Special Issue Article

Journal of AUTOMOBILE ENGINEERING

Proc IMechE Part D:
J Automobile Engineering
2021, Vol. 235(4) 1101–1112
IMechE 2020
Article reuse guidelines:
sagepub.com/journals-permissions
DOI: 10.1177/0954407020906627
journals.sagepub.com/home/pid

Enable faster and smoother spatio-temporal trajectory planning for autonomous vehicles in constrained dynamic environment

Long Xin[1*], Yiting Kong[1*], Shengbo Eben Li[1], Jianyu Chen[2], Yang Guan[1], Masayoshi Tomizuka[2] and Bo Cheng[1]

Abstract

Trajectory planning is of vital importance to decision-making for autonomous vehicles. Currently, there are three popular classes of cost-based trajectory planning methods: sampling-based, graph-search-based, and optimization-based. However, each of them has its own shortcomings, for example, high computational expense for sampling-based methods, low resolution for graph-search-based methods, and lack of global awareness for optimization-based methods. It leads to one of the challenges for trajectory planning for autonomous vehicles, which is improving planning efficiency while guaranteeing model feasibility. Therefore, this paper proposes a hybrid planning framework composed of two modules, which preserves the strength of both graph-search-based methods and optimization-based methods, thus enabling faster and smoother spatio-temporal trajectory planning in constrained dynamic environment. The proposed method first constructs spatio-temporal driving space based on directed acyclic graph and efficiently searches a spatio-temporal trajectory using the improved A* algorithm. Then taking the search result as reference, locally convex feasible driving area is designed and model predictive control is applied to further optimize the trajectory with a comprehensive consideration of vehicle kinematics and moving obstacles. Results simulated in four different scenarios all demonstrated feasible trajectories without emergency stop or abrupt steering change, which is kinematic-smooth to follow. Moreover, the average planning time was 31 ms, which only took 59.05%, 18.87%, and 0.69%, respectively, of that consumed by other state-of-the-art trajectory planning methods, namely, maximum interaction defensive policy, sampling-based method with iterative optimizations, and Graph-search-based method with Dynamic Programming.

Keywords

Spatio-temporal trajectory planning, directed acyclic graph, convex feasible set, autonomous driving, dynamic environment

Date received: 30 July 2019; accepted: 9 December 2019

Introduction

In the past few decades, autonomous driving has achieved considerable development of state-of-the-art technologies in planning, control, and sensing.[1,2] Trajectory planning is of great importance in decision-making for autonomous driving. It generates collision-free trajectory toward the destination with measurements from perception and localization.[3] It is well demonstrated that the controller uses a sequence of motion-level commands to guide the autonomous vehicle along the predetermined trajectory robustly even at driving limits.[4] Therefore, we focus on aspects of planning to improve tracking performance such as safety, comfort, and economy.[4–6] If the trajectory is spatio-temporal rather than purely geometrical, those objectives can be better guaranteed in constrained dynamic environment.[4,7,8] Moreover, the conflict-free robust control and cooperative control for multiple connected

[1]State Key Laboratory of Automotive Safety and Energy, School of Vehicle and Mobility, Tsinghua University, Beijing, China
[2]Department of Mechanical Engineering, University of California–Berkeley, Berkeley, CA, USA.

*These authors contributed equally to this work.

Corresponding author:
Shengbo Eben Li, State Key Laboratory of Automotive Safety and Energy, School of Vehicle and Mobility, Tsinghua University, Beijing 100084, China.
Email: lishbo@tsinghua.edu.cn

vehicles also benefit from the fast generation of spatio-temporal trajectory.[1,2] Current methods for spatio-temporal trajectory planning are broadly classified into three main categories: sampling-based methods, graph-search-based methods, and optimization-based methods.

Sampling-based methods first connect the points sampled randomly or deterministically and select the desired trajectory among a series of candidates by collision check and performance evaluation. The spatio-temporal trajectory is generated with samples drawn in action space.[9] LaValle[10] proposed a random sampling-based method called rapidly exploring random tree (RRT), which was characterized by high planning efficiency but poor trajectory feasibility. Kuwata et al.[11] used a closed-loop system to improve the stability of standard RRT, where planning feasibility is guaranteed by simulations with the low-level controller in advance. The deterministic sampling-based method proposed by Macfarlane and Croft[12] sampled in a predetermined acceleration range, resulting in jerk-bounded trajectories. Gu et al.[13] further improved the smoothness of the spatio-temporal trajectory by branching over the polynomial speed profiles sampled deterministically in a short term. However, in the case of small sample size, the sampling-based methods can generate trajectories far from the global optimum. Otherwise, the planning is intractable and time-consuming due to the curse of dimensionality. Besides, the collision check does not actually consider the dynamic obstacles parameterized by time, probably leading to poor feasibility of the spatio-temporal trajectory.

Graph-search-based methods search the desired trajectory under some performance requirements on a discretized map. Originally, those search methods can only deal with static obstacles. But currently, more and more research works have focused on the generation of the spatio-temporal trajectory with a consideration of dynamic obstacles in the future. State lattice proposed by Pivtoraiko et al.[14] is the most widely used discrete graphs composed of kinematically feasible motion primitives and time information. The spatio-temporal trajectory can be obtained using classical graph-search-based methods in such a state lattice.[15] The Dijkstra algorithm is widely used for finding the shortest trajectory.[16] A*, an extension of the Dijkstra algorithm, is a fast heuristic search algorithm and the basis of many implementations, such as the Junior (hybrid A*)[17] and Boss (AD*)[18] autonomous vehicles. To search more efficiently in a changing environment, D*[19] and D* Lite[20] are proposed to replan the trajectory using previous information. Although the graph-search-based methods generate the trajectory near the global optimum, its higher order derivatives are still discontinuous. However, the creation of state lattice takes a lot of time, leading to low planning efficiency.[21]

Optimization-based methods are concerned with mathematical problem-solving to obtain the minimal cost or objective function with multiple constraints. The trajectory is iteratively refined until the termination or convergence conditions are met.[22] Shibata et al.[23] employed the potential field approach to generate the obstacle-avoidance trajectory based on velocity vectors. They determined the optimal parameter values for the velocity potential function by a numerical optimization method called quasi-Newton. Keller et al.[24] represented the spatio-temporal trajectory by a timed elastic band (TEB) function and optimized it in constrained action space. They formulated the problem as a nonlinear least-squares problem and solved it using the Levenberg–Marquardt algorithm, a numerical optimization method. Gao et al.[25] applied a local optimization-based method called model predictive control (MPC) to smooth a coarse predefined trajectory. They minimized the cost function by sequential quadratic programming (SQP), where the problem was divided into a sequence of quadratic programming (QP) subproblems with linearized constraints. However, the optimization-based methods are often time-consuming, especially with non-convex environmental constraints.

Overall, current research works on trajectory planning provide insightful progress but still have limitations. Sampling-based methods can alleviate local optimality problem by exploring the whole space and finding a global optimum. However, it suffers from curse of dimensionality and high computational expense, leading to resolution suboptimality. Graph-search-based methods can effectively generate the trajectory near the global optimum, but it is usually not curvature continuous and does not satisfy model feasibility for trajectory execution. Optimization-based methods can guarantee feasibility with model information and are computationally acceptable by finding local optimum through a few iterations. However, the predetermined reference trajectory is required and needs proper design and initialization.

Therefore, this paper proposes a novel hybrid planning framework composed of two modules, which preserves the strength of both graph-search-based methods for avoiding local optimality and optimization-based methods for local refinement, thus enabling faster and smoother spatio-temporal trajectory planning in constrained dynamic environment. The proposed method first constructs spatio-temporal driving space based on directed acyclic graph (DAG) and efficiently searches a spatio-temporal trajectory using the improved A* algorithm. Then taking the search result as reference, locally convex feasible driving area is designed and MPC is applied to further optimize the trajectory with a comprehensive consideration of vehicle kinematics and moving obstacles.

The rest of the paper is organized as follows. Section "Proposed method" shows the framework of the proposed planning method. Section "Reference trajectory generation" introduces the reference trajectory generation. Section "Trajectory smoothing" introduces the trajectory smoothing. Section "Simulation setup"

presents the simulation setup in detail. Section "Results" presents the results for overtaking and yielding. Section "Conclusion" concludes this paper with a summary of the contributions and perspectives.

Proposed method

To generate the smooth spatio-temporal trajectory efficiently in constrained dynamic environment, a hybrid planning method is proposed in this paper. As shown in Figure 1, its framework consists of two procedures: reference trajectory generation with a search-based method and local refinement and smoothing with an optimization-based method:

1. *To generate a reference trajectory with a search-based method.* The driving map is represented by a three-dimensional (3D) spatio-temporal map instead of the general grid map, which is constructed by DAG. The position of dynamic obstacles in the future can be illustrated in such a driving map. Then, the improved A* algorithm is used to search in action space to efficiently get the best collision-free trajectory depending on the designed performance requirements.
2. *To smooth the reference trajectory with an optimization-based method.* The reference trajectory is not smooth enough due to the discrete action space and state space. Therefore, MPC is adopted for further optimization. However, the presence of dynamic obstacles generally brings non-convex feasible driving areas for the ego vehicle. It increases the computational difficulty and influences the whole planning efficiency. Then, the method for reconstruction of convex feasible sets is presented previously to ensure a fast optimization. A kinematic model is introduced to make sure the higher order derivatives of the trajectory are continuous and accordant with the vehicle kinematics. Overall, the optimized trajectory is smooth enough, leading to good tracking performance and reasonable control cost.

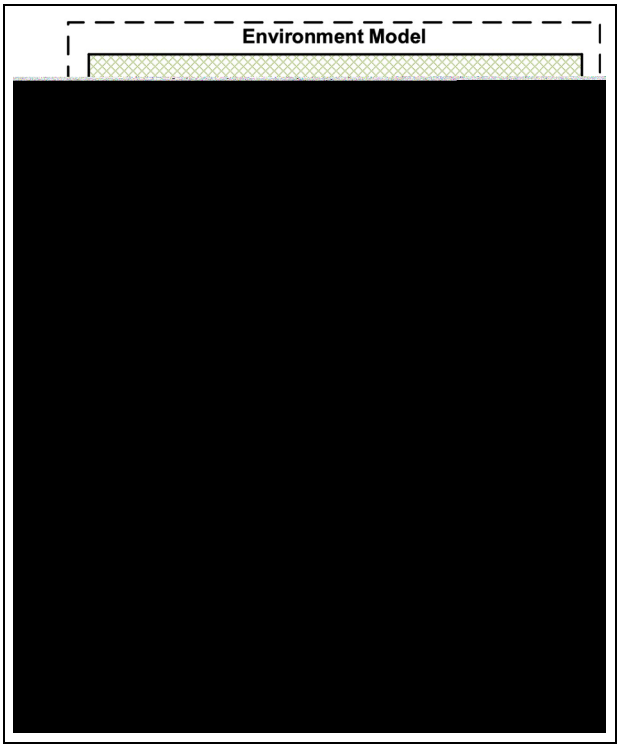

Figure 1. Framework of the proposed planning method.

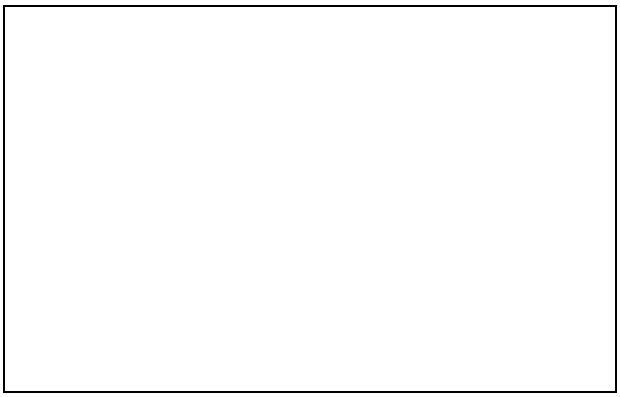

Figure 2. Two-dimensional spatial grid map for a two-lane road.

Reference trajectory generation

This section presents the generation of the reference trajectory, which contains the spatio-temporal information toward the destination in dynamic environment. The driving map represented by a 3D spatio-temporal map is first established with the help of DAG. Then, the improved A* algorithm is used to search in action space to efficiently get the best collision-free trajectory on the driving map.

Spatio-temporal map construction

The spatio-temporal trajectory provides the ego vehicle not only the position but also the kinematic information, leading to the good tracking performance of the vehicle controller. To generate a reasonable trajectory in the presence of moving obstacles, their positions are necessary to be parameterized by time. However, graph-search-based algorithms generally do not take into account their positions in the future due to the lack of temporal information. Therefore, a timeline is added into the grid map, and then the 3D spatio-temporal map is established with the help of DAG.

DAG consists of the points and the edges directed from one point to another along which it is impossible for any point to loop back to itself.[21,26] A sequence of points in such topological ordering represents the change of vehicle states over time, namely, the spatio-temporal trajectory. Then the grid map with directed

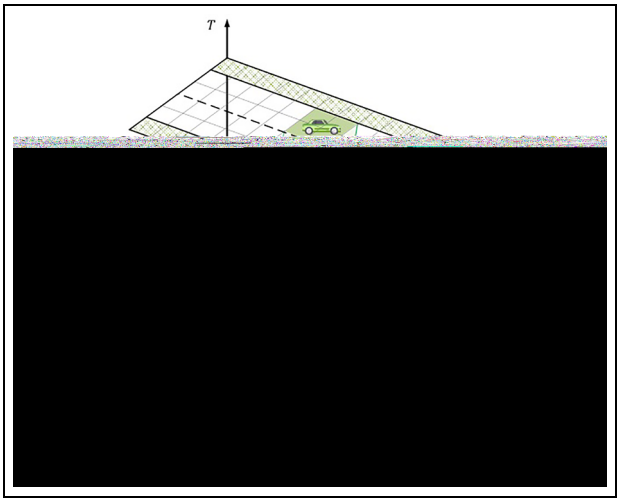

Figure 3. Three-dimensional spatio-temporal map constructed by DAG.
DAG: directed acyclic graph.

acyclic edges connecting the neighboring points is exactly a DAG.

First, the two-dimensional (2D) spatial grid map is established by dividing the environment space into finitely many squares with the same size. As shown in Figure 2, the ego vehicle is considered as a mass point occupying the grid where its geometric center is located. The girds occupied by other vehicles take the vehicle size into account so that the lateral and longitudinal distances from the ego vehicle are large enough to avoid collisions. To improve search efficiency, there are some preliminaries concerned with the elimination of useless squares. It is impossible for the ego vehicle to go through those areas without collision even with the maximum steering angle: for example, the circular areas following the neighboring vehicles and those before the destination, which are depicted as gray-shaded parts. The radius of the areas, namely the minimum turning radius for the ego vehicle, is tangent to the direction of vehicle velocity.

The 2D spatial grid map is extended along the added timeline, resulting in a 3D spatio-temporal map. Its multiple map layers are parallel to each other as shown in Figure 3. Several directed edges connect the two states in neighboring map layers at different time steps. The edge is in a single direction as time goes forward, which meets the requirements of the DAG. In Figure 3, a sequence of points connected by several solid blue lines illustrates the spatio-temporal trajectory discretized with a time step size Δt. Such a trajectory contains both the position and the time information. Specifically, it can represent a stop at a specific time, while it is impossible for general grid maps.

In the 3D spatio-temporal map, the positions for the dynamic obstacle at different time steps are depicted as the light green part in Figure 3. The future position of surrounding vehicles is assumed to be given by trajectory prediction, which is another vital module in the decision-making system.[27] Then, in the future, the proposed planning method can take the dynamic unfeasible region into account, while the classical graph-search-based methods based on the grid map only consider the static obstacles in each search step. The reference trajectory without collisions is, thus, feasible in a longer time horizon. Since the good suitability and scalability of the 3D spatio-temporal map established offline, only the unfeasible squares are specified from scenarios to scenarios. The unfeasible squares can be eliminated quickly so that the increase in obstacle numbers has little impact on overall processing time.

Moreover, the established 3D spatio-temporal map contains information not only about the vehicle states $\boldsymbol{s}$ but also about the actions $\boldsymbol{A}$, which are depicted as the blue directed lines. The slope of its projection in plan XOT denotes the lateral velocity $v_x = \Delta l_x/\Delta t$, while the slope of its projection in plan YOT denotes the longitudinal velocity $v_y = \Delta l_y/\Delta t$. The angle between its projection on the plan XOY and the axis Y approximates the yaw angle. Those actions are calculated depending on the determined neighboring states. In turn, the possible states can be expanded from the current state if the action space is determined (the blue solid line and the dotted line in Figure 3). Note that it is possible to adopt different curves to connect the two states at adjacent map layers. Since the time step is assumed to be short enough, the velocity is assumed constant. Then, the straight lines are used in this paper and the spatio-trajectory of the ego vehicle is depicted as the light blue part.

Trajectory search algorithm

One of the graph-search-based methods called A* algorithm is introduced to find the best reference path because of its high efficiency. It usually deals with situations with fixed obstacles.[28] The basic A* algorithm used for a grid configuration space is restricted to the spatial trajectory. Then, in this paper, the A* algorithm is improved to be applicable on the established 3D spatio-temporal map.

In this paper, the vehicle states are the collection of the lateral and longitudinal position x, y, velocity v, and yaw angle θ, which is denoted as $\boldsymbol{s}_t = \{x_t, y_t, v_t, \theta_t\}$. The discrete action space includes possible acceleration and yaw angle, which is denoted as $\mathcal{A} = \{\boldsymbol{A}, \boldsymbol{\Theta} | \boldsymbol{A} = \cup_i a_i, \boldsymbol{\Theta} = \cup_i \theta_i\}$. The expansion for ego vehicle states is formulated in equation (1)

$$\begin{aligned} \theta_{t+1} &= \theta_t + \theta_i \\ v_{t+1} &= v_t + a_i \times \Delta t \\ x_{t+1} &= x_t + v_{t+1} \times \Delta t \times \cos\theta_{t+1} \\ y_{t+1} &= y_t + v_{t+1} \times \Delta t \times \sin\theta_{t+1} \end{aligned} \tag{1}$$

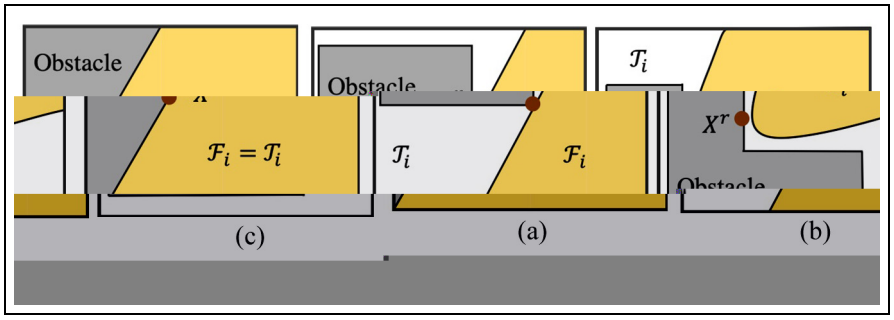

Figure 4. Three types of convex feasible sets: (a) Γ_i is convex, (b) $R\backslash\Gamma_i$ is convex, and (c) neither Γ_i nor $R\backslash\Gamma_i$ is convex.

$$\phi_i(X) \geqslant \phi_i(X^r) + \Delta\phi_i(X^r)\times(X - X^r) \tag{6}$$

Then the convex feasible superset is defined as

$$\mathcal{F}_i(X^r) := \{X|\phi_i(X^r) + \nabla\phi_i(X^r)\times(X - X^r) \geqslant 0\} \tag{7}$$

3. If neither Γ_i nor its complement $R\backslash\Gamma_i$ is convex, function ϕ_i is neither convex nor concave. An auxiliary function is adopted as

$$\widetilde{\phi}_i(X) : = \phi_i(X) + \frac{1}{2}(X - X^r)^T H_i(X - X^r) \tag{8}$$

where $\widetilde{\phi}_i(X)$ is convex so that

$$\begin{aligned}\widetilde{\phi}_i(X) &\geqslant \widetilde{\phi}_i(X^r) + \nabla\widetilde{\phi}_i(X)\times(X - X^r)\\ &= \phi_i(X^r) + \nabla\phi_i(X)\times(X - X^r)\end{aligned} \tag{9}$$

Then the convex feasible superset is defined as

$$\begin{aligned}\mathcal{F}_i(X^r) : = \Bigg\{X|\phi_i(X^r) + \nabla\phi_i(X)\times(X - X^r) \geqslant \frac{1}{2}\\ (X - X^r)^T H_i(X - X^r)\Bigg\}\end{aligned} \tag{10}$$

As shown in Figure 4, all the constraints of dynamic obstacles to reference trajectory can be reconstructed as the above-mentioned three types of convex feasible sets, which guarantees high efficiency in the following optimization step.

Trajectory optimization method

The reference trajectory is optimized by MPC, and the problem is formulated as

$$\begin{aligned}\min_{U_n}\quad & \sum_{n=1}^{T} \omega_1\times\left\|X_n - X_n^r\right\|_Q^2 + \omega_2\times\|U_n\|_R^2\\ \text{s.t.}\quad & X_0 = X^{\text{start}}, X_N = X^{\text{target}}\\ & X_n = X_{n-1} + F(X_{n-1}, U_{n-1})\times\Delta t\\ & U_{\min} \leqslant U_n \leqslant U_{\max}\\ & \phi(X_n) \geqslant 0, \forall n = 1, \ldots, N-1\end{aligned} \tag{11}$$

where T denotes the prediction horizon, X^{start} denotes the initial position, X^{target} denotes the destination, X_n^r denotes the reference state of the spatio-temporal trajectory, U_n denotes the control input with limits of $(U_{\min}, U_{\max})$, Q and R are the weighting matrices for each component of states and control inputs, ω_1 and ω_2 are the weights on two different objectives, $F(\cdot)$ is the vehicle state-space model, and $\phi(\cdot)$ denotes the state constraints for each state.

Expand the nonlinear function $F(X_n, U_n)$ as a Taylor series around the reference point

$$F(X_n, U_n) : = F(X_n^r, U_n) + \text{Jac}_F(X_n, U_n)\times(X_n - X_n^r) \tag{12}$$

where $\text{Jac}_F(\cdot)$ is the Jacobian matrix of $F(\cdot)$. Then, the state-space model is linearized as

$$\begin{aligned}X_n &= X_{n-1} + F(X_{n-1}, U)\times\Delta t\\ &\triangleq A_n X_{n-1} + B_n U_{n-1} + C_n\end{aligned} \tag{13}$$

where A_n, B_n, and C_n are the time-invariant matrices.

The problem-solving is time-consuming if the dynamic constraints ϕ are concave. With the convex feasible set construction method introduced previously, problem (11) is reformulated as

$$\begin{aligned}\min_{U_n}\quad & \sum_{n=1}^{T} \omega_1\times\left\|X_n - X_n^r\right\|_Q^2 + \omega_2\times\|U_n\|_R^2\\ \text{s.t.}\quad & X_0 = X^{\text{start}}, X_N = X^{\text{target}}\\ & X_n = X_{n-1} + A_n X_{n-1} + B_n U_{n-1} + C_n\\ & U_{\min} \leqslant U_n \leqslant U_{\max}\\ & X_n \in \mathcal{F}(X_n^r), \forall n = 1, \ldots, N-1\end{aligned} \tag{14}$$

Simulation setup

Scenario

The proposed trajectory planning method is demonstrated in different scenarios based on two-lane roads with a lane width of 4 m and a length of 100 m. A dotted line separates the two lanes, which allows vehicles to cross. The ego vehicle is driving on the downside lane while the other two vehicles are in different lanes,

respectively. The surrounding vehicle on the upside land drives in the same or opposite direction depending on specific situations. The map is homogeneously discretized by regular squares with a size of $0.2\,\mathrm{m} \times 0.2\,\mathrm{m}$, leading to the facility of meshing and the efficiency of computation. A* algorithm is applied to search the constructed spatio-temporal map in the discrete action space, where $\boldsymbol{A} = \{-3.5, -1.75, 0, 1.75, 3.5\}\,\mathrm{m/s^2}$ and $\boldsymbol{\Theta} = \{-\pi/2, -\pi/4, 0, \pi/4, \pi/2\}$ (see Figure 3).

Cost function in reference trajectory search

The cost function $Cost(\boldsymbol{s}_t)$ of the improved A* algorithm includes mainly two parts: the accumulated cost $J(\boldsymbol{s}_t)$ and the heuristic function $H(\boldsymbol{s}_t)$. In this paper, the accumulated cost $J(\boldsymbol{s}_t, a_t)$ to move from the initial state to the current state $\boldsymbol{s}_t$ consists of four indexes, namely, efficiency E, safety S, comfort C, and reasonableness D

$$J(\boldsymbol{s}_t) = w_E E(\boldsymbol{s}_t) + w_S S(\boldsymbol{s}_t) + w_C C(\boldsymbol{s}_t) + w_D D(\boldsymbol{s}_t) \tag{15}$$

where w_E, w_S, w_C, and w_D are the corresponding weights on the four indexes.

The efficiency E evaluates the search speed of the 3D spatio-temporal map, which is defined as the displacement in a step

$$E(\boldsymbol{s}_t) = \{(x_t - x_{t+1})^2 + (y_t - y_{t+1})^2\}^{1/2} \tag{16}$$

The safety index S is designed as the change of artificial potential field (APF) value

$$S(\boldsymbol{s}_t) = APF_{Node_t} - APF_{Node_{t-1}} \tag{17}$$

where the higher the APF value, the safer the ego vehicle.[32] It is calculated depending on the distance that the ego vehicle keeps from the obstacles and their speed difference, as defined in the following equation

$$\begin{aligned} APF &= \frac{c_1}{\left(\frac{\Delta d}{\Delta v + c_2} - c_3\right)^{c_4}} \\ \Delta d &= y_{\mathrm{front}} - y_{\mathrm{ego}} \\ \Delta v &= v_y - v_y^{\mathrm{front}} \end{aligned} \tag{18}$$

where c_1, c_2, c_3, and c_4 are the parameters, Δd denotes the difference in longitudinal distance between the ego car and the front vehicle, and Δv denotes their difference in longitudinal velocity.

The comfort C is defined as the deviation of longitudinal velocity from the desired value

$$C(\boldsymbol{s}_t) = |v_t - v_{\mathrm{desired}}| \tag{19}$$

The reasonableness D is defined as the deviation

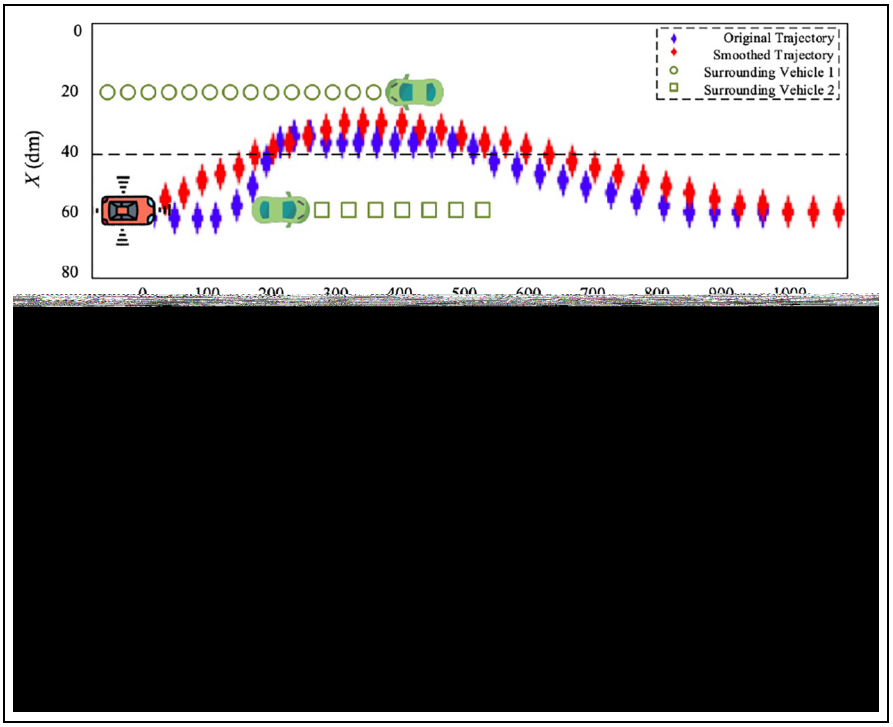

Figure 7. Simulation results for scenario II where the ego vehicle overtakes a fast oncoming vehicle: (a) planned spatio-temporal trajectory, (b) planned speed profile, and (c) planned front-wheel steering angle.

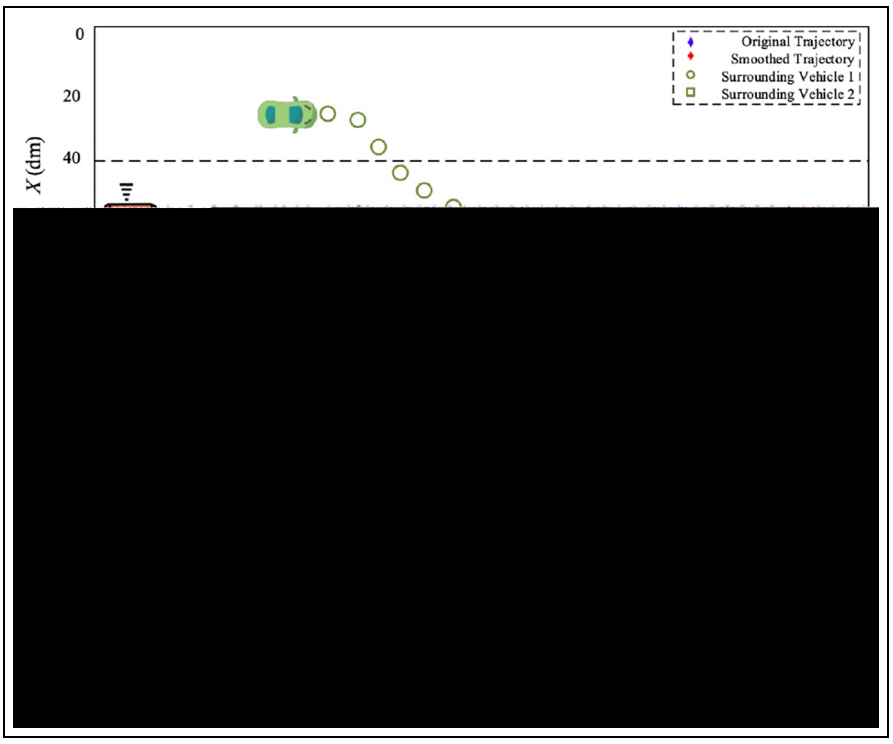

Figure 8. Simulation results for scenario III where the ego vehicle yields to a sudden cut-in vehicle: (a) planned spatio-temporal trajectory, (b) planned speed profile, and (c) planned front-wheel steering angle.

The last scenario is yielding courteously to a pedestrian as shown in Figure 9. The ego vehicle starts from a fixed position (0 m, 6 m) and ends at the destination (100 m, 6 m). It drives along the road at an initial speed of 15 m/s with a limit of 20 m/s. A pedestrian is crossing the road from an initial position (55 m, 6 m) at a speed of 1 m/s, which lasts for 1–7 s. In theory, the ego vehicle could have taken a lane change to overtake the

Figure 9. Simulation results for scenario IV where the ego vehicle yields to a pedestrian: (a) planned spatio-temporal trajectory, (b) planned speed profile, and (c) planned front-wheel steering angle.

Table 1. Comparison of planning time.

Scenario	Time (ms)			
	Our method	Chen et al.[34]	Xu et al.[35]	Ziegler and Stiller[22]
Aggressive overtaking	28	51	160	4519
Courtesy yielding	31	52	165	4524
Emergency cut-in	30	52	162	4520
Courtesy stop	35	55	170	4527

pedestrian, but it stops to wait for the pedestrian to pass by and then continues driving toward the end. This is because the safety index is assigned a huge weight in the cost function. The optimization method is applied to smooth the reference trajectory, leading to a smooth speed reduction rather than an emergency stop.

Furthermore, the proposed planning method is also evaluated in terms of computational cost. Three other methods are introduced for comparison. The first method, maximum interaction defensive policy (MIDP), combines the sampling-based and optimization-based methods. It generates the spatio-temporal trajectory by connecting the current vehicle states and the next states through the optimal speed profile. The optimal speed profile is selected according to an optimization problem among a set of speed profiles sampled by cubic polynomials.[33] Another method called sampling-based method with iterative optimizations (SAIO) is based on sampling-based methods with post-optimization, where the trajectory is connected by speed profile sampled by quartic polynomials and the optimization is iterative.[34] The last method called Graph-search-based method with Dynamic Programming (GADP) is a combination of graph-search-based and optimization-based methods. It selects spatio-temporal trajectory on the state lattices using the dynamic programming (DP) algorithms.[21]

Table 1 shows the total computation time consumed by different planning methods. It can be seen from the table that the proposed method only took 31 ms on average, which is only 59.05%, 18.87%, and 0.69% of that consumed by MIDP, SAIO, and GADP, respectively.

Conclusion

This paper proposed a hybrid framework for fast and smooth spatio-temporal trajectory planning for autonomous vehicles in constrained dynamic environment.

First, a spatio-temporal driving map is constructed using DAG, which can illustrate both search space and future positions of dynamic obstacles. Based on the driving map, a spatio-temporal trajectory is efficiently searched through action space with the help of the improved A* algorithm. Then, taking the search result as the reference trajectory, convex feasible driving area for autonomous vehicles considering surrounding vehicles is designed and MPC is applied for model feasibility. The proposed framework takes a comprehensive consideration of key advantages of both graph-search-based and optimization-based trajectory planning methods while reducing their limitations. Simulation results show that it only takes about 31 ms on average for planning, while the planned trajectory, velocity, and steering angle curves are reasonable and continuous. Specifically, there is neither an emergency stop nor an abrupt steering change in all simulated scenarios. Compared to the state-of-the-art methods, the proposed method has the following advantages: (1) be able to generate spatio-temporal trajectory rather than just geometrical path, (2) be able to generate trajectory which is kinematically smooth to follow, and (3) enjoy a much higher efficiency while guaranteeing optimum.

This study provides a promising approach for fast and smooth trajectory planning for autonomous vehicles in constrained dynamic environment. The preliminary research opens various perspectives for future research: (1) to enhance the proposed method in order to deal with multiple stochastic moving obstacles, (2) to consider the interaction between the ego vehicle and the surrounding vehicles, and (3) to generalize the proposed method in different driving scenarios, for example, intersections and unstructured roads.

Declaration of conflicting interests

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding

The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: This study was supported by the National Key R&D Program of China (grant no. 2016YFB0100906) and NSF China (grant nos 51575293, U1664263, and 51622504).

ORCID iD

Yiting Kong https://orcid.org/0000-0002-7812-1457

References

1. Ni J, Hu J and Xiang C. Robust control in diagonal move steer mode and experiment on an X-by-wire UGV. *IEEE/ASME T Mech* 2019; 24: 572–584.
2. Biao X, Eben LS, Yougang B, et al. Distributed conflict-free cooperation for multiple connected vehicles at unsignalized intersections. *Transport Res C*: Emer 2018; 93: 322–334.
3. Liu C, Li SE and Hedrick JK. Measurement dissemination-based distributed Bayesian filter using the latest-in-and-full-out exchange protocol for networked unmanned vehicles. *IEEE T Ind Electron* 2017; 64: 8756–8766.
4. Ni J, Zhou J, Hu J, et al. Robust path following control at driving/handling limits of an autonomous electric racecar. *IEEE T Veh Technol* 2019; 68: 5518–5526.
5. Li SE, Guo Q, Xu S, et al. Performance enhanced predictive control for adaptive cruise control system considering road elevation information. *IEEE T Intell Veh* 2017; 2: 150–160.
6. Ni J, Wang W, Hu J, et al. Relaxed static stability for four-wheel independently actuated ground vehicle. *Mech Syst Signal Pr* 2019; 127: 35–49.
7. Li SE, Gao F, Li K, et al. Robust longitudinal control of multi-vehicle systems—a distributed H-infinity method. *IEEE T Intell Transp* 2017; 19: 2779–2788.
8. Li SE, Li G, Yu J, et al. Kalman filter-based tracking of moving objects using linear ultrasonic sensor array for road vehicles. *Mech Syst Signal Pr* 2018; 98: 173–189.
9. Frazzoli E, Dahleh MA and Feron E. Real-time motion planning for agile autonomous vehicles. *J Guid Control Dynam* 2002; 25: 116–129.
10. LaValle SM. *Rapidly-exploring random trees: a new tool for path planning*. Technical report, TR98-11, 1998. Ames: Iowa State University (Department of Computer Science).
11. Kuwata Y, Teo J, Fiore G, et al. Real-time motion planning with applications to autonomous urban driving. *IEEE T Contr Syst T* 2009; 17: 1105–1118.
12. Macfarlane S and Croft EA. Jerk-bounded manipulator trajectory planning: design for real-time applications. *IEEE T Robotic Autom* 2003; 19: 42–52.
13. Gu T, Dolan JM and Lee JW. Automated tactical maneuver discovery, reasoning and trajectory planning for autonomous driving. In: *Proceedings of the 2016 IEEE/RSJ international conference on intelligent robots and systems*, Daejeon, South Korea, 9–14 October 2016, pp.5474–5480. New York: IEEE.
14. Pivtoraiko M, Knepper RA and Kelly A. Differentially constrained mobile robot motion planning in state lattices. *J Field Robot* 2009; 26: 308–333.
15. Kushleyev A and Likhachev M. Time-bounded lattice for efficient planning in dynamic environments. In: *Proceedings of the 2009 IEEE international conference on robotics and automation*, Kobe, Japan, 12–17 May 2009, pp.1662–1668. New York: IEEE.
16. Bohren J, Foote T, Keller J, et al. Little Ben: the Ben Franklin racing team's entry in the 2007 DARPA urban challenge. *J Field Robot* 2008; 25: 598–614.
17. Montemerlo M, Becker J, Bhat S, et al. Junior: the Stanford entry in the urban challenge. *J Field Robot* 2008; 25: 569–597
18. Ferguson D, Howard TM and Likhachev M. Motion planning in urban environments. *J Field Robot* 2008; 25: 939–960.
19. Rufli M and Siegwart R. On the application of the D* search algorithm to time-based planning on lattice graphs. In: *Proceedings of the 4th European conference on mobile robotics*, Mlini/Dubrovnik, Croatia, 23–25

September 2009, pp.105–110. Zürich: Eidgenössische Technische Hochschule.
20. Koenig S and Likhachev M. Fast replanning for navigation in unknown terrain. *IEEE T Robot* 2005; 21: 354–363.
21. Ziegler J and Stiller C. Spatiotemporal state lattices for fast trajectory planning in dynamic on-road driving scenarios. In: *Proceedings of the 2009 IEEE/RSJ international conference on intelligent robots and systems*, St. Louis, MO, 11–15 October 2009, pp.1879–1884. New York: IEEE.
22. Gu T. *Improved trajectory planning for on-road self-driving vehicles via combined graph search, optimization & topology analysis*. PhD Thesis, Carnegie Mellon University, Pittsburgh, PA, 2017.
23. Shibata N, Sugiyama S and Wada T. Collision avoidance control with steering using velocity potential field. In: *Proceedings of the 2014 IEEE intelligent vehicles symposium*, Dearborn, MI, 8–11 June 2014, pp.438–443. New York: IEEE.
24. Keller M, Hoffmann F, Hass C, et al. Planning of optimal collision avoidance trajectories with timed elastic bands. *IFAC Proc Vol* 2014; 47: 9822–9827.
25. Gao Y, Lin T, Borrelli F, et al. Predictive control of autonomous ground vehicles with obstacle avoidance on slippery roads. In: *Proceedings of the ASME 2010 dynamic systems and control conference*, pp.265–272. New York: American Society of Mechanical Engineers.
26. Karampiperis P and Sampson D. Adaptive instructional planning using ontologies. In: *Proceedings of the 2004 IEEE international conference on advanced learning technologies*, Joensuu, 30 August–1 September 2004, , vol. 1, Cambridge, MA, 12–15 September 2010, pp.126–130. New York: IEEE.
27. Xin L, Wang P, Chan CY, et al. Intention-aware long horizon trajectory prediction of surrounding vehicles using dual LSTM networks. In: *Proceedings of the 2018 21st international conference on intelligent transportation systems*, Maui, HI, 4–7 November 2018, pp.1441–1446. New York: IEEE.
28. Duchoň F, Babinec A, Kajan M, et al. Path planning with modified a star algorithm for a mobile robot. *Procedia Engineer* 2014; 96: 59–69.
29. Ni J, Hu J and Xiang C. Envelope control for four-wheel independently actuated autonomous ground vehicle through AFS/DYC integrated control. *IEEE T Veh Technol* 2017; 66: 9712–9726.
30. Liu C, Lin CY and Tomizuka M. The convex feasible set algorithm for real time optimization in motion planning. *SIAM J Control Optim* 2018; 56: 2712–2733.
31. Liu C, Lin CY, Wang Y, et al. Convex feasible set algorithm for constrained trajectory smoothing. In: *Proceedings of the 2017 American control conference*, Seattle, WA, 24–26 May 2017, pp.4177–4182. New York: IEEE.
32. Bing H, Gang L, Jiang G, et al. A route planning method based on improved artificial potential field algorithm. In: *Proceedings of the 2011 IEEE 3rd international conference on communication software and networks*, Xi'an, China, 27–29 May 2011, pp.550–554. New York: IEEE.
33. Chen J, Tang C, Xin L, et al. Continuous decision making for on-road autonomous driving under uncertain and interactive environments. In: *Proceedings of the 2018 IEEE intelligent vehicles symposium*, vol. 68, Changshu, China, 26–30 June 2018, pp.1651–1658. New York: IEEE.
34. Xu W, Wei J, Dolan JM, et al. A real-time motion planner with trajectory optimization for autonomous vehicles. In: *Proceedings of the 2012 IEEE international conference on robotics and automation*, Saint Paul, MN, 14–18 May 2012, pp.2061–2067. New York: IEEE.