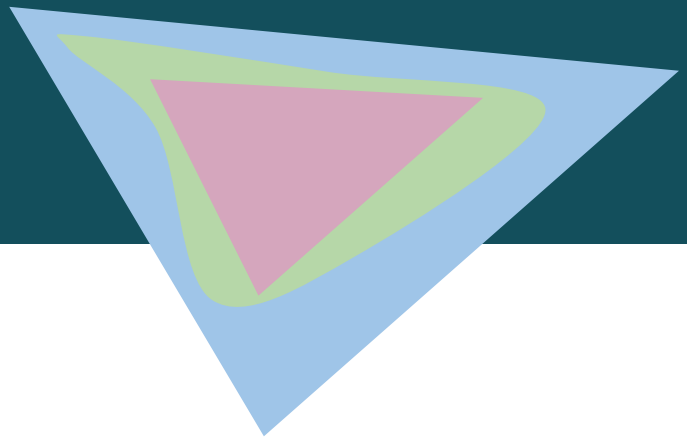
Conversion of OSM data to RDF Data and the use of Simplified Geometries when computing Spatial Relations

Iradj Solouk



Part I
Problems

Problems

- Turning OSM Data to RDF
  - Memory efficiency

Problems

- Turning OSM Data to RDF
  - Memory efficiency
  - Structure of OSM Data

Structure of OSM Data

- Nodes, Ways, Relations
- IDs
- May have OSM Tags
  - (key, value) pairs like “Amenity = restaurant”

Structure of OSM Data

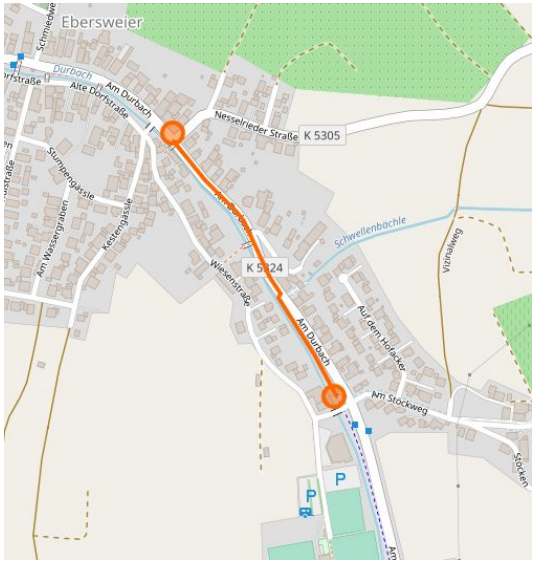
- Nodes, Ways, Relations
- IDs
- May have OSM Tags
  - (key, value) pairs like “Amenity = restaurant”
- Nodes are points in space(longitude, latitude)
- Ways are polylines/polygons
  - No lon/lat, but IDs to nodes
- Relations are groups of possibly Nodes, Ways and/or Relations
  - ID and role pairs are given as references

Problems

- Turning OSM Data to RDF
  - Memory efficiency
  - Structure of OSM Data
    - Computing explicit geometry by resolving references

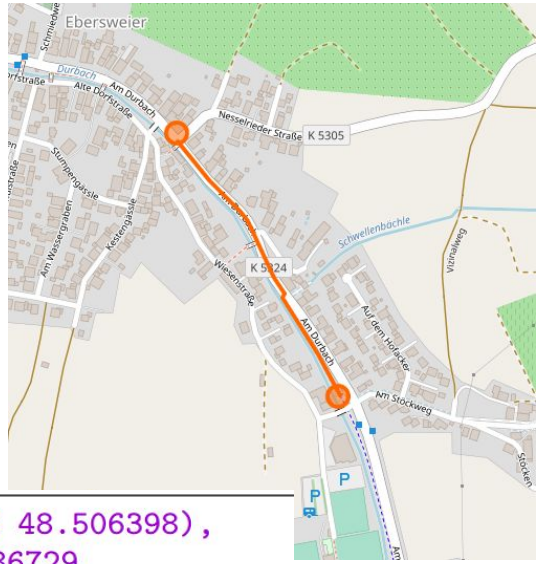
Example of converted element

```
<relation id="13038953" …>

  <member type="node" ref="8964237082" role="guidepost"/>
  <member type="way" ref="220021661" role=""/>
  <member type="way" ref="34611919" role=""/>
  <member type="node" ref="8964237098" role="guidepost"/>
  …
</relation>
```



Example of converted element

```
<relation id="13038953" …>

  <member type="node" ref="8964237082" role="guidepost"/>
  <member type="way" ref="220021661" role=""/>
  <member type="way" ref="34611919" role=""/>
  <member type="node" ref="8964237098" role="guidepost"/>
  …
</relation>
```




```
osmrel:13038953 geo:hasGeometry "GEOMETRYCOLLECTION(POINT(7.986672 48.506398),
    POINT(7.989988 48.502834),  LINESTRING(7.986689 48.506284,7.986729
    48.506251,7.987331 48.505752,7.987984 48.505333,7.988140 48.505199,7.988266
    48.505046,7.988351 48.504922,7.988450 48.504726,7.988661 48.504433,7.988891
    48.504216,7.988844 48.504160,7.989198 48.503798,7.989576 48.503397,7.990016
    48.502844))"^^geo:wktLiteral .
```

Problems

- Turning OSM Data to RDF
  - Memory efficiency
  - Structure of OSM Data
    - Computing explicit geometry by resolving references
    - Computing spatial relations: contains & within

Problems

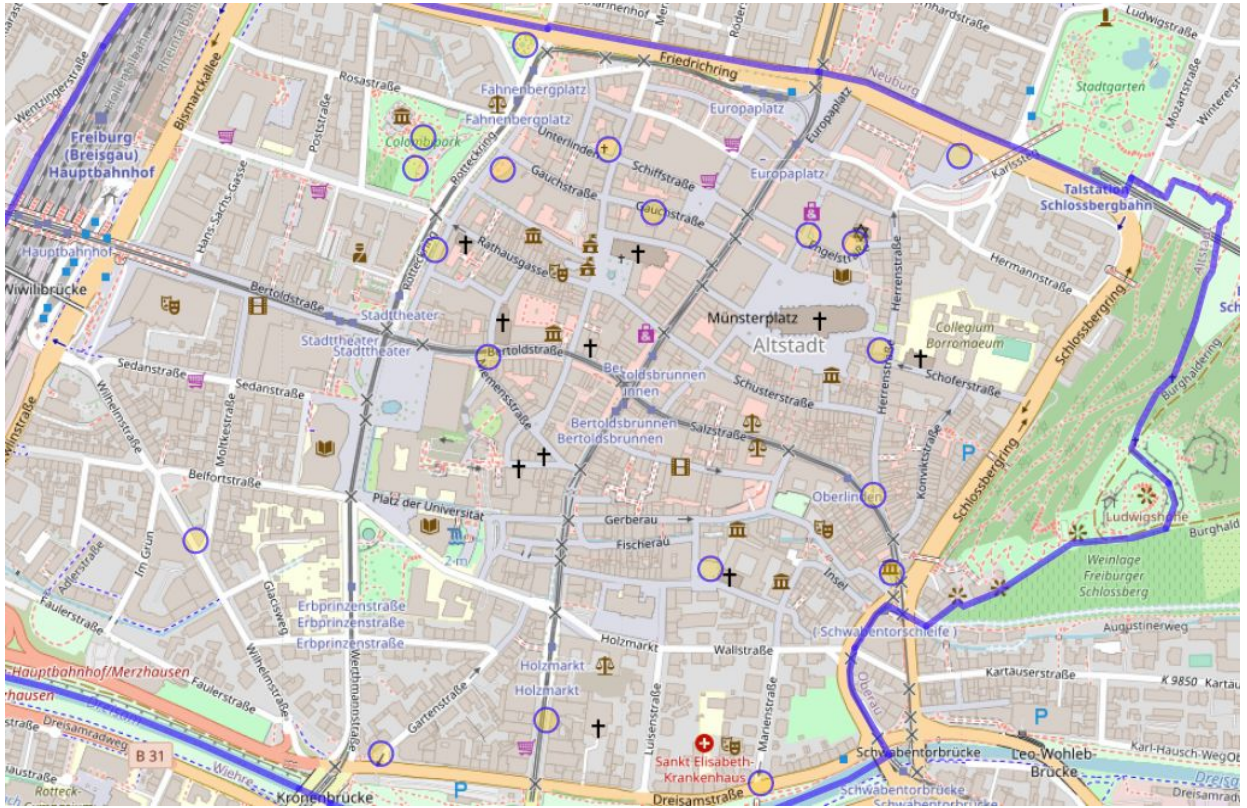
- Turning OSM Data to RDF
  - Memory efficiency
  - Structure of OSM Data
    - Computing explicit geometry by resolving references
    - Computing spatial relations: contains & within
    - Computing way-clusters

Way-cluster

- Streets are fragmented

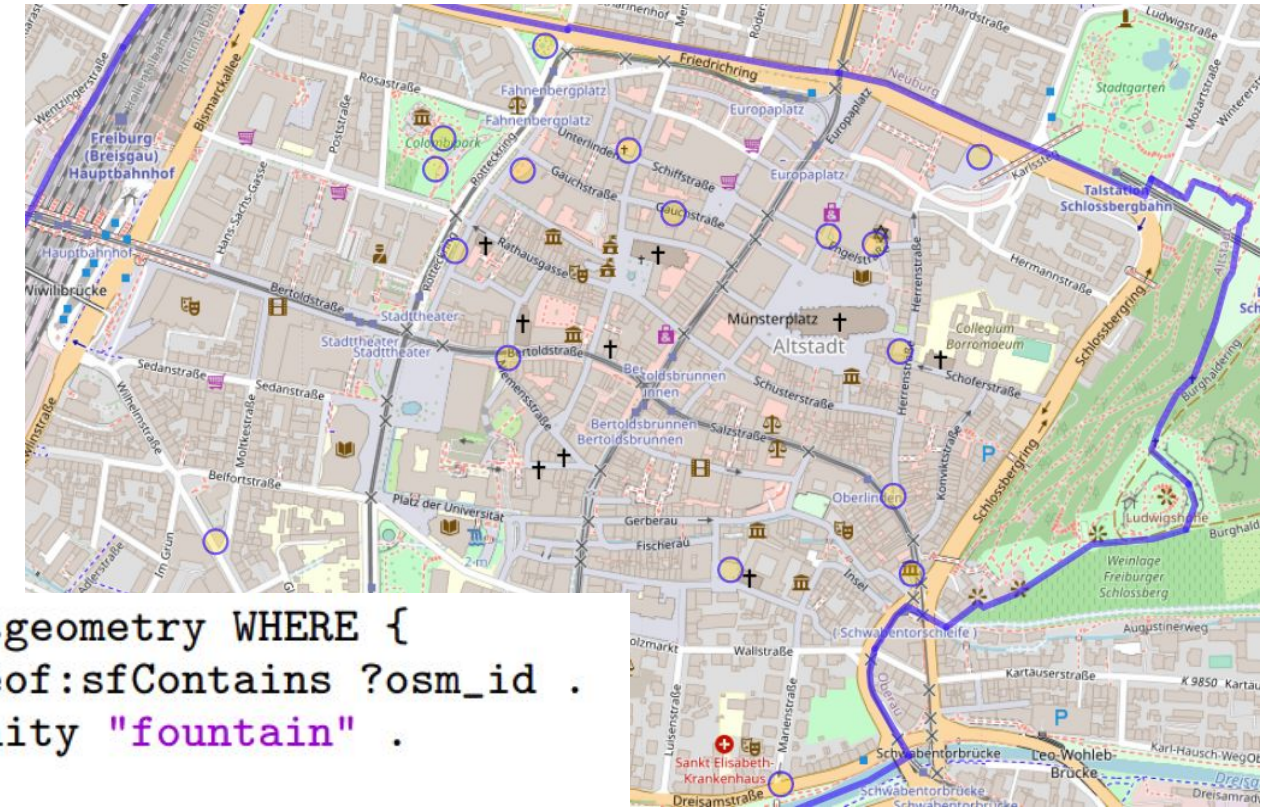
Possible Application - Overpass Turbo

```
area[name="Altstadt"]
;
node
  [amenity=fountain]
  (area);
out;

rel(1960176);
way(r);
out geom;
```



Possible Application - Overpass Turbo

```
area[name="Altstadt"]
;
node
  [amenity=fountain]
  (area);
out;

rel(1960176);
way(r);
out geom;
```




```
SELECT ?osm_id ?hasgeometry WHERE {
  osmrel:1960176 geof:sfContains ?osm_id .
  ?osm_id osmt:amenity "fountain" .
}
```

PartII

Solutions

Solutions

- Turning OSM Data to RDF
  - Memory efficiency
  - Structure of OSM Data
    - Computing explicit geometry by resolving references
    - Computing spatial relations: contains & within
    - Computing way-clusters

Memory efficiency

- Serialize the data with STXXL
  - Tries to mimic STL containers and algorithms
  - Not in-memory but on-disk
  - Sorting is possible, but only POD structures can be used

Solutions

- Turning OSM Data to RDF
  - ~~Memory efficiency~~
  - Structure of OSM Data
    - Computing explicit geometry by resolving references
    - Computing spatial relations: contains & within
    - Computing way-clusters

Resolving geometry

- E.g. resolving geometry of ways
  - Sort node vector(containing lon/lat) by ID
  - Sort way vector(containing reference ID) by reference ID

Solutions

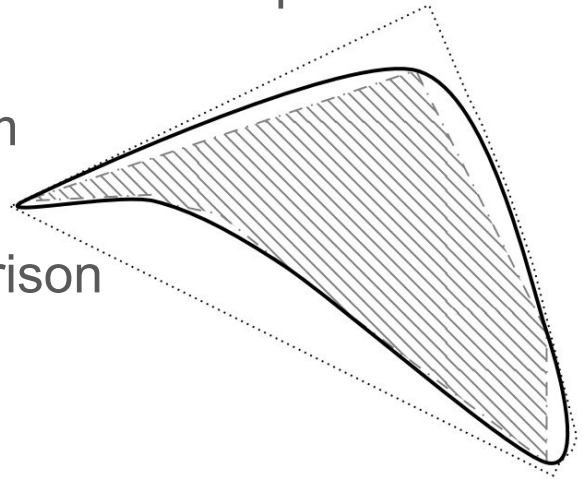
- Turning OSM Data to RDF
  - ~~Memory efficiency~~
  - Structure of OSM Data
    - ~~Computing explicit geometry by resolving references~~
    - Computing spatial relations: contains & within
    - Computing way-clusters

Spatial relations

- Spatial relations only wrt. elements tagged as *administrative boundaries*
- Using Boosts Rtree as a spatial index

Spatial relations

- Spatial relations only wrt. elements tagged as *administrative boundaries*
- Using Boosts Rtree as a spatial index
- Using overestimated and underestimated polygon to infer spatial relations: contains & within
  - Modified Ramer-Douglas-Peucker Algorithm
  - Simplified Polygons have less points
- Use directed acyclic graph for transitive comparison



Solutions

- Turning OSM Data to RDF
  - ~~Memory efficiency~~
  - Structure of OSM Data
    - ~~Computing explicit geometry by resolving references~~
    - ~~Computing spatial relations: contains & within~~
    - Computing way-clusters

Way-clusters

- Sort wayrefs by name ID and reference ID
- Find the connected components
- Intersect tags for connected components

Part III
Evaluation

Point reduction from using simplified polygons

	Freiburg	Baden-Württemberg	Germany	Europe
averaged point reduction for overestimations	82.4%	81.6%	76%	74.1%
averaged point reduction for underestimation	76%	74.79%	69.45%	71.8%

Coverage by simplified polygons

	Freiburg	Baden-Württemberg	Germany
Accepts via underestimation	1035909	2976069	19637209
Rejections via overestimation	1964445	9108820	59858222
Regular accepts/rejections	251566	1219006	13687931
Relative coverage by simplifications	92.26%	90.84%	85.31%

Runtimes

	Freiburg	Baden-Württemberg	Germany	Europe*
Printing tags	3min	12min	88min	678min
Compute boundaries	5s	34s	9min	86min
Create RTree and DAG	6s	1min43s	58min	373min
Store and process OSM nodes	4min	24min	828min	22min

Runtimes

	Freiburg	Baden-Württemberg	Germany	Europe*
Compute way-clusters	13s	1min23s	14min30s	264min
Process and print OSM ways	7min41s	64min	2458min	325min
Process and print OSM relations	10min	79min	1741min	276min
Net Runtime	23min	184min	5237min	2021min

Thank you for your attention!