

# Automatically Building Linking Relations between Lane-Level Map and Commercial Navigation Map Using Topological Networks Matching

Lu Tao<sup>1</sup>, Pan Zhang<sup>2,3</sup>, Lixin Yan<sup>4</sup> and Dunyao Zhu<sup>3,5</sup>

<sup>1</sup>(Graduate School of Informatics, Nagoya University, Japan)

<sup>2</sup>(School of Geodesy and Geomatics, Wuhan University, China)

<sup>3</sup>(Wuhan KOTEI Informatics Co., Ltd, China)

<sup>4</sup>(School of Transportation and Logistics, East China Jiaotong University, China)

<sup>5</sup>(GNSS Research Centre, Wuhan University, China)

(E-mail: [tao.lu@a.mbox.nagoya-u.ac.jp](mailto:tao.lu@a.mbox.nagoya-u.ac.jp))

The lane-level map, which contains the lane-level information severely lacking in widely used commercial navigation maps, has become an essential data source for autonomous driving systems. The linking relations between lane-level map and commercial navigation map can facilitate an autonomous driving system mapping information between different applications using different maps. In this paper, an approach is proposed to build the linking relations automatically. The different topology networks are first reconstructed into similar structures. Then, to build the linking relations automatically, the adaptive multi-filter algorithm and forward path exploring algorithm are proposed to detect corresponding junctions and paths, respectively. The approach is validated by two real data sets of more than 150 km of roads, mainly highway. The linking relations for nearly 94% of the total road length have been built successfully.

## KEYWORDS

1. Lane-Level Map.
2. Commercial Navigation Map.
3. Topological Network Matching.
4. Linking Relations.

Submitted: 8 June 2018. Accepted: 14 April 2020. First published online: 20 May 2020.

1. INTRODUCTION. In recent years, autonomous driving technology has drawn more and more attention. The autonomous vehicle comprises several systems, including perception and planning systems. Underlying these systems there is a public model: the map (Bender et al., 2014). However, current widely used commercial navigation maps cannot meet the demands of high level applications due to their serious lack of accuracy, contents and completeness (Tao et al., 2013), hence the necessity of developing detailed lane-level maps has been widely recognised (Naranjo et al., 2009; Nedevschi et al., 2013). In industry,

lane-level maps have been deployed in autonomous vehicles, including by Google (Guizzo, 2011) and Daimler (Ziegler et al., 2014).

A lane-level map (LLM) that represents the detailed lane information on a road is anticipated to provide higher resolution data to autonomous driving systems. Currently, there are some LLM models available, such as the route network definition file (RNDF, DARPA Urban Challenge, 2007), OpenDRIVE (Dupuis, 2010), Lanelets (Bender et al., 2014), and the smart map (Liu et al., 2015). Meanwhile commercial navigation maps (CNM) are widely used in human driving navigation today. However, CNMs model roads as polylines, which causes loss of information about the road. Intuitively, CNM has ample global attributes that go beyond on-board sensor visibilities and LLM has highly defined local lane-level data. The autonomous driving system inevitably involves both global and local data. For example, the hierarchical planning system, which consists of mission planning and motion planning, has been employed in some autonomous vehicles (Leonard et al., 2008; Fan et al., 2018). Mission planning is performed over the road network to find a user-defined optimal route between origin and destination; motion planning generates an appropriate path to achieve local objectives (Pendleton et al., 2017). CNM and LLM can offer global and local information to autonomous driving systems and the linking relations between CNM and LLM can greatly facilitate them, especially the hierarchical planning system. For instance, in order to guide an autonomous car, Alonso et al. (2011) proposed a two-level cartographic system to provide different linked map data. Analogously, in the well-known open source autonomous driving platform Autoware, CNM was used to search for a route, then the route was reflected onto a three-dimensional map to guide the autonomous vehicle (Kato et al., 2015). The linking relations between LLM and CNM are of great importance for mapping information between different applications using different map data among autonomous driving systems. They also facilitate data sharing and interoperation between LLM and CNM, such as CNM's error detection and correction via LLM. Unfortunately, due to the different data capturing and processing methods and rules for the different raw data that are used to generate CNM and LLM, it is impossible even for one map supplier to make explicit one-to-one mapping between CNM and LLM. In this paper, an approach is proposed to build the linking relations automatically.

The remainder of this paper is organised as follows. In Section 2, the literature on road networks matching is reviewed. Section 3 gives an introduction and comparison of LLM and CNM. In Section 4, the proposed road networks matching approach is elaborated. The linking relations are built in Section 5. In Section 6, the experiments are conducted and the results are discussed. Finally, Section 7 provides the conclusion and future works.

**2. LITERATURE ON ROAD NETWORKS MATCHING.** In GIS (geographic information systems), geographic data set integration is the process of establishing relations between corresponding object instances in different geographic data sets of a certain region (Uitermark and Cadastre, 1996). Saalfeld (1988) initially proposed an automatic method to consolidate two vector maps. Doytsher et al. (2001) detected counterpart nodes and constructed paths between the nodes at first; then detected the corresponding paths by average distance. Similarly, Volz (2006) detected 'seed nodes' and then matched the emanating edges through a weighted sum function. Due to being governed by many fixed thresholds and weightings, the approach might fail when applied to other map databases; and it was a nontrivial work to determine the thresholds and weightings. Distance is commonly used to

measure the similarity of geometric objects. Mustière and Devogele (2008) used Euclidean and Hausdorff distances to pre-match nodes and arcs. Devogele (2002) cast the process of matching homologous objects as identifying homologous points using Frechet distance. Aiming to speed up the matching process, Safra et al. (2006) detected homologous roads merely relying on the polyline's endpoint location. The distance based methods might be invalid if the maps were of great positional difference. Unlike these approaches, Zhang et al. (2005) chained the line objects that shared the same attribute together to construct reference lines to match road networks. Following corresponding objects detection, some operations were performed to achieve particular goals, such as shifting one map to align with another (Doytsher et al., 2001), improving the positional accuracy (Devogele, 2002), detecting the difference between the datasets (Koukoletsos et al., 2012; Abdolmajidi et al., 2015), and building the linking relations for homologous objects (Volz and Walter, 2004; Volz, 2006).

Although the above approaches matched the networks with different positional accuracy, date and details, etc., there was a common feature of the networks – using a polyline to represent a road. The road networks matching could be regarded as the process of detecting homologous one-dimensional linear objects. However, in LLM, the road is no longer treated as a one-dimensional linear object. To date, there are few available methods designed to cope with automatic networks matching for LLM and CNM.

### 3. LLM AND COMMERCIAL NAVIGATION MAP.

3.1. *LLM.* LLMs have a prominent role not only in improved positional accuracy, but also in higher resolution. LLM represents many geometrical entities that are not present in CNM, such as lane centreline. An instance of LLM is shown in Figure 1 that includes detailed LLM models of the road.

The LLM model is illustrated by Figure 2. *Road* is composed of a sequence of *RoadSections*. Within a *RoadSection*, some attributes of the road remain constant. For example, in Figure 2, *Road1* consists of *RoadSection1* and *RoadSection2*; the driver can change lanes freely in *RoadSection2*, but it is forbidden in *RoadSection1*. *RoadSection* contains all the lanes inside it. *Lane boundaries* and *lane centreline* form the *lane*. *Lane centrelines* are connected at *lane node*. *Lane centreline* and *lane node* form the lane-level network of LLM. The leftmost line of the road is named the *road reference line*, denoted as *r*. In order to maintain the continuity of the road, only one instance of *r* is defined between two adjacent junctions.

3.2. *LLM's cartographic primitives and network.* In LLM, lane node and lane centreline are the cartographic primitives and the atomic construction elements of the network. The lane-level network is a directed graph, denoted as  $TN_{LLM}$ , where lane node is vertex and lane centreline is edge. The sets of lane nodes and lane centrelines are denoted as *LN* and *LCL* respectively, and  $TN_{LLM} = (LN, LCL)$ .

3.3. *CNM.* The node/link model has a significant advantage in supporting navigation and is the prevailing network model for navigation maps (Goodchild, 2000). In CNM, the road is modelled as a link and represented by its centreline. The node is the topological junction of two or more links. The properties of the road network are inherently modelled as attributes of nodes and links. Various navigation oriented data are attached to the CNM network, such as guiding data, addresses and POI (points of interest). Using the centreline

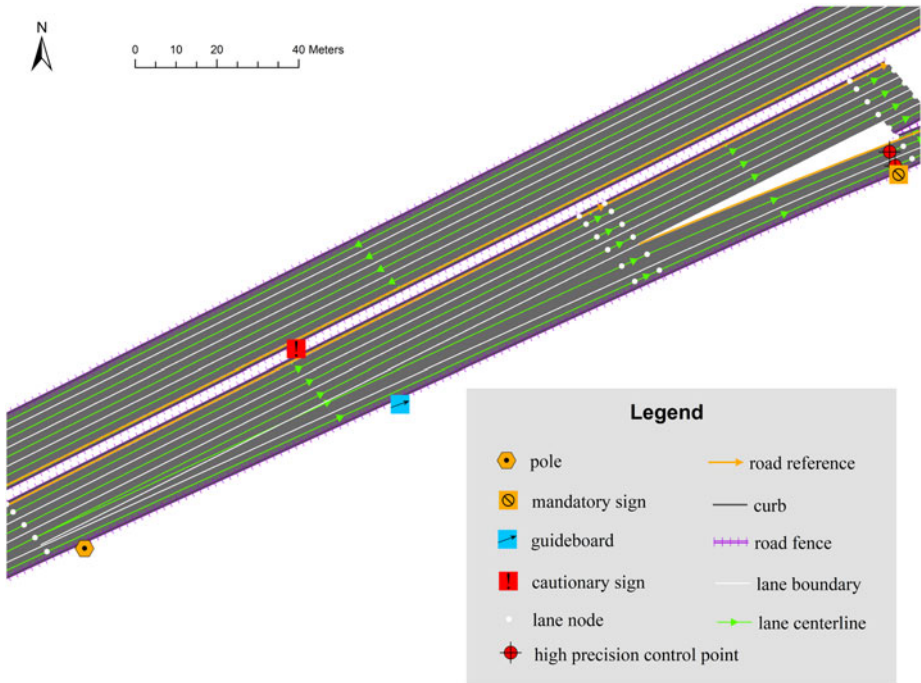


Figure 1. An instance of LLM.

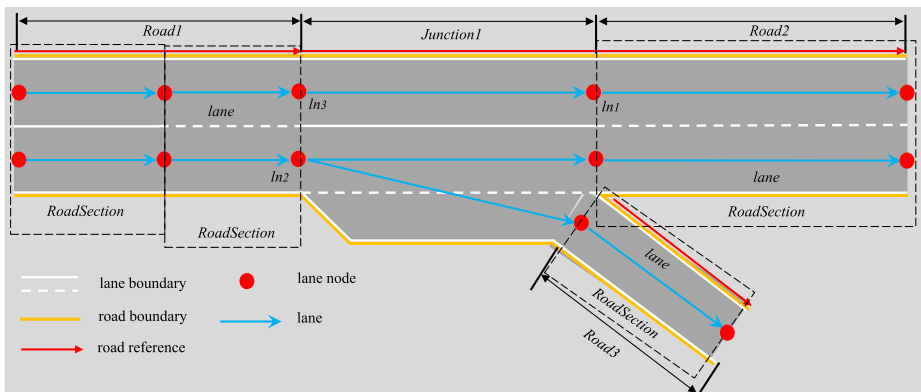


Figure 2. Model of LLM.

to represent the road causes the CNM to lose most lane information. The road section of Figure 2 is simplified in CNM, as Figure 3 shows.

3.4. *CNM's cartographic primitives and network.* The cartographic primitives and atomic construction elements of CNM are node and link, denoted as *node* and *link*; they form the road-level network  $TN_{CNM} = (NODE, LINK)$ , where *NODE* and *LINK* denote the set of nodes and links, respectively.

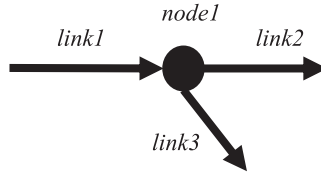


Figure 3. Model of CNM.

Table 1. Comparison of LLM and CNM.

Item	CNM	LLM
Purpose	human driving navigation	autonomous driving
Resolution	road level	lane level
Precision	metre level	centimetre level
Data contents		
Address	✓	×
POI	✓	×
guiding data	✓	×
road marking	×	✓
road attribute	rich	poor
lane attribute	poor	rich
geometry	road level	lane level
topology	road level	lane level
Functions		
global routing	✓	×
lane-level planning	×	✓
ADAS	✓	✓
fusion with sensor data	×	✓
precise positioning assistance	×	✓

3.5. *Comparison of LLM and CNM.* Table 1 summarises the difference between LLM and CNM. It can be found that LLM and CNM mainly differ in purpose, resolution and precision; in data content and function aspects, LLM and CNM are complementary, which gives a strong reason to link them together. The most significant difference between LLM and CNM is their networks. Compared with the road-level network  $TN_{CNM}$ ,  $TN_{LLM}$  is a more complicated lane-level network. Figure 4 illustrates the difference through real data, where the red is  $TN_{CNM}$  and the black is  $TN_{LLM}$ .

4. *MATCHING THE NETWORKS OF LLM AND CNM.* In this paper, a finite connected road sequence is called a path. The junction of LLM and CNM is respectively denoted as *jun* and *cnmjun*. It is hypothesised that there is one and only one path between two adjacent junctions. Based on the hypothesis, the principle of the approach is depicted in Figure 5. For network1 and network2, if the homologous junction pairs could be distinguished, e.g.,  $\langle jun1, cnmjun1 \rangle$ , then the paths between the two adjacent homologous junction pairs, e.g.,  $\langle jun1, cnmjun1 \rangle$  and  $\langle jun2, cnmjun2 \rangle$ , are seen as corresponding, e.g.,  $\langle path1, path1' \rangle$ .

The overall process of the approach is shown in Figure 6, where the solid and dotted arrow indicate the process and the dependence of process, respectively. First,  $TN_{LLM}$  and

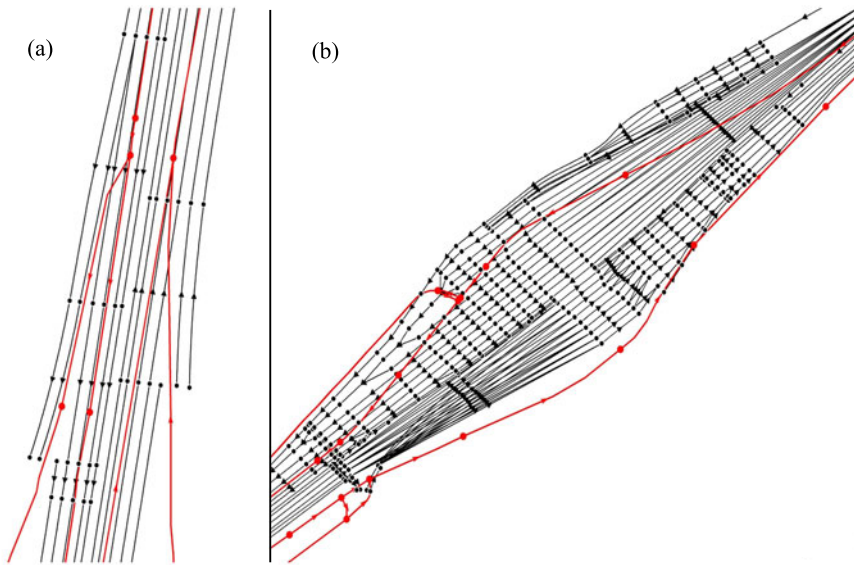


Figure 4.  $TN_{LLM}$  (black) and  $TN_{CNM}$  (red).

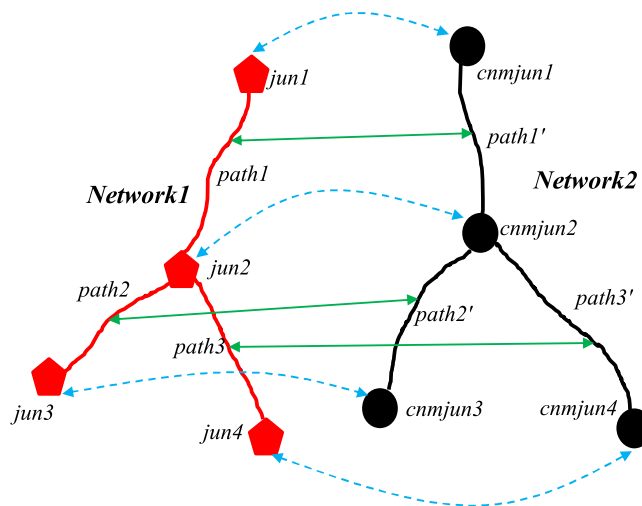


Figure 5. Principle of the proposed approach.

$TN_{CNM}$  are reconstructed into the graphs with common structure (Section 4.1), then corresponding junctions are detected using a strategy called multi-filter (Section 4.2). Following that, in stages 3 and 4, a so-called forward path exploring algorithm is developed to find and validate the corresponding paths (Section 4.3). Finally, the linking relations of networks of LLM and CNM are built automatically (Section 5).

4.1. *Topology reconstruction for  $TN_{LLM}$  and  $TN_{CNM}$ .* Generally, the networks to be matched are preprocessed to build specific structures to pave the way for network matching

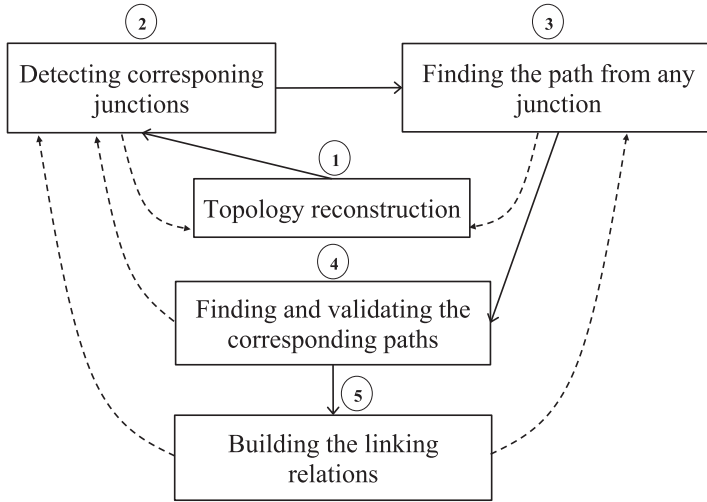


Figure 6. The processes and their dependence.

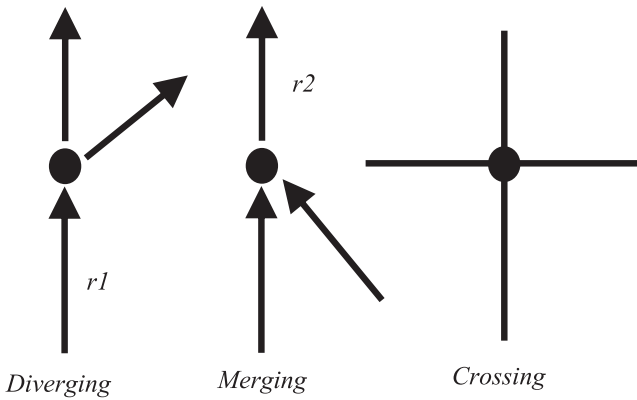


Figure 7. Types of junction.

(Volz, 2006; Mustière and Devogele, 2008; Abdolmajidi et al., 2015; Zhang et al., 2016). In this paper, topology reconstruction is performed initially to eliminate the topological inconsistencies between CNM and LLM.

Physically, a junction connects at least three road segments. In this paper, junctions are classified into three broad families: diverging, merging and crossing. As shown in Figure 7, the diverging junction connects one incoming road segment and two outgoing road segments; the merging junction connects two incoming road segments and one outgoing road segment; and the crossing junction connects more than three road segments.

For both  $TN_{LLM}$  and  $TN_{CNM}$ , road and junction are regarded as atomic construction elements to construct new networks. In  $TN_{LLM} = (LN, LCL)$ , a junction is defined as a set of lane nodes. As defined above, the diverging/merging junction owns the sole incoming/outgoing road (the crossing junction is regarded as several independent junctions; each junction uses an inside virtual road to connect the incoming and outgoing road).



Therefore, Equation (1) is used to identify all the lane nodes of the same junction. Diverging, merging and crossing junctions are denoted by  $jun_{div}$ ,  $jun_{mer}$  and  $jun_{cro}$ , respectively. Lane node (denoted as  $ln$ ) has a junction type attribute that is returned by function type ( $ln$ ). For example, in Figure 2, based on our specification, type ( $ln_2$ ) = *Diverging*, yet type ( $ln_1$ ) = *Null*. For lane centreline set  $setLCL$ , the road set it belongs to is denoted as  $setRoad$ , namely  $setRoad = \text{Belongto}(setLCL)$ . Function  $\text{InLCLs}(ln)/\text{OutLCLs}(ln)$  returns the lane centreline set incoming to/outgoing from  $ln$ ; and  $\emptyset$  represents the empty set. For example, in Figure 2, based on Equation (1), the lane nodes,  $ln_2$  and  $ln_3$ , of *Junction1* can be identified and collected from  $LN$  because  $\text{type}(ln_2) = \text{type}(ln_3) = \textit{Diverging}$  and  $\text{Belongto}(\text{InLCLs}(ln_2)) \cap \text{Belongto}(\text{InLCLs}(ln_3)) = \textit{Road1} \neq \emptyset$ .

$$\begin{aligned}
 jun_{div} &= \{ln | \text{type}(ln) = \textit{Diverging}, \cap \text{Belongto}(\text{InLCLs}(ln)) \neq \emptyset\} \\
 jun_{mer} &= \{ln | \text{type}(ln) = \textit{Merging}, \cap \text{Belongto}(\text{OutLCLs}(ln)) \neq \emptyset\} \\
 jun_{cro} &= \left\{ ln | \text{type}(ln) = \textit{Crossing}, \bigcap_{i \neq j} (\text{Belongto}(\text{InLCLs}(ln_i))) \right. \\
 &\quad \left. \cap \text{Belongto}(\text{OutLCLs}(ln_j)) \neq \emptyset \right\}
 \end{aligned} \tag{1}$$

The topological connection of  $ln$  is the set of the incoming and outgoing lane centrelines, as expressed in Equation (2).

$$\text{TopoLN}(ln) = \text{InLCLs}(ln) \cup \text{OutLCLs}(ln) \tag{2}$$

Similarly, the topological connection of a junction is the set of the incoming and outgoing roads. For the lane centreline set  $\text{TopoLN}(ln)$ , the road set it corresponds to can be obtained by  $\text{Belongto}(\text{TopoLN}(ln))$ . For a junction  $jun$ , all the connecting roads can be extracted by Equation (3), where  $x$  denotes the number of lane nodes of  $jun$ .

$$\text{TopoJun}(jun) = \bigcup_i^x \text{Belongto}(\text{TopoLN}(ln_i)), ln_i \in jun \tag{3}$$

Through Equations (1–3), the connection of the junction can be extracted; for instance, in Figure 2, it can be known that *Junction1* connects *Road1*, *Road2* and *Road3*, because  $\{Road1, Road2, Road3\} = \bigcup_{i=1}^2 \text{Belongto}(\text{TopoLN}(ln_i)), ln_i \in \{ln_2, ln_3\}$ . Suppose  $JUN_{DIV}$ ,  $JUN_{MER}$  and  $JUN_{CRO}$  are the set of diverging, merging and crossing junctions respectively, namely  $jun_{div} \in JUN_{DIV}, jun_{mer} \in JUN_{MER}, jun_{cro} \in JUN_{CRO}$ , then  $JUN = JUN_{DIV} \cup JUN_{MER} \cup JUN_{CRO}$ , where  $JUN$  is the vertex set,  $jun \in JUN$ . Taking  $r$  as the edge (road reference line represents road, geometrically),  $R$  denotes the edge set. Consequently, the reconstructed network of  $TN_{LLM}$  is  $TN_{LLM\_rec} = (JUN, R)$ .



For the CNM network  $TN_{CNM} = (NODE, LINK)$ , the nodes are directly marked as different junctions according to Equation (4).

$$\begin{aligned} cnmJUN_{DIV} &= \{node | \text{Indegree}(node) = 1, \text{Outdegree}(node) = 2\}; \\ cnmJUN_{MER} &= \{node | \text{Indegree}(node) = 2, \text{Outdegree}(node) = 1\}; \\ cnmJUN_{CRO} &= \{node | \text{Indegree}(node) + \text{Outdegree}(node) > 3\}; \\ cnmNODE &= \{node | node \notin cnmJUN_{DIV}, node \notin cnmJUN_{MER}, node \notin cnmJUN_{CRO}\}. \end{aligned} \quad (4)$$

Function  $\text{Indegree}(node)$  and  $\text{Outdegree}(node)$  return the in-degree and out-degree of  $node$ ;  $cnmJUN_{DIV}$ ,  $cnmJUN_{MER}$ ,  $cnmJUN_{CRO}$  and  $cnmNODE$  denote the set of diverging, merging, crossing and general nodes, respectively. Through Equation (4),  $TN_{CNM}$  is reconstructed into  $TN_{CNM\_rec} = (cnmJUN, LINK)$ , where  $cnmJUN = cnmJUN_{DIV} \cup cnmJUN_{MER} \cup cnmJUN_{CRO} \cup cnmNODE$ .

4.2. *Detecting the corresponding junctions in  $TN_{LLM\_rec}$  and  $TN_{CNM\_rec}$ .* Most of the topological difference has been eliminated through topology reconstruction. Thus, the design of the corresponding junction detection algorithm should consider the geometric inconsistencies between LLM and CNM. There are two ways to make the algorithm stronger: (1) not using sensitive thresholds and (2) avoiding using weightings based on limited experience. Taking into account the principles, the so-called multi-filter algorithm is designed for homologous junction detection.

4.2.1. *The filters.* As Tobler's first law indicates: near things are more related than distant things. In reality, spatially close objects usually have similar spatial features. It is hard to distinguish the junctions if only spatial characteristics are considered because some spatial metrics are usually correlated, as two nearby road segments generally have a similar direction due to the limitation of terrain. Due to the geometric inconsistencies between LLM and CNM and the spatial features correlation between near objects, heterogeneous metrics selected from different aspects, like geometry, topology and semantics, should be utilised to distinguish the junctions. In this paper, the following five filters are used.

- (1) *Euclidean Distance (ED)*: *ED* is a geometric metric. For each  $jun \in JUN$ , the nearest 10 candidate junctions (denoted as  $cnmjunc_{cd}$ ) in  $TN_{CNM\_rec}$  are selected to initialise the candidate set, *Setcandidates*, where the corresponding junction of  $jun$  is anticipated to be picked out.
- (2) *Topological Structure (TS)*: *TS* is a topological metric and characterised as diverging, merging and crossing as shown in Figure 7. The  $cnmjunc_{cd}$  with different *TS* against  $jun$  will be rejected from the candidate set.
- (3) *Type of Main Road (TMR)*: The road incoming to a diverging junction ( $r1$  in Figure 7) or outgoing from a merging junction ( $r2$  in Figure 7) is defined as the main road. Because the junction has a sole main road, the characteristics of the main road can be used to distinguish the junction efficiently. *TMR* is a semantic metric. In this paper, road is roughly classified as: highway and non-highway. The candidate with different *TMR* against  $jun$  will be kicked out. *TMR* is just applied to the junction whose degree is 3.
- (4) *Direction of Main Road (DMR)*: *DMR* is a geometric metric related to the main road. For the road incoming to a junction, its direction vector consists of the point at a

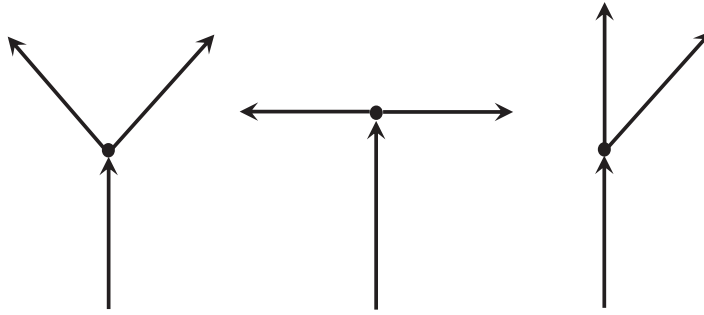


Figure 8. Different shapes of junctions.

distance of 10 m behind its end point (as initial point) and the end point (as terminal point). For the road outgoing from a junction, its direction vector consists of the start point (as initial point) and the point at a distance of 10 m front of its start point (as terminal point). The direction vectors of the main roads of *jun* and *cnmjunc<sub>cd</sub>* are compared; the candidate that induces an angle greater than 30 degrees is rejected. *DMR* is also just applied to the 3-degree junction.

- (5) *Geometrical Shape (GS)*: Some inappropriate candidates can be kicked out via the above filters, however, it is hard to distinguish the junctions with the same *TS*, *TMR* and *DMR*, as shown in Figure 8. To cope with this situation, a junction shape similarity measurement function is proposed. *GS* is the third geometric metric. The junction's shape is formed by a set of vectors and the angles between vectors reflect its shape, as shown in Figure 8. Taking the direction vectors of the roads connecting to *jun*, the normalised direction vector set is  $\mathbf{V}_{\text{jun}} = \{\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_n\}$ ; and the normalised direction vector set of *cnmjunc<sub>cd</sub>* is  $\mathbf{V}_{\text{cnmjunc}_{cd}} = \{\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_m\}$ . Then  $\mathbf{simM} = \mathbf{V}_{\text{jun}}^T \times \mathbf{V}_{\text{cnmjunc}_{cd}}$ , and the similarity measure function is:

$$\text{Sim}(jun_k, cnmjunc_{cd}) = \frac{\sum_{j=1}^n \max(\text{simM}(j))}{\max(m, n) \cdot (d/a)} \quad (5)$$

In Equation (5),  $\text{simM}(j)$  returns  $j$ -th row of  $\mathbf{simM}$ ;  $d$  is the distance between *jun* and *cnmjunc<sub>cd</sub>*, and  $a$  is the scaling parameter. In this paper,  $a = 50$  m, the similarity of junctions within 50 m will be enhanced; otherwise, diminished.

4.2.2. *Adaptive multi-filter algorithm for corresponding junction detection.* In this algorithm, the five filters are employed to eliminate the inappropriate candidates, gradually.

The entire process of multi-filter algorithm is explained by Figure 9. First, the candidate set, *Setcandidates*, is initialised; then the conflict candidates are rejected by *TS*, *TMR* and *DMP* filters. The three filters are of distinguishable and error-free characteristics; for example, it is very easy to recognise two junctions of different *TS*; and whether LLM or CNM, it is almost impossible to assign wrong *TS* and *TMR* related attributes to the released maps. The *DMR* of 30 degrees threshold is relaxed enough to avoid misjudgement. After the screening process, if there are more than one candidates, the best matching will be directly determined by *GS* filter. In our experiments, most of the corresponding junctions were found by *ED*, *TS*, *TMR* and *DMR* filters and only four junctions were determined by

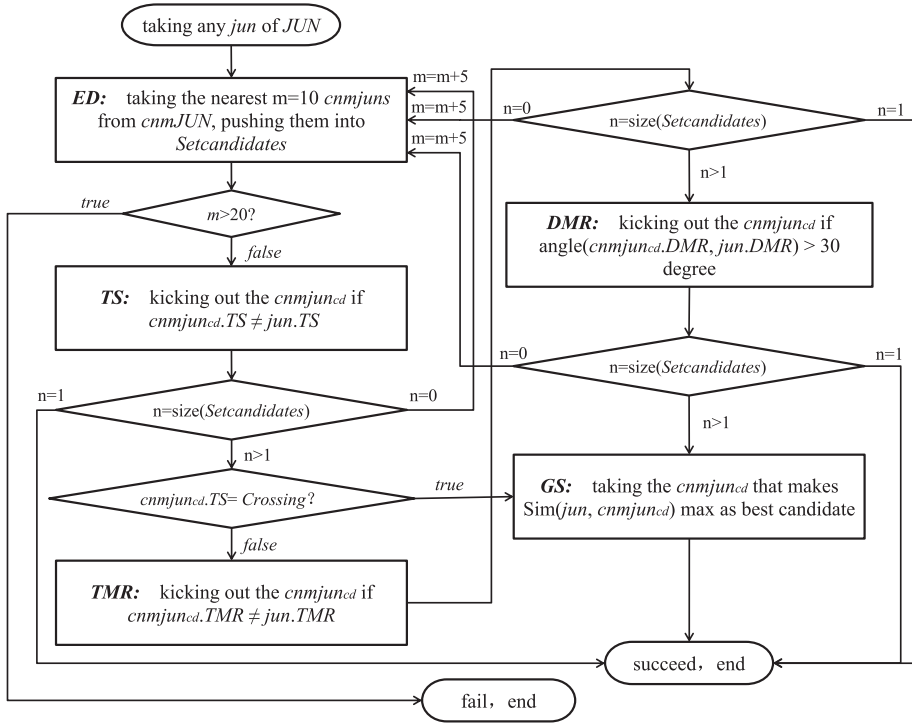


Figure 9. Flowchart of multi-filter algorithm.

GS. The final survival is considered as the corresponding junction of *jun*. If no one survives, new candidates will be pushed into *Setcandidates*. The pushing is not more than two times, otherwise the algorithm fails.

4.3. Finding the corresponding paths in  $TN_{LLM\_rec}$  and  $TN_{CNM\_rec}$ . The strategy Figure 10 illustrates is used to find the corresponding paths between two adjacent corresponding junction pairs in  $TN_{LLM\_rec}$  and  $TN_{CNM\_rec}$ .

- (1) First, starting from any junction *junS* of  $TN_{LLM\_rec}$ , *Pexplorer* (forward path exploring algorithm, see Figure 11) walks along the road until reaching the matched adjacent junction *junA*. The road passed through is marked as an element of the path, e.g.  $path_{LLM} = \{r1, r3\}$ .
- (2) Based on the junction correspondence built in Section 4.2, it is easy to find the corresponding junction of *junS* in  $TN_{CNM\_rec}$ , namely *cnmjuns*.
- (3) Similarly, in  $TN_{CNM\_rec}$ , starting from *cnmjuns*, *Pexplorer* walks to the matched adjacent junction *cnmjuna*, and the passed path is  $path_{CNM} = \{link1, link3, link4\}$ .
- (4) Finally, the correspondence of *junA* and *cnmjuna* will be validated. If they are corresponding,  $path_{LLM}$  and  $path_{CNM}$  are regarded as corresponding; otherwise, the algorithm fails.

*Pexplorer* is explained through the pseudocode in Figure 11. Function  $OutR(jun)/InR(jun)$  is used to obtain the roads outgoing from/incoming to junction *jun*;  $JunRin(r)$  is used to find the junction that road *r* is incoming to. If *jun* is matched,  $isMatched(jun)$  returns true, else

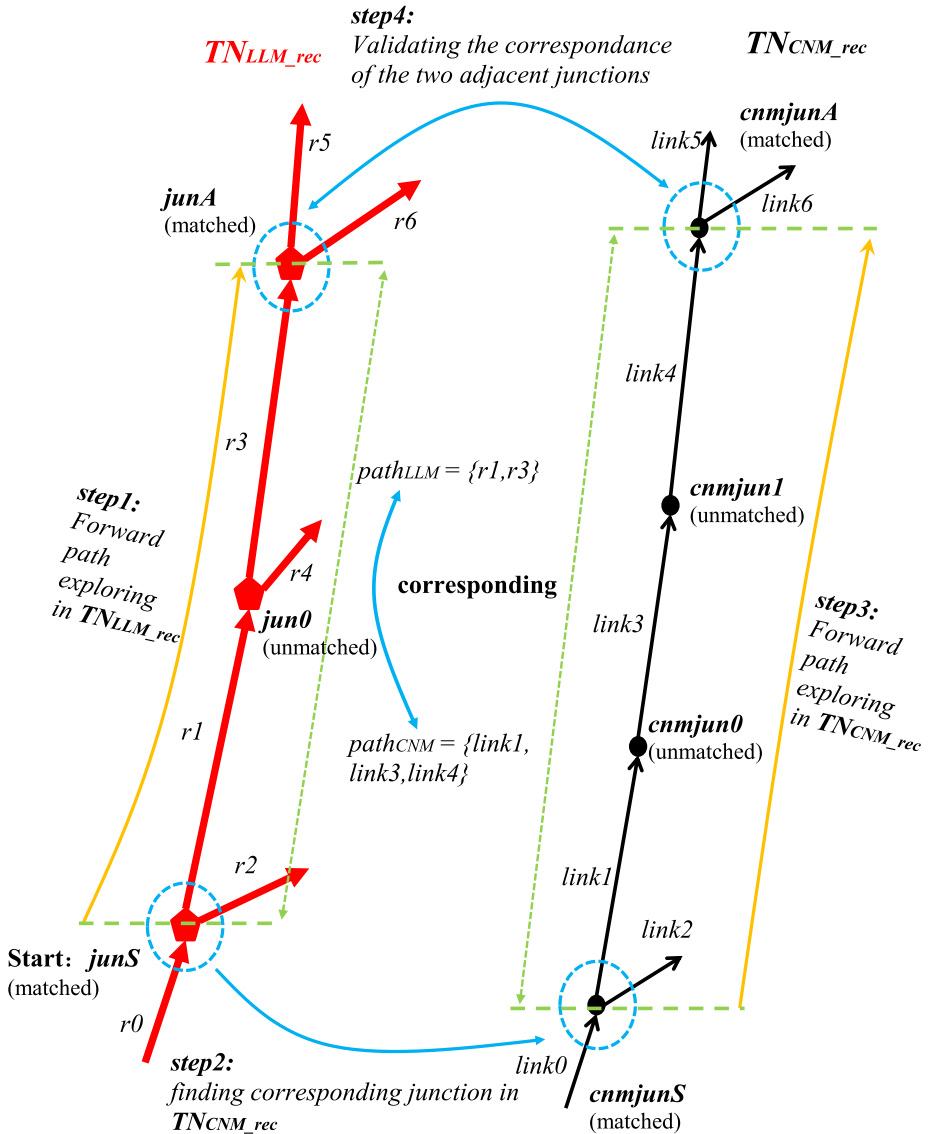


Figure 10. Finding the corresponding paths in  $TN_{LLM\_rec}$  and  $TN_{CNM\_rec}$ .

false. In  $TN_{LLM\_rec}$ , *Explorer* takes  $junS$  as input, and outputs the adjacent junction  $junA$  and the path between  $junS$  and  $junA$ ,  $path_{LLM}$ . And so it does for  $TN_{CNM\_rec}$ .

5. BUILDING THE LINKING RELATIONS. The relationship of road correspondence is related to the element numbers of  $path_{LLM}$  and  $path_{CNM}$ . If both  $path_{LLM}$  and  $path_{CNM}$  contain only one element, it is 1:1 (1 to 1) relationship; if one or both of them contain more than one element, then they are 1:m (1 to m) or m:n (m to n) relationships. The 1:1 relationship is explicit; however, relationships of 1:m and m:n provoke ambiguities.

<b>Forward Path Exploring Algorithm (<i>Pexplorer</i>)</b>	
<b>Input:</b>	$junS$ ( $cnmjunS$ )
<b>Output:</b>	$junA$ ( $cnmjunA$ ), $path_{LLM}$ ( $path_{CNM}$ )
<b>Define:</b>	
	$OutR(jun) = \text{Belongto}(\text{OutLCL}(ln))$ , where $ln \in jun$ .
	$InR(jun) = \text{Belongto}(\text{InLCL}(ln))$ , where $ln \in jun$ .
	$JunRIn(r) = jun$ , where $r \in InR(jun)$ ; if no $jun$ found return $NULL$ .
	$isMatched(jun) = true$ if $jun$ is matched with some $cnmjun$ ; else $false$ .
1	$path_{LLM} = NULL$
2	<b>while</b> (1) <b>do</b>
3	$outRs = OutR(junS)$
4	$jun = NULL$
5	<b>for</b> $i = 0$ To $size(outRs)$ <b>do</b>
6	$jun = JunRIn(outRs(i))$
7	<b>if</b> $jun \neq NULL$ <b>Then</b>
8	$path_{LLM} = path_{LLM} \cup outRs(i)$
9	<b>else</b>
10	$i++$
11	<b>end if</b>
12	<b>end</b>
13	<b>if</b> $!isMatched(jun)$ <b>Then</b>
14	$junS = jun$
15	<b>continue</b>
16	<b>else</b>
17	<b>break</b>
18	<b>end if</b>
19	<b>end</b>

Figure 11. Pseudocode of forward path exploring algorithm.

For instance, for  $path_{LLM} = \{r1, r3\}$  and  $path_{CNM} = \{link1, link3, link4\}$ , it is hard to know which road section in LLM  $link3$  corresponds to. In this section, a uniform and explicit method is proposed to describe the road correspondence.

5.1. *Length proportion based projection.* It is easy to know the corresponding road section if there is a locational referencing scheme, and one knows where the section starts and ends. The locational reference scheme in this paper is similar to the link and node locational reference proposed by Nyerges (1990).

Start and end positions are used to project each link of  $path_{CNM}$  to  $path_{LLM}$ . For the corresponding paths:  $path_{LLM} = \{r_1, r_2, \dots, r_m\}$ ,  $m \geq 1$  and  $path_{CNM} = \{link_1, link_2, \dots, link_n\}$ ,  $n \geq 1$ , each  $link_i \in path_{CNM}$  can derive the locational referencing positions with

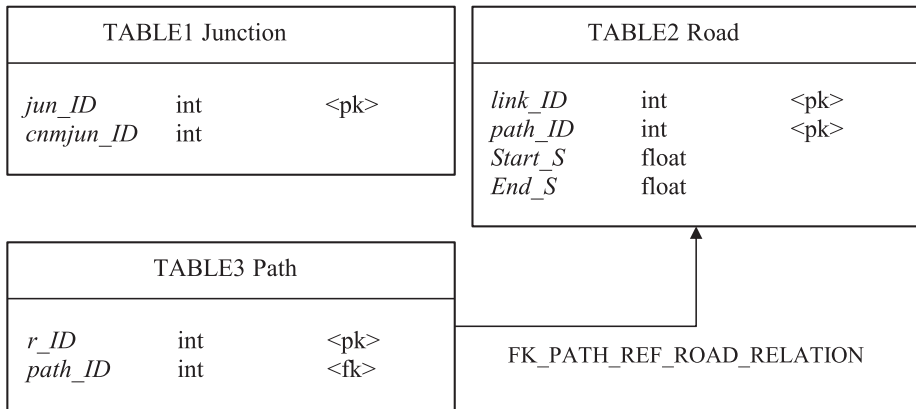


Figure 12. Physical data model for the linking relations.

respect to  $path_{LLM}$  through Equation (6).

$$\begin{aligned}
 Start\_S &= \frac{\sum_{j=1}^{i-1} \text{Length}(link_j)}{\sum_{j=1}^n \text{Length}(link_j)} \sum_{j=1}^m \text{Length}(r_j), \\
 End\_S &= \frac{\sum_{j=1}^i \text{Length}(link_j)}{\sum_{j=1}^n \text{Length}(link_j)} \sum_{j=1}^m \text{Length}(r_j)
 \end{aligned} \tag{6}$$

5.2. *Physical data model for the linking relations.* The linking relations’ physical data model is given in Figure 12, where the junction relations table and road relations table store the junction and road correspondence, and the path table records the roads of  $path_{LLM}$ .

## 6. EXPERIMENTS AND RESULTS.

6.1. *Study areas and data.* Two data sets were used to validate the approach. Test area A was a section of highway (mainly) in Beijing, northern China, with a total length of about 120 km. Test area B was a section of beltway (mainly) in Changsha, Hunan Province, southern China, with a total length of about 30 km. The LLM data of areas A and B were captured in 2015 and 2016. The CNM data of areas A and B were released in 2013 and 2016. It should be pointed out that the CNM and the LLM were made by two different map suppliers.

The LLM contained data of captured roads and the vicinities, whereas the CNM contained the whole city’s data. For reducing unnecessary calculation, a 500 m buffer around the captured roads was created, and only the CNM data within the buffer was used. The experimental data sets are shown in Figure 13, where the red is  $TN_{LLM}$  and the black is  $TN_{CNM}$ . The microscopic views of where the blue arrows point are on the lower right. The statistics of the used  $TN_{LLM}$  and  $TN_{CNM}$  of each data set are summarised in Table 2. Quantitative differences of data sets A and B were caused by: (1) the lengths of the captured roads, (2) map specification change and (3) different cities.

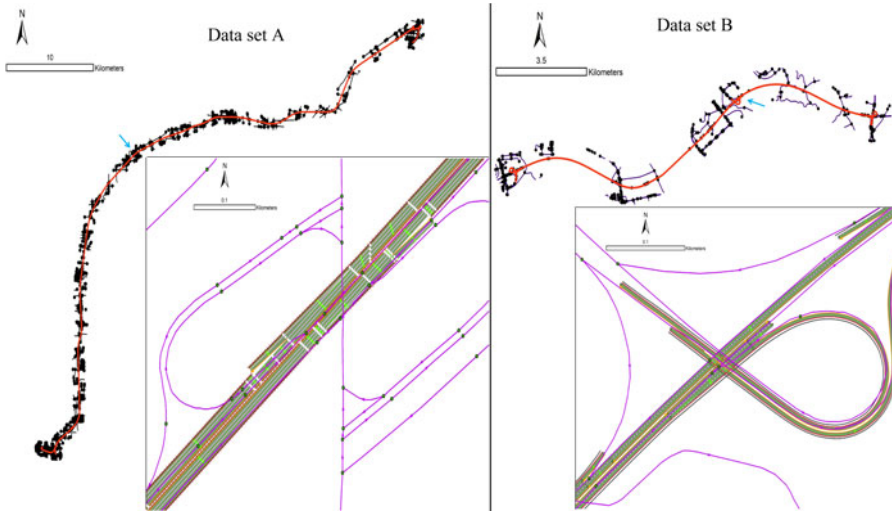


Figure 13. Test data sets.

Table 2. Statistics of data sets A and B.

Networks	Data set A		Date set B	
	$TN_{LLM}$ (2015)	$TN_{CNM}$ (2013)	$TN_{LLM}$ (2016)	$TN_{CNM}$ (2016)
Node number	–	9677	–	2542
Link number	–	36,387	–	7737
Lane node number	5664	–	131	–
Lane centreline number	5593	–	125	–

6.2. *Evaluation.* The classical indicators of precision and recall defined in Equation (7) were used to assess the quality of the experimental results.

$$\begin{aligned}
 recall &= \frac{|SC|}{|SA|}, & precision &= \frac{|SC|}{|SR|}, \\
 recall_{length} &= \frac{\sum_{r \in SC} length(r)}{\sum_{r \in SA} length(r)}, & precision_{length} &= \frac{\sum_{r \in SC} length(r)}{\sum_{r \in SR} length(r)} \quad (7)
 \end{aligned}$$

where  $SC$  denotes the set of correct pairs appearing in the results;  $SA$  denotes the set of all correct pairs among the whole data set; and  $SR$  is the set of results. Generally, the longer road is of greater importance than the shorter. Therefore, length-based recall and precision were employed as another two indicators.

6.3. *Results.* The approach in this study was coded in C++ and run on a general personal computer. It was completed in an acceptable time ( $A = 114$  s,  $B = 11$  s). The results were checked manually and are summarised in Table 3.

In data set A, the reconstructed network  $TN_{LLM\_rec}$  consisted of 76 junctions and 172 roads, where 68 junctions and 69 roads were correctly matched. The length of correctly matched roads was nearly 113 km. In data set B, 11 junctions and 17 roads were correctly matched among the total 26 junctions and 40 roads. The total length of correctly matched



Table 3. Statistics of results.

	Test data	Correctly matched	Unmatched	Mismatched	Total
Data set A	<i>jun</i> number of $TN_{LLM\_rec}$	68	8	0	76
	<i>r</i> number of $TN_{LLM\_rec}$	69	103	0	172
	<i>r</i> length of $TN_{LLM\_rec}$ (m)	112,950.88	7659.05	0	120,609.93
Data set B	<i>jun</i> number of $TN_{LLM\_rec}$	11	15	0	26
	<i>r</i> number of $TN_{LLM\_rec}$	17	23	0	40
	<i>r</i> length of $TN_{LLM\_rec}$ (m)	28,953.78	1851.42	0	30,805.20

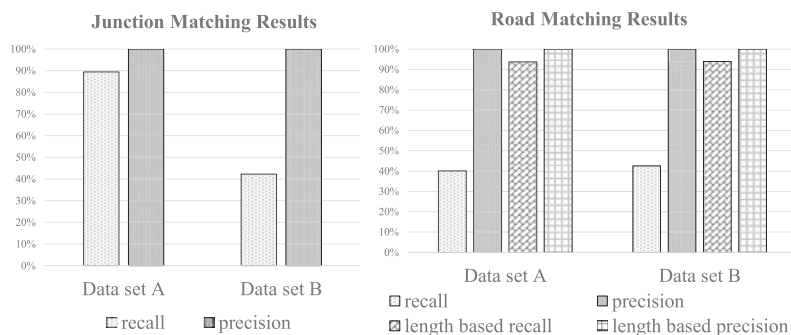


Figure 14. Recall and precision of the results.

roads was nearly 29 km. All the unmatched junctions were confirmed by the operator. In data set A, the eight unmatched junctions of LLM were built after 2013, thus they were not present in the CNM. They could not and should not be matched. Compared with the approach proposed by Mustière and Devogele (2008), our approach did not fail in this situation; the homologous parts of the roads had been matched. In data set B, where the LLM's specification had changed slightly against that of data set A, 15 pseudo junctions near the actual junctions were constructed to make  $TN_{LLM\_rec}$  connective. Matching pseudo junctions was also not necessary.

Most of the roads connecting with the unmatched junctions were matched successfully because *Pexplorer* could pass through the roads and mark them as elements of the path, which explains the high length-based recall undergoing low recall of junction matching in data set B, see Figure 14. In Table 3, the average lengths of unmatched roads of data set A and B are 74 m and 80 m, respectively; however, the average lengths of matched roads are 1,637 m and 1,703 m. This indicates most of the important roads were matched. It was found that most of the unmatched roads were located at the highway interchanges, where very few of them were captured because of the limited range of on-board sensors. These roads were represented as dead ends in  $TN_{LLM}$ .

As Figure 14 shows, regarding road matching, although the recall is below 50%, the length-based recall is nearly 94%, with precision of 100%. That means most of the roads in LLM have been matched correctly.

Compared with the methods that built linking relations semi-automatically (Volz and Walter, 2004), in this paper, the linking relations were built automatically. Tables 4–6 give a short extract (incomplete) of the linking relations between LLM and CNM for

Table 4. Examples of junction relations.

<i>jun_ID</i>	<i>cnmjun_ID</i>
0	221
1	158
2	157
3	156
4	130
5	131
7	132
8	136
12	84

Table 5. Examples of paths.

<i>r_ID</i>	<i>path_ID</i>
1010003	0
1010005	1
2010016	2
3010003	3
9010003	3
10010005	3
10010012	3
18010202	4
18010204	5

Table 6. Examples of road relations.

<i>link_ID</i>	<i>path_ID</i>	<i>Start_S</i>	<i>End_S</i>
339012	0	0	66·25
339009	0	66·25	97·42
318849	1	0	116·58
318855	2	0	610·52
319607	3	0	319·19
337679	3	319·19	1117·16
337704	3	1117·16	1436·34
337707	3	1436·34	1755·53
430523	3	1755·53	3989·84

data set A. Compared with the approaches that can only deal with 1:1 correspondence (Devoegele, 2002) or 1:1 and 1:2 correspondence (Volz, 2006), our approach is able to build the relationships of 1:1 (e.g.,  $\{r\_ID:101005\}:\{link\_ID:318849\}$ ), 1:m (e.g.,  $\{r\_ID:101003\}:\{link\_ID:339012,339009\}$ ), and m:n (e.g.,  $\{r\_ID:301003,901003,\dots\}:\{link\_ID:319607,337679,\dots\}$ ). With Tables 5 and 6, each link (e.g., *link\_ID*: 339012) of CNM is linked to the corresponding road section of LLM (the road section from 0 m to 66·25 m of *r\_ID*:101003). Conversely, with the length of road (e.g., *r\_ID*:301003, length 500 m), it is easy to derive the corresponding link section in CNM (whole of *link\_ID*:319607 and section of *link\_ID*:337679 from 0 m to 180·81 m).

7. CONCLUSION. In this paper, an approach has been proposed to build linking relations automatically between LLM and CNM. In order to facilitate network matching, similarly-structured networks are built for LLM and CNM, initially. The adaptive multi-filter algorithm and forward path exploring algorithm are proposed to detect corresponding junctions and roads, respectively. The algorithms are able to match the greatly-differing networks of LLM and CNM and achieve high recall and precision. This approach can also build 1:1, 1:m and m:n linking relationships between LLM and CNM. With the linking relations, for the link in CNM it is easy to find the corresponding road section in LLM, and vice versa. The linking relations built in this paper can facilitate autonomous driving systems, especially the hierarchical planning system, and pave the way for interoperation and data sharing between LLM and CNM.

There are still some issues to be improved. As most map suppliers mainly focus on highway LLMs currently, this paper aims to build linking relations automatically for LLMs and CNMs of highway, primarily; in future, the approach should be enhanced to address urban road networks. The topology reconstruction targets eliminating topological inconsistencies to help with network matching; however, it may fail in topologically complex areas (like the tollgate in Figure 4(b)), complex intersections (like roundabouts), and areas with incomplete/wrong connections. The topology reconstruction logic needs to be enhanced and the non-topological matching methods, like geometric and semantic methods, should be explored. Finally, the spatial features were treated in a two-dimensional plane in this paper; in future, the altitude of the object should also be considered.

#### ACKNOWLEDGEMENTS

This work was supported by the National Key Research and Development Program: New Energy Vehicles Special Project ‘Research on Key Technologies and Legally Social Issues for Autonomous Driving Electric Vehicle Applications’ (2018YFB0105202).

#### REFERENCES

- Abdolmajidi, E., Mansourian, A., Will, J. and Harrie, L. (2015). Matching authority and VGI road networks using an extended node-based matching algorithm. *Geo-Spatial Information Science*, 18, 65–80. <https://doi.org/10.1080/10095020.2015.1071065>.
- Alonso, J., Milanés, V., Onieva, E., Pérez, J., González, C. and de Pedro, T. (2011). Cartography for cooperative manoeuvres with autonomous land vehicles. *The Journal of Navigation*, 64, 141–155. <https://doi.org/10.1017/S0373463310000275>.
- Bender, P., Ziegler, J. and Stiller, C. (2014). Lanelets: Efficient Map Representation for Autonomous Driving. In: *Intelligent Vehicles Symposium Proceedings, 2014 IEEE*. IEEE, pp. 420–425.
- DARPA Urban Challenge. (2007). Route network definition file (RNDF) and mission data file (MDF) formats. *Tech. Rep., Defense Advanced Research Projects Agency*.
- Devogele, T. (2002). A new merging process for data integration based on the discrete Fréchet distance. In: *Advances in Spatial Data Handling*. Springer, pp. 167–181.
- Doytsher, Y., Filin, S. and Ezra, E. (2001). Transformation of datasets in a linear-based map conflation framework. *Surveying and Land Information Systems*, 61, 159–169.
- Dupuis, M. (2010). Opendrive format specification. VIRES Simulationstechnologie GmbH.
- Fan, H., Zhu, F., Liu, C., Zhang, L., Zhuang, L., Li, D., Zhu, W., Hu, J., Li, H. and Kong, Q. (2018). Baidu Apollo EM Motion Planner. ArXiv Prepr. ArXiv180708048.
- Goodchild, M. F. (2000). GIS and transportation: status and challenges. *GeoInformatica*, 4, 127–139. <http://dx.doi.org/10.1023/A:1009867905167>.
- Guizzo, E. (2011). How Google’s self-driving car works. *IEEE Spectrum Online*, 18, 1132–1141.

- Kato, S., Takeuchi, E., Ishiguro, Y., Ninomiya, Y., Takeda, K. and Hamada, T. (2015). An open approach to autonomous vehicles. *IEEE Micro*, 35, 60–68. <https://doi.org/10.1109/MM.2015.133>.
- Koukoletsos, T., Haklay, M. and Ellul, C. (2012). Assessing data completeness of VGI through an automated matching procedure for linear data. *Transactions in GIS*, 16, 477–498. <https://doi.org/10.1111/j.1467-9671.2012.01304.x>.
- Leonard, J., How, J., Teller, S., Berger, M., Campbell, S., Fiore, G., Fletcher, L., Frazzoli, E., Huang, A., Karaman, S., Koch, O., Kuwata, Y., Moore, D., Olson, E., Peters, S., Teo, J., Truax, R., Walter, M., Barrett, D., Epstein, A., Maheloni, K., Moyer, K., Jones, T., Buckley, R., Antone, M., Galejs, R., Krishnamurthy, S. and Williams, J. (2008). A perception-driven autonomous urban vehicle. *Journal of Field Robotics*, 25, 727–774. <https://doi.org/10.1002/rob.20262>.
- Liu, L., Wu, T., Fang, Y., Hu, T. and Song, J. (2015). A Smart Map Representation for Autonomous Vehicle Navigation. In: *Fuzzy Systems and Knowledge Discovery (FSKD), 2015 12th International Conference On*. IEEE, pp. 2308–2313.
- Mustière, S. and Devogele, T. (2008). Matching networks with different levels of detail. *GeoInformatica*, 12, 435–453. <https://doi.org/10.1007/s10707-007-0040-1>.
- Naranjo, J. E., Jiménez, F., Aparicio, F. and Zato, J. (2009). GPS and inertial systems for high precision positioning on motorways. *The Journal of Navigation*, 62, 351. <https://doi.org/10.1017/S0373463308005249>.
- Nedevschi, S., Popescu, V., Danescu, R., Marita, T. and Oniga, F. (2013). Accurate ego-vehicle global localization at intersections through alignment of visual data with digital map. *IEEE Transactions on Intelligent Transportation Systems*, 14, 673–687. <https://doi.org/10.1109/TITS.2012.2228191>.
- Nyerges, T. L. (1990). Locational referencing and highway segmentation in a geographic information system. *ITE Journal*, 60, 27–31.
- Pendleton, S., Andersen, H., Du, X., Shen, X., Meghiani, M., Eng, Y., Rus, D. and Ang, M. (2017). Perception, planning, control, and coordination for autonomous vehicles. *Machines*, 5, 6. <https://doi.org/10.3390/machines5010006>.
- Saalfeld, A. (1988). Conflation automated map compilation. *International Journal of Geographical Information Systems*, 2, 217–228. <https://doi.org/10.1080/02693798808927897>.
- Safra, E., Kanza, Y., Sagiv, Y. and Doytsher, Y. (2006). Efficient Integration of Road Maps, In: *Proceedings of the 14th Annual ACM International Symposium on Advances in Geographic Information Systems*. ACM, pp. 59–66.
- Tao, Z., Bonnfait, P., Fremont, V. and Ibanez-Guzman, J. (2013). Mapping and Localization Using GPS, Lane Markings and Proprioceptive Sensors, In: *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference On*. IEEE, pp. 406–412.
- Uitermark, H. T. and Cadastre, D. (1996). The Integration of Geographic Databases: Realising Geodata Interoperability Through the Hypermap Metaphor and a Mediator Architecture, In: *Proceedings of the Second Joint European Conference & Exhibition on Geographical Information (Vol. 1): From Research to Application through Cooperation*. IOS Press, pp. 92–95.
- Volz, S. (2006). An iterative approach for matching multiple representations of street data. *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 36, 101–110.
- Volz, S. and Walter, V. (2004). Linking Different Geospatial Databases by Explicit Relations, In: *Proceedings of the XXth International Society for Photogrammetry and Remote Sensing (ISPRS) Congress, Comm. IV*. pp. 152–157.
- Zhang, M., Shi, W. and Meng, L. (2005). A Generic Matching Algorithm for Line Networks of Different Resolutions. In: *Workshop of ICA Commission on Generalization and Multiple Representation. Computing Faculty of Coruña University-Campus de Elviña, Spain*. Citeseer.
- Zhang, M., Yao, W. and Meng, L. (2016). Automatic and Accurate Conflation of Different Road-Network Vector Data towards Multi-Modal Navigation. *ISPRS International Journal of Geo-Information*, 5, 68. <https://doi.org/10.3390/ijgi5050068>.
- Ziegler, J., Bender, P., Schreiber, M., Lategahn, H., Strauss, T., Stiller, C., Dang, T., Franke, U., Appenrodt, N. and Keller, C. G. (2014). Making Bertha drive—An autonomous journey on a historic route. *IEEE Intelligent Transportation Systems Magazine*, 6, 8–20.

## APPENDIX A. ABBREVIATIONS

The definitions of the abbreviations used in this paper are listed below.

Abbreviation	Meaning	Abbreviation	Meaning
CNM	Commercial navigation map	<i>link</i>	Link of CNM
<i>cnmJUN</i>	Junction set of CNM	LLM	Lane-level map
<i>cnmjun</i>	Junction of CNM	<i>ln</i>	Lane node
<i>cnmjunA</i>	Matched adjacent junction in CNM	<i>LN</i>	The set of lane nodes of LLM
<i>cnmjun<sub>cd</sub></i>	Candidate junction in $TN_{CNM\_rec}$	<i>NODE</i>	The node set of CNM
<i>cnmJUN<sub>CRO</sub></i>	Crossing junction set of CNM	<i>node</i>	Node of CNM
<i>cnmJUN<sub>DIV</sub></i>	Diverging junction set of CNM	<i>path</i>	The path between two adjacent junctions
<i>cnmJUN<sub>MER</sub></i>	Merging junction set of CNM	<i>path<sub>CNM</sub></i>	An explored path in CNM
<i>cnmjunS</i>	Starting junction of <i>Pexplorer</i> in CNM	<i>path<sub>LLM</sub></i>	An explored path in LLM
<i>cnmNODE</i>	General node set of CNM	<i>Pexplorer</i>	Forward path exploring algorithm
<i>DMR</i>	Direction of main road filter	<i>r</i>	Road reference line
<i>ED</i>	Euclidean distance filter	<i>R</i>	Road reference line set of LLM
<i>GS</i>	Geometrical shape filter	<i>SA</i>	All corrected pairs among whole data set
<i>jun</i>	Junction of LLM	<i>SC</i>	The set of correct pairs appearing the matching results
<i>JUN</i>	Junction set of LLM	<i>Setcandidates</i>	Candidate set in multi-filter algorithm
<i>junA</i>	Matched adjacent junction in LLM	<i>setLCL</i>	A certain lane centreline set, a subset of <i>LCL</i>
<i>JUN<sub>CRO</sub></i>	Crossing junction set of LLM	<i>setRoad</i>	A certain road set A subset of <i>R</i>
<i>jun<sub>cro</sub></i>	Crossing junction of LLM	<i>SR</i>	The set of matching results
<i>JUN<sub>DIV</sub></i>	Diverging junction set of LLM	<i>TMR</i>	Type of main road filter
<i>jun<sub>div</sub></i>	Diverging junction of LLM	<i>TN<sub>CNM</sub></i>	Topology network of CNM
<i>JUN<sub>MER</sub></i>	Merging junction set of LLM	<i>TN<sub>CNM\_rec</sub></i>	Reconstructed $TN_{CNM}$
<i>jun<sub>mer</sub></i>	Merging junction of LLM	<i>TN<sub>LLM</sub></i>	Topology network of LLM
<i>junS</i>	Starting junction of <i>Pexplorer</i> in LLM	<i>TN<sub>LLM\_rec</sub></i>	Reconstructed $TN_{LLM}$
<i>LCL</i>	The set of Lane centreline of LLM	<i>TS</i>	Topological structure filter
<i>LINK</i>	Link set of CNM		