

Földrajzi adatbázis készítése GSM alapú helymeghatározáshoz

Pázmány Péter Katolikus Egyetem
Információs Technológiai Kar
Mérnök Informatikus BSc

Vinis Ádám

Témavezetők:
Lukács Gergely
Tihanyi Attila

2010

Nyilatkozat

Alulírott Vinis Ádám Lajos, a Pázmány Péter Katolikus Egyetem Információs Technológiai Karának hallgatója kijelentem, hogy ezt a szakdolgozatot meg nem engedett segítség nélkül, saját magam készítettem, és a szakdolgozatban csak a megadott forrásokat használtam fel. Minden olyan részt, melyet szó szerint, vagy azonos értelemben, de átfogalmazva más forrásból átvettem, egyértelműen a forrás megadásával megjelöltem. Ezt a Szakdolgozatot más szakon még nem nyújtottam be.

Budapest, 2010. 05. 17.

.....
Vinis Ádám Lajos

Tartalomjegyzék

Tartalomjegyzék	3
Ábrajegyzék	4
Tartalmi összefoglaló.....	6
Abstract	7
1. Bevezetés	8
2. Helymeghatározás.....	10
2.1. Műholdas helymeghatározás (GPS).....	10
2.2. Mobilkommunikáció (GSM)	11
2.2.1. A GSM rendszer alapjai.....	11
2.2.2. GSM alapú helymeghatározás	12
3. Földrajzi adatok kezelése.....	14
3.1. Térinformatika	14
3.2. Adatbázis-kezelő rendszerek földrajzi kiterjesztése	15
3.2.1. Áttekintés	15
3.2.2. PostgreSQL, PostGIS	16
3.3. Google Earth és a KML formátum	20
3.4. OpenStreetMap (OSM).....	22
4. GSM helymeghatározás.....	25
4.1. Bevezetés	25
4.2. Adatkezelés	26
4.2.1. Architektúra	26
4.2.2. Szoftverkomponensek.....	28
4.2.3. Adatbázis séma	29
4.2.4. Adattartalom	30
4.3. Helymeghatározás.....	31
4.3.1. Alapeset	32
4.3.2. Jelszintek figyelembevétele nélkül	35
4.3.3. Főkomponens analízis segítségével	37
5. Térképészeti adatokkal bővített helymeghatározás.....	48
5.1. Bevezetés	48
5.2. Adatkezelés	48
5.3. Bővített helymeghatározás.....	54
5.3.1. Helymeghatározás utcaterképpel.....	54
5.3.2. Helymeghatározás tömegközlekedési hálózattal.....	58
6. Összefoglalás és kitekintés	60
7. Irodalomjegyzék	62

Ábrajegyzék

1. ábra - Tábla létrehozása 2 dimenziós geometriai mezővel	19
2. ábra – Sor beszúrása geometriai adatokkal	19
3. ábra - MakePoint függvény használata	20
4. ábra – Hossz számítás	20
5. ábra – Területszámítás	20
6. ábra – Példa egy egyszerű KML fájlra	21
7. ábra – Példa egy hosszabb általam készített KML fájlra.....	22
8. ábra - OSM térkép készítése	24
9. ábra - Architektúra diagram	27
10. ábra - Sémadiagram.....	29
11. ábra - Budapesti mérési adatok.....	31
12. ábra - Kültéren elvégzett mérések pontossága	32
13. ábra.....	33
14. ábra.....	34
15. ábra.....	34
16. ábra.....	35
17. ábra - Egy felmért terület	36
18. ábra – 7 bázisállomás itt együtt látszik.....	36
19. ábra – A 7 bázisállomás által besugárzott terület középpontja	37
20. ábra - Középpont	39
21. ábra - Pontok az analízishez	39
22. ábra - Pontok lekérdezése	40
23. ábra – Bázisállomások lekérdezése	40
24. ábra – Bázisállomások vételi szintjei adott pozíciókon (táblázatrészlet)	40
25. ábra – Jelszintek lekérdezése	42
26. ábra – A komponensek súlyai.....	43
27. ábra – A főkomponens-analízis első 3 főkomponensének táblázata (részlet)	44
28. ábra - 1. főkomponens változása.....	44
29. ábra - Második főkomponens változása.....	45
30. ábra - Harmadik főkomponens változása.....	46

31. ábra - Első három főkomponens	46
32. ábra - Új tábla létrehozása	49
33. ábra - SRID átírása	49
34. ábra - SRID ellenőrzése	49
35. ábra - OSM sémadiagram	50
36. ábra - OSM adatok.....	52
37. ábra – GSM és OSM adatok	53
38. ábra - Magyarországi OSM adatok	54
39. ábra - 7 bázisállomás egyszerre látszik	55
40. ábra - Közepont	55
41. ábra – SQL lekérdezés: helymeghatározás vetítése a legközelebbi utcára	56
42. ábra - Helymeghatározás utcára vetítve	57
43. ábra - Helymeghatározás az utca meghatározásával	58
44. ábra - Az utca koordinátái és neve	58
45. ábra - Buszmegálló helymeghatározás	59
46. ábra - A következő megálló pozíciója és neve	59

Tartalmi összefoglaló

(Egy-két oldalas tartalmi összefoglaló magyarul és angolul (aki akarja, elkészítheti még további nyelve(ke)n is). A magyar tartalmi kivonat *be nem kötött* másolatát 1 példányban mellékelni kell.)

- nagyobb project része – cél gsm alapú helymeghat vizsgálata
- helymegh – állomások, jelszintek használatával
- munka keretében – földr adatb – mérési adatok betölt – vizsgálatok
- helymeghatározási módok – 1-1 mondatban
- további pontosítás pl. utcára, tömegközl.
- infrastruktúra sokféle elemzést tesz lehetővé – sok földrajzi művelet áll rendelkezésre jól skálázódik – háttérben álló adatbázisnak köszönhetően
- korlát: pontos áttekintést még nem sikerült szerezni a pontosságról (pozitívan fogalmazva)

Abstract

sasd

(same in Engilsh)

1. Bevezetés

Az emberiség fejlődése során mindig is fontos szerepet töltött be a helymeghatározás, hogy megtudjuk, merre járunk, merre kell továbbhaladnunk. Hosszú fejlődés vezetett az iránytűn, különböző térképeken át a mai csúcstechnológiákig, amilyen például a GPS azaz a Global Positioning System, magyarra fordítva Globális Helymeghatározó Rendszer.

Ezt a GPS rendszert az Amerikai Egyesült Államok Védelmi Minisztériuma fejlesztette ki katonai célokra. A katonai célú felhasználás mellett később megjelent a mindennapi életben történő felhasználás, és manapság már egy teljes körűen elterjedt technológiáról beszélhetünk. A GPS rendszer indulásakor igen nagy összegeket emésztett fel a 24 Navstar műhold fellövése miatt. Ennek ellenére manapság ingyenesen használható a műholdrendszer egy megfelelő vevőegységgel. A GPS megtalálható a drágább személyautókban, újabb okostelefonokban beépítve, vagy vásárolhatunk magunknak egy különálló GPS vevőkészüléket.

Felmerülhet a kérdés, ha ilyen olcsó és elterjedt helymeghatározó technológia a GPS rendszer, miért is kéne más technológiák felé fordulni? Azért, mert ennek a rendszernek is vannak hátrányai. Ilyen hátrányok például, hogy a GPS vevőkészülékek viszonylag magas ára, vagy például csak szabadtéren használható. Mi vehetné fel a versenyt tehát ezzel a technológiával, és mi javíthatná ki a hibáit? A GSM alapú helymeghatározás. Ez a technológia a már meglévő GSM mobilkészülékeken alapul és versenyképes helymeghatározást kíván biztosítani a GPS-el szemben. Magyarországon már most is több mint 10 millió mobil előfizető van, azaz több mint amennyi ember él az országban. A mobiltelefonok egyre okosabbak, egyre több szolgáltatást nyújtanak, GSM alapú helymeghatározás pedig egy igen népszerű és hasznos alkalmazása lehet a mobilkészülékeknek. A GSM alapú helymeghatározás kutatásával egyetemünkön régóta foglalkoznak a hallgatók és a tanárok és már eddig is számos szép sikert értek el. Az én munkám a csapatban földrajzi adatbázis készítése volt ehhez a GSM alapú helymeghatározáshoz, valamint ennek az adatbázisnak a felhasználás a helymeghatározás kutatásának támogatására.

A földrajzi adatbázisok földrajzi koordináták felhasználásával számos kutatási, vizsgálati, tesztelési lehetőséget kínálnak, ami a GSM alapú helymeghatározás kutatásakor hasznosnak bizonyult. Munkám során megismerkedtem GSM alapú helymeghatározás az egyetemi csapat által eddig elért eredményeivel, ezután megismerkedtem a földrajzi

adatbázisok használatával. Ezek után felépítettem a saját földrajzi adatbázisomat és ezt használtam a GSM helymeghatározási feladatok vizsgálatára.

2. Helymeghatározás

Ide még írnék egy kis felvezetőt a fejezettel kapcsolatban

2.1. Műholdas helymeghatározás (GPS)

„A NAVSTAR GPS (NAVigation Satellite Timing And Ranging Global Positioning System-globális helymeghatározó rendszer navigációs műholdakkal idő- és távolságmeghatározás útján) elvét az Egyesült Államokban dolgozták ki katonai navigációs célokra, 1973-ban. Az első műhold fellövésére 1978-ban került sor, a rendszer szolgáltatásai hivatalosan 1995-ben indultak meg. A GPS rendszer a felhasználó helyzetét távolságmérés alapján határozza meg. A működés alapfeltétele az idő igen pontos mérése és a Föld körüli pályán keringő műholdak helyzetének pontos ismerete. A technikai és technológia fejlődése éppen a múlt század 80-as, 90-es éveiben tette lehetővé, hogy e két feltételt egyszerre teljesíteni lehessen.” [1]

A rendszer legfontosabb jellemzői:

- A GPS rendszerben ismert helyzetű Föld körüli pályákon keringő műholdak jeleket sugároznak a Föld felszíne felé. A földi vevőkészülék ezeknek a jeleknek a mérési adataiból, illetve az általuk szállított információk feldolgozásából meghatározza a saját helyzetét. A rendszer tehát aktív műholdakkal és passzív földi vevőkészülékkel működik.
- A GPS rendszer működéséhez feltétlenül szükséges az, hogy a vevőkészülék antennája és a műholdak között ne legyen akadály, ez azt jelenti, hogy beltéri helymeghatározásra a GPS rendszer nem alkalmas.
- A GPS rendszer működésének alapfeltétele az időmérés pontossága. Minden műholdon igen pontos cézium és rubídium atomórák találhatók, melyek abszolút pontossága eléri a 10^{-13} - 10^{-14} értéket. Ez azt jelenti, hogy egy ilyen pontosságú óra kb. 300.000 – 3.000.000 év alatt késik vagy siet egyetlen másodpercet.

A GPS műholdak jele adatokat tartalmaz, melyek a vevőkészüléket tájékoztatják a műhold aktuális helyzetéről és a műholdon mérhető pontos időről. A rendszer minden műholdja szinkronizáltan működik, azaz óráik pontosan össze vannak hangolva, és jeleiket is pontosan azonos időben küldik a vevő felé. A távolságot a vevő igen egyszerűen határozza meg. Méri a jel érkezési idejét, és – ismerve a jel startjának időpontját – a jelterjedési idő kiszámítása után a fénysebesség ismeretében meghatározza a műhold és a vevőkészülék távolságát.

A GPS rendszer három alapvető alrendszerből épül fel:

- az űrszegmensből (műholdak),
- a felhasználói rendszerből (vevőkészülékek és szolgáltatások) és
- a vezérlőrendszerből (földi vezérlő- és monitorállomások).

Az űrszegmens teljes kiépítésben 24 műholdat tartalmaz. A műholdak számát és elrendezését úgy választották meg, hogy minél nagyobb legyen annak az esélye, hogy a Föld felszínén egy adott helyen legalább négy műholdat lehessen "látni" a vevőkészülék antennájával. Annak az esélye, hogy egy felhasználó négynél kevesebb műholdat lát egy adott helyen egy adott időben kb. 0,01%, azaz ilyen eset minden 10.000-dik felhasználóval fordul csak elő, de néhány perces várakozás után ezek a felhasználók is képesek legalább négy műholdjelet venni.

2.2. Mobilkommunikáció (GSM)

Ide még írnék egy kis felvezetőt a fejezettel kapcsolatban

2.2.1. A GSM rendszer alapjai

A GSM rendszer működésének áttekintése, a szabványok megvizsgálása fontos, hogy megértsük lehet-e egyáltalán helymeghatározást csinálni mobil eszközzel, és ha igen milyen módon. A telefon technikában már nagyon régen kialakult a gyártók és szolgáltatók együttműködési kényszere miatt az erős, jól működő szabványosítás. A mobiltelefonok rádiós interfészére vonatkozó 3GPP dokumentumokat kötelezően betartandó szabványnak tekinthetjük, és eszerint működnek a felhasználó és szolgáltató készülékei és berendezései.

[2]

A GSM telefonhálózatokban a szolgáltató egy meghatározott rácsszerkezetben helyezi el a kommunikációs csomópontokat, és egy-egy ilyen csomópont által kiszolgált besugárzott területet cellának neveznek. A szabványos rádiós kommunikációs eljárások kialakításánál úgy jártak el, hogy lehetőség legyen akár beszélgetés közbeni barangolásra is, azaz folyamatos beszélgetés alatt váltani a kiszolgáló bázisállomások között, akkor is, ha a váltás közben még szolgáltatót is váltani kell. A bázisállomások közötti váltást megfelelő körülmények között a mobiltelefon kezdeményezi. Ilyen bázisállomás váltást kizárólag akkor tud kezdeményezni, és megvalósítani egy mobiltelefon-készülék, ha a bázisállomások egyértelműen azonosíthatóak, és saját azonosítójukat folyamatosan minden körzetükben levő készülék tudomására is hozzák, erre szolgál a jelzési csatorna.

A készülék a rádióterjedési viszonyoknak folyamatos mérésével határozza meg és optimalizálja a cella választását. A készülékben elhelyezett belső algoritmus kiválasztja az optimális bázisállomást és figyeli a vételben bekövetkezett változásokat, melyek alapján folyamatosan frissíti tábláját, ami vezérli a belső algoritmuson keresztül a cellaváltást. Ahhoz hogy a mobil készülékünkkel hívást tudjunk indítani, vagy elérhetőek legyünk mások számára, a készüléknek csatlakoznia kell egy bázisállomáshoz.

Korlátozott körülmények között indítható segélyhívás a mobil készülékről akkor is, ha az nincs hálózathoz SIM kártyán keresztül csatlakoztatva, vagyis a jelzési csatorna hálózathasználat nélkül is hozzáférhető. A 3GPP TS05.08 szerinti adó és vételi paraméterek (MCC, MNC, LAC, CELL_id, RSSI, stb.) tartalmaznak olyan információkat, amiből lehetőség van becsülni azt, hogy éppen milyen távolságban tartózkodunk az ott található bázisállomásoktól.

2.2.2. GSM alapú helymeghatározás

„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.” [3]

GSM helymeghatározás aktuális cella alapján már ma is elérhető. Létezik a mobil szolgáltatók által üzemeltetett és tőlük megrendelhető helymeghatározási szolgáltatás, mely a GSM hálózat adottságainak kihasználására alapul. Ezen szolgáltatás alapja az, hogy a GSM hálózat folyamatosan tudja, hogy a bekapcsolt mobiltelefon pontosan melyik cellában van, és mivel a szolgáltató pontosan ismeri a saját maga által telepített cellák elhelyezkedését, így információt tud adni a készülék helyzetéről. A szolgáltató a saját maga által üzemeltetett hálózat változásait figyelembe tudja venni a helymeghatározáskor. A szolgáltatás igénybevétele külön szerződéssel valósítható meg, ennek megfelelően fizetési kötelezettség tartozik hozzá a felhasználó részéről, és csak a bekapcsolt, hálózatra bejelentkezett mobil készülékekről tud információt szolgáltatni.

A helymeghatározást javíthatjuk, ha nem csak a mobiltelefon által vett aktuális cella adatokat használjuk fel, hanem az összes látható cellát. Egy mobilkészülék a szabvány szerint

maximum 7 bázisállomással áll kapcsolatban[4], azonban csak egyet használ az aktuális telekommunikáció lefolytatására, ezt hívjuk aktuális cellának. A többi cella, azaz a maradék 6 db tartalékként szolgál, arra az esetre, ha romlana a vétel és esetleg a többi bázisállomástól kedvezőbb vételi szintek érhetőek el. Ezeknek a celláknak az információi a mobilkészülékből kinyerhetőek. A leggyengébb vételi teljesítményű állomások folyamatosan lecserélődnek újakra, illetve az aktuális cellánk is folyamatosan változik amennyiben mozgásban vagyunk és mozgás közben új bázisállomások jelennek meg a látóterünkben illetve némelyek eltűnnek, és a vételi szintek is folyamatosan változnak. Ezen információk alapján próbáltam a helymeghatározásomat felépíteni.

Néhány a mobiltelefonból kinyerhető adat:

- MCC: (Mobile Country Code) mobil országkód, melynek hossza 3 számjegy, és egyértelműen meghatározza a mobil előfizető hálózata szerinti országot. Magyarország mobil országkódja: MCC=216.
- MNC: (Mobile Network Code) mobil hálózati kód, melynek hossza két számjegy. Az MNC az MCC-vel együtt egyértelműen meghatározza a telefonszolgáltatást igénybe vevő végberendezés vagy előfizető honos hálózatát. Például: a T-mobile network kódja: MNC=30.
- LAC: (Location Area Code) 4 számjegyből álló azonosító, ami egy nagyobb terület azonosítására szolgál.
- CELL (vagy ci): (cell identifier) 4 hexadecimális 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 (vagy ARFCN): (Absolute Frequency Channel Number) Az a frekvencia, amin az adó sugároz.
- RSSI (vagy RXLEV): Received signal level of BCCH carrier (0..63). Az értéke a jel mért erősségéből dBm-ben, és egy offsetből áll.

3. Földrajzi adatok kezelése

Ide még írnék egy kis felvezetőt a fejezettel kapcsolatban

3.1. Térinformatika

A térinformatika (angolul GIS (Geographic Information Systems)) többféle oldalról megközelíthető. A hálózatok terjedésével egyre nagyobb hangsúlyt kap az információk elérését, 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)* [5]

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. [6]

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.

3.2. Adatbázis-kezelő rendszerek földrajzi kiterjesztése

Ide még írnék egy kis felvezetőt a fejezettel kapcsolatban

3.2.1. Áttekintés

A földrajzi adatok kezelése a számítástechnika fejlődésével, az adatbázis-kezelő rendszerek elterjedésével nagyon leegyszerűsödött. Manapság legegyszerűbben az adatbázis-kezelő rendszerekhez kapcsolva egyfajta kiterjesztésként érhetők el a földrajzi adatok kezelésére alkalmas rendszerek. A hagyományos objektum-relációs adatbázis-kezelőket felruházzhatjuk geometriai adatok kezelésére alkalmas kiegészítőkkel, amelyek általában a GIS nevet viselik.

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.)

A hagyományos adatbázis-kezelő rendszereket olyan eszközökkel lehet kiterjeszteni, amellyel hatékonyan lehet térbeli adatokat tárolni, és hatékonyan lehet térinformatikai adatokon lekérdezéseket végrehajtani. A két legelterjedtebb ilyen kiterjesztése az Oracle-nek a Spatial modul, a PostgreSQL-nek a PostGIS.

Az Oracle Spatial rendszer[7] az Oracle Database Enterprise Edition kiegészítőjeként kapható licence köteles rendszer. Az Oracle az informatika számos rétegét lefedő szereplő, nagy infrastrukturális háttérrel rendelkező vállalat. Szemben a PostgreSQL és annak kiterjesztése a PostGIS amely ugyan kisebb infrastrukturális háttérrel rendelkezik, azonban ingyenes szabad szoftver, és az Oracle Spatial-nál nagyobb földrajzi funkciókészlettel rendelkező rendszer. Számos külső megjelenítési rendszert támogat, mint például a Google Earth.

A választásom a PostGIS-el kiegészített PostgreSQL adatbázis kezelőre esett. A nagyszámú földrajzi függvény hasznos lehet munkám során, ingyenességével pedig a későbbiekben is használható lesz a rendszer.

3.2.2. PostgreSQL, 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 vállaltak, mint az amerikai DARPA, ARO és NSF.

„A PostgreSQL egyik legfontosabb erénye, hogy igyekszik minél teljesebben mértékben megvalósítani az SQL szabványt, ugyanakkor a fejlesztők munkáját segítő jó néhány kiterjesztést tartalmaz. A kereskedelmi relációs rendszerekben manapság általánosnak

tekinthető szolgáltatások (lekérdezési eredményekben végezhető halmazműveletek, a különféle SQL-parancsok szinte bármelyik részében előforduló kifejezésekbe beágyazható lekérdezések, mezők értéktartományának és egyediségének ellenőrzése, nézetek, számlálók, triggerek (valamint hasonló szerepű szabályok), többféle programozási nyelven elkészíthető tárolt eljárások stb.) a PostgreSQL-ben szintén rendelkezésünkre állnak.”[8]

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 találhatók.

A PostgreSQL egy szinte minden platformon működő nagyteljesítményű legendásan megbízható és stabil adatbázis szerver alkalmazás.

Néhány fontosabb technikai lehetőség: [9]

- 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 háttérét is szolgáltatathatják.

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 kiszűrhatja 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.

Egy térképi adatokat tartalmazó tábla létrehozására látható példa az 1. ábrán.

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

1. ábra - Tábla létrehozása 2 dimenziós geometriai mezővel

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.

A sorok beszúrását az SQL INSERT INTO paranccsal 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 2. ábrán látható módon alkalmazhatunk.

```
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));
```

2. ábra – Sor beszúrása geometriai adatokkal

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, erre egy példa látható az 3. ábrán.

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

3. ábra – MakePoint függvény használata

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

Sokféle lekérdezésre nyílik lehetőségünk. Első példa látható a 4. ábrán, ahol megkapjuk a 112-es azonosítójú vezeték hosszát.

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

4. ábra – Hossz számítás

Az 5. ábrán 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.

```
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;
```

5. ábra – Területszámítás

A vetületi rendszerek (SRID) kezelésére a PostGIS a térképi elemek leírása mellett egy kódot is tárol. 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. Én a 4326-os SRID-t használtam, mivel ez a WGS 84 (World Geodetic System) referencia rendszer alapján készült, amit a GPS rendszerek is használnak.

3.3. Google Earth és a KML formátum

A Google Föld (angol nevén: 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 megjeleníthettem azt a földgömbre vetítve a Google Earth-el.

A KML (Keyhole Markup Language) 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.

Egy egyszerű KML fájlra látható példa a 6. ábrán.

```
<?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>
```

6. ábra – Példa egy egyszerű KML fájlra

Egy példa látható egy hosszabb általam készített KML file-ra a 7. ábrán.

```

<?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>19.21766166666667,
47.43843166666667 19.21562833333333,47.4396 19.20874,47.443655
19.20802,47.44416333333333 19.20850666666667,47.44402333333333
19.20942833333333,47.443525 19.21216666666667,47.44185166666667
19.21766166666667,47.43843166666667</coordinates></LinearRing></outerBo
undaryIs></Polygon>"
  </Placemark>
</Document>
</kml>

```

7. ábra – Példa egy hosszabb általam készített KML fájlra

Az első példában (6. ábra) 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 (7. ábra) 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.

3.4. OpenStreetMap (OSM)

OpenStreetMap (rövidítve: **OSM**) egy szerkeszthető térkép az egész világról, amely főleg GPS-nyomvonalakból kiindulva épült fel, és szabad tartalom licenccel adták ki. „Ez nem csak egy szoftverprojekt. Elmegyünk a számítógép-képernyőink elöl, és felmérjük a városokat és a vidéket, hogy létrehozzuk térképeinket. Ez egy nagyon nagy közösségi térképkészítési erőfeszítés. Ennek támogatására egy wiki stílusú együttműködő szerkesztő szoftvert használunk, ami azt jelenti, hogy a térképeink folyamatosan nagyobbra nőnek és jobbak lesznek. Ha van egy GPS-egységed, akkor feltölthetsz néhány rögzített nyomvonaladat. Szintén származtathatunk térképeket a Yahoo! Aerial Imagery-ből.” [10]

Az OpenStreetMap licenc szabad (vagy majdnem szabad) hozzáférést biztosít a térképeinkhez és minden mögöttes térképadathoz, és a projekt célja, hogy elősegítse ezen adatok új és érdekes használatát. A térkép még nincs kész, de van teljes lefedettségünk néhány helyről (ahol a legteljesebben készítették a térképet). Az emberek és a kereskedelmi vállalatok már elkezdtek használni ezeket az adatokat.

Mért fejlesztik az OpenStreetMap-ot? A geográfiai adatok nem ingyenesek a világ legtöbb részén. Ahhoz, hogy ilyen adatokat megkapjunk, legtöbbször fizetni kell különféle kormányzati/önkormányzati szervezeteknek. Ha térképet vásárolsz akkor az hamar elévül, új utak épülnek, területeket átminősítenek, így akár egy év múlva már valótlan lesz a térképed. Az olcsó, elterjedt GPS technológiának köszönhetően te magad is készíthetsz saját térképet. Együttműködve másokkal, egy kis közösségi munkával létrehozható ilyen térkép mint az OpenStreetMap. A pontosság A Google Maps és más hasonló térképszolgáltatók térképei sem ingyenesen teljesen, hiszen az adatbázist nem szolgáltatják ki ami a térképük alapját képezi, csak egy megjelenítő felület használható szabadon.

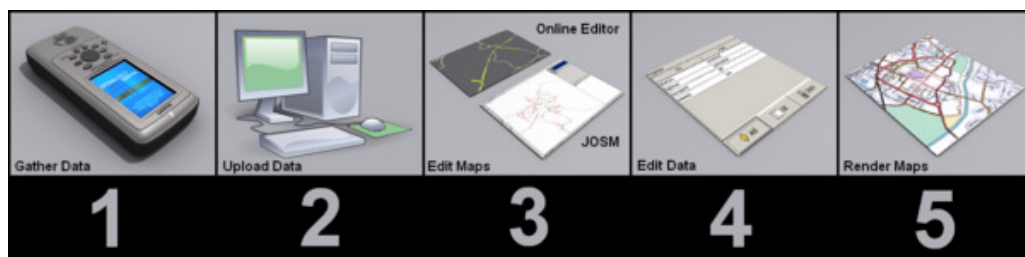
Milyen adatokat adunk hozzá? Rengeteg adatot gyűjthetünk össze és tehetünk az OSM-be: az olyan általános dolgoktól, mint például az utcanevek egészen az apró részletekig, beleértve olyan dolgokat is, mint a parkok, postaládák, sövények és kőhalmok. Mindenki mást talál fontosnak, attól függően, hogy milyen módon közlekedik. Ha úgy érzed, hogy valami segíthet az embereknek a tájékozódásban, jelöld be a térképen! A térképszerkesztők általában fokozatosan növelik a részletességet, de az alapokkal kezdenek. Így a fő úthálózat általában egy jó kiindulópont, és a kevésbé fontos dolgok, mint például a gyalogutak vagy a postaládák az utolsók között kerülnek hozzáadásra. Ez is erősen emberfüggő, nincs egyféle, mindenkire érvényes módszer.

Az OSM adatok geometriai típusai:

- Pontok (Nodes): A pontok azok, amelyeket arra használnak, hogy vonalszakaszokat rajzoljanak közöttük.
- Vonalak (Ways): Pontok egy rendezett listája, vonalszakaszokkal összekötve jelenik meg a szerkesztőben.
- Zárt vonalak (Closed Ways): A zárt vonalak olyan vonalak, amelyek egy egész hurkot alkotnak. Területek leírására használják, mint például parkok, tavak vagy szigetek.

A térkép készítésének 5 lépése (8. ábra):

1. Adatok gyűjtése
2. Adatok feltöltése
3. OSM adatok készítése/szerkesztése
4. Adatok címkézése, és részletek hozzáadása
5. Térképek megjelenítése és használata!



8. ábra - OSM térkép készítése

4. GSM helymeghatározás

Ide még írnék egy kis felvezetőt a fejezettel kapcsolatban

4.1. Bevezetés

A helymeghatározás alapjául tehát a mobilkészülék által látható maximum 7 bázisállomás és ezek vételi jelszintjei szolgálnak. A GSM helymeghatározáshoz ellentétben például a GPS rendszerrel nem tervezünk semmilyen nagyméretű beruházást megvalósítani, sem a háttér infrastruktúrát tekintve (GPS esetén például műholdak), sem a kézi eszközt tekintve, csupán egy ma is kapható okostelefont szeretnénk használni a helymeghatározáshoz. Nem kívánunk együttműködni a mobiltelefon szolgáltatókkal sem, hiszen ez mindenképp valami díjat vonna maga után.

A módszerhez szükséges tehát, hogy az adott terület ahol a helymeghatározást akarjuk használni előzőleg egy speciális eszközzel fel legyen térképezve és ezek az adatok fel legyenek töltve egy adatbázisba. Ez a speciális eszköz egy GPS vevővel és egy GSM vevővel van ellátva, és feladata, hogy a két vevőből egyszerre érkező jeleket letárolja. Az így eltárolt mérési adatokat egy adatbázisba kell aztán betölteni. Ehhez a módszerhez tehát fel kell térképezni adott esetben egy város egész utcarendszerét, ami óriási feladat. Ezt sajnos megkerülni nem tudjuk, mert a másik egyszerűbb módszer szerint a szolgáltatóktól kellene begyűjteni a bázisállomások pozícióit, és hozzá tartozó sugárzási karakterisztikáit, ám ezeket a szolgáltatók szigorú üzleti titokként kezelik, és nem adják ki senkinek. Marad tehát a területek felmérése, amely akár önkéntesek segítségével is megvalósítható, ahogy azt láthattuk az OpenStreetMap adatok gyűjtésénél, ami önkéntes alapon történik.

A helymeghatározáshoz szükséges adatokat kétféle módon kaphatja meg a mobilkészülék. Egyik esetben online módon mobilinternet segítségével internetkapcsolatban áll egy adatbázis szerverrel, amelyről lekérdezi a koordinátákat a vett bázisállomások jelei alapján és a készülék csak a térképes megjelenítést végzi. Ekkor feltételezzük, hogy a felhasználónak mobilinternetezésre alkalmas készüléke van és rendelkezik mobilinternet előfizetéssel. Ilyenkor nem feltétlenül szükséges az adatok tömörítése, használhatunk egy tetszőlegesen bő adatbázist, amely egy szerveren van elhelyezve. Egy ilyen adatbázis szerver bérlete, vagy megvásárlása ugyan költséget jelent, de ez a költség még vállalható, egy ilyen rendszer üzemeltetésekor.

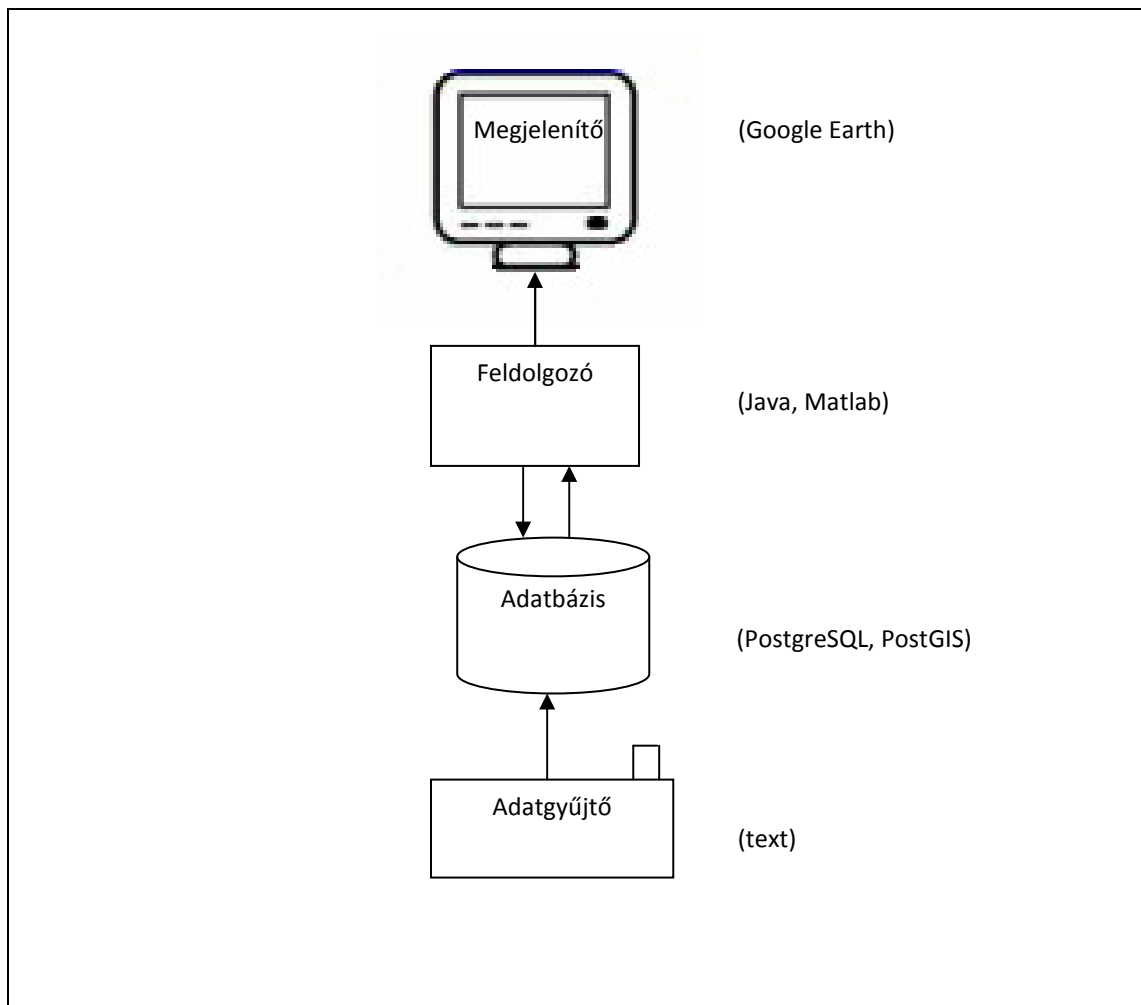
Másik esetben offline módon működik a helymeghatározás, ilyenkor nem szükséges mobilinternet hozzáférés, viszont jelentősen korlátozva vagyunk. Korlátozva vagyunk számítási teljesítményben, memóriában, akkumulátor kapacitásban. Ebben az esetben szükséges egy egyszerű adatbázis készítése a mobilkészülékre. Ez az adatbázis csak korlátozott számú adatot tud tárolni ezért mindenképpen szükség van drasztikus méretű csökkentésre az adatbázisban. Az adatbázis méretének csökkentése, az adatbázis tömörítése nagyon bonyolult mérnöki probléma, hiszen tömörítés közben figyelembe kell venni, hogy a helymeghatározás pontossága ne romoljon drasztikusan, hiszen akkor az egész rendszer használhatatlanná válik.

4.2. Adatkezelés

Ide még írnék egy kis felvezetőt a fejezettel kapcsolatban

4.2.1. Architektúra

Az architektúra a helymeghatározási feladatok vizsgálatához igazodva készült el. Az architektúra fő részei a mérésadatgyűjtő eszközök, az adatbázis, a feldolgozó rendszer és a megjelenítő rendszer. Erről egy architektúra diagram látható a 9. ábrán.



9. ábra - Architektúra diagram

A mérésekben magam is részt vettem, egy mérésadatgyűjtő eszközzel, amit a csoport egyik tagja korábban készített. Az eszköz fő részei: GSM vevőegység antennával, GPS vevőegység antennával, processzor és memória. A készülék fogadta a külvilágból érkező GSM és GPS jeleket, amelyeket egyszerre letárolta 1mp-es időközönként. Így kaptunk egy GPS koordinátát a hozzá tartozó, adott pontban fogható GSM bázisállomások jeleivel. Egyszerre egy időben maximum 7 bázisállomást láthat egy szabványos GSM készülék.

Egy már meglévő adatbázis, a kutatócsoport által készített MySQL adatbázis már a kutatásba való csatlakozásomkor rendelkezésemre állt. Ez az adatbázis sok mérési adatot már tartalmazott, ami a csoporthoz való csatlakozásom előtt lett gyűjtve, így azok már egy meglévő MySQL adatbázis kezelőben voltak feltöltve a csoport szerverén. Mivel már volt egy adott adatbázis séma, nem láttam hasznát, hogy saját ettől különböző struktúrát találjak ki, illetve a mérési adatok is. Így a meglévő struktúrát használva exportáltam az adatokat a MySQL adatbázisból és feltöltöttem az a saját Postgresql adatbázisomba, amely a PostGIS

kiterjesztéssel kiegészítve képes földrajzi adatok kezelésére amire az alap MySQL adatbázis nem lett volna használható.

A feldolgozó rendszer főként a Postgresql-be épített SQL lekérdező használata jelentette, ahol a parancsokat kézzel írtam, majd futtattam. A kapott eredményt közvetlenül egy kml fájlba illesztettem, vagy egyszerű Java programokkal, vagy Matlabbal feldolgoztam, majd a feldolgozott adatokat illesztettem egy kml fájlba.

A megjelenítésnél pedig ezt a kml fájlt nyitottam meg Google Earth-el. A kml formátum tanulmányozása után, feldolgozáskor már úgy készítettem el a fájlokat, hogy megjelenítésnél segítsen a képek értelmezését színezéssel, jelölések használatával.

A teljes műveletsorozat tehát így néz ki:

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

4.2.2. Szoftverkomponensek

Az adatbázist az ingyenesen elérhető PostgreSQL adatbázis kezelő rendszerben készítettem. Ehhez először letöltöttem a PostgreSQL 8.3-as verzióját a <http://www.postgresql.org/download/> oldalról. Ezt én Windows XP operációs rendszerre telepítettem, de természetesen a PostgreSQL-nek, mint a legtöbb ingyenes szoftvernek a Linux az alapvetően javasolt környezete. Ezután telepítettem a PostGIS nevű kiterjesztés 1.3-as verzióját a beépített Application Stack Builder segítségével. Később frissítenem kellett a PostGIS verzióját, mert számomra hasznos, új függvények jelentek meg, amelyek csak az 1.5-ös verzióban elérhetők. Ezért elindítottam Application Stack Buildert és letöltöttem vele az 1.5-ös verziót. Ezzel azonban még az én adatbázisom nem kapta meg az új függvényeket, le kellett futtatnom egy SQL parancsot ami arra hivatott, hogy frissítse a függvényeimet a saját adatbázisomban. Ezért lefuttattam ezt az SQL-t amit a frissítéskor kaptam:

```
C:\Program Files\PostgreSQL\8.3\share\contrib\postgis-1.5\postgis_upgrade_13_to_15.sql
```

Ezzel már az én adatbázisom is a frissített 1.5-ös verzió szerint működött.

A PostGIS remek dokumentációval rendelkezik, amelyet munkám során végig használtam, ezen felül pedig a felhasználók nyílt levelezési listái is sokat segítettek, ha elakadtam valahol, vagy nem éretem valaminek a működését. [11]

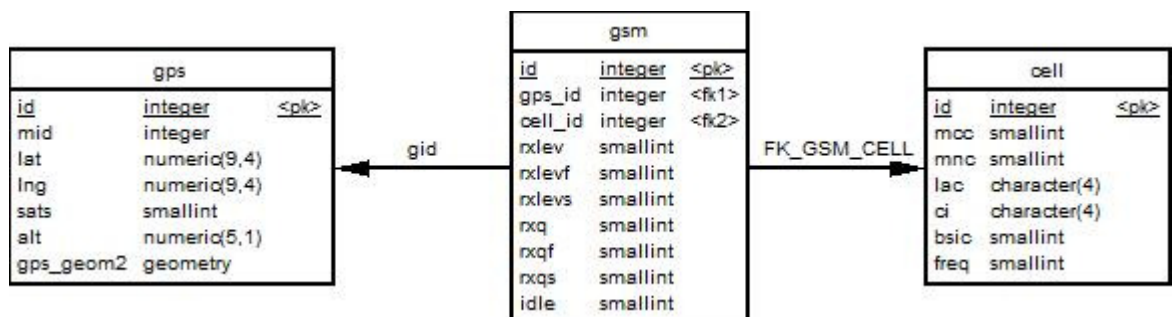
Az adatbázis és a földrajzi kiterjesztés telepítése után el kellett készítenem saját adatbázisomat, amely használni tudta a földrajzi kiterjesztéseket. Ez az adatbázis a GSM nevet kapta. Ez adatbázist a következő SQL parancs lefutatása után jött létre:

```
CREATE DATABASE GSM TEMPLATE=template_postgis
```

Az így létrejött adatbázis rendelkezik a földrajzi adatbázis kiterjesztéseivel, a speciális függvényekkel, a vonatkoztatási rendszerrel (SRID) és tudja kezelni a megfelelő geometriai típusokat.

4.2.3. Adatbázis séma

Az adatbázis sémadiagramja látható a 10. ábrán.



10. ábra - Sémadiagram

A táblák oszlopainak jelentései:

- GPS tábla:
 - id: Ez az ID azonosít minden egyes GPS mérést.
 - mid: méréshez tartozó egyéb adatok ID-je.
 - lat (Latitude): GPS koordináta szélességi foka.
 - lng (Longitude): GPS koordináta hosszúsági foka.
 - sats (Number of Satellites): a látott GPS műholdak száma
 - alt (Altitude): a tengerszint feletti magasság
 - gps_geom2: GPS koordináták tárolása geometriai (2D) pontként.
- GSM tábla:
 - id: Ez az ID azonosítja a GPS vételhez tartozó GSM bázisállomásokat.

gps_id: Kapcsolat a GPS táblához.

cell_id: Kapcsolat a CELL táblához.

rxlev (Received signal level): Értéke: 0..63 Az értéke a jel mért erősségéből dBm-ben és egy offsetből áll.

- **CELL tábla:**

id: Ez az ID azonosítja az önálló cellákat.

mcc (Mobile Country Code): mobil országcód, melynek hossza 3 számjegy, és egyértelműen meghatározza a mobil előfizető hálózata szerinti országot. A mobil országcódokat az ITU jelöli ki. Magyarország mobil országcódja: 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.

lac (Location Area Code): A 4 számjegyből álló azonosító, ami egy nagyobb terület azonosítására szolgál.

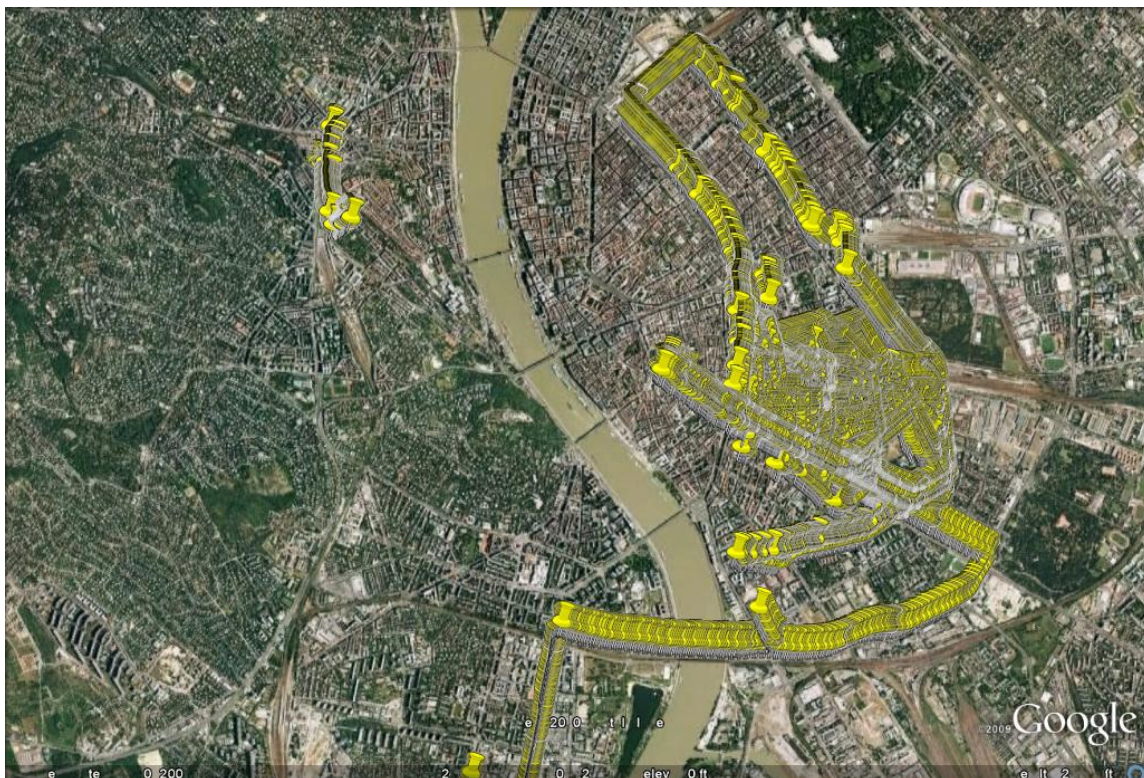
ci (Cell Identifier): 4 hexadecimalális számjegyből álló azonosító ami azonosítja a cellát.

bsic (Base station identity code): a mobil készüléket segíti a különböző szomszédos bázisállomások megkülönböztetésében. A BSIC egy úgy nevezett „színcód” ami azt jelenti, hogy a szomszédos cellák más képzeletbeli színnel vannak jelölve, és nem lehet egymás mellett két azonos színű cella.

freq (Absolute Frequency Channel Number): Az a frekvencia amin az adó sugároz, ha ennek az értéke 0 akkor az összes ehhez a cellához tartozó adat is 0.

4.2.4. Adattartalom

A GSM mérési adatokat, az adatbázisból lekérdezéssel érhetjük el. Egy lekérdezés eredménye látható a 11. ábrán, amelyen Budapesten eddig összegyűjtött mérési adatok GPS koordinátáinak pontjai láthatóak.

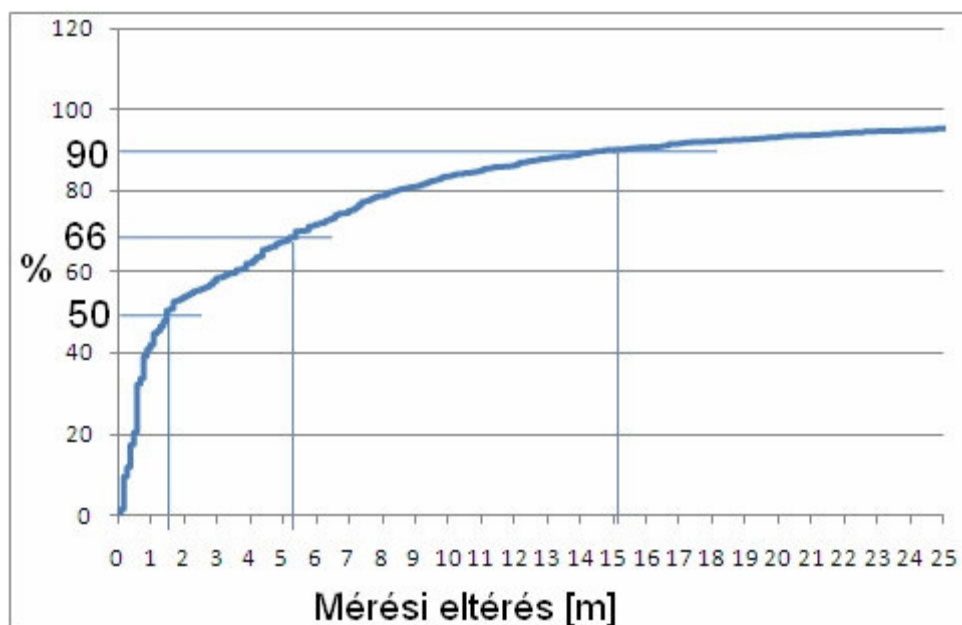


11. ábra - Budapesti mérési adatok

4.3. Helymeghatározás

A helymeghatározást mindvégig az adatbázisomra támaszkodva hajtottam végre. Egyes esetekben elméletben ugyan szűkítettem az elérhető adatok körét, de a gyakorlatban az összes tábla összes adata rendelkezésre állt. Az itt felsorolandó helymeghatározási módszerekhez tehát mindenképpen előzetesen gyűjteni kell az adott területről mérési adatokat, ezeket adatbázisba tölteni, majd ezután lehet helymeghatározást végrehajtani.

Egy kültéren elvégzett mérés pontosságának eredményei látható a 12. ábrán^[2], azaz hány százalékban volt a mérési pontosság az adott távolság alatt. Az esetek 90%-ban 15.1m-nél pontosabb volt a becslés.



12. ábra - Kültéren elvégzett mérések pontossága

Az 12. ábrán láthatjuk azt, hogy 4000 mintából vett teszt esetén a minták 50%-ban 1,5m, 66%-ban 5,3m és még a 90%-ban is 15,1m alatt volt a mérésünk pontossága. A tesztelés során a Budapest VIII kerületben dolgoztunk, ahol a magyarországi viszonyokhoz képes nagy a bázisállomások sűrűsége, ami jelentősen javítja a mérés pontosságát.

4.3.1. Alapeset

Alapesetnek vesszük azt az esetet amikor a mobilkészülékünk maximum 7 bázisállomástól érkező jeleket fogad, és ezeknek jelszintjeit is felhasználjuk a helymeghatározáshoz. A jelszintek 0 és 63 közötti értékeket vehetnek fel. Ha nagyon kicsi, vagy nem fogható a jel akkor az értéke 0-hoz közelebb, ha pedig maximálisan fogható akkor 63 az értéke. A 7 bázisállomást egy időben általában csak rövid ideig látja a mobiltelefon, mivel a telefon törekszik a minél erősebb jelű bázisállomásokot figyelni és ahogy mozgunk, távolodunk, közeledünk folyamatosan cserélődik a megfigyelt 7 bázisállomás.

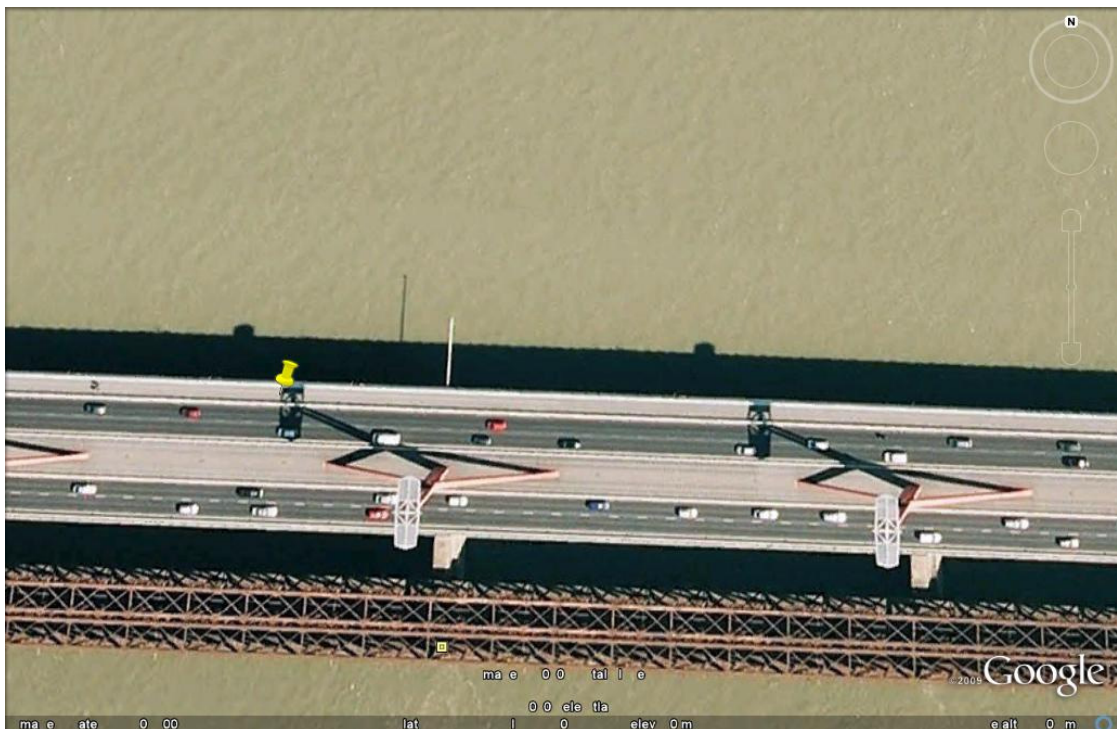
Tovább javíthatjuk tehát a helymeghatározást, ha nem csak a bázisállomások azonosítóját, hanem a bázisállomástól érkező jelszinteket is figyelembe vesszük, hiszen ezeket is veszi a mobilkészülékünk. A következőkben egy ilyen módszerről lesz szó.

A módszer lényege, hogy a mobiltelefonunk által vett adók azonosítóját (cell id) és az azonosítókhoz tartozó jelszinteket (rssi) egy csomagba tesszük (például egy megfelelően kialakított formátumú fájlba). Ezt a csomagot átküldjük a mobilkészülék internetének

segítségével egy központi adatbázis-kezelő rendszernek. Most az online esetet vizsgáljuk, vagyis amikor a helymeghatározást használó személynek van mobilinternet elérése. Az adatbázis szerver a kapott bázisállomás és jelszint adatok alapján visszaküldi a mobilkészüléknek azt a pontot vagy pontokat ahol az átküldött adótornyokat az átküldött jelszinteken az adatbázis alapján fogni lehetett. Ha nem talál az adatbázisban egyetlen pontot sem, akkor bővíti a lekérdezés halmazát oly módon, hogy a jelszintek köré egy intervallumot képez. Például első lépésben -1 és +1-es eltérést megenged a lekérdezésnél a kapott jelszintektől. Ha kapott eredményül legalább egy pontot talál, akkor sikerrel járt, ha nem akkor tovább bővíti az intervallumot. Szélsőséges esetben, akár el is hagyhatja a jelszintek figyelembevételét, ha az adatbázisunk annyira hiányos, hogy az adott helyet nehezen, nagy pontatlansággal tudjuk csak a helymeghatározást végezni. Ha talált a lekérdezés legalább egy pontot, akkor a pontnak, vagy pontoknak az adatbázis szerver kiszámolja a középpontját és ennek koordinátáit küldi el a mobiltelefon számára. A mobiltelefonnak fogadni kell ezt a csomagot, majd megjeleníteni egy térképen.

Erre egy példát fogunk látni a következőkben.

(6BS Lágymányosi híd)



13. ábra



14. ábra



15. ábra

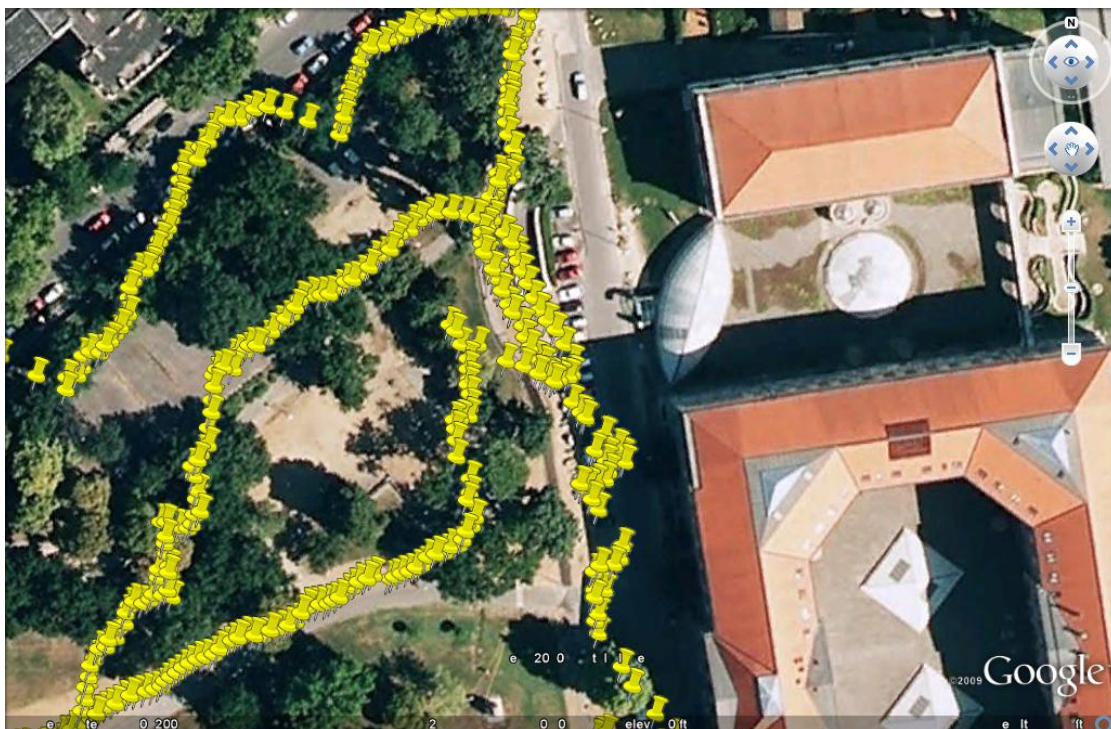


16. ábra

4.3.2. Jelszintek figyelembevétele nélkül

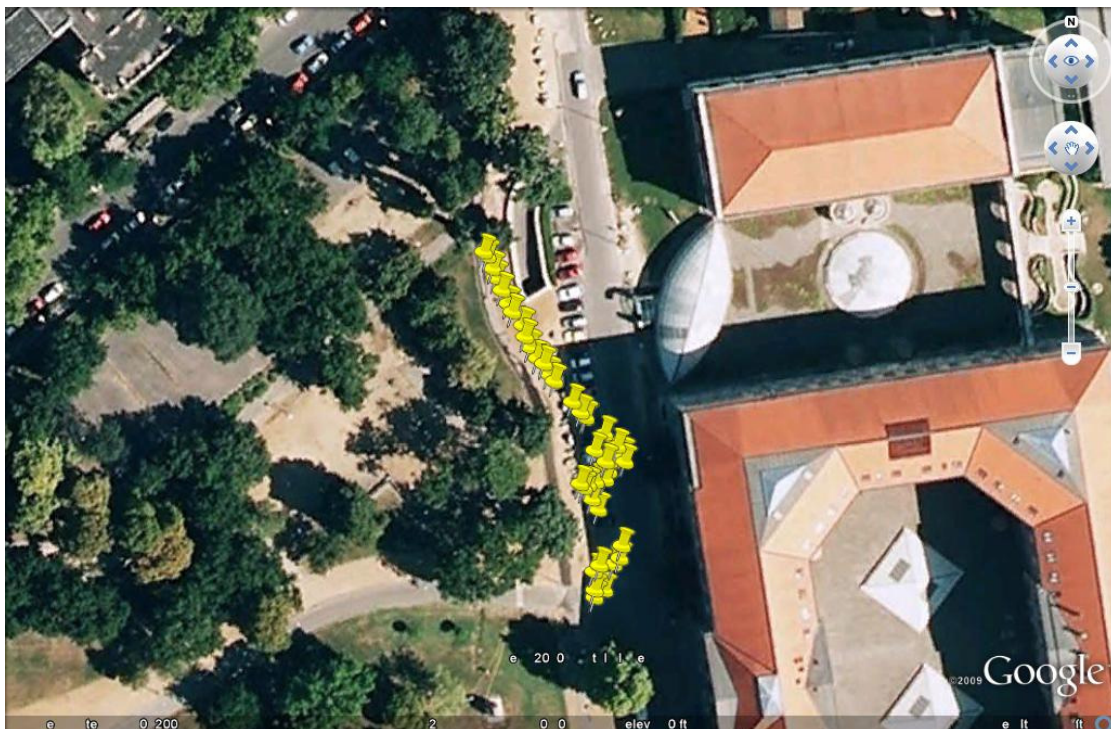
Ez a módszer az előző pontban tárgyalt módszerrel részben megegyező módszer. Itt csak az aktuális 7 bázisállomást használjuk helymeghatározáshoz, és nem vesszük figyelembe a vételi jelszinteket. Erre a szűkítésre azért van szükség, mert itt már offline módon határozzuk meg a pozíciónkat, azaz nem használunk más külső forrást csak a mobilkészülékünk erőforrásait.

A felmért terület egy részlete látható a 17. ábrán.



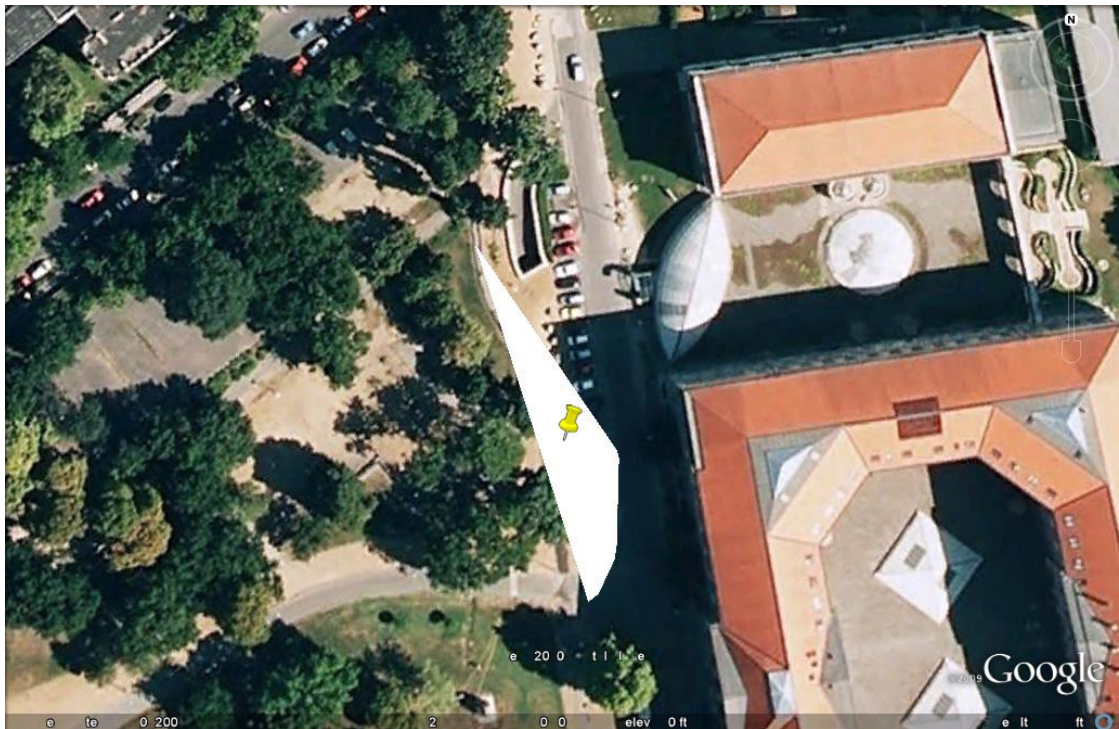
17. ábra - Egy felmért terület

Ebből az adatbázisból lekérdezéssel kiválasztunk olyan pontokat ahol a 7 db adott bázisállomás (jelen esetben a következő a 168, 157, 570, 573, 614, 618, 613 id-jű bázisállomás) együtt látszik, ami az 18. ábrán látható.



18. ábra – 7 bázisállomás itt együtt látszik

Majd ezeknek a pontoknak vesszük a középpontját, amely a 19. ábrán látható.



19. ábra – A 7 bázisállomás által besugárzott terület középpontja

Ezzel a módszerrel egy erősen leszűkített adathalmazon végezhetünk helymeghatározást. Ha az egész adatbázist ilyen módon leszűkítjük, hogy csak az egyszerre együtt látható 7 (esetenként csak 6 vagy kevesebb) bázisállomás által besugárzott terület középpontját tároljuk el, akkor ezzel valószínűleg megvalósíthatóvá válik a helymeghatározás egy ma is használt mobilkészüléken. Ez azonban további vizsgálatokat igényel, amelyet a csoport más tagja végeztek. A pontosság jelentősen romlik. A fenti példán 50 méteren beleüli a helymeghatározás pontossága városi környezetben, ezt azonban általánosságban nem jelenthetjük ki. További vizsgálatot igényelne az ezzel a módszerrel elérhető pontosság bizonyítása.

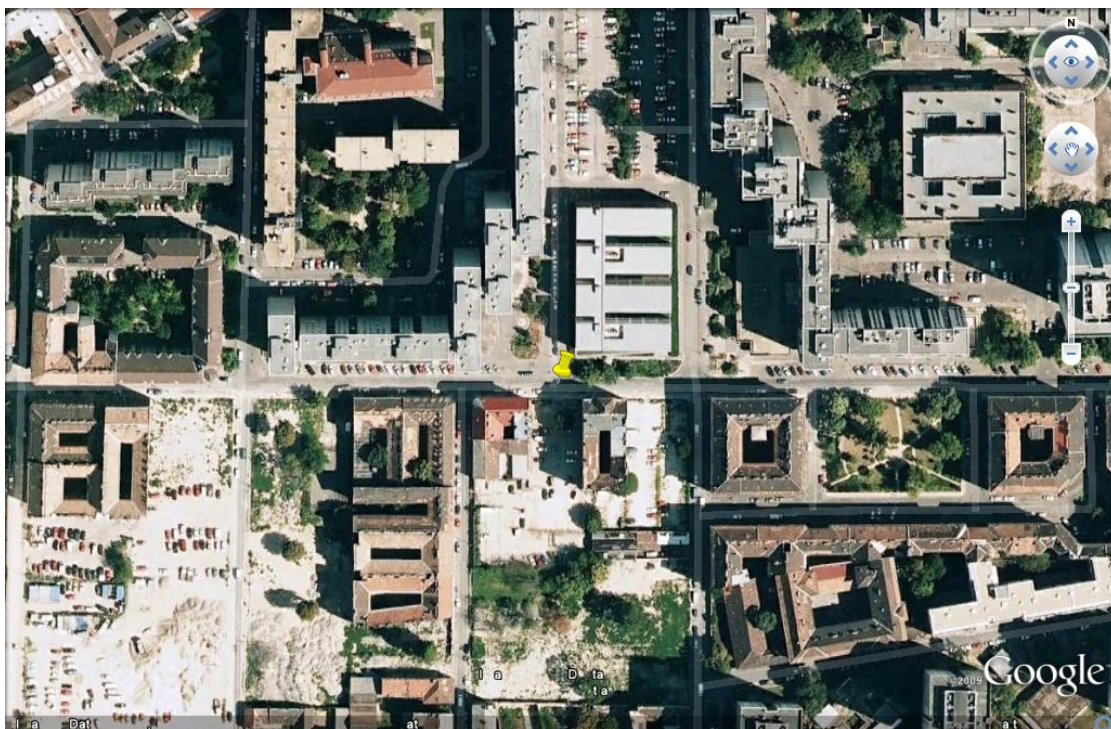
4.3.3. Főkomponens analízis segítségével

A főkomponens-analízist arra szeretném felhasználni, hogy az adatokat tömörítsem. Ez a tömörítés veszteséges lesz, de így könnyebben lehet egy térképrészletet elhelyezni a mobilkészüléken, könnyebben lehet a helymeghatározást a szűkös erőforrásokon megvalósítani.

A főkomponens-analízis (principal component analysis; PCA) módszer báziscserét alkalmazva minimális információvesztés mellett csökkenti egy vektorhalmaz dimenziószámát. A módszert Pearson dolgozta ki az 1900-as évek elején. A dimenziószám csökkentésének haszna, hogy a kapott vektorhalmaz kisebb költséggel és gyorsabban dolgozható fel. A PCA-módszer a dimenziószám csökkentését kisebb elemszámú bázisra való áttéréssel éri el. A báziscserét jelentő koordináta-transzformációnál megváltozik a pontok egymáshoz viszonyított helyzete, és ez információvesztést okoz. Az átalakítás során ezért törekedni kell arra, hogy a pontok közötti távolságviszonyok a redukált térben a lehető legjobban közelítsék az eredeti térben meglévő viszonyokat. Intuitíve érezhető, hogy a távolsági viszonyok megőrzése és a csoportképzés szempontjából az a dimenzió a legfontosabb, ahol a pontok a legjobban szeparálódnak. A PCA-eljárásban ezért úgy választjuk ki az új bázisvektort, hogy a mintahalmaz minden irányában a lehető legjobban terüljön szét. Ekkor a mintapontoknak az új b bázisvektorra vett koordinátaértékei is maximális szórást mutatnak. [12]

Az analízis során az eredeti paraméterekből (azok lineáris kombinációjával) új paramétereket hozunk létre, melyek segítségével olyan egymásra merőleges koordináta-tengelyek kaphatók, ahol az így kapott adatok sorrendben a lehető legtöbb varianciát veszik fel. Az utolsóként megkapott tengelyekre így már csak nagyon kevés variancia marad, így azok a paraméterek nem fontosak, elhagyhatók.

Főkomponens-analízist hajtottam végre a saját adatbázisom egy részén, amelynek lépéseit fogom most bemutatni, majd pedig értékelni fogom az eredményét. Első lépésként az adatbázisból kell lekérdezni egy adott területre vonatkozó földrajzi koordinátákat és GSM jelszint adatokat. Kiválasztottam egy pontot a térképen, ahol sok GSM mérési adat állt rendelkezésemre az adott pont környezetében. Ez a pont egy az Egyetem mellett lévő pont lett, amely a 20. ábra közepén látható.



20. ábra - Közepont

Ez után lekérdeztem a közép pont körül lévő pontokat 300 méteres átmérőben, hogy megnézzem elegendő adat lesz-e a PCA analízis végrehajtásához. Ezek a pontok láthatóak a 21. ábrán.



21. ábra - Pontok az analízishez

A lekérdezéshez használt SQL parancs olvasható a 22. ábrán.

```
SELECT gps.id, ST_AsKML(gps.gps_geom2)
FROM gps
WHERE ST_Distance(gps.gps_geom2,
ST_SetSRID(ST_Point(19.079261,47.486752),4326), true)<150;
```

22. ábra - Pontok lekérdezése

A pontok száma szemrevételezve elegendőnek látszik az analízis végrehajtásához. Ez után lekérdeztem ezeken a mérési pontokon milyen GSM bázisállomások látszódnak. Ez a lekérdezés látható a 23. ábrán.

```
SELECT distinct(gsm.cell_id)
FROM gps, gsm
WHERE gps.id=gsm.gps_id
AND ST_Distance(gps.gps_geom2,
ST_SetSRID(ST_Point(19.079261,47.486752),4326), true)<150;
```

23. ábra – Bázisállomások lekérdezése

Eredményül 62 db bázisállomást kaptam a következő sorszámokkal:

840, 841, 844, 845, 846, 847, 848, 849, 852, 853, 855, 856, 857, 858, 859, 860, 861, 862, 863, 864, 865, 866, 868, 871, 872, 874, 878, 879, 892, 917, 920, 926, 927, 929, 931, 959, 964, 966, 967, 968, 969, 970, 971, 972, 973, 974, 975, 976, 982, 1011, 1013, 1079, 1080, 1208, 1211, 1212, 1213, 1214, 1306, 1367, 1368, 1369

Ezekről a bázisállomásoktól vett jelszintek fogják a főkomponens analízis alapját adni. A következő összetett lekérdezés már megmutatja, hogy melyik pontban melyik bázisállomás mekkora vételi szinttel látható. Ez a lekérdezés egy kereszttáblás lekérdezés (crosstab, vagy más néven pivot table lekérdezés) melynél egyszerre két változó változtatásával táblázatos formában kapjuk a lekérdezés eredményét. Ez a két változó, a GPS tábla id-je (függetlenül, azok a pontok, amelyek az x. ábrán láthatóak) és a GSM tábla bázisállomásai (vízszintesen, a x. ábra bázisállomásai). A táblázatot pedig a az adott helyen adott bázisállomástól jövő vételi jelszint nagysága tölti ki, amelyből egy részlet látható a x. ábrán.

ID	BS_1	BS_2	BS_3	BS_4	BS_5	BS_6	BS_7	BS_8
31227	45	46				31		
31228	45	46				31		
31229	45	46				31		
31230	45	46				31		
31233	32		31			38		
31234	32		31			38		
31235	32		31			38	32	
31236	32		31			38	32	

24. ábra – Bázisállomások vételi szintjei adott pozíciókon (táblázatrészlet)

A lekérdezés amely a táblázatot előállította pedig a x. ábrán látható.


```

SELECT * FROM crosstab('
SELECT
    ta.gps_id ,
    ta.cell_id,
    tb.rxlev
FROM
    (SELECT gps_id,
        cell_id
    FROM
        (SELECT DISTINCT gps_id
        FROM
            gsm,
            gps
        WHERE
            gps.id=gsm.gps_id
        AND
            ST_Distance(gps.gps_geom2,
ST_SetSRID(ST_Point(19.079261,47.486752),4326), true)<150
        ORDER BY
            gps_id
        )
        t1,
        (SELECT DISTINCT cell_id
        FROM
            gsm,
            gps
        WHERE
            gps.id=gsm.gps_id
        AND
            ST_Distance(gps.gps_geom2,
ST_SetSRID(ST_Point(19.079261,47.486752),4326), true)<150
        ORDER BY
            cell_id
        )
        t2
    )
    ta
LEFT OUTER JOIN
    (SELECT gsm.gps_id ,
        gsm.cell_id,
        gsm.rxlev
    FROM
        gsm,
        gps
    WHERE
        gps.id=gsm.gps_id
    AND
        ST_Distance(gps.gps_geom2,
ST_SetSRID(ST_Point(19.079261,47.486752),4326), true)<150
    AND
        (gsm.cell_id=840 OR gsm.cell_id=841
OR gsm.cell_id=844 OR gsm.cell_id=845
OR gsm.cell_id=846 OR gsm.cell_id=847
OR gsm.cell_id=848 OR gsm.cell_id=849
OR gsm.cell_id=852 OR gsm.cell_id=853
OR gsm.cell_id=855 OR gsm.cell_id=856
OR gsm.cell_id=857 OR gsm.cell_id=858
OR gsm.cell_id=859 OR gsm.cell_id=860
OR gsm.cell_id=861 OR gsm.cell_id=862
OR gsm.cell_id=863 OR gsm.cell_id=864
OR gsm.cell_id=865 OR gsm.cell_id=866
OR gsm.cell_id=868 OR gsm.cell_id=871
OR gsm.cell_id=872 OR gsm.cell_id=874
OR gsm.cell_id=878 OR gsm.cell_id=879
OR gsm.cell_id=892 OR gsm.cell_id=917
OR gsm.cell_id=920 OR gsm.cell_id=926
OR gsm.cell_id=927 OR gsm.cell_id=929
OR gsm.cell_id=931 OR gsm.cell_id=959
OR gsm.cell_id=964 OR gsm.cell_id=966
OR gsm.cell_id=967 OR gsm.cell_id=968
OR gsm.cell_id=969 OR gsm.cell_id=970
OR gsm.cell_id=971 OR gsm.cell_id=972
OR gsm.cell_id=973 OR gsm.cell_id=974
OR gsm.cell_id=975 OR gsm.cell_id=976

```

```

OR      gsm.cell_id=982 OR      gsm.cell_id=1011
OR      gsm.cell_id=1013 OR      gsm.cell_id=1079
OR      gsm.cell_id=1080 OR      gsm.cell_id=1208
OR      gsm.cell_id=1211 OR      gsm.cell_id=1212
OR      gsm.cell_id=1213 OR      gsm.cell_id=1214
OR      gsm.cell_id=1306 OR      gsm.cell_id=1367
OR      gsm.cell_id=1368 OR      gsm.cell_id=1369)
)
tb
ON      ta.cell_id =tb.cell_id
AND     ta.gps_id  =tb.gps_id
ORDER BY 1,2
')
AS ct(ID integer, BS_1 smallint, BS_2 smallint, BS_3 smallint, BS_4
smallint, BS_5 smallint, BS_6 smallint, BS_7 smallint,
BS_8 smallint, BS_9 smallint, BS_10 smallint, BS_11 smallint, BS_12
smallint, BS_13 smallint, BS_14 smallint, BS_15 smallint,
BS_16 smallint, BS_17 smallint, BS_18 smallint, BS_19 smallint, BS_20
smallint, BS_21 smallint, BS_22 smallint, BS_23 smallint, BS_24
smallint, BS_25 smallint, BS_26 smallint, BS_27 smallint,
BS_28 smallint, BS_29 smallint, BS_30 smallint, BS_31 smallint, BS_32
smallint, BS_33 smallint, BS_34 smallint, BS_35 smallint,
BS_36 smallint, BS_37 smallint, BS_38 smallint, BS_39 smallint, BS_40
smallint, BS_41 smallint, BS_42 smallint, BS_43 smallint, BS_44
smallint, BS_45 smallint, BS_46 smallint, BS_47 smallint,
BS_48 smallint, BS_49 smallint, BS_50 smallint, BS_51 smallint, BS_52
smallint, BS_53 smallint, BS_54 smallint, BS_55 smallint,
BS_56 smallint, BS_57 smallint, BS_58 smallint, BS_59 smallint, BS_60
smallint, BS_61 smallint, BS_62 smallint);

```

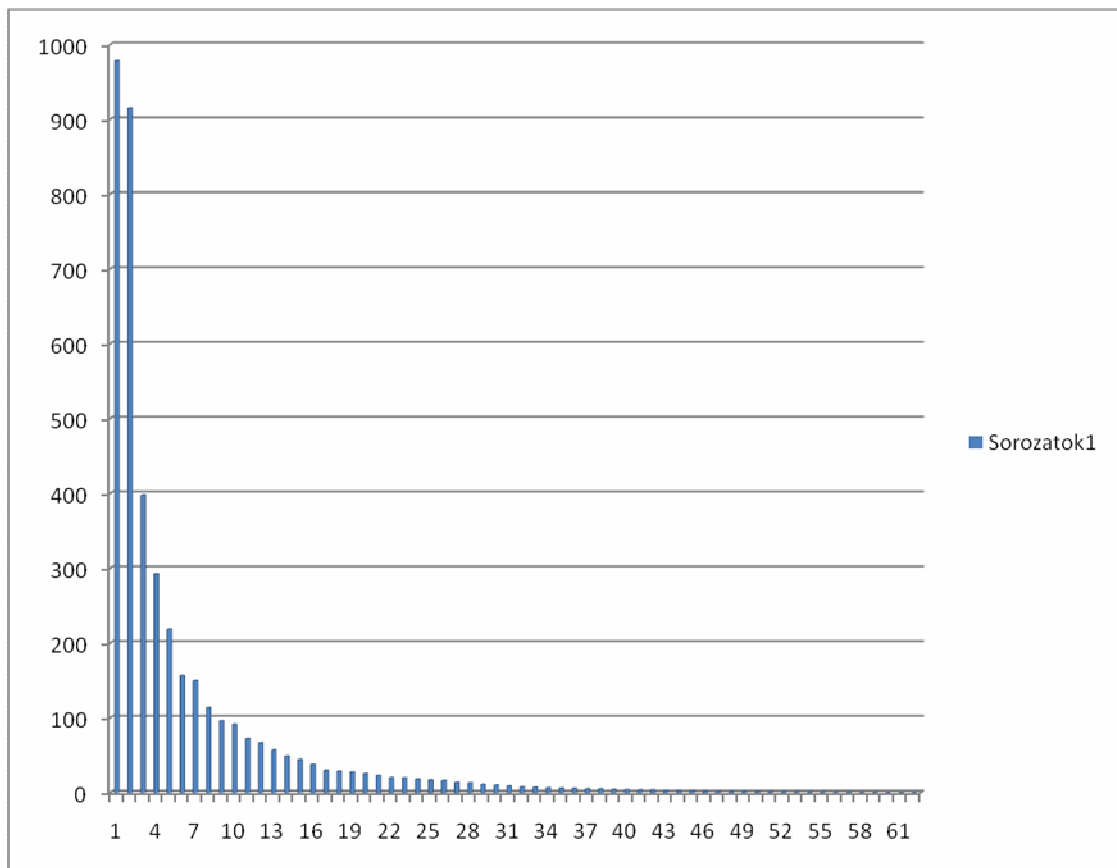
25. ábra – Jelszintek lekérdezése

A jelszinteket egy excel táblába exportáltam. A táblázatban az üresen maradt vételi jelszinteket kitöltöttem nullákkal, ez azt jelenti, hogy az adott bázisállomástól az adott helyen nem volt vételi jel.

Ezután a táblázatot betöltöttem Matlabba, hogy elvégezhessem rajta a főkomponens-analízist. A betöltés után a lefuttattam a következő parancsot:

```
[COEFF, SCORE, LATENT, TSQUARE] = princomp(PCA);
```

A princomp parancs hajtja végre az analízist, a megadott PCA táblán. Eredményül 4 táblát kapunk, amelyből 3 tábla fontos számunkra. A COEFF mátrixban kapjuk meg a transzformációs mátrixot, ami a transzformációt végezte. A SCORE mátrixban kapjuk meg az új rendszerbe transzformált koordinátákat. A LATENT vektor pedig az egyes komponensek súlyát mutatja.



26. ábra – A komponensek súlyai

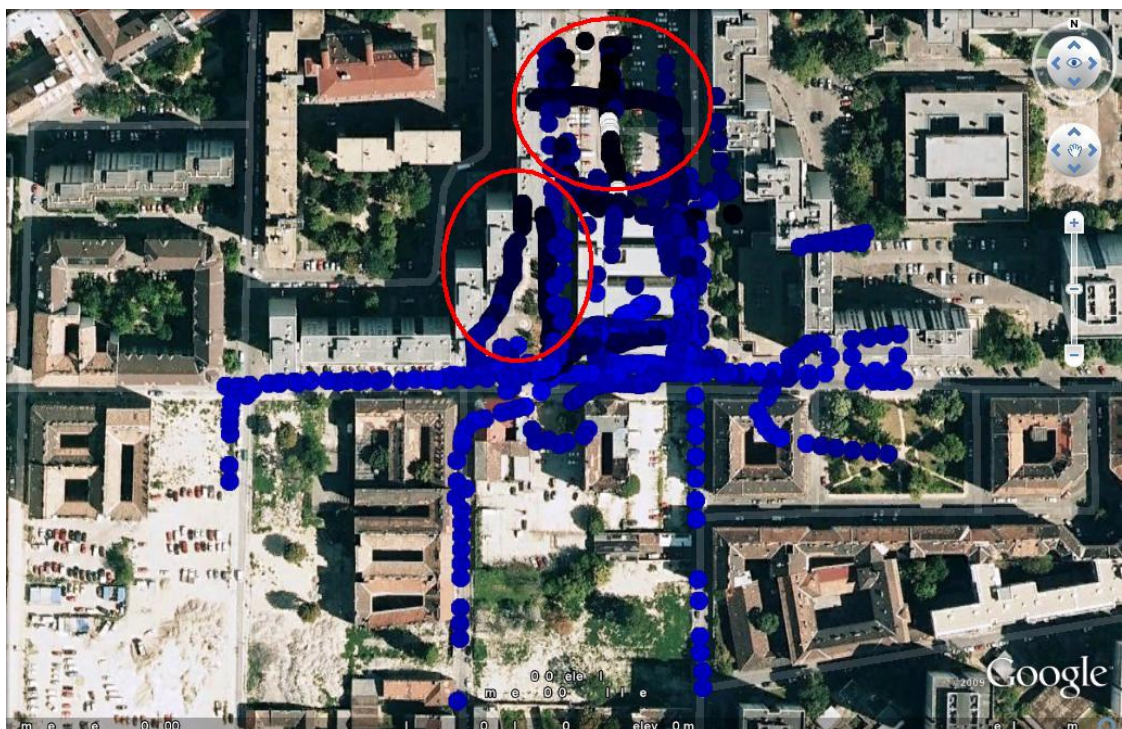
A LATENT mátrixot láthatjuk a x. ábrán. Ezen az ábrán rendre az látszik, hogy az egyes koordináta transzformációk eredményei mekkora súllyal látszanak. A magas súlyértékű koordinátákat felhasználva meghatározható pontossággal képesek vagyunk a tömörítést elvégezni. Az első két koordináta jelentősen nagyobb súllyal szerepel az eredmény létrehozásában, mint a többi, tehát a tömörítésnek ez jó alapot képez.

Ebből az ábrából kiderül, hogy a főkomponens-analízis eredménye hasznos lehet számunkra, hiszen a komponenseket nézve látható, hogy vannak lényegesebb és kevésbé lényegesebb komponensek, azaz különböző súllyal szerepelnek a komponensek.

Kiválasztottam az első három főkomponenst, ezeknek az új bázisban lévő értékeit (azaz az SCORE mátrix első 3 oszlopát) exportáltam egy excel táblázatba. Az excel táblázatban hozzáillesztettem minden ponthoz a hozzá tartozó megfelelő GPS koordinátát. Ezt követően komponensenként normalizáltam 0 és 255 közé a főkomponens értékeket, ekkor a x. ábrán látható táblázatot kaptam. Ezt egy Java programnyelven írt programmal feldolgoztam, főkomponensenként külön színezve egy-egy kml fájlba írtam. A kapott eredményt külön-külön megjelenítettem Google Earthen. Az első főkomponens változása látható a x. ábrán, a második főkomponens változása látható a x. ábrán, a harmadiké pedig a x. ábrán.

<Point><coordinates>19.07953833333332,47.488075000000002</coordinates>	34	76	192
<Point><coordinates>19.079560000000001,47.488061666666667</coordinates>	34	76	192
<Point><coordinates>19.079574999999998,47.488051666666664</coordinates>	34	76	192
<Point><coordinates>19.079585000000002,47.488061666666667</coordinates>	35	77	191
<Point><coordinates>19.079593333333332,47.488061666666667</coordinates>	35	77	191
<Point><coordinates>19.079599999999999,47.488063333333336</coordinates>	35	77	191
<Point><coordinates>19.079606666666667,47.488063333333336</coordinates>	35	77	191
<Point><coordinates>19.079611666666668,47.488064999999999</coordinates>	35	77	191
<Point><coordinates>19.079616666666666,47.48807</coordinates></Point>	35	77	191
<Point><coordinates>19.079623333333334,47.488076666666664</coordinates>	35	77	191

27. ábra – A főkomponens-analízis első 3 főkomponensének táblázata (részlet)



28. ábra - 1. főkomponens változása

A 7. ábrán a kiválasztott területen elvégzett főkomponens analízis első főkomponensének adott pontjaihoz rendelt súlyok látható. A megjelenítés során a főkomponens súlyának nagy értéke sötétkék, fekete, fehér. A kis súlyú tagok pedig világosabb kék foltokként vannak megjelenítve. Figyelmet érdemel pirossal bekeretezett rész, ahol az első főkomponens jelentős súllyal fordul elő. Ezek a területek jellemzően magas házfalak melletti területek, ahol a rádióhullámok terjedéséből következően a házfal nagy reflektáló felülete jelent kiemelést. Ebből úgy gondolhatjuk, hogy az első főkomponens ilyen nagy reflektáló felületekre jellemző adatokat tartalmaz. Ezt támasztja alá az is, hogy azokon a részekben ahol alacsonyabb téglalapítású házak, vagy üres telkek vannak ott kicsi vagy közepes értéket mutat az első főkomponens.



29. ábra - Második főkomponens változása

A 8. ábrán látható a második főkomponens változása, amely az elsővel megegyező módon, ugyan arról a területről készült. Hasonlóan látszik magas házaknál a nagy súly, de az ábra jobb oldalán ahol az első főkomponens aránylag kis értékeket mutatott az alacsony házak, illetve nyílt területnek köszönhetően a második főkomponens már jelentős értékeket mutat.

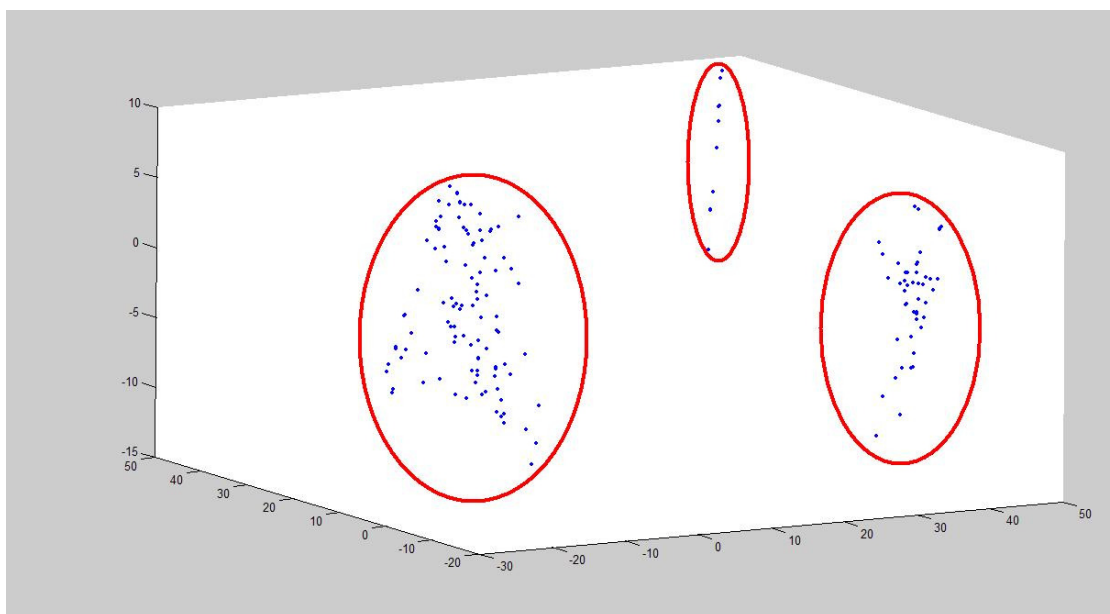


30. ábra - Harmadik főkomponens változása

Amennyiben megvizsgáljuk a harmadik főkomponenst (9. ábra) láthatjuk, hogy további jelentős változások vannak. A főkomponens transzformációnak ezen irányba már jelentősen eltér az előzőektől. A nagy házak nagy reflexiós felülete már csak közvetlenül sík terület mellett jelent nagy főkomponens súlyt és a sík területek, alacsony házak körüli rész ebben a főkomponens irányban már jelentős súlyokat kap. Ez megfigyelhető az ábrán külön jelölt területen.

Ezen eredményekből úgy tűnik, hogy a főkomponens-analízis segítségével megválasztható olyan új koordinátarendszer, ahol csökkenthető a dimenziószám és a főkomponensek alkalmasak lehetnek arra, hogy klasszifikálják, önálló részterületekre bonthassák a mérési eredményeink által létrehozott bonyolult vektorteret.

A klasszifikálással kapcsolatban további vizsgálatot végeztem. A vizsgálat során az előzőektől eltérő 780 mérési pontot dolgoztam fel, amelyekre az volt a jellemző, hogy legalább 6 bázisállomás vételében közös területet választottam ki. Az így létrejött főkomponens transzformáció egy 6x6-os transzformációs mátrixszal írható le. Elvégeztem a mérési pontok főkomponens térbe történő transzformálását. Az eredményt (azaz a főkomponensek értékeit) az első három főkomponense alapján 3 dimenziós ábrázolásban mutatom be a x. ábrán. Minden tengely egy-egy főkomponens értékeit reprezentálja.



31. ábra - Első három főkomponens

Az x. ábrán jól elkülöníthető 3 egymáshoz közeleső ponthalmaz, amely azt jelenti, hogy 3 részterületre könnyen felosztható a főkomponens transzformációt létrehozó eredeti mérési

ponthalmaz. Azonban ez a típusú matematikai transzformáció nem biztosítja azt, hogy az ilyen módon klasszifikált eredmény összefüggő ponthalmazt alkosson az eredeti mérési halmazon.

A főkomponens-analízistől elvárt dimenziószám csökkentés az előzőek alapján megvalósítható. A dimenziószám csökkentés nagy mennyiségű veszteséges adattömörítéssel valósul meg, ami a mérési adatok könnyebben tervezhető mértékű romlást eredményező tárolását tesz lehetővé. Ezeket az eredményeket a főkomponens analízis olyan módú használatával értük el, hogy egy kiválasztott területen teljes terület összes adatát figyelembe véve az összes bázisállomás összes mérési eredményére végeztük a főkomponens-analízist. Egyes méréseknél adott körülmények között nem fogható bázisállomások vételi jelszintjét 0 értékkel helyettesítettük. Ennél a módszernél további vizsgálatot kell végezni arra, hogy a főkomponens térből létrehozott ábrázolás adhat-e az egyes főkomponenseknek fizikai tartalmat.

A főkomponens-analízistől elvárt klasszifikáció tehát oly módon valósítható meg, hogy az eredeti mérési eredmények közül kiválasztottam olyan területeket, melyek esetében legalább 6 közös bázisállomás volt vehető. Ezek maximum néhány 1000 négyzetméteres területek. Ezen belül a főkomponens térben jól szeparálható részhalmazok keletkeztek, amik bizonyítják, hogy ez az eljárás klasszifikációra alkalmas. További vizsgálatot kell végezni annak érdekében, hogy a klasszifikációs eljárás eredményeül kapott halmazok hogyan helyezkednek el az eredeti mérési eredmények között.

A földrajzi adatbázis kezelők haszna ott mutatkozott meg munkám során ahol földrajzi fogalmak, földrajzi mérések kerültek elő. Ilyen volt például a 300 méter átmérőjű körben történő adatlekérdezés. Ezenkívül még az is nagy könnyebbség volt munkám során, hogy bármilyen koordináta adatot is kérdeztem le közvetlenül tudtam azt KML formátumban lekérni, így azt rögtön megjeleníteni Google Earth-en.

5. Térképészeti adatokkal bővített helymeghatározás

5.1. Bevezetés

Térképészeti adatokkal bővített GSM helymeghatározás során a meghatározott pozíciót szeretnénk még pontosítani azzal, hogy valamilyen térképi információt felhasználva a meghatározott helyet hozzákapcsoljuk valamilyen objektumhoz. Ilyen objektum lehet egy utca, egy buszmegálló, egy nevezetes épület. A következő fejezetben ilyen módszerek lesznek részletesen bemutatva.

Ezekhez a műveletekhez szükséges volt térképi adatokkal feltölteni az adatbázisomat. Térképi adatokat az OpenStreetMap ingyenesen elérhető térképészeti adatbázisából szereztem be. A PostgreSQL-hez rendelkezésre álló feltöltő script segítségével könnyedén be tudtam tölteni az adatbázisomba Magyarország összes rendelkezésre álló térképészeti adatát.

5.2. Adatkezelés

OSM adatokkal fel kellett töltenem az adatbázisomat, hogy ezzel kiegészítve segíteni tudjam a minél pontosabb, minél jobb helymeghatározást.

Első lépésként letöltöttem a magyarországi OSM adatokat erről a helyről: <http://download.geofabrik.de/osm/europe/hungary.osm.bz2>

Ez a csomag tartalmazta Magyarország határain belül eddig összegyűlt összes adatot.

A PostgreSQL adatbázisomba való adatfeltöltéshez leszedtem egy segédletet, ami az OSMtoPGSQL nevet viselte és innen töltöttem le: <http://tile.openstreetmap.org/osm2pgsql.zip>

Az adatok betöltése előtt le kellett futtatni egy SQL parancsot, ami létrehoz egy új vonatkoztatási rendszert, ez lesz a 900913-as számú SRID. Ehhez az osm2pgsql csomagban található 900913.sql fájlt futtattam le az adatbázisomon.

Ezek után elérkeztem az adatok tényleges betöltéséhez, amit a következő paranccsal tudtam végrehajtani:

```
C:\osm2pgsql>osm2pgsql -d GSM -m -p hungary_osm -U postgres -H localhost -P 5432
```

```
"C:\osm2pgsql\hungary.osm"
```

Ahhoz azonban, hogy ezeket az adatokat így használni tudjam módosítanom kellett a táblák SRID számát, azaz át kellett térnem más vonatkoztatási rendszerbe (koordináta

rendszerbe.) Az OpenStreetMap táblák feltöltése során ugyanis az adatok 900913 SRID-jű rendszerben lettek tárolva, az én GPS adataim azonban a 4326-as SRID-t használják, illetve a GoogleEarth is ezt a vonatkoztatási rendszert használja. Az áttérés más koordináta rendszerre történhetne a lekérdezések előtt is on-line módon, azonban ez rengeteg erőforrás emésztene fel feleslegesen ezért én úgy döntöttem, hogy off-line módon átirom és letárolom a megváltozott SRID-jű geometriai adatokat. Létrehoztam mind a 4 OpenStreetMap-os tábla klónját hasonló táblanéven csak egy 2-es írtam az új táblák nevének végére. A x. ábrán látható parancsot hajtottam végre az új táblák létrehozásához.

```
CREATE TABLE hungary_osm_line2 AS SELECT * FROM hungary_osm_line;  
CREATE TABLE hungary_osm_point2 AS SELECT * FROM hungary_osm_point;  
CREATE TABLE hungary_osm_polygon2 AS SELECT * FROM  
hungary_osm_polygon;  
CREATE TABLE hungary_osm_roads2 AS SELECT * FROM hungary_osm_roads;
```

32. ábra - Új tábla létrehozása

Ezzel létrehoztam 4 új táblát amely megkapta a régi táblák sémáját, és egyben fel is töltődött a régi táblák adataival.

Ezek után át kellett transzformálni a régi 900913 SRID-jű geometriai típust új 4326-as SRID-jű rendszerbe. Ezt a x. ábrán látható paranccsal csináltam meg minden táblán egyenként.

```
UPDATE hungary_osm_line2 set way = transform(SetSRID(way, 900913),  
4326);  
UPDATE hungary_osm_point2 set way = transform(SetSRID(way, 900913),  
4326);  
UPDATE hungary_osm_polygon2 set way = transform(SetSRID(way, 900913),  
4326);  
UPDATE hungary_osm_roads2 set way = transform(SetSRID(way, 900913),  
4326);
```

33. ábra - SRID átírása

Ezután egy gyors ellenőrzést hajtottam végre, hogy lássam, sikeres volt-e a transzformáció, ennek SQL parancs látható a x. ábrán.

```
SELECT SRID(way) FROM hungary_osm_line2;  
SELECT SRID(way) FROM hungary_osm_point2;  
SELECT SRID(way) FROM hungary_osm_polygon2;  
SELECT SRID(way) FROM hungary_osm_roads2;
```

34. ábra - SRID ellenőrzése

Mindenhol az új 4326-os SRID-t kaptam, ezzel a transzformáció sikeres volt.

Az adatbázis sémadiagramja látható az x. ábrán.

hungary_osm_roads2		hungary_osm_polygon2		hungary_osm_line2		hungary_osm_point2	
osm_id	integer	osm_id	integer	osm_id	integer	osm_id	integer
access	text	access	text	access	text	access	text
addr:flats	text	addr:flats	text	addr:flats	text	addr:flats	text
addr:housenumber	text	addr:housenumber	text	addr:housenumber	text	addr:housenumber	text
addr:interpolation	text	addr:interpolation	text	addr:interpolation	text	addr:interpolation	text
admin_level	text	admin_level	text	admin_level	text	admin_level	text
aerialway	text	aerialway	text	aerialway	text	aerialway	text
aeroway	text	aeroway	text	aeroway	text	aeroway	text
amenity	text	amenity	text	amenity	text	amenity	text
area	text	area	text	area	text	area	text
barrier	text	barrier	text	barrier	text	barrier	text
bicycle	text	bicycle	text	bicycle	text	bicycle	text
bridge	text	bridge	text	bridge	text	bridge	text
boundary	text	boundary	text	boundary	text	boundary	text
building	text	building	text	building	text	building	text
construction	text	construction	text	construction	text	capital	text
cutting	text	cutting	text	cutting	text	construction	text
disused	text	disused	text	disused	text	cutting	text
embankment	text	embankment	text	embankment	text	disused	text
foot	text	foot	text	foot	text	ele	text
highway	text	highway	text	highway	text	embankment	text
historic	text	historic	text	historic	text	foot	text
horse	text	horse	text	horse	text	highway	text
junction	text	junction	text	junction	text	historic	text
landuse	text	landuse	text	landuse	text	horse	text
layer	text	layer	text	layer	text	junction	text
learning	text	learning	text	learning	text	landuse	text
leisure	text	leisure	text	leisure	text	layer	text
lock	text	lock	text	lock	text	learning	text
man_made	text	man_made	text	man_made	text	leisure	text
military	text	military	text	military	text	lock	text
motorcar	text	motorcar	text	motorcar	text	man_made	text
name	text	name	text	name	text	military	text
natural	text	natural	text	natural	text	motorcar	text
oneway	text	oneway	text	oneway	text	name	text
power	text	power	text	power	text	natural	text
power_source	text	power_source	text	power_source	text	oneway	text
place	text	place	text	place	text	poi	text
railway	text	railway	text	railway	text	power	text
ref	text	ref	text	ref	text	power_source	text
religion	text	religion	text	religion	text	place	text
residence	text	residence	text	residence	text	railway	text
route	text	route	text	route	text	ref	text
service	text	service	text	service	text	religion	text
sport	text	sport	text	sport	text	residence	text
tourism	text	tourism	text	tourism	text	route	text
tracktype	text	tracktype	text	tracktype	text	service	text
tunnel	text	tunnel	text	tunnel	text	sport	text
waterway	text	waterway	text	waterway	text	tourism	text
width	text	width	text	width	text	tunnel	text
wood	text	wood	text	wood	text	waterway	text
z_order	integer	z_order	integer	z_order	integer	width	text
way_area	real	way_area	real	way_area	real	wood	text
lon_ref	text	lon_ref	text	lon_ref	text	z_order	integer
ron_ref	text	ron_ref	text	ron_ref	text	way	geometry
nwn_ref	text	nwn_ref	text	nwn_ref	text		
lwn_ref	text	lwn_ref	text	lwn_ref	text		
nwn_ref	text	nwn_ref	text	nwn_ref	text		
lwn_ref	text	lwn_ref	text	lwn_ref	text		
nwn	text	nwn	text	nwn	text		
route_pref_color	text	route_pref_color	text	route_pref_color	text		
route_name	text	route_name	text	route_name	text		
way	geometry	way	geometry	way	geometry		

35. ábra - OSM sémadiagram

A térképészeti adatokkal feltöltött táblák fontosabb oszlopainak jelentései:

- hungary_osm_line2 tábla, hungary_osm_point2 tábla, hungary_osm_polygon2 tábla, hungary_osm_roads2 tábla (közel azonosak az oszlopok):

Oszlop	Tulajdonság	Példa
way	ez a geometria, ami tárolja az adott dolog koordinátáit, ami lehet pont, poligon, törtvonal	épület (pont), utca (törtvonal), vasútvonal, folyó, kerítés, épület körvonala (zárt vonal), megyehatár, tó (terület), stb.
highway	az út típusa	autópálya, főút, mellékút, stb.
barrier	akadályok	kerítés, fal, sorompó, kapu, stb.
waterway	vizes területek	folyó, kikötő, gát, stb.
railway	vasúttal kapcsolatos dolgok	vasútvonal, villamos vonal, metróvonal, megálló, stb.
tunnel	alagút	
bridge	híd	
man_made	emberek által épített dolgok	világítótorony, szélmalom, stb.
leisure	szabadidős tevékenységhez kapcsolódó dolgok	parkok, játszóterek, stadionok, golfpályák, stb.
amenity	épületek	étterem, kávézó, iskola, könyvtár, buszpályaudvar, stb.

shop: üzletek, pl. pékség, könyvesbolt, autószalon, stb.

tourism: turistáknak szóló helyek, pl. kemping, vendégház, hotel, múzeum, vidámpark, stb.

landuse: mire használják az adott területet, pl. erdő, ipari terület, farm, katonai terület, stb.

natural: természeti képződmények, pl. barlang, különleges fa, vulkán, stb.

route: útvonal, pl. bicikli út, komp, vasút, busz, stb.

boundary: határoló vonal, pl. állam, megye, nemzeti park

sport: sportlétesítmények, pl. kosárlabda pálya, golfpálya, lóversenypálya, stb.

name: az egyed neve, ha létezik, pl. Vas megye, Fő utca, Nyugat-Dunántúl, stb.

Az OpenStreetMap adatok feltöltése után néhány próba lekérdezéssel ellenőriztem, hogy jó volt-e az adatfeltöltés, helyesek-e az adatok, illeszkednek-e a Google Earth térképére. A x. ábrán látható vékony fehér vonalak az OSM adatbázis által mutatott utak, a vastagabb átlátszó vonalak pedig a Google Earth által mutatott utak.



36. ábra - OSM adatok

Az adatok megfelelőek, kissé eltérnek egymástól, de ez mérési hibából adódóan még elfogadható. Ellenőriztem azt is, hogy azokon a területeken, ahol rendelkezésre állnak a GSM mérési adatok, ott mennyire van lefedve a terület OSM adatokkal, erre egy példa látható a x. ábrán.



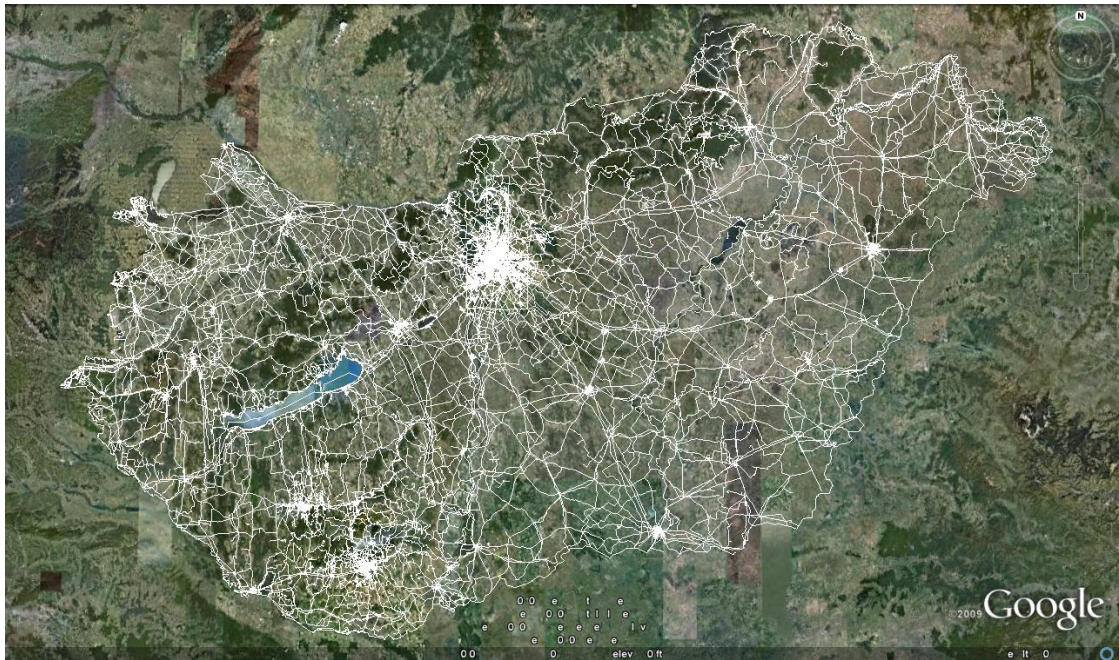
37. ábra – GSM és OSM adatok

Arra a következtetésre jutottam, hogy a betöltött adatok megfelelőek, kellőképpen lefedik a területeket, ahol helymeghatározást fogok végezni.

A 4 tábla amely feltöltésre került:

- hungary_osm_line2: mindenféle vonallal reprezentálható dolog, pl. út, vasút, mellékút, határvonalak, tavak, stb.
- hungary_osm_point2: pontok, mint pl. települések, benzinkutak, stb.
- hungary_osm_polygon2: területek, mint megyék, tavak, erdők, stb.
- hungary_osm_roads2: számmal jelzett főutak, autópályák, vasutak, közlekedési gerinchálózat

A x. ábrán szemléltetésképpen látható Magyarország térképi adatai, amelyet az OpenStreetMap adatbázisom hungary_osm_line2 táblájából kérdeztem le, a későbbiekben ezt a táblát fogom főként használni.



38. ábra - Magyarországi OSM adatok

5.3. Bővített helymeghatározás

Ide még írnék egy kis felvezetőt a fejezettel kapcsolatban

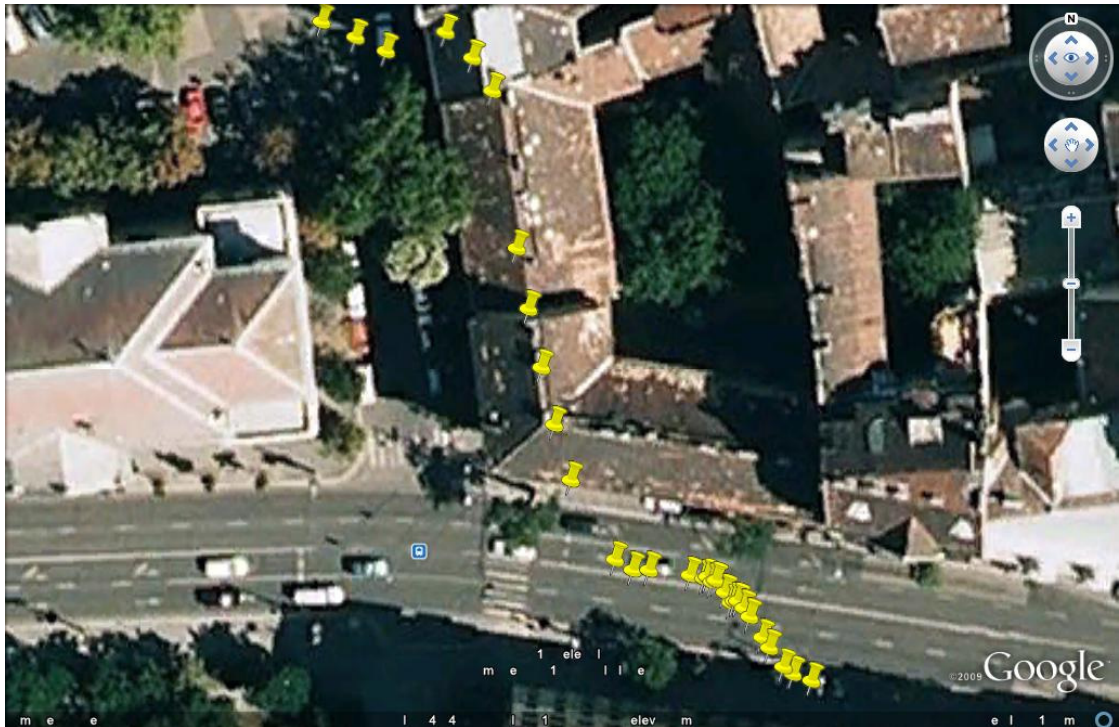
5.3.1. Helymeghatározás utcatérképpel

Helymeghatározás során gyakran előfordulhat a pontatlanság miatt, hogy a meghatározott pont nem az utcára kerül, hanem például az utca melletti telekre, pedig mi az utcán sétálunk. Ezt a problémát a személyi navigációs GPS rendszerek úgy oldják meg, hogy a helymeghatározás után mindig rávetítik a kapott pontot az utcára. Ezt a megoldást fogom én is alkalmazni.

A térképi adatokat, azaz az utcák adatbázisát az OpenStreetMap adatbázisom hungary_osm_line2 táblája fogja szolgáltatni. Itt megtalálhatóak a közösség által eddig feltérképezett utcák Magyarországról, amely jó alapot szolgáltat.

A lekérdezésem, amely az utcára vetített helymeghatározást végzi több részből tevődik össze. Felhasználom az 4.3.2. pontban bemutatott helymeghatározási megoldást, és ez alapján fogom az utcára vetítést végezni.

Első lépésként kerestem egy olyan területet, ahol az 4.3.2. pont szerinti helymeghatározás eredménye nem pont egy utcára esik. A hét bázisállomás egyszerre látszik az x. ábrán látható pontokon.



39. ábra - 7 bázisállomás egyszerre látszik

Ennek középpontja található a x. ábrán, amely láthatóan nem illeszkedik egyik utcára sem.



40. ábra - Középpont

Végrehajtva a x. ábrán látható SQL lekérdezést, megkapjuk, a pontot a legközelebbi utca legközelebbi pontjára elhelyezve. A lekérdezés alapja a 7 bázisállomásos helymeghatározási módszer, amely az első FROM utáni zárójeles SELECT utasítás, mely a 7 bázisállomás alapján

visszaadja a terület középpontját, amelyet foo.r1-nek neveztem. Ezután a pont köré képezek egy puffer mezőt 100 méteres körzetben (ST_Buffer függvény, ahol a 0.001 jelenti a 100 métert), majd erre a pufferre csinálók egy metszetet (ST_Intersects) az OSM adatokkal (r.way). Ezek a műveletek a WHERE feltételben találhatók. Végül pedig az így szűkített adathalmazon elvégzem a legközelebbi pontkeresést, azaz az ST_ClosestPoint függvényt futtatom, majd az eredményt az ST_AsKML függvénnyel kml formátumba konvertálom. Az ST_Transform függvényre azért volt szükség, mert az általam használt 4326-os SRID vonatkoztatási rendszer nem számolja elég pontosan a távolságokat, ezért az ST_ClosestPoint kiszámolásának idejére áttérek a 23700-as SRID-re amely egy távolságtartó vetítési rendszer, így pontos lesz az utcára vetítem.

```
SELECT ST_AsKML(ST_ClosestPoint(ST_Transform(r.way,23700),
ST_Transform(foo.r1,23700))) AS xy
FROM hungary_osm_line2 r,
(SELECT (ST_Centroid(ST_ConvexHull(ST_Collect(gps_geom2)))) AS r1
FROM
(Select gps.id, gps_geom2 FROM gps, gsm WHERE gsm.cell_id=861 AND
gps.id=gsm.gps_id) t1,
(Select gps.id FROM gps, gsm WHERE gsm.cell_id=980 AND
gps.id=gsm.gps_id) t2,
(Select gps.id FROM gps, gsm WHERE gsm.cell_id=899 AND
gps.id=gsm.gps_id) t3,
(Select gps.id FROM gps, gsm WHERE gsm.cell_id=917 AND
gps.id=gsm.gps_id) t4,
(Select gps.id FROM gps, gsm WHERE gsm.cell_id=856 AND
gps.id=gsm.gps_id) t5,
(Select gps.id FROM gps, gsm WHERE gsm.cell_id=977 AND
gps.id=gsm.gps_id) t6,
(Select gps.id FROM gps, gsm WHERE gsm.cell_id=855 AND
gps.id=gsm.gps_id) t7
WHERE t1.id=t2.id AND t2.id=t3.id AND t3.id=t4.id
AND t4.id=t5.id AND t5.id=t6.id AND t6.id=t7.id
)
foo
WHERE ST_Intersects(r.way,St_Buffer(foo.r1, 0.001,6))=TRUE
ORDER BY ST_Distance(foo.r1,r.way) limit 1;
```

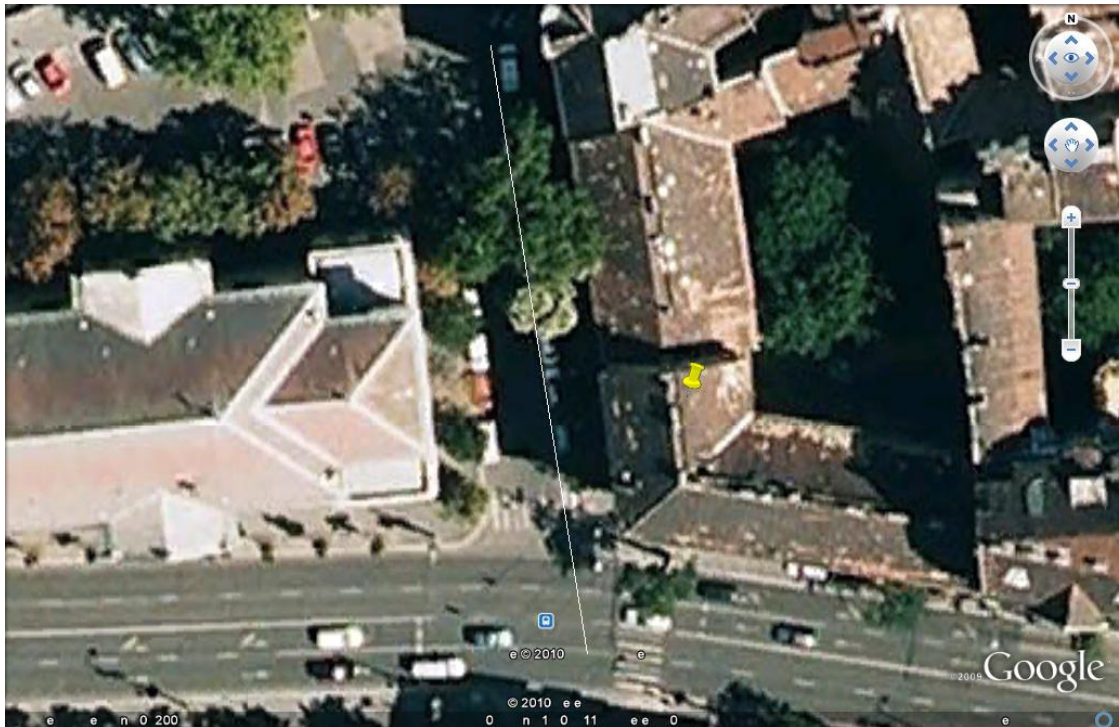
41. ábra – SQL lekérdezés: helymeghatározás vetítése a legközelebbi utcára

A lekérdezés eredménye látható a x. ábrán, ahol a sárga jel középen az eredetileg meghatározott pont, a fehér vonal az utcákat jelöli, a fehér vonal melletti sárga jel pedig a javított helymeghatározási pont.



42. ábra - Helymeghatározás utcára vetítve

Egy másik alternatíva lehet, amikor nem szeretnénk tudni, hogy pont az utca melyik részén állunk, hanem azt szeretnénk csak tudni melyik utcában tartózkodunk. Ilyenkor az x. ábrán látható lekérdezéshez nagyon hasonló lekérdezés segítségével megkaphatjuk az utcát vagy utcarészletet amely az x. ábrán látható, illetve az utca nevét, amely a lekérdezés eredményében látható a x. ábrán.



43. ábra - Helymeghatározás az utca meghatározásával

<pre><LineString><coordinates>19.076021951479436,47.490118552428569 19.076150859722709,47.489570664850127</coordinates></LineString></pre>	Horváth Mihály tér
--	-----------------------

44. ábra - Az utca koordinátái és neve

Az eredményekből tehát látszik, hogy az OSM adatokkal kiegészített adatbázis alkalmas a helymeghatározás utcával történő pontosításához.

5.3.2. Helymeghatározás tömegközlekedési hálózattal

Könnyen előfordulhat velünk, hogy tömegközlekedésen utazunk, tudjuk hányas számú járműre szálltunk fel, azonban a nagy tömegben nem látjuk a jármű kijelzőjén, vagy épp a nagy zajban nem halljuk a hangszóróból, hogy melyik állomás következik. Erre lehet egy lehetőség, ha a helymeghatározást a tömegközlekedési járművek, például buszok, villamosok, metrók, HÉV-ek következő megállóinak meghatározására használjuk.

Lássuk egy példát a megvalósításra. Tegyük fel, hogy utazunk a 160-as buszon, amely a 10-es számú főúton a városból kifelé tart. A x. ábrán alul látható sárga jel a mobilkészülékünk által meghatározott helyzetünk, a felső sárga jel pedig a következő megálló helye. A mobilkészülék tudásától függően megjeleníthetjük a következő megállót illetve az aktuális helyzetünket egy térképen, vagy egyszerűbb készülék esetében, egyszerűen csak megadhatjuk szövegesen, hogy melyik megálló következik, ugyanis az adatbázisunkból ezt is

lekérdezhető. A lekérdezés nagyon hasonló az 5.2.1. pontban használttal. Az lekérdezés eredménye látható az x. ábrán, azaz jelenleg az Orbán Balázs út nevű megálló következik.



45. ábra - Buszmegállós helymeghatározás

<Point><coordinates>19.026960700000014,47.55580369999997</coordinates></Point>	Orbán Balázs út
--	-----------------

46. ábra - A következő megálló pozíciója és neve

A módszer persze további finomítást igényel. Ha például a rendszer lát egy következő megállót, azonban a mi járatunk ott nem áll meg akkor hamisan állítanánk, hogy melyik megálló következik. Célszerű lenne, ha felszállásnál megadnánk a jármű számát és irányát. Ehhez további információkat kellene feltölteni az adatbázisba. Ezzel azonban biztosítható lenne, hogy meg tudjuk határozni a jelenlegi és a következő állomást akkor is, ha épp egy megállóban állunk, vagy a mi járatunk kihagyja a következő megállót, esetleg ha egy szakasz közös más járatokkal, amelyek letérnek a mi útvonalunkról.

6. Összefoglalás és kitekintés

Munkám során tanulmányoztam a GSM helymeghatározás eddigi módszereit, eredményeit. Megismerkedtem a helymeghatározás alapvetéseivel, a jelenleg is működő GPS helymeghatározás módszerével. Ezek után tanulmányoztam a földrajzi kiterjesztéssel rendelkező objektum relációs adatbázis-kezelő rendszereket.

Felépítettem egy földrajzi adatbázist PostgreSQL adatbázis-kezelő rendszer és ennek kiterjesztése a PostGIS segítségével. Ezt az adatbázist feltöltöttem korábban elvégzett mérési adatokkal. Ezeket az adatokat használtam a helymeghatározás lehetőségeinek vizsgálataihoz. 3 féle helymeghatározási módszert dolgoztam ki, amelyeket a felépített adatbázisom segítségével teszteltem.

A földrajzi adatbázisok és azok függvénykészletei jól használhatónak bizonyultak a helymeghatározás során. A felépített adatbázisból számtalan lekérdezést hajtottam végre ahol ilyen függvényeket használtam. Ezek nagyban megkönnyítették a munkámat, mert ezeket a függvényeket hagyományos objektum relációs adatbázis kezelő rendszerekben mással helyettesíteni igen nehéz és időigényes lett volna.

A helymeghatározást térképészeti adatok felhasználásával javítani próbáltam. Betöltöttem az adatbázisomba az OpenStreetMap közösségtől ingyenesen elérhető térképészeti adatokat. Ezen adatok felhasználásával a helymeghatározási módszereimet például a pontatlanul meghatározott helyeket utcára vetítve javítottam.

Kitekintés ide : ...

- több adat áll rendelkezésre – pl. 1000 helyen helymeghatározást végigpróbálni összehasonlítani az eredeti pontokkal
- ez az adatbázis csak a vizsgálatot támogatja – optimalizálás – sémát a lekérdezésekre optimalizálni

Tihanyi Tanár Úr tanácsai:

További munka során

FŐK1:

- A fők analízisnél olyan területen elvégezni a vizsgálatot ahol csak nagy ház van, 300-400 méter terület bekjárásával hasonló eredmények jönnek –e ki
- családi házaknál elvégezni

- üres területeken

-beltéren mennyire jó? /sok reflektáló felület/ távolba mutató ötlet.

mennyire területre és terjedési viszonyra jellemző?

FŐK2:

- elvégezni úgy hogy statisztikai módszerrel kiválogatott valami szerint egyforma területre (tpbbre) igaz-e hogy mindenhol pontosan klasszifikálódik?

Létrehoztam földr adatb, létrehoztam kapcsolatot matematikával, megmutattam hogy visszaraktam a föld koord mellé

Ide kéne 1-2 oldal összefoglalás majd a végén.

7. Irodalomjegyzék

- [1] Globális műholdas helymeghatározás alapjai
<http://www.georgikon.hu/mobilkom/GPS/alapok.htm>
- [2] Takács György, „Helymeghatározás mobiltelefonnal és mobil hálózattal,” *Híradástechnika*, LXIII. évfolyam, 2008/8
- [3] Bányai Balázs, Feldhoffer Gergely, Tihanyi Attila, „Helymeghatározás GSM hálózat felhasználásával,” Pázmány Péter Katolikus Egyetem, 2009
- [4] Michel Mouly, Marie-Bernadette Pautet, *The GSM System for Mobile Communications*, Telecom Publishing, 1992, ISBN:0945592159
- [5] Térinformatika meghatározások
<http://terinformatika.geocentrum.hu/>
- [6] Földrajzi információs rendszer
http://www.webgobe.eu/gis/GIS/foldrajzi_informacios_rendszer.html
- [7] Oracle Corporation, AZ ORACLE® SPATIAL OPCÍÓ ÉS AZ ORACLE® LOCATOR térinformatikai szolgáltatásai az Oracle Database 10g-ben
http://www.oracle.com/global/hu/database/collaterals/DATABASE10G_SPATIALLOCATOR_HUN.pdf
- [8] Schönhofen Péter, *Nyílt forráskódú adatbázis-kezelők*, SZAK Kiadó Kft., 2007, ISBN 978-963-9131-96-5
- [9] John C. Worsley, Joshua D. Drake, *Practical PostgreSQL (O'Reilly Unix)*, Command Prompt Inc, 2001, ISBN 1-56592-846-6
- [10] OpenStreetMap névjegy
<http://wiki.openstreetmap.org/wiki/HU>About>
- [11] PostGIS 1.5.1 Manual
<http://postgis.org/documentation/manual-1.5/>
- [12] Tikk Domokos, *Szövegbányászat*, Typotex Kft, 2007, ISBN 9789639664456