

# osmpg2java – Konvertierung von OSM-Datenbank-elementen zu JTS-Objekten

Doris Silbernagl<sup>1</sup>, Nikolaus Krismer<sup>1</sup>, Günther Specht<sup>1</sup>

<sup>1</sup>Institut für Informatik, LFU Innsbruck · doris.silbernagl@uibk.ac.at

**Zusammenfassung:** In dieser Arbeit wird ein Werkzeug vorgestellt, mit dem Datenbankelemente des OpenStreetMap-Projekts in Java JTS-Objekte konvertiert werden. Hauptaugenmerk liegt hierbei auf (Multi-)Polygonrelationen, da diese nicht standardmäßig explizit mit eigener Geometrie in der Datenbank gespeichert sind. Das entwickelte Werkzeug mit Namen „osmpg2java“ extrahiert sämtliche OSM-Datenbankobjekte und für Relation vom Typ Multipolygon sammelt es alle notwendigen zusammengehörigen Elemente, um gültige JTS-Objekte zu erzeugen. Die erstellten (Multi-)Polygoneometrien werden in die Datenbank gespeichert, um mathematische und geometrische Operationen darauf entweder direkt in der Datenbank selbst, aber vor allem in Java durchführen zu können.

**Schlüsselwörter:** OpenStreetMap, PostgreSQL, Java, JTS, Konvertierungsalgorithmus, Multipolygon

***Abstract:** In this work a tool that transforms database elements of the OpenStreetMap project into Java JTS objects is presented. The key aspect is to resolve the OSM (multi-)polygon relations, as those are not stored explicitly with their geometry in the database tables by default. The conversion tool called “osmpg2java” extracts all the database elements and for relations of type “multipolygon” it is able to assemble all required parts to build valid JTS objects. The generated (multi-)polygon geometries are stored into the database, allowing mathematical and geometric operations to be performed on the geometry field, either directly in the database or especially in Java itself.*

**Keywords:** OpenStreetMap, PostgreSQL, Java, JTS, Conversion algorithm, Multipolygon

## 1 Motivation

Im Bereich von freiwillig erfassten geographischen Informationen, wie sie beispielsweise im Projekt OpenStreetMap (kurz OSM, [www.openstreetmap.org](http://www.openstreetmap.org)) zur Verfügung gestellt werden, existiert die Problematik der Datenqualität (COOPER et al. 2011). Verschiedene Aspekte müssen bei der Evaluierung der Qualität betrachtet werden. Eine Definition von Parametern ist die erste Hilfestellung bei der Beurteilung der Datenqualität und wurde von VEREGIN (1999) in Angriff genommen. Neben diesen Kriterien wurden weitere hinzugefügt und von MOONEY et al. (2010) und CIEPLUCH et al. (2011) untersucht. Eine andere Methodik zur Qualitätsprüfung sind jene empirischen Studien wie etwa durch HAKLAY et al. (2010) und NEIS et al. (2010), die OSM-Daten mit jenen von kommerziellen Kartenherstellern vergleichen. Aktuellere Forschung setzt auf intrinsische Untersuchungen der OSM-Elemente, um Aussagen hinsichtlich der Datenqualität treffen zu können (GOODCHILD et al. 2012, BARRON. 2014). Ein Ansatz dazu ist die nähere Betrachtung des Zusammenhangs der Meta-Informationen mit den zugehörigen geographischen Elementen (DEVILLERS et al. 2002). Hierbei soll deren Korrelation erkennbar werden, welche Schlussfolgerungen auf deren Plausibilität und somit die Datenqualität erlauben (ALI et al. 2014). Mithilfe von mathematischen Berechnungen auf den Elementen (Umfang, Fläche, Durchmesser, Abstände etc.) können Ähnlichkeiten festgestellt werden, die mit weiterführender Analyse zusätzliche wertvolle Informationen über die Objekte selbst liefern.

Um solche Analysen durchführen zu können, ist es förderlich, die OSM-Daten in eine Datenbank zu speichern und die relevanten Informationen daraus abzufragen. Jedoch ist ein Weiterarbeiten innerhalb der Datenbank umständlich und möglicherweise auch rechenintensiv. Daher wird in dieser Arbeit ein Werkzeug mit Namen „osmpg2java“ vorgestellt. Dieses bildet OSM-Elemente direkt auf JTS-Objekte in Java ab (VIVID SOLUTIONS 2003). Für Relationen vom Typ Multipolygon, auf deren Geometrie die Berechnungen für die Analysen erfolgen, erzeugt es mittels eines Algorithmus gültige JTS-Multipolygonobjekte. Somit sind die Elemente nun außerhalb der Datenbank verfügbar und erlauben eine Verarbeitung für Analysezwecke innerhalb des Java-Codes.

Dieser Artikel teilt sich in folgende Kapitel: Abschnitt 2 beschreibt den entwickelten Algorithmus zur Konvertierung von simplen und komplexen OSM-Datenbankelementen in JTS Objekte und der Erstellung von Multipolygongeometrien. In Abschnitt 3 wird ein Beispiel zum visuellen Vergleich angeführt, um die Korrektheit des Algorithmus zu untersuchen. Ebenso ist eine Evaluation der Algorithmus-Performanz hinsichtlich definierter geographischer Regionen enthalten. Dieser Artikel schließt mit einer Zusammenfassung und einem Ausblick auf zukünftige Arbeiten.

## 2 Algorithmus: osmpg2java

In diesem Abschnitt wird auf die Funktionsweise des Algorithmus zur Konvertierung der OSM-Datenbankelemente nach JTS-Objekten näher eingegangen. Die Aufgabe des Algorithmus ist es, die Datenbankelemente von OSM auf Java JTS-Objekte abzubilden. In dieser Arbeit wurde eine PostgreSQL 9.4/PostGIS 2.2.0 Datenbank im „pgsnapshot“ Schema des OSM-Datenimportertools „osmosis“ verwendet (OSMOSIS 2015). Durch osmosis befüllte Datenbanken eignen sich für Analysezwecke von OSM (OPENSTREETMAP WIKI DBS 2016), besonders mit dem Fokus auf Qualitätsanalysen.

Für die Konvertierung werden zunächst Java-Klassen benötigt, die sämtlichen Informationen der OSM-Elemente speichern können. Der geographische Part wird dabei durch die Verwendung der JTS-Bibliothek abgedeckt. Für jedes Element kann die OSM-ID, das Tagging und die Geometrie gespeichert werden. Für die Geometrieinformation von Wegen und Punkten kann direkt der JTS-WKB-Reader verwendet werden. Daher ist dieser Konvertierungsschritt simpel und direkt durchführbar.

Der wesentlich interessantere Anwendungsfall und der Grund der Entwicklung des Algorithmus sind die speziellen Relationen-Elemente von OSM. Relationen können Wege, Punkte und sogar weitere Relationen zusammenfassen. Multipolygone sind ein solcher Spezialfall, da mit ihnen komplexe Sachverhalte der Realwelt dargestellt werden, wie beispielsweise Gebäude mit Innenhöfen oder Seen mit Inseln. Valide Multipolygone in OSM können somit sehr verschieden gestaltet sein (OPENSTREETMAP WIKI MP 2015).

Um nun Analysen auf Basis der Geometrie eines Multipolygons durchführen zu können, muss dessen geometrische Information vorhanden sein. Da die Multipolygongeometrie im vorliegenden Fall aber nicht direkt in der Datenbank gespeichert ist, wurde der hier präsentierte Erstellungsalgorithmus entwickelt. Für Relationen vom Typ Multipolygon extrahiert

das Tool sämtliche zugehörige Datenbankelemente und erzeugt daraus ein gültiges JTS-Multipolygonobjekt. Um die Geometrie zu ermitteln, müssen vor allem die Wege der Relation betrachtet werden. Diese können einerseits bereits geschlossene Flächen und somit einfache Polygone darstellen, andererseits resultieren sich berührende Wege in Summe ebenso in geschlossene Flächen. Der Algorithmus prüft sämtliche Möglichkeiten einer Multipolygonkonstruktion ab und erstellt daraus die entsprechende Geometrie. Dabei können unterschiedliche Probleme auftreten, wie beispielsweise ungültige oder unvollständige Daten. In solchen Fällen werden entsprechende Fehlermeldungen ausgegeben. Nach erfolgreichem Abschluss des Erstellungsprozesses speichert das Tool die Geometrien in eine eigene Tabelle der Datenbank. Nun können mathematische Operationen entweder direkt in der Datenbank, aber vor allem in Java durchgeführt werden. Abbildung 1 beschreibt den Algorithmus in Pseudocode.

**Initial-Bedingung:** R = Relation mit Tagging: „type=multipolygon“

```
1  LW = neue Liste der zu verbindenden Wege
2  LC = neue Liste der verbundenen Wege
3  for each w ∈ R
4  if w ist vollständig in Datenbank vorhanden und typeof(w) = way
5    if w ist geschlossen then
6      füge w zur Liste der LC hinzu
7    else if LW ist leer then
8      füge Weg zur Liste LW hinzu
9    else
10     nw = w
11     if nw berührt Element in Liste LW then
12       nw = homogenisiere Orientierung und verbinde berührende Wege
13       entferne Einzelwegeinträge der soeben verbundenen Wege in Liste LW
14     end if
15     if nw ist geschlossen then
16       füge nw zur Liste LC hinzu
17     else
18       füge nw zur Liste LW hinzu
19     end if
20   end if
21 end if
22 end for
23 if LW ist nicht leer then
24   entsprechende Fehlermeldungen werden ausgegeben (unvollständig, invalid)
25 end if
26 m = erstelle Multipolygon aus Liste LC unter Berücksichtigung der inneren und
    äußeren Wege
27 return m
```

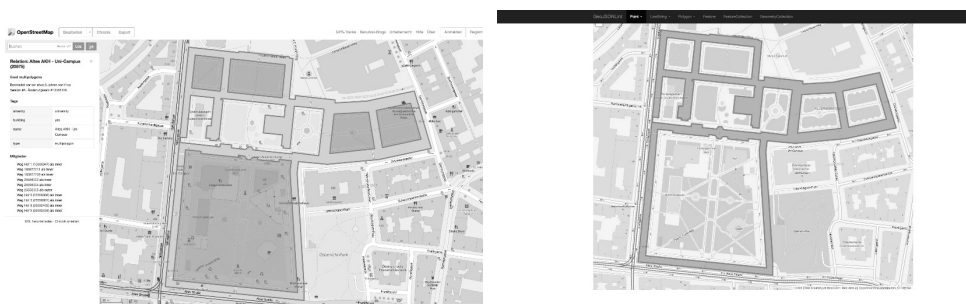
**Abb. 1:** Erstellungsalgorithmus für Multipolygon(geometrie) in Pseudocode

### 3 Evaluation

Als Datengrundlage für die visuelle und Performanz-Evaluierung wurde ein OSM-Datenexport von Österreich verwendet. Dieser wurde am 2015-12-16 von Geofabrik geladen (GEO-FABRIK 2015) und wie bereits erwähnt mit osmosis in eine PostgreSQL 9.4/PostGIS 2.2.0-Datenbank im „pgsnapshot“ Schema importiert.

#### 3.1 Visuelle Evaluierung

Die Multipolygonrelation des alten AKH-Gebäudes (Allgemeines Krankenhaus Wien) auf dessen Uni Campus hat einen äußeren Weg, der den Rahmen des Polygons definiert und das eigentliche Gebäude darstellt und Innenhöfe, die vom eigentlichen Gebäude abgezogen werden müssen. Die Besonderheit bei dieser Relation ist die Reihenfolge der einzelnen Elemente, die anders als ursprünglich angenommen, nicht mit dem äußersten Relationen-Mitglied begann, sondern mit den inneren. Daher war beim Erstellen des Multipolygons das korrekte Zusammenfügen der Wege unter Berücksichtigung der Orientierung und Reihenfolge abzuarbeiten.



**Abb. 2:** Visuelle Repräsentation des Multipolygons „Altes AKH – Uni-Campus“

Abbildung 2 enthält die grafische Visualisierung des Multipolygons: im linken Abschnitt das Originaldatenset aus OSM und im rechten Bereich das generierte in GeoJSONLint (GEOJSONLINT 2015), einem Werkzeug zur Validierung und Darstellung von GeoJSON. Hierbei fällt auf, dass die Abbildung der mit osmpg2java erstellten Geometrie exakter ist als jene von OSM und somit die eigentlichen Daten widerspiegelt: die Innenhöfe sind deutlich als freie Flächen erkennbar und das Gebäude selbst sticht durch die Färbung hervor.

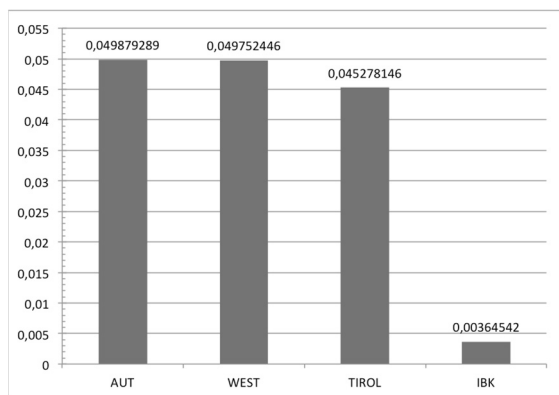
#### 3.2 Performanz Evaluierung

Dieser Abschnitt enthält eine Evaluation über die Performanz des Algorithmus zur Erstellung der Multipolygoneometrie und des Zurückspeicherns jener in die Datenbank. Die Berechnungen und sämtliche Messungen wurden auf einem virtualisierten Linux-Server mit zwei CPUs mit je 2.67 GHz und 16 GB RAM unter Verwendung von Java OpenJDK Version 1.8.0\_51 durchgeführt. Die eigentliche Performanz wird in Abbildung 3 dargestellt.

**Tabelle 1:** Testdatenbanken

Datenbank	(aufgerundete) Größe der OSM Geodaten in MB	Anzahl Relationen gesamt	Anzahl Multipolygonrelationen	Anzahl gespeicherter Multipolygongeometrien
Österreich (AUT)	15.955	82.799	47.464	41.678
Westösterreich (WEST)	4.318	20.838	12.026	11.888
Tirol	2.278	11.084	6.539	6.421
Innsbruck (IBK)	80	1.762	205	117

In Tabelle 1 sind die zu Testzwecken verwendeten Datenbanken ersichtlich. Die Regionen wurden aus dem Österreichdatenset herausgeschnitten. Westösterreich umfasst die Bundesländer Vorarlberg, Tirol und Salzburg.

**Abb. 3:** Performanz des Multipolygon-Erstellungsalgorithmus

In Abbildung 3 ist die statistische Auswertung der Performanz des Multipolygon-Erstellungsalgorithmus dargestellt. Es wurde hierbei die durchschnittliche Dauer der Verarbeitung pro Multipolygonelement in Sekunden gemessen, wobei im Rahmen einer blockweisen Verarbeitung zu einem Zeitpunkt 100 Relationen in der Java-VM vorhanden sind. Innsbruck hat eine sichtbar kürzere Bearbeitungszeit, was vermutlich auf die Größe des Datensets und damit einhergehend auf die mögliche Zwischenspeicherung des gesamten Datenbestands im Datenbankcache zurückzuführen ist. Insgesamt kann hinsichtlich der Skalierbarkeit des Algorithmus eine nahezu lineare Performanz angenommen werden.

## 4 Zusammenfassung und Ausblick

Der entwickelte Algorithmus hilft bei der Konvertierung von OSM-Datenbankelementen in JTS Objekte und erstellt dabei die notwendigen Multipolygongeometrien. Das Werkzeug

„osmpg2java“ arbeitet dabei, wie in Kapitel 3 evaluiert, effizient. Zudem werden bei invaliden und unvollständigen Daten entsprechende Fehlermeldungen ausgegeben. Eine Rekursion innerhalb des Algorithmus erlaubt die korrekte Erstellung selbst komplexester Multipolygone. Die Prüfung auf Korrektheit des Algorithmus erfolgt hier durch visuelle Vergleiche der Originaldaten mit den erstellten.

Fortführende Arbeit wird vor allem im Bereich der Fehlerbehandlung gesehen, wo beispielsweise der Algorithmus dahin gehend erweitert werden könnte, dass bei Fehlern von relevanten Daten, diese automatisiert korrigiert werden. Zudem würde die Ermittlung der optimalen Blockgröße bei der Verarbeitung die Performanz des Algorithmus steigern.

## Literatur

- ALI, A. L. & SCHMID, F. (2014), Data Quality Assurance for Volunteered Geographic Information. In: *Geographic Information Science*, Springer International Publishing, 126-141.
- BARRON, C., NEIS, P. & ZIPF, A. (2014), A comprehensive framework for intrinsic OpenStreetMap quality analysis. *Transactions in GIS*, 18 (6), 877-895.
- CIEPLUCH, B., MOONEY, P. & WINSTANLEY, A. (2011), Building generic quality indicators for OpenStreetMap.
- COOPER, A. K., COETZEE, S., KACZMAREK, I., KOURIE, D. G., IWANIAK, A. & KUBIK, T. (2011), Challenges for quality in volunteered geographical information.
- DEVILLERS, R., GERVAIS, M., BÉDARD, Y. & JEANSOULIN, R. (2002, March), Spatial data quality: from metadata to quality indicators and contextual end-user manual. In: *OEEPE/ISPRS Joint Workshop on Spatial Data Quality Management*, 21-22.
- GEOFABRIK (2015), Geofabrik: Download server for OpenStreetMap data, web based download application: <http://download.geofabrik.de/> (10/2015).
- GEOJSONLINT (2015), <http://geojsonlint.com/> (10/2015).
- GOODCHILD, M. F. & LI, L. (2012), Assuring the quality of volunteered geographic information. *Spatial statistics*, 1, 110-120.
- HAKLAY, M. (2010), How good is volunteered geographical information? A comparative study of OpenStreetMap and Ordnance Survey datasets. *Environment and planning B, Planning & design*, 37 (4), 682.
- MOONEY, P., CORCORAN, P. & WINSTANLEY, A. C. (2010), Towards quality metrics for OpenStreetMap. In: *Proceedings of the 18th SIGSPATIAL international conference on advances in geographic information systems*. ACM, 514-517.
- NEIS, P., ZIELSTRA, D., ZIPF, A. & STRUNCK, A. (2010), Empirische Untersuchungen zur Datenqualität von OpenStreetMap-Erfahrungen aus zwei Jahren Betrieb mehrerer OSM-Online-Dienste. *Angewandte Geoinformatik 2010*.
- OPENSTREETMAP WIKI MP (2015), Multipolygon Relation. <http://wiki.openstreetmap.org/wiki/Relation:multipolygon> (12/2015)
- OPENSTREETMAP WIKI DBS (2016), Databases and data access APIs. [http://wiki.openstreetmap.org/wiki/Databases\\_and\\_data\\_access\\_APIs](http://wiki.openstreetmap.org/wiki/Databases_and_data_access_APIs) (04.2016).
- OSMOSIS (2015), OSMOSIS – A command line Java application for processing OSM data. <http://wiki.openstreetmap.org/wiki/Osmosis> (10/2015).
- VEREGIN, H. (1999), Data quality parameters. *Geographical information systems*, 1, 177-189.
- VIVID SOLUTIONS (2003), Java Topology Suite. <http://www.vividsolutions.com/jts/jtshome.htm> (10/2015).