Comparison of matching methods of user generated and authoritative geographic data

E. Abdolmajidi1, J. Will1, L. Harrie1, A. Mansourian

Department of Physical Geography and Ecosystem Science, Lund University
Corresponding author: ehsan.abdolmajidi@nateko.lu.se

17th ICA Workshop on Generalization and Multiple Representation,
Vienna, Austria, 23rd Sept. 2014

Outline of the presentation:

- Related work
- Our algorithms
- Case Study
- Results
- Discussion
- Conclusions

Related work

Linear network matching (Doytsher et al. , 2001):

- Segment-based (Line-based)
  - Walter and Fritsch (1999)
  - Ludwig et al. (2011)
  - Koukoletsos et al. (2012)
- Node-based (Point-based)
  - Stigmar (2005)
  - Volz (2006)
  - Mustiere and Devogele (2008)

Segment-based Algorithms

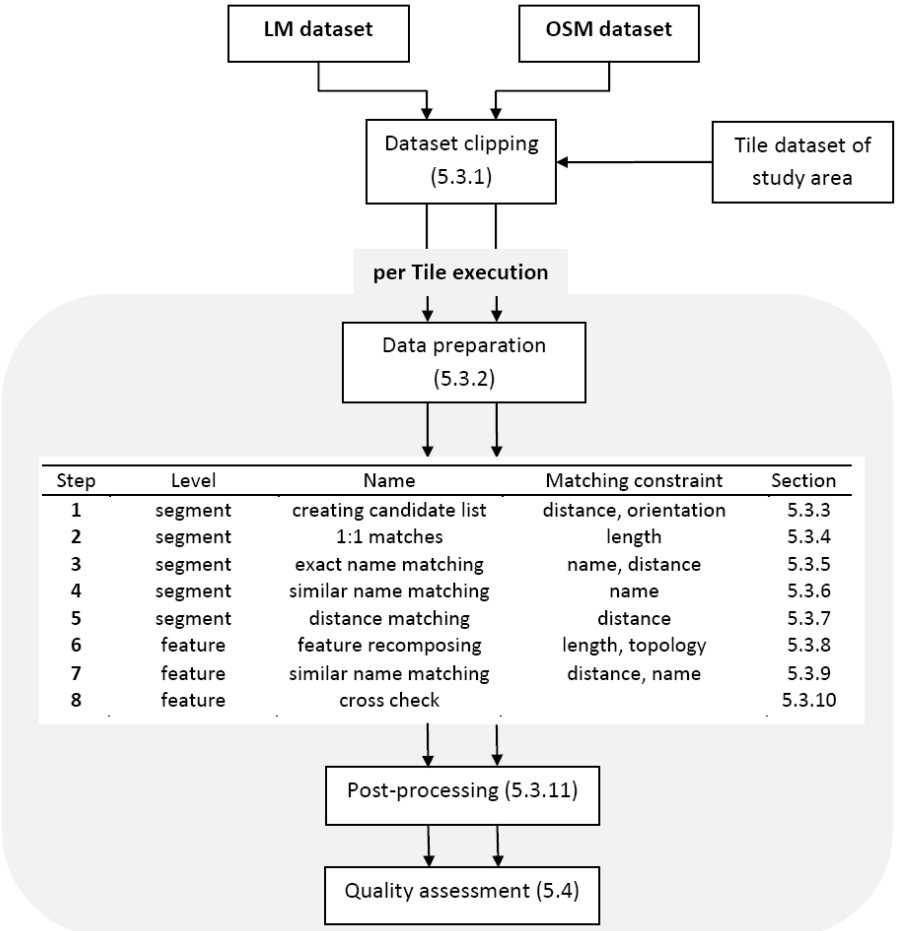
The segment-based algorithm is developed based on Koukoletsos et al. (2012) algorithm.

Matching steps at *segment Level*:

1. Buffering
2. 1:1 matching
3. Exact name matching
4. Similar name matching
5. Distance matching

At *feature level*:

6. Feature recomposing
7. VGI name similarity
8. Final check



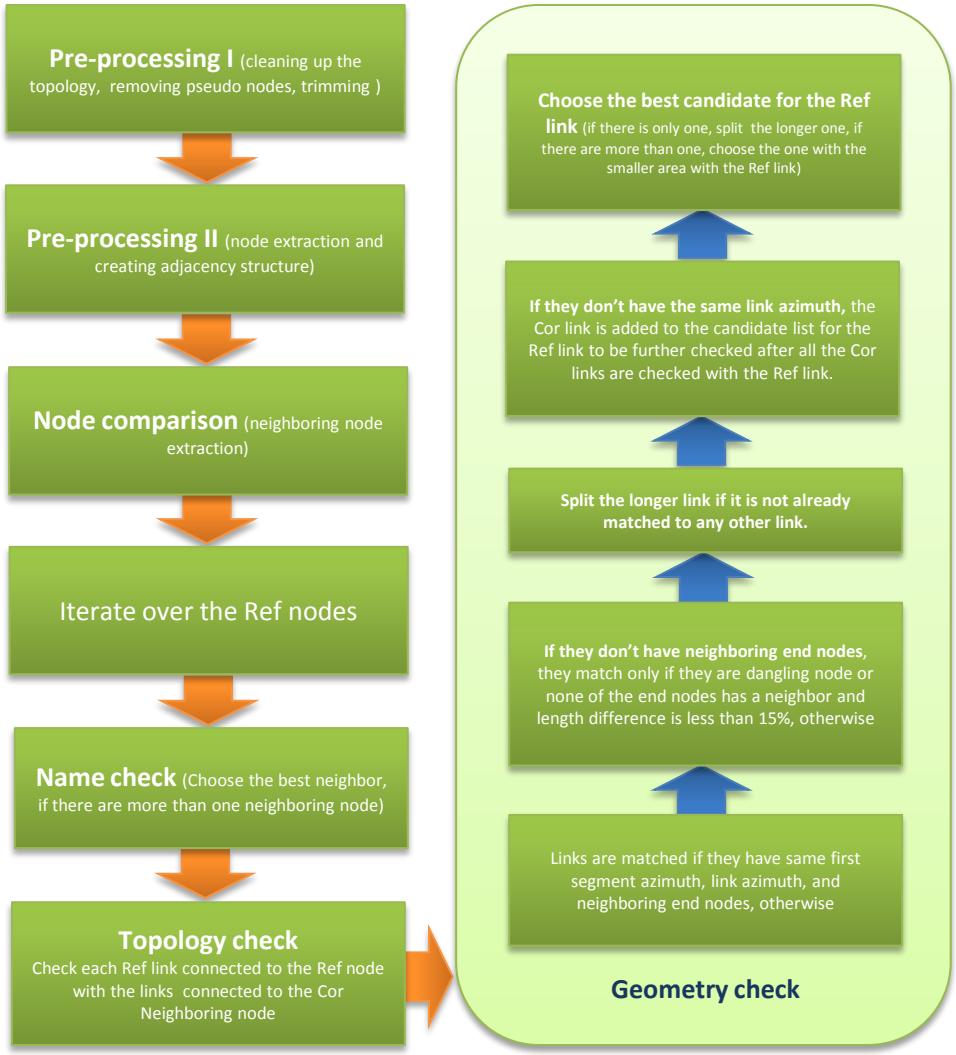
Step	Level	Name	Matching constraint	Section
1	segment	creating candidate list	distance, orientation	5.3.3
2	segment	1:1 matches	length	5.3.4
3	segment	exact name matching	name, distance	5.3.5
4	segment	similar name matching	name	5.3.6
5	segment	distance matching	distance	5.3.7
6	feature	feature recomposing	length, topology	5.3.8
7	feature	similar name matching	distance, name	5.3.9
8	feature	cross check		5.3.10

Node-based Algorithms

The node-based algorithm is developed from the scratch.

Matching steps:

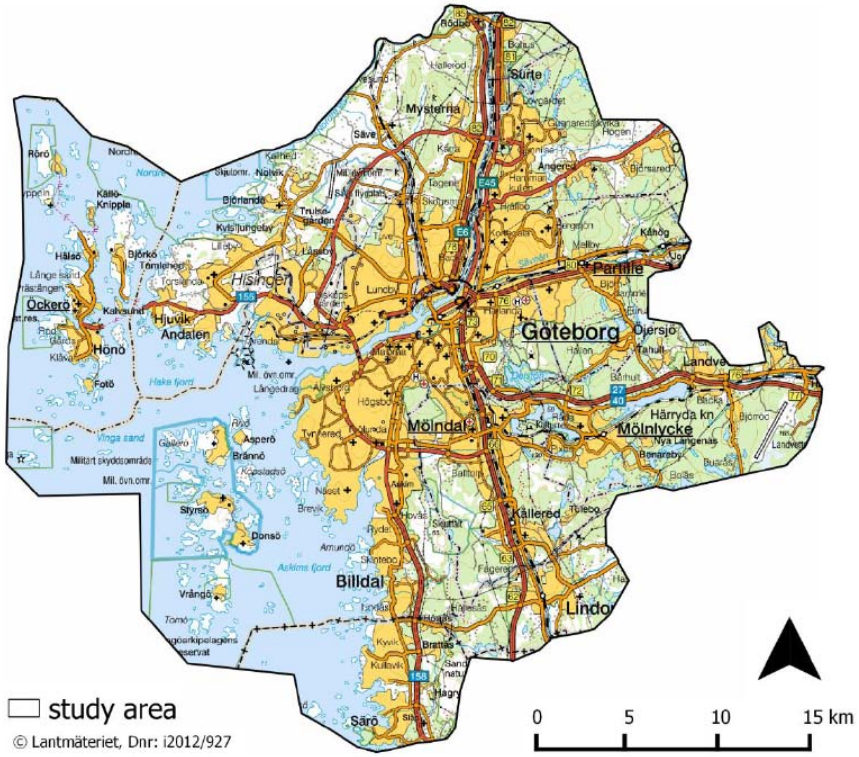
1. Node comparison
2. Name check
3. Topology check
4. Geometry check



Case Study

- Study area

- Gothenburg, Sweden
- Around 500,000 inhabitants



- Data

Authority data: real-estate map dataset from Lantmäteriet (LM)

VGI data: OpenStreetMap data (OSM)

Case Study – Implementation

- Both algorithms are developed in Python.
- The node-based was developed using Arcpy and Scipy libraries in the PyDev environment.
- The segment-based was developed using QGIS APIs in the python console of QGIS software.
- Spatial indexing: a) B-tree with depth of one in the segment-based algorithm (tiling), b) KDTree in the node-based algorithm.

Result - Matching

Segment-based algorithm

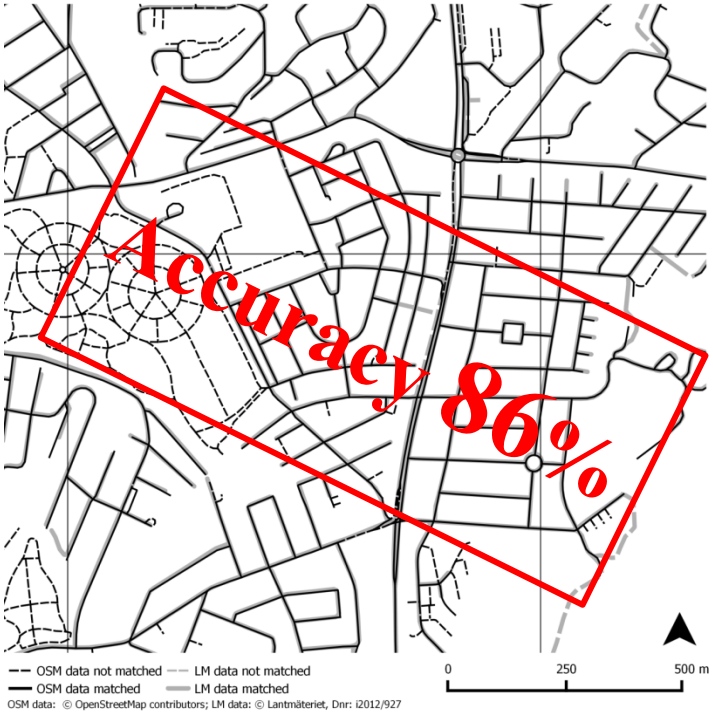
Dataset	Total length [m]	Length matched [m]
OSM	4596570	3550564 (77%)
LM	4691594	3800412 (81%)

Node-based algorithm

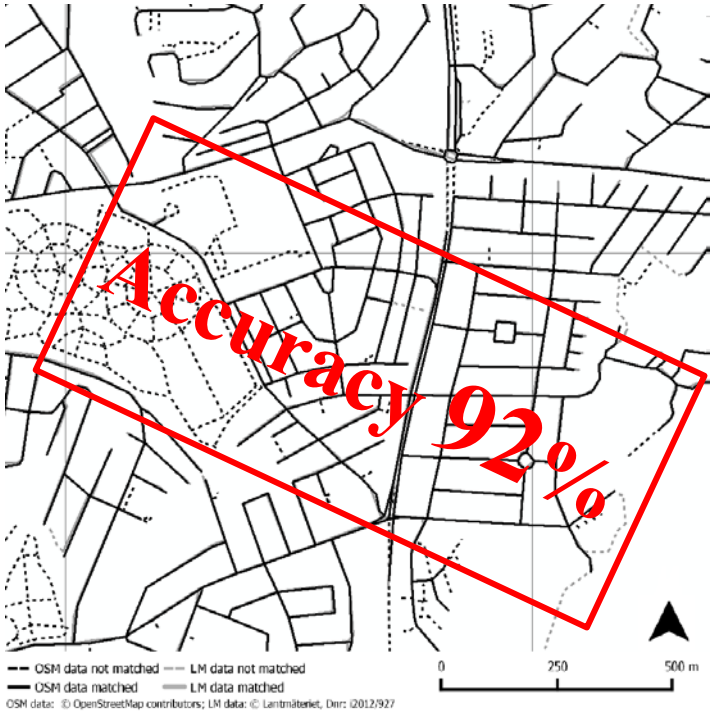
Dataset	Total length [m]	Length matched [m]
OSM	4489797	3561441(79%)
LM	4542694	3594120(79%)

Result - Accuracy

- We manually evaluated 10% of the study area.



Segment-based



Node-based

Result - Running time

Segment-based algorithm

Matching steps	Running time (Second)
Pre-processing:	10959.00
Buffering	2500.00
1:1 matching	38.00
Exact name matching	396.00
Similar name matching	141.00
Distance matching	300.00
Feature recomposing	1401.00
Feature name similarity	252.00
Final check	1514.00
Post-processing	612.00
Total	18113 (Almost 5 hours)

Node-based algorithm

Matching steps	Running time (Second)
Pre-processing I	37
Pre-processing II	161
Node comparison	50
Semantic, topology and geometry checks	180
Total	428 (Almost 7 Minutes)

Discussion – advantages and disadvantages

Segment-based

- Advantages:
  - The segment-based algorithm is a localized method around a segment which decreases the number of candidates.
  - the candidate segments are highly similar to the reference link. Hence, they need to less processing than the link candidates in node-based algorithm.

- Disadvantages:
  - The algorithm needs an essential preprocessing step in order to create the desired structure.
  - The algorithm uses buffering to create the candidate list, which is highly time-consuming.
  - It is a localized approach and therefore need broader view of the features under matching to better assign the pairs.

Node-based

- Advantages:
  - Node-based is a localized method around one node which substantial decreases the number of candidates.
  - Additionally, extracting the neighbors is very simple process.
  - The node-based is using the adjacency structure which enables to track some topological relations.

- Disadvantages:
  - needs an essential preprocessing step in order to create the desired structure (data format dependent).
  - In the node-based method, the candidate list was created based on the similarity of the neighboring nodes. Hence the similarity of the links connected to them is yet to be examined.
  - The node-based algorithm is sensitive to multi-neighboring.
  - It is a localized approach and therefore need broader view of the features under matching to better assign the pairs.

Discussion – Need for algorithm improvements

- The algorithm must be able to cope with:
  - heterogeneous geometrical representation
  - varying positional accuracy across the study area
  - complicated structures such as roundabouts and crossroads
  - data errors.

- Methods to improve the algorithm

  - To improve the node-based alg., the datasets should be enriched by graph-based and stroke-based methods before matching starts. These methods can help us to find the complex structures such as roundabouts and crossroads.
  - The varying positional accuracy can be improved by using multi buffering or cluster analyzing in order to detect the urban and rural areas.
  - Ontology and spatial ontology can be used as a data-model carrying useful information about structure, relation and classification of the features.

Conclusions

- Both the segment-based and the node-based algorithm had an accuracy of around 90% in the matching.
- The node-based algorithm is more time efficient and is therefore more suitable for huge datasets matching.
- The short-comings of the node-based can be covered by employing more processes with a few impact on the whole running time.