

OpenStreetMap

Daten nutzen



OpenStreetMap – Daten nutzen

Zur Person:

Wolfgang Hinsch

Hamburg

Vermessungsingenieur

aktiv bei OSM seit 2008

Inhalt

- OSM – Was ist das
- Anwendungsmöglichkeiten
- Das Datenmodell von OSM
- Andere Datenmodelle
- Quellen für OSM-Daten
- Werkzeuge zur Umsetzung der Daten
- Programme zur Erzeugung von Kartenbildern

OpenStreetMap – Was ist das?

- Keine offene (Straßen-) Karte
- Sondern eine geographische Datensammlung
- Warum dann der Name OpenStreetMap?

Anwendungsmöglichkeiten

- Landkarten
- Seekarten
- Routing
- Katastrophenhilfe
- Statistische Auswertungen
- Sozialgeografische Analysen
- Unterrichtszwecke

Das Datenmodell von OpenStreetMap

- Inkompatibel zu allen bestehenden GIS-Anwendungen
- Alle Koordinaten beruhen auf der Mercator-Projektion (WGS84)
- Formatiert als xml-Datei (.osm) oder komprimiert binär (.pbf, .o5m)
- Organisation, Wartung und Verzeichnis der Kodierung:

wiki.openstreetmap.org

Formen der Objekte in den OSM-Daten

- Punktförmig (Baum, Straßenlaterne, ...)
- Linienförmig (Straße, Weg, Gleis, ...)
- Flächenförmig (Wiese, See, Wald, ...)
- Netzförmig (Wanderwege, Buslinien, ...)
- Organisatorisch (Gemeinde, Schule, ...)

Das Datenmodell - Datentypen

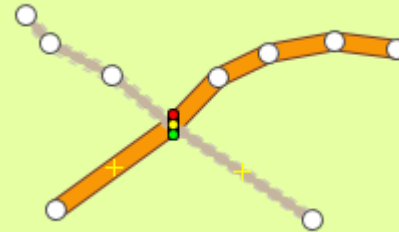
- Punkte („nodes“) als Koordinatenträger
- Linien („ways“) als Ketten von Punktnummern
- Relationen für alles, was nicht nur Punkt oder Linie ist

Aufbau der Datensätze

- Metadaten
 - ID (64 Bit)
 - Versionsnummer
 - Changeset-Nr.
 - Benutzername
 - Benutzernummer
 - Zeitstempel
 - Visible (gültig)
- Vom Datentyp abhängig
 - Nodes:
Koordinatenpaare
 - Ways:
Punktnummern
 - Relationen:
Typ der Relation und Memberliste
- Nutzdaten (Tags)
 - Wertepaare key=value

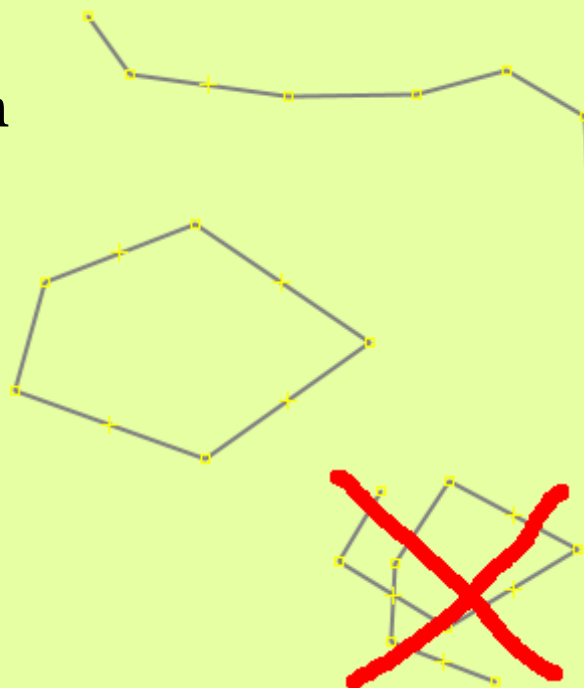
Punkt („Node“)

- Datenträger der Koordinaten
 - Meta-Tags lat=... und lon=... für Koordinaten
 - Genauigkeit 7 Dezimalstellen = ~5mm am Äquator
- Kann ein eigenständiges Objekt repräsentieren
 - Baum, Hausnummer, POI, Ort, ...
- Kann ausschließlich ein Stützpunkt einer Linie sein
- Oder eine Kombination aus beiden
 - Hauseingang, Durchlass im Zaun, Ampel, Bahnübergang, ...



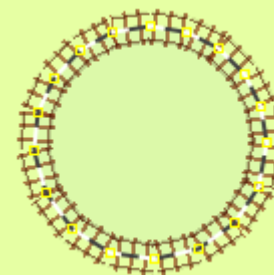
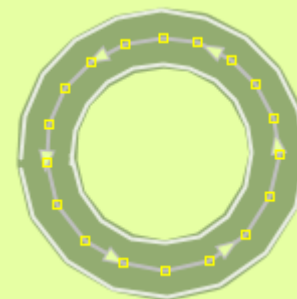
Linie („Way“)

- Stützt sich auf Punkte, hat keine eigenen Koordinaten
- Enthält max. 2000 Punkte
- Kann offen
- oder geschlossen sein
- Darf sich nicht selbst kreuzen



Linie („Way“)

- Eigenständiges Objekt
 - Ein Linienobjekt (Weg, Zaun, Kreisverkehr, ...)
 - Ein Flächenobjekt (Parkplatz, Gebäude, Wiese, ...)
 - Beides (Eingezäunte Wiese, ...)
- Oder Teil eines Objektes (Relation)
 - Als eigenes Objekt (Zaun, ...)
 - Als reine Konstruktionslinie, dann stehen die Tags in der Relation

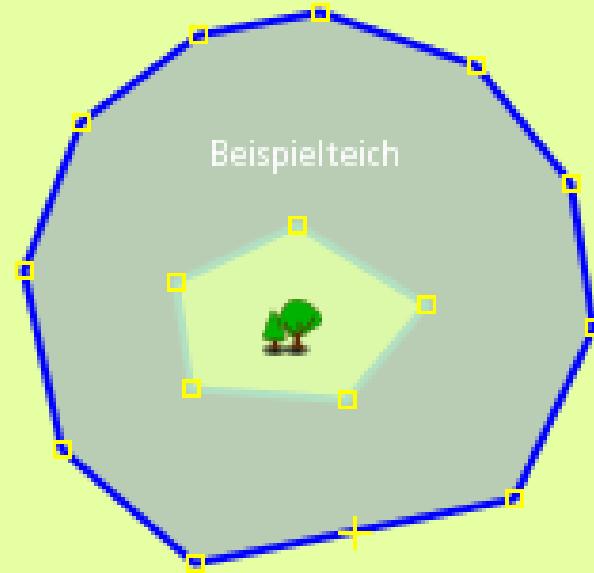


Relation

- Eine Art Container
- Kann jedes Objekt repräsentieren
- Wird definiert mit Meta-Tag `type=value` (multipolygon, ...)
- Kann Punkte, Linien und Relationen enthalten
- Keine Rekursionen
- Beschreibung durch Wertepaare (tags) wie bei Punkten und Wegen
- Verknüpfung von Objekten („Members“) durch Rollenzuweisung („Role“)
 - `type=` (node, way, relation)
 - `ref=` (ID des Elementes)
 - `role=` Rolle des Elementes

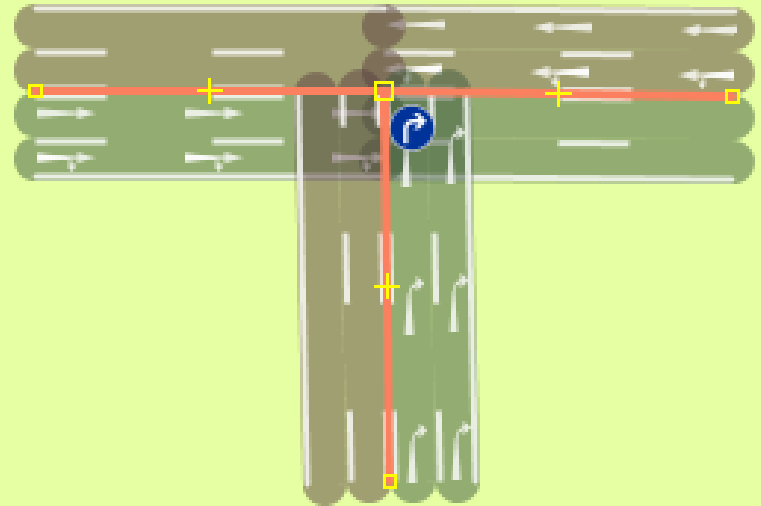
Beispiel für eine Relation: Multipolygon (See mit Insel)

- Tags:
 - type=multipolygon
 - natural=water
- Members:
 - Außenlinie des Sees als way
 - role=outer
 - Außenlinie der Insel als way
 - role=inner



Beispiel für eine Relation: Abbiegebeschränkung

- Tags:
 - type=restriction
 - restriction=only_right_turn
- Members:
 - Weg zur Kreuzung (way):
 - role=from
 - Erlaubtes Ziel (way):
 - role=to
 - Zwischenelement(e) (meistens ein node):
 - role=via



Andere Datenmodelle: Shapefiles

- Weit verbreitetes Datenformat der Firma Esri
- Besteht aus einer Serie von Dateien
- Alle gängigen Projektionen sind möglich
- Kann alternativ Punkte, Linien oder Flächen enthalten
- Die Key-Value-Paare sind als Tabelle organisiert
- Jedes Objekt enthält seine Koordinaten, keine Verknüpfungen

Andere Datenmodelle: postgresql/postgis

- Datenbank postgresql mit der Erweiterung postgis
- Kann beliebig viele Tabellen enthalten
- Tabellen können unterschiedliche Projektionen enthalten
- Innerhalb einer Tabelle nur Punkt, Linie oder Fläche
- Jedes Objekt enthält seine Koordinaten
- Mit postgis sind umfangreiche Umformungen und Auswertungen möglich

Andere Datenmodelle

- SQLite
 - Untermenge von SQL
 - Kein postgis
- Weitere
 - KML, GML
 - GEOJSON
 - GEOTIFF
 - GPX

Quellen für OSM-Daten

- Planet file
 - Komprimiert als .pbf 42 GB, unkomprimiert als .osm 964 GB
- Hosts von Datenabzügen
 - z.B. Geofabrik, Liste im Wiki
- Overpass-API
- Turbo-Overpass

Werkzeuge zur Umsetzung der Daten

- osm2pgsql - Für Webkarte Carto/CSS
- osm2pgrouting - Für Routing-Anwendungen
- mkgmap - Umsetzung für Garmin
- osm2postgresql - Umsetzung in SQL
- osmosis - Vielseitiges Werkzeug
- postgis - Weiterverarbeitung in der DB
- Qgis - Manuelle Weiterverarbeitung

Daten darstellen

- Garmin / BaseCamp / QmapShack
- Mapnik
- Maperitive
- Qgis

Daten mit darstellen QGIS

- Direkter Download in eine SQLite-DB
 - Einfach und praktisch
 - Keine weitere Datenverarbeitung
- Oder Vorverarbeitung mit postgis in postgresql
 - Freie Wahl der Projektion
 - Beliebige Bearbeitungen des Datenmaterials

Erforderliche Software

- Entweder
 - osm2postgresql
 - osmosis
- Oder
 - imposm
- postgresql mit den Erweiterungen
 - postgis
 - hstore
- Qgis
- Scribus oder Inkscape für das Finish

Kartenerzeugung mit Qgis

- Herunterladen der .pbk/.osm – Datei
- Postgresql-DB erzeugen mit
 - osm2postgresql
 - imposm
 - oder eigenen Scripten
- Ggf. mit eigenen Scripten weiterverarbeiten
- Karte mit Qgis erzeugen
- Rand, Rückseite etc. mit scribus erstellen

Anwendungen für Postgis:

- Tags auswerten
- Blinde Feldwege ausblenden
- Multipolygone erstellen
- Beschriftungslinien erzeugen
- Mittellinie für zweibahnige Straßen erzeugen
- ...

Postgis: Bahnübergänge ausdünnen (1)

```
/* Bahnübergänge zusammenfassen */
```

```
update punkte set
```

```
    ist_eisenbahn = true,
```

```
    typ_eisenbahn = 'Bahnübergang',
```

```
    display = true
```

```
where tags @> 'railway=>crossing' or tags @> 'railway=>level_crossing';
```

```
/* Bahnübergänge reduzieren auf gewünschte Elemente*/
```

```
update punkte set display = false
```

```
where ist_eisenbahn
```

```
    and typ_eisenbahn = 'Bahnübergang'
```

```
    and (tags -> 'access' = 'no'
```

```
        or tags -> 'access' = 'private'
```

```
        or tags -> 'abandoned' = 'yes'
```

```
        or tags -> 'disused' = 'yes');
```

Postgis: Bahnübergänge ausdünnen (2)

```
/* Bahnübergänge ausdünnen */
update punkte pu
  set display = false
  from (
    select id_qgis, point_g from punkte where typ_eisenbahn = 'Bahnübergang'
  ) as s1, (
    select id_qgis, point_g from punkte where typ_eisenbahn = 'Bahnübergang'
  ) as s2
where s2.id_qgis > s1.id_qgis
  and ST_Distance(s1.point_g, s2.point_g) < 40.
  and s2.id_qgis = pu.id_qgis;
```

Postgis: Straßenverzeichnis erstellen

```
select distinct po.name, ql.name, gi.name
  from polygone po,
       qlinien ql,
       gitternetz gi
 where flaechen_typ = 'Ort'
       and flaechen_detail = '8'
       and (ist_strasse or ist_weg)
       and ql.name is not null
       and po.polygon_utm32n && ql.linestring_utm32n
       and gi.geom && ql.linestring_utm32n
       and ST_Relate(po.polygon_utm32n, ql.linestring_utm32n, 'T*****')
       and ST_Relate(gi.geom, ql.linestring_utm32n, 'T*****')
 order by 1, 2;
```

Qgis

- Verarbeitet u.a. Postgis-DB, Sql-Lite, Shapefiles, Rasterdaten, WMS, ...
- Beliebig viele Layer aus der gleichen Tabelle möglich
- Signaturen regelbasiert mit Filtern erzeugen
- Ermöglicht das direkte grafische Bearbeiten der Karte
- Projektionen der Layer können automatisch angepasst werden

Vielen Dank für Ihr Interesse