

Paikkatietoaineistojen muuntaminen GPS-kohdepisteiksi GDAL/OGR -ohjelmilla

Mallisuoritus käyttämällä lähtöaineistona OpenStreetMap:in pbf- muotoista tiedostoa sekä WFS-palvelua

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

Viimeksi muutettu 5. elokuuta 2012

1. Aineistot ja ohjelmistot

Lähtöaineistot:

- Suomen OpenStreetMap-aineisto, joka on ladattu pbf-muodossa Geofabrik.de-latauspalvelusta <http://download.geofabrik.de/osm/europe/>
- Helsingin pysäköintilippuautomaattien sijainnit. Alkuperäinen aineisto ladattu sivulta <http://www.hri.fi/fi/data/helsingin-pysakointilippuautomaatit/>, mutta tässä esimerkissä käytetään WFS-palvelua <http://hip.latuviitta.org>

Ohjelmistot:

- GDAL, vähintään versio r24696. Tällä hetkellä kehitysversio, josta tulee aikanaan versio 2.0.
- Garmin POI Loader versio 2.7.1 <http://www8.garmin.com/products/poiloader/>

2. Menetelmä

Mallisuorituksessa lähtöaineistojen pisteet muunnetaan GPX-muotoon, joka on nimenomaan GPS-laitteiden väliseen tiedonsiirtoon suunniteltu tiedostomuoto. Koska muunnoksessa syntyvä GPX-tiedosto ei näytä millään tavalla mielenkiintoiseksi, niin mallisuorituksessa GPS-kohdepisteet viedään myös perille asti retki-GPS-laitteeseen. Todisteena onnistuneesta suorituksesta käytetään laitteen näytöstä otettuja valokuvia.

GPX-tiedostomuodosta on saatavilla tietoa esimerkiksi sivuilta

<http://www.topografix.com/gpx.as>

<http://www.topografix.com/GPX/1/1/gpx.xsd>

Jälkimmäisen linkin ei oleteta suuremmin kiinnostavan tämän ohjeen lukijoita, mutta se on mukana siitä syystä, ettei myöhemmin niin paljon ihmetytä, miksi eräiden tietokenttien nimiä muutetaan muuksi kuin mitä ne ovat lähtöaineistossa. Se johtuu siitä, että tietomalli määrää, että kohteen nimi on kentässä "name" ja sen kuvaus kentässä "desc". Jos lähtöaineistossa käytetään esimerkiksi kenttien nimiä "nimi" ja "kuvaus", niin se eivät kelpaa sellaisenaan GPX:ään viettäviksi.

3. Perussuoritus

A) WFS-palvelusta GPS-kohdepisteiksi

WFS-palvelu on tarkoitettu vektorimuotoisten paikkatietojen välittämiseen verkon kautta. Suomen paras avoin WFS-palvelu lienee osoitteessa <http://hip.latuviitta.org>, mikä on säälistävää, koska palvelua ylläpidetään niin laiskasti, ettei sitä millään kehtaa erityisen hyväksi mainostaa.

Mallisuorituksessa käytetään WFS-palvelua GDAL-ohjelmien avulla. Tästä aiheesta on olemassa ohje http://latuviitta.org/documents/GDAL_ja_WFS.pdf ja sen lukemista suositellaan, koska tässä käsillä olevassa ohjeessa ei suuremmin taustatietoja kerrota.

Aloitetaan keskustelu WFS-palvelun kanssa selvittämällä, mitä aineistoja palvelusta on saatavilla.

```
ogrinfo WFS:http://hip.latuviitta.org/cgi-bin/tinyows

INFO: Open of `WFS:http://hip.latuviitta.org/cgi-bin/tinyows'
      using driver `WFS' successful.
1: tows:osm_point (Point)
2: tows:osm_line (Line String)
3: tows:osm_polygon
....
67: lv:hki_parkkilippuautomaatit (Point)
```

Kiinnostuksen kohteena on tällä kertaa Helsingin parkkilippuautomaattien tiedot sisältävä aineisto, joka halutaan viedä GPS-laitteeseen. Tämän karttatason nimi WFS-palvelussa on ”lv:hki_parkkilippuautomaatit”.

Seuraavaksi kysellään lisätietoja tuosta nimenomaisesta tasosta **järkevällä ja palvelinystävällisellä tavalla**. Järkevyys saadua aikaan käyttämällä ylimääräistä parametria ”MAXFEATURES=2”. Tällä asetuksella GDAL hakee palvelusta vain kaksi ensimmäistä kohdetta, missä on aivan riittävästi aineistoon tutustumiseksi. Joillakin palvelussa olevialla karttatasoilla on satoja tuhansia kohteita, ja niiden kaikkien hakeminen pelkkää silmäilyä varten ei varmasti ole järkevää. Käytetään siis aina MAXFEATURES-parametria.

```
ogrinfo "WFS:http://hip.latuviitta.org/cgi-bin/tinyows?MAXFEATURES=2" lv:hki_parkkilippuautomaatit

INFO: Open of `WFS:http://hip.latuviitta.org/cgi-bin/tinyows?MAXFEATURES=2'
      using driver `WFS' successful.

Layer name: lv:hki_parkkilippuautomaatit
Geometry: Point
Feature Count: 2
Extent: (384783.062500, 6672911.500000) - (385116.562500, 6675190.000000)
Layer SRS WKT:
PROJCS["ETRS89 / TM35FIN(E,N)",
  GEOGCS["ETRS89",
    DATUM["European_Terrestrial_Reference_System_1989",
      SPHEROID["GRS_1980",6378137,298.257222101,
        AUTHORITY["EPSG","7019"]],
      TOWGS84[0,0,0,0,0,0,0],
      AUTHORITY["EPSG","6258"]],
    PRIMEM["Greenwich",0,
      AUTHORITY["EPSG","8901"]],
```

```

UNIT["degree",0.0174532925199433,
    AUTHORITY["EPSG","9122"]],
AXIS["Latitude",NORTH],
AXIS["Longitude",EAST],
AUTHORITY["EPSG","4258"]],
PROJECTION["Transverse_Mercator"],
PARAMETER["latitude_of_origin",0],
PARAMETER["central_meridian",27],
PARAMETER["scale_factor",0.9996],
PARAMETER["false_easting",500000],
PARAMETER["false_northing",0],
UNIT["metre",1,
    AUTHORITY["EPSG","9001"]],
AXIS["Easting",EAST],
AXIS["Northing",NORTH],
AUTHORITY["EPSG","3067"]]
Geometry Column = wkb_geometry
gml_id: String (0.0)
tunniste: Integer (0.0)
osoite: String (0.0)
vyohyke: Integer (0.0)
maksullisuus aika: String (0.0)
max_parkkiaika: String (0.0)
muu maksutapa: String (0.0)
OGRFeature(lv:hki_parkkilippuautomaatit):1
  gml_id (String) = hki_parkkilippuautomaatit.1
  tunniste (Integer) = 2
  osoite (String) = ESTERINPORTTI
  vyohyke (Integer) = 2
  maksullisuus aika (String) = ma-pe 9-19
  max_parkkiaika (String) = 4 h
  muu maksutapa (String) = (null)
  POINT (385116.55 6675189.89)
OGRFeature(lv:hki_parkkilippuautomaatit):2
  gml_id (String) = hki_parkkilippuautomaatit.2
  tunniste (Integer) = 3
  osoite (String) = ET.HESPERIANKATU 22 VP 1
  vyohyke (Integer) = 2
  maksullisuus aika (String) = ma-pe 9-19
  max_parkkiaika (String) = X h
  muu maksutapa (String) = (null)
  POINT (384783.08 6672911.55)

```

Listauksesta on korostettu pari oleellista kohtaa. Oransseilta riveiltä selviää, että aineiston oletuskoordinaattijärjestelmä on ETRS-TM35FIN (EPSG:3067), ja turkooseilla riveillä listataan aineistoon sisältyvien tietokenttien nimet ja tietotyypit. Kahden ensimmäisen parkkilippuautomaatin listauksesta nähdään mukavasti, minkä näköisiä tietoja on luvassa.

Listauksen perusteella päätetään ottaa tiedot GPX-tiedoston ”name”-kenttään WFS-palvelun ominaisuudesta ”osoite” ja ”desc”-kentän tiedot ominaisuudesta ”maksullisuus aika”. Huomioidaan, että WFS-palvelun oletuskoordinaattijärjestelmä on EPSG:3067, kun taas GPX-tiedostoissa täytyy käyttää WGS84-koordinaatteja eli järjestelmää EPSG:4326. MAXFEATURES-arvo pidetään pyynnössä mukana, koska sillä tavalla voidaan estää mielettömän suurien vastausten saaminen, mutta annetaan sille kuitenkin sen verran suuri arvo, että kaikkien pysäköintilippuautomaattien tiedot todennäköisesti saadaan. Tuskin niitä Helsingissä kymmentätuhattakaan on, joten käytetään MAXFEATURES=10000.

Nyt voidaankin jo rakentaa komento tietojen hakemiseksi ja muuntamiseksi ogr2ogr-ohjelmalla.

```
ogr2ogr -f GPX parkkiautomaatit.gpx -t_srs epsg:4326  
"WFS:http://hip.latuviitta.org/cgi-bin/tinyows?MAXFEATURES=10000" -sql "select  
osoite as 'name',maksullisuusaika as 'desc' from hki_parkkilippuautomaatit"
```

SQL-kysely "select osoite as 'name'" siis valitsee osoite-nimisestä kentästä mutta tallentaa tuloksen name-nimiseen kenttään. Kenttä "maksullisuusaika" tallennetaan nimellä "desc". Tässä kohtaa tuleekin vastaan pieni mutta tärkeä yksityiskohta, joka on vain pakko tietää. Tuloksentän nimeksi haluttava "desc" on nimittäin pakko antaa yksinkertaisten lainausmerkkien välissä, koska desc:llä on SQL:ssä erikoismerkitys "descending" eli tulosten järjestäminen käänteiseen aakkosjärjestykseen. Ilman lainausmerkkejä desc:in oletetaan tarkoittavan (väärin annettua) kommentia järjestää tulokset eikä sitä tulkita kentän nimeksi. Esimerkissä myös "name" on laitettu yksinkertaisten lainausmerkkien väliin, mutta koska name:lla ei ole erikoismerkitystä niin tämä merkkien väliin sulkeminen ei ole pakollista.

Komennon suorittamisen jälkeen levyllä on tiedosto "parkkiautomaatit.gpx". Jätetään se odottamaan ja siirrytään tekemään seuraavaa GPS-kohdepisteaineistoa.

B) OpenStreetMap-tiedostosta GPS-kohdepisteiksi

Heinäkuun loppupuolella 2012 julkaistiin GDAL:in uusi OSM-ajuri. Sen avulla OpenStreetMap-tiedostoja voidaan lukea suoraan. Ajuri tukee sekä OSM.xml- että OSM.pbf-tiedostomuotoja.

OSM-ajurin englanninkielinen ohjesivu: http://gdal.org/ogr/drv_osm.html

Suomenkielinen ohje ajurin käytöstä

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

Ennen kuin OSM-aineistosta alkaa tehdä GPS-reittipisteitä, on mietittävä muutama asia. Täytyy ensiksikin päättää, mitkä kohteet halutaan valita ja selvittää, miten ne löydetään OSM:in joskus sekavien merkintäsääntöjen avulla. Sen jälkeen on päätettävä, mitä tietoja halutaan viedä GPX-tiedoston "name"- ja "desc"-kenttiin.

Mallisuorituksessa valitaan aineistoksi huoltoasemat. OSM-aineistoon ne on merkitty avain-arvoparilla "amenity=fuel". Päätetään lisäksi, että GPX-tiedoston "name" nimi otetaan OSM:n avaimesta eli tagista "name". Kenttään "desc" viedään arvot OSM:in "addr:street"-arvoista. Päätetään vielä, että etsitään huoltoasemia vain OSM:in pistekohteista, vaikka arvellaankin, että niitä saattaisi löytyä myös aluekohteista. Hetken harkinnan jälkeen päätetään tallentaa kohteet nimellä "bensa-asemat". Luotetaan, että dieselin tankkaajatkin ymmärtävät mistä on kyse.

Kun nämä päätökset on tehty, niin GDAL:in "osmconf.ini"-ohjaustiedostoa on muokattava niin, että pistekohteille [points] otetaan käyttöön kaikki valintoihin tarvittavat ominaisuustiedot eli attributit eli avaimet eli tagit. Esimerkkitapauksessa listalta täytyy siis löytyä amenity, name ja addr:street. Lista saa olla pitempikin, ylimääräisistä avaimista ei ole haittaa.

```
# keys to report as OGR fields  
attributes=amenity,name,addr:street
```

GDAL:ille OpenStreetMap-tiedosto on täysin samanlainen tietolähde kuin edellisessä vaiheessa käytetty WFS-palvelu tai mikä tahansa muu tuettu lähde. Siten myös ogrinfo-työkalun käyttö on samanlaista kuin edellä.

```
ogrinfo finland.osm.pbf
```

```
Had to open data source read-only.  
INFO: Open of `finland.osm.pbf'  
      using driver `OSM' successful.  
1: points (Point)  
2: lines (Line String)  
3: polygons (Polygon)  
4: multilinestrings (Multi Line String)  
5: multipolygons (Multi Polygon)  
6: other_relations (Geometry Collection)
```

Seuraavaksi kysytään lisätietoja tasolta ”points”. WFS-esimerkissä saatiin sekä perustiedot että parin kohteen kaikki tiedot kätevästi yhdellä komennolla käyttämällä MAXFEATURES=2 -parametria. Muilla OGR-tietolähteillä tämä ei kuitenkaan toimi. Pelkät tason perustiedot saadaan helposti käyttämällä parametria -so (summary only). Jos halutaan vilkaista kohteille tallennettuja tietoja pikaisesti, niin voidaan antaa pelkkä ogrinfo-komento ja keskeyttää listaus Ctrc-C -näppäinyhdistelmällä. Jätetään se kuitenkin tässä esimerkissä tekemättä.

```
ogrinfo finland.osm.pbf points -so
```

```
Had to open data source read-only.  
INFO: Open of `finland.osm.pbf'  
      using driver `OSM' successful.  
  
Layer name: points  
Geometry: Point  
Feature Count: -1  
Extent: (19.024270, 59.287830) - (31.600890, 70.099590)  
Layer SRS WKT:  
GEOGCS["WGS 84",  
  DATUM["WGS_1984",  
    SPHEROID["WGS 84",6378137,298.257223563,  
      AUTHORITY["EPSG","7030"]],  
    TOWGS84[0,0,0,0,0,0,0],  
    AUTHORITY["EPSG","6326"]],  
  PRIMEM["Greenwich",0,  
    AUTHORITY["EPSG","8901"]],  
  UNIT["degree",0.0174532925199433,  
    AUTHORITY["EPSG","9108"]],  
  AUTHORITY["EPSG","4326"]]  
osm_id: String (0.0)  
name: String (0.0)  
amenity: String (0.0)  
addr_street: String (0.0)  
other_tags: String (0.0)
```

Kaikki tarvittava tieto on nyt koottu ja voidaan kirjoittaa komento GPX-tiedoston luomiseksi

```
ogr2ogr -f GPX bensa-asetat.gpx finland.osm.pbf -sql "select name,addr_street as  
'desc' from points where amenity='fuel'"
```

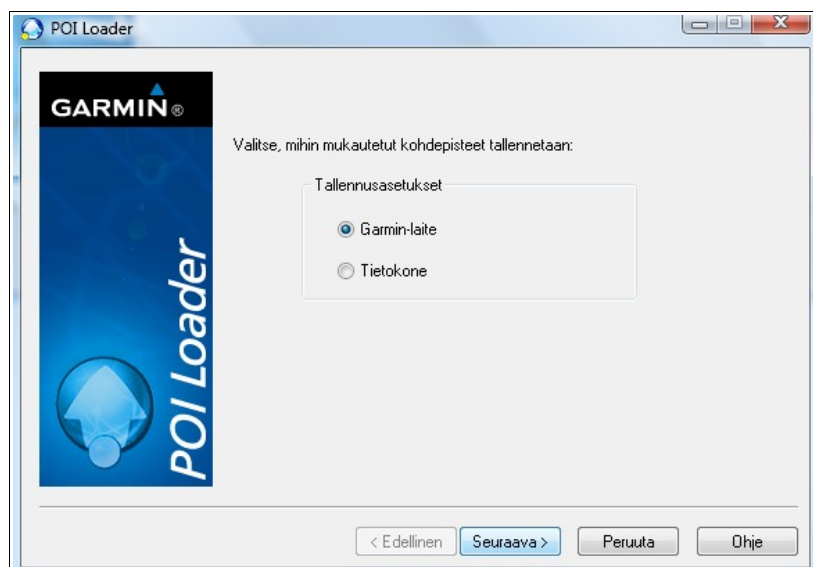
Tässäkin esimerkissä on mukana kikka, joka täytyy tietää. OSM-tiedoissa katuosoitteelle käytetään avainta addr:street. Kaksoispisteet ovat kuitenkin monissa tiedostomuodoissa ja erityisesti kaikissa XML:ään perustuvissa kiellettyjä merkkejä attribuuttien nimissä, joten GDAL muuntaa ne automaattisesti alaviivoiksi. Siksi siis ”select addr_street” vaikka alkuperäisessä OSM-aineistossa käytetäänkin nimeä ”addr:street”.

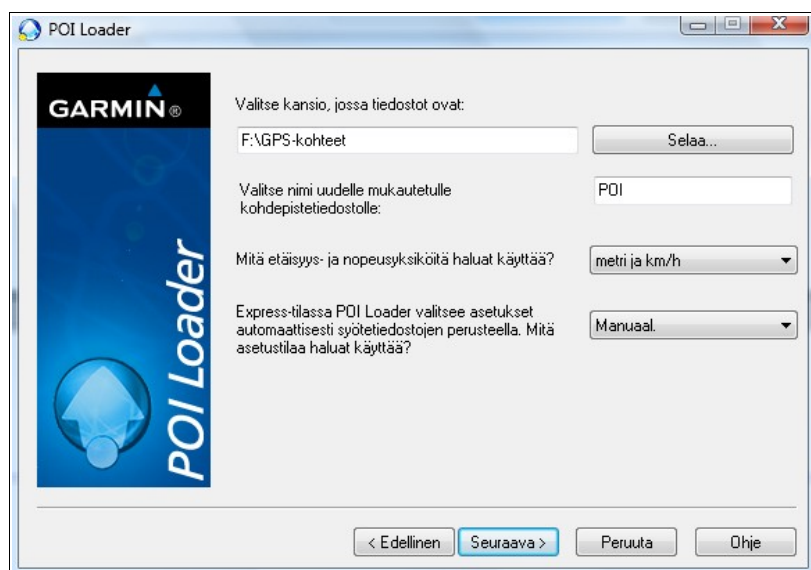
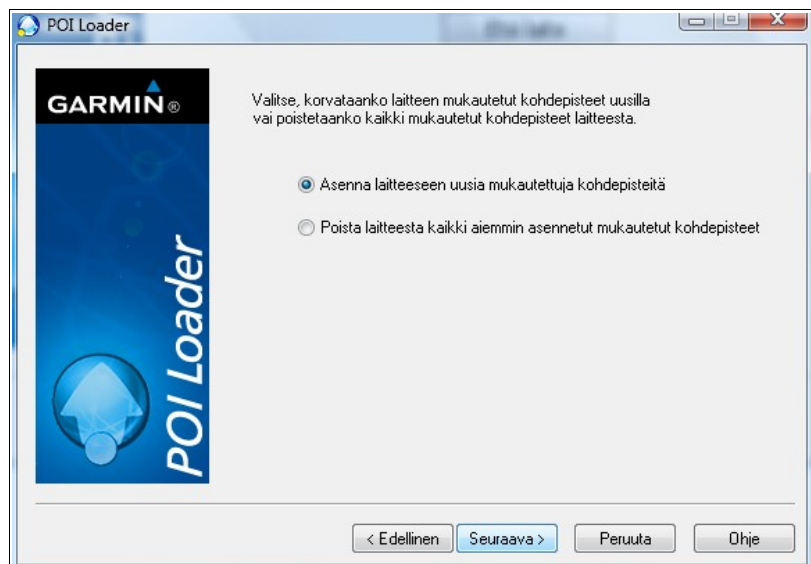
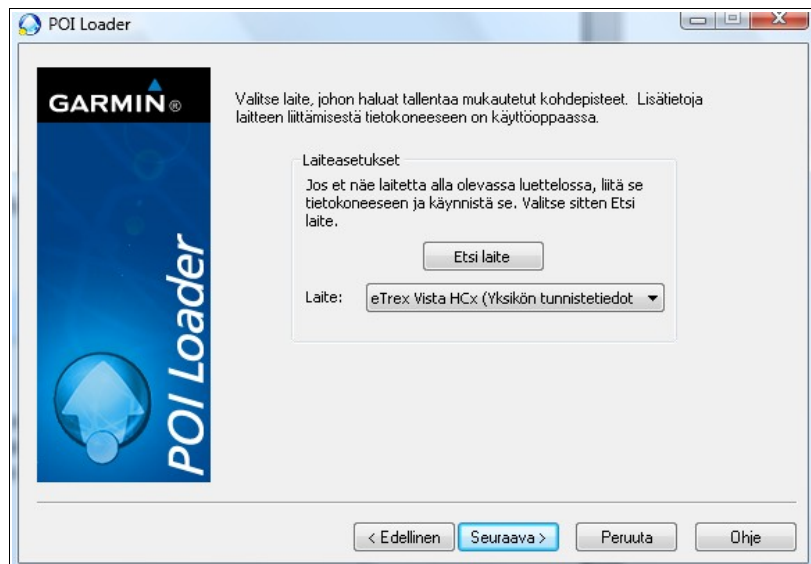
4. Kohdepisteiden vieminen GPX-tiedostosta GPS-laitteelle

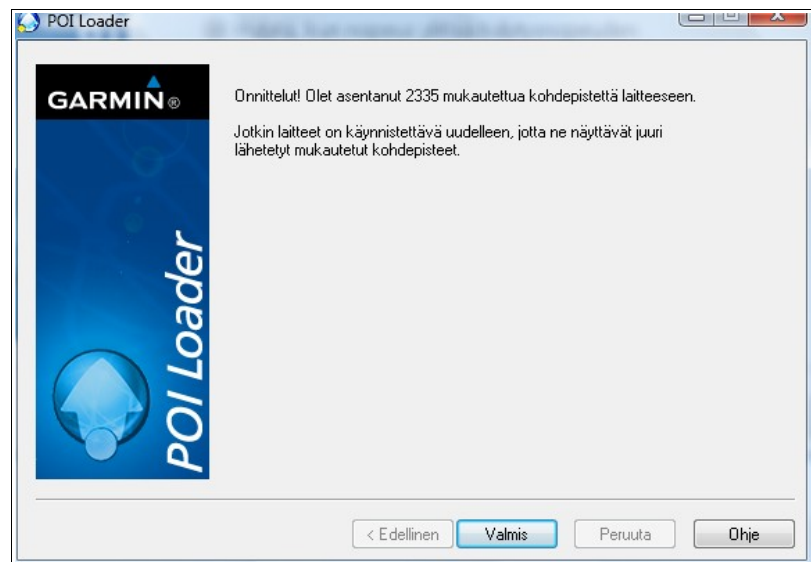
Seuraavaksi näytetään kädestä pitäen ja suurten kuvaruutukaappausten avulla, kuinka GDAL:lla luotuihin GPX-tiedostoihin kuuluvat kohdepisteet voidaan viedä POI Loader -ohjelmalla Garmin etrex Vista Hcx -GPS laitteeseen. Olettaisin homman tapahtuvan suurin piirtein samalla tavalla muunkin merkkisillä GPS-laitteilla.

POI Loader -ohjelmalle kerrotaan hakemisto, josta GPS-kohdepisteitä sisältäviä tiedostoja etsitään. Ohjelma tulee GPX-muodon lisäksi pilkuilla eroteltuja tekstitiedostoja. Jos samassa hakemistossa missä kohdepistetiedostot ovat on myös Windows bitmap -tyylinen kuvatiedosto (koko korkeintaan 24x24 pikseliä) ja sillä on sama nimen alkuosa (bensa-asemat.gpx ja bensa-asemat.bmp), niin GPS-laite näyttää uudet kohdepisteet tällä symbolilla.

Lisätietoja POI Loaderista löytyy netistä, tässä ohjeessa sitä käytetään vain osoittamaan, että edellä luotuja GPX-tietostoja on mahdollista käyttää johonkin hyödylliseen.







5. Edistyneitä suorituksia

A) Pitkä kommentti ominaisuustietoja yhdistelemällä

Edellisissä erimerkeissä poimittiin yksinkertaisesti yksi ominaisuustieto GPS-kohdepisteen nimeksi ja toinen ominaisuustieto kohteen kuvaukseksi. Kuvaus voi kuitenkin olla GPX-standardin mukaan vaikka kuinka pitkä. Käyttäjälle voisi esimerkiksi olla hyötyä nähdä osoitteessa sekä katuosoite että osoitenumero. OSM-bensa-asemaesimerkkiä voidaan muokata niin, että käytetään SQL:n concatenate-funktiota eli kenttien yhdistämistä.

```
ogr2ogr -f GPX bensa-asetat2.gpx finland.osm.pbf -sql "select  
name,CONCAT(addr_street,' ',addr_housenumber) as 'desc' from points where  
amenity='fuel'"
```

CONCAT(addr_street,' ',addr_housenumber) kirjoittaa yhteen pötköön arvon kentästä addr_street, yhden välilyönnin sisältävän merkkijonon ja arvon kentästä addr_housenumber.

B) Monirivinen kommentti ominaisuustietoja yhdistelemällä ja kikkailemalla

GPX-tiedoston kuvaus (desc) voi olla myös monirivinen. Ikävä kyllä ogr2ogr-ohjelman komentoriviltä on ainakin Windowsilla hyvin konstikasta ellei sitten mahdotonta antaa käskyä lisätä rivinvaihtomerkkejä. GPX-tiedostomuoto on kuitenkin pelkkää tekstiä, joten rivinvaihdot voidaan järjestää kirjoittamalla haluttuun rivinvaihtokohtiin joku harvinainen kirjainyhdistelmä ja vaihtamalla ne sitten sopivalla tekstinkäsittelyohjelmalla rivinvaihdoiksi. Seuraavassa esimerkissä annetaan rivinvaihtokohtiin merkkiyhdistelmä ”\n”.

```
ogr2ogr -f GPX bussipysäkit.gpx finland.osm.pbf -sql "select name as 'name',  
CONCAT('ref; ',ref,'\n','roskis; ',waste_basket,'\n','katos; ',shelter) as  
'desc' from points where highway='bus_stop'"
```

Etsi-korvaa-toiminnon jälkeen yhden bussipysäkin kuvaus GPS-tiedostossa näyttää tältä:

```
<wpt lat="60.356638" lon="25.1294777">  
  <name>Haukitie</name>  
  <desc>ref; V8618  
        roskis; yes  
        katos; yes</desc>  
</wpt>
```

Erikoismerkkejä, kuten kaksoispiste ja ääöÄÄÖ ei myöskään saa menemään läpi komentoriviltä, mutta sama etsi-korvaa-jälkikäsittelykikka toimii niilläkin.

C) Kohteiden poimiminen suorakaiteen muotoiselta alueelta

GPS-laitteen muisti saattaa täyttyä koko maan bussipysäkeistä, mutta jos käyttäjä aikoo pysytellä pienemmällä alueella, niin ogr2ogr-ohjelmalle voi antaa -spat -parametrin avulla suorakaiteen muotoisen rajauksen. Edellä oleva komento voidaan muokata poimimaan pääkaupunkiseuden bussipysäkit näin:

```
ogr2ogr -f GPX bussipysäkit.gpx finland.osm.pbf -sql "select name as 'name',  
CONCAT('ref; ',ref,'\n','roskis; ',waste_basket,'\n','katos; ',shelter) as  
'desc' from points where highway='bus_stop'" -spat 23.78 59.58 25.68 61.01
```

Kannattaa huomata, että rajauksessa annettavat koordinaatit täytyy antaa lähtöaineiston käyttämän koordinaattijärjestelmän mukaan. Aikaisemmin esitetyssä WFS-palveluesimerkissä pitää siis aluerajaus antaa EPSG:3067 -koordinaatteina.

D) Eri tietotyyppiä olevien ominaisuuksien ketjuttaminen CONCAT-funktiolla

Palataan WFS-palvelusta haettavien parkkilippuautomaattien esimerkkiin. Automaattien lisätiedot olisi hyödyllistä kirjoittaa kaikki ”desc”-kenttään, mutta jos sitä yrittää, niin saa aikaan vain virheilmoituksen. Syynä on tietotyyppien yhteensopimattomuus.

```
osoite: String (0.0)  
vyöhyke: Integer (0.0)  
maksullisuus aika: String (0.0)  
max_parkkiaika: String (0.0)  
muu_maksutapa: String (0.0)
```

Huomataan, että ”vyöhyke” on kokonaislukutyyppiä (integer) kun taas muut ovat merkkijonoja. Mutta ei hätää, vyöhykkeen tietotyyppi voidaan muuttaa SQL-lausekkeessa. Kas näin se käy:

```
ogr2ogr -f GPX parkkiautomaatit2.gpx -t_srs epsg:4326  
WFS:http://hip.latuviitta.org/cgi-bin/tinyows -sql "select osoite as  
'name',CONCAT('Klo; ',maksullisuus aika,'\n','P-aika;  
' ,max_parkkiaika,'\n','Muuta; ',muu_maksutapa,'\n','Vyohyke; ',CAST(vyohyke AS  
character)) as 'desc' from hki_parkkilippuautomaatit"
```

E) Muiden GPX-tiedostomuodon ominaisuuksien hyödyntäminen

GPX-tiedostomuoto tukee kohdepisteille seuraavia ominaisuustietokenttiä:

name (nimi), **cmt** (kommentti), **desc** (kuvaus), **src** (lähde), **link** (linkki) , **sym** (symboli), **type** (tyyppi)

Voi olla, että opiskelemalla aihetta pystyisi tekemään hienoja juttuja esimerkiksi link- ja sym-kenttien avulla. Lisäksi GPX-standardi sallii toimittajakohtaisten lisäominaisuuksien käytön. Näiden avulla voidaan esimerkiksi määrätä hälytyskäyttökohteita lähestyttäessä. Tavallisimmin tätä ominaisuutta varmaankin käytetään hälyttämään nopeusvalvontakameroista, mutta turvallisuutta voisi parantaa enemmän jos hälytys tulisi kouluja tai karikoita lähestyttäessä.

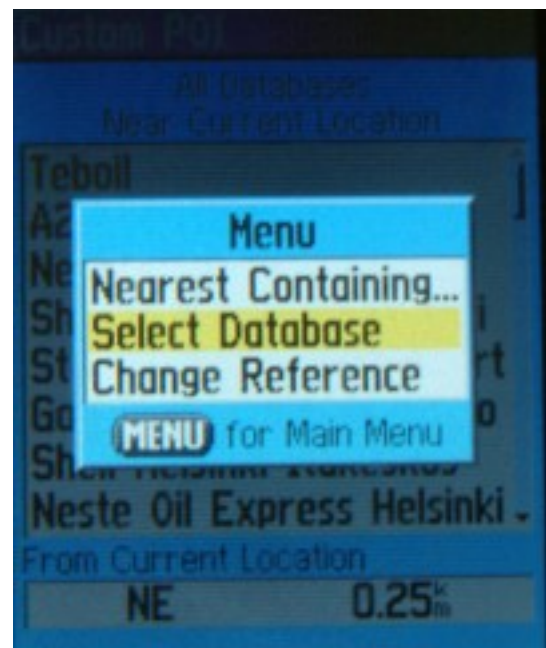
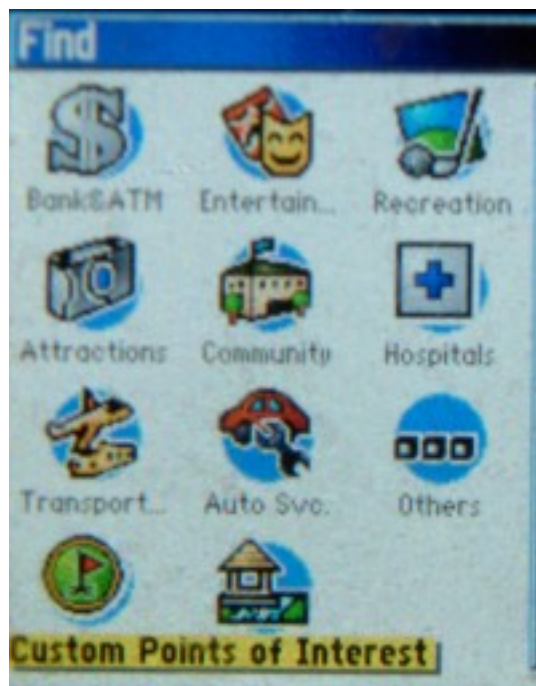
Nopeammin, paremmin ja halvemmalla eli tulosten arviointi

Tässä esitetyt mallisuoritukset ovat nopeita ja ilmaisia, joten parempaan tulokseen pääseminen vaatii lähinnä laadun parantamista.

- Kiistämätön laatuongelma mallisuorituksessa on se, että skandinaavisia kirjaimia ja muita erikoismerkkejä ei voida lähettää GPX-tiedoston kuvauskenttään suoraan komentoriviltä. Rivinvaihtomerkki kuuluu näihin erikoismerkkeihin, joten monirivisten kommenttien tekeminen ei onnistu ilman erillistä tekstinkäsittelyvaihtoa.
- OGR:n SQL-tuki ei myöskään ole erityisen laaja, ja siitä syystä kaikki sinänsä oikein muodostetut SQL-kyselyt eivät välttämässä toimi. Aiheesta kiinnostuneiden kannattaa lukea http://www.gdal.org/ogr/ogr_sql.html
- Laatuongelmaksi voidaan katsoa sekin, että komentorivi ei ole kaikkien mielestä paras käyttöliittymä.

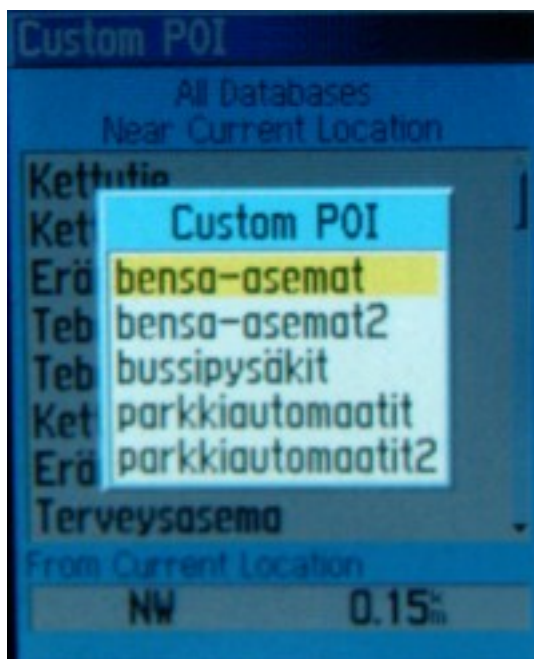
Ylipäätään en oikein usko, että mallisuorituksesta sellaisenaan tulisi suurten massojen suosikkia, koska se vaatii sen verran paljon näpertelyä. Ohjelmointitaitoinen ihminen pystyy kuitenkin halutessaan varmasti käärimään esitetyt toiminnallisuuden kauniiseen ja helppokäyttöiseen kuoreen.

Lopputulokset GPS:n näytöllä



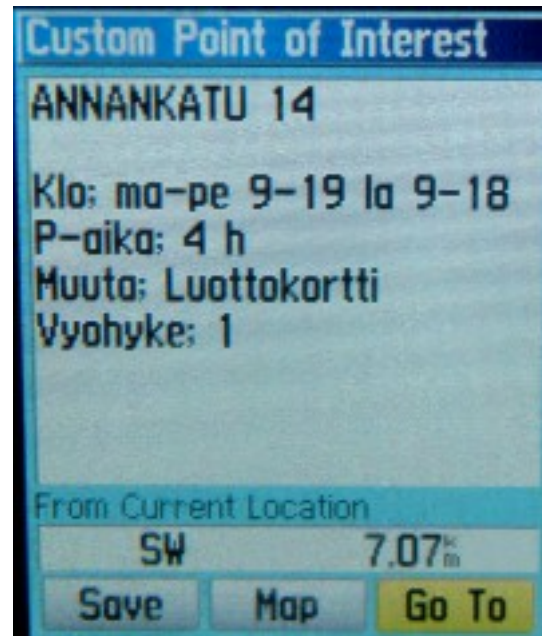
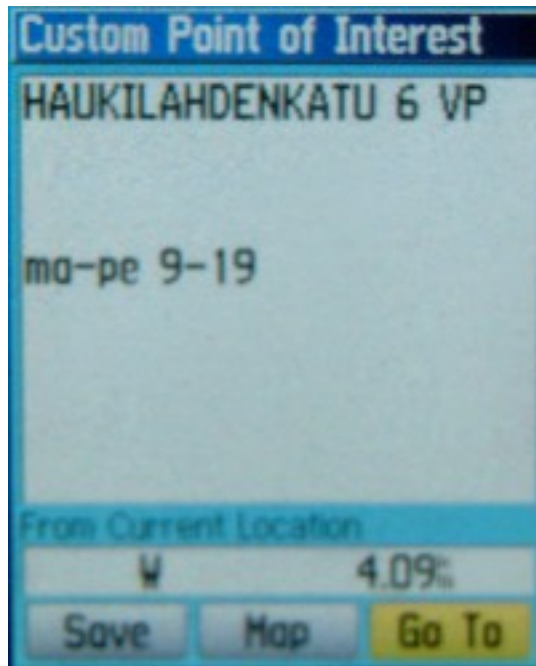
Vasemmalla: GPS-kohdepisteet asennuvat räätälöityihin kohteisiin.

Oikealla: Jos kohteisiin vietiin useita GPX-pisteitä, niin jokainen niistä muodostaa oman tietokannan.



Vasemmalla: Haun kohteena oleva tietokanta voidaan valita.

Oikealla: Parkkiautomaattitietokanta valittuna, kohteet näkyvät listalla etäisyyden mukaan järjestettynä.



Vasemmalla: Perussuorituksen mukaan tuotujen parkkilippuautomaattien ominaisuudet näytöllä.
Oikealla: Edistyneellä komennolla tuotujen parkkilippuautomaattien ominaisuustiedot.



Vasemmalla: Edistyneellä komennolla tuodun bussipysäkin tiedot.
Oikealla: Bensa-asema GPS:n karttanäytöllä, symbolina räätälöity "bensa-asemat.bmp"