This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMECH.2015.2493980, IEEE/ASME Transactions on Mechatronics

Real-time trajectory planning for autonomous urban driving: framework, algorithms and verifications

Xiaohui Li*, Zhenping Sun*, Dongpu Cao†, Zhen He*, and Qi Zhu*
*the College of Mechatronic Engineering and Automation, National University of Defense Technology, Changsha, 410073, China
†the Department of Automotive Engineering, Cranfield University, Bedford MK43 0AL, U.K.

***Abstract*—The paper focuses on the real-time trajectory planning problem for autonomous vehicles driving in realistic urban environments. To solve the complex navigation problem, we adopt a hierarchical motion planning framework. Firstly, a rough reference path is extracted from the digital map using commands from the high-level behavioral planner. The conjugate gradient nonlinear optimization algorithm and the cubic B-spline curve are employed to smoothen and interpolate the reference path sequentially. To follow the refined reference path as well as handle both static and moving objects, the trajectory planning task is decoupled into lateral and longitudinal planning problems within the curvilinear coordinate framework. A rich set of kinematically-feasible path candidates are generated to deal with the dynamic traffic both deliberatively and reactively. In the meanwhile, the velocity profile generation is performed to improve driving safety and comfort. After that, the generated trajectories are carefully evaluated by an objective function, which combines behavioral decisions by reasoning about the traffic situations. The optimal collision-free, smooth and dynamically-feasible trajectory is selected and transformed into commands executed by the low-level lateral and longitudinal controllers. Field experiments have been carried out with our test autonomous vehicle on the realistic inner-city roads. The experimental results demonstrated capabilities and effectiveness of the proposed trajectory planning framework and algorithms to safely handle a variety of typical driving scenarios, such as static and moving objects avoidance, lane keeping and vehicle following, while respecting the traffic rules.**

***Index Terms*—Autonomous urban driving, real-time trajectory planning, static obstacles and moving objects avoidance.**

I. Introduction and State-of-the-Art

AUTONOMOUS driving technologies have great potentials to improve driving safety by reducing traffic accidents and fatalities caused by human errors, enhance driving efficiency by reducing traffic congestions, as well as provide mobility for people who are not able to drive [1], [2], [3]. Fully autonomous driving is generally identified as the ultimate goal of driver assistance systems in the future [4]. The past three decades have witnessed the significant development of autonomous driving technologies, which have drawn unprecedentedly considerable attention from both academia and industry. Tremendous research efforts have been contributed towards the ambitious goal of realizing fully autonomous driving on realistic roads [5], [6], [7].

Particularly in the last decade, with the advent of advances in sensors, computer technologies, artificial intelligence, autonomous driving related research topics have become extraordinarily active in both the robotics community and the automotive industry [8], [9], [10], [11], [12]. Among these well-known research projects, the DARPA Urban Challenge (DUC) in 2007 could be recognized as a turning point in demonstrating the potentials of autonomous vehicles driving in urban environments. After that, research groups in both universities and companies are continuously studying autonomous driving technologies to investigate necessary technologies for autonomous driving on public roads. Very recently, there has existed a number of impressive research projects on testing autonomous vehicles driving in urban and highway environments [13], [14], [15], [16], [17], [18], [19].

When autonomous ground vehicles (AGVs) advance towards the realistic urban road traffic, they are required to be capable of handling various complex maneuvers, such as lane keeping, vehicle following, lane changing, merging, avoiding both static and dynamic objects, interacting with other traffic participants while complying with the traffic rules. Developing a robotic vehicle to have these functionalities requires the systematic integration of state-of-the-art technologies in perception, localization, decision-making, motion planing and control. As one of these core technologies, motion planning plays a critical role in guaranteeing driving safety and comfort. In general, the challenges of developing a reliable and robust on-road motion planner lie in the following factors: 1) generating dynamically-feasible trajectories in realtime with limited onboard computational resources; 2) dealing with the unpredictably changing surrounding environments with limited sensing range and visibility as well as the uncertainty and noise in the perception and localization systems; 3) interacting with other traffic participants, such as vehicles, cyclists, and pedestrians.

Based upon the amount of previous research on this topic in the literature, this paper focuses on solving the trajectory planning problem in urban driving scenarios in a practical way instead of a theoretical way.

A. Related Work

In recent years, a significant amount of work has been dedicated to solving the motion planning problem for AGVs. These approaches can be roughly categorized into two classes. One focuses on computing collision-free paths using deterministic graph-search, such as Hybrid A* in [20] and [21], state-lattice approaches in [22] and [23], or random sampling-based algorithms, like Rapidly-exploring Random Tree (RRT)

[24]. Most of these graph-search approaches are capable of computing long-term collision-free paths in cluttered environments and preventing the vehicle from getting stuck into local minima. These powerful path-based motion planners are suitable for AGVs navigating in complex unknown environments at low speeds. However, most of these search-based algorithms assume that environments are static in each plan cycle. In addition, the generated paths are often comprised of concatenated short-term precomputed motion primitives without considering velocity planning. As a result, they may easily result in stop-and-redirect motions. In addition, they are usually too computationally demanding to run in real time or react to typical dynamic traffic situation in urban environments. Therefore, these graph-search algorithms are not suitable for vehicles driving in urban and highway environments with moving objects. When vehicles drive in structured environments, the feasible paths are often strictly constrained by the geometry of roads, which significantly reduces the solution space of drivable paths. However, due to the presence of dynamic traffic participants, the motion planner is required to explicitly take both lateral and longitudinal movements into account.

There has been substantial research on real-time local trajectory generation for AGVs driving in urban environments. A well-known and efficient strategy follows a discrete optimization scheme. A well-known and efficient strategy follows a discrete optimization scheme. To generate dynamically-feasible trajectory candidates, a model-predictive motion planner is proposed in [11]. The control inputs are assumed to be curvature polynomials, which significantly reduce the solution space and guarantee the expressiveness of the generated trajectories as well. Using the same idea, the approach in [25] generates a finite set of nudges and swerves with different lateral offsets shifting from a reference path. A similar approach is adopted in [26] considering both lateral and longitudinal movements within the street-relative coordinate . Considering both vehicle motion model and the associated closed-loop feedback control laws, a sampling-based motion planner is proposed in [27], which ensures that the generated trajectories could be accurately tracked. A similar strategy is also referred in [24].

Based on the Boss work in DARPA Urban Challenge [11], a extended trajectory planner is proposed for highway automated driving using spatiotemporal lattice conforming to the road geometry [28]. Graphic Processor Units (GPUs) are utilized to generate thousands of long-term trajectory candidates in real-time. The motion planner demonstrated its ability to handle imminent situations in structured environments. However, always simultaneously considering both spatial and temporal dimensions with high-resolution makes the search space prohibitively large. Additionally, it is difficult to ensure the temporal consistency between consecutive replans. Besides, the approach assumes that the other traffic participants' motions could be precisely predicted within a finite horizon. Based on his work, to reduce the computation time while retaining the performance of the spatiotemporal motion planner, a hierarchical motion planner is proposed to address the complex motion planning problems in both urban and highway driving within one framework [29], [30], [31], [32].

Instead of applying the discrete optimization scheme, very recent work in [33] formulated the trajectory planning problem as a nonlinear optimization problem with a number of inequality constraints. Extracting a reference corridor using the digital map, vision-based localization information, and high-level decision process, the trajectory planner explicitly considered hard constraints imposed by both static obstacle and dynamic traffic participants, as well as the physical constraints, like maximal steering curvature. The sequential quadratic programming (SQP) method is applied to solve the computationally demanding optimization problem in the continuous state space.

Most of the previous work does not take uncertainties of the prediction and interactions with other traffic participants into account. In [34] and [35], to assess the safety of the planned paths, uncertainties originating from the measurements and possible behaviors of other traffic participants are considered in a stochastic way. Besides, the interaction with other traffic participants, as well as the limitation of driving maneuvers due to the road geometry are also explicitly taken into account. In [36], a compact representation for the on-road environment, the dynamic probability drivability map (DPDM), is presented for predictive lane change and merge driver assistance during highway and urban driving environments. The uncertainties of both ego-vehicle and other participants is predicted based on Gaussian propagation in [37].

B. Contributions and Novelties

Based upon the aforementioned previous work, the main contribution of this paper is the introduction of a practical real-time trajectory generation framework and algorithms for AGVs driving in realistic urban environments. To handle various dynamic urban traffic situations both deliberatively and reactively, we adopt a ”divide-and-conquer” strategy. More precisely, a hierarchical framework is employed. The high-level behavioral planner is responsible for reasoning about complex dynamic traffic situations and making deliberate discrete maneuver decisions, such as lane following, lane changing, vehicle following, overtaking a slow-moving vehicles and so forth. Using the behavioral decision, the trajectory generation algorithm assumes responsibility for generating dynamically-feasible and collision-free trajectories, which could be easily tracked by low-level tracking controllers. This paper focuses on investigating the real-time trajectory generation algorithm.

The rough reference path is extracted from the digital map using the lane-level accurate localization information via the LiDAR-based localization method, which is similar to the approaches proposed in [38] and [39]. To improve driving comfort and reduce the control efforts, the conjugate gradient nonlinear optimization algorithm and cubic B-spline algorithms are employed to smoothen and interpolate the reference path sequentially. In realistic urban environments, AGVs are required to always pay attention to the unpredictably changing traffic situations, generating dynamically-feasible trajectories while avoiding both static and moving objects in

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMECH.2015.2493980, IEEE/ASME Transactions on Mechatronics

real time. To achieve this, the trajectory planner is performed to handle the dynamic traffic situations in a reactive manner by explicitly considering constraints imposed by the vehicle kinematics and dynamics, as well as environments. Using the deliberative decisions generated by the behavioral planner, the trajectory planner is able to focus on the solution space, where the optimal solution is mostly like to exist. Besides, aggressive actions could also be avoided in most non-imminent traffic situations. Then, based on the curvilinear coordinate framework, the trajectory planning task is decoupled into spatial path planning and velocity planning subtasks. Instead of using the optimization scheme to produce a sole optimal trajectory in each replan cycle, the trajectory planner is capable of generating a rich set of sub-optimal candidates, which could efficiently overcome the noise in the perception and localization systems. Besides, it ensures that the vehicle is able to safely stop in imminent situations. To guarantee driving safety and improve driving comfort, an objective function with a set of cost terms with definite physical meanings is carefully designed to select the best trajectory candidate for execution.

The remainder of the paper is organized as follows. Section II describes the system framework. The trajectory planning algorithms is introduced in Section III. Section IV presents the trajectory evaluation and optimization approach. The experimental results in realistic urban scenarios are discussed in detail in Section V. Section VI draws conclusions and suggests future work.

II. System Framework

An overview of the software system architecture for the AGV is outlined in Fig. 1. The system is comprised of a variety of modules, such as sensors, a digital map, task files, the perception and localization systems, task planner, route planner, behavioral planner, trajectory planner, and tracking controllers, low-level actuators control and human machine interface (HMI). Each module communicates with other modules via an specific publish/subscribe message passing protocol based on the Inter Process Communication Toolkit.

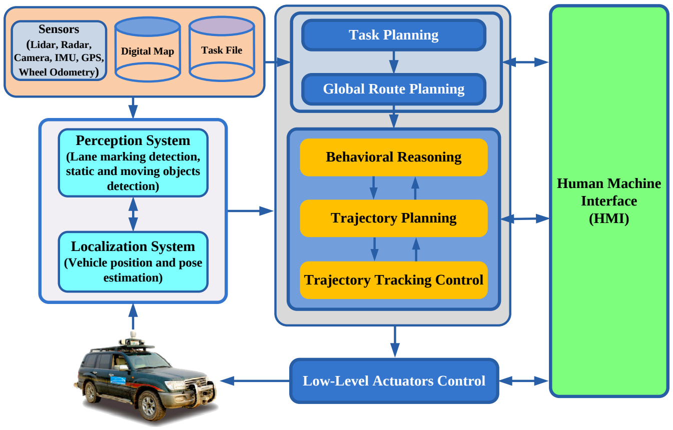

Sensors (Lidar, Radar, Camera, IMU, GPS, Wheel Odometry)
Digital Map
Task File
Task Planning
Global Route Planning
Perception System (Lane marking detection, static and moving objects detection)
Localization System (Vehicle position and pose estimation)
Behavioral Reasoning
Trajectory Planning
Trajectory Tracking Control
Human Machine Interface (HMI)
Low-Level Actuators Control

Fig. 1. The software system architecture for the AGV.

Sensors, such as LiDARs, radars, cameras, provide the sensing information of surrounding environments in real time. Additionally, other sensors, like the GPS combined with inertial measurement unit (IMU) and the wheel encoders are used to obtain the vehicle's rough localization information. The sensing information is mainly applied for two purposes: one is for the perception system, such as detecting lane markings, traffic lights, as well as static and dynamic objects (e.g. pedestrians, cyclists and the other vehicles); the other is to realize accurate and robust localization. To achieve lane-level accurate and reliable localization, many researchers take advantage of mapping and localization technologies using GPS, IMU, combined with online 3D points data from laser scanners [38], [40], [41] or camera vision data [42], [43], [44].

In this paper, online sensing 3D points data from the HDL-64E Velodyne LiDAR combining with a high-resolution 3D map data recorded by manual driving have been applied to estimate the vehicle's position and pose in real time. Based on this localization approach, the rich information of a prior digital map could be exploited. In practice, we use a manually-constructed detailed digital map, which provides abundant traffic information, such as lanes information (e.g. the position, the number, and the topological relations) and traffic regulations (e.g. speed limits). The details of the localization and map-constructed approach are not the main focus of this paper.

The results of perception and localization systems are illustrated in Fig. 2, the mother map is created offline using 3D points data of the LiDAR. The cells with magenta color are online detected obstacles, including both dynamic and static obstacles. The rectangular block represents the moving object with arrows indicating its moving direction. The while lines indicate the lanes based on the digital map. In practice, we find that the localization method plays a critical role in guaranteeing our autonomous vehicle safely driving in the dynamic urban environments during the day and night, with sun or rain.

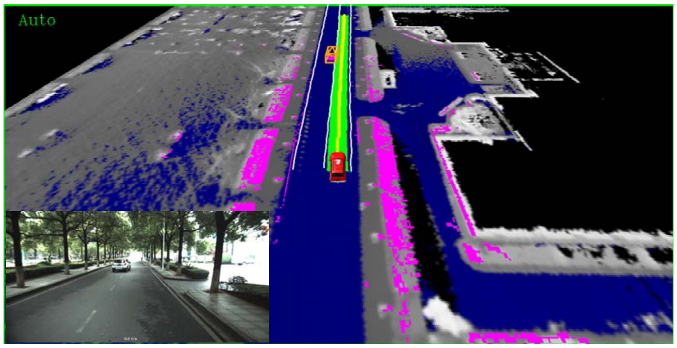


Fig. 2. Illustration of perception and localization systems.

Using the digital map and the accurate localization information, the global route planner could compute the fastest reference lane either online or offline while complying with the task planning results. The objective of the behavioral planner is to make informed decisions relying upon the online sensing information and the planned reference lane. This paper focuses on the trajectory planning module, which plays a critical role in guaranteeing the driving safety and comfort.

As the flow chart of the trajectory planning algorithms shown in Fig. 3, the reference path is derived first. Instead of using the centerline of the reference lane directly, we employ a nonlinear optimization approach to smoothen the centerline, then interpolate it using the cubic B-spline curve. In this way, a smooth reference path (or baseline) could be obtained. As referred in [13], to mimic human-like driving behaviors in urban environments, the vehicle trajectory could be naturally decoupled into the lateral movement and the longitudinal movement. In light of this, the trajectory planning task is decomposed into spatial path planning and velocity planning subtasks based on the curvilinear coordinate instead of the Cartesian coordinate. A rich set of trajectory candidates are generated based on the reference path. After that, collision test is performed to trim the trajectory candidates colliding with obstacles. Then, a situation-aware trajectory evaluation method is applied to derive the optimal trajectory, which is transformed into low-level actuator commands by the trajectory tracking controller.

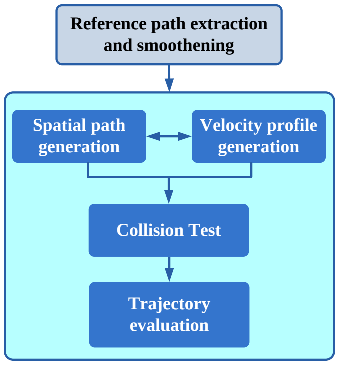

Reference path extraction and smoothening
Spatial path generation
Velocity profile generation
Collision Test
Trajectory evaluation

Fig. 3. The flow chart for trajectory planning.

III. System Algorithms

Note that the proposed algorithm mainly focuses on addressing the navigation problem in structured environments with lanes. It is not able to handle complex navigation problems with dense obstacles, such as free-zone navigation problems. In these situations, the graph-search algorithm can be employed to compute a rough collision-free path firstly to deal with the complex environmental constraints. Once the collision-free reference path is obtained, the proposed trajectory generation framework and algorithms could be used to generate smoother and dynamically-feasible trajectory to be easily tracked by low-level controllers.

A. Reference path smoothing and interpolation

In structured environments, the reference path could be directly calculated using the right boundary and left boundary of the desired lane. In practice, we found that the curvature of the calculated centerline is usually neither continuous nor smooth. It cannot be easily tracked by the low-level tracking controllers. Tracking the discontinuous centerline can easily result in control overshoots and oscillations. In addition, when the vehicle is negotiating a curvy lane or negotiating a tight turn, following the centerline requires more control efforts and results in larger lateral accelerations. To overcome these issues and improve the driving comfort, the conjugate gradient optimization method is employed to smoothen the reference path in a continuous space. A similar method was referred in [20] and in [30]. To reduce the computational burden and achieve real-time performance, the obstacle-avoidance constraints are not explicitly considered during solving the optimization problem. Given a set of discrete waypoints of the initial reference path as seed vertices $\mathbf{x}_i = (x_i, y_i), (i = 1, ..., N)$, which could be slightly tuned to minimize the following objective function:

$$\arg\min\{\omega_1 \sum_{i=1}^{N-1} (\Delta\mathbf{x}_{i+1} - \Delta\mathbf{x}_i)^2 + \omega_2 \sum_{i=1}^{N-1} (\frac{\Delta\theta_i}{|\Delta\mathbf{x}_i|})^2 + f(\mathbf{x}_i)\} \tag{1}$$

where $\Delta\mathbf{x}_i = \mathbf{x}_i - \mathbf{x}_{i-1}, (i >= 2)$ is the displacement vector at the vertex $\mathbf{x}_i = (x_i, y_i)$; $\Delta\theta = \arccos(\frac{\Delta\mathbf{x}_{i+1}\Delta\mathbf{x}_i}{|\Delta\mathbf{x}_{i+1}||\Delta\mathbf{x}_i|})$ is the vector direction change at the vertex $\mathbf{x}_i$;ω_1, ω_2 are the corresponding weights, respectively; $f(\mathbf{x}_i)$ is a differentiable nonlinear function. Its value is around zero when the point $\mathbf{x}_i$ is within the boundary of the desired lane, but goes to infinity when the point is out of the boundary.

The first term minimizes the the vector change between the adjacent vectors, and the second term minimizes cumulative curvature along the path. Both of them are performed to improve the smoothness of the reference path. The third cost term guarantees that the optimized reference path meets the constraints imposed by the road boundary conditions. The road boundary conditions could be obtained from the digital map. The weights of the cost terms and the iteration have impacts on the smoothness of the path. Like the other nonlinear optimization methods, the conjugate gradient algorithm only returns a local optimal solution. In practice, ω_1 is set to 10 and ω_2 is set to 1, which are empirically obtained. Additionally, The maximal iteration is limited to 400 to meet the real-time requirement. In future, the machine learning techniques will be employed to tune these parameters adaptively.

The number of vertices are set to be same before and after smoothing process. In practice, it is found that the distance between the vertices are too large at times. To obtain dense waypoints, we use the cubic B-spline for interpolation. The discrete points of the optimized reference path could be employed as the control points. Using the discrete control points $P_i(i = 0, 1, \cdots, n)$, the piece-wise cubic B-spline curve could be written as

$$C(u) = \sum_{i=0}^{n} P_i G_{i,3}(u) =$$

$$\frac{1}{6}\begin{bmatrix} u^3 & u^2 & u & 1 \end{bmatrix} \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 0 & 3 & 0 \\ 1 & 4 & 1 & 0 \end{bmatrix} \begin{bmatrix} P_i \\ P_{i+1} \\ P_{i+2} \\ P_{i+3} \end{bmatrix} u \in [0, 1] \tag{2}$$

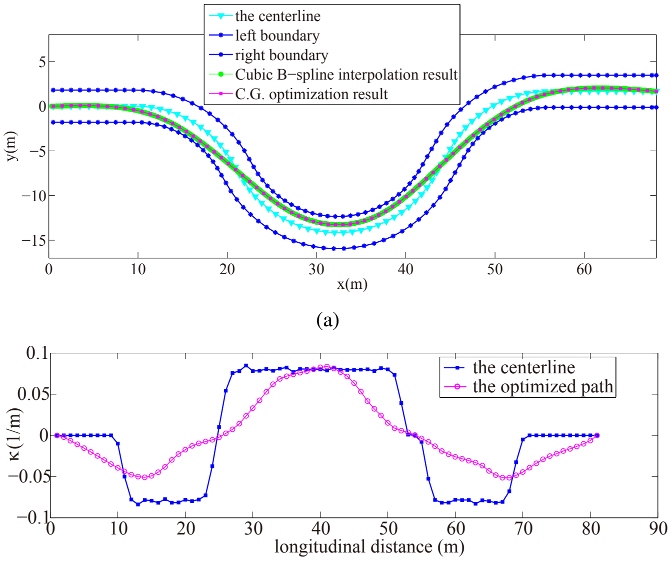
where $G_{i,3}(u)$ is the B-spline basic function.

Since the path generated using cubic B-spline approach could be written as a closed-form expression. Four dimensional states (x, y, θ, κ) of each point along the smooth path could be analytically calculated. The tangent angle $\theta(i)$ and the curvature $\kappa(i)$ of the waypoints along the path are defined as

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMECH.2015.2493980, IEEE/ASME Transactions on Mechatronics

$$\theta_i = \tan^{-1}(\frac{\dot{x}}{\dot{y}}) \tag{3}$$

$$\kappa(t) = \frac{\dot{x}\ddot{y} - \ddot{x}\dot{y}}{(\dot{x}^2 + \dot{y}^2)^{3/2}} \tag{4}$$



(a)

(b)

Fig. 4. The conjugate gradient optimization algorithm is employed to smooth the centerline. Then dense waypoints are obtained using cubic B-spline interpolating method. (a) the original centerline and the optimized reference path, (b) the curvature of the original path and the optimized reference path

As illustrated in Fig. 4 (a) and (b), the reference path (green line) with equidistant waypoints are obtained using smoothing and interpolating algorithms mentioned above. It can be seen from Fig. 4 (b), the curvature of the original path is discontinuous and changes sharply. After optimization, the curvature becomes much smoother. Meanwhile, the optimized reference path is able to be restricted within the road boundary. To achieve real-time performance, environmental constraints are not explicitly taken into account to solve the nonlinear optimization problem. Both environmental constraints and vehicle constraints will be explicitly considered by the trajectory planner, which is investigated in the subsequent subsections.

B. Spatial path generation based on the curvilinear coordinate framework

Learning from human drivers' behaviors in structured environments, it is necessary for the trajectory planner to generate trajectories which are aligned with the road geometry. In this way, the AGV is able to drive much safer and interact with other traffic participants more friendly. Besides, the solution space could be significantly reduced as well. To achieve this, the road geometry model could be represented using the curvilinear coordinate framework instead of using the Cartesian coordinate framework. Another benefit is that the lateral and longitudinal movements of vehicles can be separately handled. Therefore, we decompose the trajectory planning task into spatial path generation and velocity profile generation subtasks.

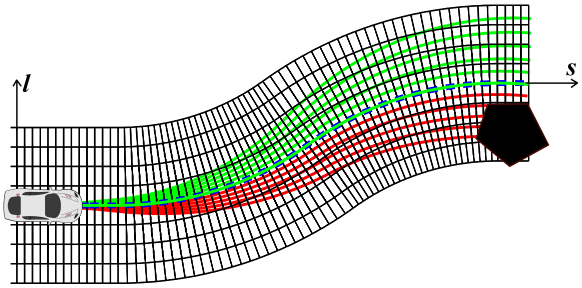


Fig. 5. A rich set of path candidates are generated to follow the centerline while avoiding static obstacles.

After applying the smoothing and interpolation algorithms mentioned above, the refined reference path is taken as the base frame. To follow the base frame while avoiding obstacles, [11] proposed a model-predictive motion planner, which sampled a finite set of terminal states laterally shifting from the base frame firstly. Then the path generation problem was formulated as a two-point boundary value problem [45]. To reduce solution space while respecting vehicle motion constraints, the steering control input space is parameterized using curvature polynomials. Though the approach considers terminal states' constraints, it ignores the path geometry during the approaching process. As a consequence, it is not able to generate long-term trajectories to be aligned with a curvy road, such as a curvy right-angle turn. As we mentioned before, it is critical to generate trajectories being aligned with the road geometry in structured environments. To solve the problem, we refer to the path generation strategies mentioned in [25], [26], [46]. As shown in Fig. 5, the sampling-based motion planning strategy is employed to generate a set of path candidates with different lateral offsets shifting from the base frame. In this way, the vehicle is capable of avoiding static obstacles as well as being aligned with the geometry of the base frame. The details are described as follows.

Based on the curvilinear coordinate framework, the sampling terminal states could be represented using two parameters, one is the longitudinal distance s_f along the base frame and the other is the lateral deviation l_f from the base frame. The longitudinal distance s_f determines how fast the vehicle is regulated to be aligned with the base frame. In practice, it could be adaptively tuned according to the vehicle velocity without violating the maximal lateral acceleration as well. The lateral offset l_0 is the perpendicular distance between the vehicle current position and the corresponding point (the nearest point) along reference path.

According to the differential flatness control theory referred in [47], polynomial spline can be applied to smooth the transition from the current lateral offset l_0 to the sampling terminal offset l_f. To achieve smooth transition from the initial lateral offset to the sampling terminal offset, we take advantage of the cubic polynomial, which is similar to the algorithm referred in [46].

$$l(s) = c_0 + c_1 s + c_2 s^2 + c_3 s^3 \tag{5}$$

The first derivative and second derivate of the lateral deviation $l(s)$ with respect to arc length s could be obtained as

$$\frac{dl}{ds} = c_1 + 2c_2 s + 3c_3 s^2 \tag{6}$$

$$\frac{d^2 l}{ds^2} = 2c_2 + 6c_3 \tag{7}$$

Based on the initial boundary constraints, we can obtain

$$l(s_0) = l_0, l(s_f) = l_f \tag{8}$$

Besides, $\theta(s)$ is the heading angle difference between the vehicle and the base frame. It could be derived by the first derivative of the lateral deviation $l(s)$ with respect to the arc length s.

$$\theta(s) = \arctan(\frac{dl}{ds}) \tag{9}$$

To ensure the generate paths to be aligned with the base frame, the heading angle of the generate path at the preview distance s_f is set to be same as that of the base frame. Since the vehicle's initial heading angle $\theta(s_0)$ may be different from the tangent direction of the base frame at the current position, it can be written as

$$\theta(s_0) = \theta_0, \theta(s_f) = 0 \tag{10}$$

θ_0 is the initial heading difference between the vehicle and the base frame. Since the number of unknown parameters is equal to that of equations, the closed-form solution of the unknown parameters $\{c_0, c_1, c_2, c_3\}$ could be obtained.

Now, in order to represent the path in the Cartesian coordinate, we use the relationship between the Cartesian coordinate and the curvilinear coordinate [48]. Since the curvature of the base frame could be calculated using (4), then we calculate the curvature of the generated path κ_b accordingly.

$$\kappa(s) = \frac{1}{Q}(\kappa_b + \frac{(1 - l\kappa_b)(d^2 l/ds^2) + \kappa_b (dl/ds)^2}{Q^2}) \tag{11}$$

with

$$Q = \sqrt{(dl/ds)^2 + (1 - l\kappa_b)^2}$$

Note that the path length s of the generate path is measured along the base frame rather than the generate path itself. To avoid singularity, $(1 - l\kappa_b)$ is required to be greater than zero. In other words, using this method is not able to generate feasible paths when the lateral offset l_s is greaterer than the radius of the corresponding point along the base frame. The curvature state is the invariant intermediate variable between the curvilinear coordinate framework and the Cartesian coordinate framework. The spatial path in the Cartesian coordinate could be numerically solved using the vehicle kinematic differential equations (12). For s is the path length along the base frame, the coefficient Q is taken into account in the following equations.

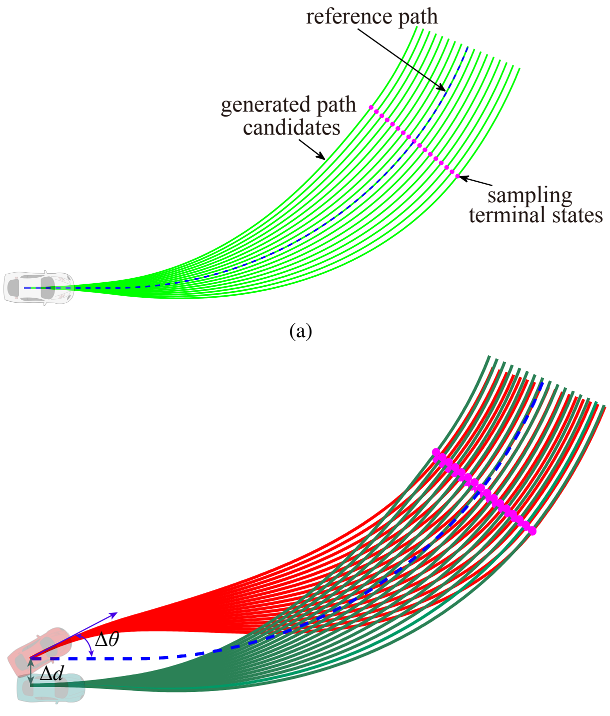
$$\dot{x}(s) = Q\cos(\theta(s))$$

$$\dot{y}(s) = Q\sin(\theta(s))$$

$$\dot{\theta}(s) = Q\kappa(s))$$

$$\kappa(s) = max\{\kappa_{\min}, \min\{\kappa(s), \kappa_{\max}\}\} \tag{12}$$

where $\kappa_{\min}$ and $\kappa_{\max}$ indicate the mechanical constraint imposed by the vehicle's steering actuator. When vehicles drive at a high speed, it is necessary to generate long-term paths to avoid reactive maneuvers. To achieve this, $d^2 l/ds^2 = 0$ and $dl/ds = 0$, when $(s >= s_f)$. In this way, the generated paths are aligned with the reference path and keep a fixed lateral offset l_f after they arrive at the sampling terminal states. As shown in Fig. 6(a), the spatial path candidates are generated by connecting the initial state with the sampling terminal states. Figure 6(b) shows that when the vehicle has the lateral deviation (curves with dark green color) or a certain heading deviation (curves with red color) from the base frame at the initial position. The proposed path generation approach is capable of generating spatial path candidates, which could regulate the vehicle to be aligned with the base frame gradually.



(a)

(b)

Fig. 6. Path candidates generation results. (a) $l_0 = 0$ and $\theta_0 = 0$,(b) $l_0 = \Delta d$ and $\theta_0 = \Delta\theta$

Since the closed-form solution could be obtained, the aforementioned spatial path generation algorithm is computationally efficient. In practice, to enhance the probability of the presence of the collision-free path in complex environments, we densely sample the longitudinal distance s_f and the lateral deviation l_f to generate a rich set of path candidates.

C. Velocity profile generation

The path planning method proposed in [8] and [46] did not explicitly consider the velocity planning problem. However, velocity planning has a great influence on driving safety and comfort, especially for vehicles driving on realistic urban scenarios with other dynamic traffic participants. Therefore, the velocity profile generation is required to be carefully assigned on each point along the generated path. Besides, constraints, such as longitudinal and lateral acceleration limits, speed limits by traffic rules, are also required to be explicitly taken into account. Considering these constraints could considerably reduces the solution space of the velocity planning as well as allow the velocity planner to concentrate on the space where the optimal solution is most likely to exist. Therefore, the velocity generation is performed after the spatial path generation.

To obtain a closed-form solution, in this paper we develop a hierarchical velocity profile generation strategy. Firstly the maximal allowed speed is carefully determined. Secondly, the trapezoidal velocity profile is employed to generate the integral speed curve. At last, to improve comfort of the longitudinal control, we employ the polynomial splines, which are utilized to smoothen the linear velocity profile and generate the acceleration-continuous velocity curve.

To guarantee driving safety and improve its comfort, a variety of constraints are explicitly taken into account to determine the maximal allowed velocity.

(i) The maximal allowed velocity V_{limit1}

It has to respect constraints imposed by the high-level behavioral planner, traffic regulations and so forth.

(ii) The maximal allowed lateral acceleration Acc_{MaxLat}

$$V_{\text{limit2}}^2\,|\kappa(s)| \le Acc_{\text{MaxLat}}$$

where $\kappa(s)$ is the path curvature value. To prevent the lateral force acting on tires from entering into the nonlinear or saturated zone, alleviate sideslip effects as well as maintain lateral stability, the Acc_{MaxLat} is required to be carefully restricted considering vehicle lateral dynamics.

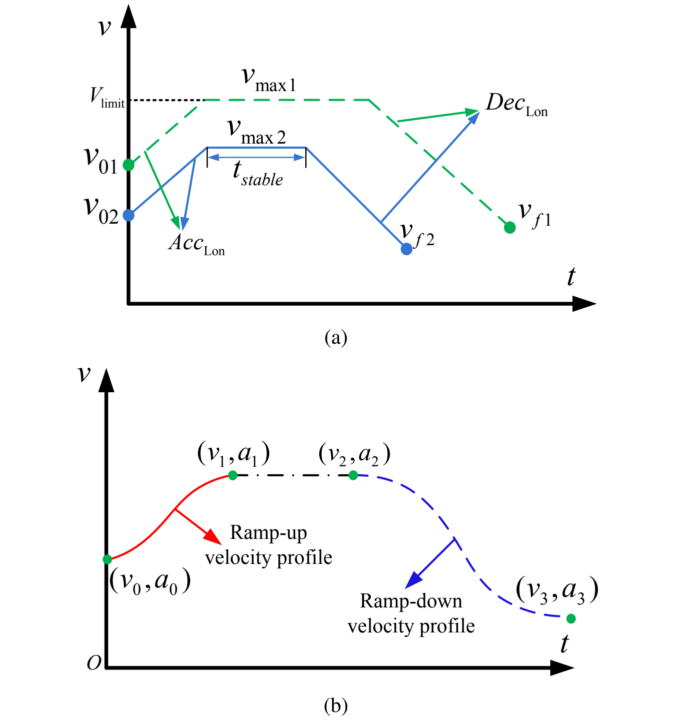
(iii) The longitudinal acceleration limit Acc_{MaxLon} and deceleration limit Dec_{MaxLon}

$$\frac{V_{\text{limit3}}^2 - v_0^2}{2Acc_{\text{MaxLon}}} + \frac{V_{\text{limit3}}^2 - v_f^2}{2Dec_{\text{MaxLon}}} + D_{\text{safe}} = S$$

where v_0 is the initial velocity, v_f is the terminal speed, and S is the path length. The terminal speed v_f is determined by the high-level behavioral planner, which reasons about traffic conditions and rules. Additionally, we also consider the safety distance D_{safe}, which involves the response time delay t_{delay} and imminent braking distance $D_{imminent}$. Using the well-known constant-time-gap law, the safety distance D_{safe} can be represented as a function of v_f.

$$D_{\text{safe}} = t_{\text{delay}} v_0 + D_{\text{imminent}}(v_f)$$

As shown in Fig. 7, the trapezoidal profile is employed for the velocity profile generation. To obtain a closed-form solution, we assume that ramp-up and ramp-down courses are with constant acceleration and deceleration values. According to our practical experience, to avoid oscillations of the longitudinal control, the velocity is set to keep at the steady speed v_{stable} for a certain period of time longer than t_{stable}. Therefore, v_{stable} does not necessarily reach the maximal allowed speed V_{limit}, such as the blue solid velocity curve shown in Fig. 7. Given the initial velocity v_0, terminal velocity v_f, the path length S, the maximal allowed velocity V_{limit}, and user-specified parameters $\{Acc_{\text{lon}}, Dec_{\text{lon}}, t_{\text{stable}}\}$, the closed-form solution of the velocity profile can be derived. At the same time, these parameters could also be parameterized to generate a rich set of velocity profile candidates.



(a)

(b)

Fig. 7. Velocity profile generation results. (a) trapezoidal velocity profile generation,(b) velocity profile smoothing

For comfort purpose, we further smoothen the trapezoidal velocity profiles to guarantee the continuity of the acceleration. Inspired by the parametric velocity profile generation method in [30], the velocity profile is parameterized using cubic polynomials. In contrast to the approach proposed in [30] and [26], we adopt the trapezoidal velocity profile and smooth the ramp-up and ramp-down processes separately.

$$v(t) = v_0 + at + bt^2 + ct^3 \tag{13}$$

According to (13), the acceleration $acc(t)$ and path length $s(t)$ can be derived through first derivative and integration of the velocity expression respectively. Given the initial velocity v_0 and initial acceleration a_0, the terminal velocity v_f and the terminal acceleration a_f, and path length s_f, the unknown parameters $\{a, b, c, t_f\}$ in (14) could be analytically solved via the following equations.

$$\begin{gathered} acc(0) = a = a_0 \\ v(t_f) = v_0 + at_f + bt_f^2 + ct_3^2 = v_f \\ acc(t_f) = a + 2bt_f + 3ct_f^2 = a_f \\ s(t_f) = v_0 t_f + \frac{1}{2}at_f^2 + \frac{1}{3}bt_f^3 + \frac{1}{4}ct_f^4 = s_f \end{gathered} \tag{14}$$

The terminal speed v_f is determined using the behavioral decision, which reasons about traffic situations. For instance, when the behavioral decision is vehicle following, the terminal speed will be the speed of the nearest front vehicle in our current lane (if its speed does not exceed the maximal allowed speed). In other words, the speed planner uses a speed control policy similar to ACC (Adaptive Cruise Control). When the behavioral decision is stop at the stop-line, the terminal velocity will be zero.

As shown in Fig. 7, we solve the acceleration-continuous S-shaped ramp-up velocity profile (the red solid line) and ramp-down velocity profile (the blue dash line) separately. Combining the spatial path with the corresponding velocity profile, the trajectories are produced. Note that the acceleration of the cubic polynomial speed profile will exceed the acceleration of the linear velocity profile at some points. In practice, the acceleration and deceleration values of the ramp-up and ramp-down profiles of the trapezoidal profile are set to be very conservative values to alleviate the negative effects. As referred in [49], these parameters could be tuned by learning from human demonstrations to achieve human-like driving.

IV. Trajectory Evaluation and Optimization

After the trajectory candidates have been generated, collision test is performed to shorten or cut off the trajectories colliding with static and moving obstacles. To achieve this, the local trajectory planning map is represented as an occupancy grid map using the online sensing data. As shown in Fig. 8, the rectangular shape of the vehicle is approximated by a set of circles [33]. Using this approach, the computational complexity of the collision test can be significantly reduced. To meet the safe requirements, the distance between obstacles and the centers of circles are required to be greater than the circle radius.

Considering the noise in the localization, perception systems, the radius could be enhanced to add safety margin between the vehicle and the obstacles. To assess the safety of the generated trajectories, both static obstacles and moving objects are required to be considered. To avoid collision with moving objects, it is necessary to predict the movements of the other moving objects. Due to the imperfect perception and prediction information, it is impossible to accurately predict the movement of the other objects within a large prediction horizon. In practice, the prediction horizon could be tuned with the vehicle speed and surrounding driving environments. Besides, the interaction with other traffic participants, as well as the limitation of driving maneuvers due to the road geometry can also be considered, as referred in [34] and [35].

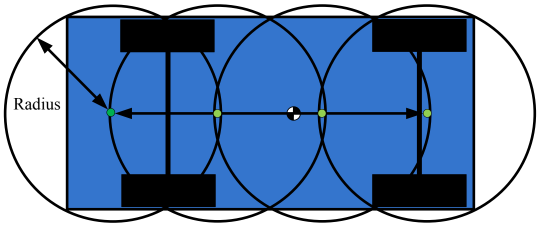


Fig. 8. The vehicle shape is approximated using four circles for collision test.

After the collision test is performed, the remaining trajectories are further evaluated and optimized via a carefully designed objective function comprised of four weighted cost terms. It is written as

$$i^* = \underset{i=1}{\overset{N}{argmin}}(\omega_p J_p(\tau_i) + \omega_s J_s(\tau_i) + \omega_d J_d(\tau_i) + \omega_c J_c(\tau_i)) \tag{15}$$

where $J\{p, s, d, c\}$ are cost terms, $\omega\{p, s, d, c\}$ are the corresponding weighted factors, and $\tau_i (i = 1, ..., N)$ is the ith trajectory candidate. The cost value is normalized to be within [0, 1]. The cost term J_p reflects the preference of the longer trajectory, which could prevent over-reactive actions resulting from too short-sighted trajectories.

$$J_p = (S_{\max} - S(\tau_i))/S_{\max} \tag{16}$$

where $S(\tau_i)$ is the path length of the trajectory candidate τ_i along the base frame, and $S_{\max}$ is the path length threshold. If the path length is greater than $S_{\max}$, J_p is set to be zero.

To enhance the comfort of the lateral movements, smoothness criterion is considered by the cost term J_s, which is the integration of curvature along the trajectory candidate.

$$J_s = \frac{1}{s_f}\int_0^{s_f} \frac{\|\kappa(\tau_i(s))\|}{\kappa_{\max}} ds \tag{17}$$

To ensure that the vehicle could follow the guidance of the base frame, the cost term J_d penalizes trajectory which deviates from the base frame.

$$J_d = \frac{1}{s_f}\int_0^{s_f} \frac{|D(\tau_i(s))|}{D_{\max}} ds \tag{18}$$

where $D(\tau_i(s))$ is the lateral deviation distance and D_{max} is the deviation distance threshold.

The cost term J_c penalizes the path inconsistency between consecutive replans. The inconsistency between adjacent replans can easily result in oscillations, overshoots, even instability of the control system. To ensure consistency between replans, we minimize the sampling terminal lateral offset of chosen path between consecutive replans.

$$J_c = \frac{|l(\tau_i) - l_p(\tau_{i^*})|}{l_{\max}} \tag{19}$$

where $l(\tau_i)$ is the terminal lateral offset of the current trajectory, $l_p(\tau_{i^*})$ is the terminal lateral offset of the previous selected optimal trajectory.

Evaluating the generated paths with considering four cost terms mentioned above will inevitably yield the issue to make a compromise among safety, smoothness, accuracy, and consistency. In practice, the weight factors of the four cost terms are empirically tuned based on the behavioral decision results and prior information of the digital map. For our future research, machine learning techniques will be employed to adaptively tune these cost terms.

As shown in Fig. 9, the occupancy grid map is adopted to represent the environment based upon the sensing information from the perception system. The grids with red color are static obstacles. The purple curve indicates the reference path (base frame). A rich set of dynamically-feasible trajectory candidates are generated being aligned with the reference path. As the bottom-right color bar illustrated, the color of the trajectory

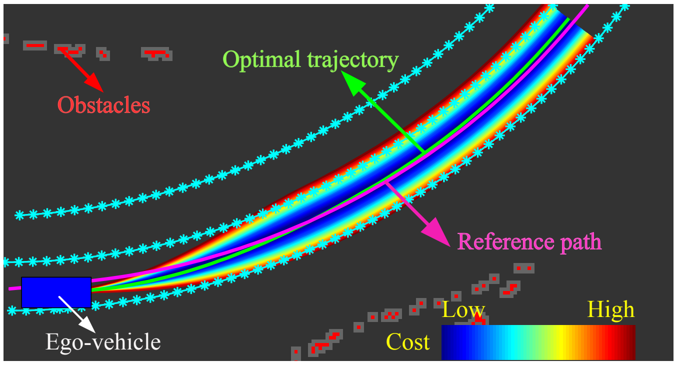


Fig. 9. Trajectory candidates evaluation.

indicates its cost value. The green curve is the optimal trajectory after evaluated by the aforementioned objective function. It will be transformed into actuators' commands by the trajectory tracking controller.

V. Urban Road Testing and Verifications

To verify the applicability and efficiency of the proposed local trajectory planning framework, we conducted field experiments on our test autonomous vehicle on the realistic urban roads. As shown in Fig. 10, our test autonomous vehicle is a modified commercial SUV, equipped with a variety of sensors, such as LiDARs, cameras, and radars. The raw data from these sensors is fused in the perception system, which provides the surrounding environments in realtime. The steering, brake, throttle, shifter actuators, as well as horn and turn lights of the vehicle have been modified to be drive-by-wire. All of these actuators could be controlled by the commands from the computer running control algorithms using the CAN Bus.

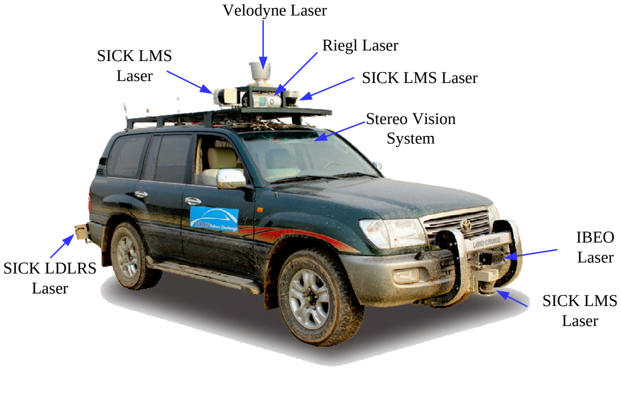


Fig. 10. Our test autonomous vehicle.

The proposed local trajectory planning algorithms are implemented in C++, which runs on a PC (Intel(R) Core(TM) i7@2.70GHz, 4.0GB memory) with Linux system. The proposed algorithm is executed with a cycle set to 100ms, which is limited by the frequency of the updating surrounding information from the perception system. We admit that the planning cycle of 100ms is not fast enough to deal with unexpected moving objects in imminent situations. The generated trajectory is followed by the low-level trajectory tracking controller. The maximal path length is set to 80m, which is limited by the sensing range of the perception system. In practice, we find that this sensing range could satisfy the mediate-speed (40kmh) navigation requirements in urban environments.

Figure 11 depicts the execution time of the planning algorithm obtained from a part of our experimental record data. The maximal number of generated trajectory candidates is set to 500. The total planning time mainly includes the reference path optimization time and trajectory planning time. It can be seen from Fig. 11 that the trajectory planning consumes around 20ms in each plan cycle, while the time spent on reference path optimization process is around 50ms. The traffic speed limit of the test road is 25km/h. The road width of our test environments is narrow (single lane width is less than 3m). There are oncoming vehicles there. The cyclists also share the road with cars. Besides, there are pedestrians walking along the lane. Sometimes there are baby carriages there. For safety purpose, the maximal allowed speed is set to 22km/h.

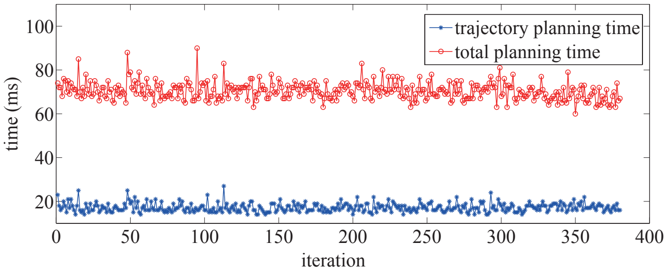
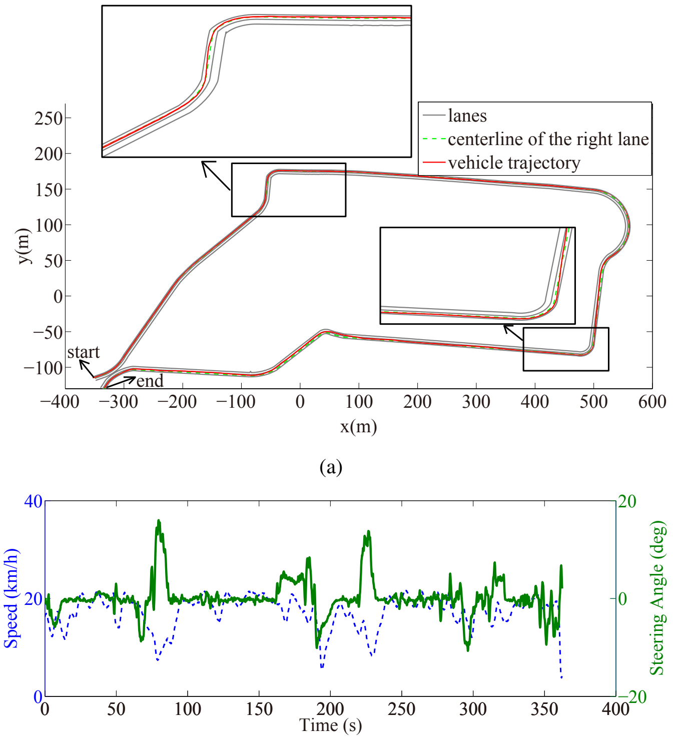


Fig. 11. Computation time of the proposed trajectory planner.

We have tested the proposed trajectory planning algorithms on our test autonomous vehicle driving thousands of kilometers with different traffic and weather conditions. A part of the experimental results in the realistic urban environments are chosen to illustrate the continuity of between replanning cycles of our proposed algorithm. As illustrated in Fig. 12(a), the vehicle runs on the road with double lanes. The vehicle keeps running on the right lanes in order to respect the traffic rules. The corresponding velocity and steering wheel angle profiles are shown in Fig. 12(b). It can be seen that the vehicle decelerates when it approaches a curvy turn and accelerates again during leaving the turn.

A variety of real-world driving scenarios are chosen from our test experiments to verify and evaluate the proposed local trajectory planning framework and algorithms. As shown in Fig. 13(a), a part of the current lane is blocked by a static vehicle stopping along the roadside in front. The command from the behavioral planner is lane following. In this situation, instead of following the centerline, the local trajectory planner selects a collision-free path which keeps a certain safe distance from the static obstacle. Figure 13(b) shows a similar traffic scenario. Comparing to the situation in Fig. 13(a), most part of the current lane is blocked by the stopping vehicle in front. To overtake the stopped car, the behavioral planner issues a lane changing maneuver command. A set of trajectory candidates are generated aligned with the reference path extracted from the adjacent lane.



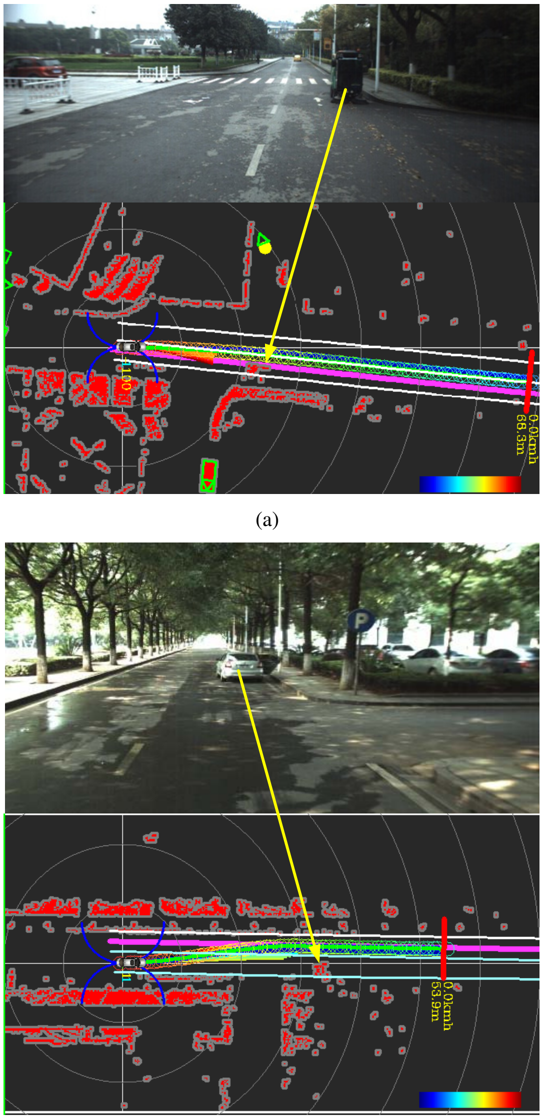
(a)

(b)

Fig. 12. Lane following results. (a) vehicle trajectory, (b) velocity and the steering angle of the front wheel profiles

Figure 14 depicts pedestrians and motorcycles avoidance scenarios. As shown in Fig. 14(a), when our vehicle is making a right turn, two motorcycles are moving alongside our AGV's current lane. One runs in the opposite direction, the other runs in the same direction as our vehicle. The behavioral decision is lane following. Instead of executing a sudden full braking action, the trajectory planner takes advantage of the free space and selects a trajectory candidate which keeps a certain safe margin from both of them. The safety margin is typically recognized as the hard constraint. In practice, for the sake of safety, the safety margin to the static obstacles (e.g. curb) and the moving objects (e.g. pedestrians, cyclists, vehicles) is tuned according to the vehicle's velocity. As illustrated in Fig. 14(b), there is a moving vehicle in front of the current lane. The vehicle's speed is 20.1kmh, which is less than the maximal allowed speed. In this situation, the behavioral planner issues a vehicle following command. In the meanwhile, a pedestrian is walking alongside the lane. Therefore, the trajectory planner is required to generate a safe trajectory following the front vehicle while avoiding the pedestrian.

As Fig. 15(a) shows, a pedestrian is walking across the lanes without caring about our AGV. The planner predicts the future movements of the pedestrian within a finite time horizon of 3s based on his current speed and direction. The behavioral planner decision is lane following. Hence, it avoids generating aggressive motions to pass the pedestrian from his right side. Besides, the adjacent lane is taken up by the on-coming traffic participants. So it makes the on-coming traffic participants feel uncomfortable if the AGV performs a double-lane-change maneuver to overtake the pedestrian. Instead the AGV slows down and stops in front of the pedestrian. The Fig. 15(b) shows the corresponding velocity generation profile.

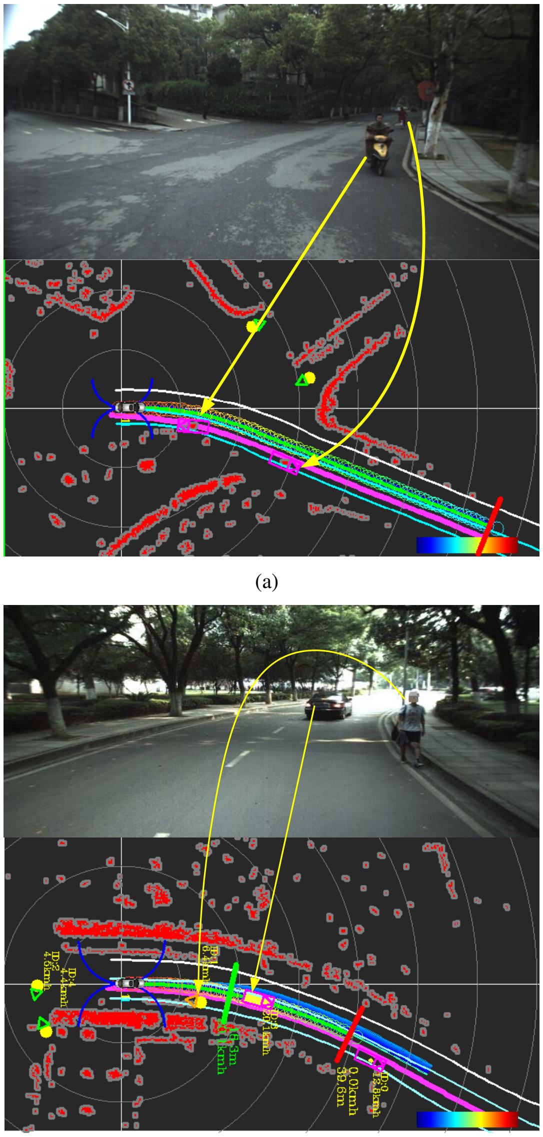


(a)

(b)

Fig. 13. Avoiding static obstacles in front of the current lane.

Figure 16(a) shows the trajectory planning results when the AGV negotiates a tight left turn with a moving vehicle in front. To interact with the oncoming vehicles naturally, we use the digital map information and assume that the other vehicles will continuously drive along their current lanes at their current speeds within a finite time horizon of 3s. For simplicity, we do not predict the lane changing maneuvers of the other vehicles. As shown in Fig. 16(a), since the adjacent lane is taken up by the oncoming traffic participants, the trajectory planner does not generate trajectory candidates passing through the adjacent lane. The vehicle selects a collision-free feasible trajectory, which is aligned with the centerline. The green line behind the front vehicle in our lane illustrates its speed and distance from the AGV along the centerline. In this situation, it is treated as a moving object instead of a static obstacle. It can be seen from



(a)

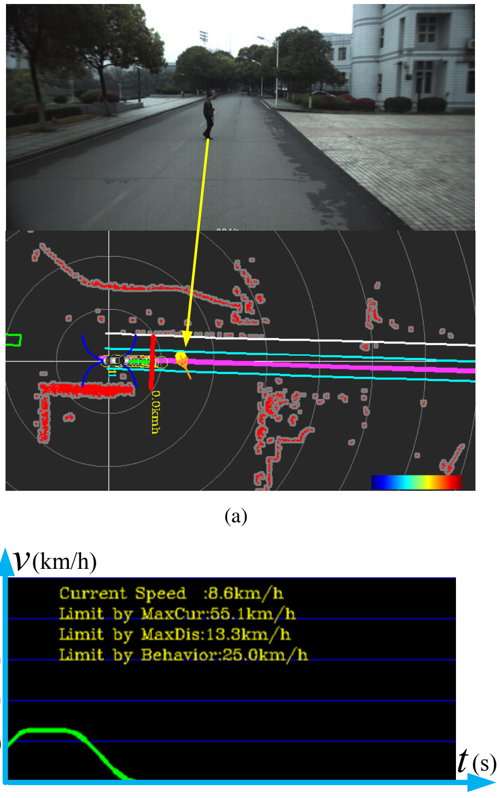
(b)

Fig. 14. Avoiding cyclists and pedestrians in the lanes.

Fig. 16(b), the maximal speed limited by the curvature is 14.8 km/h, which is less than the front vehicle's speed. Therefore, the maximal speed limited to 14.8km/h.

As shown in Fig. 17, an aggressive cutting-in vehicle in the adjacent lane performs a sudden right lane change into our vehicle's current lane. Since its current speed 26.6 km/h is greater than the maximal allowed speed 22km/h, the behavioral planner issues a lane following maneuver. Note that if the speed of the front vehicle is less than maximal allowed speed, adaptive cruise control speed profile generation strategy will be activated. For simplicity, velocity planner mainly considers the speed and distance of the front vehicle, which is nearest to our vehicle. In the future work, we will consider the speeds of all of the other vehicles in front of the AGV.

The experimental results mentioned above show the robustness of the proposed trajectory planning framework and its abilities of safely handling a variety of situations in realistic urban environments. Both the spatial path planning and velocity planning have to be explicitly considered to ensure driving safety and comfort. Due to length limit, only a small part of our experimental results are described in this paper. When a vehicle drives in dynamic urban driving environment, we find that one of the most difficult problems to deal with is moving objects, such as unexpected pedestrians, cyclists, on-coming vehicles, crossing vehicles, sudden cutting-in vehicles. When the vehicle drives on the realistic roads, the other moving objects around the AGV have to be carefully considered. Also, the static obstacles, such as curbs, parking cars, stopped pedestrians have to be carefully taken into account as well.



(a)

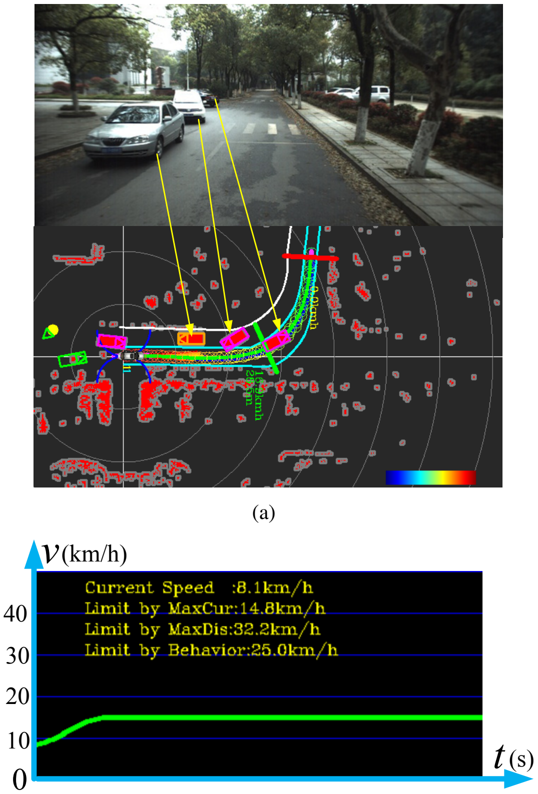
(b)

Fig. 15. Pedestrian avoidance. (a) avoiding a pedestrian passing across the front lane, (b) velocity planning profile

Due to the speed limits of the traffic regulation in our experimental tests, we demonstrate the performance of the proposed algorithm only in relatively low speed traffic situations. Besides, for safety purpose, the strategy we use is very conservative. We admit that there are still a lot of traffic situations we cannot handle smoothly and naturally as our human drivers, such as traffic free intersections, imminent situations, construction zones and so on. The prediction of the other moving vehicles and interactions with them have to be explicitly considered to achieve reliable and robust fully-autonomous driving in a larger range of realistic urban traffic scenarios.

VI. Conclusions and Future work

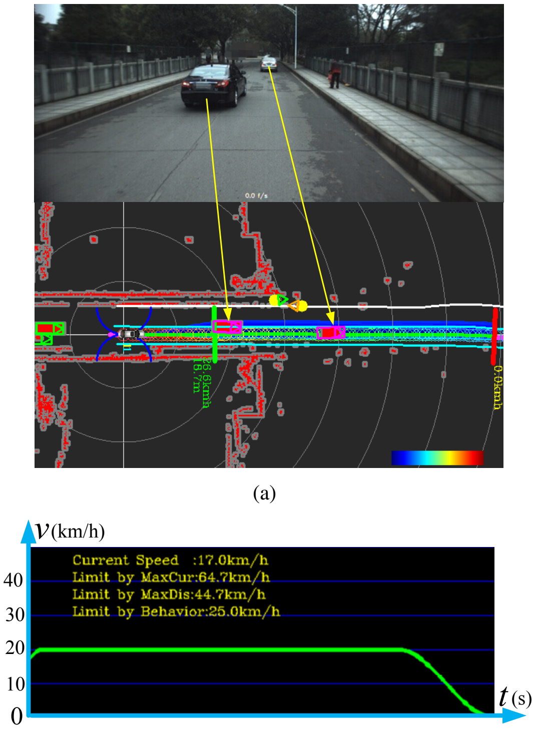
Safely and reliably navigating in realistic dynamic urban scenarios with autonomous vehicles remains a challenging



(a)

(b)

Fig. 16. Negotiating a tight left turn. (a) avoiding the opposite traffic participants and follow the front vehicle, (b) velocity planning profile



(a)

(b)

Fig. 17. A vehicle following scenario. (a) vehicle following and lane keeping, (b) velocity planning profile

problem. This paper has proposed an efficient real-time trajectory planning framework for autonomous driving in realistic urban environments. The presented trajectory generation algorithms employs a hierarchical planning strategy, which uses a reference path extracted from the digital map relying on the LiDAR-based localization technology. The reference path is refined and interpolated via the nonlinear optimization algorithm and parametric algorithm, respectively. Dividing the trajectory planning into spatial path generation and velocity profile generation has significantly reduced the solution space as well as enabled the planner to generate a rich set of dynamically-feasible trajectory candidates to handle a variety of urban driving scenarios, such as lane keeping, vehicle following, overtaking, static and dynamic objects avoidance. Although the presented algorithms have focused on addressing the trajectory planning problems for autonomous driving in urban environments. As far as our experience gained from the experimental tests goes, it could also easily be extended to solve general local path planning problems in unstructured and highway environments.

For future research, the proposed trajectory planning framework will be tested on a broader range of realistic urban traffic scenarios driving at higher speeds in order to verify and evaluate its performance more sufficiently. We also realize the present limits of the proposed trajectory planning algorithms, which have not explicitly considered the inherent uncertainty and noise arising from perception, localization and control systems. Understanding and predicting the maneuvers and movements of the other traffic participants , as well as interacting with them naturally are essential and challenging problems for autonomous driving in real-world traffic scenarios. To achieve safe and robust autonomous navigation performance, we will explicitly take uncertainty and noise in the prediction of both ego-vehicle and the other traffic participants into account in our future research. Furthermore, machine learning techniques will be introduced to adaptively tune a variety of parameters being involved in the trajectory planning algorithms.

ACKNOWLEDGMENT

This work was supported by the National Nature Science Foundation of China under grand No. 90820302. The authors would like to thank the members in the laboratory for their assistance during the experiments implementation. We also would like to thank anonymous reviewers for their constructive comments and suggestions that improved the paper.

REFERENCES

[1] J. Wang, J. Steiber, and B. Surampudi, “Autonomous ground vehicle control system for high-speed and safe operation,” *International Journal of Vehicle Autonomous Systems*, vol. 7, no. 1, pp. 18–35, 2009.

[2] H. Yu, K. Meier, M. Argyle, and R. Beard, “Cooperative path planning for target tracking in urban environments using unmanned air and ground vehicles,” *Mechatronics, IEEE/ASME Transactions on*, vol. 20, no. 2, pp. 541–552, April 2015.

[3] H. Wang and X. Liu, “Adaptive shared control for a novel mobile assistive robot,” *IEEE/ASME Transactions on Mechatronics*, vol. 19, no. 6, pp. 1725–1736, Dec 2014.

[4] K. Bengler, K. Dietmayer, B. Farber, M. Maurer, C. Stiller, and H. Winner, “Three decades of driver assistance systems: Review and future perspectives,” *IEEE Intelligent Transportation Systems Magazine*, vol. 6, no. 4, pp. 6–22, 2014.

[5] E. D. Dickmanns, B. Mysliwetz, and T. Christians, "An integrated spatio-temporal approach to automatic visual guidance of autonomous vehicles," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 20, no. 6, pp. 1273–1284, 1990.

[6] C. Thorpe, T. Jochem, and D. Pomerleau, "The 1997 automated highway free agent demonstration," in *IEEE Conference on Intelligent Transportation System*, 1997, pp. 496–501.

[7] A. Broggi, M. Bertozzi, A. Fascioli, and G. Conte, *the Experience of the ARGO Autonomous Vehicle*. World Scientific, 1999.

[8] S. Thrun, M. Montemerlo, H. Dahlkamp, D. Stavens, A. Aron, J. Diebel, P. Fong, J. Gale, M. Halpenny, and G. Hoffmann, "Stanley: The robot that won the darpa grand challenge," *Journal of field Robotics*, vol. 23, no. 9, pp. 661–692, 2006.

[9] C. Urmson, C. Ragusa, D. Ray, J. Anhalt, D. Bartz, T. Galatali, A. Gutierrez, J. Johnston, S. Harbaugh, and W. Messner, "A robust approach to high-speed navigation for unrehearsed desert terrain," *Journal of Field Robotics*, vol. 23, no. 8, pp. 467–508, 2006.

[10] U. Ozguner, C. Stiller, and K. Redmill, "Systems for safety and autonomous behavior in cars: The darpa grand challenge experience," *Proceedings of the IEEE*, vol. 95, no. 2, pp. 397–412, 2007.

[11] C. Urmson, J. Anhalt, D. Bagnell, C. Baker, R. Bittner, M. Clark, J. Dolan, D. Duggins, T. Galatali, and C. Geyer, "Autonomous driving in urban environments: Boss and the urban challenge," *Journal of Field Robotics*, vol. 25, no. 8, pp. 425–466, 2008.

[12] M. Montemerlo, J. Becker, S. Bhat, H. Dahlkamp, D. Dolgov, S. Ettinger, D. Haehnel, T. Hilden, G. Hoffmann, B. Huhnke *et al.*, "Junior: The stanford entry in the urban challenge," *Journal of Field Robotics*, vol. 25, no. 9, pp. 569–597, 2008.

[13] J. Levinson, J. Askeland, J. Becker, J. Dolson, D. Held, S. Kammel, J. Z. Kolter, D. Langer, O. Pink, and V. Pratt, "Towards fully autonomous driving: Systems and algorithms," in *IEEE Intelligent Vehicles Symposium*, 2011, pp. 163–168.

[14] F. Saust, J.-M. Wille, B. Lichte, and M. Maurer, "Autonomous vehicle guidance on braunschweig's inner ring road within the stadtpilot project," in *IEEE Intelligent Vehicles Symposium*, 2011, pp. 169–174.

[15] A. Geiger, M. Lauer, F. Moosmann, B. Ranft, H. Rapp, C. Stiller, and J. Ziegler, "Team annieway's entry to the 2011 grand cooperative driving challenge," *IEEE Transactions on Intelligent Transportation Systems*, vol. 13, no. 3, pp. 1008–1017, 2012.

[16] J. Wei, J. M. Snider, J. Kim, J. M. Dolan, R. Rajkumar, and B. Litkouhi, "Towards a viable autonomous driving research platform," in *IEEE Intelligent Vehicles Symposium*, pp. 763–770.

[17] P. Furgale, U. Schwesinger, M. Rufli, W. Derendarz, H. Grimmett, P. Muhlfellner, S. Wonneberger, J. Timpner, S. Rottmann, and B. Li, "Toward automated driving in cities using close-to-market sensors: An overview of the v-charge project," in *IEEE Intelligent Vehicles Symposium*, pp. 809–816.

[18] J. Ziegler, P. Bender, M. Schreiber, H. Lategahn, T. Strauss, C. Stiller, T. Dang, U. Franke, N. Appenrodt, and C. Keller, "Making bertha drive-an autonomous journey on a historic route," *IEEE Intelligent Transportation Systems Magazine*, vol. 6, no. 2, pp. 8–20, 2014.

[19] A. Broggi, P. Cerri, S. Debattisti, M. C. Laghi, P. Medici, M. Panciroli, and A. Prioletti, "Proud-public road urban driverless test: Architecture and results," in *IEEE Intelligent Vehicles Symposium*, 2014, pp. 648–654.

[20] D. Dolgov, S. Thrun, M. Montemerlo, and J. Diebel, "Path planning for autonomous vehicles in unknown semi-structured environments," *The International Journal of Robotics Research*, vol. 29, no. 5, pp. 485–501, 2010.

[21] D. Fassbender, A. Mueller, and H.-J. Wuensche, "Trajectory planning for car-like robots in unknown, unstructured environments," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2014, pp. 3630–3635.

[22] M. Pivtoraiko, R. A. Knepper, and A. Kelly, "Differentially constrained mobile robot motion planning in state lattices," *Journal of Field Robotics*, vol. 26, no. 3, pp. 308–333, 2009.

[23] M. Likhachev and D. Ferguson, "Planning long dynamically feasible maneuvers for autonomous vehicles," *The International Journal of Robotics Research*, vol. 28, no. 8, pp. 933–945, 2009.

[24] Y. Kuwata, S. Karaman, J. Teo, E. Frazzoli, J. How, and G. Fiore, "Real-time motion planning with applications to autonomous urban driving," *IEEE Transactions on Control Systems Technology*, vol. 17, no. 5, pp. 1105–1118, 2009.

[25] S. Thrun, M. Montemerlo, H. Dahlkamp, D. Stavens, A. Aron, J. Diebel, P. Fong, J. Gale, M. Halpenny, and G. Hoffmann, "Stanley: The robot that won the darpa grand challenge," *Journal of Field Robotics*, vol. 23, no. 9, pp. 661–692, 2006.

[26] M. Werling, S. Kammel, J. Ziegler, and L. Groll, "Optimal trajectories for time-critical street scenarios using discretized terminal manifolds," *The International Journal of Robotics Research*, vol. 31, no. 3, pp. 346–359, 2012.

[27] U. Schwesinger, M. Rufli, P. Furgale, and R. Siegwart, "a sampling-based partial motion planning framework for system-compliant navigation along a reference path," in *IEEE Intelligent Vehicles Symposium*, 2013, pp. 391–396.

[28] M. McNaughton, C. Urmson, J. M. Dolan, and J.-W. Lee, "Motion planning for autonomous driving with a conformal spatiotemporal lattice," in *IEEE International Conference on Robotics and Automation*, 2011, pp. 4889–4895.

[29] T. Gu and J. M. Dolan, "On-road motion planning for autonomous vehicles," in *Intelligent Robotics and Applications*. Springer, 2012, pp. 588–597.

[30] T. Gu, J. Snider, J. M. Dolan, and J.-w. Lee, "Focused trajectory planning for autonomous on-road driving," in *IEEE Intelligent Vehicles Symposium*, 2013, pp. 547–552.

[31] J. Wei, J. M. Snider, T. Gu, J. M. Dolan, and B. Litkouhi, "A behavioral planning framework for autonomous driving," in *IEEE Intelligent Vehicles Symposium*, pp. 458–464.

[32] T. Gu, J. M. Dolan, and J.-W. Lee, "On-road trajectory planning for general autonomous driving with enhanced tunability," in *Proceedings of the 13th International Conference on Intelligent Autonomous Systems*, 2014.

[33] J. Ziegler, P. Bender, T. Dang, and C. Stiller, "Trajectory planning for berthała local, continuous method," in *IEEE Intelligent Vehicles Symposium*, 2014, pp. 450–457.

[34] M. Althoff, O. Stursberg, and M. Buss, "Model-based probabilistic collision detection in autonomous driving," *IEEE Transactions on Intelligent Transportation Systems*, vol. 10, no. 2, pp. 299–310, 2009.

[35] M. Althoff and J. M. Dolan, "Online verification of automated road vehicles using reachability analysis," *IEEE Transactions on Robotics*, vol. 30, no. 4, pp. 903–918, 2014.

[36] S. Sivaraman and M. M. Trivedi, "Dynamic probabilistic drivability maps for lane change and merge driver assistance," *IEEE Transactions on Intelligent Transportation Systems*, vol. 15, no. 5, pp. 2063–2073, 2014.

[37] W. Xu, J. Pan, J. Wei, and J. M. Dolan, "Motion planning under uncertainty for on-road autonomous driving," in *IEEE International Conference on Robotics and Automation*, 2014, pp. 2507–2512.

[38] J. Levinson, M. Montemerlo, and S. Thrun, "Map-based precision vehicle localization in urban environments." in *Robotics: Science and Systems*, vol. 4, 2007, p. 1.

[39] J. Levinson and S. Thrun, "Robust vehicle localization in urban environments using probabilistic maps," in *IEEE International Conference on Robotics and Automation*, 2010, pp. 4372–4378.

[40] A. Joshi and M. 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.

[41] H. Choi, K. W. Yang, and E. Kim, "Simultaneous global localization and mapping," *IEEE/ASME Transactions on Mechatronics*, vol. 19, no. 4, pp. 1160–1170, 2014.

[42] J. Ziegler, H. Lategahn, M. Schreiber, C. G. Keller, C. Knoppel, J. Hipp, M. Haueis, and C. Stiller, "Video based localization for bertha," in *IEEE Intelligent Vehicles Symposium*, 2014, pp. 1231–1238.

[43] J. Fabian and G. Clayton, "Error analysis for visual odometry on indoor, wheeled mobile robots with 3-d sensors," *IEEE/ASME Transactions on Mechatronics*, vol. 19, no. 6, pp. 1896–1906, Dec 2014.

[44] G. Panahandeh and M. Jansson, "Vision-aided inertial navigation based on ground plane feature detection," *IEEE/ASME Transactions on Mechatronics*, vol. 19, no. 4, pp. 1206–1215, 2014.

[45] T. Howard, M. Pivtoraiko, R. A. Knepper, and A. Kelly, "Model-predicitve motion planning several key developments for autonomous mobile robots," *IEEE Rototics and Automation Magazine*, pp. 64–73, 2014.

[46] K. Chu, M. Lee, and M. Sunwoo, "Local path planning for off-road autonomous driving with avoidance of static obstacles," *IEEE Transaction on Intelligent Transportation System*, vol. 13, no. 4, pp. 1599–1616, 2012.

[47] M. J. Van Nieuwstadt and R. M. Murray, "Real time trajectory generation for differentially flat systems," *International Journal of Robust and Nonlinear Control*, vol. 8, pp. 995–1020, 1996.

[48] T. D. Barfoot and C. M. Clark, "Motion planning for formations of mobile robots," *Robotics and Autonomous Systems*, vol. 46, no. 2, pp. 65–78, 2004.

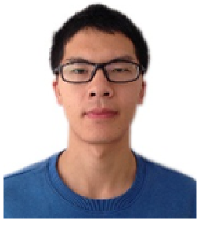
[49] T. Gu and J. M. Dolan, "Toward human-like motion planning in urban environments," in *IEEE Intelligent Vehicles Symposium*, 2014, pp. 350–355.

Xiaohui Li received his B.S. degree in electrical engineering from Harbin Institute of Technology, P. R. China, in 2009. Currently he is working toward the Ph.D. degree with the College of Mechatronic Engineering and Automation, National University of Defense Technology, P. R. China. His research activities include the hardware and software architecture design for autonomous ground vehicles. His research interests include artificial intelligence, motion planning, and control for intelligent systems.

Zhenping Sun received the Ph.D. degree in Pattern Recognition and Intelligent System from the College of Mechatronic Engineering and Automation, National University of Defense Technology, P. R. China, in 2004. He is now an associate professor. His research activities include the hardware and software architecture design for autonomous ground vehicles. His research interests include robotic motion planning, intelligent control of autonomous vehicles.

Dongpu Cao received the Ph.D. degree from Concordia University, Canada, in 2008. He is currently a Lecturer at Center for Automotive Engineering, Cranfield University, UK. His research focuses on electric and hybrid vehicles, vehicle dynamics and control, driver modeling, and intelligent vehicles, where he has contributed more than 80 publications and 1 US patent. He received the ASME AVTT2010 Best Paper Award and 2012 SAE Arch T. Colwell Merit Award. Dr. Cao serves as an Associate Editor for IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS,IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, and IEEE TRANSACTIONS ON INDUSTRIAL ELECTRONICS. He has been a Guest Editor for VEHICLE SYSTEM DYNAMICS, IEEE/ASME TRANSACTIONS ON MECHATRONICS, and IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS. He serves on the SAE International Vehicle Dynamics Standards Committee and a few ASME, SAE, IEEE technical committees.

Zhen He received his B.S. degree in Computer Science from Southwest Jiaotong University, P. R. China, in 2011 and then in 2013 the M.S. degree in Control Science and Engineering from National University of Defense Technology, P. R. China, where he is currently working toward the Ph.D. degree. His research interests include machine learning, statistical modeling and motion planning.



Qi Zhu received the B.S. and M.S. degree in control science and engineering at the College of Mechatronic Engineering and Automation, the National University of Defense Technology, Changsha, P. R. China in 2011 and 2014 respectively. Currently he is a Ph.D. student at the same university. His research interests are decision making and motion planning for autonomous ground vehicles under uncertainties as well as human driver behavior modeling.