

Önálló laboratórium beszámoló

Dolgozat címe:

.....

Konzulens(ek) neve:

(Külső cég neve:

címe:

A Hallgató a kitűzött feladatot megfelelő színvonalon és a kiírásnak megfelelően

teljesítette

nem teljesítette

Konzulens aláírása

Hallgató neve:

Képzés:

Leadás dátuma:



Pázmány Péter Katolikus Egyetem Információs Technológiai Kar

Turisztikai alkalmazás készítése GSM alapú helymeghatározás

Készítette: Gangl Zsolt

Konzulens: Tihanyi Attila

PPKE-ITK
2012.

Tartalom

1. Bevezetés	4
2 Technikai háttér	5
2.1 GSM alapú helymeghatározás	5
2.2 Földrajzi adatbázisok	7
2.3 Google Earth és a KML kiterjesztés	8
3. GSM és GPS alapú mérések és eredmények ábrázolása.....	11
3.1 Bevezetés	11
3.2 A felépített rendszer.....	11
3.3 Adatbázis séma	11
3.4 Ábrázolt adatok.....	14
4. Konklúzió.....	24
5. Irodalomjegyzék	25

1. Bevezetés

A 2011/2012-es tanév tavaszi félévében vettem fel az Önálló laboratórium elnevezésű tárgyat, amelyen a választott témám címe a következő: Turisztikai alkalmazás készítése GSM alapú helymeghatározás.

A projekt témavezetője Tihanyi Attila és a keze alatt egy csoportban hatan dolgoztunk. Hasonló projekten az egyetemen belül már többen foglalkoztak, ezért nem kellett az alapoktól elkezdni a munkát, eredményeik javarészt a rendelkezésünkre álltak. Mivel a körünkben nem volt olyan, aki már korábban foglalkozott ezzel a témával részletesebben azért a munkát az eddigi eredmények tanulmányozásával kezdtük. A későbbiekben elosztottuk egymás között a feladatokat, amelyeket mi határoztunk meg, melyek a projekt szempontjából fontos területek, ahol eredményeket tudunk elérni. Ilyen témákat találtunk: jelentős előrelépés lehet, ha csökkentjük az adatbázis méretét, rendszerintegritásokat keresünk, további méréseket végzünk, illetve a mért és későbbiekben eredményül kapott adatokat ábrázoljuk.

Én az eredmények ábrázolását választottam témának. Feladataim közé tartozott, hogy a mérési eredményeket, kapott adatokat, sejtéseket ábrázoljam, és rendszerezzem.

2 Technikai háttér

Az alábbi fejezetben a GSM alapú helymeghatározásról, a földrajzi adatbázisokról és a Google által fejlesztett Google Earth programról és a hozzá szorosan kapcsolódó kml fájlformátumról lesz szó. Leginkább a GSM alapú helymeghatározás témakörét fejtettem ki, fundamentális jelentősége miatt.

2.1 GSM alapú helymeghatározás

[1] A GSM alapú helymeghatározás egyre fontosabb lehet a világunkban, elvégre a mobil eszközök szinte mindenkinek a zsebében ott lapulnak. A mobil helymeghatározó rendszerek lehetnek végberendezés alapúak, hálózat alapúak és alapulhatnak a végberendezés és hálózat együttműködésén is. A rádiós rendszereken alapuló helymeghatározásban alapkérdés, hogy fix elemek rádiójelét veszi a vándorlóobjektum és a fix résztől vett jeleket visszaküldi, vagy vándorló objektum által kisugárzott jeleit veszi a fix rész több alkalmas eleme. Előfordulhat még ezek kombinációja, de még az is, hogy ezekből kiindulva a helymeghatározás egészen más utakra tér.

Leszármaztatott autonóm helymeghatározás

Tegyük fel, hogy a vándorló objektum (mobiltelefon) egy ismert helyről indul, amelyet például egy GPS módszerrel határozott meg. Az ismert kiinduló ponttól kezdve folyamatosan méri a mobiltelefon a pillanatnyi gyorsulásának nagyságát és irányát. A mozgás pillanatnyi gyorsulásából az eltelt idő alapján kiintegrálható az aktuális helyzet. Hasonlóan tovább az újabb és újabb helyzetek is meghatározhatók az idő múlásával. A módszer óriási előnye, hogy a mobiltelefon akkor is tovább mérni és számolni képes helyzetét, ha elvesztette a kapcsolata a GPS vagy mobil hálózattal.

Helymeghatározás bázisállomás-azonosítók alapján

A bázisállomások azonosítóját (Cell-ID) az adó kisugározza, a mobiltelefon veszi és felhasználja működése során. A cellaazonosítóhoz a hálózat üzemeltető hozzá tud rendelni egy olyan földrajzi koordinátát, amely az ellátott terület közepét jellemzi, ami persze nem szükségszerűen azonos a bázisállomáshelyével. Ennél módszernél a helyváltoztatáson vett cellaazonosítók követésével tovább finomítható a helymeghatározás. Az adatsor alapján a mozgásirányt, mozgási sebességet is bele tudja kalkulálni a helymeghatározásba a rendszer és ezzel a cellaméretnél pontosabb becslés adható a pillanatnyi helyzetre.

Helymeghatározás ívmetszéssel

Ha meg tudjuk határozni két különböző bázisállomástól a tényleges távolságunkat, akkor helyzetünk két körív mentén adódik, amelyek rendszerint két pontban metszik egymást. Egy harmadik bázisállomástól távolságunkat meghatározva közülük már kiválasztható a valódi helyzetünket leíró pont. A távolságmérés alapulhat a jelerősség mérésén, mivel a vevő által vett rádiófrekvenciás jel szintje függ az adó és a vevő távolságától, az adó teljesítményétől, az adóantenna nyereségétől, iránykarakterisztikájától, a vevőantenna nyereségétől és a terjedést befolyásoló tényezőktől. Városi környezetben az Okumura-Hata modellt használják a terjedési veszteség számítására.

Helymeghatározás a háromszögelés elvével

Ez a rádióállomások (például kalózkodók) bemérésének hagyományos módszere. Különböző helyeken (esetünkben különböző bázisállomásokon) megméri, hogy a keresett adó (esetünkben a mobilkészülék) jele milyen irányból érkezik. Elvileg két méréssel a helyzet meghatározható, a háromszög egy oldalának és két szögének ismerete alapján. Több méréssel a meghatározás hibája csökkenthető. Több, egymás melletti vevőantennát antennavektorként használva, az iránymérés az időkülönbségek alapján kifinomultan megoldható.

Helymeghatározás elve az észlelt időkülönbség mérése alapján

Tegyük fel, hogy a bázisállomások órái szinkronizáltak működnek. A mobilkészülékhez egy egyszerre elküldött jelsorozat a távolságtól függő időkésséssel érkezik. A mobiltelefon saját pontos, de nem szinkronizált órája alapján ugyanazt a jelsorozatot generálva korrelációs módszerrel azt tudja kellő pontossággal mérni, hogy különböző bázisállomásokról egyszerre küldött kódsorozatot milyen időeltéréssel vesz. A bázisállomások helye a hálózatüzemeltető számára ismert. Azok a pontok, amelyeknek távolságkülönbsége egy bázisállomás-pártól állandó, egy hiperbola mentén helyezkednek el. A mobilkészülék helye tehát a hiperbolák metszéspontjában van. Az időkülönbségek meghatározhatók fordított irányú terjedés alapján is. A mobiltelefonból kisugárzott rádiójelet a különböző bázisállomások különböző időpontban veszik. A rendszer meg tudja határozni az időkülönbségeket és ebből a mobiltelefon helyzetét.

Mobil rendszerekkel segített GPS

A mobilhálózattal segített GPS arra szolgál, hogy az eredeti rendszer több hiányosságát segítse kiküszöbölni. A hagyományos GPS rendszer egyik problémája ott jelentkezik, ahol a vételi viszonyok kedvezőtlenek (például nagyvárosok magas épületekkel beépített szűk utcáiban vagy fák alatt, ahol

gyengén és sok visszaverődéssel zavartan érkezik a jel). További probléma, hogy bekapcsolás után percekig eltarthat, míg a vevő összekapcsolódik a rendszerrel, különösen kedvezőtlen vételi viszonyok esetén. Az A-GPS rendszerben a mobilhálózat Assistance Server (AS) egysége többféle módon segíti a helymeghatározást.

- A szerver jó közelítéssel tudja a telefon helyzetét a Cell-ID alapján.
- Az AS-nek vannak jó vételi helyen elhelyezett vevői, az ott vett adatokból pontosan tudja a GPS holdak aktuális pozícióját, ezt le tudja küldeni a mobiltelefonnak, amely ezek ismeretében könnyebben, gyorsan felkapcsolódhat.
- A pontosan bemért helyzetű mérővevői által vett jelek alapján hibakorrekcióhoz is tud adatokat szolgáltatni.
- A szerverben hatalmas számítási erőforrások állnak rendelkezésre, így pontosabban és gyorsabban elvégzi a helymeghatározáshoz és hibakorrekcióhoz szükséges számításokat, mint a mobiltelefon, s ezzel a telepét is kíméli. A kiszámolt pontos adatokat pedig leküldi a mobiltelefonnak.

2.2 Földrajzi adatbázisok

A földrajzi adatbázisok jelentősége a térinformatika által meghatározott eredmények tárolásának megfelelő módjában rejlik. Magyarul a térinformatika szó a "geographic information system" angol kifejezést helyettesíti (rövidítése angolul: GIS). A térinformatika együttesen kezeli a grafikai (térkép, űrfelvétel, légi fotók stb.) és leíró (tematikus adatokat tartalmazó) adatbázisokat, és különböző típusú elemzéseket tud végezni. Ezeknek az elemzéseknek az eredményeit grafikailag meg tudja jeleníteni, például térkép vagy digitális dombormodell formájában. Magát a térinformatikát több szempontból is megközelíthetjük [2]:

- 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)
- 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)
- A GIS egyidejűleg teleszkópja, mikroszkópja, számítógépe és xerox-gépe a térbeli adatok elemzésének és szintézisének. (Abler, 1988)

Ilyen térinformatikai rendszerekkel lehet létrehozni földrajzi adatbázisokat. Jelentősége a különböző hivatkozási rendszerek közötti átmenetek megvalósítása, így lehetőség van adatbázisok felépítésére. Ilyen relációs adatbázis kezelő rendszer a PostgreSQL, amelyet a projekthez szorosan kapcsolódik, legtöbbünk adatbázisa, és a közös adatbázis is ezen alapszik.

[3] A PostgreSQL, más néven Postgres egy relációs adatbázis-kezelő rendszer (angol rövidítéséből: (O)RDBMS). Licencét tekintve szabad szoftver, a fejlesztést önkéntesek végzik közösségi alapon. A PostgreSQL fejlesztése a Kaliforniai Berkeley Egyetem gondozásában indult Ingres néven a nyolcvanas években az azonos nevű cég szponzorálásával. 1996-tól a szoftver fejlesztése elhagyta az egyetem falait és megindult a közösségi alapú fejlesztés útján. A 6.0 változatot már PostgreSQL névvel adták ki.

[4] A PostGIS egy kiegészítő csomag a PostgreSQL-hez. 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. Ami miatt előnyös volt a projekt során a PostGIS alkalmazása:

- 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.

2.3 Google Earth és a KML kiterjesztés

[5] A Google Earth, vagy Google Föld egy Google által fejlesztett program, ami virtuális földgömbként használható. A Föld háromdimenziós modelljére mértékhelyes műholdképek, légi felvételek és térinformatikai adatok vannak vetítve. A programot eredetileg a Keyhole cég fejlesztette ki, majd 2004-ben megvásárolta a Google. A Google Föld Fizetős verziói 2009-ben megszűntek. [6] Jelenleg a következő rendszereken érhető el: Android 3.0+, Windows 2000, XP, Vista, 7, Mac OS X, Blackberry Storm, iOS és Linux.

[7] A Google Earth-tel lehetőség van saját adatok ábrázolására, saját fájlformátumon keresztül, ez a kml. A kml (Keyhole Markup Language) formátum 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.

Ez a program a projektben azért volt mindenképpen hasznos, mert viszonylag könnyen és egyszerűen lehetett olyan állományokat készíteni, amellyel ábrázolni lehetett a mért és eredményként kapott pontokat, akár színezve is.

Példa egy kml fájlra, ami az egyetem bejáratát jelölő pont:

```
<?xml version="1.0" encoding="UTF-8"?>
<kml xmlns="http://earth.google.com/kml/2.0">
  <Placemark>
    <description>PPKE-ITK</description>
    <name>PPKE-ITK</name>
    <Point>
      <coordinates>19.079556, 47.486872</coordinates>
    </Point>
  </Placemark>
</kml>
```

Az első sorban az xml fejléc található, a másodikban a kml névtér deklaráció, ehhez kapcsolódik a </kml> záró elem. A <Placemark> és </Placemark> között foglalnak helyet a különböző kml objektumok. Ha egyszerre több objektumot szeretnénk egy kml fájlba rendezni, akkor ezen kívül használni kell a <Folder></Folder> tag-et is. A <Placemark>-on belül található <description> és <name> tag-eken belül egyéb adatokat találhatunk, ezek a megjelenítés során látszódnak is. Ezek után találhatók az objektumok, jelen esetben <Point>, ami pontszerű objektumot takar, de lehet akár vonal, vagy poligon. Magát a helyet a <coordinates> tag-en belüli értékek adják, hosszúsági (Longitude), majd a szélességi (Latitude) fokok.

Példa egy más jellegű kml fájlra, ez az egyetem épületét elfedő poligon:

```
<?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>egyetem.kml</name>
    <Placemark>
      <description></description>
      <name></name>
      <Polygon>
        <outerBoundaryIs>
          <LinearRing>
            <coordinates>
              19.079936,47.487417
              19.079917,47.486867
              19.079375,47.486867
              19.079383,47.487425
            </coordinates>
          </LinearRing>
        </outerBoundaryIs>
      </Polygon>
    </Placemark>
  </Document>
```

```
</LinearRing>
</outerBoundaryIs>
</Polygon>
</Placemark>
</Document>
</kml>
```

Ez a példa már az elején különbözik az előzőtől, mert tartalmazza a gx: Google által használt kiegészítő névteret is. A <Polygon> tag határozza meg, hogy milyen jellegű az ábra, az <outerBoundaryIs> tag jelenti, hogy egy poligon külső határát adjuk meg <LinearRing> típusú objektummal. LinearRing esetén fontos, hogy a kezdőpont egyben a végpont is, ezért fontos a pontok egymás utáni sorrendje.

3. GSM és GPS alapú mérések és eredmények ábrázolása

Ebben a fejezetben a turisztikai alkalmazás projekten belüli feladatom közvetlen technikai háttere és az általam megvalósított feladatokról lesz szó. A fejezetben sok kép található, mert a vizuális megfigyelés és ez alapján rendszerezés adat a feladat jelentős részét.

3.1 Bevezetés

Az önálló labor témámat ez a fejezet határozza meg, ezzel foglalkoztam a legtöbbet. Azért volt lényeges, hogy a mért vagy eredményként kapott értékek ábrázolva legyenek, hogy szemmel is ellenőrizhető legyen a kapott eredményeink helyessége, ezzel bizonyítva, hogy jó irányba haladnak az elképzeléseink, legfőképpen a tömörítés terén. Mivel többen több irányba kezdtek el kutakodni az adatredukció témájában, azt reméltük, hogy a kapott eredmények valamiféle hasonlóságot mutatnak, minden esetben azonosak lesznek a meghatározó pontok.

3.2 A felépített rendszer

A feladat megvalósításának a kezdetén még nem volt elérhető online adatbázis (mobilhely.ikt.ppke.hu), azért a saját gépemen hoztam létre egy lokális adatbázis szerveret, amely képes földrajzi koordinátákat könnyedén kezelni.

Ez azzal kezdődött, hogy feltelepítettem Windows 7 operációs rendszerre a PostgreSQL adatbázis rendszert, amit kiegészítettem a PostGIS-sel, hogy megfelelő módon lehessen a földrajzi koordinátákat kezelni. Az adatbázisom alapja a korábban mért adatok egy mentése volt, mely megfelelő számú mérést tartalmazott a produktív munka elkezdéséhez. A félév során ez az adatállomány folyamatosan bővült és végül a szerver is elkészült. Jelentős előrelépés volt, hogy most már a szerveren is PostgreSQL környezet van PostGIS-sel kiegészítve.

A saját számítógépen még szükség volt egy Java fejlesztő környezetre. Ennek köszönhetően össze tudtam kapcsolni az adatbázisból történő lekérdezéseket és egy kml fájl létrehozását. Ehhez szükséges volt egy ODBC (Open DataBase Connectivity), egy szabványos adatbázis hozzáférési metódus hozzáadása az adott projekthez

3.3 Adatbázis séma

Az önálló labor során a régebben kialakított adatbázis sémát használtam fel, ami MySQL alapú volt. Mivel én PostGIS-sel kiegészített PostgreSQL-ben dolgoztam a félév során, ezért némiképp át kellett

alakítanom a táblákat létrehozó parancsokat. Én leginkább a GPS és GSM méréseket tartalmazó táblákkal foglalkoztam melyek felépítése a következőképpen néz ki:

GSM tábla:

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

- id: egy saját azonosító ID, minden bejegyzéshez tartozik
- gid: GPS ID, a GPS táblában ezen keresztül megtalálható az adott mérés helykoordinátája
- cid: Cell table ID, kapcsolódási pont a cell táblához
- rxlev: rssi, a vett GSM jel erőssége

A többi attribútum a mérő eszköztől függően megjelenik vagy nem, ezért nem részletezném őket, mert jellemzően a telefonos méréseknél, amivel én is képes voltam méréseket készíteni, ezekhez az attribútumokhoz tartozó értékek hiányoznak.

GPS tábla:

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

- id: saját azonosító ID, minden GPS mérési bejegyzéshez tartozik
- mid: measure ID, meghatározza, hogy melyik mérés során készült a rekord
- lat: latitude, az észak-déli szélességi fok
- lng: longitude, a kelet-nyugati hosszúsági fok
- sats: a látott műholdak száma
- alt: altitude, a tengerszint feletti magasság

Ebben a táblában még nem szerepel a geometria típus, ezt később kell hozzáadni, elég az adatok feltöltése után, csak a későbbiek folyamán futtatni kell egy UPDATE statement-et. A geometria típust a PostGIS egy függvényével lehet hozzáadni egy adott táblához, ez az eljárás az AddGeometryColumn(). Az általam használt kifejezés a következőképpen néz ki:

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

Az eljárás a következő paraméterekkel rendelkezik:

- gps: az a tábla neve, ahová a geom típusú oszlopot be akarom szúrni
- gps_geom2: az új oszlop elnevezése
- POINT: a geometria típusa
- 2: a pont dimenziószáma
- 4326: ez az SRID (Spatial Reference IDentifier), a használt koordináta rendszer azonosítója

A geometria típusa lehetne még LINESTRING, POLYGON, vagy ezek többszörös multi változatai (pl.: MULTIPOINT) és GEOMETRYCOLLECTION, ami a többféle geometria gyűjtőneve. Az SRID-kről annyit érdemes tudni, hogy a jelentős földrajzi azonosítási rendszernek, illetve minden országnak van saját SRID-je. Magyarországa a 23700 számú SRID. A GPS rendszerek a 4326-os SRID-t használják, így kompatibilis lesz az adatbázis a Google Earth-el. Miután megtörtént az adatok feltöltése a következő UPDATE statement-et kell lefuttatni, hogy a most létrehozott oszlopnak értékeket adjunk.

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

Ez a következőképpen működik:

- az adatbázisban megtalálhatók a lng (hosszúsági) és lat (szélességi) fok értékek, a következő formában: 1913.0251; 4726.3264
- ezt az értéket osztom 100-al

- a tizedes részt osztom 60-al és szorzom 100-al
- eredmény: 19.217085; 47.438773

Erre azért van szükség, mert a vevő által rögzített érték első két számjegye a fok, a második két számjegye a szögperc, a további számok a szögperc törtrészei, de a Google más számformátumot használ (amit az eredmény mellett láthatunk), ezért szükséges az átalakítás.

3.4 Ábrázolt adatok

A félév során folyamatosan bővítettük és rendszereztük az adatbázist, ezzel az volt a célunk, hogy minél jobban megismerjük a mérési eredményeket, felismerjük a különböző hibákat, rájövünk az összefüggésekre, különböző rendszerintegritások vegyünk észre. Erre egy kézenfekvő módszer, hogy a mért adatokat ábrázoljuk, különböző szűréseket végzünk, majd az így kapott eredményt is ábrázoljuk. Így felfedezhetők olyan változások is, amelyek esetlegesen egy lekérdezés eredményeként nem szembetűnőek, de ábrázolva a kapott adatokat szemmel láthatóak.

A feladatom fő része a többiek által eredményül kapott adatok ábrázolása volt, ezen kívül olyan speciális lekérdezések elkészítése, amelyek kihasználják a földrajzi adatbázis nyújtotta előnyöket (beépített geometriai függvények, gyorsindexelés a geometriai adatoknak, stb.). Ezeket általában egyszerűen képernyőn megjeleníthető formátumra hoztam (Google Earth-re rajzolással), de volt olyan is, hogy az eredményül kapott adatokkal tovább dolgoztam, színeztam, tovább szűrtem, és csak ezek után ábrázoltam.

Az ismerkedést a környezettel a már előre mások által megírt kml fájlok ábrázolásával, tanulmányozásával kezdtem, majd egyszerű kml fájlokat írtam, mint például az egyetem bejáratát jelölő pont.

Az egyetem bejárata:



A mérések ábrázolását egyszerű lekérdezésekkel folytattam, mint például az összes budapesti mérés helyének ábrázolása. A mérések GPS koordinátáit a következő SQL paranccsal tudtam lekérdezni:

```
SELECT ST_AsKML(gps_geom2) FROM gps;
```

Az ST_AsKML() függvény készíti el a geometria típusú adatokból KML formátumú szöveges kimenetet, amely így néz ki:

```
<Point><coordinates>19.092421666666667,47.478088333333332</coordinates>
</Point>
<Point><coordinates>19.092388333333332,47.478101666666667</coordinates>
</Point>
<Point><coordinates>19.083918333333333,47.482643333333336</coordinates>
</Point>
<Point><coordinates>19.084021666666668,47.482705000000003</coordinates>
</Point>
```

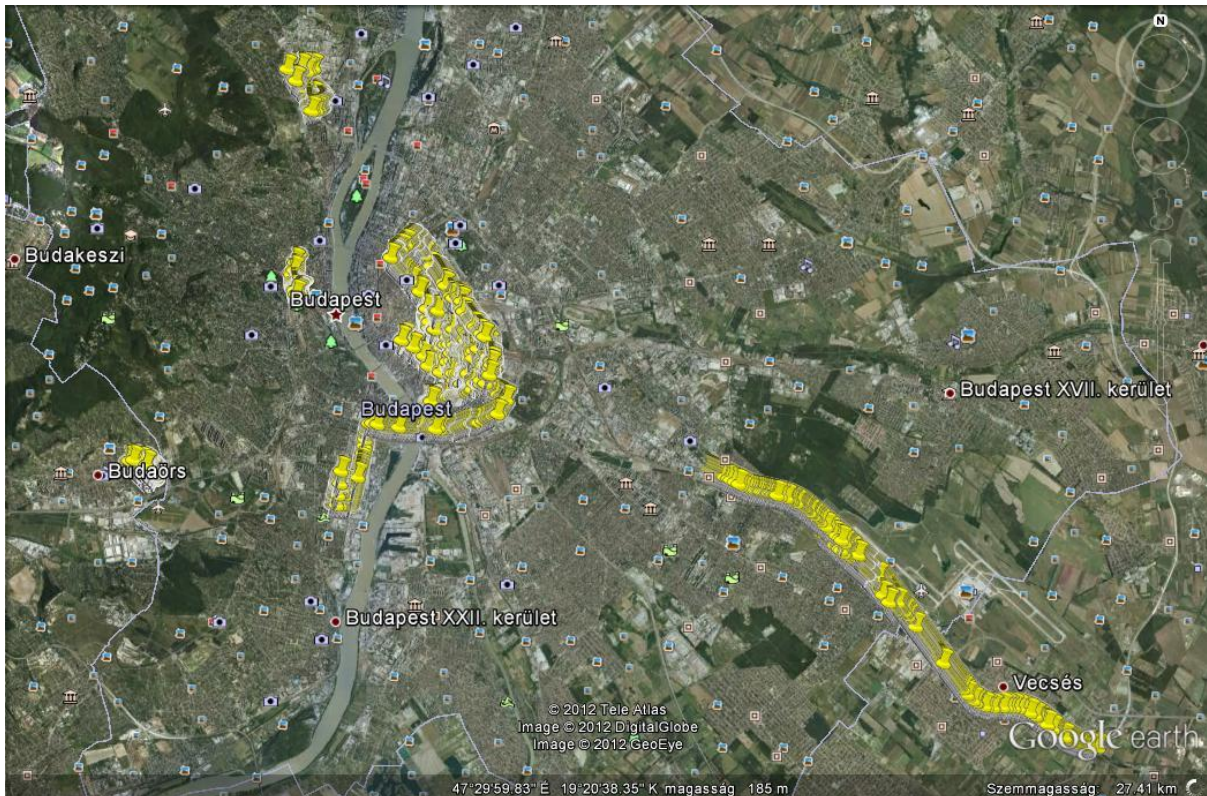
Ennek a lekérdezésnek az eredményét a Java programon belül, aminek a feladata volt, hogy a kimenetet ellássa szabályos kml fej és lábrésszel, még szűrtem, hogy csak a budapesti mérések kerüljenek a kml fájlba, a következő elágazással:

```

double bplng1 = 18.932686;
double bplng2 = 19.304075;
double bplat1 = 47.367297;
double bplat2 = 47.553653;
if(((lng3<bplng2) && (lng3 > bplng1)) && ((lat3<bplat2) && (lat3>bplat1))){
    pw.println("<Placemark>");
    pw.println(res.get(i));
    pw.println("</Placemark>");
}

```

Az így kapott Budapest.kml fájlban található pontok a következőképpen néznek ki:



A későbbiekben feladatomná vált a mások által elért eredmények ábrázolása különböző kikötésekkel.

[8] Réti Dániel a GSM tábla alapján három csoportba sorolta a GPS koordinátákat:

- Az adott pontban hét cella ID egyezik
- Az adott pontban hétnél kevesebb cella látható
- Az adott pontban hétnél több cella látható

Ezeket a koordináta csoportokat színekkel különböztettem meg egymástól, sárgával jelöltem, ha hét cella ID egyezik, feketével, ha hétnél kevesebb cella ID látható, pirossal, ha hétnél több. Mindezt ábrázolva, a képen az egyetem épülete, és a parkoló látható. Jól megfigyelhető, hogy egyes esetek milyen gyakran fordulnak elő. A pirossal jelölt csoport egy rendszerintegritás, alapvetően egy eszköz

nem láthat hétnél több cellát egyszerre, de az is megfigyelhető, hogy általában egy eszköz érzékeli a hét cellát.

Az egyetem épülete és a parkoló, három csoportba sorolva a mérési pontokat:



Ebből továbblépés volt, hogy olyan csoportokat ábrázoljak, amelyek közös tulajdonsága az, hogy az érzékelt hét cella ID megegyezik. Ehhez szükség lekérdezés megegyezett az előzővel:

```
SELECT ST_AsKML(gps_geom2), minim FROM gangl_3;
```

GPS adatokat kérünk le és a minim mező lényegében a csoportazonosító. A színek meghatározásához ez állt rendelkezésemre, elsődleges cél az volt, hogy két megegyező szín ne szerepelhessen az ábrázolás során. Ezt egy HashSet segítségével oldottam meg, továbbá fontos volt a láthatóság szempontjából, hogy ne minden szín szerepelhessen a képen. A következő ciklus segítségével színeztam a csoportokat:

```
for (int i = 0; i < res_7.size(); i++) {  
    pw.println("<Placemark>");  
    pw.println("<Style><IconStyle>");  
    if(tmp==Integer.parseInt(res_7_min.get(i))) {  
        pw.println("\t<color>" + color + "</color>");  
    }  
    else {
```

```

tmp=Integer.parseInt(res_7_min.get(i));
String color2="ff";
for (int j = 0; j < 6; j++) {
    color2+=Colors[randomGenerator.nextInt(15)];
}
if(!s.add(color2)){
    while(!s.add(color2)){
        for (int j = 0; j < 6; j++) {
            color2+=Colors[randomGenerator.nextInt(15)];
        }
    }
}
color=color2;
}
}

```

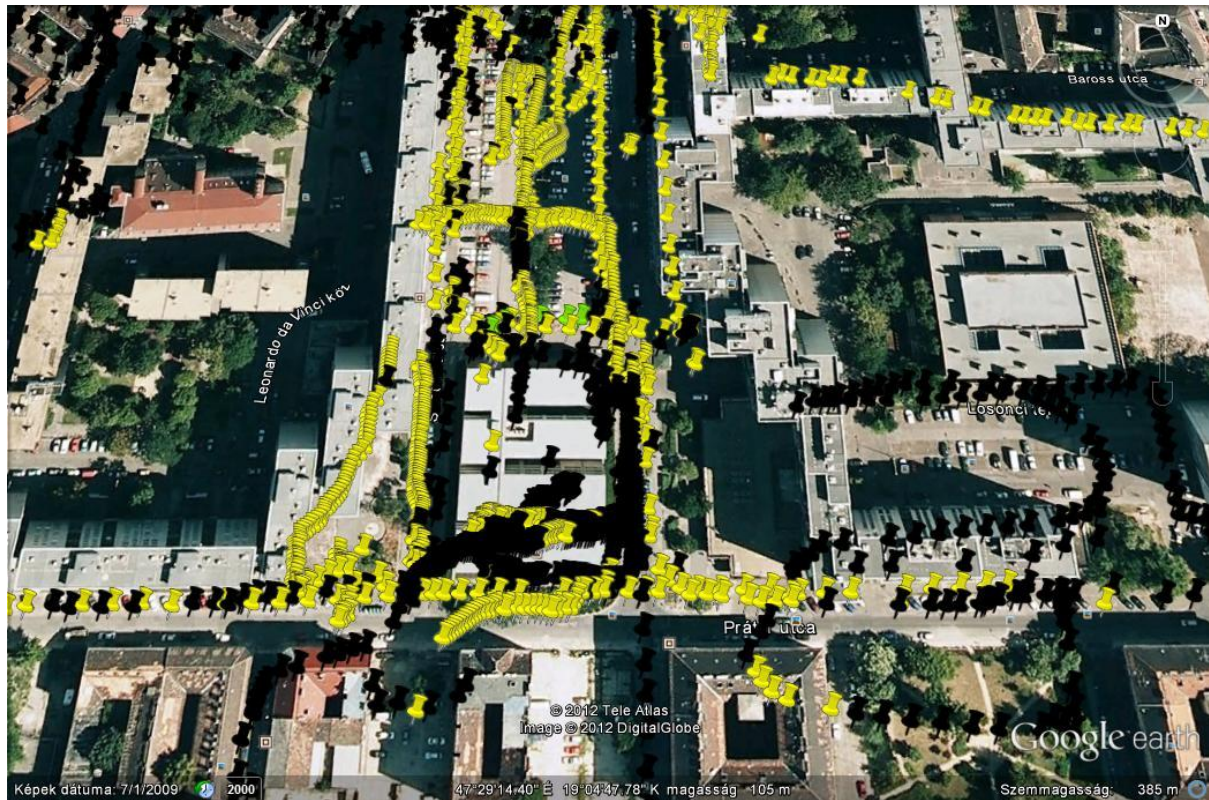
Az eredményül kapott kml fájl egy részlete ábrázolva:



Ezen szűrések alapján megállapítható, hogy a mért adataink között 1635 csoport van, amiben közösek a cella ID-k. A könnyebb kezelhetőség érdekében színek szerint külön kml állományokba rendeztem. Ezekben a fájlokban három szín található meg: sárga, fekete zöld. A zöld szín jelenti azokat a pontokat, ahol a hét cella ID egyezik, a sárga pontok jelentése, hogy hétnél kevesebb cella látható, a fekete pontok a másik csoportba tartozó pontokat jelentik. A 1600 és 1634 közötti sorszámú kml fájlok ábrázolásából egy videó fájlt is készítettem, ami egy útvonalat ír le, melynek főbb állomásai:

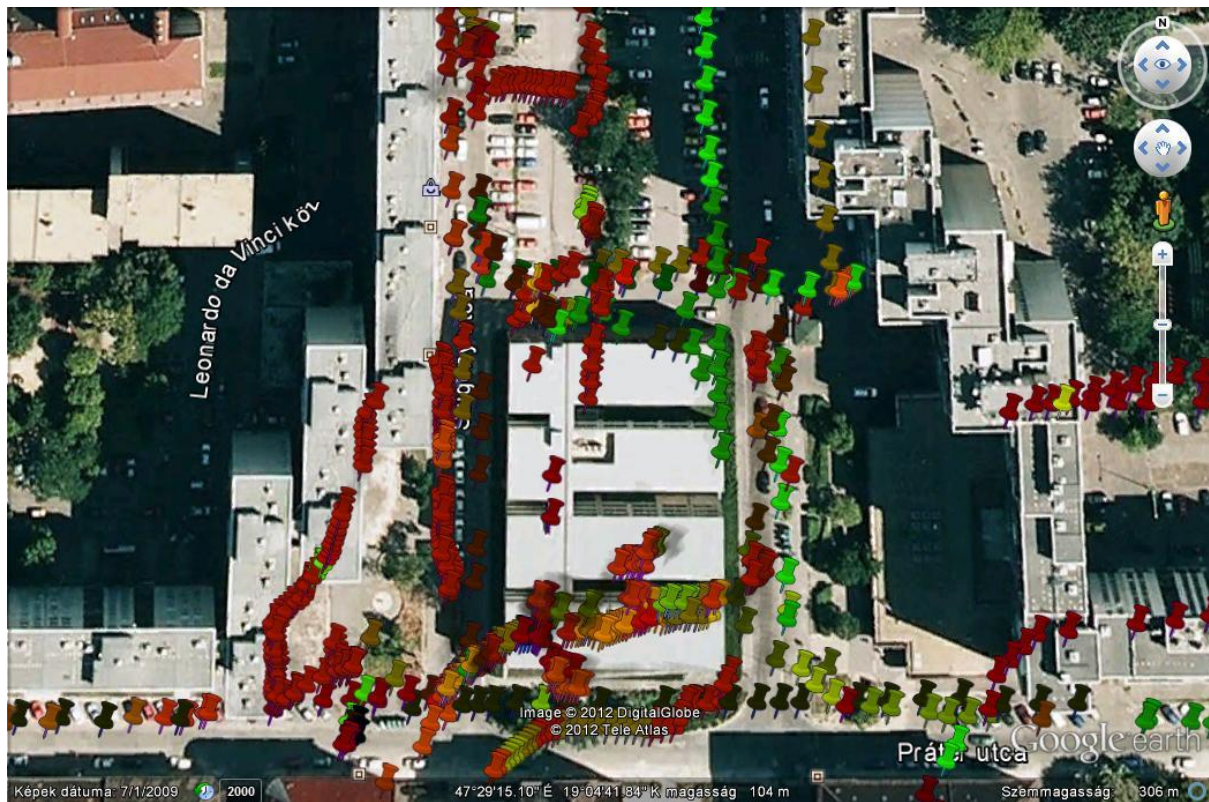
6-os út és a Galvani utca sarka, Szerémi út, Lágymányosi híd, Könyves Kálmán körút, Üllői út, Orczy út, Ludovika tér, végül ez egyetem környéke.

Az 1634.kml fájl:



A csoportból többen több úton indultak el a mérési pontok közös tulajdonságainak keresésével, [9] Markó Zoltán főkomponens analízist hajtott végre az adathalmazon, melynek célja az volt, hogy ugyancsak 1600 körüli darabszámú csoportba lehessen szétválasztani az adatbázist. Ezt 6,7 %-os threshold-dal tudta elérni.

Az így kapott eredmények ábrázolása az egyetem környékén:



[8] Réti Dániel másik elképzelés volt, hogy az is lehet közös tulajdonság, ami alapján tömöríteni lehet, ha a pontosan 6 cella ID-t tároló bejegyzésekben RX szint nagyság szerint sorba rendezzük a cella ID-ket, majd szűrjük az így kapott eredmény az alapján, hogy hány darab közös cella ID található a bejegyzésekben. Az alábbiakban ez a szűrőfeltétel az volt, hogy legalább négy cella ID megegyezzen.

A szűrés alapján a következő módon néznek ki a mérési pontok az egyetem környékén:



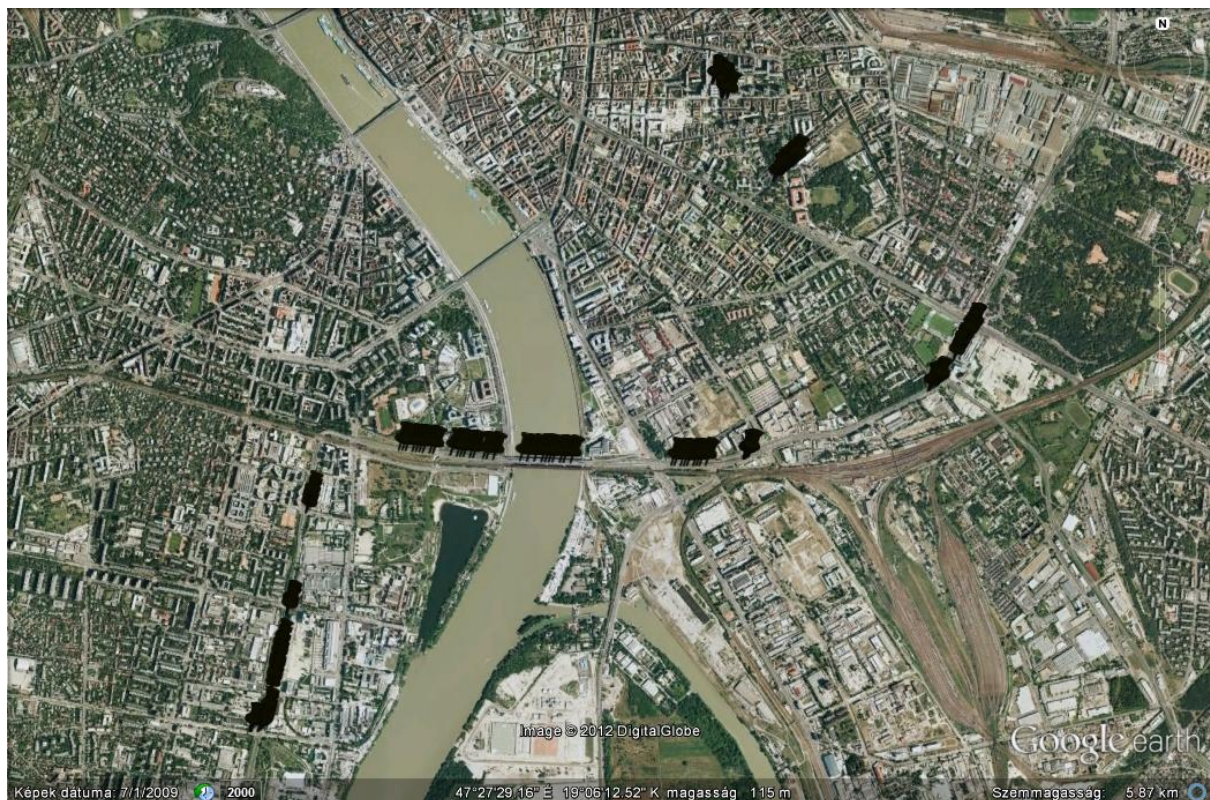
Az eddigi eredmények összehasonlítása végett létrehoztam még egy java programot. Működése a következő: abból a lekérdezésből, ami a hét egyező cella ID-t határozta meg és csoportosítva volt színek szerint, elmentettem azon pontok koordinátáit, amelyek az 1600 és az feletti sorszámú csoportokba tartoztak. Ezek után megvizsgáltam, hogy ezek a koordináták szerepelnek-e a főkomponens analízis során kapott eredménnyel, illetve Réti Dániel szűrése szerint.

Eredményül a következő kimeneteket kaptam: a hét cella egyezése során a 35 színből 224 pont került elmentésre. Ebből a főkomponens analízis által kapott eredmény 163 pontban mutatott egyezést, a szűrések alapján 161 esetben adta ugyanazt a pontot. Ezekből az eredményekből kml fájlokat készítettem és ábrázoltam, és a következő képeket kaptam:

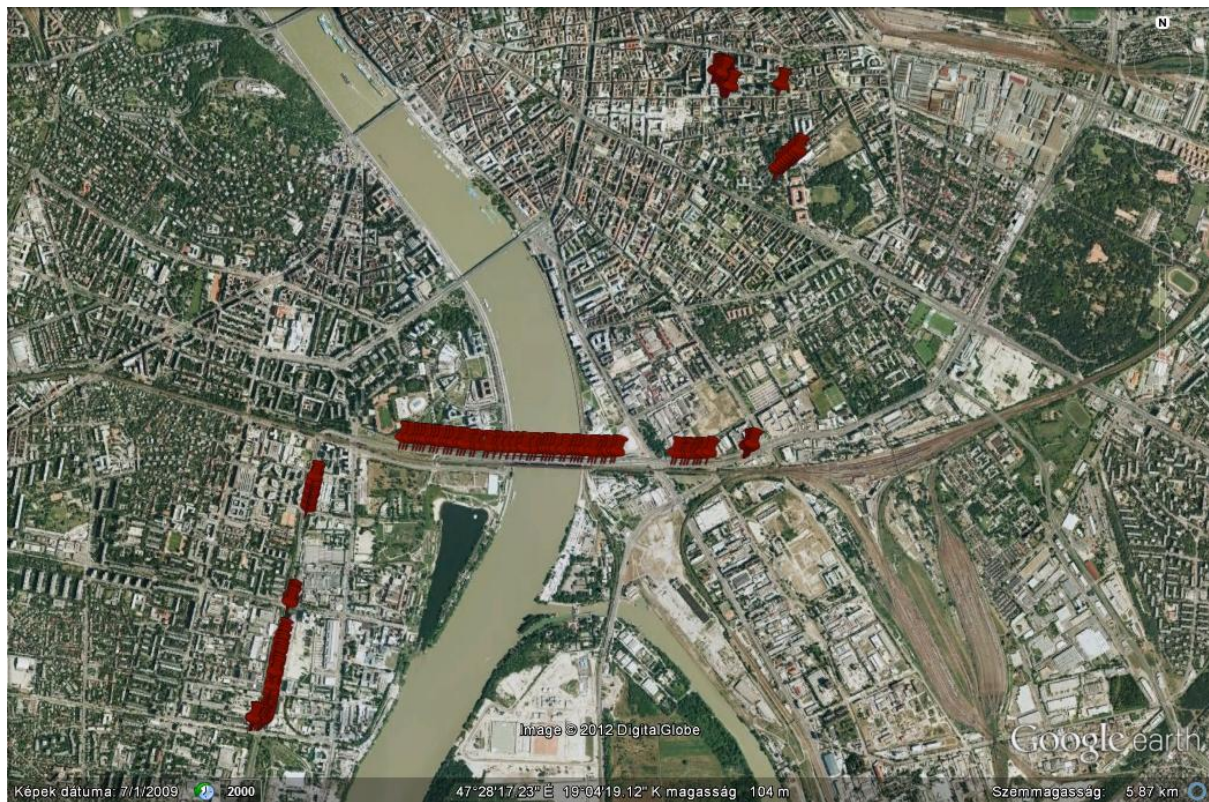
A hét egyező cella ID alapján:



A szűrés alapján:

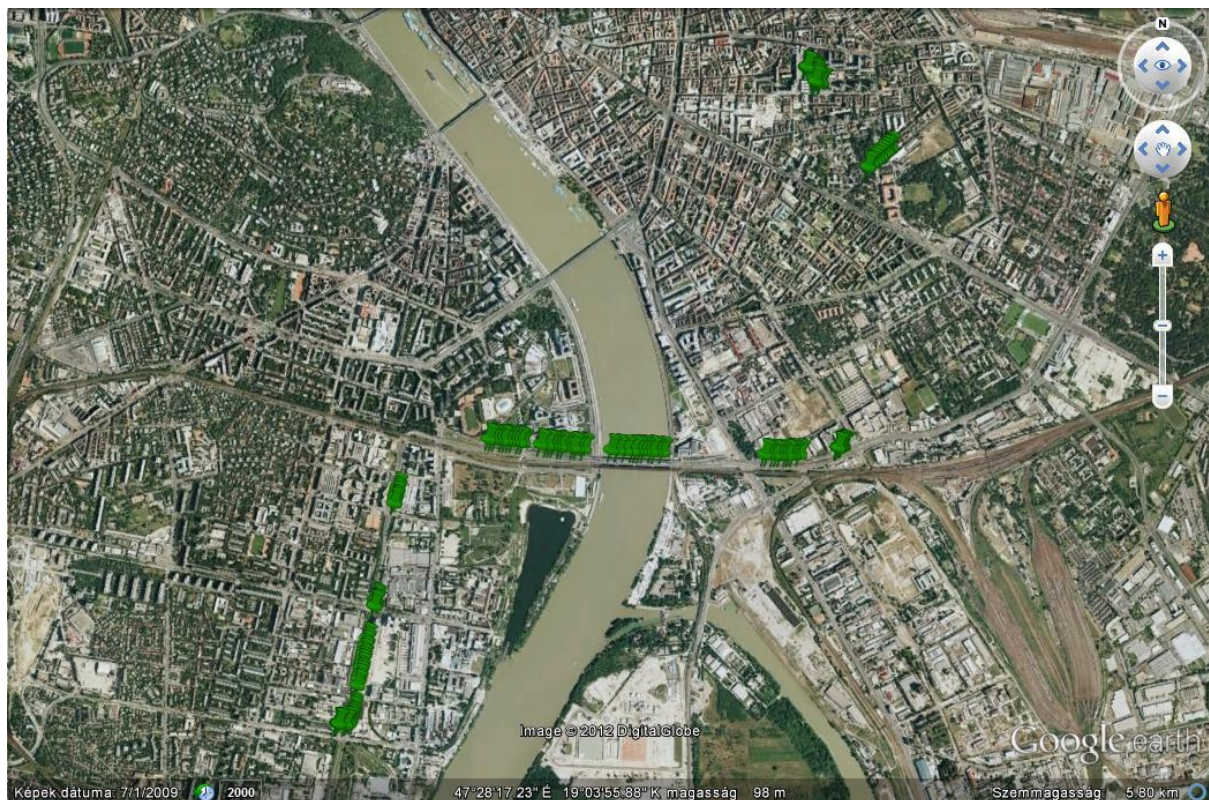


A főkomponens analízis alapján:



Legvégül összehasonlítottam, hogy a főkomponens analízis és a szűrések során kialakult pontok közül melyek egyeznek meg. Eredményül azt kaptam, hogy 120 esetben ad a két módszer közös pontot.

Egyező pontok:



4. Konklúzió

Értékelhető eredményként azt tudom listázni, hogy a főkomponens analízis és a szűrési algoritmus közel azonos mértékben képes az adatbázis mértékének a csökkentésére, de jelentős különbségek találhatók az adatredukció után kapott adatbázisokban eltárolt információkban.

Ezen különbségek megértése és magyarázatának megtalálása előrébb viheti az adatbázis tömörítési eljárások használatát, egy hasonló jellegű adatbázisok esetén, így a célul kitűzött turisztikai alkalmazás megfelelően tömörített offline adatbázissal készülhet el.

5. Irodalomjegyzék

- [1] Takács Gy. 2008. Helymeghatározás mobiltelefonnal és mobil hálózattal. Híradástechnika 63. évf. (8): p. 20-27
- [2] GIS figyelő (2012. május 10.): http://gisfigyelo.geocentrum.hu/kisokos/kisokos_gis.html
- [3] Wikipédia (2012. május 11.): <http://hu.wikipedia.org/wiki/PostgreSQL>
- [4] HupWiki (2012. május 11.): <http://wiki.hup.hu/index.php/PostGIS>
- [5] Wikipédia (2012. május 13.): http://hu.wikipedia.org/wiki/Google_F%C3%B6ld
- [6] Google Earth (2012. május 13.): <http://www.google.com/intl/hu/earth/index.html>
- [7] Wikipédia (2012. május 13.): http://hu.wikipedia.org/wiki/Keyhole_Markup_Language
- [8] Réti Dániel önálló labor beszámolója
- [9] Markó Zoltán önálló labor beszámolója