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)
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
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

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Total length [m]</th>
<th>Length matched [m]</th>
</tr>
</thead>
<tbody>
<tr>
<td>OSM</td>
<td>4596570</td>
<td>3550564 (77%)</td>
</tr>
<tr>
<td>LM</td>
<td>4691594</td>
<td>3800412 (81%)</td>
</tr>
</tbody>
</table>

### Node-based algorithm

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Total length [m]</th>
<th>Length matched [m]</th>
</tr>
</thead>
<tbody>
<tr>
<td>OSM</td>
<td>4489797</td>
<td>3561441 (79%)</td>
</tr>
<tr>
<td>LM</td>
<td>4542694</td>
<td>3594120 (79%)</td>
</tr>
</tbody>
</table>
We manually evaluated 10% of the study area.

**Segment-based**

Accuracy 86%

**Node-based**

Accuracy 92%
### Result - Running time

#### Segment-based algorithm

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

#### Node-based algorithm

<table>
<thead>
<tr>
<th>Matching steps</th>
<th>Running time (Second)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pre-processing I</td>
<td>37</td>
</tr>
<tr>
<td>Pre-processing II</td>
<td>161</td>
</tr>
<tr>
<td>Node comparison</td>
<td>50</td>
</tr>
<tr>
<td>Semantic, topology and geometry checks</td>
<td>180</td>
</tr>
<tr>
<td>Total</td>
<td>428 (Almost 7 Minutes)</td>
</tr>
</tbody>
</table>
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.