Research
Intelligent Transportation—Article

A Flexible Multi-Layer Map Model Designed for Lane-Level Route Planning in Autonomous Vehicles

Kun Jiang, Diange Yang *, Chaoran Liu, Tao Zhang, Zhongyang Xiao

State Key Laboratory of Automotive Safety and Energy, Department of Automotive Engineering & Collaborative Innovation Center of Intelligent New Energy Vehicles, Tsinghua University, Beijing 100084, China

A B S T R A C T

An increasing number of drivers are relying on digital map navigation systems in vehicles or mobile phones to select optimal driving routes in order to save time and improve safety. In the near future, digital map navigation systems are expected to play more important roles in transportation systems. In order to extend current navigation systems to more applications, two fundamental problems must be resolved: the lane-level map model and lane-level route planning. This study proposes solutions to both problems. The current limitation of the lane-level map model is not its accuracy but its flexibility; this study proposes a novel seven-layer map structure, called as Tsinghua map model, which is able to support autonomous driving in a flexible and efficient way. For lane-level route planning, we propose a hierarchical route-searching algorithm to accelerate the planning process, even in the presence of complicated lane networks. In addition, we model the travel costs allocated for lane-level road networks by analyzing vehicle maneuvers in traversing lanes, changing lanes, and turning at intersections. Tests were performed on both a grid network and a real lane-level road network to demonstrate the validity and efficiency of the proposed algorithm.

1. Introduction

A digital map navigation system assists drivers or intelligent vehicles to select the optimal route, given an origin and a destination [1,2]. The first application of a vehicle navigation system, in a BMW passenger car, can be traced back to 1994. Since then, the benefits in travel efficiency provided by navigation systems have been widely accepted. At present, most vehicle navigation systems are based on road-level maps with limited and low-precision information [3]. In recent research, the development of advanced driver-assistance systems and autonomous driving technology requires increasing assistance from digital maps. In the literature, many intelligent driving functions have been developed based on the use of digital maps.

These map-supported intelligent driving functions can be further categorized into two groups as follows:

Road-level map-supported functions. These functions are usually designed to assist human drivers, and the required accuracy is usually approximately 10 m. For example, the electronic horizon program uses road slope angles to adjust the driving strategy to save energy [4]. The curve speed warning system in Refs. [4,5] is another example of a map-enabled active safety system. The common point of these systems is that the required information is stored in a road-level map, which ignores the lane details.

Lane-level map-supported functions. These functions can acquire lane-level details of the environment from digital maps in order to enhance the vehicle's intelligence. When a map reaches the lane level, the navigation system can offer further driving assistance with greater accuracy. A representative example of lane-level map-supported driving system was displayed in the Defense Advanced Research Projects Agency (DARPA) 2007 Urban Challenge, in Bohren et al.'s work, a road network definition file was created to describe lane-level driving conditions [6].

This study aims to develop a lane-level map-supported route-planning system, which is designed to support autonomous

* Corresponding author.
E-mail address: ydg@tsinghua.edu.cn (D. Yang).

driving. The target uses of conventional navigation systems are human drivers, who are responsible for choosing the trajectory in real time. However, when the target user is an autonomous vehicle, it is essential for the navigation system to provide more detailed guidance to help the vehicle accomplish its driving tasks. As illustrated in Fig. 1, road-level navigation can only provide assistance in mission planning. Due to the limited accuracy of a road-level map, the generated road-level trajectory is more like a series of driving mission instructions than a specific trajectory that an autonomous vehicle can follow. Such guidance is accurate enough for human drivers. However, in order to drive autonomously, it is necessary to know more about the exact positions where the vehicle should continue going straight or turn. As a result, autonomous vehicles supported by road-level navigation must be equipped with a powerful real-time perception and decision-making system, which greatly increases the onboard computation burden. In contrast, lane-level navigation is able to provide a reference trajectory that can actually be followed by an autonomous vehicle in the absence of other vehicles or obstacles. The key difference between lane-level navigation and road-level navigation is the ability of the former to provide an exact trajectory as the input of control, without the help of an environment perception system. Although a lane-level navigation system cannot replace a real-time perception and decision system, it can greatly release its computation burden and reduce the risk of system failure. Researchers at Carnegie Mellon University have also mentioned the importance of lane-level navigation in their behavioral planning framework [7], in which a traffic-free reference path is required to facilitate the control of vehicle motion.

A lane-level navigation system consists of three major parts: a lane-level map, lane-level localization, and lane-level route planning. With a differential global position system (DGPS) with real-time kinematic (RTK) technology, the global navigation satellite system (GNSS) localization accuracy can reach the centimeter level in open areas. In the case of a lost signal, camera-based or LiDAR-based feature-matching technology could be employed to obtain an accurate position. In our previous study, we employed an inertial measurement unit (IMU) as supplement to the global positioning system (GPS) in order to improve the accuracy and robustness of the localization. In particular, we employed the kinematic bicycle model and the unscented Kalman filter (UKF) algorithm to fuse the measurements from GPS and IMU, as shown in Eq. (1):

$$\text{Production model}: \begin{bmatrix} \dot{X} \\ \dot{Y} \\ \dot{v}_x \\ \dot{\phi} \\ \ddot{\phi} \end{bmatrix} = \begin{bmatrix} \sin(\phi) \cdot v_x \\ \cos(\phi) \cdot v_x \\ a_x - g\sin(\theta) \\ \dot{\phi} \\ 0 \end{bmatrix} + \boldsymbol{\delta}(\text{noise})$$

$$\text{Measurement model}: \begin{bmatrix} X_{\text{gps}} \\ Y_{\text{gps}} \\ v_{\text{gps}} \\ v_{\text{wheel}} \\ \phi_{\text{gps}} \\ \dot{\phi}_{\text{gyro}} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ v_x \\ \phi \\ \dot{\phi} \end{bmatrix} + \boldsymbol{\delta}(\text{noise})$$

(1)

where $X_{\text{gps}}, Y_{\text{gps}}, v_{\text{gps}}, v_{\text{wheel}}, \phi_{\text{gps}}, \dot{\phi}_{\text{gyro}}$ are the measurements of the GPS receiver and IMU; X and Y are the states of vehicle position; v_x is the speed along x-axis; ϕ is the clockwise angle between the north and the vehicle direction; $\dot{\phi}$ is the angular velocity; g is the acceleration of gravity; θ is road pitch angle; and $\boldsymbol{\delta}$ is the noise vector.

More details about localization can be found in the literature and will not be discussed here. This study focuses on the other two parts: the lane-level map and the lane-level route planning, in order to find answers to these two questions: ① What does a so-called lane-level map look like? and ② How is a lane-level map used to generate lane-level route planning?

The definitions of a lane-level map, high-definition (HD) map, or high-precision map are still ambiguous. In general, all these concepts refer to a map that is more accurate and more suitable for autonomous driving than a conventional map. In this study, "lane-level" means that the accuracy is greater than 0.5 m and the differences between different lanes are distinguishable. The lane-level map is becoming a hot research topic, particularly because the measurement of accurate map data is becoming much easier. With the development of high-precision localization sensors and other advanced equipment for data collection, including inertia measurement units and LiDAR, it is possible to establish an enhanced map for the navigation and localization of vehicles [8–10]. With the multi-sensor fusion technique, lane-level accuracy localization has become accessible to many developers [11]. Betaille et al. [12,13] have created an enhanced map for lane-level vehicle navigation that uses a series of shape points to
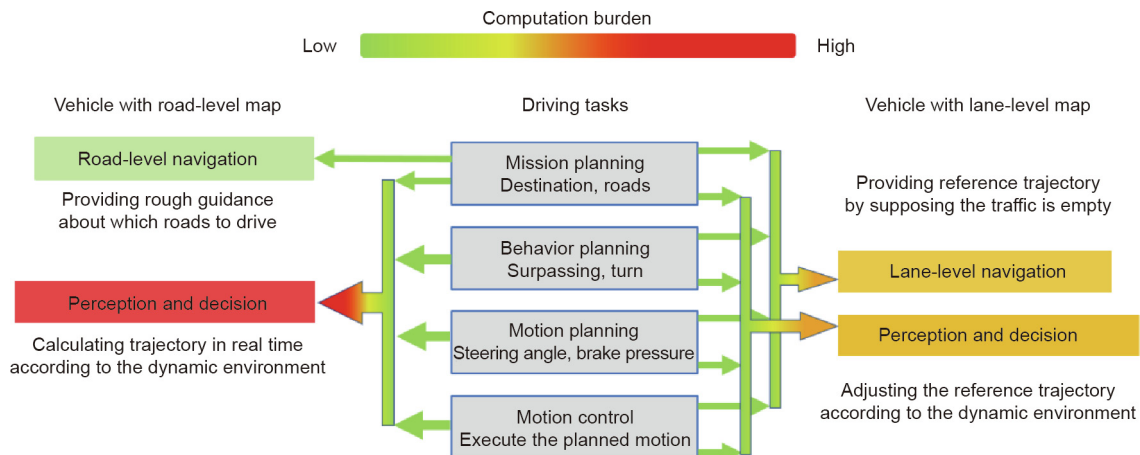

Fig. 1. Different strategies for autonomous driving with road-level navigation and lane-level navigation, respectively.

represent several lane types. Schindler et al. [14] have used circular arc splines to generate a high-precision map that includes road markings, landmarks, and other additional attributes. Liu et al. [15] and Guo et al. [16] have designed a lane-level map-generation method based on sensors and an open street map.

Current studies on lane-level maps mainly generate geometrical representations for lanes with high accuracy. However, the current limitation of lane-level maps is not accuracy but flexibility. Lane-level maps contain a great deal of detailed lane information, such as the slope angle, curvature, and position of all lanes, most of which is not necessary for a given navigation task. For better efficiency, a lane-level map should be more flexible in providing detailed lane information only when necessary. In this study, a multi-layer digital map model is developed to make the digital map more flexible.

Lane-level route planning is more challenging than road-level route planning in two aspects: the evaluation of travel cost and the optimization of efficiency. The conventional road network is usually simplified as the shortest path problem in graph theory, which is conventionally solved by Dijkstra's, A*, and other algorithms [17]. However, in practical navigation applications, these algorithms should be modified to consider the particular properties associated with real road networks [3]. Many road-level route-planning algorithms have been developed in the literature; an overview is given in Ref. [18]. When it comes to the lane level, however, the navigation task is not only to find the shortest path; more factors should be considered, such as the position of lane changing. Without a comprehensive evaluation method, it is difficult to find the optimal path, but relatively easy to find an unfeasible path. The efficiency of route planning is another key aspect that should be considered.

One feasible strategy to achieve efficient lane-level route planning is proposed in Ref. [19], and is based on a hierarchical lane-oriented three-dimensional (3D) road network model. Three candidate routes are first calculated on the roadway centerline network using the *k*-shortest path algorithm. Next, the optimal route is calculated on the roadway-based network, and the navigable route is finally determined on the lane-level network. This strategy is simple and feasible. However, it is possible to lose the distinctive accuracy that could be obtained from lane-level networks. For example, the first three candidate routes may not cover the optimal route if distinct lane features are ignored in the first routing stage. In this study, we design a new travel cost based on the detailed information provided by the lane-level map. In particular, a refined model of travel costs that considers interaction turning and lane changing is achieved, and the optimal route associated with vehicle maneuvers at the lane level is determined. A new lane-level route-planning algorithm is then developed based on the new travel cost models.

The rest of this paper is organized as follows: The next section starts by introducing the architecture of the map model we are proposing, and then explains the relationship between road-level and lane-level maps. Refined modeling of the lane-level travel costs is discussed in Section 3. Next, the lane-level route-planning algorithm is proposed. Experiments are performed on both a grid network and a real lane-level digital map, and the results are presented in Section 5. Finally, we provide conclusions and directions for further research in Section 6.

2. Adaptive accuracy map model

2.1. Map architecture

Owing to the development of LiDAR and other high-precision sensors, digital map accuracy is now able to reach the centimeter level, which has led to an explosion of map data. Several new map standards such as OpenDRIVE [20] and NDS (i.e., navigation data standard) [21] are employed for lane-level routing. However, the overwhelming level of complexity and the confidentiality of these new standards make it difficult for them to be practically used at present. Thus, an applicable map model that can take full advantage of high-precision map data is essential. Note that a map with higher accuracy does not necessarily mean improved navigation. For different navigation goals, the required accuracy can be relatively different. In order to make the digital map capable of supporting all types of navigation in an efficient way, we propose a seven-layer map model, called as Tsinghua map model (TM model), which is illustrated in Fig. 2. Each layer contains different types of data, and is dedicated to different navigation tasks.

Layer 1: Road layer. This layer is designed for road-level mission planning; it provides road-level information that can be used in global road-level route planning [17]. This layer represents conventional static digital map data.

Layer 2: Traffic information layer. This layer is designed for dynamic road-level navigation; it provides road-level dynamic traffic data on events such as traffic jams and road construction that can be used in congestion avoidance and dynamic global path planning.

Layer 3: Road–lane connection layer. This layer is designed for lane-level route planning, providing a topological connection between the road-level network and lane-level network. It can help to topologically project the road-level path into the lane-level path. As it has no detailed information on each lane, this layer is relatively small, and is able to find potential lanes quickly.

Layer 4: Lane layer. This layer is designed for lane-level navigation, and provides high-definition lane-level geometry, lane-level traffic rules, road-signs, and other lane-related information. Layers 3 and 4 can be combined to obtain lane-level optimal trajectory planning that considers the road slope, curvature, and traffic rules.

Layer 5: Map feature layer. This layer is designed for self-localization lane-level navigation; it provides high-definition feature data, which can be used for map-based localization or perception.

Layer 6: Dynamic objects container layer. This layer is designed for dynamic local trajectory planning, and provides dynamic information on moving obstacles observed by other vehicles or infrastructure. It works as a standardized interface for the fusion of multiple sensors (LiDAR, camera, vehicle-to-everything (V2X), etc.) in different vehicles. With information from other road users, the dynamic situation can be considered to generate safe local trajectory.

Layer 7: Intelligent decision support layer. This layer is designed for autonomous driving, and provides driving knowledge data. An intelligent driving support system can be embedded into this layer in order to learn a driver's logic and driving behaviors.

One significant benefit of using the seven-layer map is the possibility to load data in a flexible way. For road-level navigation, the data from Layers 1 and 2 are enough. For lane-level navigation with RTK GPS, Layers 3 and 4 should be added. If the GPS does not work, Layer 5 can help to localize the vehicle. For cooperative driving, Layer 6 should be added to obtain information transferred by V2X communication. For autonomous driving, Layer 7 provides decision-making support.

With the proposed TM model, the use of map data is more flexible, map information would be invoked only when it is required by the navigation task. In the following subsection, the lane-level map model is introduced. It should be noted here that Layers 5, 6, and 7 are designed for map-based localization, perception, and decision-making, which are outside the scope of this study.
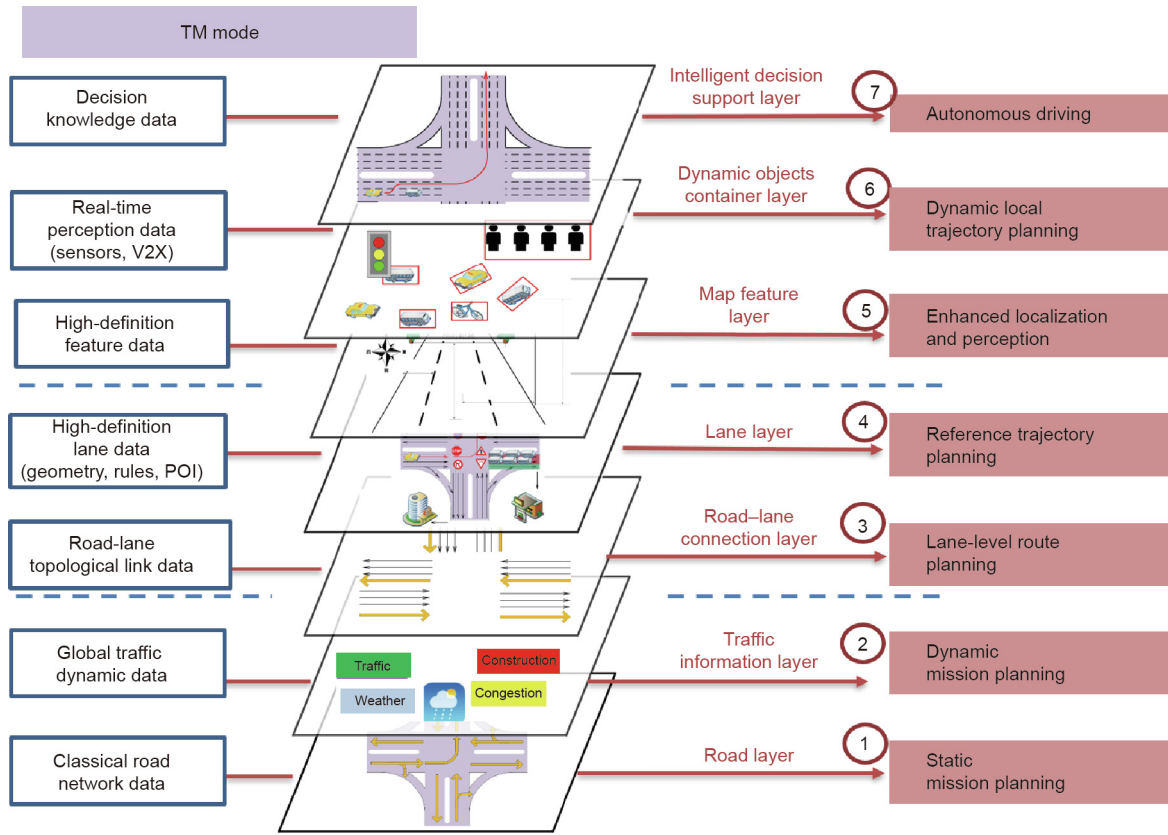
Fig. 2. A seven-layer adaptive accuracy map architecture for autonomous driving. V2X: vehicle-to-everything; POI: point of interest.

2.2. The lane-level map model

The lane-level road network should contain the geometric and topological details of roads and intersections at the lane level, as represented by Layer 4 in Fig. 2. A variety of lane-level map models have been proposed in the literature to expand a digital map with detailed lane information, including refined representations of the lane geometry and intersection geometry. Lane geometry is usually represented by clothoids [13], arc splines [14], or cubic Hermite splines (CHSs) [10], while road intersections are described using a topological method [15,19,22]. In comparison with adding more detailed information into the map, it is equally important to make the map more flexible and efficient. For the design of an efficient lane-level navigation system, we propose the construction of a lane-level map model with a four-layer structure, comprised of Layers 1, 2, 3, and 4, as shown in Fig. 2.

The mathematical representation of each layer in Fig. 3 is introduced in the following paragraphs.

2.2.1. Road layer

Without loss of generality, we consider that the road network is composed of a set of roads $\{r_j\}_{j=1}^{a}$ and a set of intersections $\{c_i\}_{i=1}^{b}$, a and b are the numbers of the roads and intersections, respectively. The road layer is usually expressed as follows:

$$W = \left(\{r_j\}_{j=1}^{a}, \{c_i\}_{i=1}^{b}\right)$$
$$c_i = (P_{c,i}, E_{c,i}, T_{c,i}) \tag{2}$$

where W is the representation of the road layer; $P_{c,i}$ is the set of road-level nodes entering the ith intersection and $E_{c,i}$ is the set of road-level nodes leaving this intersection. $T_{c,i}$ is the road-level traffic matrix indicating the topological connection between two nodes.

$$T_i = \begin{array}{c} \\ E_{c,j,1} \\ \vdots \\ E_{c,j,m} \end{array} \begin{array}{ccc} P_{c,j,1} & \cdots & P_{c,j,n} \\ \begin{bmatrix} t_{i,1,1} & \cdots & t_{i,n,1} \\ \vdots & \ddots & \vdots \\ t_{i,1,m} & \cdots & t_{i,n,m} \end{bmatrix} \end{array} \tag{3}$$

where m and n are the number of the nodes leaving and entering the intersection, respectively. The element $t_{i,n,m}$ identifies whether a vehicle can drive from the entering node $P_{c,j,n}$ to the leaving node $E_{c,j,m}$, and $t_{i,n,m} = (f_i, m_i)$, in which f_i denotes whether (= 1) or not (= 0) a vehicle may drive from $P_{c,j,n}$ to $E_{c,j,m}$ and m_i is the way of passage, such as turning left, turning right, going straight, or performing a U-turn.

The road-level road is expressed as follows:

$$r_i = (P_{r,i}, E_{r,i}, Q_{r,i}) \tag{4}$$

where P_r is the set of road-level nodes entering this road intersection, and E_r is the set of road-level nodes leaving this road intersection. Q_r comprises the road class k_r and the road length l_r.

2.2.2. Traffic information layer

Traffic information is essential for the routing algorithms in order to avoid traffic jams and improve global traffic efficiency. Global traffic dynamics can be obtained through an analysis of floating vehicle data. In this study, the traffic situation is represented by the vector of the recorded average speed.

$$V_r = [V_r(t_0), \ldots, V_r(t_i), \ldots, V_r(t_k)] \tag{5}$$

where $V_r(t_i)$ is the average speed during the time interval $[t_i, \ t_i + \Delta t]$, t_k is the time of last recorded speed.
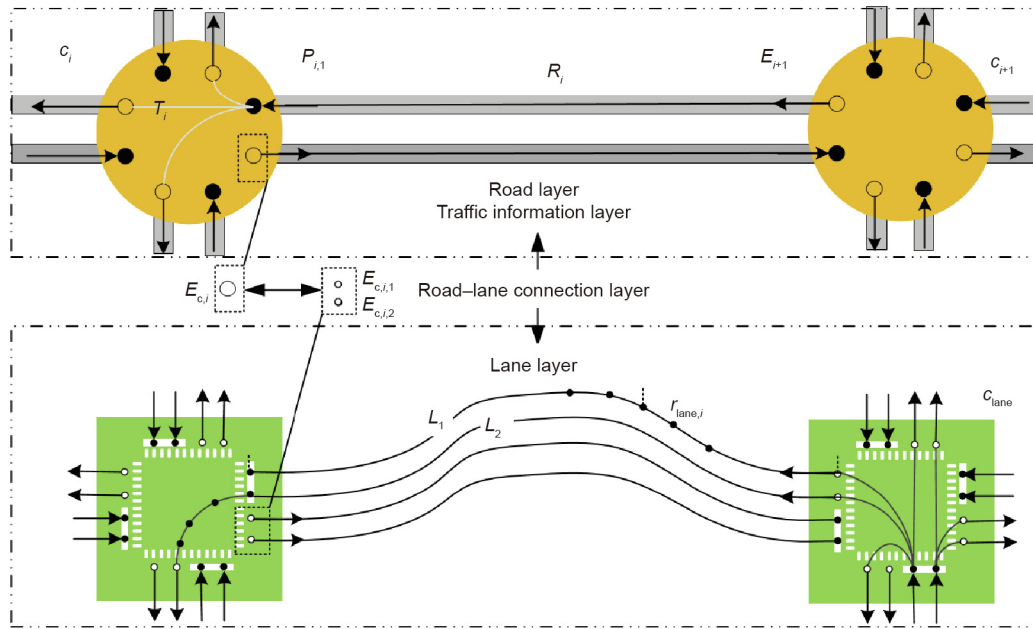
Fig. 3. The four-layer lane-level map model for lane-level navigation.

The history of the average speed data can be used as experience data to estimate the travel cost. It can be also be used to predict the future travel speed.

$$\hat{V}_r = \left[\hat{V}_r(t_{k+1}), \dots, \hat{V}_r(t_{k+i}), \dots, \hat{V}_r(t_{k+n}) \right] \tag{6}$$

where $\hat{V}_r(t_{k+i})$ is the predicted speed during the time interval $[t_k + i\Delta t, \ t_i + (i+1)\Delta t]$.

2.2.3. Road–lane connection layer

The role of this layer has already been explained in the previous subsection. This layer can be regarded as a connection layer between the road-level information and the lane-level information, which connects the upper and lower layer. This layer is the key to a flexible and accurate architecture, as it provides channels to load high-precision data and preserves the advantages of conventional maps. Thus, the connection layer is expressed as follows:

$$(P_r, E_r) = (P_{lane,1}, P_{lane,2}, \dots, P_{lane,n}, E_{lane,1}, E_{lane,2}, \dots, E_{lane,m}) \tag{7}$$

where $P_{lane,n}$ and $E_{lane,m}$ represent the lane-level entering and leaving nodes, respectively. Each lane-level node contains the location of the node and the traffic direction of the node, noted as $(u_N, \ v_N)$.

2.2.4. Lane layer

The lane layer contains more accurate and complete information.

$$W_{lane} = \left(\{r_{lane,j}\}_{j=1}^{a_{lane}}, \{c_{lane,i}\}_{i=1}^{b_{lane}} \right) \tag{8}$$

where W_{lane} represents the lane-level layer; a_{lane} and b_{lane} are the numbers of traffic lane and lane-level intersections; $r_{lane,j}$ represents a traffic lane belonging to one physical road, where the lane is defined as follows:

$$r_{lane,j} = (S_{lane}, s, c_s, c_e, Q_{lane}) \tag{9}$$

where S_{lane} is the parameter set of the lane shape, which is defined by the CHS. The lane curve can be expressed by a parametric function $f_{CHS}(u, S_{lane})$, in which u is the position of the vehicle. s denotes the lateral sequence number of the lane in the set of lanes belonging to the same road. The sequence is determined so that number 1 is

assigned to the rightmost lane. c_s and c_e are the indices of the start and end intersections, respectively, according to the traffic direction of the lane. Q_{lane} comprises the lane attributes such as the total length L and the speed limit V_{limit}, along with other geometric and traffic attributes, such as the traffic sign $B(u)$. $B(u) = 1$ indicates that a stop sign is located at $f_{CHS}(u, S_{lane})$.

The lane-level intersection $c_{lane,i}$ is defined similarly to the road-level intersection, as follows:

$$c_{lane,i} = (P_{lane,i}, \ E_{lane,i}, \ T_{lane,i}) \tag{10}$$

where $T_{lane,i}$ is the lane-level traffic matrix, which represents the geometric and topological characteristics of the joint between the entering and exiting lanes (i.e., λ_i and λ_j) at the intersection. The joint lane is defined as follows:

$$t_{ij} = (S_t, \lambda_i, \lambda_j, Q_t, f_m) \tag{11}$$

where S_t is the parameter set of a possible trajectory of a vehicle traveling from lane λ_i to λ_j and is also defined by CHS. λ_i and λ_j are the indices of the entering and exiting lanes. Q_t comprises attributes associated with the joint lane, such as length L and traffic light B_s. $B_s = 1$ means that the traffic associated with the joint lane is controlled by a traffic light. f_m describes whether the lane λ_i to λ_j can feasibly be connected. Note that the lane-level traffic restraints at the intersections are inherently described by the traffic matrix.

3. Modeling of lane-level travel cost

Given detailed information on the lane-level road network, a more refined model based on travel costs can be achieved. In general, the cost in terms of travel time can be categorized into three types: ① static, ② dynamic, and ③ statistic [23]. Static cost depends on the stationary properties of the road network, whereas dynamic and statistic costs are affected by other traffic participants. In this study, we focus on the static cost in order to explore the improvement in cost modeling derived from the refined road network data. There are many aspects to route planning, such as travel distance, time, fuel consumption, and various mixed indicators. Here, we focus on time; all the costs appearing in this study are travel time. Details will be discussed below.

3.1. Travel cost of lane changing

The travel time along a single lane can easily be estimated according to the lane attributes, as follows:

$$f_{\text{cost}}(r_{\text{lane}}) = L/V_{\text{limit}} \tag{12}$$

where L and V_{limit} are involved in Eq. (9). However, in most cases, a driver may not follow a single lane but may make lane changes. The cost of a lane change is discussed as follows. A lane change is generally performed because of the different properties of two lanes. (Note that dynamic factors are not considered in this study.) We consider two common scenarios in which lane changes could occur.

3.1.1. Lanes with different speed limits

In this case, a vehicle traveling in a lane with a lower speed limit is assumed to make a lane change as soon as possible in order to travel at a higher speed. We assume that the lane change is performed at the beginning of the lanes; accordingly, a transit edge is added between two start points, as shown in Fig. 4(a).

The cost of the transit edge should consider the delays derived not only from the extra length traveled but also from the accelerating course. The cost can be obtained by the following equation:

$$f_{\text{cost}}(r_{\text{lane},i} \rightarrow r_{\text{lane},j}) = \frac{(V_i - V_j)^2}{2a_v V_i} + \frac{\Delta s}{V_i} \tag{13}$$

where a_v is the vehicle acceleration and Δs is the extra length caused by lane changing, which can be approximated by the lane width.

3.1.2. Lanes with different traffic constraints

A vehicle traveling on a lane with a higher speed limit may also change to other lanes in order to make a turn to follow the traffic constraint assigned to that lane. As shown in Fig. 4(b), this lane change may generally be made close to the intersection ahead, and is therefore described by a transit edge between the end nodes of the two lanes. The additional cost is obtained and expressed as follows:

$$f_{\text{cost}}(r_{\text{lane},i} \rightarrow r_{\text{lane},j}) = \frac{(V_i - V_{\text{traffic}})^2}{2a_v V_i} + \frac{\Delta s}{V_i} \tag{14}$$

where V_{traffic} is the speed limit caused by the traffic restraints, rather than the speed limit road sign.

Figs. 4(c) and (d) show the transit edges that model the possible lane changes. In the case where a road contains more than two lanes, transit edges between two adjacent lanes should be generated.
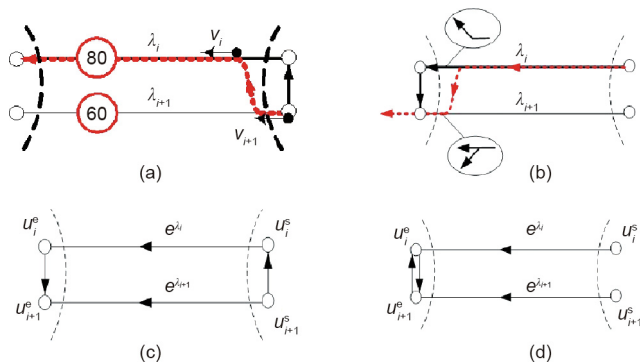
3.2. Travel cost at an intersection

The cost of turns at intersections also plays an important role in the entire cost of a route, particularly in urban areas. This issue has been considered in previous studies [23]; however, it can be improved using detailed data on intersections, which are described at the lane level. The turning cost should include not only the travel length at the intersection, but also the reduction in speed. The vehicle movement across an intersection can be divided into three sections, as shown in Fig. 5.

3.2.1. Approaching an intersection

In general, a vehicle decelerates when approaching an intersection. The time delay due to deceleration can be obtained by the following:

$$f_{\text{cost}}(c_{\text{appro}}) = \frac{(V_i - v_c)^2}{2a_v V_i} \tag{15}$$

where v_c is the vehicle speed at the end of the entering lane. It should be affected by various traffic features assigned at that point. v_c is equal to zero if the road sign is "stop." In most cases, v_c is equal to the turning speed v_t, which will be discussed below.

3.2.2. Turning at an intersection

The time of turning at an intersection can be divided into two parts: the time of waiting for signals and the time of turning behavior. The time of waiting for signals is a subject that is more related to road traffic management. In this study, the focus is on the modeling of driving behaviors; therefore, we simplify the problem by assuming that the signal waiting time is given as t_{wait}. After entering an intersection, a vehicle accelerates from v_c to the turning speed v_t and travels along the joint lane. The travel time is calculated by the following:

$$f_{\text{cost}}(c_{\text{turn}}) = \frac{(v_t - v_c)^2}{2a_v V_i} + \frac{L_T}{v_t} + t_{\text{wait}} \tag{16}$$

where L_T is the length of the joint lane. The turning speed is affected by several factors that are associated with the turning type and the intersection configurations. Thus, we define a model for turning speed that considers the influence of the attributes of an intersection:

$$v_t = v_b(1 - \kappa r_{\min}) \tag{17}$$

where v_b is the base speed obtained by $\min(V_i, V_j)$, and κ is the average curvature of the joint lane, which can be calculated along the joint lane curve [24]. $r_{\min}$ is the minimum turning radius (e.g., 6 m for a typical passenger car).
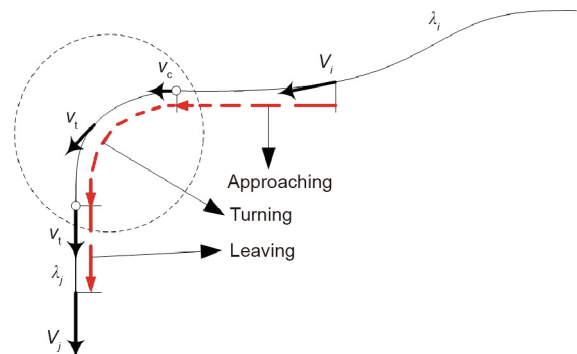

Fig. 4. Modeling of lane change cost. (a) Lane change due to different speed limits (km·h^{-1}); (b) lane change due to different traffic constraints; (c) transit edges between lanes with different speed limits, represented by e in the figure; (d) transit edges between lanes with different traffic constraints.

Fig. 5. Modeling the turning cost. v_c: the vehicle speed at the end of the entering lane; v_t: the turning speed.

3.2.3. Leaving the intersection

When leaving the intersection, the vehicle accelerates to reach the speed of the exiting lane. The time delay can be obtained by the following:

$$f_{\text{cost}}(c_{\text{leave}}) = \frac{(V_j - v_{\text{t}})^2}{2a_{\text{v}}V_j} \tag{18}$$

Finally, the total travel cost at the intersection is calculated by the following:

$$f_{\text{cost}}(c_{\text{inter}}) = f_{\text{cost}}(c_{\text{appro}}) + f_{\text{cost}}(c_{\text{turn}}) + f_{\text{cost}}(c_{\text{leave}}) \tag{19}$$

4. Lane-level route planning

Now we consider the route-planning problem on the lane-level road network. The detailed attributes assigned to separate lanes enable us to determine the optimal route while considering vehicle maneuvers at the lane level, as shown in Fig. 6.

First, we add nodes at the start and end points of each lane. The whole road network can therefore be summarized as a directed graph.

$$G = (U, E) \tag{20}$$

where U is the set of nodes $\{u_i\}$ and E is the set of edges $\{e_i\}$. According to the road network topology, an edge is derived from either an entering lane lane$_i$ connecting its start node $u_{\text{start}}(e^{\text{lane}_i})$ (simplified as u_i^s) and end node $u_{\text{end}}(e^{\text{lane}_i})$ (simplified as u_i^e), or a joint lane t_{ij} connecting the end node u_i^e of the entering lane lane$_i$ and the start node u_j^s of the exiting lane lane$_j$.

Next, the weight function $f_{\text{cost}}(e)$ is defined to specify a weight to each edge. The weight indicates the cost of traversing a lane or making a turn in terms of a specified metric, such as distance or travel time. In this paper, we focus on the most natural metric, the travel time. The detailed modeling of the lane and turn costs has already been described in Section 3.

The route planning aims to find the optimal route $P = \{e_1, \ldots, e_p\}$ satisfying the following:

$$\min c(P) = \sum_{i=1}^{p} f_{\text{cost}}(e_i) \tag{21}$$

where $e_i \in E$, for all lanes in the searching area, and $c(P)$ is the total cost along the route P.

As discussed earlier, the route-planning problem may be solved using classical algorithms such as Dijkstra's or the A* algorithm. However, employing these algorithms directly on a lane-level road network is not practical due to the significant increase in complexity in routing. Various heuristic rules and hierarchy structures have been exploited to realize efficient routing in large networks [25]. In this study, we also propose a hierarchical route-planning method that provides a lane-level routing result with improved efficiency. The second reason why we propose this algorithm is that many algorithms have been available for road-level route planning for decades, which may be convenient for us. Thus, we regard this lane-level route-planning problem as a process of determining a lane-level route from a predetermined starting point to a predetermined destination point. At first, we plan a route on the road level, as shown in Fig. 7(a); next, lane-level costs are calculated on lane-level layers; and finally, lane-level nodes making up the route are determined, as shown in Fig. 7(b).

4.1. Route planning on the road-level layer

Assuming that nodes O and D are the predetermined origin and destination, respectively, the route planning on the road level, N_{r}, can be formulated as a search for road-level intersection sequences.

$$N_{\text{r}} = (O, N_1, N_2, \ldots, N_i, \ldots, N_k, D) \tag{22}$$

where N_i represents the road-level intersection node, k is the number of nodes.

Our road-level map model is similar to the conventional map model for navigation. The A* algorithm is employed in this navigation step [18].

4.2. Route planning on the lane-level layer

Path refinement should be realized based on the obtained road-level intersection sequences, according to our proposed hierarchical route-planning method. The lane-level route planning can be divided into two steps: ① selecting lane-level nodes based on road-level routing results, and ② connecting lane-level nodes according to the lane-level travel cost.

4.2.1. Selection of lane-level route nodes

According to the lane-level map model introduced in Section 2, each road intersection has lane-level nodes entering and exiting the area. The map data in the road-lane connection layer are
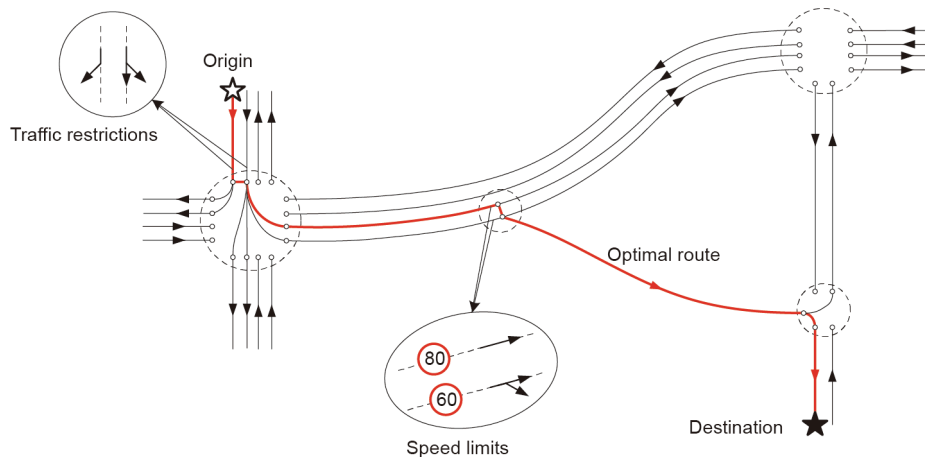

Fig. 6. The optimal route on the lane-level road network (represented by the red line). Traffic restrictions and speed limits (km·h^{-1}) associated with traffic lanes should be taken into consideration in determining the optimal route.
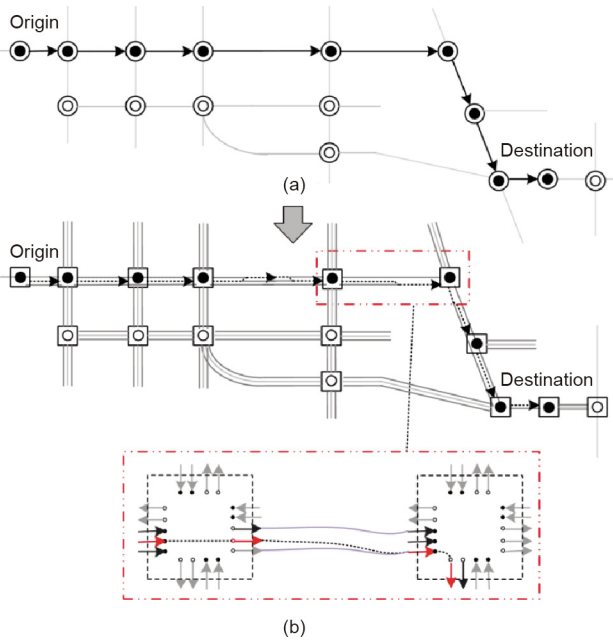
Fig. 7. Hierarchical route planning based on the multi-layer map. (a) Road-level route planning; (b) lane-level route planning.

designed to support the selection of lane-level route nodes from road-level route nodes. After obtaining the output of the road-level route planning, such as $N_1, N_2, \ldots, N_k$, the simplest way to obtain the refined nodes is to read all the lane-level nodes in $N_1, N_2, \ldots, N_k$; however, this will bring in unrelated nodes and reduce the efficiency of route searching.

To eliminate unrelated lane-level nodes, we propose the extraction of lane-level nodes according to the following methods, as shown in Fig. 8.

First, only the lane-level nodes that can connect two consecutive road-level nodes are selected. For a road-level intersection node N_i, its entering lane-level nodes should be connected to the last road-level intersection node N_{i-1}. Similarly, its exiting lane-level nodes should be connected to the next road-level intersection node $N_{i,i+1}$. The entering and exiting lane-level nodes of N_i are denoted as $N_{i,i-1}$ and $N_{i,i+1}$, obtained by the following equations:

$$N_{i,i-1} = P_c(N_i, N_{i-1}) \tag{23}$$

$$N_{i,i+1} = E_c(N_i, N_{i+1}) \tag{24}$$

where P_c represents the road-level entering node and E_c represents the road-level exiting node, both determined by the road-level route planning result.

Second, the constraints introduced by the traffic rules or road conditions should be considered. Imagine that in an intersection,
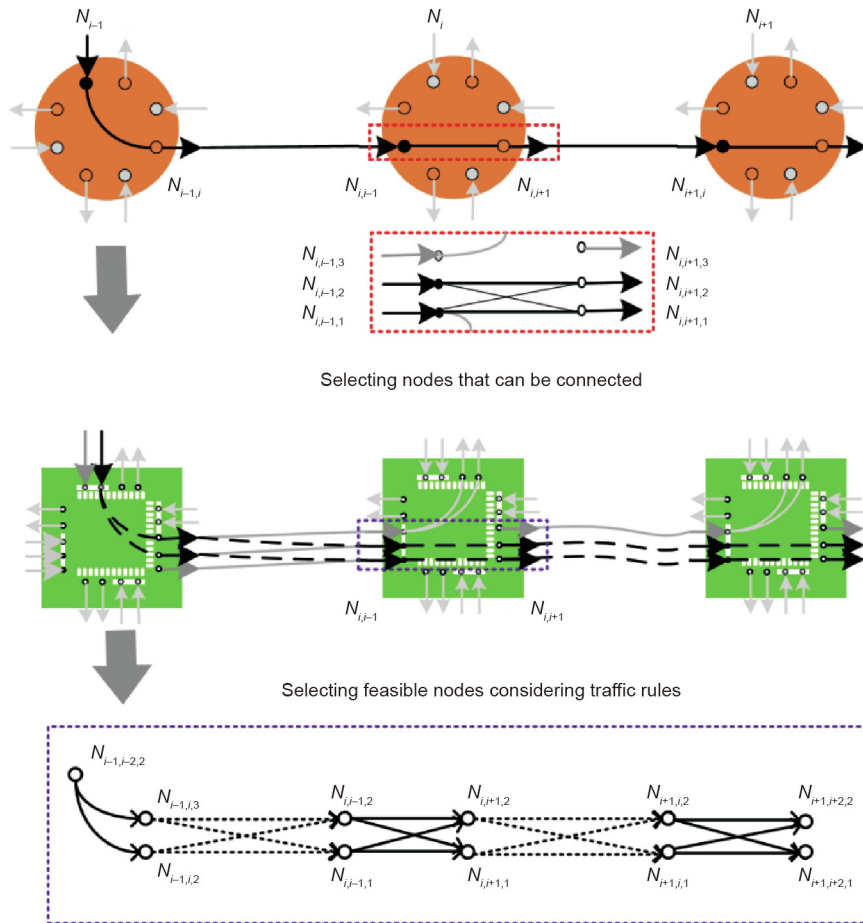

Fig. 8. An illustration of the method of selecting lane-level route nodes.

the left turning is determined by the road-level nodes sequence, and this road-level node includes three lane-level entering nodes. According to the traffic rules, the lane-level nodes that are most to the left are the only feasible lane-level nodes. To consider similar situations in the route planning, it is necessary to use the traffic matrix to eliminate unfeasible nodes. The data stored in the traffic matrix are explained by Eqs. (10) and (11). Given the specific travel task, from $N_{i,i-1}$ to $N_{i,i+1}$, the traffic matrix in N_i can be further simplified as $t(N_{i,i-1}, N_{i,i+1})$.

$$t(N_{i,i-1}, N_{i,i+1}) = t(N_{i,i-1}, N_{i,i+1}) \bigcup t(N_{i,i-1,n}, N_{i,i+1,m});$$
$$\text{if } f_m(N_{i,i-1,n}, N_{i,i+1,m}) = 1 \tag{25}$$

Only the feasible entering and exiting lanes are included in the traffic matrix. In this way, the number of nodes to be searched is reduced. Suppose that in the routing node N_i, the number of feasible entering and exiting lanes are n and m, respectively; then the traffic matrix in N_i can be formulated as the following equation.

$$T(N_{i,i-1}, N_{i,i+1}) = \begin{matrix} & N_{i,i-1,1} & \cdots & N_{i,i-1,n} \\ \begin{matrix} N_{i,i+1,1} \\ \vdots \\ N_{i,i+1,m} \end{matrix} & \begin{bmatrix} t(N_{i,i-1,1}, N_{i,i+1,1}) & \cdots & t(N_{i,i-1,n}, N_{i,i+1,1}) \\ \vdots & \ddots & \vdots \\ t(N_{i,i-1,1}, N_{i,i+1,m}) & \cdots & t(N_{i,i-1,n}, N_{i,i+1,m}) \end{bmatrix} \end{matrix} \tag{26}$$

The pseudocode of the selection rule is also listed in Algorithm 1.

Algorithm 1. Pseudocode: Selection of lane-level route nodes

1 N_r is known, set $N_P = \varnothing$
2 for $i = 2, \ldots, k-1$
3 set $N_{i,i-1} = \varnothing, \quad N_{i,i+1} = \varnothing$,
4 $N_i, N_{i-1} \to P_c \to N_{i,i-1}, N_i, N_{i+1} \to E_c \to N_{i,i+1}$
5 set $t(N_{i,i-1}, N_{i,i+1}) = \varnothing$
6   for $x = 1, \ldots, n_{i,i-1}$
7   for $y = 1, \ldots, n_{i,i+1}$
8   if $f_x(N_{i,i-1,x}, N_{i,i+1,y}) = 1$
9       $t(N_{i,i-1}, N_{i,i+1}) = t(N_{i,i-1}, N_{i,i+1}) \cup [N_{i,i-1,x}, N_{i,i+1,y}]$
10   else $N_{i,i-1} = N_{i,i-1}/N_{i,i-1,x}, N_{i,i+1} = N_{i,i+1}/N_{i,i+1,y}$
11   find the lane-level node which has no link in this road node then delete it;
12   end if;
13   end for;
14 end for;
15 $N_P = N_P \cup [N_{i,i-1}, N_{i,i+1}, t(N_{i,i-1}, N_{i,i+1})]$
16 end for;
17 return N_P

N_p: the route planning result on the lane level, which is a set of lane-level nodes.

4.2.2. Route planning on selected lane-level nodes

After the potential route nodes are selected, the main differences between the road-level and lane-level route planning are mainly the travel cost model, as described in Section 3, and the direction of searching. Our lane-level direct graph is unidirectional, because the global direction is given by the route planning at the road level. Learning from conventional path-finding algorithms, we designed a stepwise algorithm to find the series of optimal lane-level nodes, as illustrated in Fig. 9. For each step, the calculation of the costs from the start point to the current point is completed; then the last lane-level node from which the minimum cost can be realized is stored, denoted as $P_f(N_{i,i-1,m})$. Thanks to the unidirectional graph, it is not necessary to estimate the cost from the current node to the destination, which makes the

algorithm more efficient. Details about lane-level path finding are provided in Algorithm 2.

Algorithm 2. Pseudocode: Route planning on the lane-level layer

1 set $i = 1, j = k$ (number of road-level nodes in addition to O and D), $N_{0,1} = (O), N_{k+1,k} = (D), N_l = \varnothing$
2 while $i < j$
3   for $N_{i,i-1,x} \in N_{i,i-1}$
4     $g(N_{i,i-1,x}) = \infty, P_f(N_{i,i-1,x}) = \varnothing$,
5   for $N_{i-1,i,y} \in N_{i-1,i}$
6      $g'(N_{i,i-1,x}) = g(N_{i-1,i,y}) + t_{lc}(N_{i-1,i,y}, N_{i,i-1,x})$
7   if $g'(N_{i,i-1,x}) < g(N_{i,i-1,x})$
8       $g(N_{i,i-1,x}) = g'(N_{i,i-1,x})$
9       $P_f(N_{i,i-1,x}) = (N_{i-1,i,y})$
10   end if;
11   end for;
12 end for;
13   for $N_{i,i+1,z} \in N_{i,i+1}$
14     $g(N_{i,i+1,z}) = \infty, \quad P_f(N_{i,i+1,z}) = \varnothing$,
15   for $N_{i,i-1,x} \in N_{i,i-1}$
16      $g'(N_{i,i+1,z}) = g(N_{i,i-1,x}) + t_{lc}(N_{i,i-1,x}, N_{i,i+1,z})$
17   if $g'(N_{i,i+1,z}) < g(N_{i,i+1,z})$
18       $g(N_{i,i+1,z}) = g'(N_{i,i+1,z})$
19       $P_f(N_{i,i+1,z}) = (N_{i-1,i,z})$
20 $i = i + 1$;
21   end if;
22   end for;
23 end for;
24 if $i = j$, return path $N_l = \{D, P_f(D), P_f[P_f(D)], \ldots, O\}$;
25   end if;
26 end

g: travel cost from start to previous node; g': travel cost from start to current node; t_{lc}: travel cost of this step.

5. Experimental validation

5.1. Simulation tests on a grid network

We performed simulation tests on a grid network in order to evaluate the effectiveness of the proposed route-planning algorithm. As shown in Fig. 10, a 5 km grid network containing eight roads in the horizontal direction and eight roads in the vertical direction was built. All roads between every two adjacent intersections were bidirectional, and three parallel lanes were included in each direction in one road, as shown in Fig. 11. The traffic rule we designed on the three entering lanes in an intersection was left, straight, and straight/right from the innermost lane to the outermost lane. The lane parameters are given in Table 1. The start point and destination point, indicated by red circles, were located as shown in Figs. 10 and 11.

Given the road-level map data in Fig. 10 and the lane-level map data in Fig. 11, we applied the method described in Sections 2 and 3 to find the optimal route in an efficient way. To better present the effectiveness of our proposed method, the direct lane-level routing algorithm was also performed, which involved directly searching for routes in the lane-level map without the hierarchical process. The routing results of our proposed method are illustrated by red lines in Figs. 12 and 13, where the former presents the road-level routing results and the latter presents the lane-level routing results. In Fig. 12, it can be seen that the
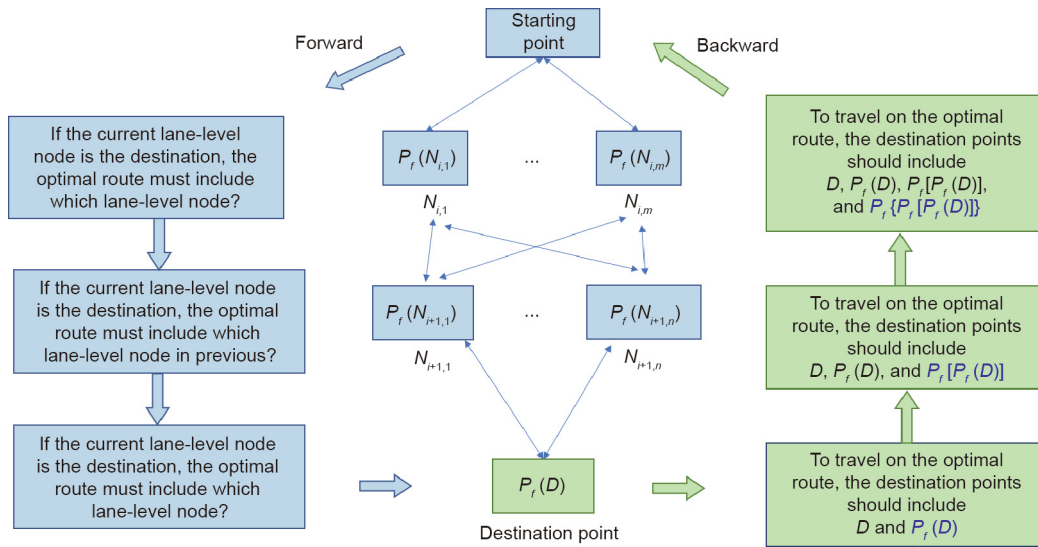
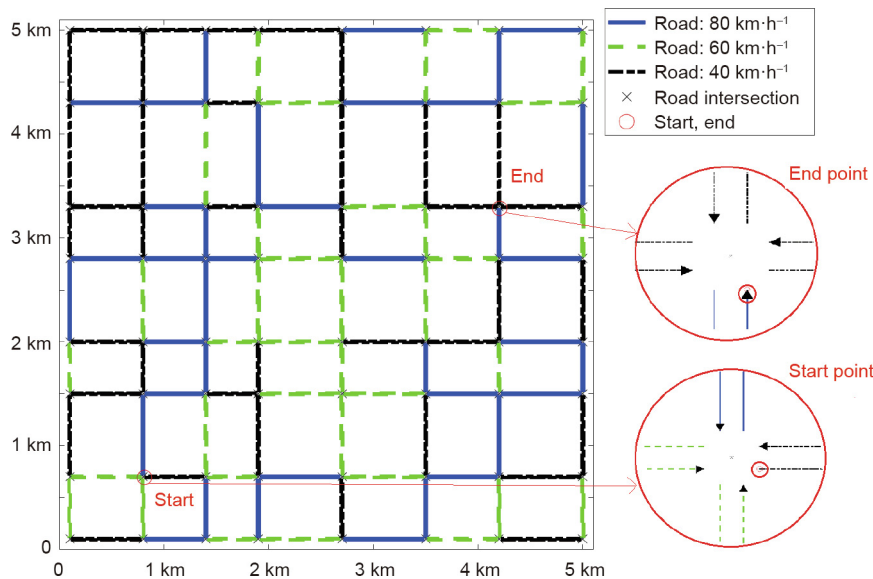**Fig. 9.** The proposed stepwise algorithms to find optimal route nodes.

Fig. 10. Setup of the simulation test: A road-level map, where traffic situations are indicated by different travel speeds for each road.

chosen path avoids all black lanes when it can choose to do so—for example, at Point 3 of Fig. 12. This is because the black lanes are the slowest lanes, with a speed limit of 40 km·h⁻¹. In Fig. 13, it can be seen that the proposed algorithm has successfully chosen the correct lanes to both respect the traffic rules and run on the fastest lane. For example, at Point 2 of Fig. 13, the planned path has changed to the outer lane to turn right; at Point 3 of Fig. 13, the planned path has changed into the inner lane after passing the intersection in order to reduce the travel time. This result reveals that the multi-layer map model is efficient for hierarchical lane-level route planning, and that the proposed planning method and lane-level cost model is capable of finding the lane-level optimal route in a road network.

To demonstrate the improved efficiency of our proposed method, we compared the computation time of different routing algorithms in Table 2. The computation time of the proposed hierarchical routing algorithm can be divided into three parts: initial-

izing the directed graph, road-level routing, and lane-level routing refinement. In comparison, the computation time of the direct lane-level routing algorithm contains two parts: initializing the directed graph, and lane-level routing. The time cost of initializing a directed graph decides the map model's ability to support routing algorithms. As shown in Table 2, our proposed method uses much less time to generate the directed graph than the direct lane-level routing: Our time cost is only 2% of the method of the direct lane-level routing. This is due to the complexity of the lane-level map data. Table 2 also shows that the core routing time cost of our proposed method is 41.5% less than that of the direct lane-level routing. Although the hierarchical routing method divides the routing into two steps, the computation burden is much lower than that of the direct routing method. In conclusion, the simulation test validated the proposed hierarchical routing as having better efficiency in supporting lane-level navigation than the direct routing method.
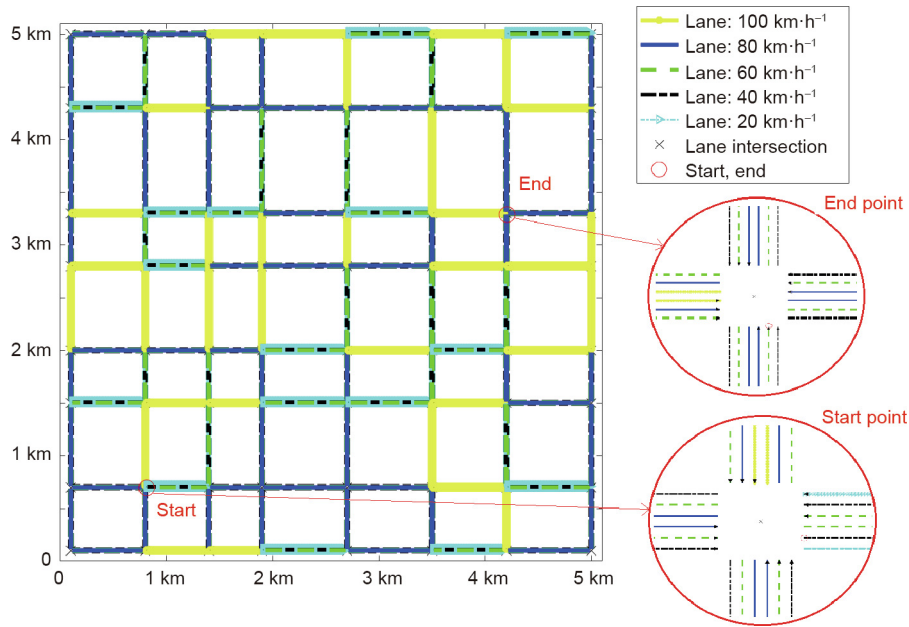
Fig. 11. Setup of the simulation test: A lane-level map, where traffic situations are indicated by different travel speeds for each lane.

Table 1
Setup of parameters in the simulation test.

Parameters	Value
Lane numbers in one road	3
Lane width	4 m
Intersection width	14 m
Average speed limit of road: v_x	Randomly chosen from {80 km·h^{-1}, 60 km·h^{-1}, 40 km·h^{-1}}
Speed limit of each lane	Inner: v_x + 20 km·h^{-1}; middle: v_x; outer: v_x − 20 km·h^{-1}
Allowed behavior	Inner: only left; middle: only forward; outer: forward or right

5.2. Tests on a real road network

Tests were also performed on small areas of the real road network in order to validate our travel cost model and the hierarchical

routing algorithm. In our previous study, a lane-level digital map of a small urban area in Milan was built [25], as shown in Fig. 14. The map contains 44 lanes and 16 intersections. Yellow lines with a gray background indicate parallel lanes where lane changes are permitted (i.e., lanes that are separated by dashed lane markings).

The travel cost of a determined route can be calculated according to the digital map and verified against the real travel time. Two routes were selected, as shown in Fig. 14. One route, indicated by green lines, is 906 m long and contains eight intersections, three traffic lights, and four yield signs. The other route, indicated by blue lines, is 1132 m long and contains nine intersections, four traffic lights, and three yield signs. We drove an experimental car over the selected routes several times during off-peak hours. The travel time for each trip was recorded and compared against the modeled travel cost. The results are shown in Table 3. It can be seen that in some trips, the travel time values were significantly larger than others (i.e., the first trip of Route 1 and the third trip of Route 2). This was due to unexpected traffic delays such as vehicle queuing
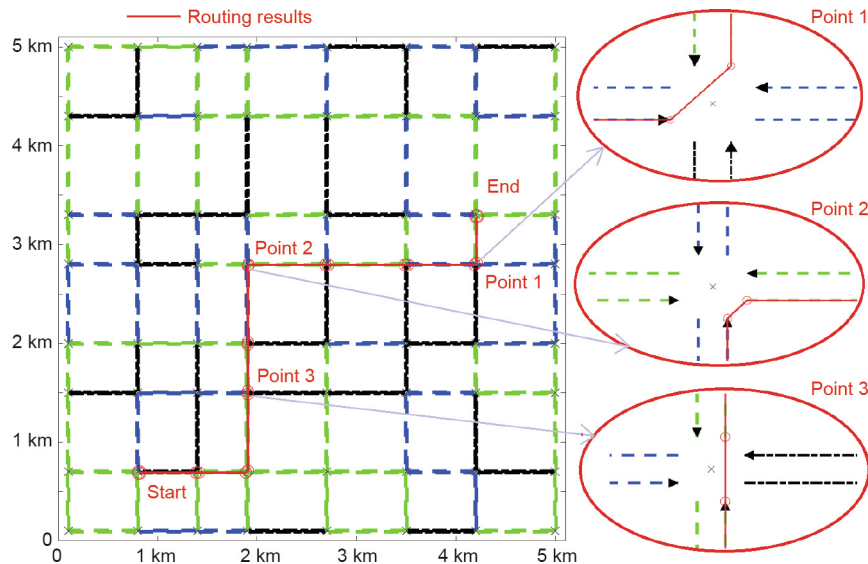

Fig. 12. Road-level routing result: The proposed method has chosen the fastest routes. (See Fig. 10 for an explanation of the speed limit of each road.)
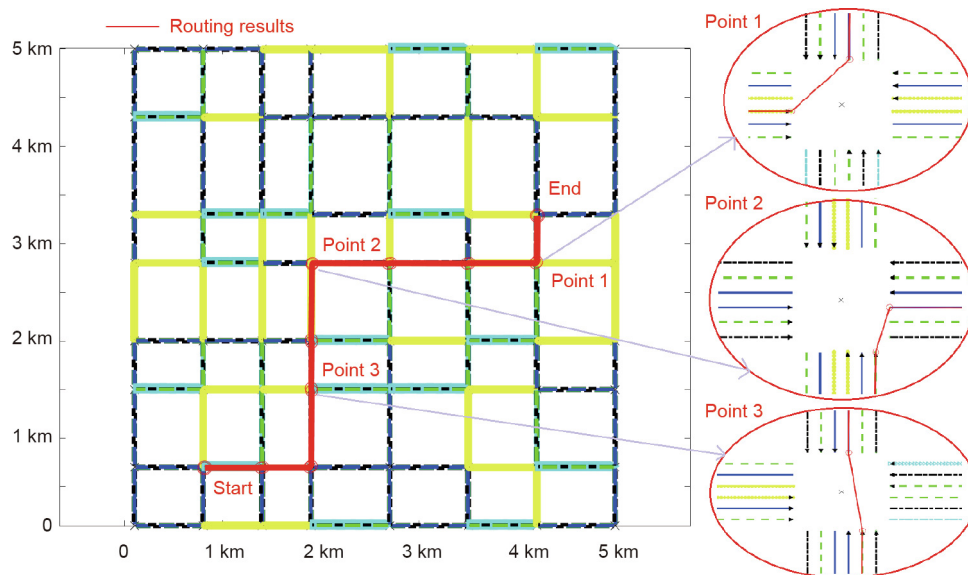
Fig. 13. Lane-level routing result: The proposed method has chosen the correct lane with respect to the traffic rules. (See Fig. 11 for an explanation of the speed limit.)

Table 2
Time costs of different routing methods in the simulation test.

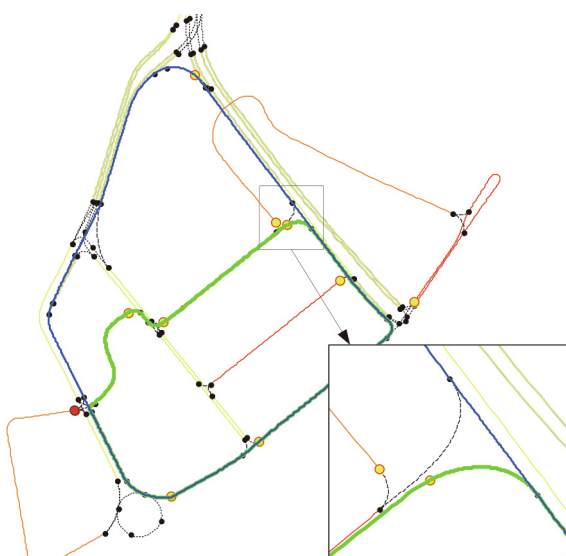|	Proposed hierarchical routing			Direct routing on lane-level map	
	Directed graph over road-level map	Road-level routing	Road-level to lane-level routing	Directed graph over lane-level map	Lane-level routing
Number	224 (node)	768 (edge)	54 (node)	672 (node)	3072 (edge)
Time cost (s)	0.012	0.029	0.002	0.603	0.053
Core routing time cost (s)		0.031			0.053

Fig. 14. Tests of the travel cost modeling on a real road network. Lanes are represented by lines, whose colors denote the various speed limits, the limit of yellow lanes is 90 km·h^{-1} and the limit of other lanes is 50 km·h^{-1}. Dashed curves indicate joint lanes that model non-signal-controlled turns, while dotted curves indicate joint lanes with signal controls. The stop and yield signs are separately indicated by red and yellow circles.

at an intersection. These delays were caused by dynamic traffic factors that are not considered in this study. Therefore, we ignored these values and derived the average time of multiple trips as the real travel time. The real travel times were found to be very close to the modeled costs, demonstrating the effectiveness of the cost-modeling approaches.

Lane-level route-planning tests were also performed on a lane-level digital map of a small urban area in Huizhou City, as shown in Fig. 15, where different color lines represent lanes with different speed limits. The routing results are shown by the red lines in Fig. 16. In general, the real map data are less complicated than the simulation data, as they have fewer intersections and lane nodes. It is clear that the proposed method has chosen the correct lanes to arrive at the destination. Table 4 compares the routing time costs of different routing methods in this urban area. The routing efficiency has been improved by 54.8%, compared with direct lane-level routing.

6. Conclusions and further research

This study aimed to develop a lane-level map-supported route-planning algorithm. First, the architecture of a seven-layer flexible and accurate digital map was proposed to explain how the digital map can enable navigation applications from road level to lane level. Next, refined modeling of travel costs (e.g., the cost of traversing a lane, making a lane change, or making a turn) was achieved by analyzing the vehicle movement on the lane-level road network. A graph for routing can be built by assigning the costs to the network. In order to accelerate the routing process, a hierarchical routing was implemented on both road-level and lane-level graphs in order to efficiently determine the optimal route. Simulation tests were performed to demonstrate the accuracy and effectiveness of the complete hierarchical routing algorithm. Tests were also performed on a real lane-level map to verify the cost modeling.

HD lane-level maps are believed to play a significant role in the development of autonomous driving vehicles. Many map data providers have already developed high-accuracy map data acquisition

Table 3
Test results of the travel cost modeling.

	Real travel time (s)						Modeled cost (s)
	1	2	3	4	5	Average	
Route 1	—	141.3	142.1	163.6	166.0	153.2	149.3
Route 2	151.5	154.4	—	150.1	156.3	153.1	150.1

"—" indicates anomalies caused by unexpected traffic delays that are not included in this study; thus, these values were excluded from the averages.
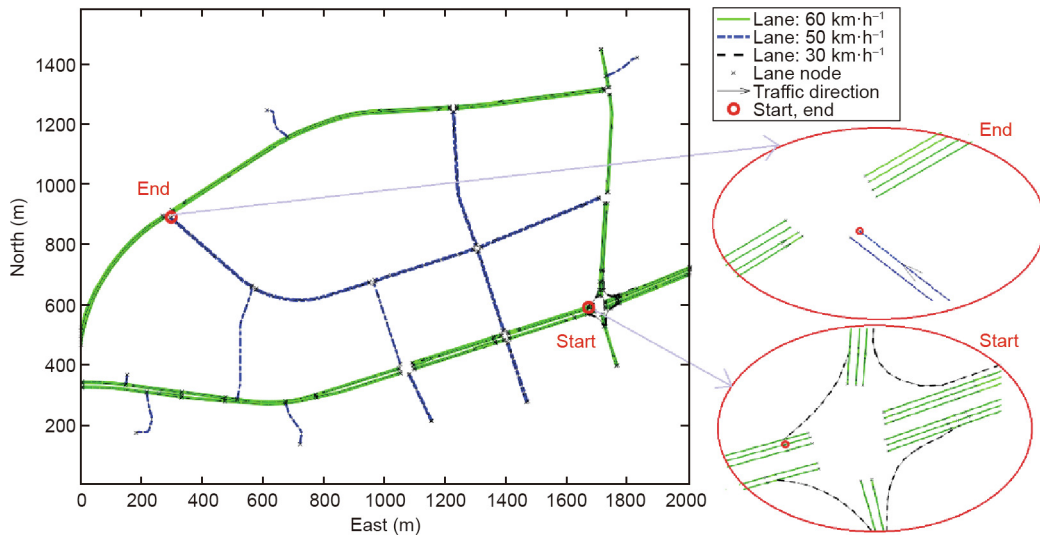

Fig. 15. Illustration of real lane-level map data of Huizhou City. The color of the lane indicates the speed limit of the lane.
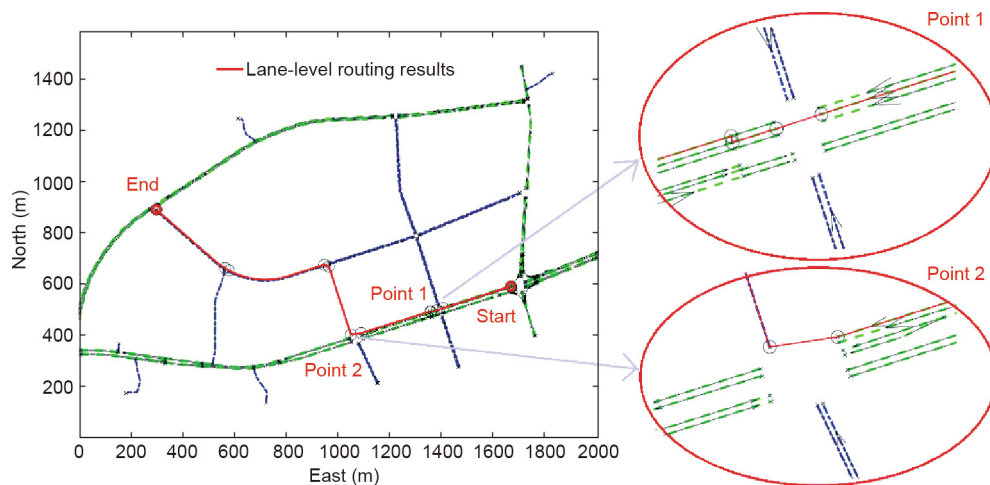

Fig. 16. Illustration of lane-level routing results over real map data of Huizhou City.

Table 4
Time costs of different routing methods in a real road test.

	Proposed hierarchical routing		Direct routing on lane-level map
	Road-level routing	Lane-level refinement	Lane-level routing
Node number	67	14	190
Time cost (s)	0.012	0.002	0.031
Total time cost (s)	0.014		0.031

equipment dedicated to the construction of HD maps. However, many problems remain to be addressed before wide-scale application is possible—such as the high cost of constructing and using an HD map. At present, a centimeter-level HD map can only be constructed using expert data, which is usually acquired by means of multi-layered LiDAR and is very difficult to update online. In future, AI technology will be employed to extract useful information from user data, which is likely to reduce the construction cost of HD maps. Meanwhile, the computation cost of using an HD map is equally problematic. The multi-layer map data structure and the hierarchical routing algorithms described in this study have the potential to reduce the computation cost.

To increase the utility of lane-level route planning in practical navigation applications, further research will expand the current work in two directions: ① More refined modeling of uncertain travel costs will be performed. In this study, we model the travel costs in a deterministic way; however, the uncertainties associated with

traffic conditions, weather, and other factors could also be included. The inclusion of uncertain information would lead to a more refined representation of costs using, for example, fuzzy numbers [26]. A balance should be obtained between routing complexity and efficiency. ② A further speed increase will be established for large road networks. This study addresses the lane-level route-planning problem on small or medium-sized road networks. Further speed-up techniques are required if the routing is performed on a large network. A simple strategy exists, which replaces bounded costs with their average values and applies advanced speed-up techniques such as contraction hierarchies [27]. The feasibility of this strategy will be verified in further studies. Furthermore, various existing speed-up techniques may be modified so that an optimal routing can be achieved directly on large lane-level road networks.

Acknowledgements

Compliance with ethics guidelines

Kun Jiang, Diange Yang, Chaoran Liu, Tao Zhang, and Zhongyang Xiao declare that they have no conflict of interest or financial conflicts to disclose.

References

[1] Skog I, Handel P. In-car positioning and navigation technologies—a survey. IEEE Trans Intell Transp Syst 2009;10(1):4–21.
[2] Lecce VD, Amato A. Route planning and user interface for an advanced intelligent transport system. Iet Intell Transp Syst 2011;5:149–58.
[3] Schultes D. Route planning in road networks. Saarbrücken: VDM Verlag; 2008.
[4] Ress C, Balzer D, Bracht A, Löwenau J. Adasis protocol for advanced in-vehicle applications. Meas Sci Rev 2008;11(5):154–9.
[5] Durekovic S, Smith N. Architectures of map-supported ADAS. In: Proceedings of the 2011 IEEE Intelligent Vehicles Symposium (IV); 2011 Jun 5–9; Baden-Baden, Germany. New York: IEEE; 2011. p. 207–11.
[6] Bohren J, Foote T, Keller J, Kushleyev A, Lee D, Stewart A, et al. Little Ben: the Ben Franklin Racing Team's entry in the 2007 DARPA Urban Challenge. J F Robot 2008;25(9):598–614.
[7] Wei J, Snider JM, Gu T, Dolan JM, Litkouhi B. A behavioral planning framework for autonomous driving. In: Proceedings of Intelligent Vehicle Symposium; 2014 Jul 8–11; Dearborn, MI, USA. New York: IEEE; 2014. p. 458–64.
[8] Zhang J, Singh S. LOAM: LiDAR odometry and mapping in real-time. In: Proceedings of the Robotics: Science and Systems Conference; 2014 Jul 12–16; Berkeley, CA, USA; 2014.
[9] Joshi A, James MR. Generation of accurate lane-level maps from coarse prior maps and LiDAR. IEEE Intell Transp Syst Mag 2015;7(1):19–29.
[10] Chen A, Ramanandan A, Farrell JA. High-precision lane-level road map building for vehicle navigation. In: Proceedings of the IEEE/ION Position, Location and Navigation Symposium; 2010 May 4–6; Indian Wells, CA, USA. New York: IEEE; 2010. p. 1035–42.
[11] Vu A, Ramanandan A, Chen A, Farrell JA, Barth M. Real-time computer vision/DGPS-aided inertial navigation system for lane-level vehicle navigation. IEEE Trans Intell Transp Syst 2012;13(2):899–913.
[12] Crash Avoidance Metrics Partnership. Enhanced digital mapping project final report. Report. Washington: United States Department of Transportation, Federal Highway Administration and National Highway Traffic and Safety Administration; 2004.
[13] Betaille D, Toledo-Moreo R. Creating enhanced maps for lane-level vehicle navigation. IEEE Trans Intell Transp Syst 2010;11:786–98.
[14] Schindler A, Maier G, Janda F. Generation of high precision digital maps using circular arc splines. In: Proceedings of the 2012 IEEE Intelligent Vehicles Symposium; 2012 Jun 3–7; Alcala de Henares, Spain. New York: IEEE; 2012. p. 245–51.
[15] Liu J, Cai B, Wang Y, Wang J. Generating enhanced intersection maps for lane level vehicle positioning based applications. Procedia Soc Behav Sci 2013;96:2395–403.
[16] Guo C, Meguro J, Kojima Y, Naito T. Automatic lane-level map generation for advanced driver assistance systems using low-cost sensors. In: Proceedings of the 2014 IEEE International Conference on Robotics and Automation (ICRA); 2014 May 31–Jun 7; Hong Kong, China. New York: IEEE; 2014. p. 75–82.
[17] Zeng W, Church RL. Finding shortest paths on real road networks: the case for A*. Int J Geogr Inf Sci 2009;23(4):531–43.
[18] Delling D, Sanders P, Schultes D, Wagner D. Engineering route planning algorithms. In: Lerner J, Wagner D, Zweig KA, editors. Algorithmics of large and complex networks. Berlin: Springer; 2009. p. 117–39.
[19] Zhu Q, Li Y. Hierarchical lane-oriented 3D road-network model. Int J Geogr Inf Sci 2008;22(5):479–505.
[20] OpenDRIVE® v1.4 format specification. Report. Bad Aibling: VIRES Simulationstechnologie GmbH; 2015 Nov. Report No.: VI2014.106.
[21] Martin Schleicher. NDS association: right in time—a standardized system is the key for the future autonomous drive. In: Proceedings of the Electronics in Vehicles 2015, 17th International Congress; 2015 Oct 14; Baden, Germany; 2015.
[22] Zhang T, Yang D, Li T, Li K, Lian X. An improved virtual intersection model for vehicle navigation at intersections. Transp Res Part C 2011;19(3):413–23.
[23] Volker L. Route planning in road networks with turn costs. Interpret A J Bible Theol 2008:1–30.
[24] Zhang T, Arrigoni S, Garozzo M, Yang D, Cheli F. A lane-level road network model with global continuity. Transp Res Part C Emerg Technol 2016;71:32–50.
[25] Fu L, Sun D, Rilett LR. Heuristic shortest path algorithms for transportation applications: state of the art. Comput Oper Res 2006;33:3324–43.
[26] Zhang X, Wang Q, Adamatzky A, Chan FTS, Mahadevan S, Deng Y. A Biologically inspired optimization algorithm for solving fuzzy shortest path problems with mixed fuzzy arc lengths. J Optim Theory Appl 2014;163:1049–56.
[27] Geisberger R, Sandersn P, Schultes D, Delling D. Contraction hierachies: faster and simpler hierachical routing in road networks. In: McGeoch CC, editor. WEA 2008: experimental algorithms. Heidelberg: Springer; 2008. p. 319–33.