

Virtual Lane Boundary Generation for Human-Compatible Autonomous Driving: A Tight Coupling between Perception and Planning

Binbin Li, Dezhen Song, Ankit Ramchandani, Hsin-Min Cheng, Di Wang, Yiliang Xu, and Baifan Chen

Abstract—Existing autonomous vehicle (AV) navigation algorithms treat lane recognition, obstacle avoidance, local path planning, and lane following as separate functional modules which result in driving behavior that is incompatible with human drivers. It is imperative to design human-compatible navigation algorithms to ensure transportation safety. We develop a new tightly-coupled perception-planning framework that combines all these functionalities to ensure human-compatibility. Using GPS-camera-lidar sensor fusion, we detect actual lane boundaries (ALBs) and propose availability-reasonability-feasibility (ARF) threefold tests to determine if we should generate virtual lane boundaries (VLBs) or follow ALBs. If needed, VLBs are generated using a dynamically adjustable multi-objective optimization framework that considers obstacle avoidance, trajectory smoothness (to satisfy vehicle kinodynamic constraints), trajectory continuity (to avoid sudden movements), GPS following quality (to execute global plan), and lane following or partial direction following (to meet human expectation). Consequently, vehicle motion is more human compatible than existing approaches. We have implemented our algorithm and tested under open source data with satisfying results.

I. INTRODUCTION

As more and more companies are developing autonomous vehicles (AVs), it is important to ensure that the driving behavior of AVs is human-compatible because AVs will have to share roads with human drivers in the years to come. When planning motion for an AV, we can adjust speed and trajectory in many possible ways but not all plans guarantee human compatibility, which requires the understanding of human decision process. A human driver is far better than an AV when handling complex situations. A human driver can avoid obstacles and still respect lane markings (LMs) and traffic cones to a large degree. A human driver can override lane boundaries (LBs) in appropriate scenarios: lane markings (LMs) may disappear or be blocked by construction or parked vehicles, LMs may not be consistent with the traveling direction, a vehicle may be traveling too fast, thus being temporarily unable to follow the sudden changes in LMs, etc. In fact, there is a tight coupling between perception for scene understanding and motion planning, which involves finding an optimal trajectory under multiple objectives.

B. Li, D. Song, A. Ramchandani, H. Cheng, and D. Wang are with CSE Department, Texas A&M University, College Station, TX 77843 US. (Emails: binbinli@tamu.edu; dzsong@cse.tamu.edu; ankit61@tamu.edu; hmcheng@tamu.edu; ivanwang@tamu.edu)

Y. Xu is with Tencent America, Palo Alto, CA 94306 US. (Email: yiliangxu@tencent.com)

B. Chen is with School of Automation, Central South University, Changsha 410083 China. (Email: chenbaifan@cse.edu.cn)

This work was supported in part by National Science Foundation under NRI-1526200, 1748161, and 1925037.

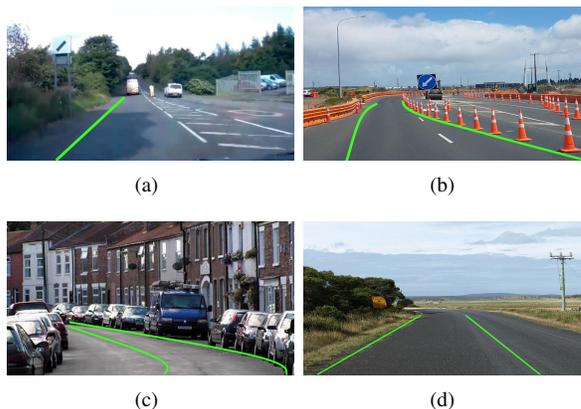


Fig. 1. We generate virtual lane boundaries for autonomous driving to ensure human compatible driving under complex road conditions: (a) current lane lacks left side lane boundary, (b) traffic cones alter roads, (c) parked cars block streets, (d) there are no LMs at all. Green curves are the VLBs generated by our algorithm (best viewed in color).

However, traditional navigation design in AVs treats functionalities such as lane recognition, obstacle avoidance, local path planning, and lane following as separate modules which results in unnatural driving behavior from a human perspective. For example, a low-level obstacle avoidance as reflex behavior often emphasizes speedy response instead of incorporating in-depth LB understanding. The resulting obstacle avoidance may not be human-compatible.

We propose a new tightly-coupled perception-planning framework to improve human-compatibility. Using GPS-camera-lidar multi-modal sensor fusion, we detect actual lane boundaries (ALBs) and propose availability-reasonability-feasibility tests to determine if we should generate virtual lane boundaries (VLBs) or follow ALBs. When needed, VLBs are generated using a dynamically adjustable multi-objective optimization framework that considers obstacle avoidance, trajectory smoothness (to satisfy vehicle kinodynamic constraints), trajectory continuity (to avoid sudden movements), GPS following quality (to execute global plan), and lane following or partial direction following (to meet human expectation). The resulting trajectory is more human compatible than existing approaches, especially when coping with difficult conditions (see Fig. 1).

We have implemented our algorithm and tested it with the KITTI open source data set. The source codes have been released on Github™. The results have shown that our algorithm automatically and dynamically switches between VLBs and ALBs. The ratio of time the VLB dominated segments range from 29% to 100% depending upon road scenarios. Our multiple-objective tightly-coupled perception-

planning framework produces high quality trajectories in city environments.

II. RELATED WORK

Our research is related to LB detection and tracking, local path planning, and obstacle avoidance.

LB detection and tracking plays an important role in autonomous driving, which has been studied for years [1], [2]. Andrade et al. [3] propose to detect and track LBs by using Hough transform and a shape-preserving spline interpolation. Li et al. [4] introduce predictive random sample consensus (RANSAC) to fit and track LBs in the presence of heavy noise and outliers. Petrovai et al. [5] apply stereovision to track 3D LBs. Huang et al. [6] detect and estimate multiple LBs by fusing calibrated video imagery and laser range data for a moving vehicle. Joshi et al. [7] use a 1D Laplacian filter to extract and track LBs from 3D lidar data. Kang et al. [8] propose a probabilistic decision-making algorithm to track curbs that uses interacting multiple model method for autonomous mobile robot navigation. Most existing methods detect and track LBs as an isolated perception problem. In this work, we tightly couple perception with planning by generating VLBs in sensor space while considering vehicle size and kinodynamic constraints.

Traditionally, obstacle avoidance is often designed as a low level reflex for a robot to stay away from obstacles. Obstacle avoidance for autonomous driving involves planning the AV's trajectory by satisfying control objectives subject to non-collision constraints. Many methods for obstacle avoidance have been proposed [9], [10]. Khatib [11] designs artificial potential field to represent the obstacles so that a robot reaches the goal without colliding with obstacles. Song et al. [12] construct a vision vector space to facilitate motion planning to avoid obstacles by fitting the dynamic requirement of a motorcycle. Kahlouche et al. [13] employ optical flow to get the information about the robot environment for visual obstacle avoidance. Sgorbissa et al. [14] integrate a prior knowledge of the environment with local perceptions, and guarantee that the robot can never be trapped in deadlocks even when operating within a partially unknown dynamic environment. For simple mobile robots in slow speed, obstacle avoidance does not have to be built on sophisticated perception model. However, an AV has to follow traffic rules and handle conflicting goals to meet human expectations.

Local path planning produces a collision-free path for AVs based on a predefined global route and *in situ* information from on-board sensors [15]. Compared with the grid-based methods [16], the sampling-based methods [17] are more widely used to find a collision-free path due to the high-speed driving requirement. Likhachev et al. [18] present a graph-based planning and re-planning algorithm, which is able to produce bounded sub-optimal solutions to speed up decision time. Chu et al. [19] propose to generate an optimal path for off-road autonomous driving with static obstacles. Li et al. [20] employ a hierarchical planning strategy by extracting a reference path from the lidar-based localization map. Bai et al. [21] utilize an intention-aware online planning approach

for AVs to drive near pedestrians safely, efficiently, and smoothly. Ma et al. [22] propose an efficient sampling-based planning method, which introduces a rule-template set based on the traffic scenes and an aggressive extension strategy of search tree. However, these dedicated planning approaches seek to find an optimal trajectory in the free space to avoid static or dynamic obstacles. The trajectory generated may not be compatible with human drivers.

III. PROBLEM DEFINITION

The vehicle is equipped with a frontal view camera, a lidar, and a GPS receiver, which is the common sensory configuration for AVs. Prior maps, such as Google™ Maps or OpenStreetMaps™ [23], are used as a part of the inputs. We have the following assumptions,

- a.1 The camera is pre-calibrated, and the nonlinear distortion of images has been removed.
- a.2 All sensor readings are synchronized.
- a.3 The coordinate system transformations between any two sensors are known by prior calibration.

All coordinate systems are right hand system and common notations are defined as follows,

- $\{\mathcal{L}\}$ defines the lidar coordinate system with x -axis pointing in the vehicle forward direction, y -axis pointing to the left, and z -axis pointing upward. $\mathbf{P}_{i,t} = [x_{i,t}, y_{i,t}, z_{i,t}]^T \in \mathcal{R}^3$ is the i -th 3D lidar point with respect to $\{\mathcal{L}\}$ at time $t \in \{0, 1, \dots, T\}$, and $\mathcal{P}_t := \{\mathbf{P}_{i,t}\}$ is the set of lidar points at time t .
- $\{\mathcal{C}\}$ defines the camera coordinate system with x -axis pointing to the right of the vehicle lateral direction and z -axis pointing forward coinciding with the front-view camera's principal axis.
- $\{\mathcal{I}\}$ defines image coordinate system. Let $\mathbf{p}_{k,t} = [u \ v]^T \in \{\mathcal{I}\}$ be the k -th pixel point in image \mathbf{I}_t at time t , where (u, v) is the image coordinate.
- $\{\mathcal{W}\}$ defines the world coordinate system which overlaps with $\{\mathcal{L}\}$ at the vehicle starting position.

Denote the left and right LBs in $\{\mathcal{W}\}$ by ${}^{\mathcal{W}}\mathbf{L}_l$ and ${}^{\mathcal{W}}\mathbf{L}_r$ at time t , respectively. Note that left superscript in this paper describes the coordinate system for the corresponding variable. With the assumptions and notations defined, our problem is defined as follows,

Problem 1: Given a prior map, current GPS position, and *in situ* camera and lidar inputs, and velocity profile and global route from a global planner, recognize, generate and track LBs ${}^{\mathcal{W}}\mathbf{L}_l$ and ${}^{\mathcal{W}}\mathbf{L}_r$ in $\{\mathcal{W}\}$, or report when the VLBs cannot be generated.

IV. ALGORITHM

Fig. 2 shows the system diagram. It mainly contains the following blocks: A) Free-space detection, B) VLB generation where we perform ALB detection and also determine how we should generate VLBs, and C) VLB registration where we track the LBs through an extended Kalman filter (EKF) and re-project VLBs in $\{\mathcal{W}\}$. We start with the free-space detection.

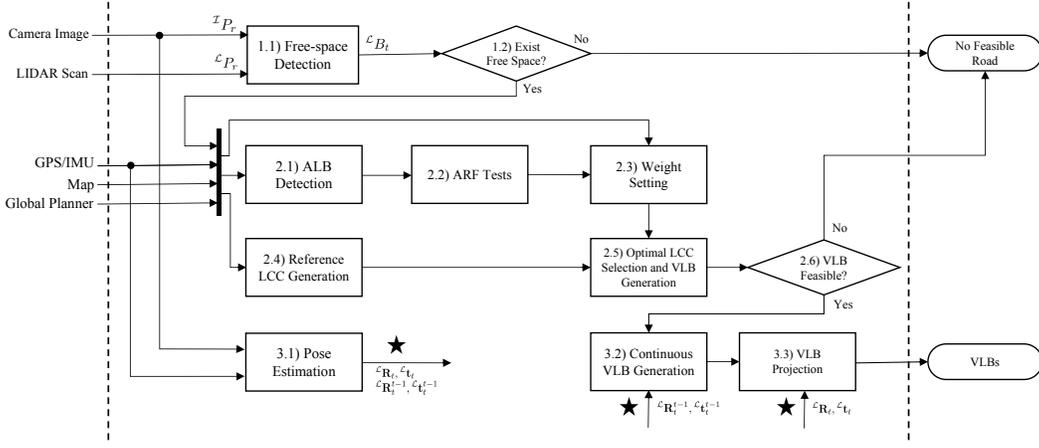


Fig. 2. System diagram. The solid star represents the output of pose estimation, which is also the input to the continuous LB generation and LB projection.

A. Free-space Detection

The free-space is collision free surface in front of the vehicle which can be defined by road edges and obstacle boundaries. We detect free space in both camera and lidar modalities and extract free-space surface boundary in $\{\mathcal{L}\}$ (see Box 1.1 in Fig. 2).

We start with recognizing road surface in both image and lidar data based on our prior work [24] where we have employed camera-lidar fusion to obtain road surface pixel set $\mathcal{I}P_r$ in image coordinate $\{\mathcal{I}\}$ using the appearance classification. We also have the corresponding 3D point lidar point cloud set $\mathcal{L}P_r \subset \mathcal{P}_t$ for $\mathcal{I}P_r$. For each point $\mathbf{p}_r \in \mathcal{I}P_r$ and its corresponding lidar point $\mathbf{P}_r \in \mathcal{L}P_r$, we have the projection relationships between them, $\tilde{\mathbf{p}}_r = c_p \mathbf{K} [{}^c_{\mathcal{L}}\mathbf{R} \quad {}^c_{\mathcal{L}}\mathbf{t}] \tilde{\mathbf{P}}_r$ and $\mathbf{P}_r = c_q [\mathbf{K}^c_{\mathcal{L}} \mathbf{R}]^{-1} \tilde{\mathbf{p}}_r - {}^c_{\mathcal{L}}\mathbf{R}^{-1} {}^c_{\mathcal{L}}\mathbf{t}$, where c_p and c_q are scalars, a vector with symbol ‘ \sim ’ on top is in its homogeneous representation, \mathbf{K} is the intrinsic camera matrix under the pin hole model, and ${}^c_{\mathcal{L}}\mathbf{R}$ and ${}^c_{\mathcal{L}}\mathbf{t}$ are the rotation matrix and translation vector between $\{\mathcal{L}\}$ and $\{\mathcal{C}\}$, respectively. We also use two more inputs from [24]: the road surface model with coefficient vector \mathbf{H}_r^* which is acquired by fitting points in $\mathcal{L}P_r$ to a polynomial model, and $d_{\perp}(\mathbf{H}_r^*, \mathbf{P}_{i,t})$ which is the shortest distance for a point $\mathbf{P}_{i,t}$ to the road surface.

Building on these prior results, we design a two-step approach to obtain 3D free-space surface boundary points. 1) We only keep lidar points $\mathcal{L}C_t$ with small elevation difference to the surface model, $\mathcal{L}C_t = \{\mathbf{P}_{i,t} | c_l \leq d_{\perp}(\mathbf{H}_r^*, \mathbf{P}_{i,t}) \leq c_u, \mathbf{P}_{i,t} \in \mathcal{P}_t\}$, where c_l and c_u are thresholds. 2) We compute the average surface normal of each pixel’s neighbor set and use it to determine if it is on the smooth surface. Let us detail the second step here.

For each point $\mathbf{P}_{i,t} \in \mathcal{L}C_t$, we can find its neighbor set $\mathcal{L}E_i$ by selecting the K -nearest neighbors (KNNs) [25] with an upper bound d_r . $\mathcal{L}E_i = \{\mathbf{P}_{j,t} | \|\mathbf{P}_{i,t} - \mathbf{P}_{j,t}\| \leq d_r, \mathbf{P}_{j,t} \in \mathcal{L}C_t\}$, where index variable $j \in \mathcal{N}$ satisfies $j \neq i$ and $1 \leq j \leq K$. Next we apply methods in [26] to extract surface normal for the neighbor set to determine if $\mathbf{P}_{i,t}$ is a smooth road point. Define $\hat{\mathbf{C}}_{i,t} = \frac{1}{|\mathcal{L}E_i|} \sum_{\mathbf{P}_{j,t} \in \mathcal{L}E_i} \mathbf{P}_{j,t}$ to be the 3D

centroid of $\mathcal{L}E_i$, and

$$d_e(\mathbf{P}_{i,t}) = \frac{1}{|\mathcal{L}E_i|} \sum_{\mathbf{P}_{j,t} \in \mathcal{L}E_i} \frac{\|\mathbf{P}_{i,t} - \mathbf{P}_{j,t}\|}{\|\mathbf{P}_{i,t}\|}, \quad (1)$$

to be the normalized average distance for all points in $\mathcal{L}E_i$ to $\mathbf{P}_{i,t}$. Define $d_s(\mathbf{P}_{i,t}, \mathbf{P}_{j,t}) = \|\mathbf{P}_{j,t} - \hat{\mathbf{C}}_{i,t}\| / \|\mathbf{P}_{i,t}\|$ as the normalized distance for the point $\mathbf{P}_{j,t}$ to remove scale effect. Define a weight value $w_{j,t}$ for the $\mathbf{P}_{j,t}$ to be

$$w_{j,t} = \begin{cases} \exp\left\{-\frac{d_s(\mathbf{P}_{i,t}, \mathbf{P}_{j,t})^2}{d_e(\mathbf{P}_{i,t})^2}\right\}, & \text{if } d_s(\mathbf{P}_{i,t}, \mathbf{P}_{j,t}) \geq d_e(\mathbf{P}_{i,t}), \\ 1, & \text{otherwise.} \end{cases}$$

Let λ_1, λ_2 and λ_3 be the eigenvalues of the correlation matrix $\sum_{j=1}^{|\mathcal{L}E_i|} w_{j,t} (\mathbf{P}_{j,t} - \hat{\mathbf{C}}_{i,t})(\mathbf{P}_{j,t} - \hat{\mathbf{C}}_{i,t})^{\top}$, and suppose $\lambda_1 \leq \lambda_2 \leq \lambda_3$. According to [27], a point on smooth road surface has λ_1 significantly smaller than the other two; for a free-space surface boundary point, λ_1 and λ_2 are substantially smaller than λ_3 . Therefore, we can use this property to obtain boundary point set $\mathcal{L}F_t$ by thresholding,

$$\mathcal{L}F_t = \{\mathbf{P}_{i,t} | \lambda_3 / \sum_{i=1}^3 \lambda_i \geq \lambda_d, \mathbf{P}_{i,t} \in \mathcal{L}C_t\}, \quad (2)$$

where λ_d is the threshold. Inspired by [28], we can further remove noisy points in set $\mathcal{L}F_t$ by examining surface normal vector directions. For a point $\mathbf{P}_{i,t} \in \mathcal{L}F_t$, we compute the average surface normal $\theta_{i,t}$ as follows, $\theta_{i,t} = \frac{1}{|\mathcal{L}E_i|} \sum_{j=1}^{|\mathcal{L}E_i|} \arctan \frac{|z_{i,t} - z_{j,t}|}{\sqrt{|x_{i,t} - x_{j,t}|^2 + |y_{i,t} - y_{j,t}|^2}}$. Note that points on the road surface have small variations in z direction which means small $\theta_{i,t}$ values. Therefore, we can identify boundary/obstacle points by thresholding on $\theta_{i,t}$ and obtain free-space surface boundary point set $\mathcal{L}B_t$ as follows,

$$\mathcal{L}B_t = \{\mathbf{P}_{i,t} | \theta_{i,t} \geq \theta_v, \mathbf{P}_{i,t} \in \mathcal{L}F_t\}, \quad (3)$$

where θ_v is the threshold.

Next, we need to verify if there is available free space in front of the vehicle given vehicle kinodynamic and size constraints. We apply a state lattice planner [29] to generate a set of seven candidate arc trajectories $\{\mathcal{L}L_{l_p}\}_{l_p=1}^7$ that evenly cover curvatures in the allowable range given the current speed. The length of arc is the braking distance. Let d_v be

the haft width of the vehicle. We evaluate all points in the region swiped by the vehicle if following the arc ${}^{\mathcal{L}}\mathbf{L}_{l_p}$ which is set ${}^{\mathcal{L}}P_{l_p} = \{\mathbf{P}_{i,t} | \min \|\mathbf{P}_{i,t} - \mathbf{P}_w\| \leq d_v, \mathbf{P}_{i,t} \in \mathcal{P}_t, \mathbf{P}_w \in {}^{\mathcal{L}}\mathbf{L}_{l_p}\}$. Denote the logic OR operator by \vee . If we have,

$$\bigvee_{l_p} \{ {}^{\mathcal{L}}P_{l_p} \cap {}^{\mathcal{L}}B_t = \emptyset \} = 1, \quad (4)$$

then the free space exists (see decision box 1.2 in Fig. 2) and we move on to next step. Otherwise, there is no feasible road and global planner needs to be notified to re-plan route. The global planner concerns overall routing and is not the concern of this paper.

B. VLB Generation

VLBs and corresponding lane center curves (LCCs) regulate how the vehicle can move. Generating them is equivalent to local planning but with tight coupling to perception and vehicle kinodynamic constraints. By tight coupling we mean that LCCs and VLBs are evaluated directly and locally in the sensor space without an additional world model. We have to answer two important questions here: 1) when should we decide to deviate from ALBs? and 2) how to generate VLBs to balance multiple requirements to be human compatible?

1) *LB representation*: Before we dive into details, let us define LCC and the information obtained from ALB as shown in our prior work [24]. In fact, it is also possible to use lane detection methods from other existing works. From [24], we obtain ALB and the corresponding LCC ${}^{\mathcal{L}}\mathbf{L}_a$ is represented as cubic B-spline curves that are made of l piecewise polynomial functions,

$${}^{\mathcal{L}}\mathbf{L}_{a,l}(s) = \mathbf{a}_{l,0} + \mathbf{a}_{l,1}s + \mathbf{a}_{l,2}s^2 + \mathbf{a}_{l,3}s^3 \quad (5)$$

be the l -th curve segment where $\{\mathbf{a}_{l,j} | l = 1, 2, \dots, n_c - 3, j = 0, 1, 2, 3\}$ are 3-vectors for polynomial coefficients, $0 \leq s \leq s_e$, n_c is the number of the control points for the spline curve, and $s_e = n_c + 3$ is the maximum knots. Subscript a indicates this LCC is from ALB. As shown in [24], for a given LCC and a lane width, it is trivial to obtain the left and right LBs ${}^{\mathcal{L}}\mathbf{L}_l(s)$ and ${}^{\mathcal{L}}\mathbf{L}_r(s)$, respectively, and vice versa.

2) *Examining ALB quality*: For question 1), we determine if the vehicle should follow ALBs using *availability, reasonability, and feasibility* (ARF) tests (see Box 2.2 in Fig. 2). For *availability*, we examine if ALBs provide a sufficiently long trajectory to follow.

$$\int_0^{s_e} \|{}^{\mathcal{L}}\mathbf{L}'_a(s)\| ds \geq l_{\min}, \quad (6)$$

where $\|\cdot\|$ is the vector l^2 -norm and l_{\min} is the trajectory length threshold.

For *reasonability*, we check if the LCC ${}^{\mathcal{L}}\mathbf{L}_a$ heading agrees with the vehicle's current heading. Let $\mathbf{n}_v \in \mathcal{R}^3$ point to the vehicle's driving direction at time t , and $\mathbf{n}_u \in \mathcal{R}^3$ be the first derivative of the LCC ${}^{\mathcal{L}}\mathbf{L}_a(s)$ when $s = 0$, respectively. Let $\langle \cdot, \cdot \rangle$ represent the inner product between two vectors. For a threshold $\beta_l = 10^\circ$, if

$$\arccos \frac{\langle \mathbf{n}_v, \mathbf{n}_u \rangle}{\|\mathbf{n}_v\| \|\mathbf{n}_u\|} \leq \beta_l, \quad (7)$$

then the current LCC ${}^{\mathcal{L}}\mathbf{L}_a$ is reasonable.

For *feasibility*, we want to make sure that the curvature of the LCC is compatible with the current vehicle speed. We precompute a look-up table offline considering the vehicle speed and the curvature. Let $\langle \cdot \times \cdot \rangle$ represent vector cross product. Let κ_{\max} be the maximum allowable LCC curvature for the vehicle given the current forward speed v_t . We have a feasible LCC if

$$\frac{\| \langle {}^{\mathcal{L}}\mathbf{L}'_a(s) \times {}^{\mathcal{L}}\mathbf{L}''_a(s) \rangle \|}{\|{}^{\mathcal{L}}\mathbf{L}'_a(s)\|^3} \leq \kappa_{\max}. \quad (8)$$

ARF test results are used to set weights in selecting LCCs for VLB and will be detailed later in Section IV-B.4.

3) *VLB generation*: For question 2), to generate human-compatible VLBs, we need to a) respect partial information from ALB, b) follow GPS waypoints, c) avoid dynamic and stationary obstacles, and d) consider vehicle kinodynamic constraints.

Therefore, we need the planned GPS trajectory as a seed. From the current GPS reading and the prior map, we can extract a set of GPS way points to represent the road ahead. The number of points depends on the velocity of the vehicle and the minimum number needed to construct a cubic B-spline representation. We can project these 2D map points onto the road surface model to obtain 3D points. Applying cubic B-spline fitting and coordinate transformation, we obtain its representation in current lidar coordinates to be ${}^{\mathcal{L}}\mathbf{L}_g(s)$ where subscript g means this is from GPS reference. Note that LCC of VLBs should start with the endpoint of previous LCC (denoted by ${}^{\mathcal{L}}\mathbf{L}^-(s)$) at time $t - 1$ which happens when $s = s_e$. ${}^{\mathcal{L}}\mathbf{L}_g(s)$ and ${}^{\mathcal{L}}\mathbf{L}^-$ do not necessarily overlap. A minimum distance parallel shift of ${}^{\mathcal{L}}\mathbf{L}_g(s)$ allows point ${}^{\mathcal{L}}\mathbf{L}^-(s_e)$ be located on the shifted ${}^{\mathcal{L}}\mathbf{L}_g(s)$. The shifted ${}^{\mathcal{L}}\mathbf{L}_g(s)$ is cropped to start at the point and serve as the seed trajectory for candidate trajectory generation. In fact, the shifted ${}^{\mathcal{L}}\mathbf{L}_g(s)$ does not need to be collision free. The new trajectory along with velocity profile and vehicle size are then used to generate candidate trajectories by sampling on lattice using [29], which provide us a set of candidate LCCs ${}^{\mathcal{L}}\mathbf{L} \subset \mathcal{L}_c$ considering the vehicle's kinodynamic constraints. Of course, any candidate LCC ${}^{\mathcal{L}}\mathbf{L}$ also have to pass our ARF tests. If none of the candidate LCC pass ARF tests, the system reports "no feasible road" to the global planner.

We then select the best candidate LCC by minimizing a cost function $C({}^{\mathcal{L}}\mathbf{L})$ (see Box 2.5 in Fig. 2)

$${}^{\mathcal{L}}\mathbf{L}^* = \arg \min_{{}^{\mathcal{L}}\mathbf{L} \subset \mathcal{L}_c} C({}^{\mathcal{L}}\mathbf{L}), \quad (9)$$

that is designed to consider human compatibility by integrating smoothness f_s , obstacle avoidance f_o , GPS trajectory following f_g , trajectory continuity f_c , and ALBs f_a as follows

$$C({}^{\mathcal{L}}\mathbf{L}) = f_s({}^{\mathcal{L}}\mathbf{L}) + w_2 f_o({}^{\mathcal{L}}\mathbf{L}, {}^{\mathcal{L}}B_t) + w_3 f_g({}^{\mathcal{L}}\mathbf{L}, {}^{\mathcal{L}}\mathbf{L}_g) + w_4 f_c({}^{\mathcal{L}}\mathbf{L}, {}^{\mathcal{L}}\mathbf{L}^-) + w_5 f_a({}^{\mathcal{L}}\mathbf{L}, {}^{\mathcal{L}}\mathbf{L}_a), \quad (10)$$

where w_2, \dots, w_5 are non-negative weighting variables.

Function $f_s(\mathcal{L}\mathbf{L})$ controls the smoothness of the LCC [30],

$$f_s(\mathcal{L}\mathbf{L}) = \int_0^{s_e} \|\mathcal{L}\mathbf{L}'(s)\|^2 ds + w_1 \int_0^{s_e} \|\mathcal{L}\mathbf{L}''(s)\|^2 ds, \quad (11)$$

where w_1 is a non-negative weight variable, $[0, s_e]$ define spline parameter range for the LCC.

Function $f_o(\mathcal{L}\mathbf{L}, \mathcal{L}B_t)$ is the cost related to the clearance to boundary/obstacle $\mathcal{L}B_t$ set in (3). Let $d_o^* = \min_{\mathbf{P}_{i,t} \in \mathcal{L}B_t, 0 \leq s \leq s_e} \|\mathcal{L}\mathbf{L}(s) - \mathbf{P}_{i,t}\|$, be the shortest distance between a candidate LCC and a road edge point $\mathbf{P}_{i,t} \in \mathcal{L}B_t$, we have

$$f_o(\mathcal{L}\mathbf{L}, \mathcal{L}B_t) = \begin{cases} 0 & \text{if } d_o^* \geq d_r, \\ \frac{c_b}{d_l - d_r} (d_o^* - d_r) & \text{if } d_l < d_o^* < d_r, \\ \infty & \text{otherwise.} \end{cases} \quad (12)$$

where c_b is linear cost coefficient for distance to obstacle, d_l and d_r define the distance interval where the linear cost function is applied.

Cost function $f_g(\mathcal{L}\mathbf{L}, \mathcal{L}\mathbf{L}_g)$ wants the output trajectory to be similar to that of the GPS trajectory,

$$f_g(\mathcal{L}\mathbf{L}, \mathcal{L}\mathbf{L}_g) = \int_0^{s_e} \|\mathcal{L}\mathbf{L} - \mathcal{L}\mathbf{L}_g\|^2 ds. \quad (13)$$

Cost function $f_c(\mathcal{L}\mathbf{L}, \mathcal{L}\mathbf{L}^-)$ maintains continuity of LCC from prior period $\mathcal{L}\mathbf{L}^-$ (noting it has been transformed to current $\{\mathcal{L}\}$ coordinate),

$$f_c(\mathcal{L}\mathbf{L}, \mathcal{L}\mathbf{L}^-) = \int_0^{s_e} \|\mathcal{L}\mathbf{L} - \mathcal{L}\mathbf{L}^-\|^2 ds. \quad (14)$$

This cost function helps avoid sudden motion and makes the LCC more compatible with human drivers.

Cost function $f_a(\mathcal{L}\mathbf{L}, \mathcal{L}\mathbf{L}_a)$ regulates the LCC to be close to ALBs,

$$f_a(\mathcal{L}\mathbf{L}, \mathcal{L}\mathbf{L}_a) = \int_0^{s_e} \|\mathcal{L}\mathbf{L} - \mathcal{L}\mathbf{L}_a\|^2 ds. \quad (15)$$

This function regulates LCC to follow ALBs as much as possible which makes LCC to meet human expectation better.

4) *Weight settings*: Non-negative weighting variables w_1 , w_2 , w_3 , w_4 , and w_5 play an important role in regulating the LCC. This is done before LCC generation (see Box 2.3 in Fig. 2).

w_1 and w_4 control the smoothness of the resulting LCC. They should be an increasing function of velocity due to vehicle kinodynamic constraints. They are also related to driving status. If the vehicle decides to make a turn or switch lanes as instructed by the global planner, then we set them to be zero since we do not need to follow the previous direction.

w_2 controls how conservative the vehicle should be in obstacle avoidance. It should be a function of the relative velocity to obstacles. For example, the existence of a cyclist demands higher w_2 settings.

w_3 controls GPS following quality. It should be determined by how good the prior map quality is. If the road is under construction and the prior map has not been updated, then we should reduce w_3 to allow more deviation from the original map.

w_5 is adjusted according to ARF test results. If ALBs do not exist or are not reasonable (i.e. fail the first two tests of ARF tests), we set w_5 to be 0 because there is no trustable $\mathcal{L}\mathbf{L}_a$. However, w_5 remains positive if ALBs are infeasible due to vehicle kinodynamic constraints. If ALBs pass all ARF tests, then w_5 is set to the highest value to ensure good lane following.

C. VLB Registration

So far, we have obtained LCCs which are computed in local lidar coordinates and are piece-wise polynomials over time. To ensure a smooth and continuous trajectory in $\{\mathcal{W}\}$, we apply an EKF to track VLBs to generate and register continuous curves (see Boxes 3.2 and 3.3 in Fig. 2).

We need the coordinate transformation from time $t-1$ to t . This can be obtained using the optimization-based multi-sensor state estimator [31] (see Box 3.1 in Fig. 2)). Denote the rotation matrix and translation vector with respect to $\{\mathcal{W}\}$ at time t by ${}^{\mathcal{L}}\mathbf{R}_t$ and ${}^{\mathcal{L}}\mathbf{t}_t$, respectively. Let ${}^{\mathcal{L}}\mathbf{R}_{t-1}^t$ and ${}^{\mathcal{L}}\mathbf{t}_{t-1}^t$ be the relative rotation matrix and translation vector from $t-1$ to t , respectively. We have ${}^{\mathcal{L}}\mathbf{R}_{t-1}^t = {}^{\mathcal{L}}\mathbf{R}_{t-1} {}^{\mathcal{L}}\mathbf{R}_t^{-1}$ and ${}^{\mathcal{L}}\mathbf{t}_{t-1}^t = {}^{\mathcal{L}}\mathbf{t}_{t-1} - {}^{\mathcal{L}}\mathbf{R}_{t-1}^t {}^{\mathcal{L}}\mathbf{t}_t$. Recall the cubic B-spline curves are made of piecewise polynomial functions, and each polynomial function needs four control points to satisfy its continuity properties. We sort the control points for the optimal LCC $\mathcal{L}\mathbf{L}^*$ according to the increasing order of their distance to the origin of $\{\mathcal{L}\}$. let ${}^{\mathcal{L}}P_* = \{\mathbf{P}_1, \mathbf{P}_2, \dots, \mathbf{P}_{n_c}\}$ be the control point set for the $\mathcal{L}\mathbf{L}^*$. Here, $\|\mathbf{P}_p\| < \|\mathbf{P}_q\|$ for $\mathbf{P}_p, \mathbf{P}_q \in {}^{\mathcal{L}}P_*$ if $p < q$. Let

$$\mathbf{z}_{p,t} = [\mathbf{P}_p^\top, \mathbf{P}_{p+1}^\top, \mathbf{P}_{p+2}^\top, \mathbf{P}_{p+3}^\top]^\top, \quad (16)$$

be the observations for the LCC for $p = 1, 2, \dots, n_c - 3$. Define a state vector

$$\mathbf{x}_{p,t-1} = [\mathbf{a}_{p,3}^\top, \mathbf{a}_{p,2}^\top, \mathbf{a}_{p,1}^\top, \mathbf{a}_{p,0}^\top]^\top, \quad (17)$$

through (5) for the LCC at time $t-1$. The state transition function is just the coordinate system transformation between adjacent time epochs,

$$\mathbf{x}_{p,t} = \mathcal{R}_{t-1}^t \mathbf{x}_{p,t-1} + \begin{bmatrix} \mathbf{0}_{3 \times 1} \\ \mathbf{0}_{3 \times 1} \\ \mathbf{0}_{3 \times 1} \\ \mathcal{L}\mathbf{t}_{t-1}^t \end{bmatrix} + w_t, \quad (18)$$

where w_t has zero mean and covariance \mathbf{Q}_t , and $\mathcal{R}_{t-1}^t = \text{diag}({}^{\mathcal{L}}\mathbf{R}_{t-1}^t, {}^{\mathcal{L}}\mathbf{R}_{t-1}^t, {}^{\mathcal{L}}\mathbf{R}_{t-1}^t, {}^{\mathcal{L}}\mathbf{R}_{t-1}^t)$ is a diagonal block matrix. According to [32], the observation function is

$$\mathbf{z}_{p,t} = \begin{bmatrix} \mathbf{0}_{3 \times 3} & \frac{2}{3}\mathbf{I} & -\mathbf{I} & \mathbf{I} \\ \mathbf{0}_{3 \times 3} & -\frac{1}{3}\mathbf{I} & \mathbf{0}_{3 \times 3} & \mathbf{I} \\ \mathbf{0}_{3 \times 3} & \frac{2}{3}\mathbf{I} & \mathbf{I} & \mathbf{I} \\ 6\mathbf{I} & \frac{11}{3}\mathbf{I} & 2\mathbf{I} & \mathbf{I} \end{bmatrix} \mathbf{x}_{p,t} + m_t, \quad (19)$$

where m_t is zero mean and has the covariance $\mathbf{\Omega}_t$, and \mathbf{I} is a 3×3 identity matrix. We continuously predict and update the EKF as more data comes in, and register the optimal LCC by

$$\mathcal{W}\mathbf{L}^*(s) = {}^{\mathcal{L}}\mathbf{R}_t^\top \mathcal{L}\mathbf{L}^*(s) - {}^{\mathcal{L}}\mathbf{t}_t, \quad (20)$$

from $\{\mathcal{L}\}$ to $\{\mathcal{W}\}$. We also apply (20) to obtain the left VLB $\mathcal{W}\mathbf{L}_l(s)$ and the right VLB $\mathcal{W}\mathbf{L}_r(s)$, respectively.

TABLE I
%VLBS ON KITTI DATASET

Sequence	Duration	length	% ($w_5 = 0$)
2011_09_26_drive_0035	13 s	60.41 m	100%
2011_09_26_drive_0039	40 s	297.09 m	100%
2011_09_26_drive_0051	44 s	255.42 m	92%
2011_09_26_drive_0056	30 s	419.95 m	29%

V. EXPERIMENTS

We have implemented the proposed method in C++ and shared it on Github¹. It is tested on a Laptop PC with an Intel[®] Core[™]i7-3517U CPU@1.90GHz and 8 GB RAM. We test our approach using the KITTI dataset [33], which contains images covering a variety of street scenes captured from a vehicle driving around the city of Karlsruhe.

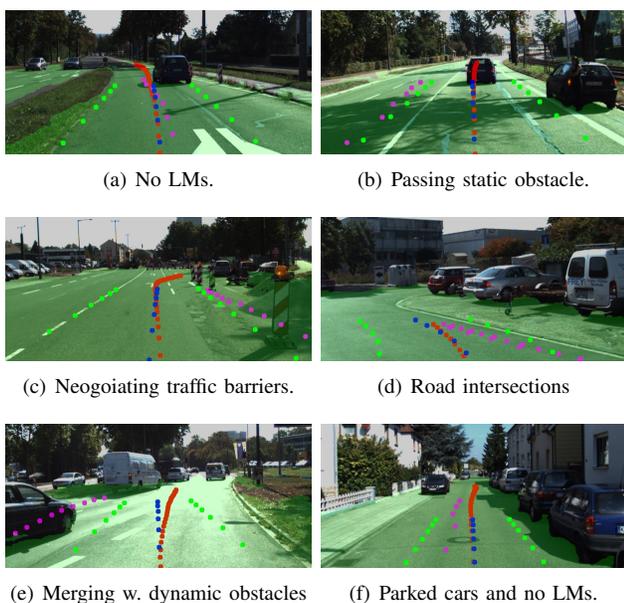
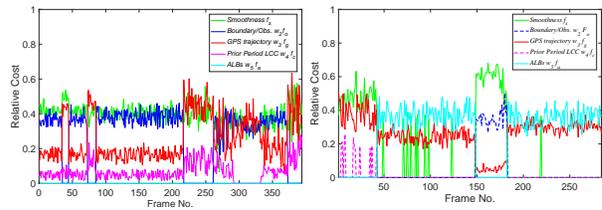


Fig. 3. Sample algorithm outputs for six different scenarios (Best viewed in color).

We have tested our algorithm on four different sequences of two categories from KITTI dataset including city and residential area (see Tab. I). In all cases, our algorithm can generate feasible LCC and VLBs to guide the vehicle. In last column, we track the ratio when $w_5 = 0$ because it indicates that the vehicle decides to deviate from ALBs. The ratio varies from 29% to 100% due to different road scenarios. Some road segments have great ALBs and do not need VLB generation as much (e.g. the fourth row) while some roads do not have ALBs at all (e.g. the first two rows).

Sample outputs are shown in Fig. 3. The green masked area is the free space detected by the algorithm. Four different dotted lines are drawn on the six figures: purple lines represent GPS waypoints from Google maps, blue lines are the algorithm output LCCs ${}^{\mathcal{W}}\mathbf{L}^*(s)$, red lines are the high precision GPS recording of actual human driving the vehicle which can be viewed as the human decision counterpart,



(a) 2011_09_26_drive_0039 (b) 2011_09_26_drive_0056

Fig. 4. Contribution to LCC cost by different components.

and two green lines are the left LB ${}^{\mathcal{W}}\mathbf{L}_l(s)$ and the right LB ${}^{\mathcal{W}}\mathbf{L}_r(s)$, respectively. It is clear that way points from Google maps are too lousy to be used as direct navigation guides, as indicated by the poor quality of purple lines. When comparing our algorithm outputs to the GPS recording of the human driving, blue lines are quite in agreement with red lines with the only exception in Fig. 3(e). Note that red lines extend beyond blue lines due to different trajectory length which does not mean that they do not agree. Even in Fig. 3(e), both the blue line and the red line are viable choices. In all cases, our algorithm can generate LCCs that are compatible with human expectations.

Fig. 4 further illustrates how different components contribute to the VLB LCC selection in (10) using the second and fourth sequence in Tab. I. The plots are the normalized ratios in the overall objective function value. During the computation, the weight settings for the optimization problem are set as $w_1 = 1$, $w_2 = 1$, $w_3 = 2$, $w_4 = 0.2$ and $w_5 = 5$ for non-zero cases to balance the multiple objectives in the LCC selection. It is clear that every component in (10) plays a role in determining LCC.

The more interesting part is the dynamic change of ratios, as shown by Fig. 4(a) which really exposes the inner-works of VLB generation. First, there are no ALBs in the entire sequence and w_5 has to be zero during 100% of the time. Second, both f_s (green solid line) and $w_2 f_o$ (blue dashed line) are relatively high throughout the entire sequence because it is important to avoid obstacles and maintain smooth motion during the driving. A close look reveals that there are four sudden drops for $w_2 f_o$. Two short segments are located at frames 34–85, one long segment appears at frames 216–254 and the last one is at frames 374–394. These are due to the fact that there are no obstacles at the time and the road is empty. Consequently, the vehicle relies more on GPS trajectory following and we can see that ratio of $w_3 f_g$ increases. It means that the algorithm automatically falls back to rely on other available information when there are no LMs and no obstacles, which is desirable. The $w_4 f_c$ usually has a segment of being zero at frames 293–334 because the vehicle is make a 90° turn and actively set w_4 to be zero. Similar scenario happens at the beginning of Fig. 4(b). The sequence in Fig. 4(b) has high quality ALBs mostly and only needs to rely on VLBs 29% of the time. It is clear that $w_5 f_a$ remains high at frames 44–149 and 184–293 where the AV relies a lot on the ALB following. In addition, the reason that we have a segment of VLBs at frames 150–183 is due to a

¹<https://github.com/bli-tamu/LDRT>

parked vehicle occupying part of the road which is shown in Fig. 3(b).

VI. CONCLUSION AND FUTURE WORK

We reported our development of a new tightly coupled perception and planning framework to enable AVs to consider multiple conflicting goals simultaneously and generate human-compatible navigation trajectories. We built on our prior work to detect free space using camera-lidar sensor fusion and proposed ARF tests to determine whether the AV should simply follow ALBs or generate VLBs by taking into account vehicle kinodynamic constraints, obstacle avoidance, smooth motion, GPS trajectory following, respecting direction of LMs in a multi-objective optimization framework with dynamically adjustable weights for different road scenarios. We implemented our algorithm and the test results confirmed our design.

In the future, more physical experiments will be conducted and we will incorporate more functionalities such as velocity planning to make navigation decisions more human-like and human-compatible.

ACKNOWLEDGMENT

Thanks to C. Chou, S. Yeh, A. Kingery, and A. Angert for their inputs and contributions to the NetBot Laboratory, Texas A&M University.

REFERENCES

- [1] A. B. Hillel, R. Lerner, D. Levi, and G. Raz, "Recent progress in road and lane detection: a survey," *Machine vision and applications*, vol. 25, no. 3, pp. 727–745, 2014.
- [2] K. Abdulrahim and R. A. Salam, "Traffic surveillance: A review of vision based vehicle detection, recognition and tracking," *International journal of applied engineering research*, vol. 11, no. 1, pp. 713–726, 2016.
- [3] D. C. Andrade, F. Bueno, F. R. Franco, R. A. Silva, J. H. Z. Neme, E. Margraf, W. T. Omoto, F. A. Farinelli, A. M. Tusset, S. Okida *et al.*, "A novel strategy for road lane detection and tracking based on a vehicle's forward monocular camera," *IEEE Transactions on Intelligent Transportation Systems*, no. 99, pp. 1–11, 2018.
- [4] Y. Li and N. R. Gans, "Predictive ransac: Effective model fitting and tracking approach under heavy noise and outliers," *Computer Vision and Image Understanding*, vol. 161, pp. 99 – 113, 2017.
- [5] A. Petrovai, R. Danescu, and S. Nedevschi, "A stereovision based approach for detecting and tracking lane and forward obstacles on mobile devices," in *Intelligent Vehicles Symposium (IV), 2015 IEEE*. IEEE, 2015, pp. 634–641.
- [6] A. S. Huang, D. Moore, M. Antone, E. Olson, and S. Teller, "Finding multiple lanes in urban road networks with vision and lidar," *Autonomous Robots*, vol. 26, no. 2-3, pp. 103–122, 2009.
- [7] A. Joshi and M. R. James, "Generation of accurate lane-level maps from coarse prior maps and lidar," *IEEE Intelligent Transportation Systems Magazine*, vol. 7, no. 1, pp. 19–29, 2015.
- [8] Y. Kang, C. Roh, S.-B. Suh, and B. Song, "A lidar-based decision-making method for road boundary detection using multiple kalman filters," *IEEE Transactions on Industrial Electronics*, vol. 59, no. 11, pp. 4360–4368, 2012.
- [9] A. Mukhtar, L. Xia, and T. B. Tang, "Vehicle detection techniques for collision avoidance systems: A review," *IEEE Trans. Intelligent Transportation Systems*, vol. 16, no. 5, pp. 2318–2338, 2015.
- [10] A. Pandey, S. Pandey, and D. Parhi, "Mobile robot navigation and obstacle avoidance techniques: A review," *Int Rob Auto J*, vol. 2, no. 3, p. 00022, 2017.
- [11] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," in *Autonomous robot vehicles*. Springer, 1986, pp. 396–404.
- [12] D. Song, H. N. Lee, J. Yi, and A. Levandowski, "Vision-based motion planning for an autonomous motorcycle on ill-structured roads," *Autonomous Robots*, vol. 23, no. 3, pp. 197–212, 2007.
- [13] K. Souhila and A. Karim, "Optical flow based robot obstacle avoidance," *International Journal of Advanced Robotic Systems*, vol. 4, no. 1, p. 2, 2007.
- [14] A. Sgorbissa and R. Zaccaria, "Planning and obstacle avoidance in mobile robotics," *Robotics and Autonomous Systems*, vol. 60, no. 4, pp. 628–638, 2012.
- [15] D. González, J. Pérez, V. Milanés, and F. Nashashibi, "A review of motion planning techniques for automated vehicles," *IEEE Trans. Intelligent Transportation Systems*, vol. 17, no. 4, pp. 1135–1145, 2016.
- [16] J. Ziegler, P. Bender, M. Schreiber, H. Lategahn, T. Strauss, C. Stiller, T. Dang, U. Franke, N. Appenrodt, C. G. Keller *et al.*, "Making bertha drive-an autonomous journey on a historic route," *IEEE Intell. Transport. Syst. Mag.*, vol. 6, no. 2, pp. 8–20, 2014.
- [17] T. Gu, J. Atwood, C. Dong, J. M. Dolan, and J.-W. Lee, "Tunable and stable real-time trajectory planning for urban autonomous driving," in *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*. IEEE, 2015, pp. 250–256.
- [18] M. Likhachev, D. I. Ferguson, G. J. Gordon, A. Stentz, and S. Thrun, "Anytime dynamic a*: An anytime, replanning algorithm," in *ICAPS*, 2005, pp. 262–271.
- [19] K. Chu, M. Lee, and M. Sunwoo, "Local path planning for off-road autonomous driving with avoidance of static obstacles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 13, no. 4, pp. 1599–1616, 2012.
- [20] X. Li, Z. Sun, D. Cao, Z. He, and Q. Zhu, "Real-time trajectory planning for autonomous urban driving: Framework, algorithms, and verifications," *IEEE/ASME Transactions on Mechatronics*, vol. 21, no. 2, pp. 740–753, 2016.
- [21] H. Bai, S. Cai, N. Ye, D. Hsu, and W. S. Lee, "Intention-aware online pomdp planning for autonomous driving in a crowd," in *Robotics and Automation (ICRA), 2015 IEEE International Conference on*. IEEE, 2015, pp. 454–460.
- [22] L. Ma, J. Xue, K. Kawabata, J. Zhu, C. Ma, and N. Zheng, "Efficient sampling-based motion planning for on-road autonomous driving," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 4, pp. 1961–1976, 2015.
- [23] M. Haklay and P. Weber, "Openstreetmap: User-generated street maps," *Ieee Pervas Comput*, vol. 7, no. 4, pp. 12–18, 2008.
- [24] B. Li, D. Song, H. Li, A. Pike, and P. Carlson, "Lane marking quality assessment for autonomous driving," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Madrid, Spain, October, 1-5, 2018.
- [25] J. M. Keller, M. R. Gray, and J. A. Givens, "A fuzzy k-nearest neighbor algorithm," *IEEE transactions on systems, man, and cybernetics*, no. 4, pp. 580–585, 1985.
- [26] R. B. Rusu, Z. C. Marton, N. Blodow, M. Dolha, and M. Beetz, "Towards 3d point cloud based object maps for household environments," *Robotics and Autonomous Systems*, vol. 56, no. 11, pp. 927–941, 2008.
- [27] S. Gumhold, X. Wang, and R. S. MacLeod, "Feature extraction from point clouds," in *IMR*. Citeseer, 2001.
- [28] A. Y. Hata, F. S. Osorio, and D. F. Wolf, "Robust curb detection and vehicle localization in urban environments," in *Intelligent vehicles symposium proceedings, 2014 IEEE*. IEEE, 2014, pp. 1257–1262.
- [29] T. M. Howard, C. J. Green, A. Kelly, and D. Ferguson, "State space sampling of feasible motions for high-performance mobile robot navigation in complex environments," *Journal of Field Robotics*, vol. 25, no. 6-7, pp. 325–345, 2008.
- [30] T. Mörwald, J. Balzer, and M. Vincze, "Modeling connected regions in arbitrary planar point clouds by robust b-spline approximation," *Robotics and Autonomous Systems*, vol. 76, pp. 141–151, 2016.
- [31] T. Qin, P. Li, and S. Shen, "Vins-mono: A robust and versatile monocular visual-inertial state estimator," *IEEE Transactions on Robotics*, vol. 34, no. 4, pp. 1004–1020, 2018.
- [32] G. E. Farin and G. Farin, *Curves and surfaces for CAGD: a practical guide*. Morgan Kaufmann, 2002.
- [33] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The kitti dataset," *The International Journal of Robotics Research*, vol. 32, no. 11, pp. 1231–1237, 2013.