2017 IEEE Intelligent Vehicles Symposium (IV)
June 11-14, 2017, Redondo Beach, CA, USA

Spatially-Partitioned Environmental Representation and Planning Architecture for On-Road Autonomous Driving

Wei Zhan[1], Jianyu Chen[1], Ching-Yao Chan[2], Changliu Liu[1] and Masayoshi Tomizuka[1]

***Abstract*— Conventional layered planning architecture temporally partitions the spatiotemporal motion planning by the path and speed, which is not suitable for lane change and overtaking scenarios with moving obstacles. In this paper, we propose to spatially partition the motion planning by longitudinal and lateral motions along the rough reference path in the Frenét Frame, which makes it possible to create linearized safety constraints for each layer in a variety of on-road driving scenarios. A generic environmental representation methodology is proposed with three topological elements and corresponding longitudinal constraints to compose all driving scenarios mentioned in this paper according to the overlap between the potential path of the autonomous vehicle and predicted path of other road users. Planners combining A* search and quadratic programming (QP) are designed to plan both rough long-term longitudinal motions and short-term trajectories to exploit the advantages of both search-based and optimization-based methods. Limits of vehicle kinematics and dynamics are considered in the planners to handle extreme cases. Simulation results show that the proposed framework can plan collision-free motions with high driving quality under complicated scenarios and emergency situations.**

I. Introduction

Motion planning is one of the most challenging areas in autonomous driving, which should comprehensively consider the following factors:

1) driving qualities such as comfort and time-efficiency;
2) soft traffic rules such as speed limit and lane keeping;
3) hard traffic rules such stop sign and traffic signal;
4) avoiding collisions with all obstacles;
5) vehicle kinematics feasibility such as path curvature;
6) vehicle dynamics feasibility such as tire friction circle and engine traction limits.

For on-road driving in urban areas or on highway, the planner need to optimize 1) and 2), and guarantee 3) to 6) in a variety of driving scenarios with different types of roadway geometries and various kinds of road participants as moving obstacles.

There has been a considerable amount of research efforts devoted to motion planning of autonomous vehicles [1][2][3]. Some of the work were focused on designing a planner in a spatiotemporal framework. For example, a state-lattice-based spatiotemporal representation and search algorithm was proposed in [4] for motion planning of highway driving. A trajectory planner based on a single spatiotemporal optimization was proposed in [5]. The approach was applicable to various kinds of scenarios, but it still needed an ad hoc process for specific situations. The initial value of the numerical optimization significantly affected the final result due to the existence of local optima. Also, the runtime was not desirable for time-critical replanning.

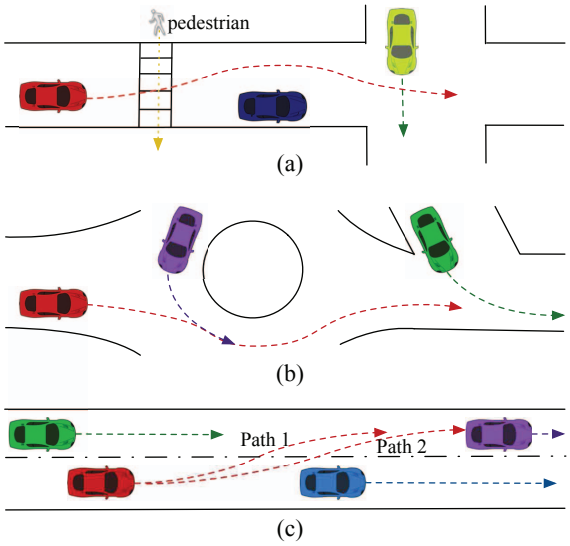


Fig. 1: The path of the host vehicle (red car) does not depend on the speed profile in: (a) scenarios with static obstacles (blue car) and moving obstacles crossing the path (pedestrian, green car), and (b) scenarios with merging obstacles (purple car in roundabout and green car at T-intersection). The path depends on the speed profile for (c) lane change or overtaking scenarios with moving obstacles (all other cars).

It is computationally intractable to include all factors in 1)-6) precisely in a single spatiotemporal framework to handle various kinds of scenarios. For search-based methods, wasteful explorations exist in the search space of trajectories. For optimization-based approaches, the complicated expressions of feasibility and safety constraints result in undesirable runtime. Therefore, a layered planning architecture temporally partitions the spatiotemporal planning by the path and speed profile planning, which significantly reduces computational complexity.

An elastic-band-based path planner and a speed-constraint-based temporal planner were proposed in [6] to generate trajectories in complicated scenarios for urban driving. Multiple on-road driving scenarios were handled in [7] by interpolating piecewise-linear paths with quintic Bézier curve,

[1]W. Zhan, J. Chen, C. Liu and M. Tomizuka are with the Department of Mechanical Engineering, University of California, Berkeley, CA 94720 USA (e-mail: wzhan, jianyuchen, changliuliu, tomizuka@berkeley.edu).

[2]C. Chan are with California PATH, University of California Berkeley, CA 94804 (e-mail:cychan@berkeley.edu).

and designing a velocity planner based on model predictive control. A driving strategy planning method via A* search was proposed in [8] by providing a long-term rough speed profile on a predefined lane. The predefined lane acted as the planned path and [8] essentially followed the conventional temporally-partitioned architecture.

Temporally-partitioned architecture is suitable for scenarios in Fig. 1 (a) and (b), in which path and speed profile can be well decoupled. However, for scenarios in Fig. 1 (c), the temporally-partitioned architecture is not applicable. Normally, scenarios in Fig. 1 (a), (b) and (c) may appear simultaneously in one planning horizon for on-road driving, thus our primary motivation is as follows:

Motivation 1: propose a novel planning architecture instead of the temporally-partitioned one, to handle on-road driving scenarios which include, but are not limited to scenarios in Fig. 1.

The combinatorial aspect of planning was considered in [9] based on the planner in [5]. Difficulties still existed in generating generic constraints. A behavioral planner was proposed in [10] with spatiotemporal candidate strategies and reaction prediction of surroundings. Adaptive cruise control and lateral control were required to achieve the desirable strategy. Safety constraints was only in the behavioral strategy level, and the final trajectory generated by the controllers may be unsafe in highly constrained environments. Thus our second motivation is:

Motivation 2: create generic environmental representation for each planning layer to generate safety constraints so that the final trajectory is guaranteed to be collision-free.

Besides the normal scenarios in Fig. 1, extreme scenarios may exist to push the autonomous vehicles to the boundaries of the vehicle kinematics and dynamics. A planner was proposed in [11] for highly constrained environments such as narrow roads with sharp turns. Planners for emergency maneuvers were designed in [12] and [13]. These planners targeted at desirable performance for specific extreme scenarios, but lack the portability for normal on-road driving scenarios. Extreme scenarios may appear simultaneously with normal ones, thus our third motivation is:

Motivation 3: include factors in 5) and 6) in the planning framework to deal with situations pushing the vehicle to the kinematics boundary such as U-turn, or dynamics boundary such as emergency maneuvers.

To the best of our knowledge, this work is the first to propose spatially-partitioned environmental representation and planning architecture, which are generic for all types of aforementioned on-road driving scenarios (see in Fig. 1). The planning framework comprehensively takes into account factors 1) to 6), which can also deal with extreme cases with guaranteed safety and feasibility.

The remainder of the paper is organized as follows. Section II provides the architecture of the proposed planning framework and Section III discusses generic representation of the environment for a variety of scenarios; the generic planning framework based on the proposed environmental representation is discussed in Section IV; simulation results are provided in Section V; and Section VI concludes the paper.

II. Overview of the Architecture

The motion planning module is presented in Fig. 2 from the viewpoint of the overall system architecture. The motion planning module receives the traffic-free reference path and desirable speed from the routing and reference planning module, as well as the driving task from the task planning module. At the final stage, the motion planning module outputs a trajectory satisfying factors 1)-6) described in Section I for the control module to execute. The perception and localization module provides the necessary inputs to all three stages of planning and control.

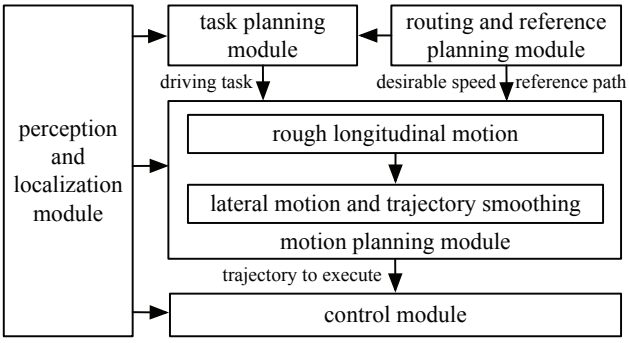


Fig. 2: Overview of the architecture

The processing of the routing and reference planning module can be completed partially offline, and will be further discussed in Section III-A. The task planning module determines the combinatorial aspect of planning in long term, such as following or overtaking a bicyclist, changing or keeping the lane, etc. Receiving a driving task does not mean that the final decision has been made. The task planning module may compare the costs of potential choices and select the best choice to execute, or plan ambiguous next-step motions with several potential decisions according to the intention probabilities of other road participants [14].

This paper is focused on the design of the motion planning module. Instead of adopting the temporally-partitioned architecture, we propose a novel spatially-partitioned planning architecture since the information associated with the temporal dimension is extremely important for any potential layer to deal with moving obstacles. Due to the nonholonomic motion constraints of the vehicle, longitudinal motion plays a more significant role than the lateral motion given the reference. Longitudinal constraints are often associated with higher priority, such as traffic signal, stop sign, car following, speed limit, turning speed on curvy road, yielding those who hold the right-of-way, etc. Also, when lane change and overtaking scenarios with moving obstacles (shown in Fig. 1 (c)) are involved, longitudinal motion is more influential in deciding the trajectory.

Therefore, we propose to spatially partition the spatiotemporal planning by the longitudinal and lateral motions along the reference. Such architecture also makes it possible to

linearize the safety constraints for each layer. We suggest that the planning framework should first plan long-term (at least 10 s) rough longitudinal motions according to the driving task and longitudinal constraints on safety and feasibility. Given the rough longitudinal motions, the rough short-term lateral motions are determined and smooth trajectories are further planned by considering both lateral and longitudinal constraints on safety and feasibility.

III. Generic Environmental Representation

In order to design a generic motion planning framework, we need a simplified and general representation of the environment to represent the longitudinal constraints for the long-term longitudinal motion planning. The methodology of generic environmental representation is proposed in this section, which is applicable for all types of scenarios mentioned in Section I. .

A. Representation in Frenét Frame

The lane centerline is a baseline as the traffic-free reference path (refer to [15] for comfortable and human-like improvement). A Frenét Frame can be defined on the reference path (shown in Fig. 3), in which the motion in Cartesian coordinates $(x(t), y(t))$ can be represented with the longitudinal position along the path $s(t)$, and lateral deviation to the path $d(t)$.

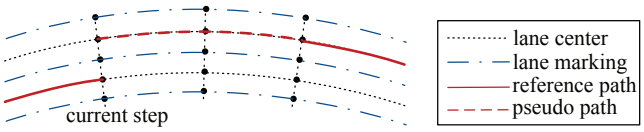


Fig. 3: Representation within Frenét Frame

Note that when changing the lanes, the reference path is switched. In order to keep the scale consistency of $s(t)$, a pseudo path is created on the target lane at the current time step for the horizon if a lane-change task is received (as shown in Fig. 3). The real reference path starts at the point where the lane change should be completed, and the decision on when to start and complete the lane change is left to the motion planning module.

Suppose $v_{\text{limit}}(s)$ is the speed limit introduced by traffic law, and $\kappa(s)$ is the curvature of the reference path in cartesian coordinate. The desirable speed v^{d} can be written as $v^{\text{d}}(s) = \min\left\{v_{\text{limit}}(s), \sqrt{a_{\text{lat}}^{\text{des}}/\kappa(s)}\right\}$, where $a_{\text{lat}}^{\text{des}}$ is the desirable lateral acceleration.

B. Boundaries and indicators

To guarantee safety, an autonomous vehicle needs to create the longitudinal boundaries at each time step for longitudinal motion planning. There are three categories of the boundaries according to the topological overlap of the reference path and the prediction of the obstacles.

Suppose $s_i(t)$ is the predicted longitudinal position (along the reference path of the host vehicle) of the i-th obstacle (can also be a stop bar for stop sign or traffic signal) which may affect the trajectory of the host vehicle. $v(t)$ and $v_i(t)$ are the speeds of the host vehicle and the i-th obstacle along the reference path, respectively. Then the safety factors can be represented by the following boundaries, as well as activeness indicators.

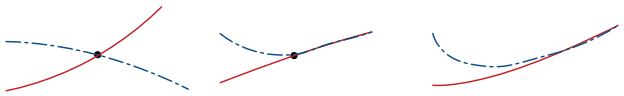
An obstacle can generate two kinds of boundaries. One is *front bound* blocking the host vehicle, which is written as $f_i(s(t), v(t), s_i(t), v_i(t)) \leq 0$; The other is *rear bound* forcing the host vehicle to pass, which is written as $g_i(s(t), v(t), s_i(t), v_i(t)) \leq 0$. More explanations on the boundaries can be found in Section III-C.

The change of the lateral positions of the host vehicle and obstacles can make the aforementioned longitudinal boundaries ineffective. Also, when lane change or overtaking is completed, the original target vehicle will be behind the host vehicle and the *rear bound* is no longer considered. Moreover, after a full stop near the stop sign, the boundary created by the stop bar can be discarded if the intersection is clear. Therefore, an indicator function is needed to indicate the activeness of the boundary. The boundaries of each obstacle have corresponding indicator functions $I_{fi}^{\text{act}}(t)$ and $I_{gi}^{\text{act}}(t)$, indicating whether the boundary is *active*, *potentially active* or *inactive* at each time step.

The *active* range of the boundary f_i or g_i can be written as $\left[t_{fi}^{\text{s}}, t_{fi}^{\text{e}}\right]$ or $\left[t_{gi}^{\text{s}}, t_{gi}^{\text{e}}\right]$ with start and ending time. If an obstacle only creates one boundary, it can be written as $[t_i^{\text{s}}, t_i^{\text{e}}]$.

C. Topological elements

Given the reference path and motion prediction of the obstacles (including traffic signal phase, stop sign, static and moving obstacles), the topological relationship between the potential paths of the host vehicle and the predicted path of the obstacle can be decomposed to three basic topological elements (shown in Fig. 4).



(a) point-overlap (b) line-overlap (c) undecided-overlap

Fig. 4: Topological elements to represent the relationship of the potential path of the host autonomous vehicle (red solid lines) and predicted path of the obstacles (blue dashed lines).

(a) Point-overlap corresponds to scenarios with crossing traffic (see scenarios in Fig. 1 (a) as examples), traffic signal or stop sign. When the prediction of the i-th obstacle crosses the reference path, it forms a *front bound* or a *rear bound*, that is,

$$f_i(t) = s(t) + \Delta s - s_i(t) \leq 0, \tag{1}$$

$$g_i(t) = s_i(t) + \Delta s - s(t) \leq 0, \tag{2}$$

where Δs is a safe margin containing the geometry of the obstacle. The *active* range of the boundary can be obtained from prediction of crossing traffic occupying the reference path, or the red-light phase of traffic signal. For motions at a

stop sign, the boundary is not *inactive* until a full stop $v = 0$ appears in a position range before the stop bar.

(b) Line-overlap corresponds to scenarios of merging and car following (see scenarios in Fig. 1 (b) as examples), including mergings at roundabout, highway ramp, intersections, etc. These scenarios have fixed merging or demerging point, which forms a *line-overlap* between the prediction and reference path.

If the obstacle is the front obstacle to follow, then it is a *front bound*

$$f_i(t) = s(t) + \frac{v(t)^2}{2\left|a_{\text{long}}^{\min}\right|} + \Delta s - s_i(t) - \frac{v_i(t)^2}{2\left|a_i^{\min}\right|} \leq 0, \quad (3)$$

where $a_{\text{long}}^{\min}$ and $a_i^{\min}$ are the maximum deceleration of the host vehicle and the i-th obstacle, respectively. The *active* range starts when the obstacle reaches the fixed merging point, and ends when it reaches the fixed demerging point.

If the obstacle follows the host vehicle, then it is a *rear bound*

$$g_i(t) = s_i(t) + \frac{v_i(t)^2}{2\left|a_i^{\min}\right|} + \Delta s - s(t) - \frac{v(t)^2}{2\left|a_{\text{long}}^{\min}\right|} \leq 0, \quad (4)$$

The *active* range starts when the host vehicle reaches the fixed merging point, and ends after a specific time range when the obstacle is regarded as a vehicle to follow the host.

(c) Undecided-overlap corresponds to lane change and overtaking scenarios with moving obstacles (see scenarios in Fig. 1 (c) as examples). These scenarios do not have fixed merging or demerging point since the desirable path depends on the speed profile. It is undecided where the overlap of the prediction and the reference path starts or end.

For obstacles to follow on the original lane or the target lane, (3) can be used as a *potentially active* boundary. When the host vehicle leaves the original lane and reaches the target lane, the boundary corresponds to the obstacle to follow on the original lane is *inactive* and that of the target lane is *active*.

When the host vehicle plans to cut in front of an obstacle, (4) can be used as a *potentially active* boundary. When the host vehicle reaches the target lane, the boundary becomes *active*. Then the boundary is *inactive* when the obstacle follows the host.

For an oncoming vehicle on the lane used temporarily by the host vehicle to accomplish the overtaking maneuver, (1) can be used as a *potentially active* boundary. When the host vehicle reaches the lane of the oncoming vehicle, the boundary is *active*. When the host vehicle completely leaves the lane of the oncoming vehicle, the boundary is *inactive*.

IV. Motion Planning Framework

In this section, planners for long-term longitudinal motion, short-term lateral motion and trajectory smoothing are designed. Optimization-based methods perform well in providing motions with high driving quality, but they are inefficient in dealing with nonlinear and nonconvex terms. Search-based approaches are efficient in handling complex terms, but hard to provide smooth motions. In this section, search-based and optimization-based methods are combined to find desirable motions for the long-term rough speed and short-term trajectory. We first use A* search to find a rough reference, then formulate a quadratic programming (QP) optimization problem which can be solved extremely fast with global minimum.

A. Long-term longitudinal motions

The generic representation described in Section III is used to plan long-term rough speed via A* search, as is inspired by [8]. The state of this problem is defined as $z = [s, v, t, I^{\text{act}}]^T$, where t is the time stamp which will increase by Δt_l once an action is applied, and I^{act} is a vector containing all relevant activeness indicators within the horizon.

The action is discretized as a set of accelerations $a = [-2, -1, 0, 1]$ m/s^2. Such discretization was used in [8] which well covered normal actions. When no sequence of action can be found within the action set to obtain a collision-free speed profile, the planner will calculate the minimum constant acceleration or deceleration to avoid collision, which corresponds to an emergency situation. s and v are updated by applying a constant acceleration in a period of sampling time Δt_l. Since actions are discretized and the sampling time is fixed, there will be lots of repeated states. Thus graph search is applied instead of tree search to prevent revisiting visited states.

The cost function of A* search basically includes penalty for accelerations and deviation from the desired speed, which is expressed as

$$\min_a \quad \sum_{k=1}^{T_l/\Delta t_l} w_1 a(k)^2 + w_2 \left|v(k) - v^{\text{d}}(s(k))\right| + I^{\text{c}}(z(k)),$$

where I^{c} is a function indicating whether any of the constraints related to driving task and collision avoidance is violated. At each step, if the boundary is *active*, the collision avoidance conditions are checked directly. A large cost will be added to I^{c} if there is a violation to the boundary. During forward expansion of the nodes, if the boundary is *potentially active*, the planner will check if the switch conditions are satisfied. When the algorithm expands a node satisfying the switch condition of a boundary, the indicator will be set as *active* or *inactive* at the current step. Inevitable Collision States (ICS) in the sense of all *active* boundaries are used as a heuristic.

To smooth the reference rough speed profile in the short-term horizon and reduce the sampling time, we formulate an optimization problem. Suppose $s_l(t)$ is the position provided by the long-term rough speed planning layer with sampling time Δt_l, and $s(t)$ is the position provided by the short-term speed smoothing layer with sampling time Δt_s, where $\Delta t_l = n\Delta t_s$. Then the optimization problem can be written as the QP form,

$$\min_s \quad \sum_{t=\Delta t_s}^{T_s} w_1 \ddot{s}(t)^2 + w_2 \dddot{s}(t)^2 + \sum_{t=\Delta t_l}^{T_s} w_3 \left(s_l(t) - s(t)\right)^2,$$
$$\text{s.t.} \quad A_s s \leq b_s,$$

where T_s is the short-term preview horizon, $\ddot{s}$ and $\dddot{s}$ are longitudinal acceleration and jerk, corresponding to the driving comfort. A_s and b_s are constructed via linear hard constraints, such as longitudinal acceleration $a_{\min} \leq \ddot{s}(t) \leq a_{\max}$, and collision avoidance for (1)-(4) by setting $v = v_i$. An alternative for designing the longitudinal motion planner can be found in [16].

B. Short-term lateral motions

A simple illustration of the trajectory planner is shown in Fig. 5. Given the longitudinal positions from the smoothed speed profile, we can project them on the reference path, and create points with the same longitudinal positions in a Frenét Frame with lateral sample interval h. Then an A* search algorithm is designed with constraints on collision avoidance to lateral obstacles and feasibility according to the vehicle kinematics and dynamics.

Suppose $q_r(t)$ is the position of the rear axle center of the vehicle in A* search. $e(t)$ represents the edge connecting the points $q_r(t)$ and $q_r(t - \Delta t_s)$. The cost for A* search is set as

$$\min_{q_r} \sum_{t=\Delta t_s}^{T_s} w_1 \|e(t)\| + w_2 |d(t)| + I_{\text{colli}}(e(t)),$$

where I_{colli} is a function to set the cost to be a large value when lateral collision exists. Then heuristic is "holonomic-with-obstacles" distance used in [17], which can be calculated by dynamic programming. The admissibility can be easily proved and the optimality can be guaranteed.

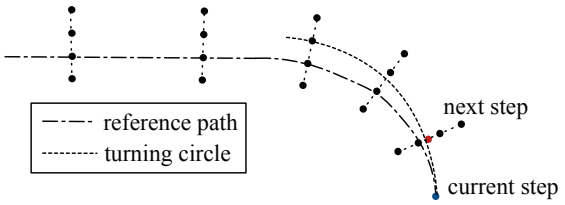


Fig. 5: Illustration of the A*-based trajectory planner with tight turns. The turning circle corresponds to the maximum curvature $\kappa_{\max}$. When expanding the nodes of the next step, the point with the minimal cost may be out of the kinematics boundary. Then the A* algorithm will check the red point on the kinematics boundary, and it may have the lowest cost. Then the point on the boundary will be expanded so that the autonomous vehicle can plan a trajectory with a maximal steering angle.

When the vehicle is making a tight turn such as a U-turn, it is possible that the minimum cost node under forward expansion is out of the kinematics boundary of the vehicle. If so, the A* algorithm will also check the point on the kinematics boundary, as is shown in Fig. 5. Bicycle model can also be used as the kinematic model when lateral acceleration is low [18].

Since time stamps are associated with the each point, we can also check whether vehicle dynamics limits are satisfied. G-G diagram (shown in Fig. 6 (a)) is widely used for the high-level control of racing cars [19] to push

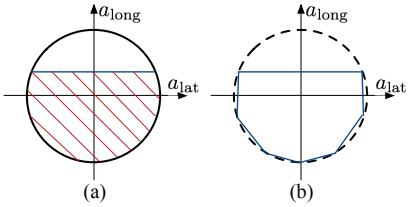


Fig. 6: (a) illustrates of G-G diagram. The circle corresponds to tire friction circle, and the blue line which bounds the maximum longitudinal acceleration corresponds to the engine traction limit. (b) illustrates the linearized approximation of G-G diagram.

the vehicle dynamics to the limits. When expanding the node, longitudinal and lateral acceleration can be obtained and we can check the dynamics feasibility with emergency maneuvers according to the G-G diagram. More detailed discussions on kinematic and dynamic vehicle model near limits can be found in [20].

C. Trajectory smoothing

Finally, the trajectory smoothing for the autonomous vehicles can be formulated as a QP problem. The motion representation can be found in Figure 7 (a). $q(t) = [x(t) \quad y(t)]^T$ is the position vector of the rear axle center of the vehicle at time step t. The velocity, acceleration and jerk $\dot{q}$, $\ddot{q}$ and $\dddot{q}$ can be obtained via backward differences, and yaw angle $\phi(t) = \arctan \dot{y}(t)/\dot{x}(t)$.

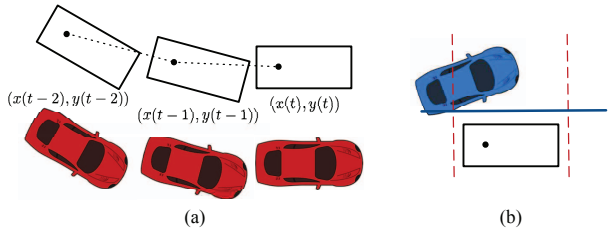


Fig. 7: (a) Motion representation of the QP problem. (b) Linearization of lateral obstacle avoidance constraints: the rough trajectory obtained from A* provides the rough position and yaw angle at each time step, from which we draw the rough vehicle body represented by a rectangle. We first draw two lines (red dashed lines) which are perpendicular to the rough direction with specific margin to the vehicle body. Then obstacles in the area between the two lines are considered as lateral obstacles at the current time step. Then the boundary line (blue solid line), which is parallel to the direction, is pushed as far as possible until it hits the obstacle. Then the boundary line can be used as a linearized safety constraints for lateral obstacles.

By taking into account the driving comfort and position error to the rough trajectory as the objective of optimization, the QP problem can be written as

$$\min_{q} \sum_{t=\Delta t_s}^{T_s} w_1 \|\ddot{q}(t)\|^2 + w_2 \|\dddot{q}(t)\|^2 + w_3 \|q_r(t) - q(t)\|^2,$$

$$\text{s.t.} \quad Aq \leq b,$$

where A and b are constructed via linear hard constraints. The linearized constraints can be found in Section IV-A for

longitudinal obstacles and in Fig. 7 (b) for lateral obstacles. Given the rough direction of the vehicle, boundaries for vehicle dynamics can be constructed with linearized G-G diagram shown in Fig. 6 (b).

V. Illustrative Examples

In this section, exemplar use cases are shown to illustrate the performance of the proposed generic representation and motion planning framework. *Scenario 0* is not necessarily practical in real world, but involves all the three topological elements in one planning horizon to explain how the generic environmental representation works (without planning results). *Scenario 1* is a practical lane change scenario with a pedestrian to yield, which contains two topological elements to test the planning capability of the proposed framework. *Scenario 2* is an extreme case to test the capability of the proposed framework to handle emergency situations. When running the algorithm in python on a laptop with Intel Core i7-6600U 2.6 GHz CPU, the runtime corresponding to the worst case can be bounded within 0.2 s with $T_l = 10$ s, and 0.1 s with $T_l = 9$ s. Particularly, the A* search for rough speed profile dominated the runtime.

A. Scenario 0

An exemplar scenario in Fig. 8 to explain how the environmental representation works. The environmental representation in the position-time domain is shown in Fig. 9, in which T_l is the long-term planning horizon. Note that in order to represent the boundaries in Eq. (3) and (4) in the position-time domain, the speed of the host vehicle in Eq. (3) and (4) is set to be $v = v_i$. Note that it is not the real speed of the host vehicle.

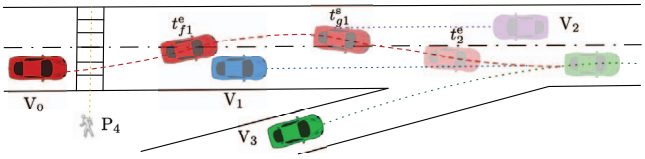


Fig. 8: A complicated exemplar scenario: the task of the host autonomous vehicle V_0 (red) is to overtake V_1 (blue) in front, return to the original lane to avoid the oncoming V_2 (purple) in the opposite lane, and then to follow the V_3 (green) that is merging into the lane from a ramp junction. V_0 does not have to yield to the pedestrian P_4 since it is relatively far away from the crosswalk.

The activeness indicators of the boundaries corresponding to P_4 (*point-overlap*) and V_3 (*line-overlap*), namely I^{act}_4 and I^{act}_3, are *active* within specific time periods (represented by solid lines). The indicators of the boundaries corresponding to V_1 and V_2 (*undecided-overlap*), I^{act}_{f1}, I^{act}_{g1} and I^{act}_2, are *potentially active* (represented by dashed lines). Given a rough speed profile of V_0 (speed profile candidates in search-based algorithms), the *active* range of boundaries of V_1 and V_2 can be determined. Before t^{e}_{f1}, V_0 follows V_1 and I^{act}_{f1} is *active*. Between t^{e}_{f1} and t^{s}_{g1}, V_1 should be on the right side of V_0, and V_1 can be regarded as a lateral obstacle. After t^{s}_{g1}, V_0 merges into the original lane, and V_1 can be

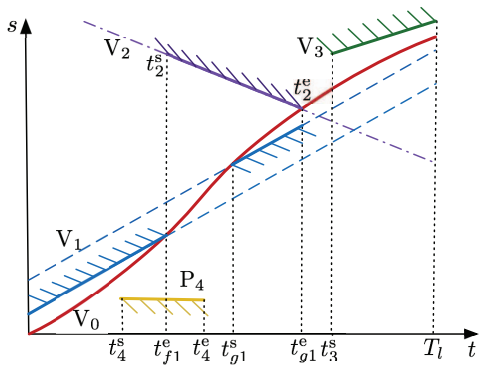


Fig. 9: Environmental representation in position-time domain

regarded as a longitudinal obstacle behind. I^{act}_{g1} is *active* to achieve a safe and courteous cut-in, and it should be *inactive* after V_0 is completely on the target lane (t^{e}_{g1}). Between t^{e}_{f1} and t^{e}_{g1}, I^{act}_{f2} is *active* since V_0 is on the lane of V_2, which decides t^{s}_2 and t^{e}_2.

B. Scenario 1

The first practical scenario that we used for the proposed planning framework was a challenging lane change with pedestrian yielding shown in Fig. 10. The parameter settings were as follows. The long-term preview horizon for rough speed profile planning was $T_l = 10$ s, and the sampling time was $\Delta t_l = 1$ s. The short-term preview horizon for speed profile smoothing and trajectory planning was $T_s = 5$ s, and the sampling time was $\Delta t_s = 1/3$ s. Lateral position sample interval $h = 0.25$ m. A rectangle was used to represent the vehicle body. $v_{\mathrm{limit}} = 10$ m/s and $\kappa_{\max} = 0.2$ m^{-1}.

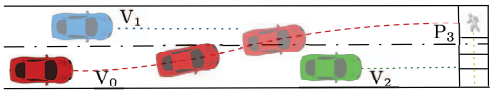


Fig. 10: The task planner required the host autonomous vehicle (V_0) to merge to the target lane on the left ahead of a vehicle on that lane (V_1), which is relatively slow (5 m/s). Before completing the lane change, V_0 had to follow a vehicle on its lane (V_2), which is quite slow (3 m/s). Also, a pedestrian (P_3) approached the crosswalk to cross the street, and V_0 had to yield to the pedestrian.

Fig. 11 reveals the boundaries for rough speed profile planning and the distance traveled by V_0 over time with the planned rough speed profile. The activeness indicators of the boundaries corresponding to P_3 (*point-overlap*), I^{act}_3, was *active* within specific time periods (represented by solid cyan lines). When the rough speed profile of V_0 is not planned, the indicators of the boundaries corresponding to V_1 and V_2 (*undecided-overlap*), I^{act}_1 and I^{act}_2, are *potentially active* (represented by blue and green dashed lines, respectively).

We used the planning framework described in Section and IV-A to plan a rough speed profile for V_0 (the solid red curve), and the *active* range of boundaries of V_1 and V_2 can be determined. At first, the boundary for following V2, $f_2 \leq 0$, remained *active*. V_0 accelerated to reach the

boundary for overtaking V_1, namely, $g_1 \leq 0$. When V_0 reached the *potentially active* boundary, I_1^{act} is *active*. When the range was enough for V_0 to accomplish a relatively courteous cut-in, V_1 can be regarded as a longitudinal obstacle behind and I_1^{act} is *inactive*. Moreover, when approaching the crosswalk, V_0 tended to make a relatively sharp brake first and accelerate when the pedestrian had not completely left the conflict zone. The reason was that such maneuver maximized the time-efficiency of V_0 so that it could maintain higher speed which was closer to the desirable speed.

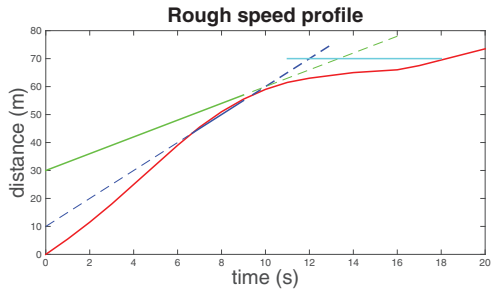


Fig. 11: Rough speed profile and boundaries

The rough speed profile helped the trajectory planner determine whether an obstacle should be a longitudinal or lateral obstacle at each time step. When I_1^{act} is *active*, V_1 was regarded as a longitudinal obstacle behind V_0. Before that, V_1 was a lateral obstacle on the left; after that, V_1 was no longer considered. When I_2^{act} is *active*, V_2 was a longitudinal obstacle to follow. After that, V_2 was a lateral obstacle on the right.

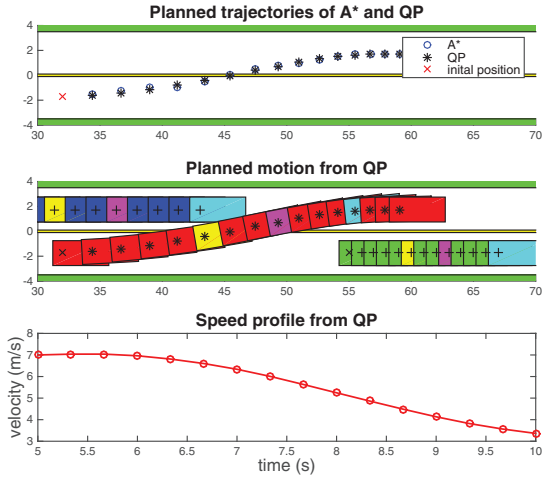


Fig. 12: Planned trajectories

Given the long-term speed profile, the A* search obtained a rough trajectory, and QP smoothened it to get the final trajectory. The trajectories obtained by A* and QP at time $t = 5$ s for a short-term horizon $T_s = 5$ s were compared in Fig. 12. As is demonstrated by the figure, the rough trajectory generated by A* provided a fairly good reference, and QP locally smoothened the trajectory to make it comfortable. Fig. 12 also demonstrated the relationship of the motions of V_0, V_1 and V_2 within the preview horizon. The three vehicles were marked with the same color at three representative time steps (yellow for $t = 6.66$ s, purple for $t = 7.66$, and cyan for $t = 9$ s). During $t = 6.66 - 9$ s, the boundary of V_1 is *active*, which means that V_0 reached the target lane at $t = 6.66$ s, and became the front vehicle of V_1 at $t = 9$ s.

C. Scenario 2

The second practical scenario considered an emergency maneuver at a two-way-stop intersection as is shown in Fig. 13. The parameter settings were the same as Section V-B, and acceleration limits were $a_{\max} = 2.5$ m/s^2 and $a_{\min} = -7$ m/s^2.

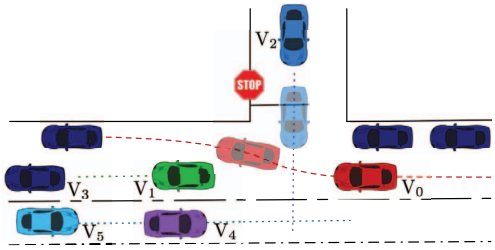


Fig. 13: Illustration of the second exemplar scenario: the host autonomous vehicle V_0 (red) was passing a two-way-stop intersection, following V_1 (green) and holding the right of way against any vehicle from the perpendicular cross street controlled by a stop sign. Deep blue cars, including V_3, were all static obstacles. V_2 (blue) was violating the stop sign with a relatively high speed. Since the parked cars blocked the view of V_0, it cannot see V_2 until it was too late to apply emergency braking. V_0 had to accelerate to avoid being hit by V_2. However, V_3 temporarily blocked the lane and V_1 had to slow down and come to a stop. Also, there were oncoming vehicles V_4 and V_5. Hence, V_0 had to swerve to the parking lane on the right side to maneuver around. When reaching the parking lane, it had to slow down immediately to come a stop after escaping the conflict with V_2.

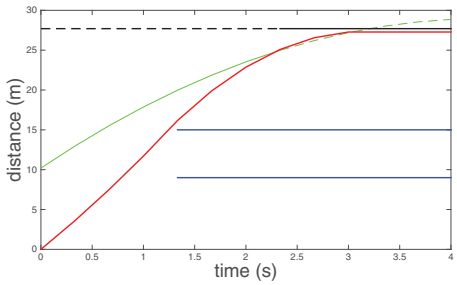


Fig. 14: Rough speed profile and boundaries

When applying the method in Section IV-A, no feasible solution can be found within the set of acceleration. Therefore, the rough speed planner resorted to plan a course of collision-free motions with a combination of constant acceleration and deceleration within $a_{\max}$ and $a_{\min}$. Fig. 14 reveals that it is impossible to apply emergency stop to make the distance under the lower blue bound created by V_2. Therefore, it has to accelerate to pass the upper blue bound. Fig. 15

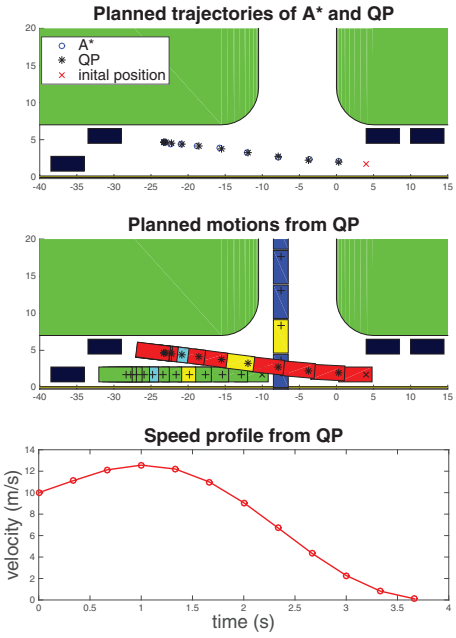


Fig. 15: Planned trajectories

demonstrates the trajectories generated by A* and QP, which reveals the performance of the planner to generate safe and feasible trajectories under an emergency situation. The three vehicles were marked with the same color at representative time steps (yellow for $t = 1.33$ s and cyan for $t = 2.33$ s).

VI. Conclusion

A spatially-partitioned environmental representation and planning architecture was proposed for on-road autonomous driving in this paper, which can generate safe and feasible trajectories with high driving qualities when dealing with a variety of complicated driving scenarios. The planning framework spatially partitioned longitudinal and lateral motions instead of path and speed in the temporally-partitioned architecture. Three topological elements and corresponding longitudinal boundaries were proposed to compose the representation of all types of on-road driving scenarios discussed in this paper. A* search plus quadratic programming (QP) planners were designed for long-term longitudinal motions and short-term trajectory to leverage the advantages of each method, and the linearization of safety and feasibility constraints obtained from the proposed methodology. As was demonstrated in the simulation results, the proposed planning framework can generate high-quality and collision-free trajectories in the long term in complicated driving scenarios and emergency situations.

References

[1] B. Paden, M. p, S. Z. Yong, D. Yershov, and E. Frazzoli, "A Survey of Motion Planning and Control Techniques for Self-Driving Urban Vehicles," *IEEE Transactions on Intelligent Vehicles*, vol. 1, no. 1, pp. 33–55, Mar. 2016.

[2] D. Gonzlez, J. Prez, V. Milans, and F. Nashashibi, "A Review of Motion Planning Techniques for Automated Vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 4, pp. 1135–1145, Apr. 2016.

[3] C. Katrakazas, M. Quddus, W.-H. Chen, and L. Deka, "Real-time motion planning methods for autonomous on-road driving: State-of-the-art and future research directions," *Transportation Research Part C: Emerging Technologies*, vol. 60, pp. 416–442, Nov. 2015.

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

[5] J. Ziegler, P. Bender, T. Dang, and C. Stiller, "Trajectory planning for Bertha - A local, continuous method," in *2014 IEEE Intelligent Vehicles Symposium Proceedings*, Jun. 2014, pp. 450–457.

[6] 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 *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Sep. 2015, pp. 250–256.

[7] X. Qian, I. Navarro, A. de La Fortelle, and F. Moutarde, "Motion planning for urban autonomous driving using Bézier curves and MPC," in *Intelligent Transportation Systems (ITSC), 2016 IEEE 19th International Conference on*. IEEE, 2016, pp. 826–833.

[8] C. Hubmann, M. Aeberhard, and C. Stiller, "A generic driving strategy for urban environments," in *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*, Nov. 2016, pp. 1010–1016.

[9] P. Bender, O. S. Tas, J. Ziegler, and C. Stiller, "The combinatorial aspect of motion planning: Maneuver variants in structured environments," in *2015 IEEE Intelligent Vehicles Symposium (IV)*, Jun. 2015, pp. 1386–1392.

[10] J. Wei, J. Snider, T. Gu, J. Dolan, and B. Litkouhi, "A behavioral planning framework for autonomous driving," in *2014 IEEE Intelligent Vehicles Symposium Proceedings*, pp. 458–464.

[11] D. Fassbender, B. C. Heinrich, and H.-J. Wuensche, "Motion planning for autonomous vehicles in highly constrained urban environments," in *Intelligent Robots and Systems (IROS), 2016 IEEE/RSJ International Conference on*. IEEE, 2016, pp. 4708–4713.

[12] M. Keller, C. Ha, A. Seewald, and T. Bertram, "A Model Predictive Approach to Emergency Maneuvers in Critical Traffic Situations," in *2015 IEEE 18th International Conference on Intelligent Transportation Systems*, Sep. 2015, pp. 369–374.

[13] J. Salvado, L. M. Custdio, and D. Hess, "Contingency planning for automated vehicles," in *Intelligent Robots and Systems (IROS), 2016 IEEE/RSJ International Conference on*. IEEE, 2016, pp. 2853–2858.

[14] W. Zhan, C. Liu, C. Y. Chan, and M. Tomizuka, "A non-conservatively defensive strategy for urban autonomous driving," in *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*, pp. 459–464.

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

[16] C. Liu, W. Zhan, and M. Tomizuka, "Speed profile planning in dynamic environments via temporal optimization," in *2017 IEEE Intelligent Vehicles Symposium (IV)*, Jun. 2017.

[17] 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, Apr. 2010.

[18] P. Polack, F. Altché, B. d'Andréa Novel, and A. de La Fortelle, "The kinematic bicycle model: a consistent model for planning feasible trajectories for autonomous vehicles?" in *2017 IEEE Intelligent Vehicles Symposium (IV)*, Jun. 2017.

[19] J. Funke, P. Theodosis, R. Hindiyeh, G. Stanek, K. Kritatakirana, C. Gerdes, D. Langer, M. Hernandez, B. Muller-Bessler, and B. Huhnke, "Up to the limits: Autonomous Audi TTS," in *2012 IEEE Intelligent Vehicles Symposium (IV)*, Jun. 2012, pp. 541–547.

[20] F. Altché, P. Polack, and A. de La Fortelle, "A simple dynamic model for aggressive, near-limits trajectory planning," in *2017 IEEE Intelligent Vehicles Symposium (IV)*, Jun. 2017.