Rakennusrekisterikartasta osoite- ja postinumerokartaksi

Jukka Rahkonen, http://latuviitta.org

Viimeksi muutettu 18. syyskuuta 2012

Helsingin kaupungin rakennusrekisteriote v. 2012 on ollut saatavilla avoimena aineistona 12. syyskuuta 2012 alkaen. Aineistoon liittyviä linkkejä:

Aineiston esittely Helsinki Region Infoshare -sivustolla
http://www.hri.fi/fi/data/helsingin-kaupungin-rakennusrekisterin-ote-62012/

Helsingin kaupungin avoimien aineistojen lataussivu
http://kartta.hel.fi/avoindata/

Aineiston käyttöoikeudet
http://kartta.hel.fi/avoindata/aineistot/Kartta_avoindata_kayttoehdot_v02_3_2011.pdf

Aineiston haltuunotto

Aineisto on ladattavissa MapInfo TAB -muodossa, eikä sen käyttöönotossa ole mitään ihmeellistä. Jos käytetään avoimen lähdekoodin GDAL-ohjelmistoa ja tavoitteena on muuntaa aineisto johonkin UTF-8-merkistökoodausta käyttävään järjestelmään, niin muunnos on tehtävä mutkan kautta käyttämällä GML-muotoa välivaiheena. Lähtöaineistossa käytetty merkistökoodaus on ISO-8859-1.

```
ogr2ogr -f gml -t_srs epsg:3067 hki_rakennukset_2012.gml
rakennukset_Helsinki_06_2012_etrsgk25.tab --config OGR_FORCE_ASCII no
```

Avataan syntynyt GML-tiedosto, jonka ensimmäinen rivi ilmoittaa virheellisesti, että tiedosto käyttäisi UTF-8-koodausta. Muokataan ensimmäistä riviä niin, että merkistökoodaus ilmoitetaan oikein.

Ennen korjausta
```
<?xml version="1.0" encoding="utf-8" ?>
```
Korjauksen jälkeen
```
<?xml version="1.0" encoding="iso-8859-1" ?>
```

Lisää lukemista MapInfosta, GDAL:sta ja merkistökoodauksesta on ohjeessa
http://latuviitta.org/documents/Mapinfo_GDAL_ogr2ogr_ja_UTF-8.pdf

Seuraava komento tekee korjatusta GML-tiedostosta Spatialite-tietokannan

```
ogr2ogr -f sqlite -dsco spatialite=yes -dsco init_with_epeg=yes -t_srs epsg:3067
-s_srs epsg:3067 hki_rakennukset_2012.sqlite -nlt multipolygon
hki_rakennukset_2012.gml
```

Aineiston jalostaminen, osa 1: Tee osoitepisteaineisto

Rakennusrekisteriaineistossa rakennukset on esitetty aluekohteina ja yhtenä rakennusten ominaisuustietona on rakennuksen osoite, joka on kirjoitettu kokonaisena merkkijonona yhteen kenttään, esimerkiksi ”Pohjoisranta 21 00170 HELSINKI”.

Tehdään aineistosta huvin ja harjoituksen vuoksi pisteaineisto, jolla annetaan ominaisuustiedoksi lyhytosoite kadunnimi-osoitetunnus ja postinumero omaan kenttäänsä. Edellisen esimerkin osoitekenttään tulisi siis ”Pohjoisranta 21” ja postinumerokenttään ”00170”.

Vaihe 1: Aluekohteista pisteitä

Koska tiedot nyt ovat tiedokannassa, niin luodaan pisteaineisto tietokannassa. Spatialiten funktio ST_PointOnSurface luo pisteen, joka on jossain lähtögeometriana olevan alueen sisällä.

```
CREATE TABLE hki_rakennusosoite_2012
AS SELECT OGC_FID, ST_PointOnSurface(geometry) as geometry,
rakennustunnus, osoite
FROM hki_rakennukset_2012
```

Vaihe 2: Pitkän osoitteen osittaminen

Pitkän osoitteen voi jakaa osiin esimerkiksi käyttämällä aineiston tekstinkäsittelyohjelmassa, koska Spatialite osaa sekä viedä että tuoda tekstitiedostoja.

Alkutilanne (huomaa tyhjä merkkijono geometry-ominaisuustiedolla):

```
OGC_FID,geometry,osoite
1,,Pohjoisranta 21 00170 HELSINKI
```

Tehdään kaksi etsi-korvaa -toimintoa: ” HELSINKI” muutetaan muotoon ”,HELSINKI” (välilyönnin sijaan pilkku) ja vastaavasti muutetaan ” 00” muotoon ”,00”.

Lopputilanne:

```
OGC_FID,geometry,osoite,postinro,kaupunki
1,,Pohjoisranta 21,00170,HELSINKI
```

Tämä on ilman muuta hölmö tapa merkkijonon pilkkomiseen, mutta tällä aineistolla se toimii, käy nopeasti ja tuottaa oikean lopputuloksen.

Seuraavaksi muokattu tekstitiedosto tuodaan takaisin kantaan ”lyhytosoite” -nimiseen tauluun, ja kantaan luodaan näkymä, johon yhdistetään edellisessä vaiheessa luodusta rakennuspistetaulusta pistegeometria ja rakennustunnus.

```
CREATE VIEW "osoitenakyma" AS
SELECT "a"."ROWID" AS "ROWID", "a"."geometry" AS "geometry",
    "a"."rakennustunnus" AS "rakennustunnus",
    "b"."osoite" AS "osoite", "b"."postinro" AS "postinro"
FROM "hki_rakennusosoite_2012" AS "a"
JOIN "lyhytosoite" AS "b" USING ("OGC_FID")
```

Lopputuloksena on näkymä, joka on määrittelyn mukainen.

```
select * from osoitenakyma limit 2;

ROWID  geometry           rakennustunnus            osoite              postinro
1      BLOB sz=60 GEOMETRY 091-001-9901-0100-001     Pohjoisranta 21     00170
2      BLOB sz=60 GEOMETRY 091-001-9901-0100-002     Siltavuorenranta 16 00170
```

Aineiston jalostaminen, osa 2: Tee postinumeroaluekartta

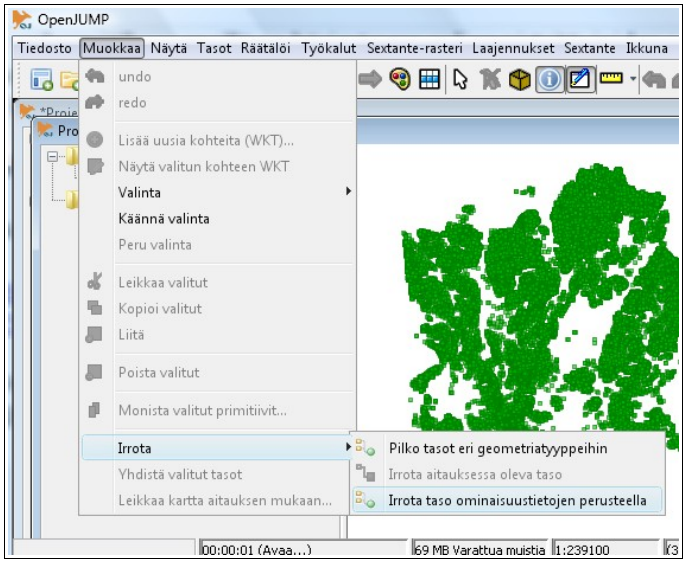
Edellisen vaiheen jälkeen meillä on käsissämme paljon pisteitä, joiden ominaisuustietoihin kuuluu postinumero. Tulee mieleen, että jos saman postinumerotiedon omaavat pisteet sulkisi kunkin oman alueensa, niin tulokseksi saataisiin postinumeroaluekartta. Ei muuta kuin kokeilemaan.

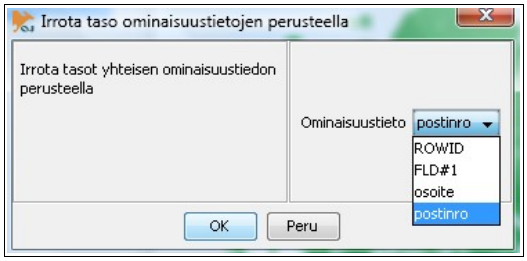
Vaihe 1: Pisteaineiston pilkkominen postinumeroalueittain

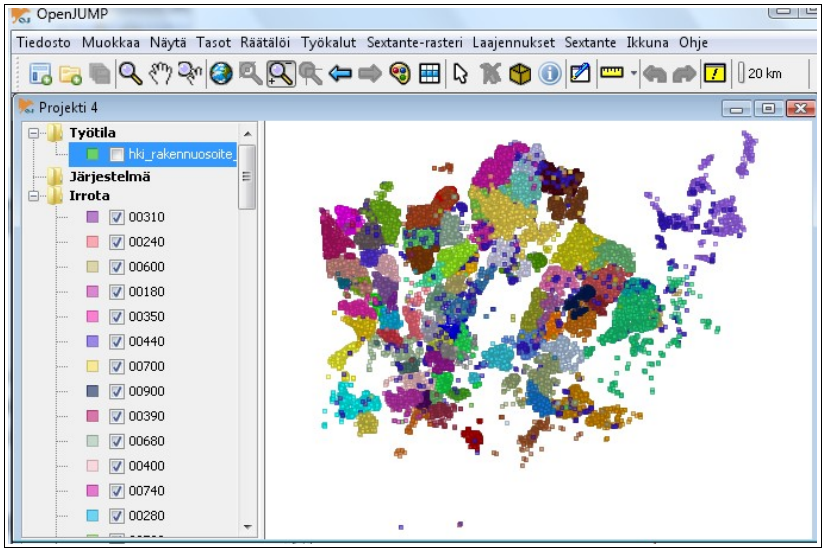
Käytetään kokeilussa OpenJUMP-ohjelmaa ja sen Conclave hull -lisäosaa.

http://openjump.org
http://www.rotefabrik.free.fr/concave_hull/concave_hull_dist.zip

Tallennetaan Spatialiten osoitepistetaulu shapefile-muotoon, avataan se OpenJUMP:ssa ja pilkotaan tasoiksi ”postinro” -kentän arvojen perusteella.

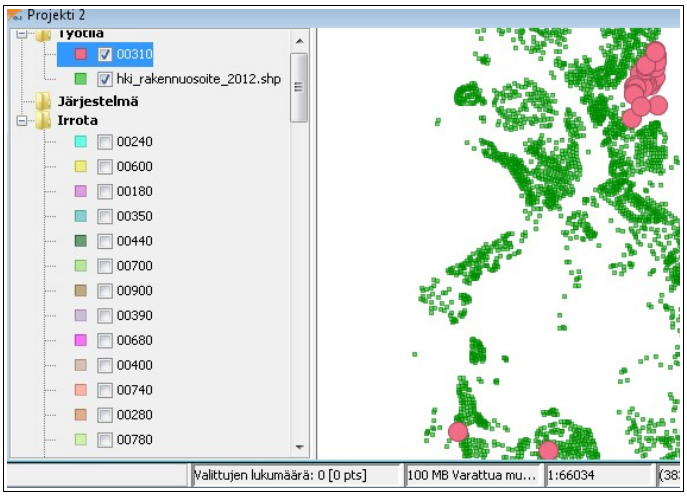






Vaihe 2: Korjataan ilmeisiä virheitä aineistosta

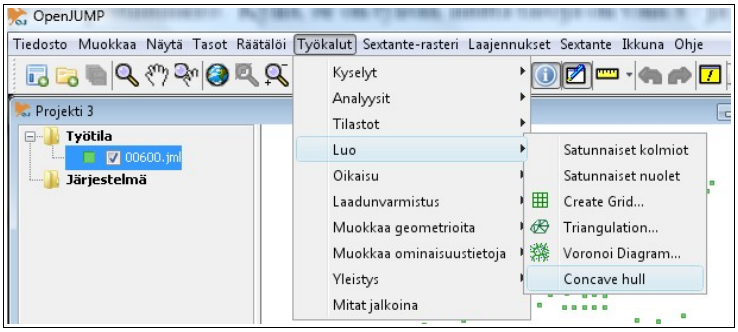
Kun silmäillään postinumeroittain pilkottuja tasoja, niin huomataan, että kaikilla tasoilla on rakennuspisteitä,joiden sijainti ja niille tallennettu postinumero eivät näytä sopivan yhteen.

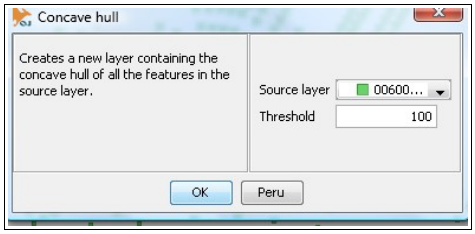


Käydään siis käsin läpi jokainen postinumerotaso ja tuhotaan niiltä ne pisteet, joille on ilmiselvästi tallennettu väärä postinumero. Kyllä, se on tylsää, mutta tasoja on vain 87 ja me teemme tätä vain huvin vuoksi.

Vaihe 3: Pisteiden sulkeminen alueiden sisälle

Lopuksi käytetään OpenJUMP:in lisäosaa ”Concave hull” jokaiselle postinumerotasolle vuorollaan.

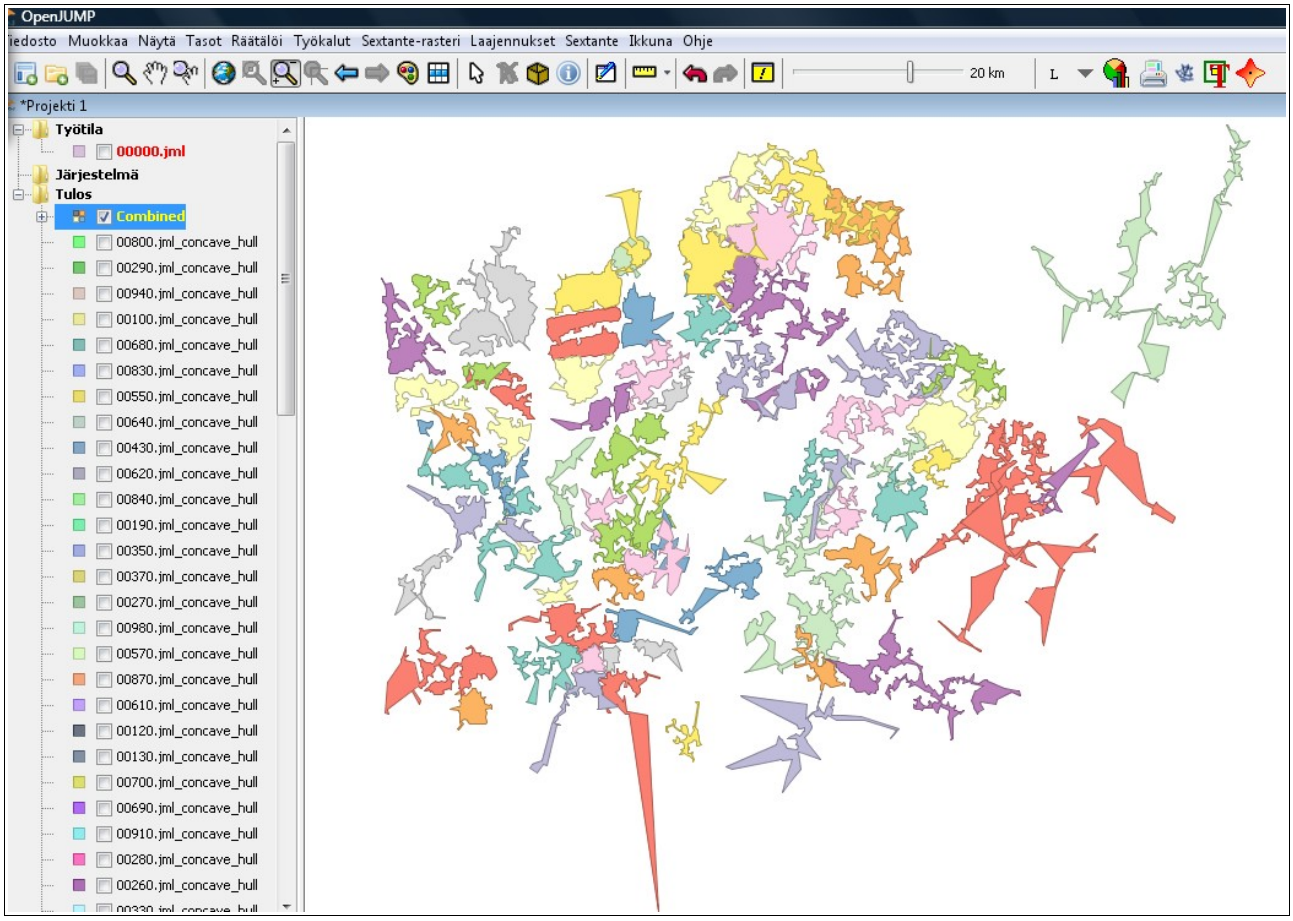




Siisteissä tapauksissa tuloksena on nätti postinumeroalue.

Toisinaan taas jäljelle näyttää jääneen pisteitä, jotka eivät oikein näyttäisi kuuluvan samaan joukkoon toisten kanssa.

Lopputuloksena on kuitenkin kartta, jossa selvästi häämöttää erillisiä ja yhtenäisiä alueita, ja jota parantelemalla epäilemättä olisi mahdollista tehdä käyttökelpoinen postinumerokartta.



Nopeammin, paremmin ja halvemmalla eli kriittinen tarkastelu

A) Menetelmä

Kenties tyhmintä, mitä tässä ohjeessa tehdään, on merkkijonon pilkkominen palasiin tekstinkäsittelyohjelmalla tietokannan ulkopuolella. Sen voisi varmasti tehdä tietokannassa SQL:llä paljon kätevämmin. Heikkonan puolusteluna voi sanoa, että jos näin pienen homman joutuu tekemään vain yhteen kertaan, niin käy nopeammin tehdä se tyhmällä tavalla, jonka osaa, kuin opetella järkevämpi tapa. Sitä paitsi, erotinmerkeillä jäsenneltyjen tekstitiedostojen siirtely tietokannasta ulos ja sisään on hyvä opetella sekin.

”Concace hull” on nykyisin tuettu sekä PostGIS- että Spatialite-tietokannoissa. OpenJUMP:ssa tehdyt postialueiden luomiset olisi nekin voitu tehdä suoraan SQL:llä. Aineistovirheiden takia oli kuitenkin mukavampaa tarkastaa omin silmin, tuottiko lopputulos eden osapuilleen järkevän alueen lopputuloksena.

Valtaosa tämän harjoituksen vaatimasta ajasta kului väärien postinumeroiden takia hassuissa paikoissa olevien pisteiden aiheuttamien ongelmien pienentämiseen. Ohjelmien vaatima prosessointiaika oli korkeintaan viisi minuuttia, joten siltä suunalta ei ole juuri saatavissa lisää tehokkuutta.

ST_PoinOnSurface tuotti jonkin verran tyhjiä geometrioita. Kyseessä voi olla virhe ohjelmassa tai sitten lähtöpolygonien geometria ei ollut kunnossa, mutta en vaivautunut selvittämään tätä sen tarkemmin. Joku muu ohjelma saattaisi selvitä tässä suhteessa paremmin kuin Spatialite.

B) Aineisto

Rakennusrekisterissä voisi kuvitella rakennusten katuosoitteiden olevan yleensä oikein, joten sen perusteella muodostettu osoiteaineisto on todennäköisesti hyvälaatuinen. Koska aineistossa rakennuksella on vain yksi osoite, niin kadun kulmatonttien rakennuksilta puuttuu toisia osoitteita. Rakentamattomille tonteille ei luonnollisesti tästä aineistosta saada osoitteita.

Postinumero on selvästi rakennusrekisterissä sellainen tieto, joka rakennukselle on tallennettava, mutta jota ei sinänsä koskaan tarvita mihinkään. Siksi aineistossa on paljon vääriä postinumeroita, mutta rekisteriä on turha moittia siitä, sillä tarpeettoman tiedon ylläpitoon ei kannata kauheasti uhrata vaivaa.

Postinumerotiedon sumeuden vuoksi tämän aineiston luulisi kiehtovan ihmisiä, jotka kehittävät menetelmiä paikkatietojen automaattista laadunvarmistusta varten. Kuka keksii kaavan, joka löytää parhaalla osumaprosentilla väärin tallennettuja postinumeroita? Ja osaisiko yhtälö jopa ehdottaa mahdollisia oikeita postinumeroita?

Lopputulokseksi saadun postinumeroaluekartan jalostaminen hyvälaatuiseksi tuotteeksi vaatisi kenttätyötä. Kartasta löytyvien epäilyttävien osoitteiden oikea postinumero pitäisi tarkistaa jollain muutta menetelmällä, esimerkiksi kysymällä asukkailta. Työ olisi varmasti tehtävissä, mutta todennäköisesti Suomen osoitteet ja samalla postinumeroalueet ehditään julkistaa avoimena datana ennen sitä.