MDPI

Article

An Area Partitioning Approach to the Conflation of Road Networks with Highly Different Level of Details

Hoa-Hung Nguyen [1] and Han-You Jeong [2,*]

[1] Department of Electrical and Electronics Engineering, Pusan National University, Republic of Korea; nguyenhoahungit@gmail.com
[2] Department of Electrical Engineering, Pusan National University, Republic of Korea; hyjeong@pusan.ac.kr
* Correspondence: hyjeong@pusan.ac.kr

Abstract: A road network represents road objects in a given geographic area and their interconnections, and is an essential component of intelligent transportation systems (ITS) enabling emerging new applications such as dynamic route guidance, driving assistance systems, and autonomous driving. As the digitization of geospatial information becomes prevalent, a number of road networks with a wide variety of characteristics coexist. In this paper, we present an area partitioning approach to the conflation of two road networks with a large difference in level of details. Our approach first partitions the geographic area by the *Network Voronoi Area Diagram (NVAD)* of low-detailed road network. Next, a subgraph of high-detailed road network corresponding to a complex intersection is extracted and then aggregated into a supernode so that a high matching precision can be achieved via 1:1 node matching. To improve the matching recall, we also present a few schemes that address the problem of missing corresponding object and representation dissimilarity between these road networks. Numerical results at Yeouido, Korea's autonomous vehicle testing site, show that our area partitioning approach can significantly improve the performance of road network matching.

Keywords: Road-network matching; matching precision; matching recall; network Voronoi area diagram; intelligent transportation systems.

1. Introduction

Geographic information systems (GIS) provide the solutions for capturing, manipulating, analyzing and visualizing the geospatial data for many application fields, such as transportation, agriculture, commerce, etc. [1,2]. Initially, government agencies have built authoritative GIS because the construction of geospatial information requires extensive and accurate surveys of the land [3,4]. Recently, as the digitization of geospatial information has recently become prevalent, some portal sites or mobile service providers have constructed proprietary GIS that combines authoritative GIS, aerial photos, mobile-mapping service (MMS) and crowdsourcing data, etc. [5,6] On the other hand, voluntary GIS, such as the *openstreetmap (OSM)*, has been constructed by the participation of voluntary user carrying a GPS-enabled mobile terminal [7]. Currently, more than 7.8 million registered users all around the world contribute to the OSM [8].

A *road network* is a subset of GIS that focuses on road objects, attributes, and their interconnectivity. It is usually represented by a graph, where a node represents an intersection, an endpoint of a road, or a point of attribute change, whereas an edge represents a road segment connecting two nodes. The road network is an important component of many Intelligent Transportation System (ITS) applications. For example, a turn-by-turn navigation establishes the shortest route connecting the origin and destination in the road network. In addition, the current road traffic information at a road segment is indexed with the corresponding identifier of road network, and then broadcast as a public transportation data, which enables novel ITS applications, such as dynamic route

Citation: Nguyen, H.-H.; Jeong, H.-Y. Title. *Sensors* **2021**, *1*, 0. https://doi.org/

Received:
Accepted:
Published:

Table 1. Characteristics of authoritative, proprietary, and voluntary road networks

Characteristics	**Authoritative** [4]	**Proprietary**[5,6]	**Voluntary** [7]
Raw dataset	Accessible	Inaccessible	Accessible
Quality	Intermediate	High	Low
Level of detail	Low	High	High
Real-time data	Available	Available	Not available
Software packages	None	Limited	Abundant

guidance [9–12] and dynamic traffic management [13–15]. In a high-precision map for autonomous driving, each lane of a road segment can be represented in connection with the corresponding edge of road network [16].

Table 1 shows the characteristics of authoritative, proprietary, and voluntary road networks. First, the authoritative road network called *node-link map (NLM)* is designed to support ITS applications in Korean major roads [4]. It provides the representation of road objects associated with public transportation data, such as average speeds, road incidents, variable-message signs, and CCTV streamings [17]. Two major limitations of the NLM are the lack of software packages for ITS applications and the low-detailed representation of road network. Second, the proprietary road network has a good characteristics to support ITS services, but the access to its raw dataset and the ITS software packages is either very limited or not possible. The voluntary road network called the *OSM road network (ORN)* provides a detailed view of road network as well as a variety of open-source software packages: map editing tools (Potlatch 2 [18] and JOSM [19]), rendering tools, (Mapnik [20] and the Tirex [21]), geocoding tools (Nominatim [22]), and especially routing tools (the open-source routing machine [23] and the Valhalla [24]). However, it has been reported that the quality of OSM object obtained from crowdsourcing can be diverse in terms of accuracy, completeness, and consistency [25].

Taking into account the above characteristics of road networks, we focus on the *road network matching (RNM)* between the authoritative and voluntary road networks for emerging new ITS services. In general, the RNM problem is the association problem between a set of objects in one road network and a set of objects in the other road network, where both object sets represent the same road entity. Since each road network has its own strengths and weaknesses, a successful road network conflation can enhance the strengths and compensate for the weaknesses. In particular, a solution to the RNM between NLM and ORN can suggest a new direction to the emerging new ITS applications through the integration of NLM-indexed real-time transportation data with OSM software packages. The challenge of RNM is how to address difference in the object representation and cartographic rule between two road networks, including the *level of detail (LoD)* at a complex intersection [26–28], missing correspondent road objects [26,28,29], and the representation of acceleration/deceleration lanes.

In this paper, we present an area partitioning approach to the conflation of two road networks with highly differenct LoD. To reduce the complexity of RNM, the preprocessing step prunes the ORN objects in minor road networks. Given the NLM graph, the *network Voronoi area diagram (NVAD)* in [30] is used to partition the map area into a set of regions centered on each NLM node. For each partitioned region, the *ORN supernode grouping (ORN-SG)* algorithm is proposed to extract the ORN subgraph of a complex intersection and to aggregate it into a supernode. The 1:1 node matching between NLM node and ORN supernode can significantly improve the matching precision. To enhance the matching recall, we also propose a few schemes that address the problems of correspondent-missing NLM nodes and the representation of acceleration/deceleration lanes. The numerical results show that our area partitioning approach can achieve much higher matching precision and recall than the existing RNM approaches. The contributions of this paper are summarized as follows:

- This paper presents a new S-RNM framework that enables the integration between public transportation data and open-source ITS applications.
- The proposed S-RNM approach achieves a high matching accuracy against the LoD difference between two RNIS datasets.
- The exception-handling schemes increase the matching cardinality while maintaining high matching accuracy against the missing correspondent and different representations of acceleration/deceleration lanes.

The remainder of this paper is organized as follows. Section 2 reviews the related literature on RNM. Section 3 describes the characteristics of two road networks and provides a generic formulation on the RNM problem. Next, we present the details of area partitioning approach in section 4. Section 5 presents a few schemes to further improve the matching recall. Numerical results are presented in section 6. Finally, we conclude this paper in section 7.

2. Related Work

The RNM is the problem of associating an object or a set of object in one dataset, called the reference dataset, to an object or a set of objects in the other dataset, called target dataset, where both objects or sets of objects represent the same road entity. These objects are called the corresponding object of the others. The requirements of the RNM are maximizing the matching probability and the matching accuracy. In the ideal situation where both datasets perfectly reflect the geographical environment, the similar sets of objects are used in two datasets and each pair of the corresponding objects have exactly the same geometric, topological and semantic attributes. Then, solving the problem of the RNM will be straightforward.

However, in reality, there are representational dissimilarities between two datasets originated from the abstraction and generalization errors during the cartographic process. The first type of representational dissimilarity is the nonexistence of corresponding object in the target dataset. To mitigate this problem, it is needed to add a new corresponding road object to the target dataset, which is beyond the scope of this paper. The second type is the attributional dissimilarity between the corresponding objects. The third type is the LoD dissimilarity between the set of corresponding objects representing the same road entity. The LoD dissimilarity does not only affect the RNM, but also requires special treatment for adapting the RNM result to the ITS applications. To address these challenging issues, the RNM approach first needs a method to assess the attributional dissimilarity between a pair of corresponding nodes as well as identifying the set of objects involving in the LoD dissimilarity. Then, based on that, the matching between two datasets can be conducted in non-iterative or iterative manner. In the following sections, we will introduce the existing approaches to addressing these representational dissimilarity as well as the matching procedure in the RNM problem.

2.1. Representational Dissimilarity

The representational dissimilarity includes the attributional dissimilarity and the LoD dissimilarity. Attributional dissimilarities is the dissimilarities in the geometric, topological, and semantic attributes between a pair of corresponding objects. The geometric dissimilarity is mainly the difference of position attribute between the corresponding nodes which also leads to the difference in edge geometric attributes. The topological dissimilarity comes from the missing or the incorrect of the connectivity to the adjacent objects. The semantic dissimilarity also originates from the missing or the incorrectness of the attribute as well as the difference in the attribute definition. To address these dissimilarities, the common approach is defining the metrics to quantify the dissimilarity of attributes and selecting the target object with the smallest dissimilarity. The dissimilarity metrics between a pair of objects can be computed based on its own attributes (i.e. single object dissimilarity metrics) or together with its nearby objects attributes (i.e. multiple object dissimilarity metrics).

The single object dissimilarity metrics involve the attribute of a single object, including the geometric metrics (e.g. the difference in position, length, shape, etc. [26,28,29,31–40]), the topological RNM metrics (e.g. the difference in the node degree, etc. [26,29,31,34,35,38,40]) and the semantic RNM metrics (e.g. the difference in road name, road class, etc. [26,28,35]). The difference in position, i.e. distance, is the most used metric in the node matching [36] and the edge matching [34,35,37–39]. While the distance in the node matching is trivial, the distance between edges can be defined in several ways. The approaches in [34,35,38] exploit the Hausdorff distance which is the maximum of the minimum distance of any point on reference edge to the target edge. The sampling-based distance used in [37] is the average distance among the pair of equidistant points on both edges. A different sampling-based distance is used in [39] where the average distance is aggregated from the equidistant points on reference edge and their closest point on the target edge. The length difference are used in [26,29,33–35,38]. There are several ways to define the shape difference. In [31], it defined as the area created from the two edges after shifting and scaling to make both edge endpoints overlapped. In [26], the shape can be defined as the cumulative angle function among the line segments along an edge and the shape difference is the gap between the two functions. The other single object dissimilarity metrics such as the differences in node degree, road name, road class are used as the sub-metrics used together with the distance, length difference and shape difference metrics [26,28,29,31,34,35,38,40]. The major problem of single object dissimilarity metrics is wrong identification of the corresponding object if the attributional dissimilarity between the reference object and its corresponding object is large.

The multiple object dissimilarity metrics aggregate the attributional dissimilarity of the matched object as well as its nearby objects [26,28,32]. The authors in [32] proposed the cluster-based matching approach in which the clusters are created by spanning from a reference node as well as the target node. This pair of center nodes are matching by comparing the aggregated difference in the distance, length and shape of each pair of edges in two clusters. The authors in [28] proposed another cluster-based matching approach in which the cluster is similarly constructed like the cluster in [32]. The reference and the target nodes are matched by comparing the structure of their clusters. In [26], the authors proposed the delimited-stroke-oriented algorithm to match the groups of connected edges that seem to be on the same road called delimited-stroke.

On the other hand, the LoD dissimilarities occur when different number of objects are used to represent a road entity. Comparing to the attributional dissimilarities, they incur the significant dissimilarity in geometrical, topological and semantic attributes among all involving objects. Therefore, the special treatment for LoD dissimilarity is necessary to achieve the correct matching. They can be classified in the edge LoD dissimilarity and the node LoD dissimilarity.

The edge LoD dissimilarity comes from the fragmenting of the road into several concatenating edges and the division of the road into two edges, each of which representing one traffic direction. The former case is addressed by buffer growing method [29,33], grouping edge to road [26], or the shortest path between nodes [31,35]. In the buffer growing algorithm [29,33], for each reference edge, the list of target edges within a given distance (buffer) is selected as matching candidates. If there is no edge entirely locating inside the buffer, due to the edge LoD dissimilarity, the buffer is extended from the current reference edge to its adjacent edge. By this way, the set of corresponding edges with LoD dissimilarity can be grouped. The approach in [26] groups the edges into strokes containing edges in both datasets that may be in the same road and performs matching on them. The last approach relies on the node matching to match the edges by searching the shortest path between two pairs of matched nodes [31,35]. To address the latter case of the edge LoD dissimilarity, the common solution is detecting and grouping them into one edge [26,28].

The node LoD dissimilarity exists when the different number of nodes are used for representing an intersection. Usually, the intersection represented several nodes in one dataset and a single node in the other dataset. In [28], these issues are overcome by grouping the nodes constituting the intersection into a single node. These nodes are detected by finding an area with high node density as suggested in [27]. In [26], the LoD dissimilarity at the roundabout is presented in which the related nodes are detected by checking circle-like shape formed by the edges connecting them. However, the existing approach either includes many false positive, incomplete and redundant matching or addresses the specific types of intersection.

To effectively address the node LoD dissimilarity, we need to rely on the distinct geometrical and topological features of the intersection with the LoD dissimilarity in the datasets rather than relying on the statistical parameters such as distance threshold [27] which are the source of incorrect grouping.

2.2. Matching Procedure

Based on dissimilarity metrics and the grouping of objects with LoD dissimilarity, the matching can be directly produced (non-iterative matching) or combining with the iterative matching to improve the matching probability and matching accuracy.

The iterative matching attempts to reduce the mismatching by expanding the matchings from the set of reliable matchings (i.e. Iterative matching from seeds [26, 34]), or repeatedly improving the matching consistency (i.e. Iterative matching for the consistency [29,38]). The iterative matching from seeds finds the matching with strict condition, and expands the matching to the nearby objects [26,34]. The strong point of iteration from seeds is that the later matched objects partly rely on the reliable matching of the seed objects. In [34], the seed node matching is selected from the matching with only one target node candidate that satisfies thresholds of the node distance and adjacent edges angle difference. In [26], the seed matching is the matching of the large strokes that satisfied the threshold on several metrics such as orientation, length, shape. However, the effectiveness of this iterative matching highly depends on condition to select the seeds. The tight condition leads to smaller amount of seeds, while the looser condition incurs the mismatching of the seed itself.

On the other hand, the iterative matching for the consistency relies on the initial matching of two datasets then gradually modifies the initial matching to improve the consistency among the matching [29,31,38]. This approach relies on the matching consistency checking which examines the set of initial matchings by the dissimilarity metrics to evaluate the consistency of the nearby objects matchings. The corresponding target object of a reference object can be changed to obtain the higher consistency. In [31], the authors proposed the adjacent matching consistency checking for the node matching. Initially, target node candidates are matched with node degree and their edges angles. The corresponding target node is decided by checking the matching consistency of the adjacent nodes of the reference node to the adjacent nodes of the target node. In [29,38], the one-hop and two-hop matching consistency checking is also proposed for edge matching. However, similar to iterative matching from seed, the iterative matching for the consistency also relies on the fixed threshold of the dissimilarity metrics to find candidate [29,31,38] or the fixed threshold of the consistency condition [31]. Therefore, both of the approaches may not effectively address the mismatching.

To address the drawbacks of the existing approaches, we need a new iterative matching approach to effectively overcome the attributional dissimilarity in our RNM problem. It should allow the relaxation of the matching condition and combine the strong points of two existing iterative matching approaches.

3. Input Road Networks and RNM Problem Specification

In this section, we describe the characteristics of two road networks, i.e. NLM and ORN, and then formulate the RNM problem.

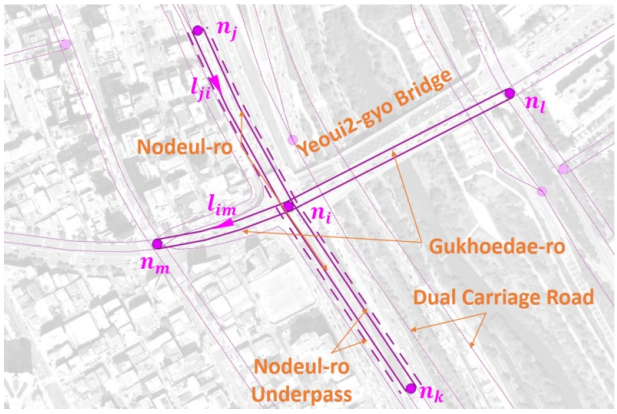


Figure 1. NLM graph representation around Yeoui2-gyo intersection

Table 2. The *road_rank* attribute in the NLM.

road_rank	**Explanation**
101	Highway
102	Urban expressway
103	National road
104	Metropolitan city road
105	Aerial or inter-province road
106	Intra-province road
107	Intra-city or island road
108	Other roads

3.1. Node-Link Map

The Korean government has initiated the national GIS project in 1995, and completed the construction of geospatial database in 2009 [41]. The NLM is the road network of this database that represents major road objects in Korea [4]. It also provides a unified identifier (ID) hierarchy to its road entity. In order to efficiently exchange the ITS information, the Korean law enforces that all ITS applications must use the NLM ID hierarchy to exchange road and traffic information [17].

Figure 1 shows NLM graph representation of Yeoui2-gyo intersection, Yeoui-do, Seoul, Korea overlaid on top of aerial view, where Gukhoe-daero (east-west road) and the access ramps of Nodeul-ro (north-south underpass) are interconnected. The NLM graph is a *directed* graph $\mathcal{G}_N = (\mathcal{N}, \mathcal{L})$, where $\mathcal{N}$ is the set of nodes representing the points at which the road characteristics are changed, such as intersection (n_i, n_l, and n_m), traffic monitoring point, administrative boundary, and the endpoints of road, overpass, and underpass (n_j and n_k). A single NLM node $n_i \in \mathcal{N}$ is used to represent a complex intersection (Yeoui2-gyo) without a detailed view of the internal road network. We define subgraph $\mathcal{G}_N(n_i) = (\mathcal{N}(n_i), \mathcal{L}(n_i))$ consists of NLM node n_i, its directly connected links (pink solid links in $\mathcal{G}_N$), and the neighbor NLM nodes (n_j, n_k, n_l, and n_m). An NLM node is placed at the crosspoint of two roads, where a road consists of two parallel links each of which represents a unidirectional road segment. In a dual carriage road, it is placed at the endpoint of two NLM links.

In the NLM, the geometric shape of a link is approximated by a sequence of concatenated line segments. For example, unidirectional links l_{ji} and l_{im} are shown by pink solid lines with triangular marks for their directions. The underpass and overpass links are also placed in parallel with the main road segment with an additional spacing between them. For example, Nodeul-ro underpass is shown by pink dashed lines in Figure 1. Each link has a set of attributes, such as *link_id, f_node, t_node, road_rank, road_type, connect, road_use*, etc., where the *road_rank* attribute represents the class of road segment as shown in Table 2, *road_type* specifies the type of road, such as overpass, underpass,

Table 3. Major *highway* tag of an OSM way.

***highway* tag group**	***highway* tag value**
Roads	*motorway, trunk, primary, secondary, tertiary, unclassified, residential, service*
Link roads	*motorway_link, trunk_link, primary_link, secondary_link, tertiary_link*
Special roads	*living_street, pedestrian, track, bus_guideway, escape, raceway, road*
Paths	*footway, bridleway, steps, path, cycleway*
Sidewalks	*sidewalk*
Cycleways	*cycleway*

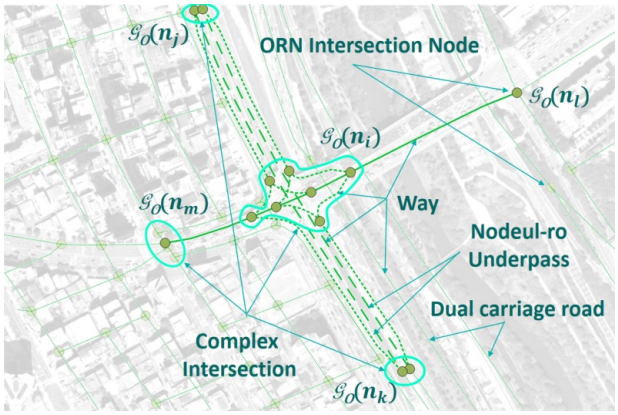


Figure 2. ORN graph representation around Yeoui2-gyo intersection

bridge, tunnel, etc., *connect* specifies the type of ramps depending on *road_rank* attribute, and *f_node* and *t_node* represent the start and end node indexes of NLM link, respectively.

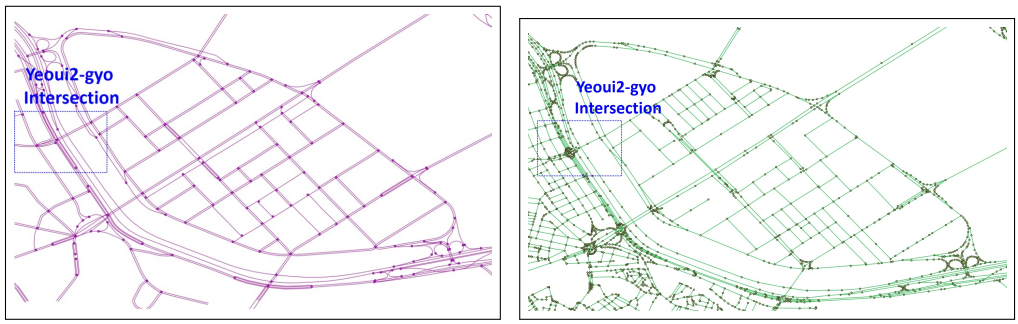
3.2. OpenStreetMap Road Network (ORN)

The ORN is a subset of OSM objects with *highway* tag, where a tag is an ordered pair of (key, value) identifying the attribute of a road object. Table 3 shows the *highway* tag of way which is classified into a few groups. In each group, the tag values are ordered from the most important to the least important. The main focus of this paper is on the *road* and *link road* groups, where the former is a way for representing a road while the latter is a way for connecting two roads in a complex intersection. We initially prune all ORN objects in *special roads*, *paths*, *sidewalks*, and *cycleways* groups that do not correspond to the NLM objects. This pruning process removes approximately 20 % of unnecessary road objects from the original ORN. Furthermore, we also remove the subgraphs for underpass/overpass in both road networks because they can be easily matched via their attribute, such as NLM *road_type*, and ORN *tunnel* and *bridge* tags.

Figure 2 shows the ORN graph representation which can be modeled by *undirected* graph $\mathcal{G}_O = (\mathcal{V}, \mathcal{E})$. Contrary to NLM graph $\mathcal{G}_N$, ORN graph $\mathcal{G}_O$ is designed to reflect the detailed road network at a complex intersection. This feature makes the ORN more suitable for ITS applications, such as navigation and autonomous driving.

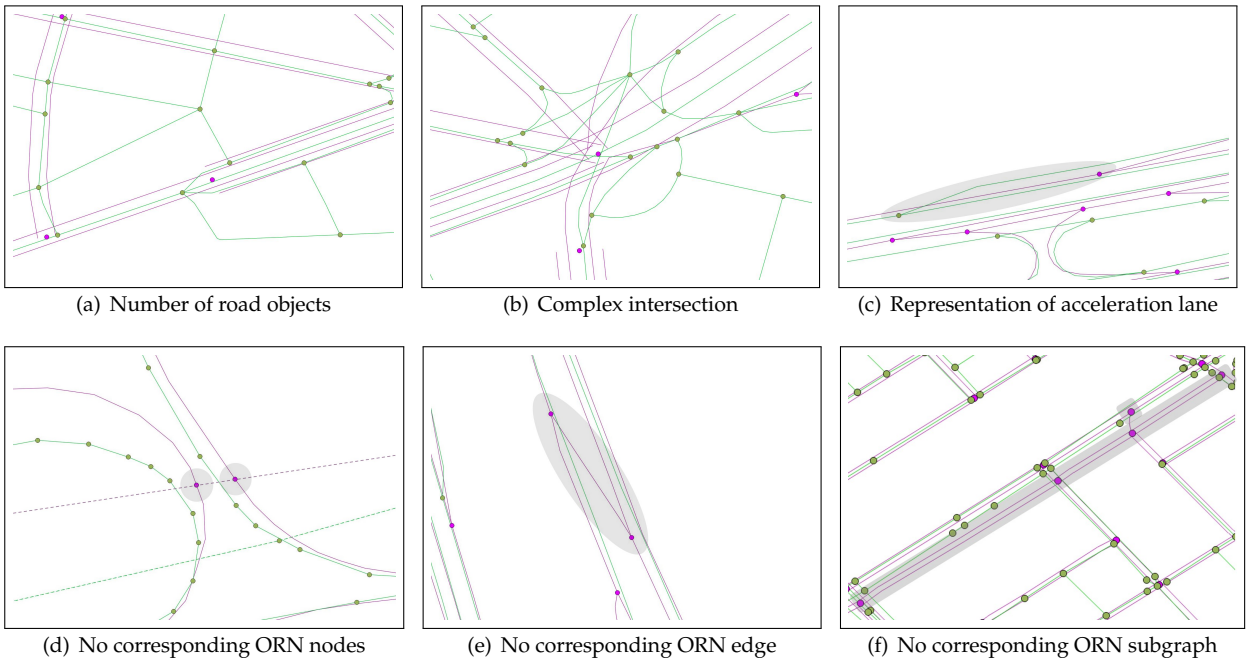
In $\mathcal{G}_O$, an ORN node $v \in \mathcal{V}$ is connected to at least three neighbor ORN nodes. In the RNM, NLM node n is associated with ORN subgraph $\mathcal{G}_O(n)$, where the ORN subgraph can be a single ORN intersection node , e.g. $\mathcal{G}_O(n_l)$, disconnected subgraphs, e.g. $\mathcal{G}_O(n_j)$ and $\mathcal{G}_O(n_k)$, or a connected subgraph, e.g. $\mathcal{G}_O(n_i)$ and $\mathcal{G}_O(n_m)$, in Figure 2. If an intersection consists of a single ORN intersection node, it is called a simple intersection; otherwise, a complex intersection.

The atomic unit for representing an ORN road is a way $w \in \mathcal{W}$ which may span multiple ORN nodes [7]. If way w includes more than two ORN nodes, it is decomposed into consecutive ORN edges $e \in \mathcal{E}$ so that each edge connects two ORN nodes only. In Figure 2, the Gukhoe-daero in the OSM subgraph $\mathcal{G}_O(n_i)$ consists of edges with *road* tag group only, shown in solid green lines, whereas all remaining edges in $\mathcal{G}_O(n_i)$ belong to *link road* tag group, represented by dotted green lines. On the other hand, all intersecting



(a) NLM graph of Yeoui-do roads (b) (Pruned) ORN graph of Yeoui-do roads

Figure 3. NLM and ORN graph representation of Yeoui-do roads



(a) Number of road objects (b) Complex intersection (c) Representation of acceleration lane

(d) No corresponding ORN nodes (e) No corresponding ORN edge (f) No corresponding ORN subgraph

Figure 4. Examples of the representational dissimilarities between NLM and ORN

edges at a simple intersection, such as $\mathcal{G}_O(n_l)$, belong to *road* tag group. In case of dual carriage road, a distinct edge used for each ORN edge whose direction is specified to the *direction* tag.

3.3. Problem Specification

Figure 3 shows the NLM and ORN graph representation of Yeoui-do roads which are given as the input of our RNM problem. The NLM in Figure 3(a) shows an abstract view consisting of major public roads only, while the ORN in Figure 3(b) shows a much more detailed representation of the road network. Given NLM $\mathcal{G}_N$ and ORN $\mathcal{G}_O$, the RNM problem is the association problem between an NLM object and a set of ORN objects that correspond to the same road entity.

Since each road network has its own representation rules of the road network, there are several differences in object representation and cartographic rule between two road networks as shown in Figures 4: Figure 4(a) shows different numbers of road objects, where the ORN shows both major and minor roads in a geographical area while the NLM displays major public roads only. Figure 4(b) illustrates different LoDs at a complex intersection, where the ORN illustrates all connecting edges at the intersection whereas the NLM aggregates them into a single node. Figure 4(c) reveals two different rules to represent acceleration (deceleration) lane, where it is a part of the mainline

road in the NLM, while it is a distinct edge with *trunk_link* tag in the ORN. Figure 4(d) shows two correspondent-missing NLM nodes which are located at the crosspoints of the administrative boundary. Figure 4(e) illustrates a correspondent-missing NLM edge due to the omission of corresponding ORN edge during the crowdsourcing process of OSM. Figure 4(f) also shows a connected correspondent-missing NLM subgraph due to the OSM crowdsourcing errors.

To achieve high matching performance, a comprehensive S-RNM solution needs to address the fundamental issues of these representational dissimilarities, as follows:

1. To identify and group the set of ORN nodes at a complex intersection in order to alleviate the LoD difference between two road networks,
2. To find a methodology to reliably match two corresponding objects each of which has its own cartographic rule, e.g. the representation of acceleration/deceleration lanes and administrative boundary nodes, and
3. To insert a new ORN node for each correspondent-missing NLM node, a new ORN edge for each correspondent-missing NLM link, and a new ORN subgraph for each connected correspondent-missing NLM subgraph.

4. Area Partitioning Approach

In this section, we present the area partitioning approach to addressing the LoD difference at a complex intersection. Given an NLM subgraph, the challenging task is to accurately extract the corresponding OSM subgraph against a wide variety of intersection topology as shown in Figure 4(b). An inaccurate OSM subgraph incurs an incorrect matching which in turn influences the accuracy of the other matchings. This propagation eventually results in severe degradation of RNM precision.

The area partitioning approach takes each NLM subgraph $\mathcal{G}_N(n_i)$ and ORN graph $\mathcal{G}_O = (\mathcal{V}, \mathcal{E})$ as the input parameters, and outputs an ORN supernode v_i^* which replaces the whole corresponding ORN subgraph $\mathcal{G}_O(n_i)$. It first partitions the map area into regions around the center of the intersection in which the corresponding ORN subgraph may exist. Then, it extracts ORN subgraph $\mathcal{G}_O(n_i)$ along the path connecting each pair of entering and exiting points across the region boundary, taking into account the turning information and geometry of intersection. Finally, the set of vertices and edges in these connecting edges constitute an ORN subgraph $\mathcal{G}_O(n_i)$ which is replaced by an ORN supernode v_i^*.

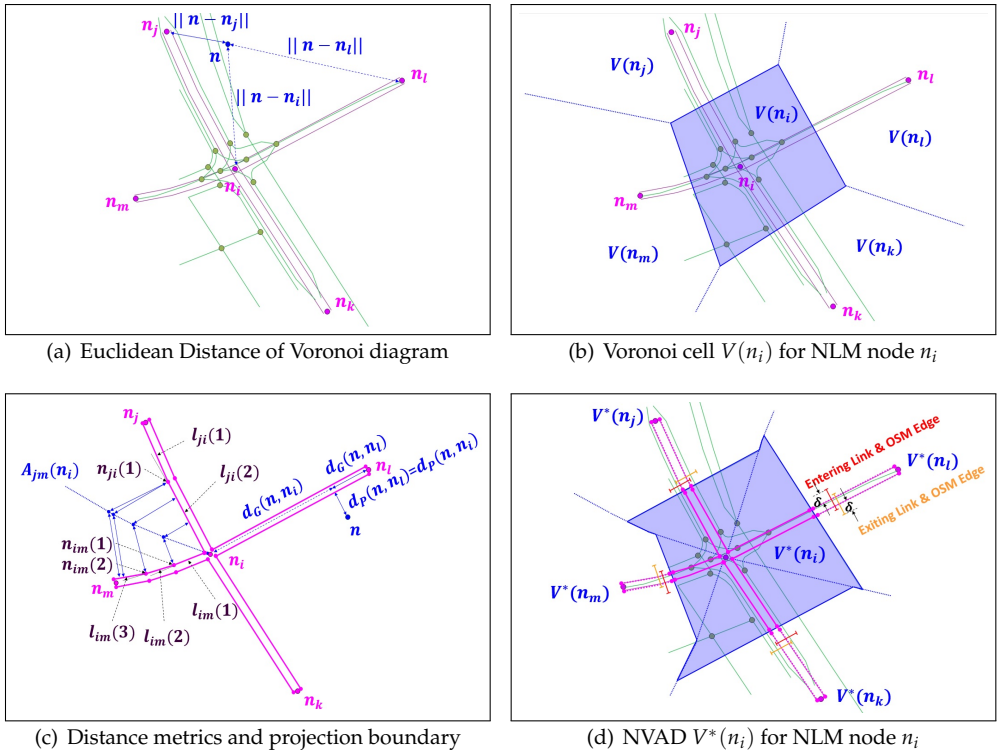
4.1. Network Voronoi Area Diagram (NVAD) for Partitioning Map Area

Given NLM subgraph $\mathcal{G}_N(n_i)$ and the corresponding map area $\mathcal{A}(n_i)$ around n_i, the first task of our area partitioning approach is to partition this area into regions, where each region is centered at an intersection in $\mathcal{N}(n_i)$. A simple method called the *Voronoi diagram (VD)* partitions the map area $\mathcal{A}(n_i)$ based on the Euclidean distance [30]. The basic idea is to associate a point $n \in \mathcal{A}(n_i)$ with the region of the closest intersection n_x, called the Voronoi cell $V(n_x)$, in terms of the Euclidean distance metric:

$$V(n_x) = \{n | \ \|n - n_x\| \leq \|n - n_y\| \quad \forall y \neq x, \ n_x, n_y \in \mathcal{N}(n_i)\}, \tag{1}$$

where $\mathcal{N}(n_i) = \{n_i, n_j, n_k, n_l, n_m\}$ for NLM graph $\mathcal{G}_N(n_i)$ in Figure 5(a). Given an ORN node $n \in V(n_j)$ in map area $\mathcal{A}(n_i)$, the Euclidean distances from three closest NLM nodes are shown in Figure 5(a). For two NLM nodes n_x and n_y ($n_y \in \mathcal{N}(n_i) \backslash \{n_x\}$), the boundary of Voronoi cells becomes a hyperplane that is equidistant from both NLM nodes. Finally, Voronoi cell $V(n_x)$ is constructed by intersecting all half-spaces in which NLM node n_x is located. For example, Voronoi cell $V(n_i)$ is illustrated with the blue transparent quadrilateral in Figure 5(b).

However, given NLM subgraph $\mathcal{G}_N(n_i)$, the Euclidean norm is no longer a fair measure to evaluate the distance between point $n \in \mathcal{A}(n_i)$ and the set of NLM nodes in $\mathcal{N}(n_i)$. This is because the Euclidean distance metric does not account for the distance



(a) Euclidean Distance of Voronoi diagram (b) Voronoi cell $V(n_i)$ for NLM node n_i

(c) Distance metrics and projection boundary (d) NVAD $V^*(n_i)$ for NLM node n_i

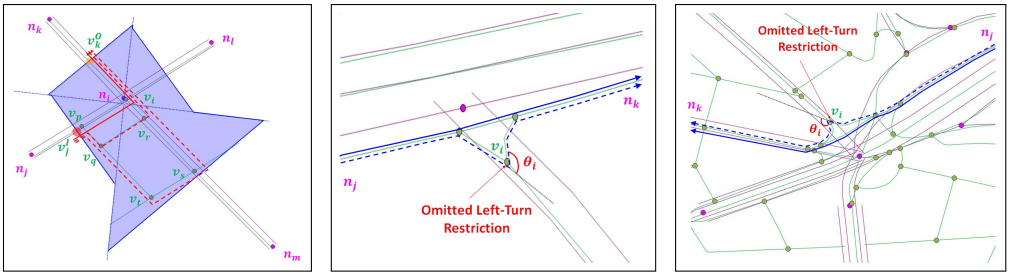
Figure 5. VD and NVAD to partition map area $\mathcal{A}(n_i)$ around NLM node n_i.

from the curved roads in $\mathcal{G}_N(n_i)$. To address this problem, our area partitioning approach adopts the *network Voronoi area diagram (NVAD)* whose measure reflects two distance factors [30]: First, if point n is on subgraph $\mathcal{G}_N(n_i)$, the distance should be the length of shortest path to NLM node $n_x \in \mathcal{N}(n_i)$ in $\mathcal{G}_N(n_i)$, called the graph distance $d_G(n, n_x)$. If point n lies in $\mathcal{A}(n_i) \backslash \mathcal{G}_N(n_i)$, the measure should also consider the projection distance $d_P(n, n_x)$ to the closest NLM link of subgraph $\mathcal{G}_N(n_i)$. Figure 5(c) shows these distances between point n and two closest intersections n_i and n_l. Consequently, the distance metric of NVAD is defined as the sum of these two distance components, i.e.,

$$\|n - n_x\| = d_G(n, n_x) + d_P(n, n_x). \tag{2}$$

To determine the NLM link onto which a given point n is projected, we choose an example of map area $\mathcal{A}_{jm}(n_i)$ surrounded by unidirectional NLM links l_{ji} and l_{im} in Figure 5(c), where the former (latter) consists of two (three) line segments. The k-th line segment and vertex of NLM link l_{ji} are denoted by $l_{ji}(k)$ and $n_{ji}(k)$, respectively, where $n_{ji}(0) = n_j$ and $n_{ji}(2) = n_i$. Our approach draws the equiangle boundary starting from the center of intersection n_i until its projection approaches the endpoint of shorter line segment $n_{im}(1)$. Notice that any points on this projection boundary are equidistant from both NLM links l_{ji} and l_{im}. In Appendix A, we demonstrate that the projection boundary curve becomes a concatenation of linear or parabolic segments. Figure 5(c) shows the resulting blue dotted projection boundary of map area $\mathcal{A}_{jm}(n_i)$.

Figure 5(d) shows all projection boundaries that partition map area $\mathcal{A}(n_i)$ into four projection areas each of which has a pair of NLM links between n_i and its neighbor NLM node. At the middle point of these links, we draw a perpendicular line that bisects the projection area. Then, NVAD cell $V^*(n_i)$ is determined by the union of the side of the line on which NLM node n_i is located, as shown by the blue transparent polygon in Figure 5(d). For each NLM link l passing through the projection boundary, we finally build a list of candidate ORN edges $\mathcal{E}_l = \{e_l(1), e_l(2), \cdots\}$ of the same direction whose distance along the boundary line is less than threshold δ. For example, in Figure 5(d),



(a) Kookmin-bank intersection (b) Yeouisangryu-IC2 (c) Yeongdeungpo rotary

Figure 6. OSM paths from the entering edge closest to l_{ji} to the exiting edge closest to l_{ik} at the boundary of intersection area $V^*(n_i)$

NLM links l_{ji} and l_{ik} have two ORN edges in their lists, while all remaining NLM links have only one ORN edge. In the next section, the candidate ORN edges will be examined to be the correspondent of an NLM link.

4.2. Construction of Candidate ORN Subgraphs

Given NVAD cell $V^*(n_i)$, allowable turn information at NLM node n_i, and set $\mathcal{E}_l$ of candidate ORN edges for NLM link l, the objective of this section is to extract the corresponding ORN subgraph $\mathcal{G}_O(n_i)$ from the ORN subgraph in $V^*(n_i)$. Our key observation is that *an intersection allows at most one path for each pair of entering and exiting roads*. Starting with null OSM subgraph $\mathcal{G}_I(n_i)$ having no OSM node and edge, the basic idea of our approach is to expand it by inserting a single OSM path that passes the intersection according to the turn specified to each pair of entering and exiting OSM edges. Without loss of generality, we focus on the construction of path p_{jk} that passes the intersection according to the specified turn between an entering OSM edge of NLM link l_{ji} and an exiting OSM edge of NLM link l_{ik}, as shown in Figure 6.

Figure 6(a) shows an example of simple intersection, where NLM node n_i connects a two-way road (l_{ki} and l_{ik}) and three one-way roads (l_{ji}, l_{il}, and l_{mi}). Due to the LoD difference, the OSM graph in $V^*(n_i)$ consists of two components: 1) the corresponding OSM subgraph $\mathcal{G}_I(n_i)$ almost overlapped with NLM subgraph $\mathcal{G}_N(n_i)$, and 2) the remaining OSM subgraph representing a minor road network around intersection n_i. Denoting by v_j^I and v_k^O the crosspoint of the boundary of polygon $V^*(n_i)$ and the entering and exiting OSM edges, respectively, there are three candidate paths in Figure 6(a): $p_{jk}(1) = v_j^I \rightarrow v_i \rightarrow v_k^O$, $p_{jk}(2) = v_j^I \rightarrow v_p \rightarrow v_q \rightarrow v_r \rightarrow v_k^O$, and $p_{jk}(3) = v_j^I \rightarrow v_p \rightarrow v_t \rightarrow v_s \rightarrow v_k^O$. Among these paths, the OSM-SG algorithm chooses the path that has smallest total turning angle. For example, path $p_{jk}(1)$ has the smallest total turning angle since it makes only one left-turn at v_i compared to three turns in other two paths. The specified left-turn at the closest OSM node v_i. This path is then inserted to OSM subgraph.

Although OSM subgraph $\mathcal{G}_I(n_i)$ is much more complex than NLM subgraph $\mathcal{G}_N(n_i)$ around a complex intersection, our key observation that each pair of OSM edges has at most one path is still valid for all complex intersections in Yeoui-do except for the blue dashed paths in Figures 6(b) and 6(c). They are originated from the crowdsourcing error that omits a left-turn restriction at OSM node v_i by the participating users, and eventually, turn out to be invalid paths. Unfortunately, these human errors are inevitable in the crowdsourcing-based OSM dataset. To exclude these exceptional paths from OSM subgraph $\mathcal{G}_I(n_i)$, our approach utilizes the second key observation that, *to smoothly connect the entering and exiting OSM edges, the geometry of connecting edge in a complex intersection is designed in a way that the curvature changes linearly with the curve length,* called the *clothoids*. Based on this observation, our approach discards a path, if it has two consecutive edges and the angle in between them abruptly changes, e.g. the blue dashed path at node v_i in Figures 6(b) and 6(c).

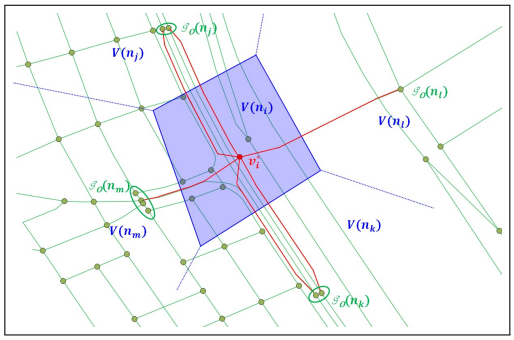


Figure 7. Supernode v_i^* replacing the whole OSM subgraph $\mathcal{G}_I(n_i)$

4.3. Supernode v_i^* for the Extracted OSM Subgraph $\mathcal{G}_I(n_i)$

For each combination of candidate OSM edges in lists $\{\mathcal{E}_l\}$, the OSM-SG algorithm counts the number of allowable turns along the path between its entering and exiting OSM edges. Then, it chooses the optimal combination that has the largest number of allowable turns, which results in the corresponding OSM subgraph $\mathcal{G}_I(n_i)$.

The final step of the OSM-SG algorithm is to extract an intersection OSM node whose degree is no less than three in $\mathcal{G}_I(n_i)$, from $\mathcal{V}(n_i)$. If there is at least one intersection OSM node in $\mathcal{G}_I(n_i)$, the OSM-SG algorithm replaces these OSM nodes with supernode $v_i^* \in \mathcal{V}^*$ located at the center of them, as shown in Figure 7. Then, node matching is straightforward one-to-one matching between the NLM node n_i and OSM subgraph $\mathcal{G}_I(n_i)$. The edge matching is also straightforward because among the candidate OSM edges in $\mathcal{E}_l$ of NLM link l (see Section 4.1), only one OSM edge is selected when choosing the optimal OSM edges combination for the corresponding OSM subgraph $\mathcal{G}_I(n_i)$.

Finally, we partition NLM graph $\mathcal{G}_N$ into the matched and unmatched NLM subgraphs: The former is denoted by $\mathcal{G}_N^* = (\mathcal{N}^*, \mathcal{L}^*)$ and the latter by $\overline{\mathcal{G}}_N = (\overline{\mathcal{N}}, \overline{\mathcal{L}})$. We also define the matching function $\mathcal{M}(obj)$ as the corresponding OSM road objects to given NLM object obj. For example, $\mathcal{M}(n_i)$ is equal to supernode v_i^* for each NLM node $n_i \in \mathcal{N}^*$, and $\mathcal{M}(l_{ij})$ is a concatenated OSM edges in set $\mathcal{E}^* = \{e^*\}$ for each NLM link $l_{ij} \in \mathcal{L}^*$.

In the following section, we introduce the exception-handling schemes of our S-RNM approach to appropriately insert OSM road objects corresponding to each connected NLM component $\overline{\mathcal{C}}_N$ in the unmatched NLM subgraph $\overline{\mathcal{G}}_N$.

5. Exception-Handling Schemes of the S-RNM

Since the OSM-SG is designed to resolve the LoD difference between NLM and OSM datasets, it cannot address the different cartographic rules (See Figures 4(c) and 4(d)), and the user errors in the OSM crowdsourcing process (See Figures 4(e) and 4(f)). The objective of this section is to present a set of exception-handling schemes that maximize the matching cardinality of the S-RNM. For each connected NLM component $\overline{\mathcal{C}}_N$ in unmatched NLM graph $\overline{\mathcal{G}}_N$, the exception-handling scheme consists of two major components: the extension of the OSM-SG to address the different cartographic rules, the insertion of the OSM supernode corresponding to unmatched NLM component $\overline{\mathcal{C}}_N$.

5.1. OSM Extended Supernode Grouping (OSM-ESG) Scheme

Figure 8 shows an example of the OSM-ESG scheme for unmatched NLM node $\overline{n}_i$ with $|\mathcal{L}(\overline{n}_i)| \geq 3$ because its corresponding OSM node v_i^* is located outside of its map area $\mathcal{A}(\overline{n}_i)$. Although all neighbor NLM nodes are correctly matched with their corresponding OSM counterparts, the difference in the placement of intersection nodes results in no intersection OSM node in its NVAD. A typical cause of this problem is the difference in the representation rule of merging lane as shown in Figure 4(c).

To resolve this problem, the OSM-ESG utilizes the matching of the neighbor NLM nodes in previous step and extends the OSM-SG at the candidate OSM subgraph con-

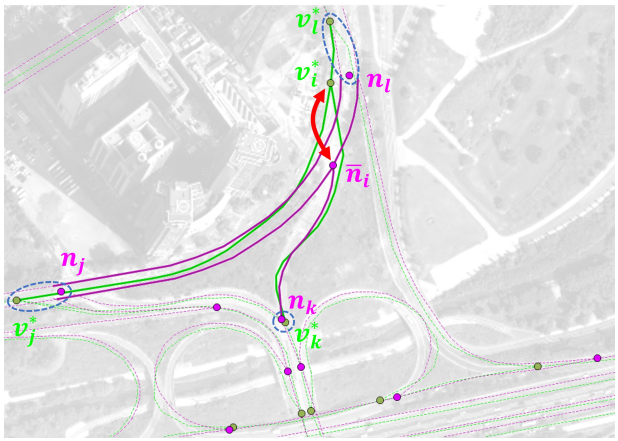


Figure 8. Example of the OSM-ESG scheme for unmatched NLM node $\overline{n}_i$, where the corresponding OSM node is in map area $\mathcal{A}(n_l)$

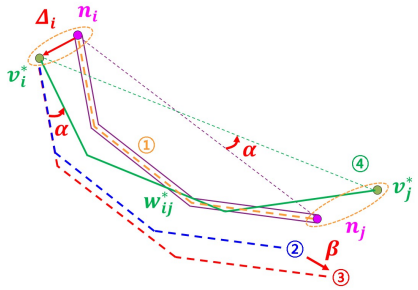


Figure 9. Example of OEI scheme for correspondent-missing NLM link $\overline{l}_{ij}$

struction step (Section 4.2). The extensions include: (1) extending the boundary of the OSM subgraph to the corresponding OSM nodes of neighbor NLM nodes, (2) replacing the boundary crosspoints of polygon $V^*(n_i)$ with the the corresponding OSM nodes of neighbor NLM nodes. In this example, the OSM-ESG constructs two OSM paths in the $V^*(n_i)$, i.e., $v_j^* \rightarrow v_l^*$ and $v_k^* \rightarrow v_l^*$,. Through the same procedure in section 4.3, the OSM-ESG scheme extracts OSM supernode v_i^* as the correspondent of unmatched NLM node $\overline{n}_i$.

5.2. Insertion of OSM Supernode for Unmatched NLM Component

A connected NLM subgraph in unmatched NLM graph $\overline{\mathcal{G}}_N$ can be an NLM link $\overline{l}_{ij}$, an NLM node $\overline{n}_i$, or a connected NLM component $\overline{C}_N$ consisting of at least two NLM road objects. We first present the *OSM edge insertion (OEI)* scheme for correspondent-missing NLM link $\overline{l}_{ij}$. For correspondent-missing NLM node $\overline{n}_i$, we propose the *NLM node projection onto the matched OSM edge (NPME)* scheme. The, we present the *sequential OSM subgraph construction (SOSC)* scheme for connected correspondent-missing NLM component $\overline{C}_N$. Finally, the internal structure of the inserted OSM supernode and the conversion of NLM attributes into OSM tags are presented.

5.2.1. OSM Edge Insertion (OEI) Scheme

Figure 9 shows an example of OEI scheme for correspondent-missing NLM link $\overline{l}_{ij}$. In this example, both NLM endpoints of $\overline{l}_{ij}$ are already matched with OSM supernodes, i.e., $\mathcal{M}(n_i) = v_i^*$ and $\mathcal{M}(n_j) = v_j^*$. The goal of this section is to insert an OSM edge e_{ij}^* between these two supernodes that corresponds to NLM link $\overline{l}_{ij}$. In general, there are three factors to be considered by the OEI scheme: 1) the displacement Δ_i between NLM node n_i and supernode v_i^*, 2) the angle difference α between NLM line segment $\overline{n_i n_j}$ and

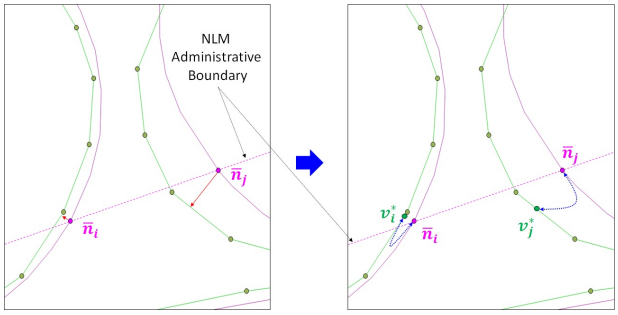
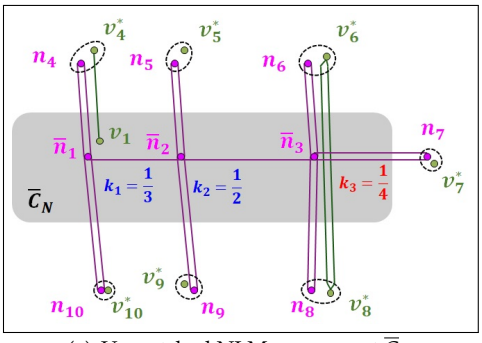
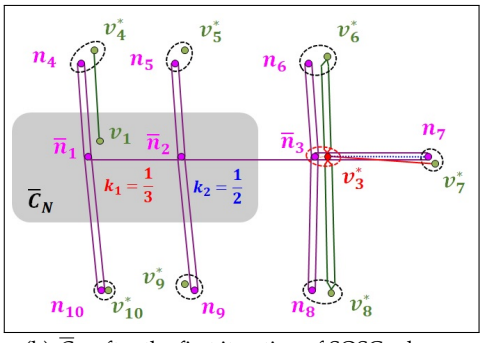


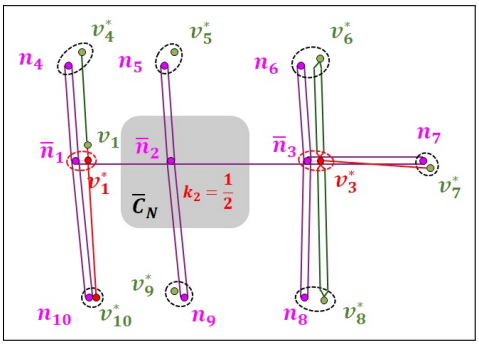
Figure 10. Example of NPCE scheme for two NLM nodes on the administrative boundary



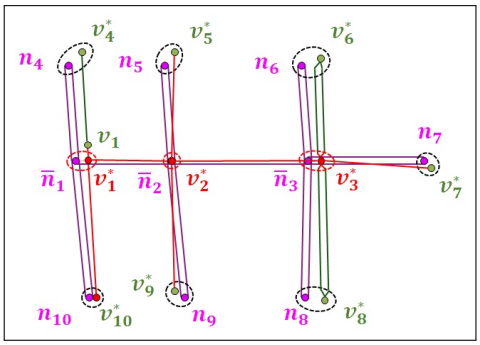
(a) Unmatched NLM component $\overline{C}_N$



(b) $\overline{C}_N$ after the first iteration of SOSC scheme



(c) $\overline{C}_N$ after the second iteration of SOSC scheme



(d) The OSM subgraph matched with NLM subgraph $\overline{C}_N$

Figure 11. Example of SOSC scheme for unmatched NLM component $\overline{C}_N$

OSM line segment $\overline{v_i^* v_j^*}$, and 3) the Euclidean distance ratio β of line segment $\overline{v_i^* v_j^*}$ to line segment $\overline{n_i n_j}$, where

$$\beta = \frac{\left\| v_i^* - v_j^* \right\|}{\left\| n_i - n_j \right\|}. \tag{3}$$

The OEI scheme first computes an orange dashed link between NLM nodes n_i and n_j which is equally distant from both NLM links $\bar{l}_{ij}$ and $\bar{l}_{ji}$. Next, it obtains a blue dashed link by shifting the orange dashed link by Δ_i so that it can start from supernode v_i^*. Then, it computes a red dashed link by scaling the blue dashed link by β. Finally, it obtains the OSM edge e_{ij}^* between two supernodes by rotating the red dashed link by α around supernode v_i^*. Since the OSM edge concept does not exist in the OSM dataset, the newly inserted OSM edge is reflected into the OSM dataset in form of a new OSM way.

5.2.2. NLM Node Projection onto the Matched OSM Edge (NPME) Scheme

Figure 10 shows an example of NPME scheme for two correspondent-missing degree-2 NLM nodes $\overline{n}_i$ and $\overline{n}_j$ on the administrative boundary. In this case, both NLM

nodes have all matched neighbor NLM nodes, but they still belong to unmatched NLM subgraph $\overline{\mathcal{G}}_N$, because there is no intersection OSM node in their NVADs. This is mainly due to the difference in their cartographic rules between NLM and OSM datasets: The former requires that an NLM node must be placed on the administrative boundary, while the latter does not. To address this mismatch problem, the NPME first retrieves the matched OSM edges of the incident links and constructs allowable paths among them. Then, the NLM node is projected on the paths and a new OSM supernode at the center of mass of projection points is inserted (See OSM node v_3^* in Figure 11(b)). Figure 10 shows two OSM supernodes v_i^* and v_j^* that are matched with $\overline{n}_i$ and $\overline{n}_j$, respectively.

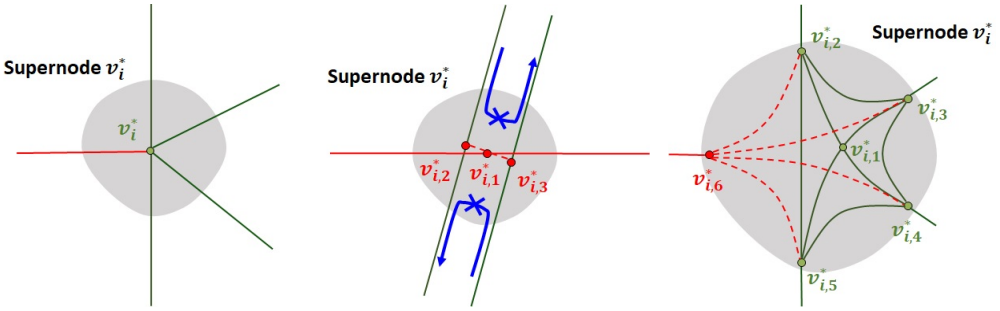
5.2.3. Sequential OSM Subgraph Construction (SOSC) Scheme

During the OSM crowdsourcing process, the whole connected NLM component $\overline{C}_N$ might be omitted due to the misinterpretation of the road network as shown in Figure 4(f). Furthermore, the OSM-SG cannot match an NLM node n if there is no intersection OSM node in the extracted OSM subgraph $\mathcal{G}_I(n)$. In other words, the degree of each node in extracted OSM subgraph $\mathcal{G}_I(n)$ is no more than two. Figure 11(a) shows an example of connected NLM component $\overline{C}_N$ consisting of three unmatched NLM nodes ($\overline{n}_1$, $\overline{n}_2$, and $\overline{n}_3$) interconnected by two unmatched NLM links. There are also 14 unmatched NLM links across the boundary of $\overline{C}_N$, each of which is connected to an NLM node in matched NLM subgraph $\mathcal{G}_N^*$. The objective of the SOSC scheme is to construct a simple OSM subgraph that corresponds to NLM component $\overline{C}_N$.

The basic idea of the SOSC scheme is to sequentially examine an unmatched NLM node in $\overline{C}_N$, and for each unmatched NLM node to construct the corresponding OSM subgraph using both OEI and NPME schemes. This scheme maintains priority queue Q that determines the order of unmatched NLM nodes sequentially extracted from $\overline{C}_N$. To better associate with the neighbor NLM nodes in matched NLM subgraph $\mathcal{G}_N^*$, the key k_i of unmatched NLM node $\overline{n}_i$ in priority queue Q is defined as the ratio of unmatched neighbor NLM nodes to all neighbor NLM nodes $|\mathcal{L}(\overline{n}_i)|$. Since the three key values are $k_1 = \frac{1}{3}$, $k_2 = \frac{1}{2}$, and $k_3 = \frac{1}{4}$ in Figure 11(a), they are extracted from Q in the order of $\overline{n}_3 \rightarrow \overline{n}_1 \rightarrow \overline{n}_2$.

At each extraction of unmatched NLM node $\overline{n}_i$ from priority queue Q, the SOSC scheme first investigates the existence of an OSM edge that is already associated with the NLM links (l_{ij} and/or l_{ji}) at the NVAD boundary of its matched neighbor NLM node $n_j \in \mathcal{N}(\overline{n}_i) \cap \mathcal{N}^*$. Figure 11(a) illustrates two examples of this OSM edge: a dual carriage edge between two neighbor supernodes v_6^* and v_8^*, and a terminating edge between OSM node v_1 and supernode v_4^*. Then, the SOSC scheme inserts a new supernode v_i^* for unmatched NLM node $\overline{n}_i$ using the NPME scheme. For example, supernode v_3^* is inserted at the center of two projection points onto the opposite OSM edges in Figure 11(b), and supernode v_1^* at the projection point in the extended OSM edge in Figure 11(c). Since there is no such OSM edge for unmatched NLM node $\overline{n}_2$ in Figure 11(d), supernode v_2^* is overlaid on NLM node $\overline{n}_2$.

Once supernode v_i^* is obtained, the SOSC scheme uses the OEI scheme to insert a new OSM edge connecting to each neighbor supernode. Finally, the unmatched NLM node $\overline{n}_i$ and a subset of its NLM links that have their corresponding OSM edges are removed from NLM component $\overline{C}_N$, and then inserted to matched NLM subgraph $\mathcal{G}_N^*$. For example, in Figure 11(b), supernode v_3^* has three neighbor supernodes v_6^* v_7^* and v_8^*, where supernodes v_6^* and v_8^* already have their own OSM edges connecting to supernode v_3^*. For the OSM edge between supernodes v_3^* and v_7^*, the OEI scheme shifts, scales, and rotates the blue dotted NLM link in the middle of two parallel NLM links $\overline{l}_{37}$ and $\overline{l}_{73}$. In Figure 11(c), the SOSC scheme also inserts an OSM edge between supernodes v_1^* and v_{10}^*. Since there is no existing OSM edge for unmatched NLM node $\overline{n}_2$ in Figure 11(d), the SOSC scheme needs to insert an OSM edge connecting to every neighbor supernode.



(a) Simple intersection (b) Dual carriage road (c) Complex intersection

Figure 12. Addition of an OSM edge to the existing OSM subgraph of supernode v_i^*

5.2.4. Internal Structure of OSM Supernode

Once a new OSM supernode v_i^* is inserted, we need to design the internal structure of the OSM supernode. Figure 12 shows a few examples of adding a new external OSM edge to the existing OSM subgraph of supernode v_i^*, where the green nodes and edges represent the existing intersection OSM nodes and edges, respectively. In addition, the new OSM nodes and edges are represented by the red nodes and edges, respectively. There are three possible cases for the addition of a new OSM edge: 1) simple intersection, 2) dual carriage road, and 3) complex intersection.

To make the resulting OSM subgraph simple for the first two cases, our exception-handling scheme restricts that all OSM paths through the intersection must intersect at one intersection OSM node. Furthermore, a new relation must be inserted into the OSM dataset in order to reflect a turn restriction between a new OSM edge and an existing OSM edge. Since there is only one intersection OSM node in a simple intersection, the new OSM subgraph for supernode v_i^* is obtained by connecting the new OSM edge to the intersection OSM node as shown in Figure 12(a). In the previous section, the supernode v_i^* for dual carriage road is placed in the middle of two parallel OSM edges. In Figure 12(b), our scheme overlays an intersection OSM node $v_{i,1}^*$ to this supernode, and then requires that all additional OSM edges must intersect at this point. To interconnect the dual carriage edges to OSM node $v_{i,1}^*$, it also inserts two internal (red dashed) OSM edges which connect this node and its projection onto two opposite OSM edges, i.e. intersection OSM nodes $v_{i,2}^*$ and $v_{i,3}^*$. To avoid the u-turns via new internal OSM edges, it is also required to add an additional OSM relation that restricts the u-turns between two dual carriage edges.

However, it is not easy to define a single intersection OSM node for connecting all OSM edges in a complex intersection due to the wide diversity of its internal structure. Figure 12(c) shows an example of OSM subgraph for complex intersection, where the set of intersection OSM nodes are partitioned into two subsets: 1) the subset $\mathcal{V}_{i,C}^*$ of *core* intersection OSM nodes whose all OSM edges are connecting to the other intersection OSN nodes, and 2) the subset $\mathcal{V}_{i,B}^*$ of *boundary* intersection OSM nodes having at least one OSM edge that connects to a neighbor supernode. For example, $\mathcal{V}_{i,C}^* = \{v_{i,1}^*\}$ and $\mathcal{V}_{i,B}^* = \{v_{i,2}^*, v_{i,3}^*, v_{i,4}^*, v_{i,5}^*\}$ in Figure 12(c). In order to add a new OSM edge regardless of the internal structure of the existing OSM subgraph, the exception-handling scheme first adds a new boundary intersection OSM node $v_{i,6}^*$, and then add a new (red dashed) OSM edge that directly connects this new node with every other boundary intersection OSM node. To reflect a turn restriction between a new OSM edge and an existing OSM edge, a new relation should be inserted into the OSM dataset similarly to the previous two cases.

5.2.5. Conversion of NLM Attributes into OSM Tags

The final step after the OSM nodes and edges are created is to convert the attributes of the NLM node and link into the appropriate tags of the corresponding OSM node and

Table 4. Mapping between NLM (*road_rank, connect*) attributes and OSM *highway* tag

road_rank	*connect*	KMF [42]	Statistical Data	
			MM-HT (Prob)	SMM-HT (Prob)
102	000	*trunk*	*primary* (0.53)	*primary_link* (0.33)
	102	*trunk_link*	*trunk_link* (0.74)	*secondary* (0.23)
103	000	*primary*	*primary* (1.0)	
	103	*primary_link*	*primary_link* (1.0)	
107	000	*tertiary*	*secondary* (0.42)	*primary* (0.31)
	107	*tertiary_link*	*secondary* (0.65)	*primary* (0.24)

way that contains the edge, respectively. For all matched road objects by the S-RNM, we first examine the statistical characteristics of mapping between NLM attributes and OSM tags. For each newly matched road object by the exception-handling scheme, we suggest the mapping strategy between NLM attribute and OSM tag using the Korean map feature [42].

We observe that an NLM node has a valid entry for every NLM attribute, such as *node_id, X, Y, node_type, road_name,* and *turn_p*, in the NLM dataset, where X and Y corresponds to longitude and the latitude, respectively. In addition, an OSM node has a non-empty entry for a few basic OSM tags, such as *node_id, lat, lon,* and *node_degree*, in the OSM dataset. In other words, only a few geographical and topological NLM attributes are sufficient to fill out all non-empty OSM tags. As a result, the exception-handling scheme simply converts these NLM attributes to the corresponding OSM tags.

Similarly, an NLM link has a non-empty entry for all NLM link attributes, such as *link_id, f_node, t_node, road_name, road_use, lanes, road_rank, connect, max_speed,* and *multi_link*, while an OSM way has three mandatory tags *link_id, name,* and *highway,* as well as a few optional tags, such as *oneway, layer, bridge, tunnel*. It is straightforward to fill out the OSM way tag, if there are two one-to-one correspondence between NLM link attribute and OSM way tag(s): $(road\_name \leftrightarrow name)$, $(road\_use \leftrightarrow oneway)$, and $(road\_type \leftrightarrow (layer, bridge, tunnel))$.

In the NLM dataset, all NLM links in Yeoui-do are classified into three categories depending on *road_rank* attribute: urban expressway (102), general national road (103), and city and country road (107). Table 4 lists the mapping between NLM (*road_rank, connect*) attributes and OSM *highway* tag over all matched road objects by the S-RNM, where KMF represents the Korean map feature of OSM in [42], and MM-HT and SMM-HT stand for most matched *highway* tag and second MM-HT from statistical data, respectively. In addition, an NLM link with $connect = 000$ is a normal NLM link, while one with $connect = road\_rank$ is a connecting NLM link. We observe that the statistical data can be much different from the KMF, except for general national road, possibly due to the misinterpretation of detailed road attributes during the OSM crowdsourcing process. Taking into account this limitation, the exception-handling scheme chooses the OSM *highway* tag based on the KMF assignment.

6. Numerical Results

In this section, we compare the numerical results of the OSM-SG algorithm and the whole S-RNM framework with three node-matching-based schemes and two edge-matching-based schemes. The node-matching-based schemes include:

- The distance-based greedy matching (DGM): For each NLM node, the DGM first finds a set of candidate OSM nodes within 100 m. Then, it applies the greedy algorithm that sequentially selects the pair of NLM node and its candidate OSM node with shortest distance between them.
- The connectivity-based matching (CM) [31]: The CM scheme extends the DGM by post-correcting the matching of an NLM node using the matching of its neighbor NLM nodes to improve the topological consistency between NLM and OSM sub-

Table 5. Statistical description of NLM and OSM datasets in Yeouido area.

Datasets	Spatial extent	Number of intersection nodes	Number of edges	Total road length
NLM	3.5 km x 2.8 km	178	449	124.74 km
OSM (Pruned)	3.5 km x 2.8 km	590	1005	140.67 km

graphs. Particularly, it re-selects the corresponding OSM node as the node which is adjacent to the most corresponding OSM nodes of the neighbor NLM nodes.

- The CM combining with an extended DBSCAN (density-based spatial clustering of applications with noise) algorithm (CM-EDBSCAN): Since the CM does not account for the LoD difference at the intersection, we combine it with an extended DBSCAN algorithm proposed in [27]. The extended DBSCAN algorithm finds the OSM nodes in a complex intersection by clustering the OSM nodes using the road network distance on $\mathcal{G}_I$ graph as described in section 2.

The two edge-matching-based schemes address the similar RNM problem between the OSM dataset and an authoritative datasets. They include:

- Line-segment-based matching (LSBM) [40]: This scheme decomposes each edge into line segments and performs the matching based on them. For each line segment in the reference dataset, the one in target within a given distance is selected as the candidate. Among them, the corresponding line segments are determined by comparing the angle difference.
- Polygon-based matching (PBM) [39]: This scheme searches for candidate edges by mapping the edges to the polygon representing the urban block in each dataset. Firstly, the polygons in two datasets are matched and the edges on each pair of corresponding polygons are considered as the candidate for edge matching. The corresponding edges are decided based on the sampling-based distance between them.

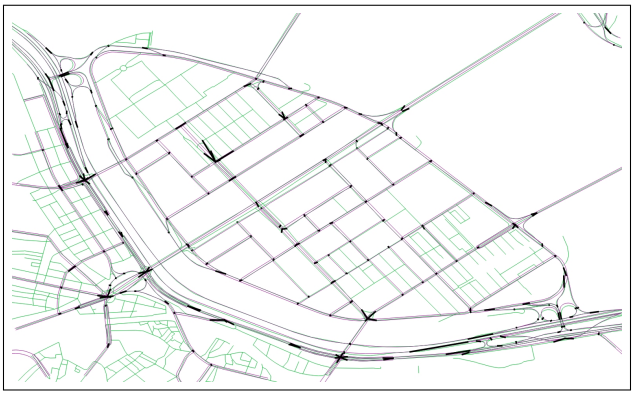
In the OSM-SG and S-RNM, the threshold δ to select the candidate OSM edges of a link is set as 34 m which is the maximum width of the general highway and local road in Korea [3].

The OSM dataset of the Yeoui-do area is extracted from the OSM website [43] on Nov. 17th, 2017 in the form of an XML file, and the NLM dataset is downloaded from the Korean ITS website [4] at the same time in the form of the shape file. Two datasets are imported to PostgreSQL database [44] for processing. The statistical descriptions of each dataset of the Yeoui-do area are shown in Table 5.

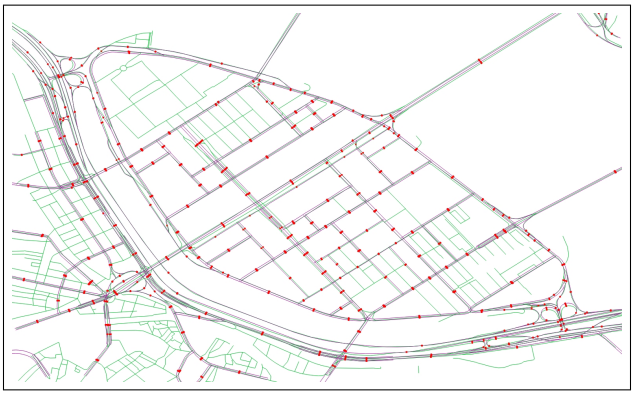
Figure 13 shows the overall matching of the S-RNM in which the black bold line in Figure 13(a) represents the node matching between the NLM nodes and the OSM nodes and the red bold line in Figure 13(b) is the edge matching between NLM links and the OSM edges. In the following sections, we first evaluate the performance of the S-RNM in terms of precision, recall, and F-score as suggested in [39,45]. Since the OSM-SG and S-RNM are the node-matching-based scheme, we focus on the analyzing the node matching result in detail. Particularly, we category that matching result into correct, incomplete, incorrect, and unmatched to analyze the features of the S-RNM. Finally, we examine the goodness of the OSM-SG in terms of missing and redundant OSM nodes in OSM subgraph.

6.1. Precision and Recall

In this section, we present the main performance metrics of RNM: precision, recall, and F-score as suggested in [39,45]. The precision indicates accuracy of a matching scheme and is defined as the ratio between the number of NLM nodes/links that are correctly matched over the total number of correctly and incorrectly matched NLM nodes/links. The recall indicates the completeness of a matching scheme and is defined



(a) Node matching results (Black line).



(b) Edge matching results (Red line).

Figure 13. Overall matching results of the S-RNM.

as the ratio between the number of correctly matched NLM nodes/links over the total number of correctly matched and unmatched NLM nodes/links. A good RNM scheme should achieve both high precision and high recall. To represent this combination, their harmonic mean, also known as F-score, is used. The formula of these three performance metrics are given as follows:

$$Precision = \frac{Correct}{Correct + Incorrect} \tag{4}$$

$$Recall = \frac{Correct}{Correct + Unmatched} \tag{5}$$

$$F - score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{6}$$

First, we present the result of these metrics in node matching among the OSM-SG, S-RNM, and the three other node-matching-based schemes. Later, the result of edge matching is also discussed among OSM-SG, S-RNM, and all other schemes.

6.1.1. Node Matching

In node matching, the correctly matched NLM node n_i is defined as the NLM node matched to the OSM subgraph $\mathcal{G}_I(n_i)$ that is exactly identical to the true corresponding OSM subgraph $\mathcal{G}_I^T(n_i)$. A redundant or missing OSM objects in OSM subgraph is considered as incorrect matching.

Figure 14 shows the precision, recall, and F-score of the node matching. We can observe that the OSM-SG and S-RNM outperform the other schemes in node matching precision. The OSM-SG improves the precision by at least 31 % compared to the DGM, CM, and CM-EDBSCAN. Particularly, the OSM-SG and S-RNM schemes achieve the

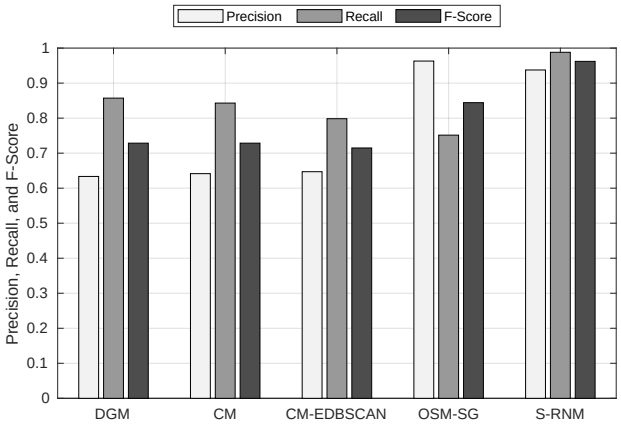
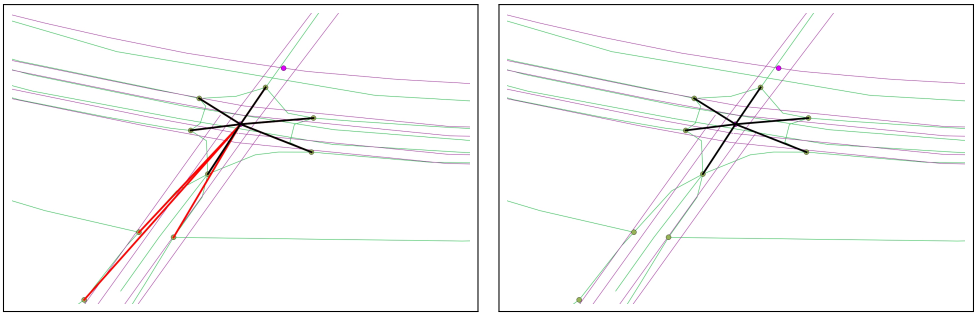


Figure 14. Precision, recall, and F-score of node matching.



(a) CM-EDBSCAN. (b) S-RNM.

Figure 15. Examples of overly grouping by CM-EDBSCAN and correctly matching in the S-RNM. The correct matching is represented by black line while the wrong matching is represented by red line.

precision of 0.94 and 0.96, respectively. The DGM is the scheme with the lowest precision as we expected. The CM improves the precision by only 0.01 compared to the DGM. The CM-EDBSCAN is expected to have a competitive matching precision with the OSM-SG since both schemes address the LoD dissimilarity at complex intersection. However, compared to the CM, the CM-EDBSCAN only increases precision by 0.01. The reason is that the CM-EDBSCAN can correctly group several OSM subgraphs but also produces many incorrect ones. The CM-EDBSCAN usually overly expands the group to the OSM nodes that do not belong to the corresponding OSM subgraph as shown in Figure 15. In this example, the CM-EDBSCAN incorrectly groups three OSM nodes belonging to the minor intersections (red lines) with the OSM nodes belonging to the complex intersection at the center (black lines). This problem does not exist in the S-RNM as shown in figure 15(b).

In figure 14, we can also observe that there is a trade-off between precision and recall in the first four schemes. The precision increases while the recall decreases among the DGM, CM, CM-EDBSCAN, and OSM-SG. However, the decreasing in the recall in the OSM-SG is much smaller than the increasing in the precision. Particularly, comparing to the DGM, CM, and CM-EDBSCAN, the OSM-SG has at most 0.11 lower matching recall, however, this amount is much less than the improvement in the precision of the OSM-SG which is at least 0.31 as mentioned before. This fact leads to the higher F-score in the OSM-SG scheme comparing to DGM, CM, and CM-EDBSCAN. With the exception-handling schemes, the S-RNM can improve the matching recall to 0.99 which is the highest among all schemes. Therefore, the S-RNM achieves the highest F-score of 0.96 and is the best scheme in terms of both precision and recall.

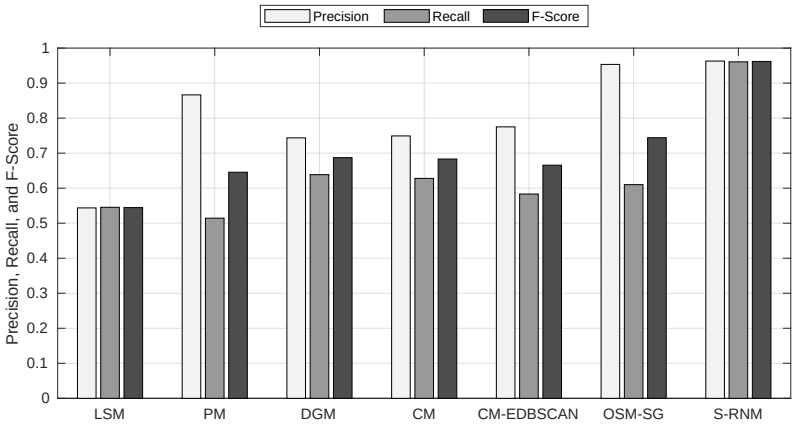


Figure 16. Precision, recall, and F-score of edge matching.

6.1.2. Edge Matching

In this section, the edge matching of the node-matching-based schemes including DGM, CM, and CM-EDBSCAN, is obtained as following. First, for the pair of NLM nodes of an NLM link, two corresponding OSM subgraphs are retrieved. Then, all the candidate OSM paths between these two OSM subgraphs are constructed. The path having the smallest sampling distance with NLM link is chosen as the corresponding OSM path. To compute the sampling distance, 10 equidistant points are sampled on both NLM link and OSM path. The sampling distance is the mean distance between each corresponding pair of equidistant points.

Figure 16 shows the precision, recall, and F-score of the edge matching. This figure shows the advantages of the node-matching-based schemes over the edge-matching-based schemes in the F-score. The PBM has the lowest recall since this scheme relies on the matching of the urban block which does not exist near the highway road. However, this scheme has relatively high precision comparing to the node-matching-based schemes DGM, CM, and CM-EDBSCAN thanks to the intermediate matching of the urban polygons. The LSBM achieves the higher recall than PM but has lower precision. This scheme matches the NLM link and OSM way based on their line segments, therefore, a single line segment of a link that is matched to the wrong OSM way can lead to the incorrect matching of a whole link.

Since the edge matching in the DGM, CM, and CM-EDBSCAN is based on the node matching, the result of the edge matching in these schemes shows the similar trend with the node matching. We can observe that the OSM-SG outperforms the DGM, CM, and CM-EDBSCAN in the precision and F-score while the S-RNM outperforms in all three metrics. In general, the S-RNM is the best schemes with the precision, recall, and F-score as high as 0.96.

6.2. Detailed Matching Results

In order to find out how the S-RNM achieves good precision and recall, we discuss another aspect of the matching results by classifying the matching of NLM nodes into correct, incorrect, incomplete, and unmatched. The correctly matched NLM node is defined in Section 6.1. The incorrectly matched NLM node is the NLM node matched to the OSM subgraph $\mathcal{G}_I(n_i)$ which has at least one OSM node $v_j \in \mathcal{G}_I(n_i)$ and $v_j \notin \mathcal{G}_I^T(n_i)$. On the other hand, the incompletely matched NLM node n_i has $\mathcal{G}_I(n_i) \subset \mathcal{G}_I^T(n_i)$. The unmatched NLM node is the NLM node matched to null OSM subgraph $\mathcal{G}_I(n_i)$.

Figure 17 shows the ratio of matching in each category for all node-matching-based schemes. First, we find out why the OSM-SG has high precision but low recall. It can be observed that the high precision of the OSM-SG comes from the fact that it has corrected the majority of incomplete and incorrect matching caused by the LoD difference at a complex intersection. This result proves the key advantage of the OSM-SG in addressing

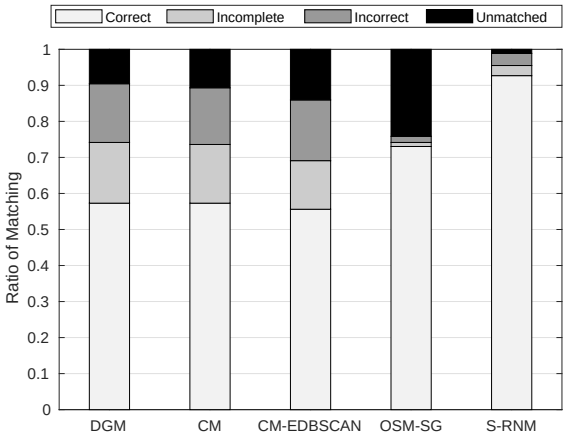


Figure 17. The number of correct, incorrect, incomplete, and unmatched NLM nodes.

the LoD difference at the intersection. On the other hand, the low recall of the OSM-SG comes from the fact that the OSM-SG isolates the incomplete and incorrect matchings into the unmatched cases. This is another advantage of the OSM-SG rather than a drawback. It is also the key requirement to apply the exception-handling schemes. Even though targeting the same LoD problem, these advantages do not exist in the CM-EDBSCAN where the ratios of incomplete and incorrect matching are almost similar to the DGM and CM schemes. Second, The result in figure 17 also shows that the exception-handling schemes can successfully insert the correct OSM subgraph for correspondent missing NLM node. Particularly, compared to the OSM-SG, the S-RNM increases the correct matching by 0.2 and reduces the unmatched case by 0.23.

6.3. Normalized Numbers of Redundant and Missing OSM Nodes

To examine the advantages of the OSM-SG algorithm in the OSM dataset aspect, we propose two performance metrics called the normalized number of redundant OSM nodes (NNRON) and normalized number of missing OSM nodes (NNMON). The redundant OSM node v_j of the identified OSM subgraph $\mathcal{G}_I(n_i)$ is the OSM node belonging to $\mathcal{G}_I(n_i)$ but not belonging to the truth OSM subgraph $\mathcal{G}_I^T(n_i)$, i.e. $v_j \in \mathcal{G}_I(n_i) \setminus \mathcal{G}_I^T(n_i)$. Then, the NNRON is defined as the total number of redundant OSM nodes of all NLM nodes divided by the total number of NLM nodes. On the other hand, the missing OSM node v_k is the OSM node not belonging to $\mathcal{G}_I(n_i)$ but belonging to the truth OSM subgraph $\mathcal{G}_I^T(n_i)$, i.e. $v_j \in \mathcal{G}_I^T(n_i) \setminus \mathcal{G}_I(n_i)$. The NNMON is similarly defined as the total number of missing OSM nodes of all NLM nodes divided by the total number of NLM nodes. These two metrics assess how accurate the node-matching-based schemes group the OSM nodes in the subgraph.

Figure 18 shows the NNRON and NNMON of the five node-based matching schemes. We can observe that the OSM-SG and S-RNM achieve at least 5.8 times lower NNRON and 8 times lower NNMON compared to other schemes. All the other schemes including CM-EDBSCAN have the NNMON of nearly 1, i.e., for average, they miss 1 OSM nodes for each matched NLM node including the simple intersection NLM node. The CM-EDBSCAN has only 0.11 lower in the NNMON compared to the DGM and CM even though it also considers the OSM grouping at the intersection like the OSM-SG and S-RNM. Especially, the CM-EDBSCAN has the highest NNRON of 0.88. It confirms our statement about CM-EDBSCAN in the previous section that it overly expands the OSM subgraph. From this result, we can conclude that our OSM-SG is essential in the RNM between NLM and OSM datasets.

7. Conclusions

This paper proposes the S-RNM approach between NLM and OSM datasets. To remove the LoD dissimilarity at the complex intersection, the OSM-SG algorithm is

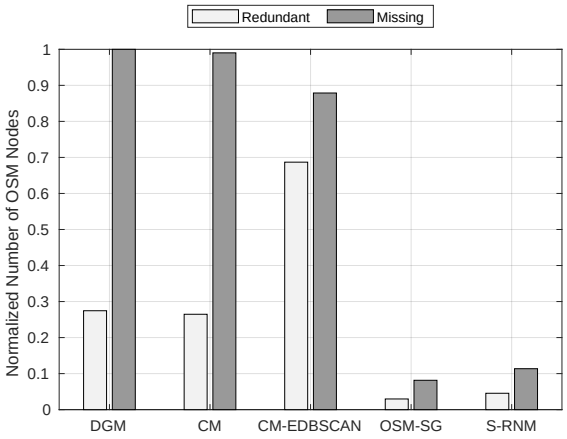
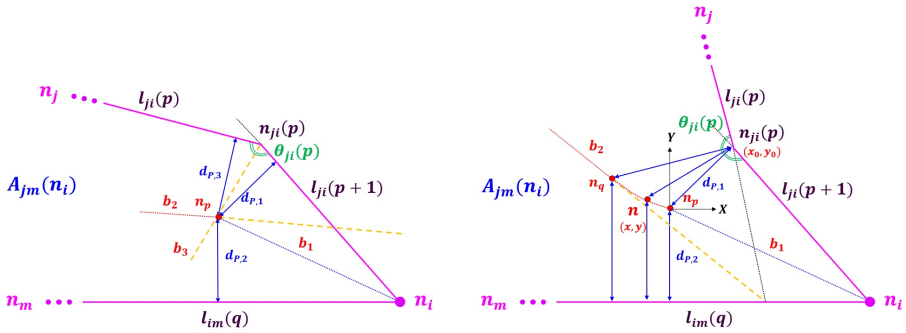


Figure 18. Normalized number of redundant OSM nodes (NNRON) and normalized number of missing OSM node (NNMON).



(a) $\theta_{ji}(p) < 180°$ (b) $\theta_{ji}(p) > 180°$

Figure A1. Construction of projection boundary in map area $A_{jm}(n_i)$

proposed to identify and aggregate the detailed connectivity of the corresponding OSM subgraph of an NLM node. The exception-handling schemes are proposed to address the matching of nodes near the acceleration/deceleration lane and to insert the missing corresponding OSM objects of NLM objects and create the appropriate structure and object attribute inside of OSM subgraph. The numerical results show that our S-RNM achieves much higher matching accuracy and cardinality than the existing RNM approaches. They also show that the OSM-SG is an essential component in the RNM between NLM and OSM datasets.

Appendix A Transient Curve of Projection Boundary

In this appendix, we identify the transient curve of projection boundary around a vertex in an intersection area. Figure A1 shows two examples of projection boundary in area $A_{jm}(n_i)$, where vertex $n_{ji}(p)$ connects two NLM line segments $l_{ji}(p)$ and $l_{ji}(p+1)$ in one projection side, and NLM line segment $l_{im}(q)$ is common on the other projection side. Starting from NLM node n_i, the projection boundary is the bisector b_1 of the angle created by $l_{ji}(p+1)$ and $l_{im}(q)$, and is illustrated by the blue dotted line in both examples. It is clear that every point on this projection boundary should have the same projection distance to l_{ji} and l_{im}, e.g. $d_{P,1} = d_{P,2}$. Our goal is to determine the point where the projection boundary deviates from b_1 and find the equidistant projection boundary between two NLM line segments $l_{ji}(p)$ and $l_{im}(q)$. Without loss of generality, we examine the projection boundary curve in two different cases: 1) The internal angle of vertex $n_{ji}(p)$ is less than 180° ($\theta_{ji}(p) < 180°$); and 2) It is greater than 180° ($\theta_{ji}(p) > 180°$).

Figure 1(a) shows an example where $\theta_{ji}(p) < 180°$. To find the point where projection boundary deviate from b_1, we draw two additional bisectors that intersect with bisector b_1 at point n_p: bisector b_2 of the angle between $l_{ji}(p+1)$ and $l_{im}(q)$ and bisector b_3 of angle $\theta_{ji}(p)$. At point n_p, the projection distance to NLM line segments $l_{ji}(p)$, $l_{ji}(p+1)$, and $l_{im}(q)$ becomes the same. After point n_p, the projection boundary deviates from b_1 and becomes the red dotted line segment b_2.

When $\theta_{ji}(p) > 180°$ as shown in Figure 1(b), bisector b_2 is similarly obtained from the crosspoint of $l_{im}(q)$ and the extended line of $l_{ji}(p)$. Next, we determine point n_q on bisector b_2 so that its distance to point $n_{ji}(p)$ is equal to the projection distance to $l_{im}(q)$. It is clear that, beyond point n_q, bisector b_2 becomes the projection boundary. The remaining problem is to determine the projection boundary between points n_p and n_q. To address this problem, we first define a Cartesian coordinate whose X-axis crossing at the origin point n_p is parallel to $l_{im}(q)$. We denote the Cartesian coordinate of point n on the transient boundary curve by (x, y). Similarly, the Cartesian coordiantes of point $n_{ji}(p)$ is denoted by (x_0, y_0). Since $y > 0$, the projection distance of point n to $l_{im}(q)$ becomes $y + d_{P,2}$ which must be equal to the distance between points n and $n_{ji}(p)$, i.e.,

$$\sqrt{(x - x_0)^2 + (y - y_0)^2} = y + d_{P,2}. \tag{A1}$$

Finally, the transient curve of projection boundary becomes a *parabola* satisfying the following equation:

$$y = \frac{(x - x_0)^2 + y_0^2 - d_{P,2}^2}{2(y_0 + d_{P,2})}. \tag{A2}$$

References

1. Geography, G. 1000 GIS Applications and Uses – How GIS Is Changing the World. https://gisgeography.com/gis-applications-uses/, accessed on 2021-12-02.
2. Clarke, K.C. Advances in Geographic Information Systems. *Computers, Environment and Urban Systems* **1986**, *10*, 175–184. doi:10.1016/0198-9715(86)90006-2.
3. Korean Ministry of Land Infrastructure and Transport. The Road Extension per Person in Korea is 2 Meters. Technical report, 2019.
4. Korean National Transport Information Center. Node Link Map. http://nodelink.its.go.kr/, accessed on 2021-12-02.
5. Google LLC. Google Map. http://maps.google.com/, accessed on 2021-12-02.
6. Daum Kakao. Daum map. http://map.daum.net/, accessed on 2021-12-02.
7. OpenStreetMap Contributors. OpenStreetMap Wiki. http://wiki.openstreetmap.org/, accessed on 2021-12-02.
8. OpenStreetMap Contributors. OpenStreetMap Wiki - Stats. https://wiki.openstreetmap.org/wiki/Stats, accessed on 2021-12-02.
9. Yuan, M.; Yang, Z.; Niu, S. Study on dynamic route guidance method of vehicle based on travel time reliability. *Proceedings - 2nd IEEE International Conference on Advanced Computer Control, ICACC 2010* **2010**, *1*, 292–295. doi:10.1109/ICACC.2010.5487007.
10. Hajiahmadi, M.; Knoop, V.L.; De Schutter, B.; Hellendoorn, H. Optimal dynamic route guidance: A model predictive approach using the macroscopic fundamental diagram. 16th International IEEE Conference on Intelligent Transportation Systems (ITSC 2013). IEEE, 2013, number Itsc, pp. 1022–1028. doi:10.1109/ITSC.2013.6728366.
11. Zuo, L.; Zhang, N.; He, Y.; Zhou, T. Research on Dynamic Route Guidance and Navigation System Based on Multi Information Feedback. *Proceedings - 2017 International Conference on Sensing, Diagnostics, Prognostics, and Control, SDPC 2017* **2017**, *2017-Decem*, 421–424. doi:10.1109/SDPC.2017.86.
12. Lin, J.; Yu, W.; Yang, X.; Yang, Q.; Fu, X.; Zhao, W. A Real-Time En-Route Route Guidance Decision Scheme for Transportation-Based Cyberphysical Systems. *IEEE Transactions on Vehicular Technology* **2017**, *66*, 2551–2566. doi:10.1109/TVT.2016.2572123.
13. Yang, Q.; Koutsopoulos, H.N.; Ben-Akiva, M.E. Simulation laboratory for evaluating dynamic traffic management systems. *Transportation Research Record* **2000**, pp. 122–130. doi:10.3141/1710-14.
14. Van Hinsbergen, C.P.; Schreiter, T.; Zuurbier, F.S.; Van Lint, J.W.; Van Zuylen, H.J. Localized extended kalman filter for scalable real-time traffic state estimation. *IEEE Transactions on Intelligent Transportation Systems* **2012**, *13*, 385–394. doi:10.1109/TITS.2011.2175728.
15. Zygouras, N.; Zacheilas, N.; Kalogeraki, V.; Kinane, D.; Gunopulos, D. Insights on a scalable and dynamic traffic management system. EDBT 2015 - 18th International Conference on Extending Database Technology, Proceedings, 2015, number March, pp. 653–664. doi:10.5441/002/edbt.2015.65.
16. Korean Land Information Platform. Precise Road Map. http://map.ngii.go.kr/ms/pblictn/preciseRoadMap.do, accessed on 2021-12-02.

17. Korean National Transport Information Center. Public Traffic Information Service. http://openapi.its.go.kr/, accessed on 2021-12-02.
18. Fairhurst, R. Potlatch 2. https://wiki.openstreetmap.org/wiki/Potlatch_2, accessed on 2021-12-02.
19. Scholz, I.; Stöcker, D. JOSM. https://josm.openstreetmap.de/, accessed on 2021-12-02.
20. Pavlenko, A. Mapnik. http://mapnik.org/, accessed on 2021-12-02.
21. Ramm, F.; Topf, J. Tirex. https://wiki.openstreetmap.org/wiki/Tirex, accessed on 2021-12-02.
22. Quinion, B.; Hoffmann, S.; Metten, M.T. Nominatim. https://nominatim.org/, accessed on 2021-12-02.
23. Luxen, D.; Vetter, C. Real-time routing with OpenStreetMap data. *Proceedings of the 19th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems - GIS '11* **2011**, p. 513. doi:10.1145/2093973.2094062.
24. Gearhart, D.; Knisely, G.; Kreiser, K.; DiLuca, K.; Nesbitt, D. Valhalla. https://github.com/valhalla, accessed on 2021-12-02.
25. SinghSehra, S.; Singh, J.; Singh Rai, H. Assessment of OpenStreetMap Data - A Review. *International Journal of Computer Applications* **2013**, *76*, 17–20, [1309.6608]. doi:10.5120/13331-0888.
26. Zhang, M. Methods and implementations of road-network matching. *PhD Dissertation* **2009**.
27. Yang, B.; Luan, X.; Li, Q. Generating hierarchical strokes from urban street networks based on spatial pattern recognition. *International Journal of Geographical Information Science* **2011**, *25*, 2025–2050. doi:10.1080/13658816.2011.570270.
28. Yang, B.; Luan, X.; Zhang, Y. A pattern-based approach for matching nodes in heterogeneous urban road networks. *Transactions in GIS* **2014**, *18*, 718–739. doi:10.1111/tgis.12057.
29. Walter, V.; Fritsch, D. Matching spatial data sets: a statistical approach. *International Journal of Geographical Information Science* **1999**, *13*, 445–473. doi:10.1080/136588199241157.
30. Okabe, A.; Boots, B.; Sugihara, K.; Chiu, S.N.; Kendall, D. *Spatial Tessellations: Concepts and Applications of Voronoi Diagrams*; Wiley Series in Probability and Statistics, John Wiley & Sons, Inc.: Hoboken, NJ, USA, 2000. doi:10.1002/9780470317013.
31. Filin, S.; Doytsher, Y. Detection of corresponding objects in linear-based Map conflation. *Surveying and land information systems* **2000**, *60*, 117–128.
32. Xiong, D.; Sperling, J. Semiautomated matching for network database integration. *ISPRS Journal of Photogrammetry and Remote Sensing* **2004**, *59*, 35–46. doi:10.1016/j.isprsjprs.2003.12.001.
33. Zhang, M.; Shi, W.; Meng, L. A generic matching algorithm for line networks of different resolutions. *Proceedings of 8th ICA Workshop on Generalisation and Multiple Representation* **2005**, pp. 1–8.
34. Volz, S. An Iterative Approach for Matching Multiple Representations of Street Data. *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences* **2006**, *36*, 101–110.
35. Lüscher, P.; Burghardt, D.; Weibel, R. Matching road data of scales with an order of magnitude difference. *Proc. XXIII International Cartographic Conference, Moscow, Russia* **2007**, pp. 1–11.
36. Krishnamurthy, G.; Devarajan, V.; Dragan, I. Nonrigid conflation for vector datasets using EM algorithm and mixture of Gaussian approach. *International Geoscience and Remote Sensing Symposium (IGARSS)* **2010**, pp. 3964–3967. doi:10.1109/IGARSS.2010.5654391.
37. Anand, S.; Morley, J.; Jiang, W.; Du, H.; Hart, G. When worlds collide : combining Ordnance Survey and Open Street Map data. AGI Geocommunity '10, 2010.
38. Yang, B.; Zhang, Y.; Luan, X. A probabilistic relaxation approach for matching road networks. *International Journal of Geographical Information Science* **2013**, *27*, 319–338. doi:10.1080/13658816.2012.683486.
39. Fan, H.; Yang, B.; Zipf, A.; Rousell, A. A polygon-based approach for matching OpenStreetMap road networks with regional transit authority data. *International Journal of Geographical Information Science* **2016**, *30*, 748–764. doi:10.1080/13658816.2015.1100732.
40. Brovelli, M.A.; Minghini, M.; Molinari, M.; Mooney, P. Towards an Automated Comparison of OpenStreetMap with Authoritative Road Datasets. *Transactions in GIS* **2017**, *21*, 191–206. doi:10.1111/tgis.12182.
41. Korean Ministry of Land Infrastructure and Transport. Spatial Information Portal. https://www.nsdi.go.kr, accessed on 2021-12-02.
42. OpenStreetMap. Ko: Map Features. https://wiki.openstreetmap.org/wiki/Ko:Map_Features, accessed on 2021-12-02.
43. OpenStreetMap Contributors. OpenStreetMap. https://www.openstreetmap.org/, accessed on 2021-12-02.
44. PostgreSQL Global Development Group. PostgreSQL. http://www.postgresql.org, accessed on 2021-12-02.
45. Hacar, M.; Gökgöz, T. A new, score-based multi-stage matching approach for road network conflation in different road patterns. *ISPRS International Journal of Geo-Information* **2019**, *8*. doi:10.3390/ijgi8020081.