



Research
Intelligent Transportation—Article

一种适用于自动驾驶汽车的多层地图模型和车道级轨迹规划

江昆, 杨殿阁*, 柳超然, 张涛, 肖中阳

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

ARTICLE INFO

Article history:

Received 5 March 2018
Revised 29 May 2018
Accepted 8 November 2018
Available online 12 March 2019

关键词

车道级地图
路径规划
清华地图模型
行车代价模型

摘要

越来越多的司机依赖于汽车或手机上的电子地图导航系统来选择最佳的驾驶路线, 以节省时间和提高安全性, 在不久的将来, 电子地图和导航系统有望在未来交通运输系统中发挥更大的作用。为了将现有的导航系统扩展到更多的应用领域, 如自动驾驶, 需要考虑在传统道路地图模型的基础上, 建立车道级地图模型和车道级轨迹规划。本研究针对传统电子地图内容不够丰富的局限性, 提出了一种全新的七层自动驾驶地图结构模型, 并将它命名为清华地图模型。考虑车辆换道、转向和直行等不同行车特点, 建立了车道级路段行车代价模型, 建立一种分层路径轨迹搜索方法, 能够在道路和车道网络中实现快速的轨迹规划, 能够很好地支持自动驾驶的车道级轨迹规划。通过在虚拟道路网络 and 实际道路网络上的测试, 充分验证了该地图模型和算法的灵活性和有效性。

© 2019 THE AUTHORS. Published by Elsevier LTD on behalf of Chinese Academy of Engineering and Higher Education Press Limited Company This is an open access article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>).

1. 引言

在给定起点和终点的情况下, 电子地图导航系统能够帮助车辆选择最优路线[1, 2]。电子地图导航系统在汽车上的首次安装应用可以追溯到1994年宝马的量产汽车上。从那时起, 导航系统在提高出行效率方面所带来的好处就被广泛接受。目前, 大多数车辆导航系统都是基于道路级电子地图, 规划路径结果主要是一系列的道路[3]。随着辅助驾驶系统和自动驾驶技术的进步, 电子地图的作用越来越重要。电子地图已成为智能汽车自动驾驶功能的基础数据支撑。电子地图在自动驾驶领域的重要应用之一便是智能导航功能, 即为自动驾驶车辆提供丰富而精确的导航信息。不同智能程度的智能导航

功能, 对地图的数据精度和内容丰富度的需求不同。目前文献里, 基于地图的智能导航功能可根据所依赖的地图类别大致分为两类:

基于道路级地图的智能导航功能。这些导航功能通常是辅助驾驶系统设计的, 所需的精度通常约为米级。例如, 电子地平线项目 (the electronic horizon program), 可以使用道路坡度来优化驾驶策略, 实现节能驾驶[4]。地图还可以辅助实现安全性能的提升, 参考文献[4, 5]中的曲线速度报警系统就是利用地图信息实现主动安全系统的另一个例子。在这些系统中, 所有的信息都是存储在道路上。

基于车道级地图的智能导航功能。由于可以从电子地图中获取车道级的环境细节信息, 这些导航系统可以

* Corresponding author.

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

提供更精确的驾驶导航信息，以便增强车辆的智能程度。2007年，美国DARPA（Defense Advanced Research Projects Agency）城市挑战赛就展示了车道级地图支持的驾驶系统，在Bohren等[6]的工作中，一个车道级的道路网络定义文件被用来提供驾驶环境的先验信息。

本项研究的目标是基于车道级地图开发面向自动驾驶的路径规划算法，为自动驾驶导航功能的开发奠定技术基础。传统的道路级导航系统的目标用户是人类驾驶员，在这种场景下人类驾驶员负责实时选择驾驶轨迹。然而新型导航系统，即车道级导航系统的目标用户是自动驾驶车辆，其必须提供更详细的路径导航信息以辅助车辆完成自动驾驶任务。两种导航系统的最大差别在于前者是基于道路级地图，而后者是基于车道级地图，具体差别如图1所示。由于道路级地图的数据精度不高，道路级导航生成的道路路径是针对驾驶员设计的驾驶任务提示，而不是自动驾驶车辆能够遵循的具体轨迹。道路级导航对于人类驾驶员来说足够准确，然而对于自动驾驶车辆而言，所提供的指引信息过于模糊。为了弥补这一缺点，基于道路级导航的车辆必须具备强大的实时环境感知和决策系统，以便实时规划出具体的行驶路径，这大大增加了车载计算单元的负担。相比之下，车道级导航则需要提供更清晰细致的引导信息，即在无障碍物的前提下，提供一条可供自动驾驶车辆跟随的实际参考轨迹。车道级导航与道路级导航的关键区别在于前者能够在不借助环境感知系统的情况下提供精确的轨迹作为控制系统的输入。虽然车道级导航系统不能替代实时感知和决策系统，但可以极大地减轻计算负担，降低系统故障风险。卡内基梅隆大学的研究人员也在他们的行为规划框架[7]中提到了车道级导航的重要性，即无

障碍物情况下的车道级参考路径可以用于提升车辆的控制性能。

车道级导航系统由车道级地图、高精度定位和车道级路径规划三大部分组成。其中高精度定位技术的研究相对成熟。利用实时动态差分全球定位系统（RTK-DGPS），全球导航卫星系统在开放区域的定位精度可达到厘米级。在信号丢失的情况下，可以使用基于相机或基于激光雷达的特征匹配技术来获得准确的位置。在之前的研究中，我们使用惯性测量单元（IMU）作为全球定位系统（GPS）的补充，以提高定位的精度和鲁棒性。具体来说，我们基于无迹卡尔曼滤波算法融合车辆动力学模型、GPS和IMU的信息，如公式（1）所示：

生产模型：

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

测量模型：

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

式中， X_{gps} 、 Y_{gps} 、 v_{gps} 、 v_{wheel} 、 ϕ_{gps} 和 $\dot{\phi}_{\text{gyro}}$ 为GPS接收机和IMU的测量值； X 、 Y 为车辆位置状态； v_x 为车辆沿x轴的速度； ϕ 为车辆朝向与正北方向之间的顺时针角度；

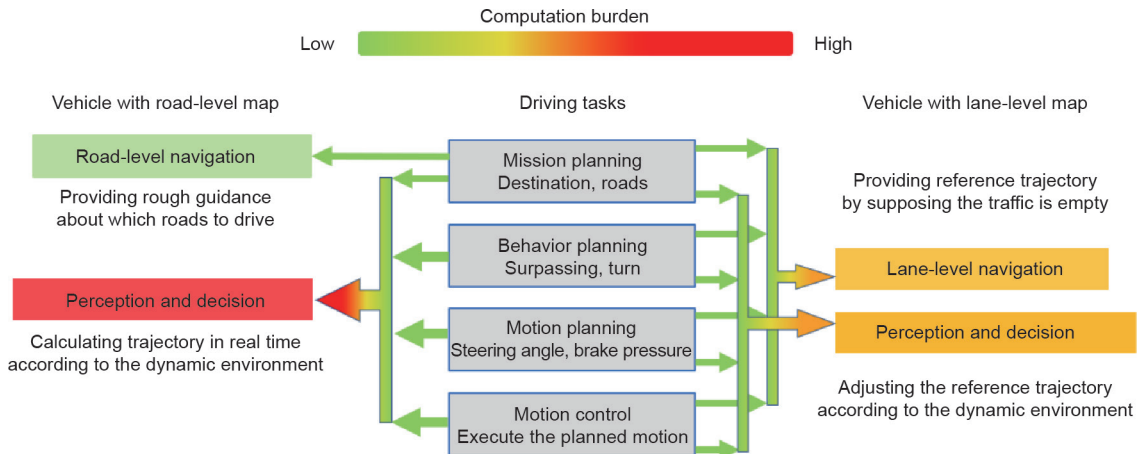


图1. 道路级导航和车道级导航的自动驾驶方案对比。

$\dot{\phi}$ 为角速度； g 为重力加速度； θ 为道路俯仰角； δ 为噪声向量。

本文的研究更加关注的是另外两个部分的研究：车道级地图和车道级路径规划算法。

车道级地图、高清地图（HD map）或高精度地图等新型地图概念目前仍然没有可被广泛接受且确定的定义，但这些概念都指出新型地图应当比传统地图更精确、更适合自动驾驶。在本研究中，我们使用车道级地图来代表新型地图，“车道级”表示地图数据精度优于0.5 m并且能够区分不同的车道。目前车道级地图正成为一个热门的研究领域。随着高精度定位传感器等先进数据采集设备的发展，建立适用于车辆导航定位的增强型地图成为可能[8-10]。随着多传感器融合技术的发展，许多开发人员都能够实现车道级精度的定位[11]。Betaille等[12, 13]通过使用一系列形状点来表示几种车道类型，初步尝试建立车道级导航地图。Schindler等[14]使用圆弧样条生成车道级地图，其中包括道路标记、地标和其他附加属性。Liu等[15]和Guo等[16]设计了一种基于传感器和OSM（OpenStreetMap）的车道级地图生成方法。目前对车道级地图的研究主要集中在对车道进行高精度的几何表示。然而面向自动驾驶的车道级地图的另一大难点是提高地图表达的灵活性，当前研究相对较少。地图表达的灵活性是指根据实际需求灵活地提供所需要的信息。车道级地图包含大量详细的环境信息，如所有车道的坡度、曲率，还有所有车道点的坐标等。然而并不是每一次调用地图的时候都需要调用所有信息，因为大部分信息对于特定导航任务是不必要的。为了提高效率，车道级地图应当能够更加灵活，只在需要时提供详细的车道信息。为了提高电子地图的灵活性，本文提出了一种多层次电子地图模型。

车道级路径规划比道路级路径规划更具挑战性，主要表现在两方面：一是行车代价模型的构建；二是路径规划效率的优化。传统的道路网络结构通常被简化为图论中的最短路径问题，经典求解方法有Dijkstra算法、A*算法等[17]。然而在实际导航应用中，这些算法需要进行优化修改以便考虑实际道路网络的特定属性[3]。目前文献中已有大量的道路级路径规划算法，参考文献[18]给出了相关的综述。然而当涉及车道级路径规划时，导航任务就不仅仅是寻找最短路径，而应该考虑更多的因素，比如车辆换道的位置。为了确保导航系统找到最优路径，且避免给出不可行的路径，行车代价模型

的构建就显得尤为重要。路径规划的效率是另一个应当被考虑的关键因素。在参考文献[19]中提出了一种基于分层三维路网模型实现高效的车道级路径规划的可行策略。首先利用 k -最短路径算法在道路中心线网络选择三条候选路径。然后，在三条候选路径中基于道路级行车代价模型计算出最优道路级路径，最后在此道路级路径对应的车道级网络上确定最终路径。这一方案虽然简单高效，但也存在无法找到最优车道级路径的风险。例如，在第一个寻路阶段所选择的三条候选路径规划可能无法覆盖最优路径。本文的贡献在于，利用车道级地图的丰富信息，设计了考虑变道和路口转向影响以及车道级驾驶行为规则限制的行车代价模型，并基于该代价模型提出了一种面向自动驾驶的车道级路径规划算法。

本文的组织结构如下：第2节将介绍本文所提出的地图模型整体结构，并解释道路级地图和车道级地图之间的关系。第3节详细讨论车道级行车代价模型的构建方法，进而在第4节介绍所提出的车道级路径规划算法。然后在模拟的道路网络和真实的车道级电子地图上分别进行了实验验证，实验结果见第5节。最后，我们在第6节中给出结论和进一步研究的方向。

2. 精度自适应的地图模型

2.1. 七层地图结构

由于激光雷达等高精度传感器的发展，电子地图的精度可以达到厘米级，电子地图的制作难度也大大降低，推动了地图数据的爆炸式增长。一些新的地图标准被用于存储车道级地图数据，如OpenDRIVE [20]和NDS [21]。然而，这些地图数据标准并没有针对自动驾驶对地图数据的调用方式进行设计，难以在自动驾驶开发中直接使用。因此，面向自动驾驶且能够充分发挥高精度地图数据优势的地图模型就变得非常重要。值得注意的是，地图数据精度越高，并不一定意味着能够更好地支撑智能导航功能。因为不同的导航应用场景，所需的地图精度是不同的。为了使电子地图能够有效地支持各种类型的导航应用，我们提出了一个七层地图模型，简称清华自动驾驶地图模型，如图2所示。每一层地图所包含的数据类型不同，所支撑的导航功能也不同。

第一层：道路级路网层。这一层地图数据主要用于道路级路径规划，提供道路级的全局导航信息[17]。这一层主要包含传统的静态电子地图数据。

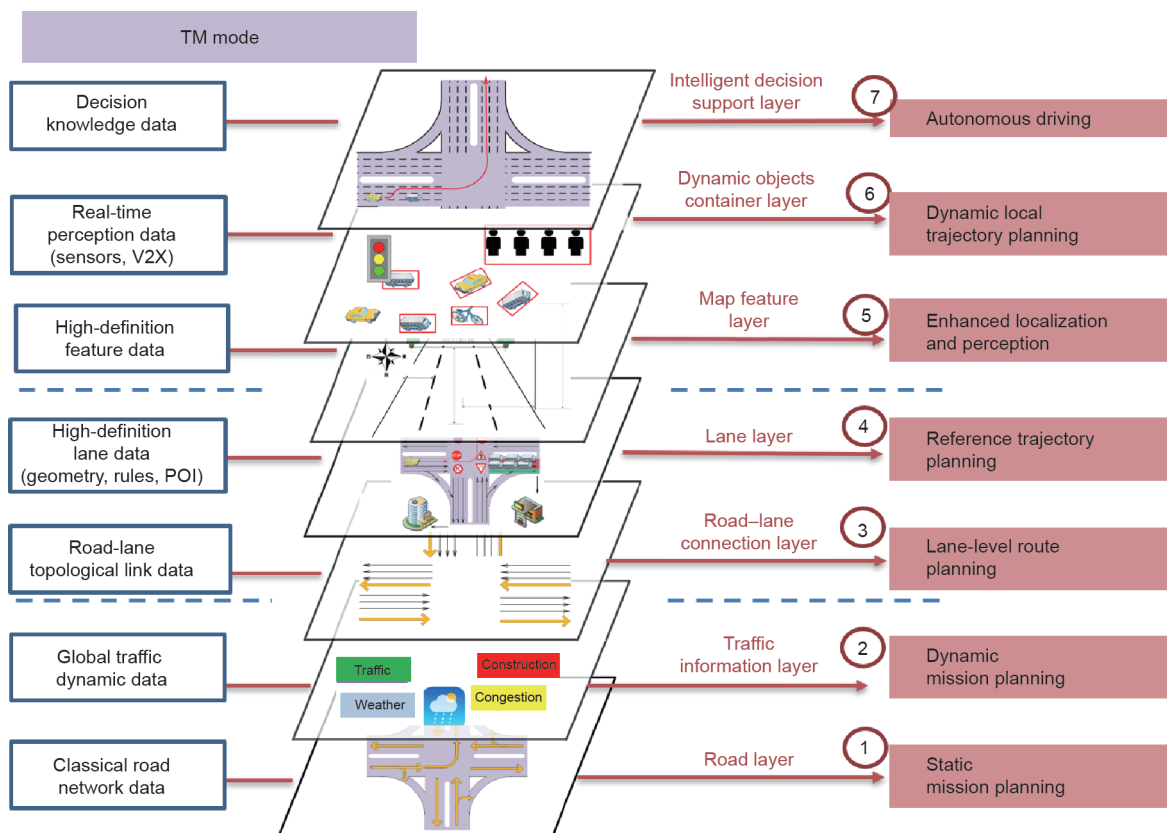


图2. 一种用于自动驾驶的七层自适应精度地图体系结构。

第二层：交通信息层。这一层地图数据主要用于动态道路级路径规划。这一层地图数据主要包括道路级动态交通数据，如交通堵塞和道路施工等事件，可用于避免拥堵的动态全局路径规划。

第三层：道路-车道连接层。这一层主要用于车道级路径规划，提供了道路级路网和车道级路网之间的拓扑连接。它能够将道路级路网拓扑映射到车道级路网中。由于不包含每个车道的详细信息，这一层数据体量相对较小，方便进行快速的候选车道级路径搜索。

第四层：车道级路网层。这一层地图数据是为车道级导航而设计的，提供了高精度的车道级几何结构、车道级交通规则、道路标志等车道相关的信息。结合第三层和第四层可以得到综合考虑道路坡度、曲率和交通规则的车道级最优路径。

第五层：地图特征信息层。这一层地图数据用于辅助自动驾驶车辆的环境感知，可以用于开发具有自主定位能力的车道级导航。这一层地图存储了高精度的特征数据，以便实现基于地图匹配的定位或感知算法。

第六层：动态感知容器层。这一层地图数据用于局部动态轨迹规划。其存储的数据包括网联车辆或基础设施共享的障碍物动态信息。同时这一层地图也可以看做

动态感知信息的容器，提供融合多车传感器信息的标准化接口，如激光雷达、摄像机等。基于这一层的信息，局部轨迹规划可以考虑动态情况以便保证安全。

第七层：智能决策支持层。这一层是为自动驾驶决策过程设计的，它能够提供驾驶决策知识数据库。这一层存储的数据来自于对驾驶员的驾驶决策行为的大量分析和学习。

本文所提出的七层地图模型的重要特点是具备精度自适应的地图数据提取能力。对于道路级导航，只需使用第一层和第二层的地图数据。对于使用差分卫星定位系统的车道级导航，我们需要在地图中加入第三层和第四层。在GPS失效的情况下，第五层可以帮助车辆实现自主定位。对于多车协同自动驾驶场景，需要添加第六层来获取网联车辆共享的信息。对于高级别自动驾驶场景，第七层数据则提供了决策支持信息。

在所提出的清华自动驾驶地图模型中，地图数据的使用更加灵活，只有在导航任务需要时才会调用相应的地图信息。这里需要说明的是第五层、第六层和第七层地图是为基于地图的定位、感知和驾驶决策功能而设计的，不在本文研究的范围之内。本文关注第一层到第四层地图数据的构建和调用方式。在接下来的章节中，我

们将介绍由第一层到第四层地图数据所构成的车道级地图模型。

2.2. 车道级地图模型

车道级路网应包含车道级路段和交叉口的几何及拓扑细节，如图2中的第四层所示。相关研究已经提出了多种表征车道细节信息的车道级地图模型的方法。车道几何信息通常用clothoids[13]、arc样条[14]或CHS (cubic Hermite splines) 曲线[10]来表示，而道路交叉口则用拓扑结构来描述[15, 19, 22]。为了支持高效的车道级路径规划算法，我们提出了一种由清华自动驾驶地图模型的前四层所组成的车道级地图模型，如图3所示。基于每一层地图的概念，本文进一步设计了各层地图数据的具体数学表达模型。

2.2.1. 道路级路网层

在不失一般性的前提下，我们考虑路网由一组道路 $\{r_j\}_{j=1}^a$ 和一组交叉口 $\{c_i\}_{i=1}^b$ 组成，其中 a 和 b 分别表示道路的个数和交叉口的个数。道路层通常可以表示为

$$W = \left(\{r_j\}_{j=1}^a, \{c_i\}_{i=1}^b \right) \quad (2)$$

$$c_i = (P_{c,i}, E_{c,i}, T_{c,i})$$

式中， W 代表道路层； $P_{c,i}$ 表示进入第 i 个交叉口的道路级节点集合； $E_{c,i}$ 表示离开该交叉口的道路级节点集合； $T_{c,i}$ 表示两个节点之间的道路级交通矩阵。

$$T_i = \begin{matrix} & P_{c,j,1} & \dots & P_{c,j,n} \\ E_{c,j,1} & \begin{bmatrix} t_{i,1,1} & \dots & t_{i,1,n} \\ \vdots & \ddots & \vdots \\ E_{c,j,m} & \begin{bmatrix} t_{i,1,m} & \dots & t_{i,n,m} \end{bmatrix} \end{bmatrix} \end{matrix} \quad (3)$$

式中， m 和 n 分别为离开交叉口和进入交叉口的节点数； $t_{i,n,m}$ 为车辆能否从驶入节点 $P_{c,j,n}$ 到驶出节点 $E_{c,j,m}$ ，并且 $t_{i,n,m} = (f_i, m_i)$ ，其中， f_i 表示能否从 $P_{c,j,n}$ 到达 $E_{c,j,m}$ ， m_i 表示行驶方案，如左转弯、右转弯、直行或者转U形弯。

道路级路网表示如下：

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

式中， P_r 为进入该交叉口的道路级节点集合； E_r 为离开该交叉口的道路级节点集合； Q_r 包括道路类别 k_r 和道路长度 l_r 。

2.2.2. 交通信息层

为了避免交通阻塞、提高全局交通效率，路径规划算法必须考虑交通信息。通过对流动车辆数据的分析，可以得到全局的交通动态。在本研究中，交通状况由记录的平均速度向量表示：

$$V_r = [V_r(t_0), \dots, V_r(t_i), \dots, V_r(t_k)] \quad (5)$$

式中， $V_r(t_i)$ 为时间间隔 $[t_i, t_i + \Delta t]$ 内的平均速度； t_k 为最

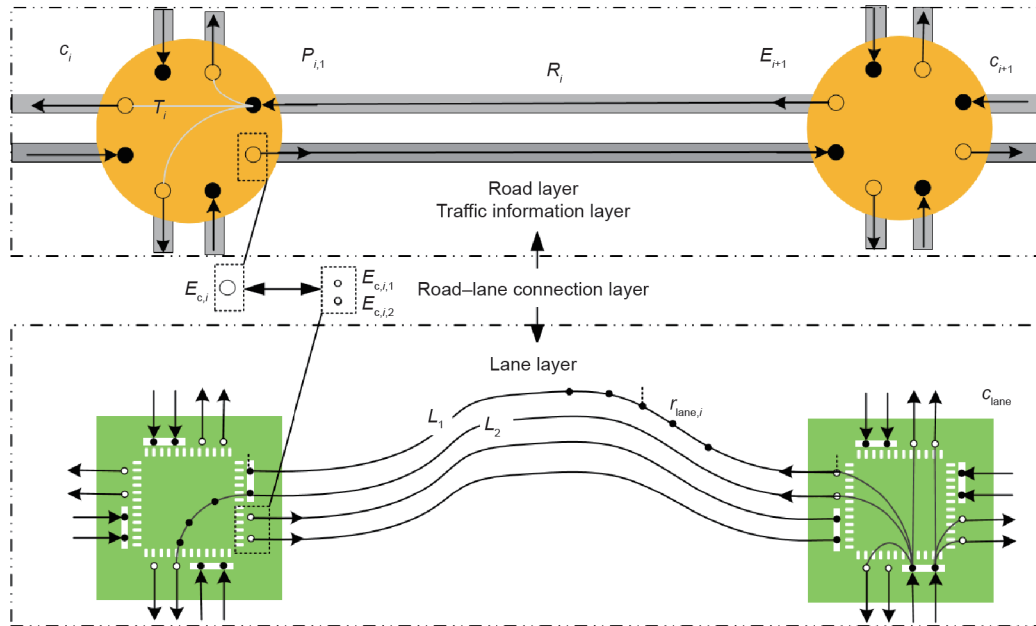


图3. 用于车道级导航的四层车道级地图模型。

新记录速度的时刻。

平均速度的历史记录可以作为经验数据来估计时间成本，同时也可以用来预测未来的行驶速度。

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

式中， $\hat{V}_r(t_{k+i})$ 为时间间隔 $[t_k + i\Delta t, t_k + (i+1)\Delta t]$ 内的预测速度。

2.2.3. 道路 - 车道连接层

这一层的作用已经在2.1节中解释过，是道路级信息与车道级信息之间的连接层。这一层是整个地图结构灵活性和精确性的关键，因为它提供了加载高精度数据的通道，并保留了传统地图的优势。因此连接层可以表示为

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

式中， $P_{\text{lane},n}$ 、 $E_{\text{lane},m}$ 分别为车道级的驶入节点和驶出节点。每个车道级节点包含节点的位置和节点的交通方向，记为 (u_N, v_N) 。

2.2.4. 车道级细节层

车道级细节层包含更精确和完整的信息。

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

式中， W_{lane} 表示车道层； a_{lane} 和 b_{lane} 分别表示车道数和车道级交叉数； $r_{\text{lane},j}$ 表示属于某一道路实体的交通车道，其中车道定义如下：

$$r_{\text{lane},j} = (S_{\text{lane}}, S, C_s, C_e, Q_{\text{lane}}) \quad (9)$$

式中， S_{lane} 为车道形状的参数集，由CHS定义。车道曲线可用参数函数 $f_{\text{CHS}}(u, S_{\text{lane}})$ 表示，其中， u 为车辆的位置。 s 表示属于同一道路的车道集合中车道的横向序号。在序列确定之后，最右边的车道被标记为1。根据车道的交通方向， c_s 和 c_e 分别表示起点和终点的交叉口。 Q_{lane} 表示车道属性，包括总长度 L 和限速值 V_{limit} ，以及其他几何和交通属性，如交通标志 $B(u)$ ， $B(u) = 1$ 表示一个停止标志位于 $f_{\text{CHS}}(u, S_{\text{lane}})$ 。

车道级交叉路口 $c_{\text{lane},i}$ 的定义与道路级交叉口相似：

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

式中， $T_{\text{lane},i}$ 为车道级交通矩阵，表示交叉路口处驶入车

道 λ_i 和驶出车道 λ_j 的联合车道的几何和拓扑特征，联合车道定义如下：

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

式中， S_t 也是由CHS定义的，表示车辆从车道 λ_i 到车道 λ_j 的可能路线的参数集合； λ_i 和 λ_j 分别表示驶入车道和驶出车道； Q_t 表示联合车道的属性，如车道长度 L 以及交通信号 B_s ， $B_s = 1$ 表示联合车道的交通正在受到交通信号的控制； f_m 表示车道 λ_i 和 λ_j 能否实际上被连接在一起。值得注意的是，十字路口的车道级交通约束本质上是由交通矩阵所描述的。

3. 车道级行车代价模型

为了在车道级地图上进行最优路径的搜索，本研究建立了车道级的行车代价模型。一般而言，行车代价可以分为三种类型：静态行车代价、动态行车代价和统计行车代价[23]。静态行车代价是路网的静态属性，而动态和统计行车代价则受其他交通参与者的影响。本研究以静态行车代价为研究重点，研究基于道路级静态路网数据获得精确的代价模型。路径规划需要考虑多方面的代价，如行程距离、时间成本、油耗等各种混合指标。本研究主要关注时间成本。下面将介绍具体的代价模型构建方法。

3.1. 换道的通行时间成本

根据车道属性可以很容易地估算出单车道行驶时间：

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

式中， L 和 V_{limit} 已在式(9)中定义。然而在大多数情况下，司机可能不会沿单车道行驶，而是会多次改变车道。注意这里不考虑动态因素。我们考虑了两种换道的常见场景。

3.1.1. 具有不同限速的车道

在这种情况下，假设行驶在限速较低的车道上的车辆为了以较高的速度行驶，将会尽快改变车道。我们假设换道是在车道开始时进行的，因此在两个起始点之间添加一个过渡线，如图4(a)所示。

不同限速的车道间换道时，不仅要考虑额外的行驶距离所产生的延时，同时还要考虑加速过程所产生的延时。时间成本可由下式得到：

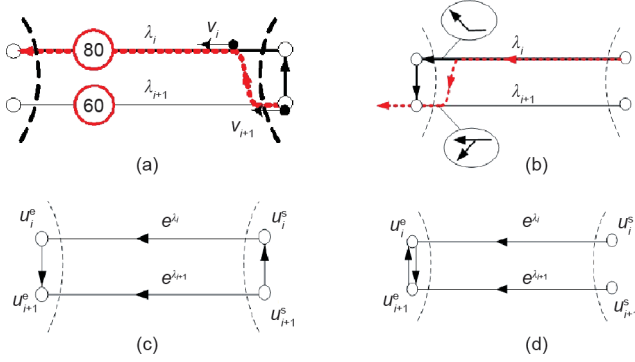


图4. 变道成本建模。(a) 由于不同限速 ($\text{km}\cdot\text{h}^{-1}$) 而换道; (b) 由于不同的交通限制而换道; (c) 不同限速车道之间的过渡线, 如图中所示; (d) 不同交通限制的车道之间的过渡线。

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

式中, a_v 为车辆加速度; Δs 为变道引起的额外长度, 可以用变道宽度近似。

3.1.2. 具有不同交通限制的车道

行驶在高速车道上的车辆, 也会因为其他原因变换到低速车道, 如为了遵循转弯车道的交通限制。如图4 (b) 所示, 这种变道一般发生在前方交叉口附近, 因此可以用两条车道末端节点之间的过渡线来描述。额外时间成本计算如下:

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

式中, V_{traffic} 表示由于交通限制导致的限速, 而不是路标指示的限速值。

图4 (c) 和 (d) 表示了两种可能的换道的过渡线。如果一条道路包含两条以上的车道, 则应生成相邻车道之间的过渡线。

3.2. 交叉路口的转弯时间成本

交叉路口转弯时间成本也是路径总代价中的重要部分, 尤其是在城市地区。在以往的研究[23]中, 已经考虑了这个问题; 然而我们可以使用交叉路口处的车道级详细数据对它加以改进。转弯成本不仅要考虑交叉口的行驶距离, 还应考虑到在交叉路口处的加减速行为。如图5所示, 车辆通过交叉口的运动可以分为三个部分。

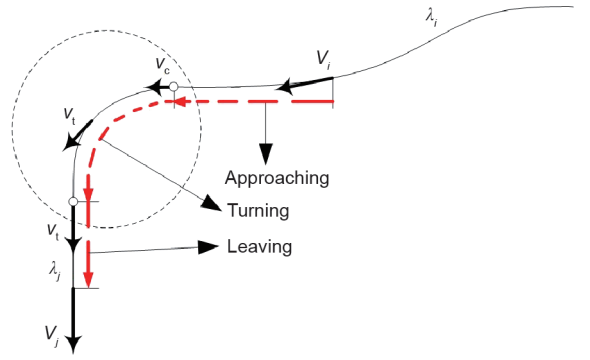


图5. 转弯成本建模。 v_c : 驶入车道末端处的车速; v_t : 转弯速度。

3.2.1. 接近路口阶段

一般来说, 车辆在接近十字路口时减速。减速引起的时间延迟可以通过以下公式计算得到:

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

式中, v_c 为驶入车道末端的车速, 主要由该位置的交通特征所决定。如果该处有“停车”交通标识, 则 v_c 为零。在大多数情况下 v_c 等于转弯速度 v_t , 这将在下面讨论。

3.2.2. 在路口转弯阶段

交叉口转弯时间可分为等待信号时间和转弯行为时间两部分。等待信号时间是一个与道路交通管理更为相关的量。本研究的重点是驾驶行为的建模, 因此我们假设在等待时间 t_{wait} 后, 车辆开始以 v_c 加速到 v_t , 然后沿着转弯车道行驶。行驶时间计算如下:

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

式中, L_T 表示转弯车道的长度。转弯速度受到转弯类型和交叉口构型等多种因素的影响。因此, 我们定义了一个考虑交叉口属性的转弯速度模型:

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

式中, $v_b = \min(V_i, V_j)$, 代表最低行驶速度; κ 代表转弯车道的平均曲率, 可以通过转弯车道曲线计算得到[24]; r_{min} 为最小转弯半径, 如对一辆典型的客车而言最小转弯半径为6 m。

3.2.3. 驶离路口阶段

当离开交叉路口时，车辆加速以达到驶出车道的速度。时间延迟可以通过如下公式计算得到：

$$f_{\text{cost}}(c_{\text{leave}}) = \frac{(V_j - v_t)^2}{2a_v V_j} \quad (18)$$

最后，交叉口的总行车代价计算如下：

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

4. 车道级路径规划

在代价模型确定之后，本节将介绍车道级路网的路径规划方法。车道级路径规划的优势在于可以充分考虑每个车道的详细属性，提供一个符合车辆驾驶行为约束的最优路径，如图6所示。

首先，我们在每个车道的起点和终点添加节点，从而整个路网可以概括为一个有向图：

$$G = (U, E) \quad (20)$$

式中， U 表示节点的集合 $\{u_i\}$ ； E 表示边的集合 $\{e_i\}$ 。根据路网拓扑结构，一条边可以从连接了起点 $u_{\text{start}}(e^{\text{lane}_i})$ （简记为 u_i^s ）和终点 $u_{\text{end}}(e^{\text{lane}_i})$ （简记为 u_i^e ）的驶入车道 lane_i 得出，也可以从连接了驶入车道 lane_i 的终点 u_i^e 和驶出车道 lane_j 的起点 u_j^s 的连接车道 t_{ij} 得出。

接下来，定义权重函数 $f_{\text{cost}}(e)$ 用于为每条边指定一个权重。权重表示以指定的度量（如距离或行驶时间）描述换道或转弯的成本。在本文中，我们关注的是时间成本这一最自然的度量。车道和转弯成本的详细模型已在第3节中描述。路径规划的目标是找到最优路线 $P =$

$\{e_1, \dots, e_p\}$ 满足以下约束：

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

式中， $e_i \in E$ 为搜索区域内的所有车道； $c(P)$ 为沿路径 P 的总成本。

如前所述，路径规划问题可以使用经典算法解决，如Dijkstra算法或A*算法。然而由于大型车道级路网过于复杂，将这些算法直接应用于长距离车道级路径搜索是不现实的。各种启发式规则和分层结构被应用于大型路网的路径规划[25]。在本研究中，我们也提出了一种高效的分层车道级路径规划方法。我们所提出的分层车道级路径规划方法是道路级规划和车道级规划进行融合。通过这样的设计，可以将文献中所积累的大量道路级路径规划重新利用起来，为车道级路径规划提供了技术基础支撑。车道级路径规划问题可视为从预定起点到预定终点的车道级最优路径搜索过程。本文所提出的分层规划步骤为，首先搜索出最优道路级路径上的节点，如图7（a）所示。然后在道路级节点所对应的车道级路网上计算车道级行车代价，最后搜索出代价最顶的车道级路径以及构成路径的车道级节点，如图7（b）所示。

4.1. 道路级路径规划

假设节点 O 和 D 分别是预定的起点和目的地，则在道路层级上的路径规划可以被建模为对道路级交叉序列的搜索。

$$N_r = (O, N_1, N_2, \dots, N_i, \dots, N_k, D) \quad (22)$$

式中， N_i 为道路层级交叉节点的节点数。

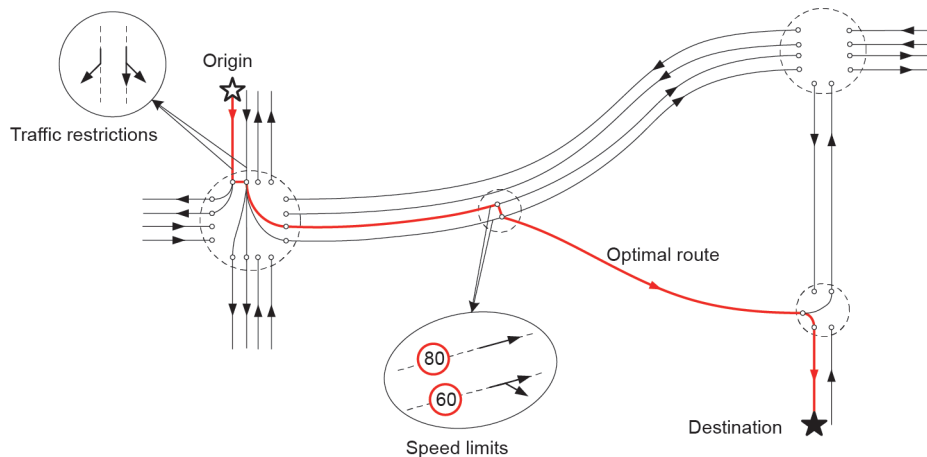


图6. 车道级路网的最优路径（用红线表示）。在确定最优路径的过程中应当考虑交通限制及速度限制（ $\text{km} \cdot \text{h}^{-1}$ ）。

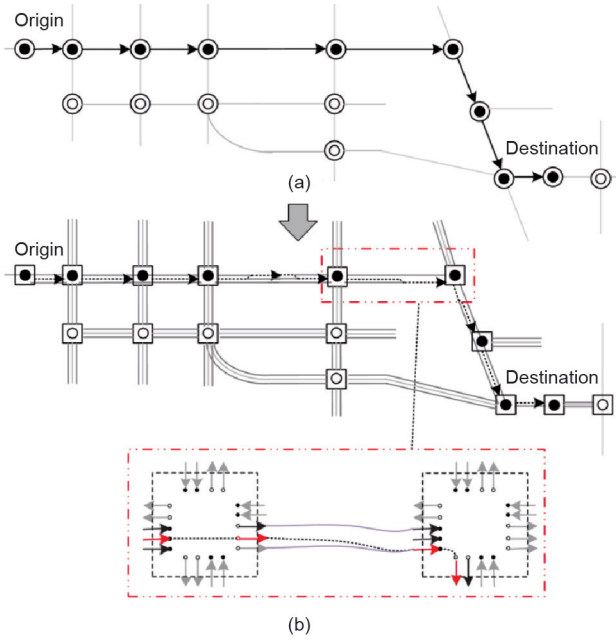


图7. 基于多层地图的分层路径规划。(a) 道路级路径规划；(b) 车道级轨迹规划。

在道路级的导航步骤中，本文使用传统的A*算法[18]。

4.2. 车道级地图路径规划

根据获得的道路节点序列，将粗略的道路级路径细化为车道级路径，具体可分为两个步骤：①根据道路级规划结果选择车道级节点；②根据车道级路径的代价选择所连接的车道级节点。

4.2.1. 选择车道级路径节点

由第2节中介绍的车道级地图模型可以知道，每个道路交叉口都包含了进入和离开该区域的车道级节点。道路-车道连接层中的地图数据结构可以支持从道路级节点映射到车道级节点。在获得道路级路径规划的结果后，如 N_1, N_2, \dots, N_k ，获取最优车道级节点的最简单方法是遍历所有关联的车道级节点；但是，这种遍历会计算大量不相关的车道级节点，从而降低路径搜索的效率。

为了减少不相关的车道级节点对路径搜索效率的影响，本文提出了一种搜索候选车道级节点的方法，如图8所示。

首先，仅选择可以连接两个连续道路级节点的车道级节点。对于道路级交叉点节点 N_i ，在进入交叉点时的车道级节点（进入节点）应连接到上一个道路级交叉点节点 N_{i-1} 。同样，最后一个车道级节点（出口节点）应连接到下一个道路级交叉点节点 N_{i+1} 。进入和退出该交

叉节点的车道级节点 N_i 表示为 $N_{i,i-1}$ 和 $N_{i,i+1}$ ，可以由以下等式计算得出：

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

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

式中， P_c 为道路级节点的入节点； E_c 为道路级交叉节点的出节点，两者均由道路级路径规划结果确定。

然后，考虑交通规则或道路条件的限制进一步筛选候选的车道级节点。如果需要在道路级节点中完成左转弯的动作，且该道路级节点包括三个车道级进入节点，如在交叉路口中，根据交通规则，最左侧的车道级节点是唯一可行的车道级节点。为了在车道级路径规划考虑类似情况，本文使用交通矩阵来消除不可行的节点。存储在交通矩阵中的数据由公式（10）和公式（11）计算和处理。对于给定的行驶任务，如从 N_{i-1} 到 N_{i+1} ，道路级节点 N_i 中的交通矩阵可以进一步简化为 $t(N_{i-1}, N_{i+1})$ 。

$$t(N_{i-1}, N_{i+1}) = t(N_{i-1}, N_{i+1}) \cup t(N_{i-1,n}, N_{i+1,m});$$

$$\text{if } f_m(N_{i-1,n}, N_{i+1,m}) = 1 \quad (25)$$

交通矩阵中只包含可行的进入和出口节点。通过引入交通矩阵，本文减少了要搜索的节点数量。假设在道路级节点 N_i 中，可行的进入和退出通道的数量分别为 n 和 m ；那么交通矩阵可以表示为下面的等式。

$$T(N_{i,i-1}, N_{i,i+1}) =$$

$$N_{i,i-1,1} \quad \dots \quad N_{i,i-1,n}$$

$$N_{i,i+1,1} \begin{bmatrix} t(N_{i,i-1,1}, N_{i,i+1,1}) & \dots & t(N_{i,i-1,n}, N_{i,i+1,1}) \\ \vdots & \ddots & \vdots \\ t(N_{i,i-1,1}, N_{i,i+1,m}) & \dots & t(N_{i,i-1,n}, N_{i,i+1,m}) \end{bmatrix} \quad (26)$$

算法1展示了道路级节点的搜索过程。

算法1 选择局域网级路由节点

```

1  $N_r$  is known, set  $N_p = \emptyset$ 
2 for  $i = 2, \dots, k-1$ 
3 set  $N_{i-1} = \emptyset, N_{i+1} = \emptyset$ 
4  $N_i, N_{i-1} \rightarrow P_c \rightarrow N_{i-1}, N_i, N_{i+1} \rightarrow E_c \rightarrow N_{i+1}$ 
5 set  $t(N_{i-1}, N_{i+1}) = \emptyset$ 
6 for  $x = 1, \dots, n_{i-1}$ 
7 for  $y = 1, \dots, n_{i+1}$ 
8 if
9  $t(N_{i-1}, N_{i+1}) = t(N_{i-1}, N_{i+1}) \cup t(N_{i-1,x}, N_{i+1,y})$ 
10 else  $N_{i-1} = N_{i-1} / N_{i-1,x}, N_{i+1} = N_{i+1} / N_{i+1,y}$ 
11 (find the lane-level node which has no link in this road node then delete it)

```

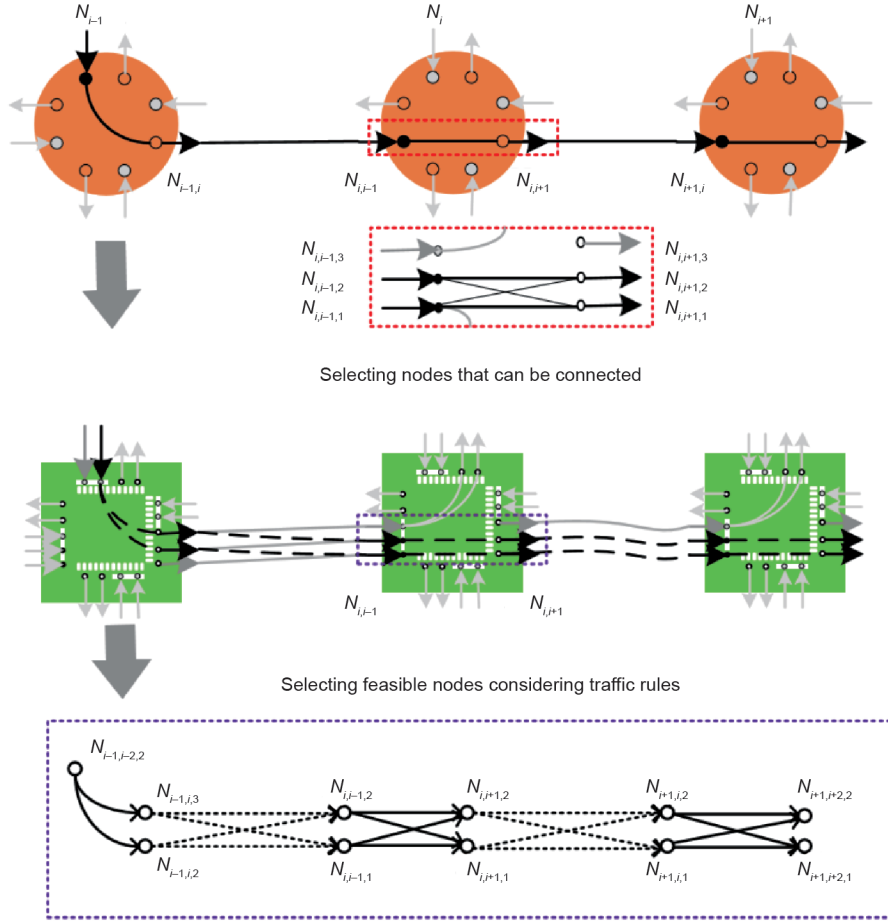


图8. 车道级节点的搜索方法。

```

12  end if;
13  end for;
14  end for;
15   $N_p = N_p \cup [N_{i,j-1}, N_{i,j+1}, t(N_{i,j-1}, N_{i,j+1})]$ 
16  end for;
17  return  $N_p$ 

```

N_p : 车道级上的路由规划结果，它是一组车道级节点。

4.2.2. 在候选车道级节点上进行路径规划

在获得候选车道级节点之后，就可以进行车道级路径规划。道路级和车道级路径规划算法之间的最主要差别在于是否包含行车代价模型的构建方法以及路径搜索的方向。行车代价模型的差别在第3节已经解释。而路径搜索方向的差别表现在道路级规划大多是双向搜索，而本文的分层规划架构下，车道级有向图是单向的。

借鉴传统的寻路算法，本文设计了一个逐步迭代的寻路算法以便获取到最优的车道级节点集，算法过程如图9所示。每一步迭代时，首先计算从起点到当前节

点的成本；然后存储可以实现最小成本的车道级节点，表示为 $P_f(N_{i,j-1,m})$ 。由于本文采用的是单向图，所以没有必要计算从当前节点到目的地的成本，这提升了算法的计算效率。算法2中提供了有关车道级轨迹规划的详细步骤。

算法2 车道级路径规划

```

1  set  $i = 1, j = k$  (number of road-level nodes in addition to  $O$  and  $D$ ),  $N_{0,1} = (O)$ ,  $N_{k+1,k} = (D)$ ,  $N_i = \emptyset$ 
2  while  $i < j$ 
3  for  $N_{i,j-1,x} \in N_{i,j-1}$ 
4   $g(N_{i,j-1,x}) = \infty$ ,  $P_f(N_{i,j-1,x}) = \emptyset$ 
5  for  $N_{i-1,j,y} \in N_{i-1,j}$ 
6   $g'(N_{i,j-1,x}) = g(N_{i-1,j,y}) + t_{lc}(N_{i-1,j,y}, N_{i,j-1,x})$ 
7  if  $g'(N_{i,j-1,x}) < g(N_{i,j-1,x})$ 
8   $g(N_{i,j-1,x}) = g'(N_{i,j-1,x})$ 
9   $P_f(N_{i,j-1,x}) = (N_{i-1,j,y})$ 
10 end if;
11 end for;
12 end for;

```

```

13 for  $N_{i,j+1,z} \in N_{i,j+1}$ 
14    $g(N_{i,j+1,z}) = \infty, P_f(N_{i,j+1,z}) = \emptyset$ 
15   for  $N_{i,j-1,x} \in N_{i,j-1}$ 
16      $g'(N_{i,j+1,z}) = g(N_{i,j-1,x}) + t_c(N_{i,j-1,x}, N_{i,j+1,z})$ 
17     if  $g'(N_{i,j+1,z}) < g(N_{i,j+1,z})$ 
18        $g(N_{i,j+1,z}) = g'(N_{i,j+1,z})$ 
19        $P_f(N_{i,j+1,z}) = (N_{i,j-1,x})$ 
20  $i = i + 1;$ 
21   end if;
22 end for;
23 end for;
24 if  $i = j$  return path  $N_1 = \{D, P_f(D), P_f[P_f(D)], \dots, O\};$ 
25   end if;
26 end

```

g : 从开始到前一个节点的行车代价; g' : 从起始节点到当前节点的行车代价; t_c : 此步骤的行车代价。

5. 实验验证

5.1. 在虚拟路网上进行仿真测试

本文首先构建了虚拟路网，并进行了仿真测试。如

图10所示，本文建立了一个长宽各为5 km的网格路网，该网络在水平方向上包含八条道路，在垂直方向上也包含八条道路。每两个相邻交叉口之间的道路都是双向的，并且每一条道路的两个方向上都包括三条平行车道，如图11所示。本文在交叉口的三条进入车道上设计的交通规则是左转、直行、直行（从最里面的车道到最外面的车道）。车道的参数见表1。如图10和图11所示，红色圆圈表示起点和终点的位置。

根据图10和图11所示的道路级路网和车道级虚拟路网数据，本文验证了分层车道级路径规划算法的可行性和高效性。为了更好地展现分层车道级算法的效率，本文将对对比直接在车道级路网进行寻路的算法计算效率。路径规划的结果可见图12和图13中的红色线条。其中，前者展示道路级路径规划结果，后者则是车道级路径规划结果。在图12中我们可以看到所规划的路径在尽量避免包含黑色车道。这是因为黑色车道是最慢的车道，速度限制为 $40 \text{ km} \cdot \text{h}^{-1}$ 。由图13可以看出，本文所提出的算法在遵守交通规则的前提下，选择了最快的车道。例如，在图13中的点2处，算法所规划的路径换道到外侧的车

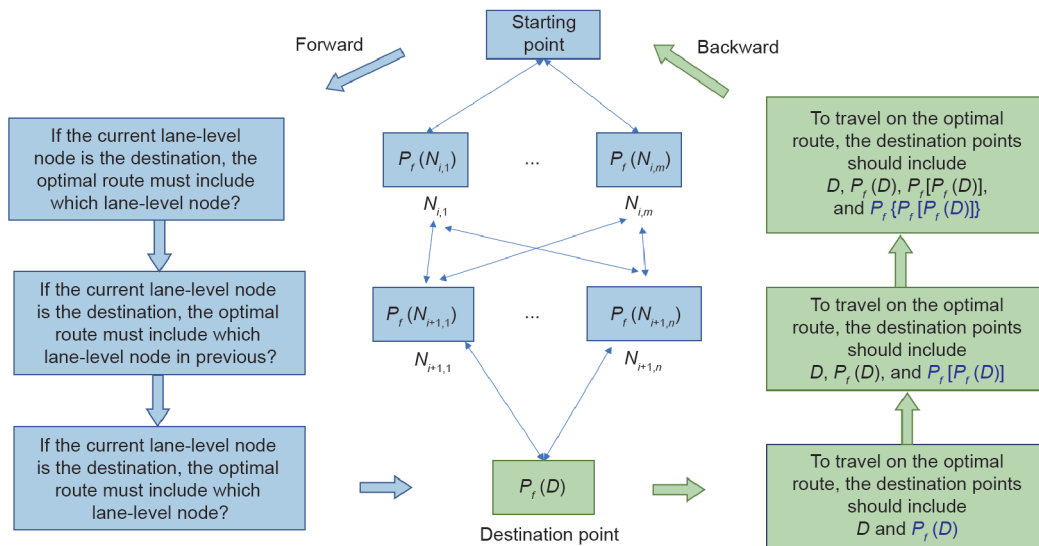


图9. 提出的逐步迭代算法，以找到最佳路径节点。

表1 仿真测试中的参数设定

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 $\{80 \text{ km} \cdot \text{h}^{-1}, 60 \text{ km} \cdot \text{h}^{-1}, 40 \text{ km} \cdot \text{h}^{-1}\}$
Speed limit of each lane	Inner: $v_x + 20 \text{ km} \cdot \text{h}^{-1}$; middle: v_x ; outer: $v_x - 20 \text{ km} \cdot \text{h}^{-1}$
Allowed behavior	Inner: only left; middle: only forward; outer: forward or right

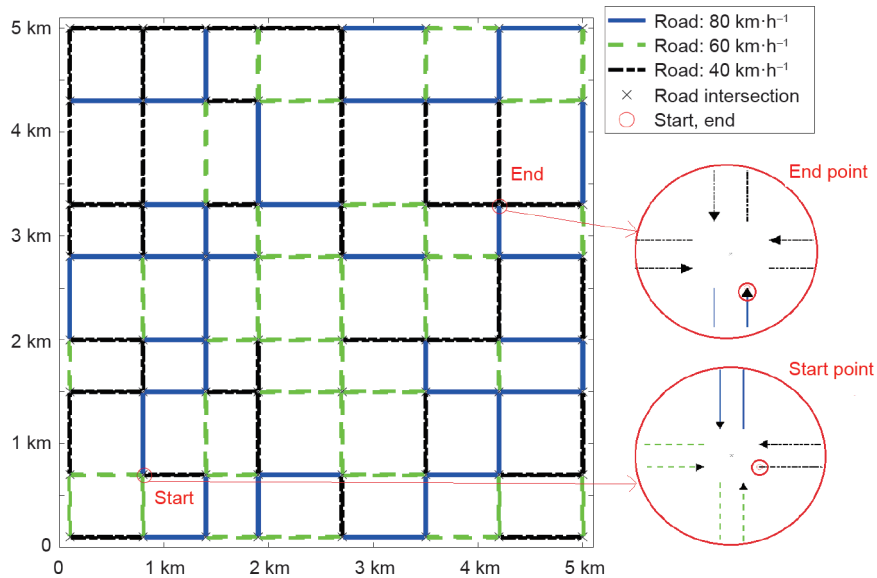


图10. 仿真测试——道路级地图，其中，交通状况由每条道路的不同行驶速度表示。

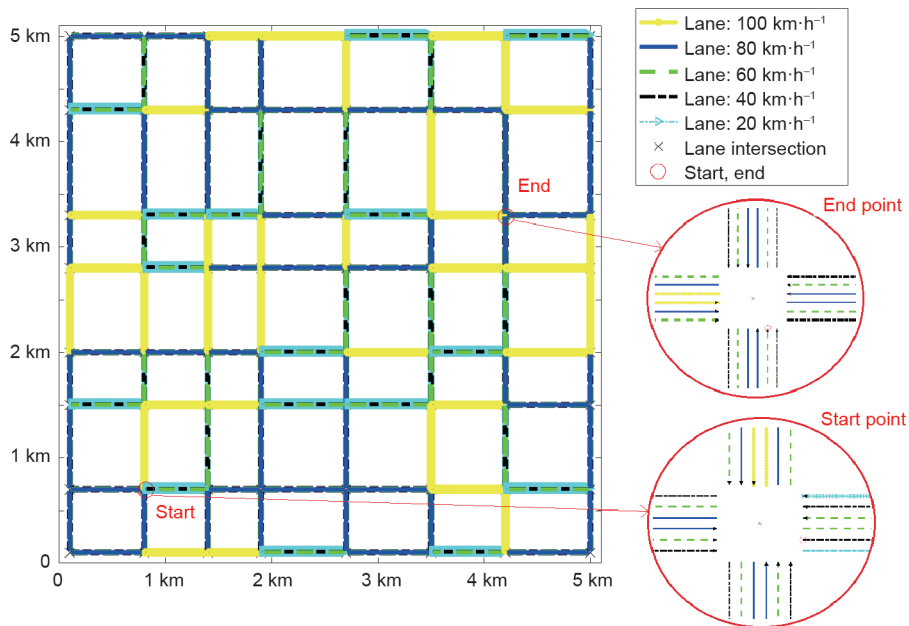


图11. 仿真测试——车道级地图，其中，交通状况由每个车道的不同行驶速度表示。

道以便之后向右转；在图13的点3处，规划的路径在经过交叉路口后换道为内侧的车道，从而减少了行驶时间。仿真测试的结果说明，多层地图模型对于分层车道级路线规划是有效的，并且本文所提出的规划方法和车道级行车代价模型能够在路网中找到车道级最优路线。

为了验证本文所提出的方法提高了路径规划的效率，本文比较了不同路径算法的计算耗时，如表2所示。本文所提出的分层路径规划算法的计算时间可以分为三个部分：初始化有向图、道路级路径规划以及车道级轨

迹规划细化。相比之下，直接车道级轨迹规划算法的计算时间包含两部分：初始化有向图和车道级轨迹规划。本文提出的算法与直接进行车道级轨迹规划相比，生成有向图的耗时大大减少：耗时仅为直接车道级规划方法的2%。这是由于直接在车道级地图上进行规划，需要处理复杂的车道级地图数据。由表2还可以看出，本文所提出的算法的核心规划耗时比直接进行车道级规划的耗时低41.5%。综上所述，仿真测试说明，本文所提出的分层规划方法在车道级轨迹规划上比直接进行规划的方法更有效。

表2 不同路径规划方法在仿真测试中的时间成本

	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	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

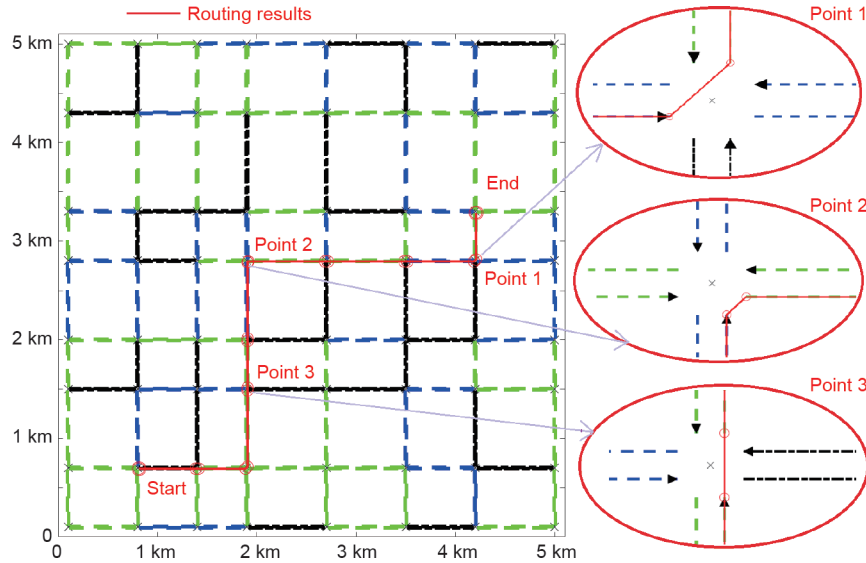


图12. 道路级规划结果。所提出的方法选择了最快的规划结果（有关每条道路的速度限制的说明参见图10）。

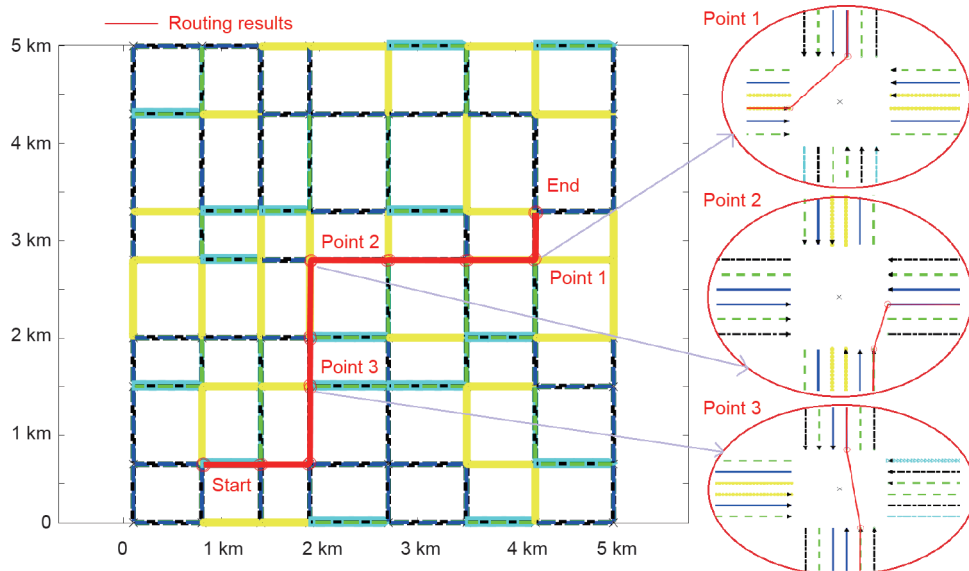


图13. 车道级规划结果。所提出的方法已根据交通规则选择了正确的车道（有关速度限制的说明参见图11）。

5.2. 真实路网数据上进行验证

本文也在真实路网上对所提出的算法进行了测试，以验证行车代价模型的准确性和路径规划算法的高效性。在课题组之前的研究中，建立了意大利米兰部分城

区的车道级电子地图[25]，如图14所示。该地图包含44个车道和16个交叉路口。具有灰色背景的黄线表示允许车道变换的平行车道。

为了验证行车代价模型的准确性，本文将首先根据

地图数据及代价模型计算某条路径的行驶时间成本，同时记录通过实际车辆的行驶时间统计数据，然后进行对比。在实验中，本文选择了两条路线，如图14所示。第一条路线，用绿线表示，长906 m，包含8个交叉路口、3个交通灯和4个让行标志。第二条路线用蓝线表示，长1132 m，包含9个交叉路口、4个交通信号灯和3个路标。试验车辆在非高峰时段进行驾驶，以便尽量符合本文研究所基于的无障碍物假设。试验结果见表3。可以看出，在某些实车测试中，实车行驶耗时明显大于其他测试（即路线1的第一次行驶和路线2的第三次行驶）。本文认为，这些耗时的原因是道路上意外的交通延误，如交叉路口的车辆堵塞。这些额外的耗时是由本研究中未

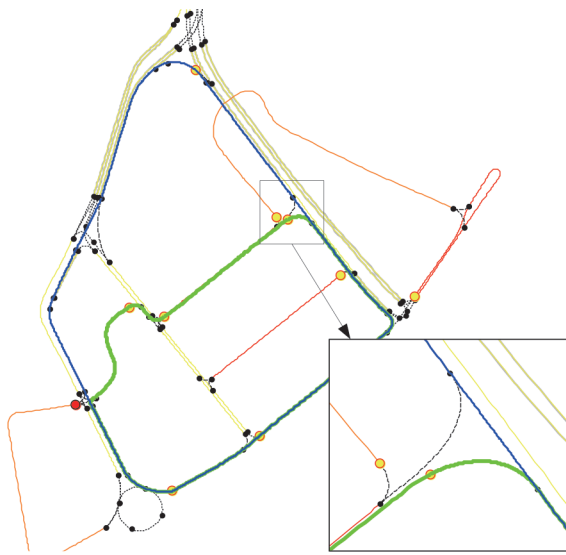


图14. 真实道路网络上的行驶成本建模测试。车道以线条表示，不同的颜色表示不同速度限制，黄色车道的限制为 $90 \text{ km}\cdot\text{h}^{-1}$ ，其他车道的限制为 $50 \text{ km}\cdot\text{h}^{-1}$ 。虚线曲线表示没有信号控制转弯的混合车道，而虚线表示具有信号控制的联合车道。停止标志和让行标志分别用红色和黄色圆圈表示。

表3 行车代价模型的测试结果

	Real travel time (s)					Average	Modeled cost (s)
	1	2	3	4	5		
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

Note: Crossed-out values are anomalies caused by unexpected traffic delays that are not included in this study; thus, these values were excluded from the averages.

表4 真实道路测试中不同路径规划方法的耗时

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

考虑的动态交通因素引起的。因此，试验数据分析时将忽略表3中的异常值，然后将多次行驶的平均时间作为实际行驶时间。试验发现实际行驶时间非常接近地图模型计算的行驶时间，这证明了本文建立的行驶成本计算方法的有效性。

本文还在惠州市城区的局部车道级电子地图上进行了车道级路径规划测试，如图15所示，其中不同颜色的线代表具有不同速度限制的车道。规划结果由图16中的红线表示。由于实际的路网往往包含较少的交叉点和车道节点，所以真实地图数据往往比4.1节中的仿真地图数据简单。从图16可以看出，本文所提出的方法选择了正确的车道，能够引导车辆到达目的地。表4比较了使用不同路径规划方法的在该城区中进行规划的时间成本。与直接进行车道级轨迹规划相比，分层规划的效率提高了54.8%。

6. 结论与展望

本研究旨在开发一种新的自动驾驶地图模型和车道级路径规划算法。首先，本文提出了一种包含了7层数据的自动驾驶电子地图架构，这种新结构的电子地图能够存储丰富的车道级信息，同时也能够实现灵活的数据调用，以便支撑从道路级到车道级的导航应用。在新型地图模型的基础上，通过分析车辆运动特性，对所规划路径的车辆行驶代价进行了分析和建模，如进行车道变换或转弯的时间成本。在行车代价建模基础上，本文计算了地图中各个路口和车道的行驶代价，并构建了面向路径规划、具有代价属性的路网地图数据。为了提升路径规划的效率，本文提出了分层车道

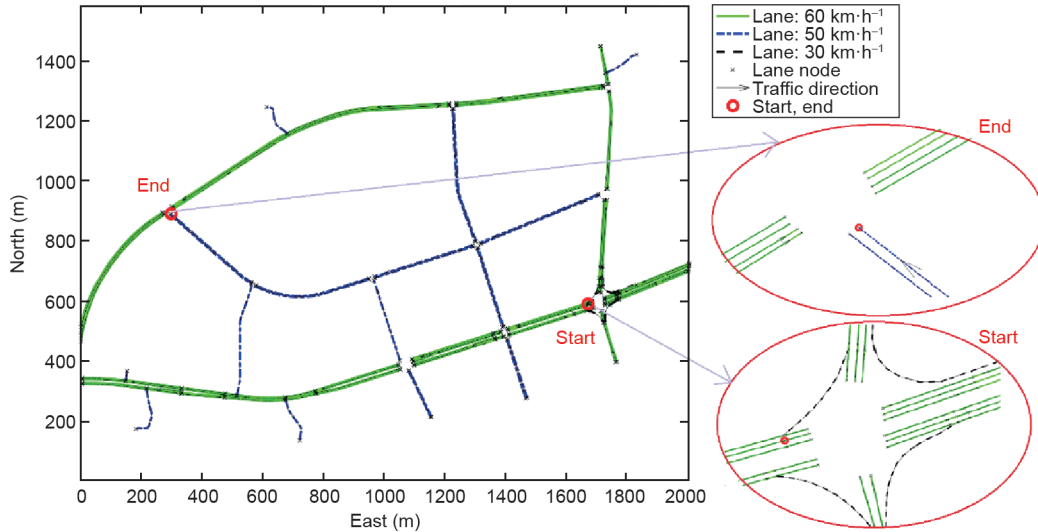


图15. 惠州市实际车道级地图数据图。车道的颜色表示车道的速度限制。

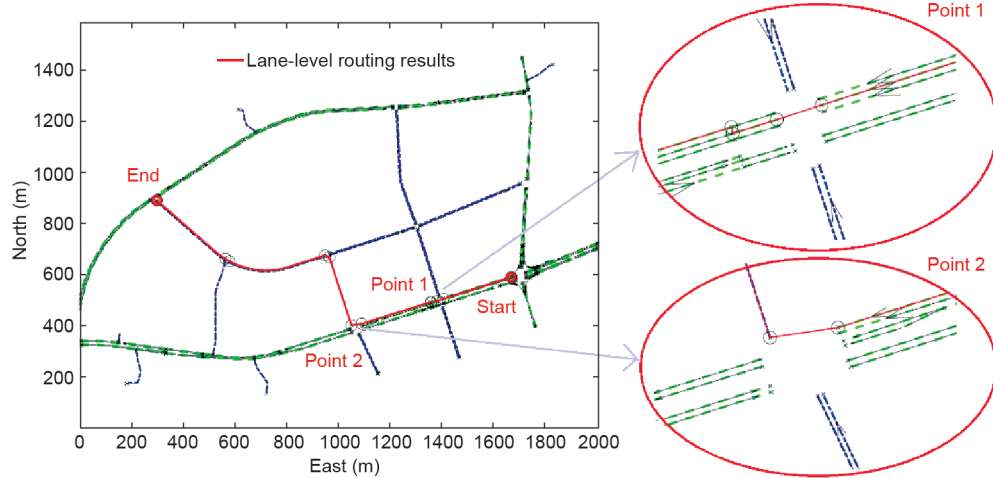


图16. 惠州市实际地图数据的车道级轨迹规划结果示意图。

级路径规划方法，并实现了道路级规划和车道级规划的融合，解决了在复杂道路网络中车道级行驶路径的快速规划。本研究中提出的多层地图数据结构和分层路径规划算法可以有效地降低基于地图的路径规划计算成本，仿真和实车的测试实验证明了分层路径规划算法的高效性。

在未来的研究里，为了提高实际导航应用中车道级轨迹规划的效率，将进一步在以下两方面开展研究工作：一是构建考虑交通不确定性的行车代价模型。在本研究中，行车代价是基于确定性的信息计算的，但是现实中的交通环境信息具有不确定性，如交通状况、天气和其他因素的不确定性。针对这些不确定性，可以通过使用模糊数[26]等方法，考虑不确定信息，从而实现更加精确的代价建模。采用这种方法进行成本建模时，我

们需要寻找路径规划时复杂性和效率之间的平衡点。二是进一步提高大型道路网络上的规划效率。如果在大型道路网络上进行路径规划，则需要更复杂的计算加速技术，特别是在候选车道级路点的搜索上，文献中已有基于路点收缩的分层寻路算法[27]等，我们在进一步的研究中将验证这些加速技术的可行性，实现在大型车道级公路网中车辆的高效路径和轨迹规划。

致谢

本项目得到了国家重点研发计划（2018YFB0105000）、国家自然科学基金（61773234和U1864203）、清华大学-丰田自动驾驶人工智能技术联合研究中心（TT2018-02）的支持，同时，北京市科技计划（D171100005117001

和Z181100005918001) 也为本项目提供了软件开发上的支持。

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. *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; Alcalá 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: VIRESS 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, Sanders P, Schultes D, Delling D. Contraction hierarchies: faster and simpler hierarchical routing in road networks. In: *MeGeoch CC, editor. WEA 2008: experimental algorithms*. Heidelberg: Springer; 2008. p. 319–33.