
  id integer NOT NULL,
  gid integer NOT NULL,
  cid integer NOT NULL,
  rxlev smallint default NULL,
  rxlevf smallint default NULL,
  rxlevs smallint default NULL,
  rxq smallint default NULL,
  rxqf smallint default NULL,
  rxqs smallint default NULL,
  idle smallint default NULL,
  PRIMARY KEY  (id)
) WITH (OIDS=FALSE);
ALTER TABLE attributes OWNER TO postgres;
```

id: minden bejegyzéshez saját ID
gid: GPS ID, kapcsolódási pont a gps táblához
cid: Cell table ID, kapcsolódási pont a cell táblához (nem a cell id)
rxlev: rssi, a vett rádió jel erőssége (0..63 között)
A többi attribútum a jelen helyzetben nem fontos, azok legtöbb esetben nullák.

CELL tábla:
SQL:

```
CREATE TABLE cell (
  id integer NOT NULL,
  mcc smallint default NULL,
  mnc smallint default NULL,
  lac character(4) default NULL,
  ci character(4) default NULL,
  bsic smallint default NULL,
  freq smallint default NULL,
  PRIMARY KEY  (id)
) WITH (OIDS=FALSE);
ALTER TABLE attributes OWNER TO postgres;
```

id: minden bejegyzéshez egyedi ID
mcc: Mobile Country Code: mobil országkód, melynek hossza 3 számjegy, és egyértelműen meghatározza a mobil előfizető hálózata szerinti országot. A mobil országkódokat az ITU jelöli ki. Magyarország mobil országkódja: 216.
mnc: Mobile Network Code: mobil hálózati kód, melynek hossza két számjegy. Az MNC az MCC-vel együtt egyértelműen meghatározza a mobil rádiótelefon szolgáltatást igénybe vevő végberendezés vagy előfizető honos hálózatát. Az MNC az

MCC-vel együtt, a mobil szolgáltatást nyújtó hálózatokkal jelzéstechnikailag kompatibilis szolgáltatás nyújtása céljából egyértelműen azonosíthat helyhez kötött telefonhálózatot vagy hálózat csoportot is. A mobil hálózati kódot a hatóság jelöli ki. A kijelölés feltételeit külön jogszabály tartalmazza.
lac: Location Area Code: A 4 számjegyből álló azonosító, ami egy nagyobb terület azonosítására szolgál.
ci: Cell Identifier: 4 hexadecimalális számjegyből álló azonosító ami azonosítja a cellát
bsic: Base station identity code: a mobil készüléket segíti a különböző szomszédos bázisállomások megkülönböztetésében. A BSIC egy úgy nevezett „színkód" ami azt jelenti, hogy a szomszédos cellák más képzeletbeli színnel vannak jelölve, és nem lehet egymás mellett két azonos színű cella.
freq: az a frekvencia amin az adó sugároz

3.3. Jelenlegi adatok

A mérési adatok gyűjtése azt a célt szolgálta, hogy jobban megismerhessük, elemezhessük a GSM rendszerek rádiós technológiáját, a rádiós jelek viselkedését, a lehetséges helymeghatározási megoldásokat minél jobban tökéletesítsük. Mérési adat gyűjtéséhez két fő adatcsoportot kellett gyűjteni. Az aktuális pozíciónkat kellett rögzíteni, ez rendszerint GPS vevő segítségével, valamint az adott pozícióban éppen vett GSM cellainformációkat.
A mérések legnagyobb részét a Kelemen Mihály csoporttagunk által erre a célra készített mérésadatgyűjtővel gyűjtöttük. Ennél az adatgyűjtőnél egy chipre volt kötve a GPS vevő és a GSM vevő, és az adatokat egy időbélyeggel ellátva tárolta el egy txt fájlban. Az adatokat egy másodpercenként rögzítette.
A másik mérésadatgyűjtő eszköz egy Samba75 típusú számítógéphez csatlakoztatható GSM vevőegység, ami USB felületen kapcsolódhat egy számítógéphez és így tudja rögzíteni a számunkra szükséges GSM adatokat. Ez a fajta adatrögzítés esetén nem keletkezik GPS koordináta, így azt nekünk kell „kézzel" hozzáadnunk. Ezt a készüléket a Robotika Laborban található Nagy Sárga Robothoz kapcsolta Feldhoffer Gergely és Rudán János, így a robot segítségével belső, épületeken belüli méréseket végeztünk, amihez a koordinátákat a robot szolgáltatta.

Felmerült az ötlet, hogy az adatbázisban érdemes lenne az adatokat ritkítani, mert a GSM vevők nem frissítenek egy másodperc alatt, hanem általában 2 vagy több másodpercenként frissítik csak a vételi szintet (rssi) és a cella információkat. Nem sikerült megegyezni abban, hogy mennyi adatot lenne érdemes kitörölni, így jelenleg a mérésekben rögzített összes adat megtalálható az adatbázisban. Eddig körülbelül 25.000 GPS mérési pont van, és ehhez kapcsolódik több mint 140.000 egyedi GSM mérési értéksor. Ez azt jelenti, hogy körülbelül minden GPS ponthoz tartozik 5-6 GSM bázisállomás bemérése. Ez megfelel a mai mobiltelefonok mérési szokásainak, azok általában 6 bázisállomás információit tartják számon minden pillanatban, ahol elérhető legalább 6 bázisállomás.
A legtöbb mérés a fővárosban készült és sajnos a GPS koordináták a szűk utcák magas házfalai alatt haladva pontatlan értéked adtak, és ezek a hibák megjelennek majd a lekérdezéseimnél is.

3.4. Lekérdezések

A feladatom fő része azok a speciális lekérdezések elkészítése amelyek kihasználva a földrajzi adatbázis nyújtotta előnyöket (beépített geometriai függvények, gyors indexelése a geometriai adatoknak, stb.) értékelhető eredményeket adtak. Akár látványosan a képernyőn megjeleníthető formában (Google Earth-re rajzolással), akár valamilyen hagyományos szöveges információt visszaadva.
Most tehát a lekérdezések következnek:

3.4.1. A teljes eddigi mérések lekérdezése :

Lekérdezem az adatbázisban található összes GPS koordinátát.
SQL:

```
SELECT ST_AsKML(gps_geom2) FROM gps WHERE lat>0 AND lng >0;
```

A WHERE feltételre azért volt szükség, hogy a 0,0 GPS koordinátájú adatokat kiszűrjem. Az ilyen 0,0 adatok azért kerülnek bele a mérésbe, mert a mérésadatgyűjtő eszközben lévő GPS vevő a bekapcsoláskor rövid ideig szinkronizál és ekkor még nem ad GPS koordinátákat, a GSM vevő azonban már veszi a jeleket és a készülék rögzíti is és ekkor a GPS helyére 0,0-t ír.
Az ST_AsKML() függvény készít a geometria típusú adatokból KML formátumú szöveges kimenetet, ezt a függvény gyakran elő fog kerülni a lekérdezéseimben, ahol az eredményt Google Earth-re kirajzolva szeretném ábrázolni.

A lekérdezés ilyen kimenetet produkált:

…
"<Point><coordinates>19.2235316666667,47.4374033333333</coordinates></Point>"
"<Point><coordinates>19.2235316666667,47.4374016666667</coordinates></Point>"
…

Ezt a szöveges kimenetet helyeztem bele egy szövegfájlba, amit az aztán az általam Java nyelven írt rövid program beolvasta és ellátta a szabványos KML fejrésszel és lábrésszel.
Az így kapott eredmény egy budapesti részlete látható az 1. ábrán.

1. ábra

Ennek a lekérdezésnek nincs komoly tartalmi jelentése, csupán azért raktam be ide, hogy látható legyen a mérési adatgyűjtéseink főbb helyszínei, valamint egy egyszerű bevezető példaként szolgáljon.

3.4.2. Adott bázisállomás vételi szintjeinek kirajzolása a pontok színezése alapján

Az általam választott bázisállomás a cid=170 azonosítójú állomás.
A pontok színezése sajnos csak 3 általam választott kvantált érték alapján történt, a helyesebb színezés az emberi szemmel is alig látható finom színátmenetes megoldás lett volna, de sajnos erre időhiány miatt nem került sor.

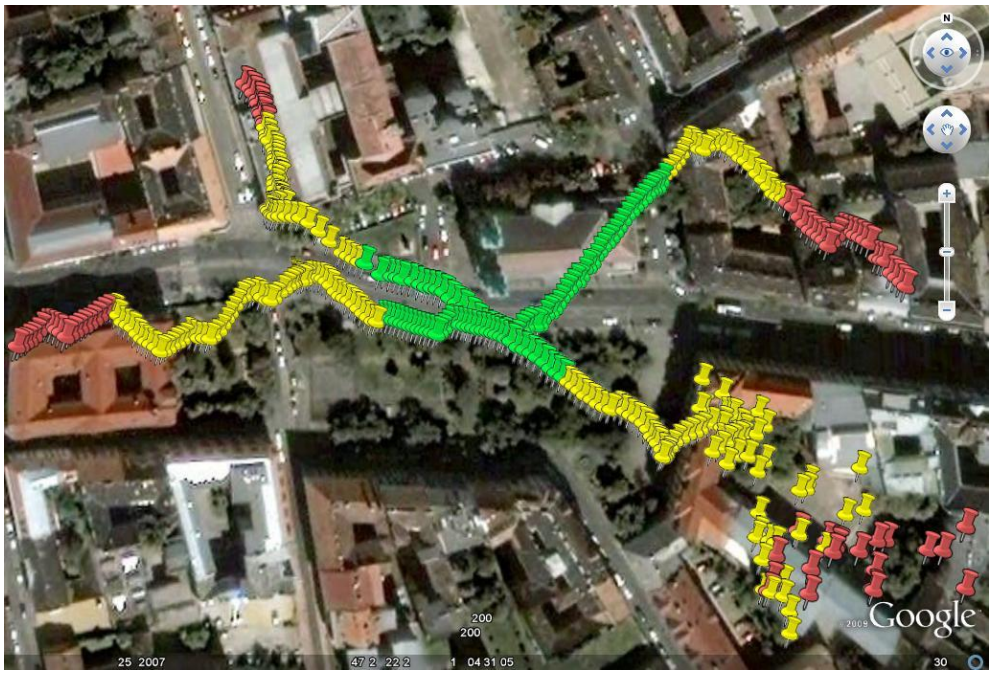
SQL:

```
SELECT ST_AsKML(gps_geom2), gsm.rxlev
FROM gps, gsm
WHERE gps.id=gsm.gid AND gsm.cid=170 AND gps.lng>0 AND gps.lat>0;
```

A kapott eredmény alakja:

…
"<Point><coordinates>19.076905,47.4887966666667</coordinates></Point>";13
"<Point><coordinates>19.07687,47.4889733333333</coordinates></Point>";24
"<Point><coordinates>19.0766333333333,47.4890566666667</coordinates></Point>";24
…

Ezt az eredményt egy rövid Java programmal dolgoztam fel, ami elvégezte a 3 színre kvantálást. Az eredmény a 2. ábrán látható.



2. ábra

Itt tehát látható, hogy a bázisállomástól távolodva, hogyan csökken a vett jel szintje. Sajnos a pontok nem teljesen az úttesten helyezkednek el ahol a mérés történt, hanem a GPS hibájából (magas házfalak mellett ment a mérést végző személy) arrébb vannak.

3.4.3. Bázisállomás helyének közelítése:

Gyenge közelítés arra, hol található egy adott bázisállomás (cid=170) a mért pontok alapján. Sajnos a közelítés nagyon elnagyolt, mert csak a pontok egymáshoz vett elhelyezkedése alapján számolja ki a középpontot és nem veszi figyelembe a vett jelszintek értékét. A későbbiekben ez pontosabban megvalósítható lenne, ha a 2D koordinátapontokat áthelyezném 3D-s reprezentációba, ahol a 3. pont a vett jelszint lenne. Így valószínűleg még jobb közelítést kapnánk a bázisállomás elhelyezkedésére. Az így maghatározott bázisállomást hívják virtuális bázisállomásnak, mert ugyan az állomás nem pont ott van ahova kiszámoltuk, de jó közelítés lehet a későbbiekben, hogy a bázisállomást ott lévőnek tekintjük.

SQL:

```
SELECT  ST_AsKML(ST_Centroid(ST_Collect(gps_geom2)))
FROM gps, gsm
WHERE gps.id=gsm.gid AND gsm.cid=26 AND gps.lng>0 AND gps.lat>0;
```

A lekérdezés eredménye: (egyetlen pont)
"<Point><coordinates>19.2127882380952,47.4413741428572</coordinates></Point>"

Az ST_Collect() függvény szolgált arra, hogy a mért pontokat egy ponthalmazba foglaljam (ez a következő függvénynek kell a bemenetére), majd ebből az ST_Centriod() függvény számolta ki a ponthalmaz közepét.
A lekérdezés eredményét csak egyszerűen bemásoltam egy szabvány KML file-ba és ezt már meg is jeleníthetem.

Az eredmény a 3. ábrán látható (ez a virtuális bázisállomás az előző lekérdezésben, a 2. ábrán látható vételi szinteket produkáló bázisállomás, érdemes összehasonlítani a kapott eredményt).

3. ábra

Látható, hogy ezzel a módszerrel, ha bemérünk egy adott területet, akkor gyors lekérdezéssel meghatározhatjuk az ott található virtuális bázisállomásokat.

3.4.4. Tartalmazás lekérdezése:

Annak lekérdezése, hogy adott bázisállomás adott jelszint feletti lefedett területét tartalmazza-e a kisebb jelszint feletti lefedett területe. Alapesetben azt gondolnánk, hogy persze egyértelműen ez mindig igaz, hisz a jelszint az adótoronytól távolodva csökken, de ismerve a jelterjedési tulajdonságokat, ez könnyen előfordulhat, hogy nem minden esetben teljesül. Elég hogy ha arra gondolunk, hogy az épületbe is pl. az ablakon keresztül behatol a jel és közel sem biztos, hogy ezért az ablak mellett nagyobb lesz a jelszint, mint az ablaktól távolodva. A rádióhullámú jelek terjedésében megfigyelhető jelterjedési mechanizmusok: reflection, scattering, diffraction. Ezek tudnak furcsa jelszinteket produkálni például egy háztömb sarkánál.

SQL:

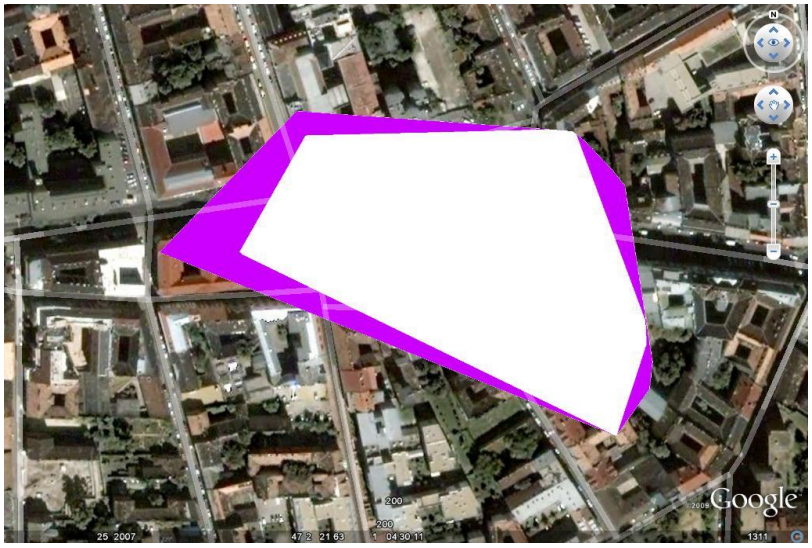
```
SELECT ST_Contains(t1.gg1,t2.gg2)
from(SELECT ST_ConvexHull(ST_Collect(g1.gps_geom2)) as gg1
       FROM gps as g1, gsm as gsm1
      WHERE g1.id=gsm1.gid AND gsm1.cid=170 AND gsm1.rxlev>10 AND
g1.lng>0 AND g1.lat>0) t1,
      (SELECT ST_ConvexHull(ST_Collect(g2.gps_geom2)) as gg2
      FROM gps as g2, gsm as gsm2
      WHERE g2.id=gsm2.gid AND gsm2.cid=170 AND gsm2.rxlev>20 AND
g2.lng>0 AND g2.lat>0) t2;
```

Ez egy egymásba ágyazott SELECT utasítás, ahol veszek két ponthalmazt, amik a két jelszint melletti mérési pontok, majd ezeket egy befoglaló geometriával „beburkolom” (ST_ConvexHull) és az így kapott sokszögeken megnézem, hogy a nagyobb jelszinthez tartozót tartalmazza-e (ST_Contains) a kisebb jelszinthez tartozó sokszög. Az eredmény egy logikai érték, igaz(t), vagy hamis(f).

Eredmény:

```
st_contains
boolean
t
```

Ehhez is készítettem ábrázolást, hogy látható legyen a lekérdezés tartalmi lényege, amit a 4.ábra szemléltet.



4. ábra

Ezzel a módszerrel lehetőség nyílik megvizsgálni azt a teóriát, mely szerint a jelszintre a gyakorlatban gondolhatok úgy, hogy a bázisállomástól távolodva folyamatosan csökken, lényegében koncentrikus körök (amely a gyakorlatban lehet akár egy babapiskóta szerű alakzat) mentén. Ezen a képen a kevés mérési adat miatt látható egy trapézszerű alakzat, a kör szerű alakzat helyett.

3.4.5. 3 bázisállomás vett jelszintje alapján hol lehetek:

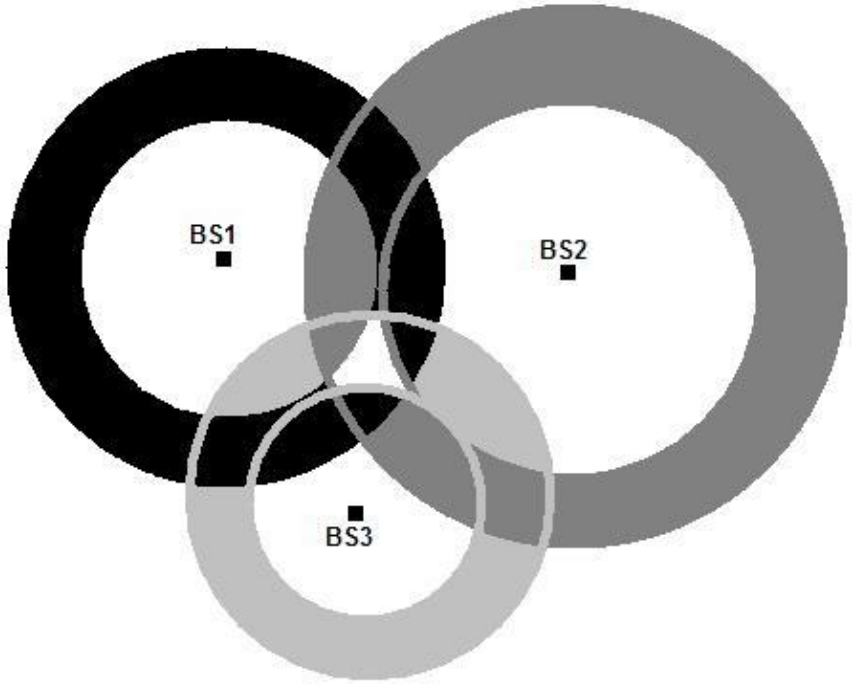
3 bázisállomás vett jelszintje(rssi) alapján meghatározom a mérésekből, hogy most hol is állhatok. A jelszinteket 5 hosszú intervallumon belül határoztam meg, amely jobban megfelel a valóságnak, ugyanis elképzelhető, hogy a mobiltelefonunk nem frissít olyan gyorsan, ahogy változik a jelszint.

SQL:

```
SELECT
ST_AsKML(ST_Intersection(ST_Intersection(tt1.gg1,tt2.gg2),tt3.gg3))
from(

SELECT ST_Difference(t1.ggg1,t2.ggg2) as gg1
from
(SELECT ST_ConvexHull(ST_Collect(g1.gps_geom2)) as ggg1
       FROM gps as g1, gsm as gsm1
      WHERE g1.id=gsm1.gid AND gsm1.cid=266 AND gsm1.rxlev>52
      AND gsm1.rxlev<56 AND g1.lng>0 AND g1.lat>0) t1,
(SELECT ST_ConvexHull(ST_Collect(g2.gps_geom2)) as ggg2
      FROM gps as g2, gsm as gsm2
      WHERE g2.id=gsm2.gid AND gsm2.cid=266  AND gsm2.rxlev>56
      AND g2.lng>0 AND g2.lat>0) t2) as tt1,

(SELECT ST_Difference(t1.ggg1,t2.ggg2) as gg2
from
(SELECT ST_ConvexHull(ST_Collect(g1.gps_geom2)) as ggg1
       FROM gps as g1, gsm as gsm1
      WHERE g1.id=gsm1.gid AND gsm1.cid=270 AND gsm1.rxlev>52
      AND gsm1.rxlev<56 AND g1.lng>0 AND g1.lat>0) t1,
(SELECT ST_ConvexHull(ST_Collect(g2.gps_geom2)) as ggg2
      FROM gps as g2, gsm as gsm2
      WHERE g2.id=gsm2.gid AND gsm2.cid=270  AND gsm2.rxlev>56
      AND g2.lng>0 AND g2.lat>0) t2) as tt2,

(SELECT ST_Difference(t1.ggg1,t2.ggg2) as gg3
from
(SELECT ST_ConvexHull(ST_Collect(g1.gps_geom2)) as ggg1
       FROM gps as g1, gsm as gsm1
      WHERE g1.id=gsm1.gid AND gsm1.cid=262 AND gsm1.rxlev>52
      AND gsm1.rxlev<56 AND g1.lng>0 AND g1.lat>0) t1,
(SELECT ST_ConvexHull(ST_Collect(g2.gps_geom2)) as ggg2
      FROM gps as g2, gsm as gsm2
      WHERE g2.id=gsm2.gid AND gsm2.cid=262  AND gsm2.rxlev>56
      AND g2.lng>0 AND g2.lat>0) t2) as tt3;
```

Ennek a nagyon hosszú SQL parancsnak a lényege, hogy a jelszintek alapján veszek egy ponthalmazt, ami az adott bázisállomástól adott szinten vett jelek pontjai. Ezt a ponthalmazt az ST_ConvexHull() függvénnyel körülveszem egy sokszöggel. Sajnos ekkor előjön az a hiba, hogy a ponthalmazom az elmélet szerint egy körgyűrűn helyezkedik el, viszont a ConvexHull függvény a kör közepét is belefoglalja, ezért

ebből kivonom az adott jelszint alatti területet ami egy kisebb kör lesz. Így már kaptam egy körgyűrűt. Ugyanezt a műveletsor elvégzem a második és harmadik bázisállomásra is. Így elvben kaptam 3 körgyűrűt. Ezek metszete már meghatároz egy összefüggő területet. Egy kis rajzzal illusztráltam, hogy mit is csinál a lekérdezés, ez az 5. ábrán látható.



5. ábra

Tehát a három körgyűrűt határoztam meg a három bázisállomástól jövő jel szintek alapján és ezek metszetét vette. Így megkaptam az 5. ábra közepén látható metszet fehér területét. Ugyanilyen fehér terület látható a 6.ábra valós lekérdezésében is, amit bekarikáztam.

Eredmény:
"<Polygon>
<outerBoundaryIs><LinearRing><coordinates>19.0265086930456,47.5004017625899
19.0265283333333,47.50042 19.02656,47.5004483333333 19.02659,47.5004666666667
19.0266683333333,47.5005133333333 19.0273233333333,47.5008583333333
19.0273483333333,47.50086 19.0274033333333,47.5007416666667
19.0273916666667,47.5007333333333 19.0267133077142,47.5004563265016
19.0265086930456,47.5004017625899</coordinates></LinearRing></outerBoundaryIs>
</Polygon>"

Ezt az eredményt belerakva egy KML file-ba a Google Earth azonnal meg tudja jeleníteni és ez nagy könnyebbséget jelent, ha ugyanezt más hagyományos adatbázis-kezelő rendszerekből kellene lekérdezni, majd megjeleníteni.

Ha mindhárom bázisállomást 52-56 közti jelszinten látjuk, akkor a 6. ábra fehér területén tartózkodhatunk (bekarikáztam, a könnyebb észrevehetőségért).

6. ábra

A kapott fehér terület a lekérdezés által adott terület, ami meghatározza a helyünket, azonban ez a terület a valóságban jóval nagyobb, a jobb, valószerűbb lekérdezéshez több mérési adat kellett volna. Néhány egymáshoz közeli bázisállomást kellett volna körülmérni, és így körgyűrűkhöz jobban hasonlító területek metszetét tudtam volna venni, így viszont a szögletes területek metszetéből csak ilyen kis területet adott vissza.
Így ezzel a lekérdezéssel lehetőségünk van egy nagyon alap változatú adatbázisból való helymeghatározásra a vett jelszintek (rssi) alapján.

4. Összefoglalás és kitekintés

Az eddig leírtakban többször is megemlítettem, hogy milyen későbbi feladatok várnak még megoldásra. Úgy vélem még sok lehetőség rejlik a Földrajzi adatbázisok használatában a GSM alapú helymeghatározás kutatásához. Valószínűleg a következő félévben tovább folytatom az adatbázis építését, a lekérdezések javítását, pontosítását, akár más módszerek kombinálásával is.

Források:

Bányai Balázs, Feldhoffer Gergely, Tihanyi Attila: Helymeghatározás GSM hálózat felhasználásával
http://www.postgis.org/
http://www.postgresql.org
http://gisfigyelo.geocentrum.hu/
http://weblabor.hu/cikkek/postgresqltelepites
http://www.linuxvilag.hu
https://fagus.inf.elte.hu/fagus/szolgaltatasok/terinformatikai-adatbazisok
http://hup.hu/