Matching authority and VGI road networks using an extended node-based matching algorithm

Ehsan ABDOLMAJIDI*, Ali MANSOURIAN, Julian WILL and Lars HARRIE

GIS Center, Department of Physical Geography and Ecosystem Science, Lund University, Sölvegatan 12, 22362 Lund, Sweden

(Received 18 March 2015; fi nal version received 27 June 2015)

The amount of volunteered geographic information (VGI) has increased over the past decade, and several studies have been conducted to evaluate the quality of VGI data. In this study, we evaluate the completeness of the road network in the VGI data set OpenStreetMap (OSM). The evaluation is based on an accurate and efficient network-matching algorithm. The study begins with a comparison of the two main strategies for network matching: segment-based and node-based matching. The comparison shows that the result quality is comparable for the two strategies, but the node-based result is considerably more computationally efficient. Therefore, we improve the accuracy of node-based algorithm by handling topological relationships and detecting patterns of complicated network components. Finally, we conduct a case study on the extended node-based algorithm in which we match OSM to the Swedish National Road Database (NVDB) in Scania, Sweden. The case study reveals that OSM has a completeness of 87% in the urban areas and 69% in the rural areas of Scania. The accuracy of the matching process is approximately 95%. The conclusion is that the extended node-based algorithm is suf fi ciently accurate and efficient for conducting surveys of the quality of OSM and other VGI road data sets in large geographic regions.

Keywords: geographic data; volunteered geographic information (VGI); OpenStreetMap (OSM); node-based matching; segment-based matching; pattern detection; Swedish National Road Database (NVDB)

1. Introduction

The amount of user-generated geographic data, the so-called volunteered geographic information (VGI) (1), has increased substantially in recent years. Thus, researchers have been investigating VGI resources in terms of quality and attributes (2–5). A common approach is to compare VGI with authority data. Most studies on VGI data quality compared the number of features and the total feature length of objects. However, there are also new studies that match individual features in the VGI and authority data sets (ADs) for the purpose of quality assessment (5, 6). In these studies, a key process is automatic feature matching. However, current matching algorithms have limitations in matching complex structures, such as complex junctions in road data sets.

The main aim of this study was to develop a more efficient and accurate algorithm for matching road networks with high matching quality, particularly for complex structures. To achieve the aim, we first conduct a comparison between the two main matching techniques: segment-based and node-based matching. Segment-based matching is based on a comparison of segments that compose a candidate list; then, the best match in the candidate list is selected by, e.g., statistical methods. Node-based matching comprises node matching and then an evaluation of geometrical and topological properties of the segments/links between the nodes. The most efficient technique is chosen and improved upon with further complex measures.

In the second part of the study, we increase the accuracy of the matching algorithm by improving the representation of topological relationships during the matching and by including pattern-detection routines for matching complex structures in the network. The final part of the study contains a case study on using our improved node-based algorithm to match OpenStreetMap (OSM) data to the Swedish National Road Database (NVDB) in Scania, Sweden. The study ends with a discussion and conclusions.

2. Related work

The literature on road-network matching provides several algorithms for finding the correspondence between features in two data sets. However, there is no consensus regarding how this matching should be performed. Road networks are very diverse in terms of geometric representation and availability of attribute data. A special problem in the matching process is the handling of complex junctions. A possible strategy that is not commonly used in the matching field is to include pattern detection as part of the matching. Therefore, we include a description of pattern detection in this literature.
review. The last part of the section addresses previous work on quality evaluations of OSM data.

2.1. Road-matching methods
Three properties are generally used to match the corresponding features in networks: geometric, topological, and non-spatial attributes. Using geometry only to compute matching pairs is not always sufficient. First, one road may have different geometries in different sources. Second, a polyline is represented by more than one point, and two polylines that represent the same road may have different numbers of segments. Consequently, there is no straightforward way to compare all shape points along two polylines or to determine whether the polylines are counterparts. Ruiz et al. have comprehensively reviewed the previous works related to geospatial polylines are counterparts. Ruiz et al. have comprehensively reviewed the previous works related to geospatial databases, in which the feature matching methods are also investigated as one of the steps (7). Here, we mainly focus on road-network-matching studies and investigate them from the matching technique perspective. Moreover, we also add more recent researches.

Java Conflation Suite (JCS) is an open source Java library which contains a road-network matching tool and uses geometric property of features (8). Its automated road-network matching of two data sets is based on a node-based approach. Within a maximum searching distance around each node of the reference data set, the best matching node of the other data set is selected based on distance. Edges are matched using the Hausdorff distance, edge length, and angle measurements. The algorithm splits matched edges when the length differences are too large to create more similar geometries. The JCS algorithm is adapted to increase the matching quality by adding three extensions specifically tailored for the involved data sets (9). In a preprocessing step, the geometry of the topological data set is simplified. Furthermore, the researchers add two steps to the original JCS algorithm to match yet unmatched segments by looking at their topological relationships first and then adding buffers around the remaining unmatched segments. The best matching pair within the buffer is finally chosen according to the distance measurement and the angular difference.

Another node-based matching algorithm is proposed by Volz (10). It starts with a rubber-sheeting algorithm to remove the geometric distortions between the data sets. Then, nodes with a high likelihood of correspondence between the data sets are identified and called seed nodes. By using the seed nodes as starting points, combined node and link matching is performed to identify one-to-one matches. Then, one-to-many matches are resolved.

Mustière and Devogele developed a node-based matching routine (including link matching) which particularly matched data sets with different levels of detail (11). The first step is to create graphs of the original networks. Then, pre-matching of nodes is performed by a distance criterion followed by pre-matching of links. These pre-matching steps result in lists of candidates. Finally, for each node in the less detailed data set, the matching is performed with the nodes in the other data sets while considering the consistency in the list of candidates.

Zhang and Meng suggested a network-based matching algorithm, which produced more robust results by adding more contextual information to the matching procedure (12). The proposed Delimited Stroke Oriented algorithm (DSO) applies a hybrid matching approach based on geometries and topologies, in which the semantic attributes are not yet considered. The reason is that the geometry and topology are always available or derivable for spatial objects, whereas semantic attributes are not. The proposed algorithm can lead to an on-the-fly matching process based on the principle of “what you can see is what you can match” (WYCISWYCM), i.e., it is a generic matching algorithm that can be utilized for various types of road networks.

Walter and Fritsch presented a segment-based method for matching road networks from different data sets and data models (13). The matching pairs are found by buffering the referent; from this buffer, potential matching elements in the other data set are identified. For the potential pairs, parameters such as angular difference, length, shape, and distance between elements are examined. The approach is based on statistical investigations between the two data sets.

Ludwig et al. developed a segment-based matching algorithm between OSM and a reference data set (5). This algorithm is partly based on the work of Walter and Fritsch (13). An initial list of matching pairs is created by buffering the reference data set with different sizes. All OSM data within a buffer segment are linked to the reference segment. For each list, similarities are calculated and ranked by considering the name and category attributes. Only the highest OSM ranks are retained as the final candidate list.

Koukoletzos et al. presented another segment-based matching algorithm which was evaluated for OSM and the Integrated Transport Network (ITN) data set from the Ordnance Survey (6). The data sets are divided into 1 km² tiles, which are computed separately to achieve better performance and to obtain a better representation of the heterogeneity of OSM in the results. The algorithm is based on a comparison of segments in which the correspondence measures are derived from the expected quality of the OSM data set. The algorithm produces robust results with low matching errors of 2% in urban areas and that of 3% in rural areas, approximately.

Yang et al. developed a link-based instance matching algorithm (similar to the segment-based algorithm, which buffers around the links instead of segments) using probabilistic matrix based on dissimilarities in shapes of the matching candidates (14). The probabilistic matrix will be then iteratively updated by the relative
compatibility coefficient of the neighboring candidate pairs until the matrix is globally consistent. The method finds the candidates using the buffer around links. It is able to achieve a robust matching result and to detect the 1:0 (Null), 1:1, and N:M pairs. Moreover, this algorithm is independent from the matching relationship. In other words, matching data set A to B leads to the same results as matching B to A. Then, Yang et al. proposed a recursive method based on hierarchical stable strokes considering the local similarities and global consistency of road features (15). Sch and Lipeck also proposed a matching algorithm considering the weighted similarity index calculated based on the geometric, semantics as well as the topology (16). The candidates are constrained by using must-match and cannot-match constraints, and a greedy approach is designed to find the local optimum instead of the global.

2.2. Pattern detection in road matching

Pattern and structure detection are widely used in several fields of geographic information science. Structures are the composition of objects in a map that carry a property, such as a shape, orientation, or functionality (17, 18). There are various methods for detecting such patterns in a network, and these methods are mostly developed or implemented in generalization applications.

Mackaness and Mackechnie presented a method for identifying road junctions using a combination of spatial clustering and graph theory (19). The authors mainly attempted to automatize the process of generalization and minimize human intervention during the process. Touya also enriched the data set by creating structural knowledge that can help detect geographical patterns and structures in the road networks (18). Savino et al. adopted an approach to detect road junctions by analyzing the cycles in the road graph and applying a morphological analysis for classifying the road junctions (20). Weiss and Weibel used the characteristics of the stroke-based method in their automatic generalization algorithm to detect roundabout patterns (21). Strokes are natural functional units within a network that represent the movement or flow path (22). Using stroke-based methods employing the good continuity for developing the strokes leads to isolated strokes for roundabouts (23).

However, in the field of matching linear network data sets, there are few works that utilize pattern detection to enrich the data set and facilitate/improve the matching process. Zhang and Meng benefited from the ability of the strokes in detecting roundabouts (using the DSO algorithm) (12). The authors also attempted to detect more patterns in their algorithm, such as dual carriageways, slip roads, and narrow passages. It seems that the road-network-matching literature fails to use the pattern detection. Thus, we decided to use pattern detection for data enrichment, which helps us employ a dedicated algorithm for those patterns.

2.3. Completeness of OSM

The evaluation of the OSM quality is mainly concentrated on completeness and positional accuracy (4). In our study, we are only interested in completeness. The (relative) completeness for linear features can be calculated by comparing the number of the summed lengths of all features of OSM and a reference data set. Haklay (4), Zielstra and Zipf (24) and Girres and Touya (3) estimated the completeness of OSM (only road objects) using the summed length. All of these studies found that OSM has a better completeness in populated regions or cities in general (25). Note that this methodology of evaluating OSM completeness is very sensitive to the choice of the reference data set.

A better evaluation of completeness can be achieved if the OSM has been matched to the reference data set. Then, the summed length of matched features divided by the total length of all features in the data set can be used for completeness measurements (6). This ratio describes the amount of data that can be found in the other data set. By using this approach, Koukoletssos et al. (6) found a completeness of OSM of 93% in urban areas (Greater London) and 59% in rural areas (West of Newcastle) (reference data set: ITN layer of MasterMap from Ordnance Survey). Using a similar approach, Ludwig et al. (5) found that the completeness of OSM in Germany varies from approximately 80% in inhabited areas to 50% in uninhabited areas (reference data set from Navteq). If only streets were considered, the completeness is as high as 97% in some populated areas.

3. Comparison of segment-based and node-based network matching

To compare the segment-based and node-based approaches for matching, one algorithm is implemented for each (Sections 3.1–3.2); then, a comparison is performed (Sections 3.3–3.5). The segment-based algorithm is similar to the algorithm introduced by Koukoletssos et al. (6). The node-based algorithm was based on ideas in (8–12, 26).

3.1. Segment-based algorithm

The segment-based algorithm has two levels: the segment level and feature level. Segments are the links between vertices, or a vertex and a node, while features are the links between two nodes. In other words, a feature consists of one or more segments. This algorithm first investigates the segments and then recomposes the segments to rebuild the features; it continues the matching process in the feature level. More detailed steps of the segment-based algorithm, apart from the data preprocessing (generalization and segmentation), are outlined below (see Ref. (27)).
Segment level:

1. Buffering: For each segment in AD, a list of possible matching segments in VGI located within a buffer around an AD segment is found. These candidates should also have a particular orientation.

2. 1:1 matching: If the list has only one candidate and its length does not exceed three times the corresponding AD segment, then the pair is considered matched.

3. Exact name matching: The road name of the AD segment is matched to the VGI candidates. If only one segment has an exact name, then this pair is regarded as a match. If several segments are found, then the closest pair is chosen as a match.

4. Similar name matching: The most similar name between VGI candidates and AD segments is identified as the road names are not always correctly spelled or might be abbreviated, particularly in VGI.

5. Distance matching: AD and VGI segments are matched based on the distances between possible matches regardless of their road name attributes.

Feature level:

1. Feature recomposing: Segments along with their matching information are transferred to the feature level. A feature is then considered matched if the matched segments constitute more than half of its length. The corresponding feature(s) are selected based on the length proportion matched to the feature(s) in the other data set.

2. VGI feature name similarity: The name similarity of the non-matched VGI feature with the AD features located within the buffer (which is twice the GPS accuracy of 10 m) around the VGI feature is determined.

3. Final check: If matching information on non-matched features in one data set exists in the other data set, then the features are assigned to their corresponding features in the other data set where they are listed as a match.

3.2. Node-based algorithm

The node-based algorithm involves the following four important checks besides preprocessing (generalization and data structuring):

1. Node comparison: This is an inherent step of node-based algorithms in which a list of neighboring VGI nodes is found for each node in the AD.

2. Topology check: Topology, the main component of the algorithm, helps verify all of the links connected to the AD node with each link connected to its neighbors (11, 26). This topological relationship is called a composition relationship between a node and lines connected to it.

3. Geometry check: To find matches between links, several conditions must be considered. The measures such as first segment direction, link direction, and length are taken into consideration.

4. Name check: If both links under investigation have names, then their names are compared. This outcome mainly occurs when an AD node has more than one neighboring node; the algorithm should decide which node is correct.

This algorithm initially identifies corresponding nodes and then matches the corresponding links connected to the nodes using topological, geometric properties and attribute information.

3.3. Material and methods of the comparison

The comparison used sample data that covered Gothenburg, Sweden, and the surrounding region (861 km²). We used a real estate map from Lantmäteriet (LM) as the AD. The data set is produced at a scale of 1:5000–1:20,000, and its positional accuracy is specified as less than 2 m (standard deviation). The OSM data (from 16 April 2014) for the same area were used as the VGI. For consistent geometry between the two data sets, all OSM features were split at intersections in the preprocessing steps.

The segment-based algorithm was implemented as a plug-in to PyQGIS using Python and the QGIS API. The node-based algorithm was mainly implemented using the ArcPy package. The latter implementation utilized the spatial module of the SciPy package (28) for creating the KD-tree index of the nodes.

3.4. Result of the comparison

The comparison was based on (1) the rate of correctly matched and unmatched links and (2) computational efficiency. Table 1 shows the results of the segment-based and node-based matching methods for the entire study area with almost 80% matched features in both data sets. The matching rate depends on the completeness of OSM and LM in the study area. In the central part of Gothenburg, OSM includes more features, while LM has a better coverage in the other parts of the study area (see Ref. (29)).

Each algorithm was run 10 times, and the average execution time was measured. The execution times are represented in Tables 2 and 3 for every step of each algorithm in seconds. The final step of the node-based matching algorithm includes the topology, geometry, and name checks for measuring the similarity of features (steps 2–4 in Section 3.2).
Table 1. Matching results of the segment-based and node-based algorithms for the study area.

| Matching method         | Data set | Total length (m) | Matched length (m) | Matching accuracy (%) |
|-------------------------|----------|------------------|--------------------|-----------------------|
| Segment-based algorithm | OSM      | 4,596,570        | 3,550,564          | 77                    |
|                         | LM       | 4,691,594        | 3,800,412          | 81                    |
| Node-based algorithm    | OSM      | 4,489,797        | 3,561,441          | 79                    |
|                         | LM       | 4,542,694        | 3,594,120          | 79                    |

Table 2. Efficiency assessment for the segment-based algorithm.

| Segment-based algorithm | Running time (s) |
|-------------------------|------------------|
| Preprocessing           | 10,959           |
| Buffering               | 2500             |
| 1:1 matching            | 38               |
| Exact name matching     | 396              |
| Similar name matching   | 141              |
| Distance matching       | 300              |
| Feature recomposing     | 1401             |
| VGI feature name similarity | 252              |
| Final check             | 1514             |
| Post-processing         | 612              |
| Total running time      | 18,113 (3 h)     |

Table 3. Efficiency assessment for the node-based algorithm.

| Node-based algorithm | Running time (s) |
|---------------------|------------------|
| Preprocessing I     | 37               |
| Preprocessing II    | 161              |
| KD-tree indexing of the nodes | 5               |
| Node comparison     | 50               |
| Geometry, topology, and name matching | 180           |
| Total running time  | 433 (7 min)      |

Substantial differences occur in the preprocessing steps of the two algorithms (Tables 2 and 3). In the preprocessing step, data structuring requires the most time. The segment-based algorithm must break down all of the features into the segment, whereas the node-based algorithm only extracts the nodes of a feature. The second highest time-consuming step is the Buffering step in the segment-based algorithm, which is the counterpart of the node comparison in the node-based algorithm. Buffering is an expensive process compared with the node comparison for selecting the candidates. Nevertheless, applying either method for a large data set without deploying any indexing method is inefficient. In this regard, a KD-tree index was used for indexing the nodes in the node-based algorithm. In the segment-based algorithm, tiling was used as Koukoletsos et al. suggested (6).

For a reliable judgment, we also performed an accuracy test. For that purpose, 10% of the features were randomly selected and assessed for both algorithms. The accuracy assessment showed that the accuracy of the node-based algorithm is 92%, and the accuracy of the segment-based algorithm is 86% in the study area. This implies that 92%, respectively, and 86% of the matched and unmatched features are correctly assigned.

3.5. Discussion of the comparison

Each strategy has advantages and disadvantages. The segment-based approach selects the segments that are within the buffer around a reference segment. Only a limited number of candidates are expected to be similar. However, the node-based method focuses on the neighboring nodes of two data sets. This approach is computationally simpler than buffering around a link or segment. However, gaining such simplicity by choosing the neighboring nodes without considering the condition of the links connected complicates the node-based approach when detecting the correct final matches. Both methods need specifically structured input data, which makes the preprocessing step necessary. In the preprocessing step, the segment-based method requires tiling the data set and splitting the links into segments, while the node-based approach also needs to extract the nodes of two data sets. As the node-based algorithm focuses on the neighboring nodes, the result can be affected if there are clusters of nodes. Thus, choosing the best neighboring node is difficult. Nevertheless, a node is a starting point for this matching approach; hence, a better node availability corresponds to more possible matches.

The accuracy assessment showed that the node-based algorithm performed somewhat better than the segment-based method. The evaluation also showed that the algorithms have similar limitations. The matching errors in both cases were caused by heterogeneous geometrical representations, varying positional accuracies across the study area, different representations of complicated structures (e.g., roundabouts, bridges, and multi-carriage roads), and data errors. In the node-based algorithm, another problematic case was ring-shaped links. A ring is a link that has the same start and end nodes. These cases were excluded from the current version of the algorithm because their link azimuth is not defined. Another reason for matching errors was the poor quality of the attribute information in the data sets (i.e., the road names). Note that the accuracy obtained by the segment-based algorithm is worse in this study than what Koukoletsos et al. (6) obtained with a similar algorithm (Section 2.1), mainly because the data sets differed.

The main difference between the methodologies is computational. Although one cannot truly compare the execution time of the algorithms (e.g., because different programming packages and indexing techniques are used), the node-based approach is computationally more efficient than the segment-based approach. Creating and comparing buffers are computationally expensive.
processes. Table 4 summarizes the above comparison between node-based and segment-based algorithm in terms of data structuring, computational complexity, link matching, and matching accuracy.

We want an accurate and efficient method for our matching procedure. The comparison shows that the node-based algorithm is more efficient and somewhat more accurate than the segment-based algorithm. Furthermore, we cannot see any major differences in the possibility of enhancing the accuracy of the algorithms (e.g., by using pattern detection). Therefore, we selected the node-based algorithm for the next step in our study.

4. Extended node-based matching algorithm

The node-based algorithm (as described in Section 3) has shortcomings at complex junctions (where it is difficult to identify corresponding nodes) and roundabouts. To improve the accuracy of the algorithm, we add features for identifying the correct link using topological reasoning and introduce specific pattern detection and matching. In this section, we describe the extended node-based algorithm in detail and note the improvements particularly. The section is organized according to the workflow of the algorithm (Figure 1).

In the description, we use the terms reference nodes and reference links for the nodes/links in the AD. The corresponding names in the VGI data set are target node and target link.

4.1. Preprocessing

Before the matching process begins, the data must be cleaned up. A main task of the cleaning procedure is the removal of pseudo-nodes (i.e., nodes that have exactly two links); nodes are removed by employing a dissolve function. The elimination of pseudo-nodes simplifies the data set and skips the unnecessary work of the algorithm. Moreover, it is essential to remove the pseudo-nodes for correct pattern detection of the roundabouts.

4.2. Step 1: complex structure detection

As the node-based algorithm is a localized approach that focuses on the nodes, the matching may be misled when the structure is complex and the representations have different levels of detail. For good matching of these complex features, the features must be treated separately in the matching process. In this study, we develop a method to address one such complex structure: the roundabout. Typical representation of roundabouts is illustrated in Figure 2. In Figure 2, the roundabout in one data set is only represented as an intersection in the other data set. The treatment is divided into two phases: complex structure detection and complex structure matching (Steps 1 and 4 in Figure 1). The pattern-matching technique is inspired by Ref. (18).

In some cases, roundabouts can be identified using attribute information in the databases. However, in many cases, this information is lacking or is of bad quality. Then, we need to detect the roundabouts using geometric and topological information. For the latter case, we develop a method to detect the roundabouts where several conditions are used. As the main perception, a roundabout is expected to be round in shape; hence, the first condition is detecting round shapes as the preliminary condition based on the normalized roundness index (NRI) (18, 30):

![Figure 1. The general flowchart of the extended node-based algorithm.](image-url)
The NRI is equal to one if the structure is a circle. However, because a roundabout is not represented as a circular shape in the data sets, a threshold is required. A test is performed on empirical data to determine that an NRI of 0.9 has a low commission error (errors are further discussed in the case study section). The other conditions for detecting roundabouts are as follows:

- Connected to at least three links.
- Not have links longer than 100 m.
- Not have an area larger than 4000 m².

The implementation of the roundabout detection is performed in three steps. In the first step, polygons are created for all faces (polygons) that are included in the road network (assuming that the network is a planar graph). In the second step, all of the faces are evaluated against the conditions above. Finally, in the third step, the centroid of the roundabout is stored, and an attribute is added to the links describing the roundabout they belong to. The centroids of roundabouts are used in the node comparison to find the possible corresponding roundabout. If there is no corresponding centroid around reference centroid, then the corresponding nodes are checked to find the corresponding intersection in the other data set. The link information attached to the roundabouts is used in the roundabout-matching process for final link matching.

4.3. Step 2: Data Structuring

In this step, the data (of both networks) are organized in adjacency lists to facilitate rapid retrieval of the data (31). In this data structure, the links contain information on the feature ID of their end nodes, the topological relations, and attribute information. To achieve a high-performance search of the neighboring nodes, the node list of each data set is spatially indexed using a KD-tree (32).

4.4. Step 3: Node Comparison

The comparison of the nodes in the two data sets could be based solely on the distance between them as follows: (1) using a fixed number of neighbors (growing buffer) or (2) using a fixed neighboring distance (a distinct threshold). The second approach implies that a single node in one data set could be linked to zero, one or several nodes in the other data set. We choose to use the second method with a threshold of 10 m. This distance is selected based on the average accuracy (α) of GPS absolute code measurements frequently used by VGI data collectors (6). In this step, the centroids of the roundabouts are also involved in the comparison. They are compared with each other to find their corresponding roundabouts. They are then compared to the nodes if there is no neighboring centroid for them. The output of this step is lists of candidate target nodes or centroids for each reference node or centroid.

4.5. Step 4: Complex Structure Matching

In this step, the detected roundabouts are matched. Ideally, the feature should be a roundabout in both data sets; hence, their centroids must be neighbors. Otherwise, if the feature is a roundabout in one data set and only ordinary junctions in the other data set (as in Figure 2), then the roundabout’s centroid must be neighbor of a node. In this case, all the links composing the roundabout are matched to the node in the other data set (a hyper-node). If there is no neighboring node, then the roundabout will be treated as normal links in the main matching algorithm.

In a case that the centroid of the roundabouts has a neighboring centroid, the roundabout–roundabout matching is deployed. Note that the roundabout might have different levels of detail in the data sets, as shown in Figure 3. Figure 3(a) and (b) depict a reference roundabout (solid line) that is more detailed than its corresponding roundabout (dashed line).

The roundabout-matching algorithm is able to distinguish between the two cases shown in Figure 3 by checking the following:

(a) The number of nodes on the roundabout. The number of links connected to them is also checked and should be 3.
(b) Topology of the links exiting the roundabout. If the links mutually share a node outside the roundabout, then Figure 3(a) has occurred. If the links share a link in between and then share a node, Figure 3(b) has happened.

The exiting links from the roundabout are then considered as an entry and exit of one street and if the target
roundabout has lower level of detail as shown in Figure 3, then algorithm knows that the two target links should be connected to one reference link. For finding the corresponding link in the target data set, the node (shown with red color in Figure 3) between Node 1 and Node 2 is found and the exiting links will be matched. The link between Node 1 and Node 2 is also matched to the target node (shown by red color) between them. The same process is applicable if the target roundabout is more detailed than the reference roundabout.

4.6. Step 5: node iteration
In this step, the algorithm iterates over each reference node. If there is no candidate target node, then the reference node is saved in a list of unchecked nodes. This list is important due to the updating nature of this algorithm in which these nodes could be matched in a later stage. If there is exactly one candidate target node, the algorithm proceeds to Step 3, the link comparison. Finally, if there are two or more candidate target nodes, then the algorithm must more thoroughly investigate the nodes, as done in Step 7.

4.7. Step 6: link matching
In this step, topological, geometric, and attribute information is used to select the corresponding links in the data sets. The starting point is the topological information (stored in the adjacency list). By using this information in conjunction with the lists of corresponding nodes, we can find candidates of corresponding links. The first candidate check is based on the following requirements:

1) The azimuths of the first line segments within the links are adequately close. Based on an empirical study of the data, we set the threshold to 45°.

2) The azimuths of the links are adequately close, i.e., is within the angular tolerance (as shown in Figure 4). This is computed based on the expected accuracy of the nodes (which is equal to the parameter \( \alpha \) above) and the length of the reference link (\( \beta \)) according to:

\[
\text{Angular tolerance} = \varphi = \frac{180}{\pi} \tan^{-1} \left( \frac{\alpha}{\beta} / 2 \right)
\]

The continuous line is the reference feature, the dashed line is the case of compared features with the GPS accuracy of \( \alpha \), and \( \varphi \) is the angular tolerance. The idea is based on Ref. (6).

3) The end nodes of the links are within \( \alpha \) meter distance from each other.

The requirements are checked sequentially. If all three requirements hold, then the links and the end nodes are matched. If the first requirement holds but not the second and third ones, the following action is performed. The target link is added to a new candidate list created for that reference link for further investigation after the other candidate links are checked and no match is found for the reference link. If no better candidate is found, then either of these conditions holds:

(a) The positional accuracy is less than 10 m at the end node.

(b) One link is shorter than the other because its network has more details, e.g., an intersecting alley.

To determine whether the condition (a) holds, we evaluate the end nodes. If the end nodes are not the neighbor of other nodes and their links can be matched, then the problem is likely only the positional accuracy of the node; hence, the links are matched.

To check for the condition (b), i.e., if one link is longer than the other because of varying levels of detail, the longer link is broken close to the end node of the other link. Then, an investigation follows by checking their new azimuth. The new links are matched if the new azimuths are in tolerance threshold. To maintain the connectivity of the matched network, the algorithm recursively matches the new links until there is no new

Figure 3. Complex roundabout matching of the algorithm links exiting the roundabout, in case (a), mutually share a node outside the roundabout, and in case (b), share a link in between and then share a node.
link to be matched. If the links are dangling, then they are matched without splitting. Note that this process is not always successful. If the algorithm cannot determine that they are similar, the longer link is then rebuilt and the target and reference links are considered unmatched.

If the geometric and topological information cannot resolve the problem of matching the links, then the attribute information is investigated. This investigation contains a check of whether the names of the links are the same in the two data sets (thus, both must be named). Note that the names of the links, particularly in the OSM data set, are likely misspelled or abbreviated. Therefore, we need a similarity measure in the comparison. A commonly used name-similarity measure is the Levenshtein distance \((33)\). This name similarity, however, does not take the lengths of the strings into account. A better index is calculated using the normalized Levenshtein distance \((34)\) (Equation (3)). Before the calculation, the strings are converted to upper letters to only investigate spelling mistakes. The normalized Levenshtein distance value is subtracted from one.

\[
\text{Normalized Levenshtein Distance} = \frac{\text{Levenshtein distance}(\text{Road\_name1}, \text{Road\_name2})}{\text{maximum(} \text{length Road\_name1}, \text{length Road\_name2})}
\]

There are some complicated cases that sometimes occur in the matching of links. One such case is whether the end node of one of the links has already been matched to another node. In that case, the best node should be selected in the final result, as discussed in the next section.

4.8. Step 7: best matching node selection
A complicated situation for this algorithm is when there is more than one candidate target node for a reference node. The node with the correct match must be determined; however, this seemingly simple decision needs a complex investigation because the problems might be caused by various geometrical and topological situations, as depicted in Figures 5–7. In Figure 5, the reference links are in solid and the target links are dashed. The reference node has two candidate nodes with the same number of links. However, in Figures 5(a) and 5(b), the link names identify the correct match (using the normalized Levenshtein distance of Equation (3)) because the geometries of the links connected to the candidate target nodes are similar. In Figure 5(c), the geometry of the links can help find the best node. If the available information is insufficient to determine the final match of the nodes, the nearest of the two nodes is selected for matching.

Figure 6 illustrates more complicated cases. In Figure 6(a) and (b), the data sets represent the same junction with different levels of detail. In these cases, the reference node is matched to the link shown in red (a hyper-node), and the rest of the links are correctly matched. Notably, before assigning any of the links to each other, their overall geometries are checked to avoid any mistakes, as could be the case in Figure 6(c) and (d). If these two cases occur, then the link names may help find the best neighbor; otherwise, the nearest node is selected as the match.

Figure 7 illustrates another case in which the junctions in the data sets are represented in different levels of detail. In this case, the number of links connected to the reference node is more than 4 and the summed number of links for the target nodes is two plus the links for the reference node. Therefore, the problem is likely related to a short link between the target nodes (the red link in Figure 7). A situation like this is addressed by investigating whether the links from all of the nodes can be matched. If so, then the reference node is matched to the joint link.

5. Case study
5.1. Study area and data
The study area is Scania, which is the southernmost province of Sweden, with a total area of 11,034 km². The study area was divided into urban and rural areas (as shown in Figure 8). Urban areas are defined as the
geographic extent of cities with at least 15,000 inhabitants (using data from Statistics Sweden)

The reference data set is the National Road Database of Sweden, NVDB (35). NVDB is produced at scale of 1:5000–1:20,000; its positional accuracy is specified as less than 2 m (standard deviation). NVDB contains all roads for vehicles in Sweden (including minor forest roads), as well as some cycle paths. NVDB is provided in the geodetic reference system SWEREF 99 TM, which is a UTM 33 projection of the Swedish implementation of ETRS 89.

The target data set is OSM from 16 April 2014. The OSM road data are not required to have a node at intersections. To make the geometries between the two data sets more similar, all OSM features are split at intersections. The OSM data are transformed from geographic coordinates in WGS 84 to SWEREF 99 TM.

Figure 9 illustrates the number of features and the length of OSM and NVDB in the study area. This figure indicates a general pattern that OSM has generally better completeness in urban areas than in rural areas (3, 4, 7, 21). We also note that the features (links) in NVDB are somewhat longer than those in OSM on average.

5.2. Implementation and computing environment

The extended node-based algorithm was implemented in Python using the ArcPy and SciPy packages (28). The test was run on a PC with Intel Core i7 CPU (3.4 GHz), 16 GB RAM, and 64-bit Microsoft Windows 7 SP1. The data sets were in shape file format so we needed to convert them to a planar graph structure for creating the adjacency list.

5.3. Results

The extended node-based algorithm was applied to the entire study area. Figure 10 shows the overall matching result: 77% of the features in NVDB are matched to 70% of the features in OSM. Also, 69% of the length of the NVDB network is assigned to 80% of the length of the OSM network. Figure 11 shows the same information divided into urban and rural areas. Of special interest is the percentage of features in NVDB that is matched to OSM; this figure is a good estimation of the completeness of OSM. From Figure 11, we estimate that OSM has a completeness of 87% in the urban areas and 69% in the rural areas in Scania.

The accuracy of the matching result was evaluated manually. The evaluation was conducted for urban and rural areas in each data set. The accuracy values, given in Table 5, are the ratio of the summation of the length or the number of features correctly matched and unmatched to the summation of the length or the number of all features in that data set (in percentage). A reference feature is correctly matched to target feature(s) if it has corresponding features in the target data set (1:1 or 1:N relation), and a reference feature is correctly unmatched to any target feature if there is no corresponding feature for it in the target data set (1:0 or Null relation).

For the efficiency assessment, the improved algorithm was applied to the study area for 10 times and the
The average running time for each step was calculated (Table 6).

The sensitivity analysis was also applied on the extended algorithm regarding the GPS accuracy and the angular tolerance. The algorithm was tested with 5, 10, and 15 m of the GPS accuracy ($\alpha$), and 40°, 45°, and 50° for the angular tolerance. The results shown in Table 7 reveal that the method is quite insensitive to the parameter values. The accuracy values are based on the ratio of the correctly matched and unmatched links to whole links in a data set.

To evaluate the influence of the roundabout detection, the algorithm was run twice; once without the roundabout detection and once with the detection. First, the results of the pattern detection are shown in Table 8.

Table 5. The accuracy of the extended node-based matching algorithm in urban and rural areas for the NVDB and OSM data sets.

| Matching area (matching item)          | NVDB (%) | OSM (%) |
|---------------------------------------|----------|---------|
| Rural area (length)                   | 94.9     | 94.7    |
| Rural area (number of features)       | 94.8     | 95.2    |
| Urban area (length)                   | 93.6     | 97.5    |
| Urban area (number of features)       | 94       | 95.2    |

Table 6. Running time for each step of the extended node-based matching algorithm.

| Step                            | Running time (s) |
|---------------------------------|------------------|
| Preprocessing                   | 3564             |
| Roundabout detection            | 7027             |
| Data structuring                | 1175             |
| Indexing                        | 5                |
| Node comparison                 | 237              |
| Roundabout matching             | 1                |
| Matching                        | 1265             |
| Total running time              | 13,274 (4 h)     |

50° for the angular tolerance. The results shown in Table 7 reveal that the method is quite insensitive to the parameter values. The accuracy values are based on the ratio of the correctly matched and unmatched links to whole links in a data set.

To evaluate the influence of the roundabout detection, the algorithm was run twice; once without the roundabout detection and once with the detection. First, the results of the pattern detection are shown in Table 8. The accuracy of the pattern-detection process used in the algorithm is 93.5 and 72.8% for NVDB and OSM, respectively. The pattern detection accuracy is the ratio of correctly detected roundabouts to the detected roundabouts in Table 8. To assess the roundabout detection process, the result was compared with the roundabout tags in OSM data and the *Cirkulationsplats* layer in NVDB. Table 8 also presents details on the omission and commission errors in both data sets.

The omission error refers to the roundabouts that the detection process missed. There are four major reasons for these omissions (Figure 12):

1. The roundabout is not round (NRI ≤ 0.9).
2. The area of the roundabout is larger than the threshold.
3. The number of entries to the roundabout is less than three.
4. The roundabout is composed of more than one face.

The roundness, area, and number of entries are the thresholds used to reduce the commission error. Hence, regulating those constraints can change the omission error. Note that changing the constraints to improve the
omission error may increase the commission errors. However, the fourth reason cannot be solved simply by using a roundness measure (NRI). The reason for this shortcoming is that the input data sets are in shape file format (i.e., not topologically structured). Therefore, we cannot detect if roads are connected in reality. Hence, the data sets must be converted to a planar graph as we needed to create the adjacency list. If we had a topologically structured input data, NRI was able to detect multi-face roundabouts as well.

In contrast, the commission errors refer to roundabouts that are detected as roundabouts by the pattern-detection process but are not classified as roundabouts in the NVDB and OSM data sets.

Common reasons for the commission errors are weak roundness or large area thresholds. Furthermore, an investigation of the commission error showed that 50 and 75% of the commission errors in the NVDB and OSM data sets, respectively, are features that are not classified as roundabouts in both data sets. Hence, the classification of these features is somewhat uncertain.

After the roundabout detection, the roundabout-matching process was able to correctly match 17 roundabouts with geometries similar to those shown in Figure 3 with an accuracy of 100%. Moreover, 28 and 25 roundabouts in the NVDB and OSM data sets, respectively, were matched to nodes with an accuracy of approximately 90%. Table 9 shows the accuracy of matching roundabouts without and with pattern detection in the refined node-based algorithm. The accuracy is the ratio of correctly matched and unmatched links to the whole links in the roundabouts.

As Table 9 illustrates, the accuracy of matching roundabouts has significantly increased. Investigating the overall accuracy of the matching algorithm, however, showed the overall accuracy improved less than 1%. That is expected as the number of links (approximately 300,000) in our data sets is drastically bigger than the number of links that build the treated roundabouts (Figure 3) in this study.

As Figure 13 shows, the link matches without the pattern detection has lower quality than matching roundabouts using pattern detection and a dedicated matching algorithm. The aligned links in former case may not be useful in data conflation and navigation representation as the connectivity between links is not retained in many cases. It occurs as the matching algorithm does not consider the functionality of the links that build up a pattern (i.e., two links may be two lanes of one street connected to the roundabout). Using pattern detection, however, allows us to retain the functionality of the pattern by identifying the difference between two roundabouts and deploying a proper treatment for that special case. Figure 13 shows the matching of a roundabout without and with pattern detection. Nevertheless, the matched links in Figure 13(a) are still considered correct as the corresponding links belong to the corresponding roundabout and the consecutive links are meaningful.

5.4. Discussion of OSM completeness
The overall matching result (Figure 10) shows that NVDB has a larger percentage of matched features,

![Figure 12](image-url) Examples of roundabouts in data sets that are not detected (omission errors). a) a roundabout composed of more than one face, b) a roundabout that is not round (NRI ≤ 0.9).
while OSM has a larger percentage of the total length. Mostly long links in OSM are matched to NVDB. Furthermore, fewer links in NVDB constitute the greatest total length of the NVDB network (Figure 9).

As a definition of completeness, a higher detailed road network is considered more complete. We distinguish between the two notions of more complete and more detailed road networks, which can augment the previous definition. A more detailed road network has more links with short lengths, which may represent less important features in a road network, such as complex road junctions, narrow passages, or driveways. A more complete road network has few links with longer lengths, such as main roads and highways.

Figure 11 shows that the percentage of matched links in urban areas for NVDB (87%) is considerably higher than that for OSM (54%). Thus, OSM in the urban area is highly detailed. However, in the rural areas, the situation is the opposite because of the lower degree of completeness of OSM in rural areas. Comparing the results of the length and number of matched links in rural and urban areas (Figure 11) reveals the same discrepancy seen in Figure 9. A smaller discrepancy in the number than in the length of the links in rural areas shows that, in rural areas, the OSM data set is missing long links (roads) (i.e., it is incomplete). However, in urban areas, a larger discrepancy in the number than in the length of matched links means that OSM has many short links compared with the NVDB data set (i.e., it has higher detail).

In this study, we estimate the completeness of OSM based on a matching approach. By using this approach, we estimate that OSM has a completeness of 69% in the rural areas and 87% in the urban areas in Scania (Figure 10). If we had only used the total length of the features in the data sets to estimate the completeness, we would have obtained OSM completeness of 81% in rural areas and 100% in urban areas (the total length of OSM is longer than that of NVDB, but by definition, the completeness can never be above 100%). We argue that estimating the completeness of OSM only based on the total line length is inappropriate.

5.5. Discussion of the quality of the extended node-based algorithm

The results of the state-of-the-art implementations of the node-based and segment-based methods revealed the capability of the node-based algorithm for improvement. Hence, the node-based algorithm was extended by involving more advanced methods in the matching steps (Steps 6 and 7 in Figure 1) and by employing complex structure detection with a specific matching process.

The added complexity of the algorithm increases the accuracy of the algorithm but deteriorates the efficiency of the algorithm. This finding is particularly true if we use the geometrical and topological method to identify the roundabouts (Table 6). In general, we can conclude that attaining a higher accuracy requires that the
algorithm must consider more special cases that will degrade the efficiency.

The geometrical and topological detection of roundabouts could be used in those cases in which there is no reliable attribute information that defines the roundabouts. In OSM and NVDB, such attributes exist, although they are not very good quality. Therefore, we attempted to provide a generic method for dealing with these complex structures. Note that the efficiency of the current implementation of pattern detection can be improved, for example, by defining clusters. A density-based cluster algorithm, such as DBSCAN, could potentially find areas with high-density nodes where the pattern detection should be triggered. In this case, fewer features will be checked.

Despite the similarity in the pattern detection for generalization and matching problems, there are substantial differences. Detecting the pattern is not the solution for the matching problem, in contrast to the generalization. In generalization, the detected pattern is simply changed to the desired structure, e.g., a roundabout in a large-scale map to a node in a small-scale map. Although, in matching, the problem is more complex as we address the varying complexity for one specific pattern. Therefore, pattern detection should be applied to both data sets.

Nevertheless, the uncertainty in the attributes and the pattern-detection process can affect the roundabout matching and matching steps. For instance, Figure 14 denotes a roundabout-like structure detected in the NVDB data set, while there is no roundabout detected in the OSM data set. The mismatch occurs because an OMS node is slightly inside the NVDB roundabout, and the dedicated roundabout-matching process mistakenly matches the NVDB roundabout to the OSM node.

In this study, we only considered roundabouts as a complex structure. Other complex structures include crossroads and dual carriageways. Furthermore, we only detected the cases illustrated in Figure 3; however, there could be various cases that we have not considered here. In cases that different complex structures are combined (Figure 15, where dual carriageways, roundabout, and a set of exit and entry are connected), one solution is to detect all the complex structures (roundabouts, multi-lane carriageways, complex crossroads, etc.) and find the relations between them. In other words, which complex structure is connected to the other one? The other solution is to provide richer data sets using ontology where the relations between instances are explicitly defined and the relations between patterns can be derived from them.

6. Conclusions

Matching VGIs with AD has become a new field with various objectives. In this study, we improved the state-of-the-art node-based network-matching algorithms. The improvement is related to better measures for finding corresponding nodes and links in the matching algorithm. Another contribution of this study is the introduction of a roundabout detection and matching process. These enhancements improved the accuracy of the algorithm; however, the efficiency of the algorithm deteriorates to some extent. The main algorithm improvement lies in pattern detection. The pattern detection presents a general approach for dealing with complex structures. If there is reliable attribute information (roundabouts in this case), then the detection process will not be necessary.

A case study on the extended node-based algorithm was performed in the province of Scania, Sweden. The case study reveals that OSM has a completeness of 87% in the urban areas and 69% in the rural areas in Scania. The accuracy of the matching was approximately 95%. The conclusion is that the extended node-based algorithm is sufficiently accurate and efficient for conducting surveys of the quality of OSM and other VGI road data sets in large geographic regions.

Acknowledgments

The first author of this study was financed by the European Union funding program, Erasmus Mundus “External Cooperation Window” (IIY action 2). The authors would like to appreciate for the funding. They are thankful to Tarafikverket and Lantmäteriet for providing Swedish road network data. The authors would also like to appreciate Cecile Duchene’s comments on pattern detection part.
Notes on contributors

Ehsan Abdolmajidi is currently a PhD student in the GIS center at the Department of Physical Geography and Ecosystem Science, Lund University. His research fields are Spatial Data Infrastructure (SDI), feature matching, semantics, and ontology. He is currently focused on utilizing the ontologies to facilitate the integration of road network data sets.

Ali Mansourian is an associate professor at the Department of Physical Geography and Ecosystem Science, Lund University, and a member of Lund University GIS Centre. His research interests are on modeling and applications of SDI, spatial web services, and spatial modeling using artificial intelligence techniques. Matching spatial data sets and databases in SDI environment is of his special interest.

Julian Will was a master student of Geomatics program in the GIS center, Lund University, from 2012 to 2014. His work focuses on developing an automated feature matching algorithm to assess the quality of OSM road network.

Lars Harrie is an associate professor at the Department of Physical Geography and Ecosystem Science Lund University and a member of Lund University GIS Centre. He received his PhD from Lund University in 2001. His research interests are Cartography, SDI, and spatial analysis. His focus is now on linking spatial data in SDI environment.

References

(1) Goodchild, M.F. Citizens as Sensors: The World of Volunteered Geography. Geo J. 2007, 69, 211–221.

(2) Al-Bakri, M.; Fairbairn, D. Assessing Similarity Matching for Possible Integration of Feature Classifications of Geospatial Data from Official and Informal Sources. Int. J. Geog. Inf. Sci. 2012, 26, 1437–1456.

(3) Girres, J.F.; Touya, G. Quality Assessment of the French OpenStreetMap Dataset. Trans. GIS 2010, 14, 435–459.

(4) Haklay, M. How Good is Volunteered Geographical Information? A Comparative Study of OpenStreetMap and Ordnance Survey Datasets. Environ. Planning B Planning Des. 2010, 37, 682–703.

(5) Ludwig, I.; Voss, A.; Krause Traudes, M. A Comparison of the Street Networks of Navteq and OSM in Germany. In Advancing Geoinformation Science for a Changing World; Geertman, S., Reinhardt, W., Toppen, F., Eds.; Springer-Verlag: Berlin, 2011; pp 65–84.

(6) Koukoletsos, T.; Haklay, M.; Ellul, C. Assessing Data Completeness of VGI through an Automated Matching Procedure for Linear Data. Trans. GIS 2012, 16, 477–498.

(7) Ruiz, J.J.; Ariza, F.J.; Ureña, M.A.; Blázquez, E.B. Digital Map Conflation: A Review of the Process and a Proposal for Classification. Int. J. Geog. Inf. Sci. 2011, 25, 1439–1466.

(8) Solutions, V. Java Conflation Suite. Available online: http://www.vividsolutions.com/JCS/ (accessed February 14, 2015).

(9) Stigmar, H. Matching Route Data and Topographic Data in a Real-Time Environment, Proceedings of Scandinavian Research Conference on Geographical Information Science, Stockholm, Sweden, June 13–15, 2005; pp 89–107.

(10) Volz, S. An Iterative Approach for Matching Multiple Representations of Street Data. Proceedings of the JOINT ISPRS Workshop on Multiple Representation and Interoperability of Spatial Data, Hanover, Germany, Feb 22–24, 2006; pp 101–110.

(11) Mustière, S.; Devogele, T. Matching Networks with Different Levels of Detail. GeoInformatica 2008, 12, 435–453.

(12) Zhang, M.; Meng, L. Delimited Stroke Oriented Algorithm-working Principle and Implementation for the Matching of Road Networks. Geo. Inf. Sci. 2008, 14, 54–53.

(13) Walser, V.; Fritsch, D. Matching Spatial Data Sets: A Statistical Approach. Int. J. Geog. Inf. Sci. 1999, 13, 445–473.

(14) Yang, B.; Zhang, Y.; Luan, X. A Probabilistic Relaxation Approach for Matching Road Networks. Int. J. Geog. Inf. Sci. 2013, 27, 319–338.

(15) Yang, L.; Zuo, Z.; Wang, R.; Ye, Y.; Hu, M. Matching Road Network Combining Hierarchical Strokes and Probabilistic Relaxation Method. Open Autom. Control Syst. J. 2014, 6, 268–276.

(16) Sch, M.; Lipeck, U.W. SimMatching: Adaptable Road Network Matching for Efficient and Scalable Spatial Data Integration. Proceedings of the 1st ACM SIGSPATIAL PhD Workshop, Dallas/Fort Worth, TX, USA, Nov 4–7, 2014; pp 1–5.

(17) Mackaness, W.; Edwards, G. The Importance of Modeling Pattern and Structure in Automated Map Generalisation. Proceedings of the Joint ISPRS/ICA Workshop on Multi-scale Representations of Spatial Data, Ottawa, Canada, July 7–8, 2002.

(18) Touya, G. A Road Network Selection Process Based on Data Enrichment and Structure Detection. Trans. GIS 2010, 14, 595–614.

(19) Mackaness, W.A.; Mackechnie, G.A. Automating the Detection and Simplification of Junctions in Road Networks. GeoInformatica 1999, 3, 185–200.

(20) Savino, S.; Rumor, M.; Lissandron, I. Road Junction Generalization in Large Scale Geodatabases. In Urban and Regional Data Management: UDMS 2009 Annual; Zlatanova, S., Fendel, E., Krek, A., Rumor, M., Eds.; CRC Press/Balkema: London, 2009; pp 205–229.

(21) Weiss, R.; Weibel, R. Road Network Selection for Small-scale Maps Using an Improved Centrality-based Algorithm. J. Spatial Inf. Sci. 2014, 9, 71–99.

(22) Thomson, R.C. The ‘stroke’ Concept in Geographic Network Generalization and Analysis. In Progress in Spatial Data Handling; Riedl, A., Kainz, W., Elmes, G.A., Eds.; Springer-Verlag: Berlin, 2006; pp 681–697.

(23) Thomson, R.C.; Richardson, D.E. The ‘good continuation’ principle of Perceptual Organization Applied to the Generalization of Road Networks. International Cartographic Exhibition, Ottawa, Canada, Aug 16–20, 1999.

(24) Zielezka, D.; Zipf, A. A Comparative Study of Proprietary Geodata and Volunteered Geographic Information for Germany. 13th AGILE International Conference on Geographic Information Science, Guimarães, Portugal, 2010.

(25) Neis, P.; Zielezka, D.; Zipf, A. The Street Network Evolution of Crowdsourced Maps: OpenStreetMap in Germany 2007–2011. Future Internet 2011, 4 (4), 1–21.

(26) Zhang, M.; Shi, W.; Meng, L.Q. A Generic Matching Algorithm for Line Networks of Different Resolutions. 8th ICA Workshop on Generalization and Multiple Representation, A Coruña, Spain, July 7–8, 2005.

(27) Will, J. Development of an Automated Matching Algorithm to Assess the Quality of the OpenStreetMap Road Network: A Case Study in Göteborg. Student Thesis Series INES, Sweden, 2014.

(28) Scipy. Available online: http://www.scipy.org/ (accessed February 20, 2015).

(29) Abdolmajidi, E.; Will, J.; Harrie, L.; Mansourian, A. Comparison of Matching Methods of User Generated and Authoritative Geographic Data. ICA Workshop on Integration, Filtering and Abstraction of User Generated or Multi-Source Spatial Content, Vienna, Austria, 2014.
(30) Campbell, J. Map Use & Analysis; McGraw-Hill: New York, 2001.
(31) Sedgewick, R. Algorithms in C; Addison-Wesley Longman: Boston, MA, 2002.
(32) Bentley, J.L. Multidimensional Binary Search Trees in Database Applications. IEEE Trans. Softw. Eng. 1979, SE-5, 333–340.
(33) Navarro, G. A Guided Tour to Approximate String Matching. ACM Comput. Surv. 2001, 33, 31–88.
(34) Serva, M.; Petroni, F. Indo-European Languages Tree by Levenshtein Distance. Europhys. Lett. 1979, SE-5, 333–340.
(35) Sweden, N.R.D.o. NVDB. http://www.nvdb.se/ (accessed February 27, 2015).