

Itellan perusosoitteiston haltuunotto ja postinumeroalueiden muodostaminen

Jukka Rahkonen, <http://latuviitta.org>

Lisenssi [Creative Commons Attribution 3.0 Unported](#)

Viimeksi muokattu 7. maaliskuuta 2013

Tiivistelmä

Itella osoitetiedot ovat olleet saatavilla 1. tammikuuta 2013 saatavilla avoimena datana. Perusosoitteisto on tekstimuotoinen luettelo, jossa jokainen katuosoite on liitetty kuntaan ja postinumeroalueeseen. Tämä aineisto ei ole paikkatietoaineisto, mutta siitä voidaan tehdä paikkatietoaineisto yhdistämällä se ”Maastotietokannan tiestö osoitteilla” -aineiston kanssa. Tässä ohjeessa neuvotaan ensimmäiset askeleen aineiston haltuunottamiseksi ja luodaan karkea postinumeroaluekartta.

Tietoa perusosoitteistosta:

<http://itella.fi/palvelutjatuotteet/postinumeropalvelut/perusosoitteisto.html>

Aineiston käyttöehdot:

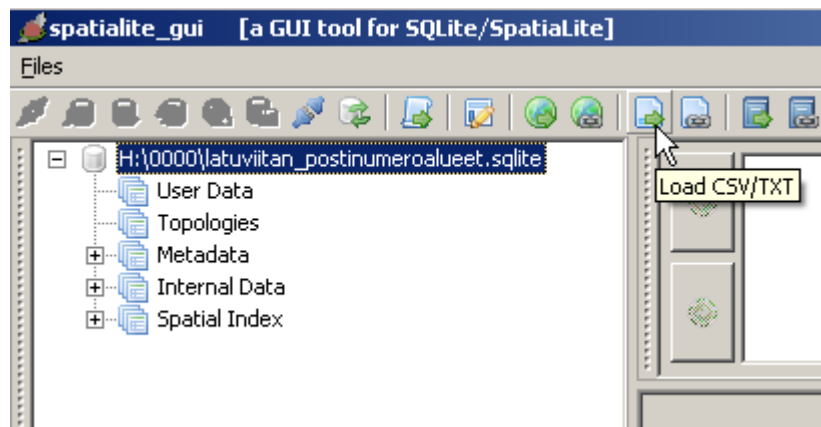
http://itella.fi/liitteet/palvelutjatuotteet/yhteystietopalvelut/postinumeropalvelut_palvelukuvaus_ja_kayttoehdot.pdf

Itellan tekstitiedosto ja sen vienti Spatialite-tietokantaan

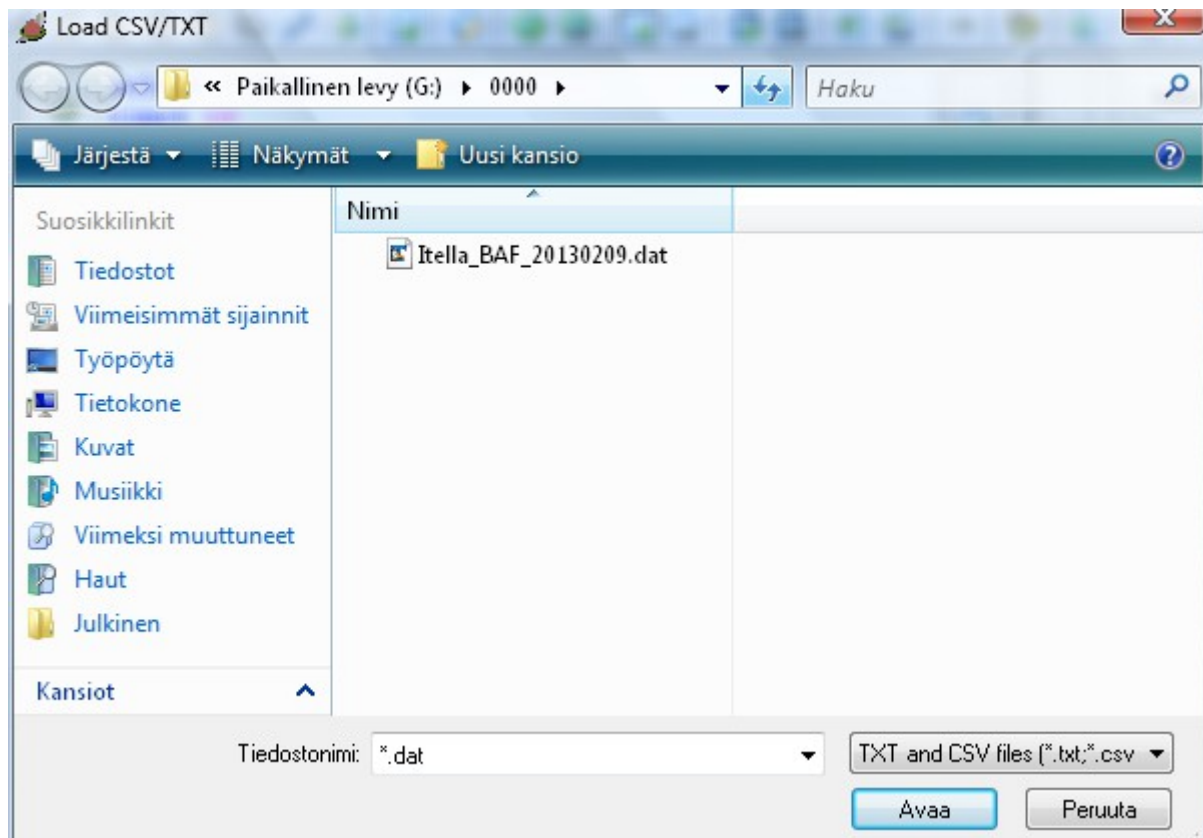
Itellan verkkosivuilta ladattavassa perusosoitteistossa jokainen osoite on kirjoitettu omalle rivilleen 256 merkkiä pitkänä määrämittaisena tekstinä. Tekstirivin rakenne on kuvattu palvelukuvauksessa. Koska tiedosto on myöhemmin tarkoitus yhdistää paikkatietoaineistoon, niin osoitetiedosto kannattaa ensimmäiseksi vielä johonkin tietokantaan, jossa on myös paikkatieto-ominaisuuksia. Seuraavassa näytetään, kuinka aineisto luetaan sisään Spatialite-tietokantaan Spatialite-gui -ohjelman avulla.

Vaiheet:

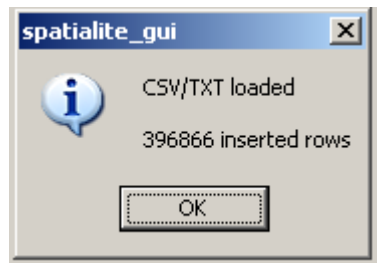
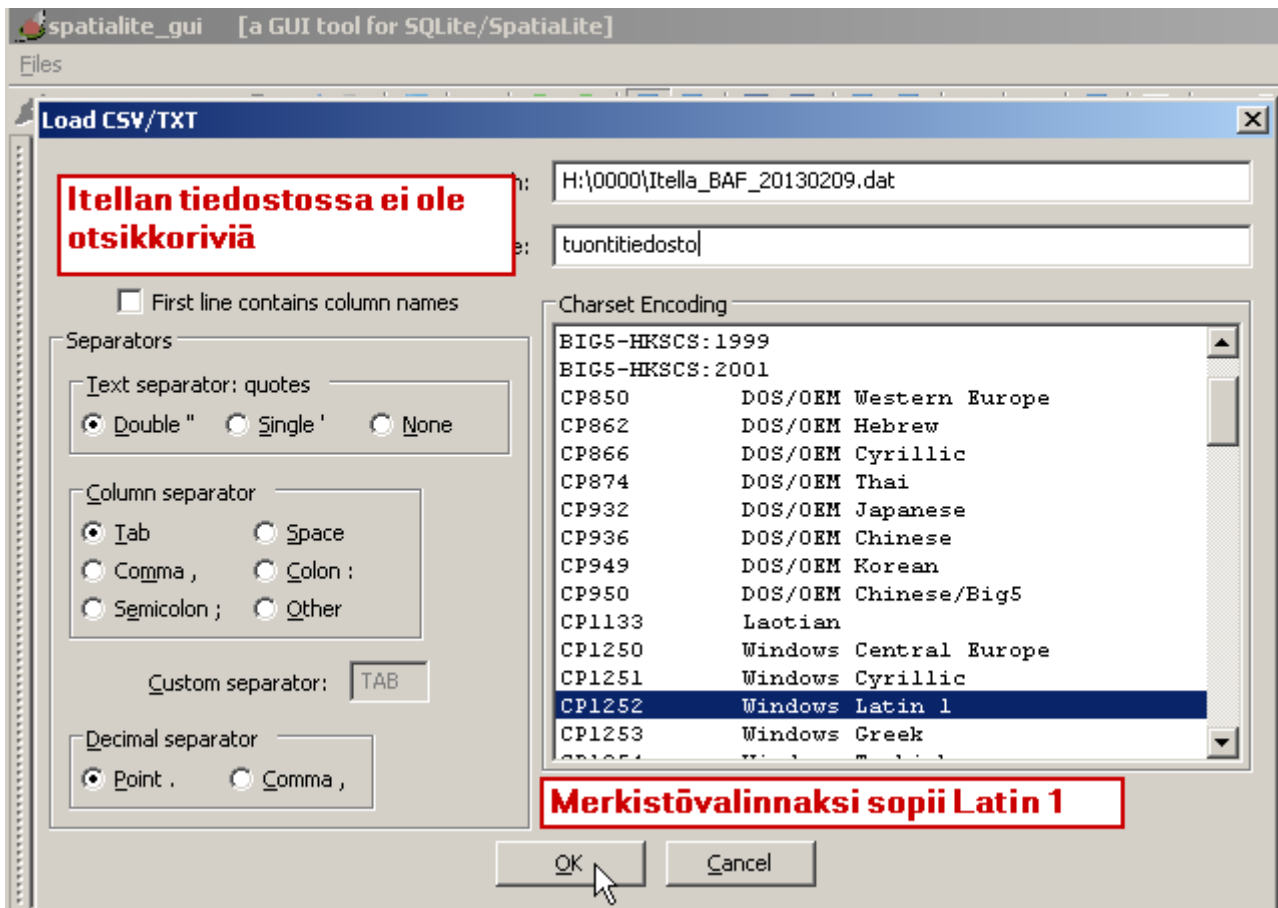
- Ladataan perusosoitteisto ja puretaan zip-arkisto
- Hankitaan Spatialite-gui -ohjelma sivustolta <http://www.gaia-gis.it/gaia-sins/>
- Luodaan tyhjä tietokanta ja luetaan Itellan tekstitiedosto tietokantaan



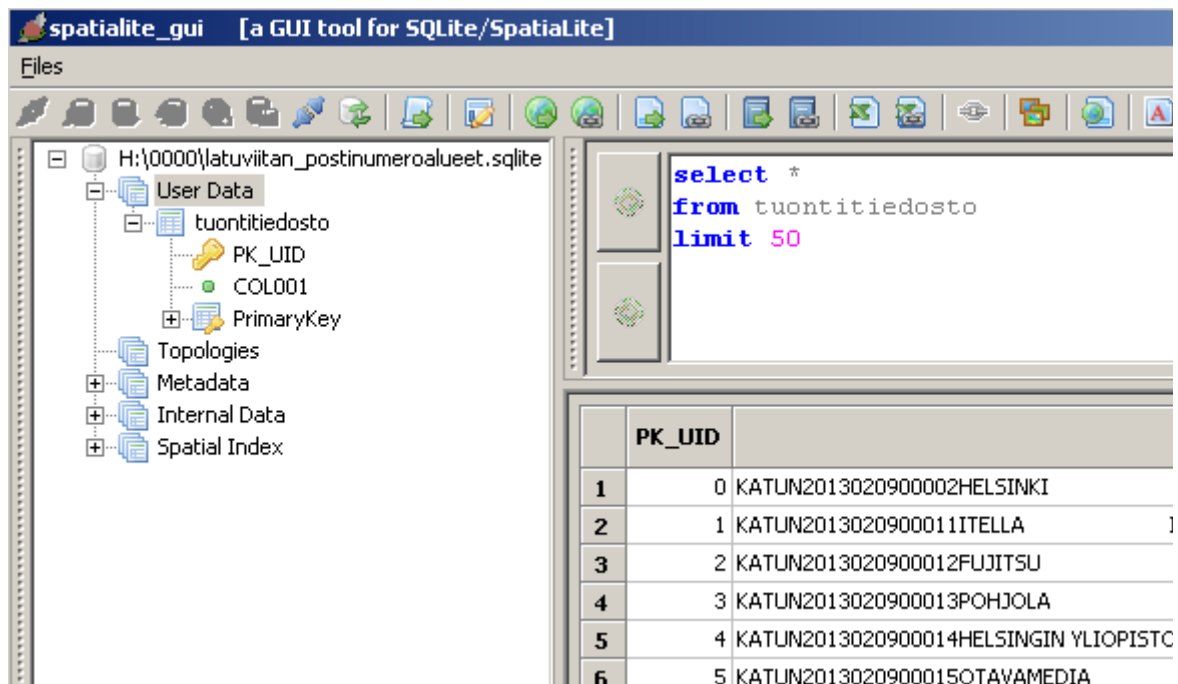
Käytetään Lataa CSC/TXT -työkalua



Itellan tiedosto, jonka tarkentimena on .dat, on houkuteltava ensin näkyviin, että sen voi valita.



Laitetaan asetukset kohdalleen ja tuodaan tiedosto tietokantaan.



Vilkaistaan, vaikuttaako tulos järkevältä. Sen jälkeen luetaan tarkasti tekstitiedoston rakenteen kuvaus Itellan ohjeesta, ja tehdään SQL-lauseet, joilla luodaan ensiksi rakenteen mukainen tietokantataulu ja sen jälkeen pilkotaan yhtenä pötkönä olevasta tekstistä oikeat pätkät niille kuuluviin kenttiin.

```
CREATE TABLE `itella` (
  `IDX` int(11) DEFAULT NULL,
  `POSTINRO` varchar(5) NOT NULL,
  `NIMIFI` varchar(30) NOT NULL,
  `NIMISV` varchar(30) NOT NULL,
  `NIMILYHFI` varchar(12) DEFAULT NULL,
  `NIMILYHSV` varchar(12) DEFAULT NULL,
  `KATUFI` varchar(30) DEFAULT NULL,
  `KATUSV` varchar(30) DEFAULT NULL,
  `TYHJA1` varchar(12) DEFAULT NULL,
  `TYHJA2` varchar(12) DEFAULT NULL,
  `KIINTIETOTYYPKDI` int(1) DEFAULT NULL,
  `PIENKIINTEISTONUMERO1` varchar(5) DEFAULT NULL,
  `PIENKIINTEISTOJAKOKIRJ1` char(1) DEFAULT NULL,
  `PIENVALIMERKKI` char(1) DEFAULT NULL,
  `PIENKIINTEISTONUMERO2` varchar(5) DEFAULT NULL,
  `PIENKIINTEISTOJAKOKIRJ2` char(1) DEFAULT NULL,
  `SUURKIINTEISTONUMERO1` varchar(5) DEFAULT NULL,
  `SUURKIINTEISTOJAKOKIRJ1` char(1) DEFAULT NULL,
  `SUURVALIMERKKI` char(1) DEFAULT NULL,
  `SUURKIINTEISTONUMERO2` varchar(5) DEFAULT NULL,
  `SUURKIINTEISTOJAKOKIRJ2` char(1) DEFAULT NULL,
  `KUNTAKDI` varchar(3) DEFAULT NULL,
  `KUNTAFI` varchar(20) DEFAULT NULL,
  `KUNTASV` varchar(20) DEFAULT NULL
) ;

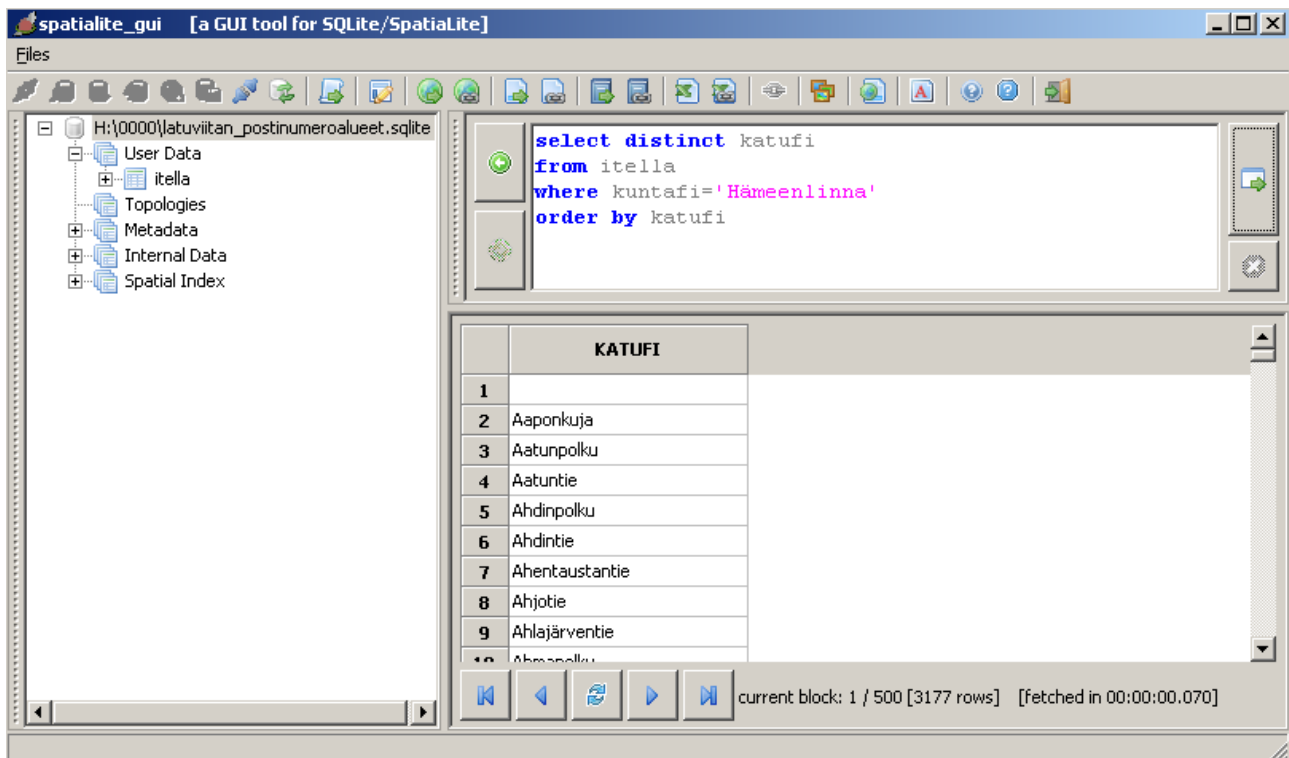
insert into itella
select
trim(substr(col001,6,8)),
trim(substr(col001,14,5)),
trim(substr(col001,19,30)),
```

```
trim(substr(col001,49,30)),
trim(substr(col001,79, 12)),
trim(substr(col001,91, 12)),
trim(substr(col001,103, 30)),
trim(substr(col001,133,30)),
trim(substr(col001,163,12)),
trim(substr(col001,175, 12)),
trim(substr(col001,187,1)),
trim(substr(col001,188,5)),
trim(substr(col001,193,1)),
trim(substr(col001,194,1)),
trim(substr(col001,195,5)),
trim(substr(col001,200,1)),
trim(substr(col001,201,5)),
trim(substr(col001,206,1)),
trim(substr(col001,207,1)),
trim(substr(col001,208,5)),
trim(substr(col001,213,1)),
trim(substr(col001,214,3)),
trim(substr(col001,217,20)),
trim(substr(col001,237,20))
from tuontitiedosto;
```

```
CREATE INDEX itella_postinro_idx
ON "itella" (postinro);
CREATE INDEX itella_nimifi_idx
ON "itella" (nimifi);
CREATE INDEX itella_nimisv_idx
ON "itella" (nimisv);
CREATE INDEX itella_katufi_idx
ON "itella" (katufi);
CREATE INDEX itella_katusv_idx
ON "itella" (katusv);
CREATE INDEX itella_kuntakdi_idx
ON "itella" (kuntakdi);
CREATE INDEX itella_kuntafi_idx
ON "itella" (kuntafi);
CREATE INDEX itella_kuntasv_idx
ON "itella" (kuntasv);
```

Väli aikaista taulua ”tuontitiedosto” ei enää tämän jälkeen tarvita ja se voidaan poistaa,

```
DROP table tuontitiedosto;
```



Tehdään pikatesti ja luodaan aakkosjärjestyksessä oleva lista taulussa esiintyvistä Hämeenlinnan kaduista. Haku kestää 7 sadasosasekuntia ja löytää 3177 riviä yhteensä 396866 rivistä. Tämä vaikuttaa hyvältä.

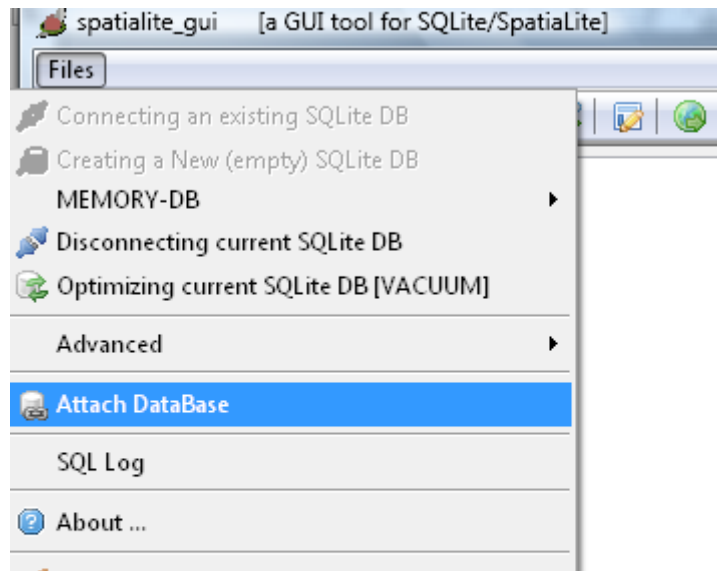
Maastotietokannan osoitteiden ja postiosoitetietokannan yhdistäminen

Vaihtoehto 1: Convex Hull

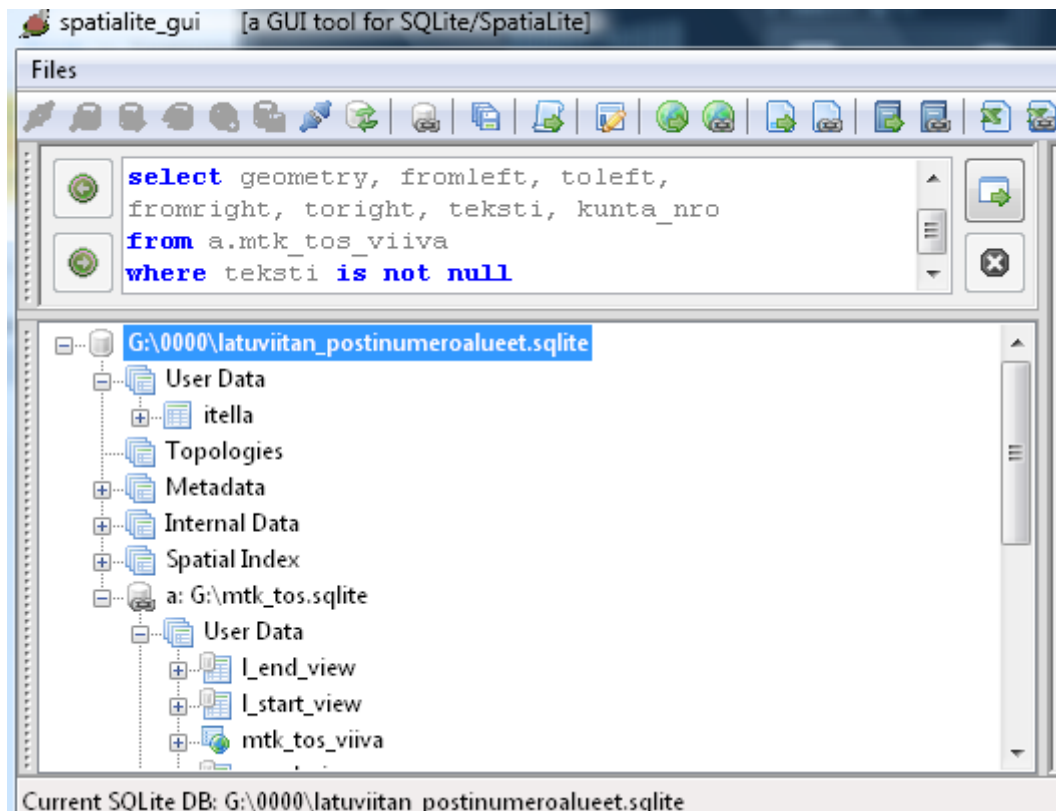
Maastotietokannan osoitteiden haltuunottoa käsitellään ohjeessa

http://latuviitta.org/documents/Maastotietokannan_osoitteiden_haltuunotto.pdf

Ohjeen vaiheessa 1. tuodaan tievektorit Spatialite-tietokantaan tauluun "mtk_tos_viivat". Kun tällainen tietokanta ja taulu on olemassa, niin tiestoviivoja voidaan lukea helposti postiosoitetietokannasta liittämällä tietokannat yhteen "attach database" toiminnolla. Tämä on mukana Spatialite-gui ohjelman versiossa 1.6.0, mutta ei sitä vanhemmissa.



Tästä löytyy ”Attach DataBase” -toiminto



Kun liitos on tehty, niin liitetty tietokanta näkyy luettelossa nimellä a: [Spatialite-tiedoston polku]. Liitetyn tietokannan tauluihin viitataan kyselyissä käyttämällä taulun nimen edessä sille liitoksessa syntynyttä tunnusta, joka tässä tapauksessa on siis a. Liitetyn taulun maastotietokannan tiet ovat siten ”a.mtk_tos_viiva”.

Vaikka liitettyjen tietokantojen tauluja onkin mahdollista käyttää kuten ne olisivat emotietokannan ikiomia tauluja, niin se ei kuitenkaan ole yhtä tehokasta. Siitä syystä käytämme tässä ohjeessa liitosta vain helpottamaan tietojen tuomista postinumerotietokantaan. Tuominen tehdään seuraavalla SQL-lauseella

```
create table mtk_tos_viiva as
select * from a.mtk_tos_viiva;
```

Liitettyä tietokantaa ei enää tarvita, joten voimme poistaa kytkennän

```
detach database a;
```

Tuonnin jälkeen uudessa tiestötaulussa ei ole yhtäkään indeksiä. Jatkossa tarvitaan ainakin paria indeksiä, joten luodaan ne.

```
CREATE INDEX mtk_tos_viiva_teksti_idx
ON "mtk_tos_viiva"
(teksti);

CREATE INDEX mtk_tos_viiva_kunta_nro_idx
ON "mtk_tos_viiva"
(kunta_nro);
```

Indeksien luominen kestää ehkä pari minuuttia indeksiä kohti.

Maastotietokannan tiestön osoitenumerot on liitetty teihin sellaisella tavalla, että niiden yhdistäminen postiosoitetietokannan osoitenumeroitten kanssa ei onnistu pienellä vaivalla. Tyydytään siis tässä harjoituksessa yhdistämään maastotietokannan teihin osoitetietokannan postinumerot käyttämällä yhdistämiseen pelkästään kuntakoodia ja tien nimeä.

Aloitetaan valmistelut tekemällä postiosoitteista lyhennetty taulu, jossa jokainen yksittäinen postinnumero-katunimi-kunta -yhdistelmä esiintyy vain kerran. Yhdistämiseen käytettäville kentille tehdään saman tien indeksit.

```
create table numerotie as
select distinct postinro, nimifi, katufi, kuntakdi
from itella
where katufi is not null;

CREATE INDEX postitie_katufi_idx
ON "numerotie"
(katufi);

CREATE INDEX postitie_kuntakdi_idx
ON "numerotie"
(kuntakdi);
```

Tarkistetaan, montako riviä jäi jäljelle.

```
select count(*) from numerotie
255487
```

Liitetään uudesta “numerotie” -taulusta tietoja “mtk_tos_viiva” -tauluun ja luodaan tuloksesta taas uusi taulu, “postitie”, tehdään pari indeksiä ja katsotaan, montako riviä tällä kertaa syntyi.


```

create table postitie as
SELECT "a"."GEOMETRY" AS "GEOMETRY", "a"."fromleft" AS "fromleft",
      "a"."toleft" AS "toleft", "a"."fromright" AS "fromright",
      "a"."toright" AS "toright", "a"."teksti" AS "teksti",
      "a"."kunta_nro" AS "kunta_nro", "b"."POSTINRO" AS "POSTINRO",
      "b"."NIMIFI" AS "NIMIFI"
FROM "mtk_tos_viiva" AS "a"
JOIN "numerotie" AS "b" ON ("a"."teksti" = "b"."KATUFI"
      AND "a"."kunta_nro" = "b"."KUNTAKDI")
      AND "a"."teksti" is not null

```

```

CREATE INDEX postitie_kunta_nro_idx
ON "postitie"
(kunta_nro);

```

```

CREATE INDEX postitie_postinro_idx
ON "postitie"
(postinro);

```

```

select count(*) from postitie
1506237

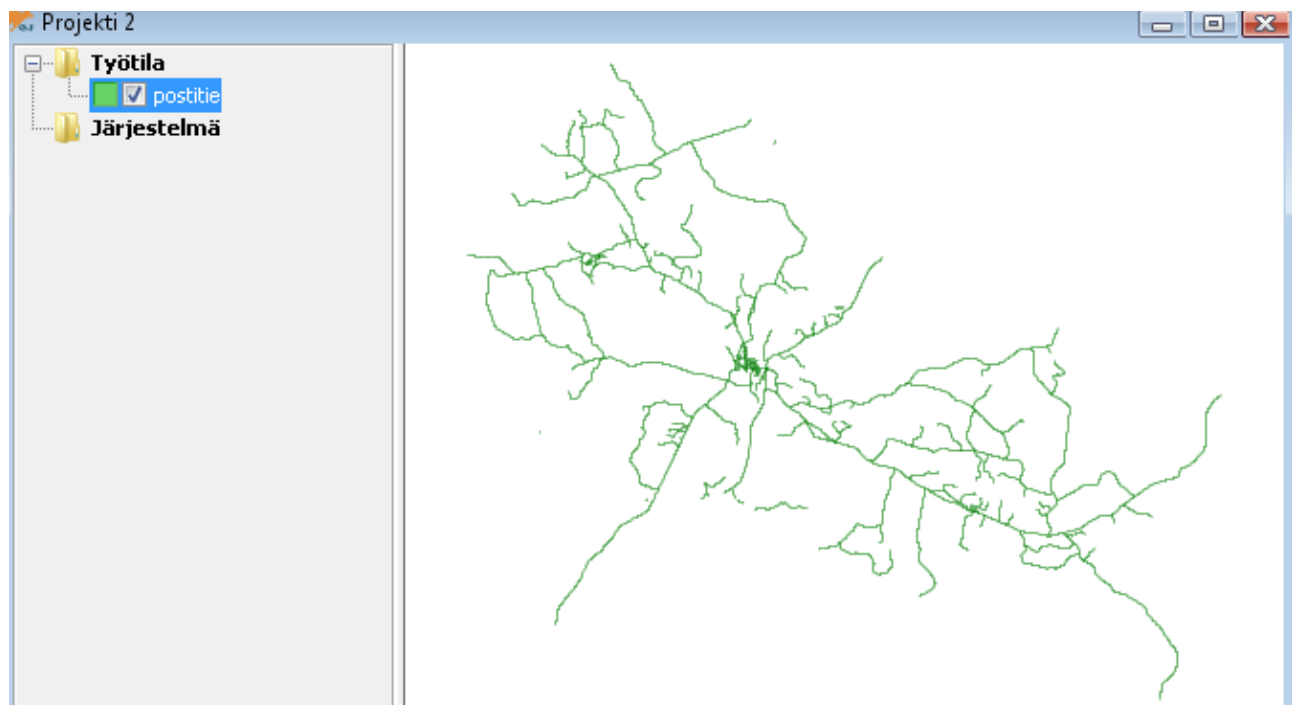
```

Nyt olla jo niin pitkällä, että jotain käyttökelpoisia tuloksia pitäisi olla näkyvillä. Testataan tulosta esimerkiksi OpenJUMP:in avulla (tätä varten on asennettava ensin laajennusosa ”DB Query Plugin”).

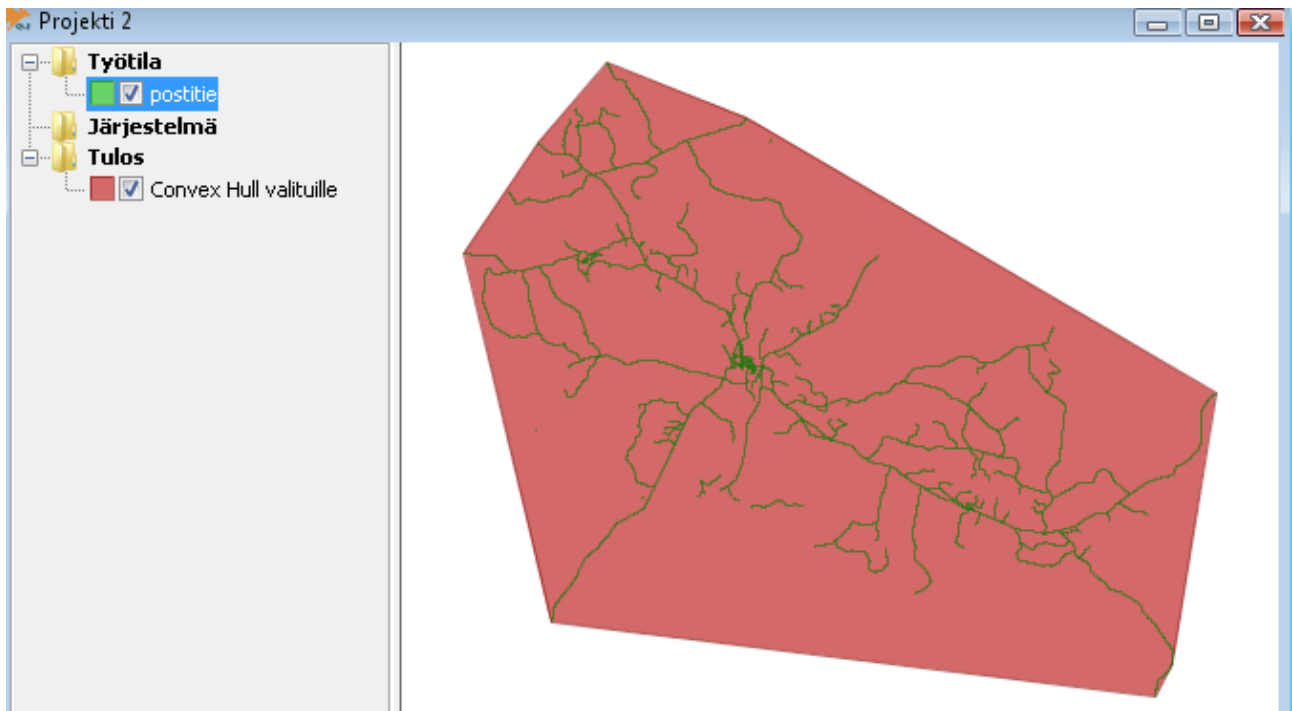
Kysely:

```
select geometry from postitie where postinro='89400';
```

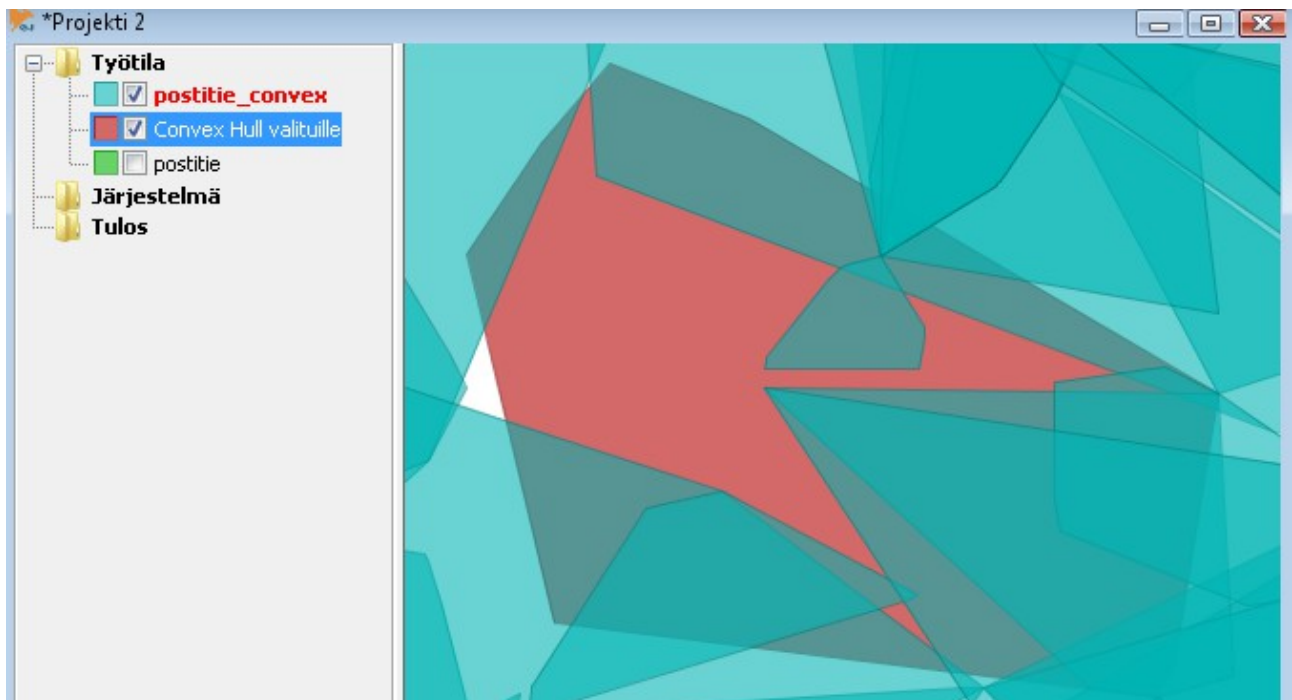
Tulos:



Tuloksessa näyttää olevan järkeä ja 89400-postinumeron saaneet tiet ovat jokseenkin yhtenäisellä ja rajatulla alueella.



Tiespajetista voidaan tehdä jonkinlainen postinumeroalue esimerkiksi Convex Hull -operaatiolla. Tulos näyttää tässä vaiheessa edelleen jokseenkin järkevältä.

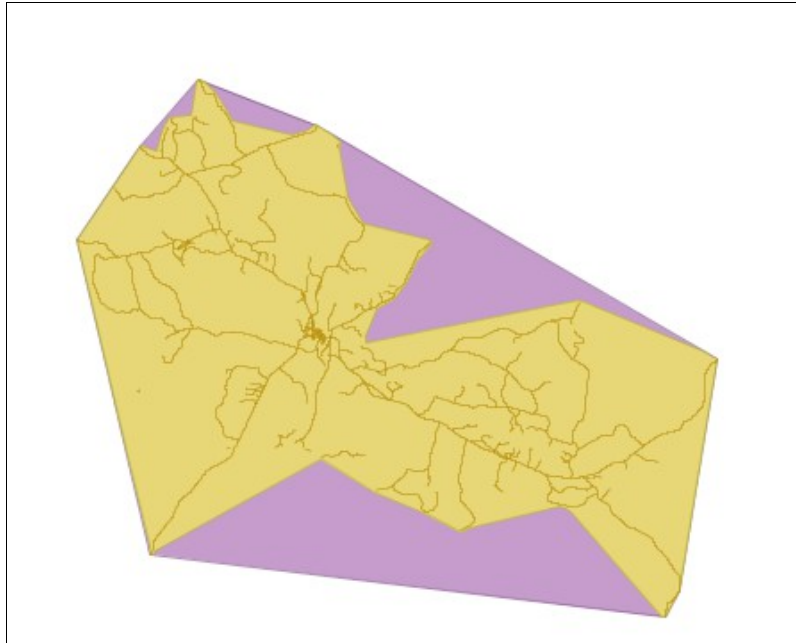


Tulos ei kuitenkaan näytä enää yhtä käyttökelpoiselta, jos lasketaan samalla menetelmällä postinumeroalueet kaikille postinumeroille. Näyttää siltä, että menetelmää on kehitettävä tai tulosta on parannettava käsityönä, jotta postinumeroalueet saataisiin sellaisiksi, etteivät ne mene päällekkäin eikä niiden välille jäisi aukkoja.

Vaihtoehto 2: Concave Hull

Concave Hull luo myös valitut kohteet sisäpuolelleen sulkevan alueen kuten convex hull -operaatiokin. Erona on se, että concave hull myötäilee kohteita tarkemmin. Convex hull:in lopputulos on kuin kohteiden ympärille olisi venytetty kuminauha, mutta concave hull:in lopputulos näyttää siltä, kuin kuminauhan sijasta olisi käytetty kutistemuovia.

PostGIS versiossa 2.0 on mukana concave hull -toiminto ja alla olevan kuvan keltainen alue näyttää kuinka se myötäilee tietyn postinumeroalueen tiestöä tarkemmin kuin violetilla esitetty convex hull.



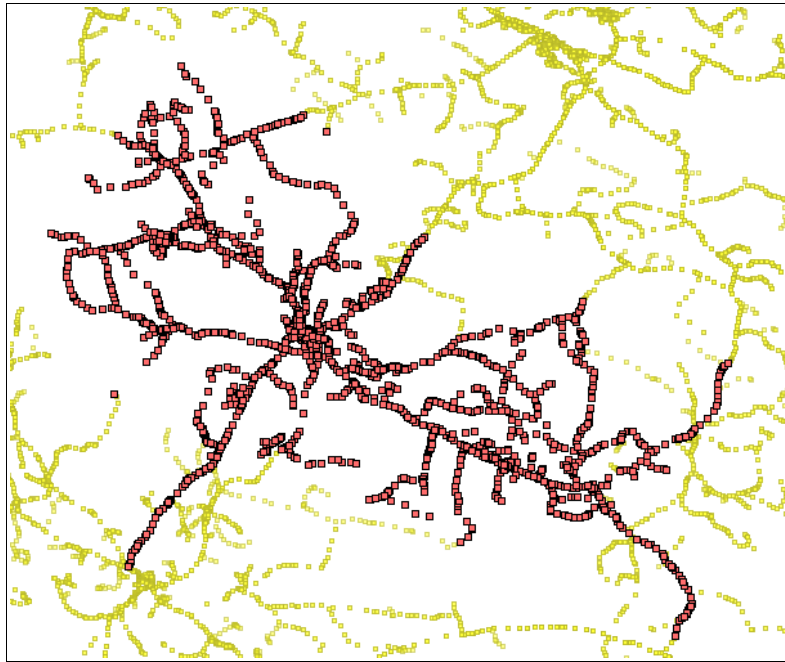
Tämä kuva puolestaan näyttää saman alueen muutkin postinumeroalueet.



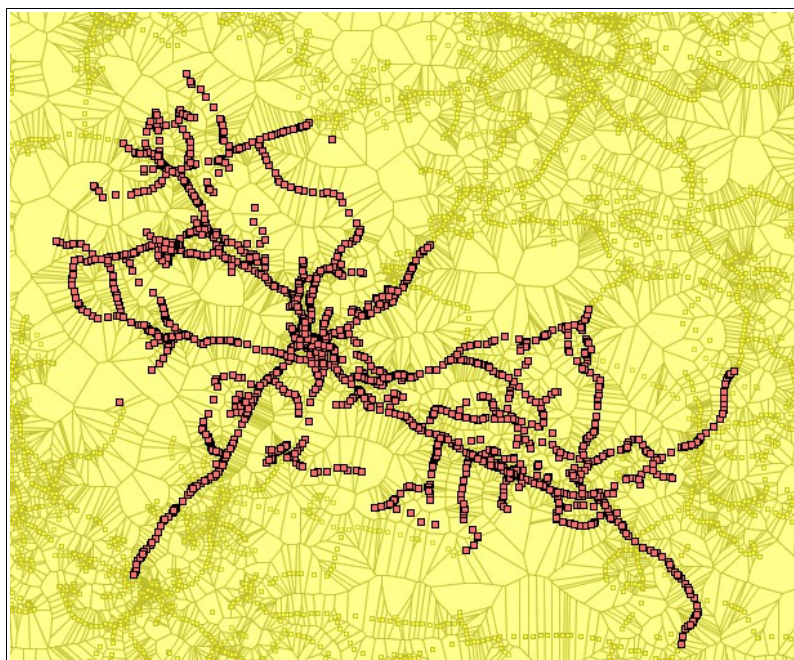
Johtopäätös Concave hull menetelmän käytöstä on, että se tuottaa vähemmän päällekkäisyyksiä ja enemmän aukkoja. Tulos saattaa sopia esittämään, missä tietty postinumeroalue on, mutta ei siihen, että etsittäisiin koordinaattien perusteella mihin postinumeroalueeseen tietty paikka kuuluu.

Vaihtoehto 2: Voronoi-diagrammin käyttö

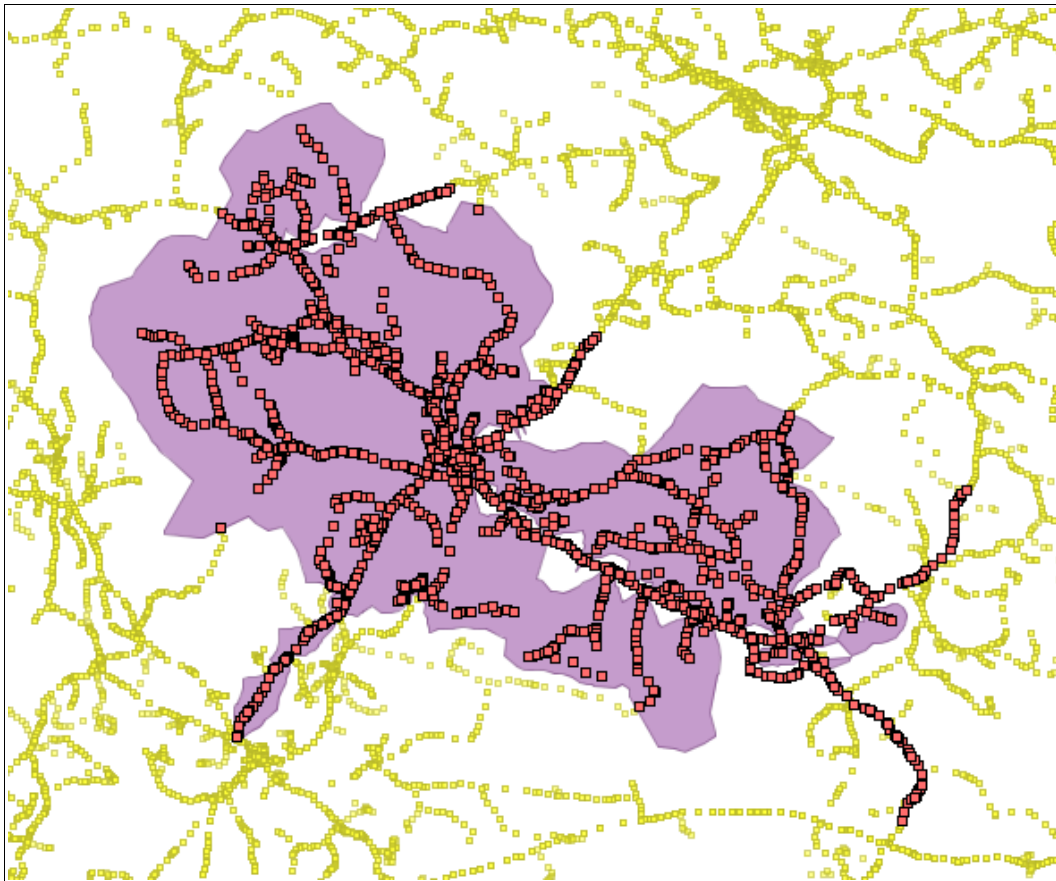
Kolmannessa vaihtoehdossa muodostetaan alueet aivan eri tavalla käyttämällä lähtöaineistona osoitepisteitä. Alla oleva kuva esittää osoitepisteitä, jotka luotu ronskilla menetelmällä osoitteellisista teistä, eli ottamalla tien keskipiste ST_centroid-funktiota käyttämällä ja siirtämällä näille pisteille tienpätkien ominaisuustiedoista postinumero. Oranssit pisteet esittävät postinumeroalueen 89400 kohteita.



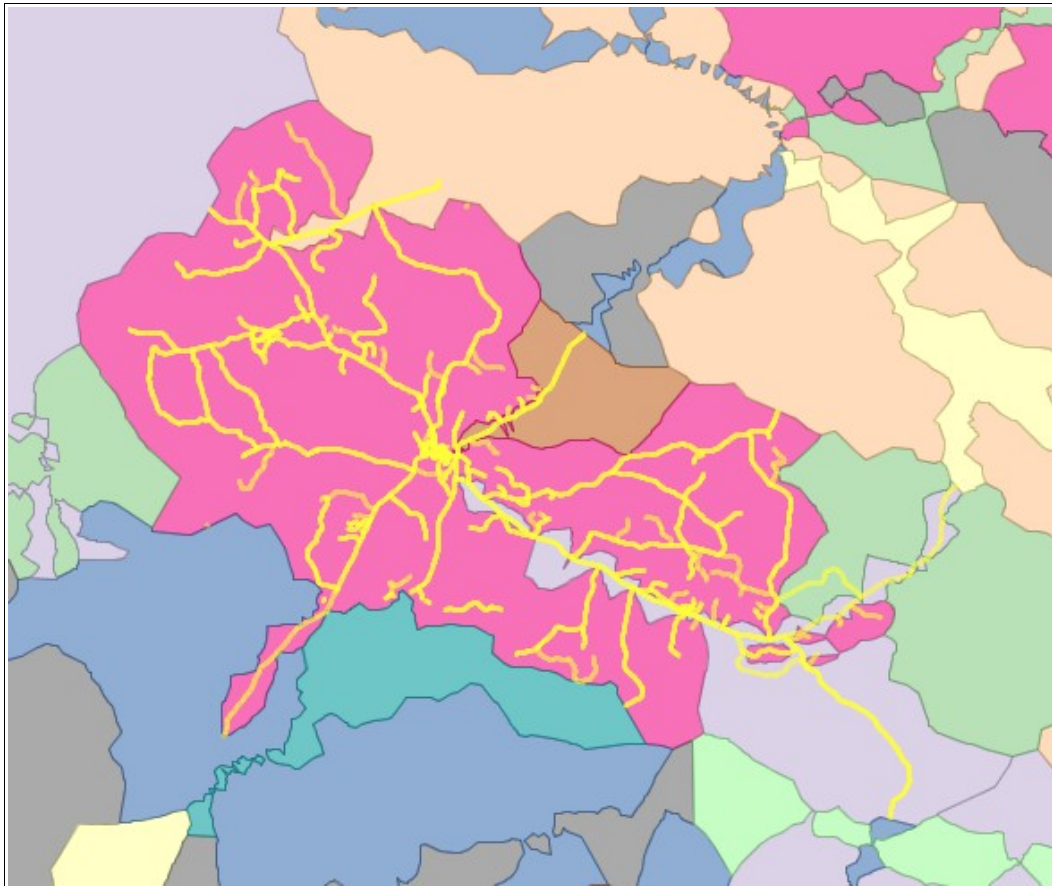
Voronoi-diagrammissa kartan alue jaetaan monikulmioihin niin, että koko alue peittyy, ja jokaisen monikulmion sisäpuolelle jää tasan yksi piste. Alla olevassa kuvassa jokainen monikulmio esittää ikään kuin sen sisällä olevan osoitepisteen vaikutusalueetta. Diagrammi on tehty OpenJUMP-ohjelmalla.



Postinumeroalue 89400 on nyt helppo tehdä sulauttamalla yhteen kaikki pikkumonikulmiot, joiden postinumeroiksi tuli analyysissä 89400. Tulos näyttää silmämääräisesti varsin järkevältä. Kuitenkin alueen itäreunalla on osoitepisteitä, jotka eivät päätyneet omaan postinumeroalueeseen, ja toisaalta alueen keskellä näyttää olevan aukkoja. Tämä johtuu siitä, että lähtöaineisto on muodostettu tällaista työtä varten liian roskilla menetelmällä. Pitkät tiet saattavat katketa keskeltä ja alkuosa voi kuulua eri postinumeroalueelle kuin loppuosa, mutta aineiston esikäsittelyssä tätä ei ole otettu huomioon, vaan nyt pitkät tiet kuuluvat muka koko pituudeltaan molempiin postinumeroalueisiin. Tämä tilanne sekoittaa monikulmiot muodostavan funktion, tai ainakin sen funktion osan, joka siirsi postinumerotiedot pisteille.



Viimeisessä kuvassa näkyy tämän alueen kaikki postinumeroalueet. Alueet eivät mene päällekkäin eikä niiden väliin jää aukkoja. Alueissa ovat todennäköisesti suunnilleen oikeilla paikoillaan. Virheistä on ilmeisesti siinä, mihin yksi alue oikeasti loppuu ja mistä toinen alkaa, mutta tilanteen korjaamiseksi pitää vain siirtää alueiden välistä yhteistä rajaviivaa, ja jos tuo työ tuntuisi oikeasti tärkeältä, niin sen kyllä pystyisi tekemään ihan käsityönäkin.



Kiitokset

Ajatuksen siitä, että Itellan tekstimuotoisesta postinumeroaineistosta voisi ehkä tehdä jotain kivaa esitti Petri Suvila. Ajatus Voronoi-diagrammin käytöstä tuli Pekka Sarkolalta.

Lopputulokset shapefile-muodossa

Convex hull -menetelmän lopputulos shapefile-muodossa:

http://latuviitta.org/documents/postinumeroalueet_2013_v01.zip

Voronoi-menetelmän lopputulos shapefile-muodossa:

http://latuviitta.org/documents/postinumeroalueet_2013_v02.zip