Mechanical Systems and Signal Processing 100 (2018) 482–500

Contents lists available at ScienceDirect

Mechanical Systems and Signal Processing

journal homepage: www.elsevier.com/locate/ymssp

Dynamic path planning for autonomous driving on various roads with avoidance of static and moving obstacles

Xuemin Hu [a], Long Chen [b,*], Bo Tang [c], Dongpu Cao [d], Haibo He [e]

[a] School of Computer Science and Information Engineering, Hubei University, Wuhan, Hubei 430062, PR China
[b] School of Data and Computer Science, Sun Yat-sen University, Guangzhou, Guangdong 510275, PR China
[c] Department of Electrical and Computer Engineering, Mississippi State University, MS 39762, USA
[d] Advanced Vehicle Engineering Center, Cranfield University, Cranfield, Bedfordshire MK430AL, UK
[e] Department of Electrical, Computer, and Biomedical Engineering, University of Rhode Island, Kingston, RI 02881, USA

ARTICLE INFO

Article history:
Received 7 April 2017
Received in revised form 30 June 2017
Accepted 14 July 2017

Keywords:
Path planning
Path generation
Path selection
Obstacle avoidance
Autonomous driving

ABSTRACT

This paper presents a real-time dynamic path planning method for autonomous driving that avoids both static and moving obstacles. The proposed path planning method determines not only an optimal path, but also the appropriate acceleration and speed for a vehicle. In this method, we first construct a center line from a set of predefined waypoints, which are usually obtained from a lane-level map. A series of path candidates are generated by the arc length and offset to the center line in the $s-\rho$ coordinate system. Then, all of these candidates are converted into Cartesian coordinates. The optimal path is selected considering the total cost of static safety, comfortability, and dynamic safety; meanwhile, the appropriate acceleration and speed for the optimal path are also identified. Various types of roads, including single-lane roads and multi-lane roads with static and moving obstacles, are designed to test the proposed method. The simulation results demonstrate the effectiveness of the proposed method, and indicate its wide practical application to autonomous driving.

1. Introduction

Autonomous driving has been the focus of significant interest from academia, industry, and the military in recent years [1–3]. Demonstrations on California highways in 1997 proved that autonomous driving was possible in cases restricted to a dedicated lane on a highway with magnets [4,5]. The Defense Advanced Research Projects Agency (DARPA) Grand Challenge and Urban Challenge have stimulated research interest in the field. Moreover, Google driverless cars have been tested on more than 1,600,000 miles of American roads. Tesla has equipped its MODEL S car with a self-driving system "autopilot system. Although Google cars, Teslas, and vehicles developed for the DARPA challenges have shown great autonomous driving performance, significant challenges still remain for the commercialization of autonomous vehicles in terms of technology and price [6,7]. There are many issues associated with autonomous driving systems in practical applications. However, these issues can be solved in the near future [8,9].

Current research on autonomous driving encompasses different fields, including perception, planning, and control [10–12]. The goal of perception is to obtain information for autonomous vehicles from their environments [13], while the goal

* Corresponding author.
E-mail address: chenl46@mail.sysu.edu.cn (L. Chen).

http://dx.doi.org/10.1016/j.ymssp.2017.07.019
0888-3270/

of control is to determine the appropriate parameters for veer, throttle, or brake systems in order to make the autonomous vehicles follow the planned path [14]. Planning is the decision making stage between perception and control. In particular, critical decision making in autonomous driving is the key to autonomy. This can be realized through planning algorithms incorporated into the middleware of the navigation system of an autonomous vehicle, situation understanding, and decision making modules. The main purpose of planning is to provide vehicles with a safe and collision-free path towards their destinations, accounting for vehicle dynamics, maneuvering capabilities in the presence of obstacles, and traffic rules and road boundaries [15,16]. Path planning methods are typically studied to allow mobile robots and autonomous vehicles to avoid obstacles [17,18]. Existing path planning algorithms can be divided into two stages: global planning and local planning. In the global planning stage, global routes and the vehicle states are determined from a digital map and localization system. In the local planning stage, a local path can be achieved based on a global route and surrounding information obtained from sensors such as cameras or radars [4,7]. In this paper, we focus on designing a local path planning method for autonomous vehicles on the basis of a predefined global route.

In recent years, many studies have been conducted on dynamic path planning. These studies can be divided into four primary categories: grid-based approaches, potential field approaches, sample-based approaches, and discrete optimization approaches.

In grid-based approaches, the environment is mapped to a set of cells, where each cell represents the presence of an obstacle at that position in the environment [19]. Optimal search algorithms such as A$*$ and D$*$ are typically used to find the global optimal path that connects the initial position of each cell to the goal position, while also avoiding obstacles. In recent years, some improved grid-based methods treating nonholonomic constraints have been proposed, by adapting applications such as executing U-turns on a blocked road, parking, and navigating roads in unstructured environments [20,21]. Path planning for first responders in the presence of uncertain moving obstacles based on algorithm A$*$ is proposed in [22]. Zuo developed a novel hierarchical path planning approach for mobile robot navigation in complex environments [23]. This type of method has all future path information known after planner execution and before vehicle motion [24]. However, the incremental nature of search algorithms causes problems with the exponential growth of computing complexity. Grid-based approaches perform well for path planning in low-speed applications, but they are not suitable for high-speed driving.

In potential field approaches, the repulsive forces to obstacles and attractive forces to the goal position are virtually assigned [25,26]. The gradient of a potential field is constructed by virtual forces. A path can then be achieved along the steepest gradient of the potential field. Ref. [27] introduces a path planning method based on a potential field where stream functions are used to plan the paths of autonomous vehicles. Daily proposed a harmonic potential field path planning approach for high speed vehicles in [28]. These potential-field-based approaches are advantageous, because their trajectories can be produced with little computation. However, in some scenarios the approaches can become trapped in the local minima of a potential field, in which case the obtained path is not optimal or a path may not be found.

Path planning approaches based on sampling are appropriate for planning in high dimensional spaces. In such approaches, a collision-free path from the initial position to the destination is constructed by sampling the configuration describing the position and orientation of a vehicle. Rapidly exploring random trees (RRTs) and RRT variants are widely used in nonholonomic path planning [29]. Obermeyer used this method to deal with a path planning problem for a single fixed-wing aircraft performing a reconnaissance mission [30]. In [31], an adaptive path planning algorithm was proposed for multiple AUVs to estimate the scalar field over a region of interest. Devaurs developed two efficient sampling-based approaches combining two RRT variants (RRT$*$ and T-RRT) to solve a complex path planning problem [32]. An efficient state-space sampling-based trajectory planning scheme was employed to smoothly follow the reference path in [16]. These approaches easily handle problems with obstacles and differential constraints (nonholonomic and kinodynamic), and have been widely used in autonomous robotic path planning [33,34]. Nevertheless, path planning approaches based on RRTs for real-time implementation require efficient guiding heuristics for their sampling configurations.

Path planning approaches based on discrete optimization have achieved great success in the field of autonomous driving in recent years. Methods that apply a finite set of paths can reduce the solution space; this allows for real-time implementation [35]. An efficient path planning method was developed with an autonomous vehicle in [36]. In this method, the planner defines the lateral offset as the perpendicular distance to a fixed-base line, enabling a vehicle to travel a road parallel to the base line. To select the optimal path, the cost function penalizes running over obstacles and distance from the current road center. The authors in [37] proposed a similar path planning approach for autonomous vehicles to avoid obstacles on roads. Phung proposed an enhanced discrete particle swarm optimization path planning for unmanned aerial vehicle (UAV) vision-based surface inspection [38]. These methods can provide a safe and smooth path for autonomous vehicles or UAVs. However, it can only treat static obstacles. Moving obstacles are not considered in these methods.

In this paper, we propose a path planning approach based on discrete optimization for autonomous vehicles. This approach can select an optimal path from a finite set of path candidates, and simultaneously determine appropriate vehicle acceleration and speed. The method uses a global route from the digital map that is obtained prior to local path planning. A center line representing the global route is constructed using a cubic spline. To generate the path candidates, a $s-\rho$ coordinate system is introduced and the directional information for the global route is blended with the maneuvering of the vehicle by adjusting the lateral offset to the center line. A novel cost function is designed and used to select an optimal path from multiple path candidates. The proposed method in this paper considers dynamic safety costs and allows a vehicle to avoid moving obstacles. Appropriate acceleration and speed for every planning can be simultaneously provided for autonomous

vehicles. Moreover, many existing methods, such as [37], are only used for single-lane roads such as country roads or racetracks; however, the proposed method in this paper can be used in cities with both single lane roads and multi-lane roads. In conclusion, the merits of the proposed method, which are highlighted by the colorful scaleboards in Fig. 1, are summarized as follows. (1) The method provides both an optimal path and an appropriate acceleration and speed for a vehicle that allows it to avoid static and moving obstacles. (2) A novel cost function is designed in this method to account for acceleration. The function causes the vehicle to change its speed to adapt to paths with different curvatures when turning, and allows the vehicle to avoid moving obstacles. (3) In the proposed method, a Gaussian convolution algorithm is developed for computing collision risks on both single-lane roads and multi-lane roads.

The remainder of this paper is organized as follows. The details of the dynamic path planning method are described in Section 2. Simulation results are given in Section 3. The conclusions of this paper are presented in Section 4.

2. Material and methods

The proposed path planning method is used to generate a safe and comfortable path (with an appropriate speed and acceleration) from an initial position towards a destination, while complying with a global route and map. Our method aims to resolve local path planning problems based on a global route and map. The global route is obtained by the high precision navigation system, and the map is downloaded from the Internet. As shown in Fig. 1, the map is composed of a set of waypoints on the road edges and topology that describes the relationships between connected roads. The process by which the map is obtained falls outside the scope of this paper; therefore, the maps used in this paper are predefined in our simulations.

Fig. 1 shows the proposed dynamic path planning method, which includes three stages: center line construction, path candidate generation, and path selection. These are performed on the basis of perceived information and the proposed algorithms. As shown in Fig. 1, the center line of the road is constructed from the center waypoints, which are the center positions of a pair of waypoints, using the method of cubic spline fitting. The path candidates, which are also described by the cubic spline, are generated by adjusting the lateral offset to the center line using the information for the current vehicle position, speed, and direction in the $s-\rho$ coordinate system. During path selection, the costs of static safety, comfortability, and dynamic safety are taken into account, and are combined with information on road edges, and static and moving obstacles for selecting the optimal path. Our method provides not only the selected path, but also the appropriate speed and acceleration for the vehicle maneuvering system. In this study, the proposed dynamic path planning algorithm is executed 15 times per second, and a new path is generated from the current vehicle position at every time step.

2.1. Center line construction

A parametric cubic spline is used to represent the road and center line, considering the continuity of the first and second derivatives of the curve and the computational complexity. Because the vehicle travels along a road that is often not a regular line, the parameters of a typical cubic spline have no physical meaning, and it is difficult to directly account for the param-

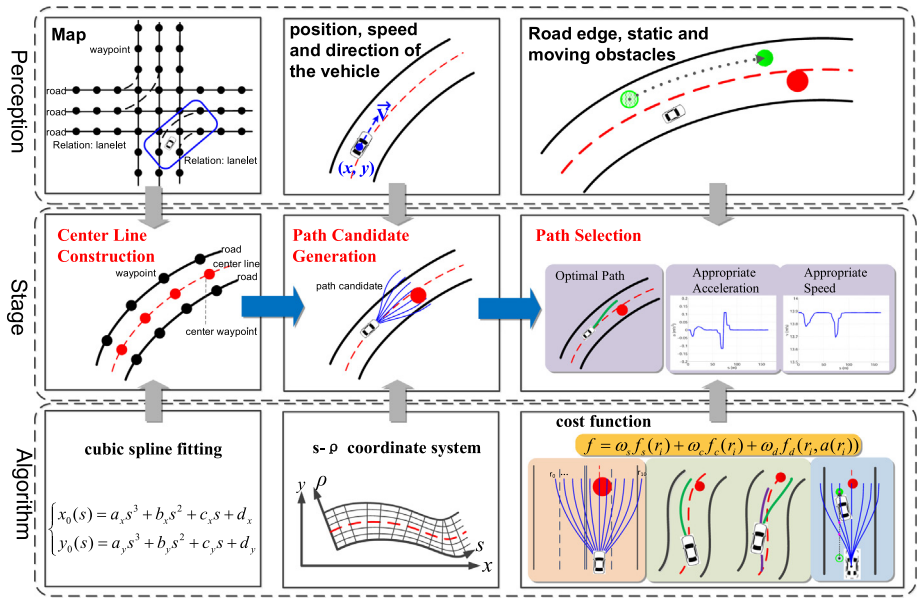

Fig. 1. Flowchart of the proposed path planning method.

eters as the representation of the center line for path planning. The vehicle trajectory is a curve and the arc length is the typical parameter for a curve. To parameterize the spline curve as the arc length, the arc lengths of all segments in the spline curve constructed from the center waypoint are numerically calculated using the method of quadrature. The spline curve describing a center line is then parameterized using the arc length of the curve [39]. Thus, the spline can be expressed as (1).

$$\begin{cases} x_0(s) = a_x s^3 + b_x s^2 + c_x s + d_x \\ y_0(s) = a_y s^3 + b_y s^2 + c_y s + d_y \end{cases} \tag{1}$$

As shown in Fig. 2(a), the center line is divided into many segments by the center waypoints, which are the centers of two waypoints. Every segment generates a cubic spline as (1), where s is the arc length for every segment, referring to the last center point as the starting point. (x_0, y_0) are the Cartesian coordinates of the point on the center line. $a_x, b_x, c_x, d_x, a_y, b_y, c_y$, and d_y are the coefficients. Through (1), the position (x_0, y_0) can be computed using the arc length s. We define a point $p_0(s, \rho)$ away from the road center at a given lateral offset ρ from the center line. In this case, every point on the road and path can be transformed from Cartesian coordinates to $s - \rho$ coordinates. In the $s - \rho$ coordinate system, the heading θ_0 and curvature κ_0 of every point on the center line can be computed using the first and second derivatives of a cubic spline with variable s. The equations are shown by (2) and (3) [40].

$$\theta_0 = \arctan \frac{dy_0}{dx_0} \tag{2}$$

$$\kappa_0 = \frac{x_0' y_0'' - x_0'' y_0'}{\sqrt{(x_0' + y_0')^3}} \tag{3}$$

where x_0', y_0', x_0'', y_0'' are the first and second derivatives of x_0 and y_0. According to (1)–(3), the coordinates, heading and curvature of any point on the center line can be found.

2.2. Path candidates generation

The path is the trajectory guiding the vehicle to follow a global route and avoid obstacles. The arc length s indicates the traveling distance on the global route, and the offset ρ can be used to measure the distance between the vehicle and an obstacle. Consequently, the proposed path planning method for avoiding obstacles is designed in the $s - \rho$ coordinate system rather than the Cartesian coordinate system.

2.2.1. Localization on the center line

To use the direction and curvature of the center line, it is necessary to find the position of the vehicle on the center line, as shown in Fig. 2(b). We first map the vehicle position from the Cartesian coordinate system to the $s - \rho$ coordinate system,

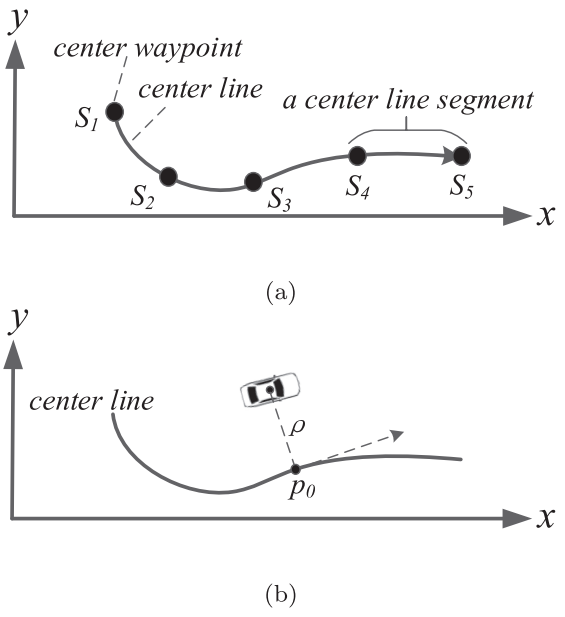

Fig. 2. Vehicle localization on the center line. (a) Center waypoints and center line segments, (b) localization on the center line.

and then determine the closest point of the center line p_0, which has the minimum distance ρ. In this paper, a method combining quadratic minimization and Newton's method is used to find p_0 [41].

2.2.2. Path candidates generation in the $s-\rho$ coordinate system

To generate path candidates, the curvature of each path is determined by the lateral offset ρ of the path, based on the curvature of the center line. As shown in Fig. 3(a), P_{init} is the original point on the center line. P_{start} and P_{end} are the start and end points on the center line, respectively, for one step of planning. P_{veh} is the start point of the vehicle. P_1 to P_5 are the end points of five path candidates, and are indicated by r_1 to r_5. It is obvious that only r_2, r_4, and r_5 are available and free of obstacles. The reason for this availability lies with the differences between the offset from the path candidate to the center line and the offset from the obstacle to the center line. Meanwhile, the positions of the obstacle and the vehicle on the center line can be expressed by the arc length s. In this case, it is necessary to design the algorithm to avoid obstacles using the parameters s and ρ. As shown in (4), a function describing the relationship between the arc length s and offset ρ is designed to provide a continuous change in the offset.

$$\rho(s)=\begin{cases} a(s-s_{start})^3+b(s-s_{start})^2+c(s-s_{start})+\rho_{start} & s\in[s_{start},s_{end}] \\ \rho_{end} \quad \texttt{others} \end{cases} \tag{4}$$

To determine the coefficients a, b, c in (4), boundary conditions are needed. In Fig. 3(b), the arc length s_{start} and offset ρ_{start} at the current position are given by finding the position on the center line. The relative angle $\Delta\theta_{start}$ is defined by the difference between the vehicle heading angle and tangent angle of the center line at the current position. Thus, the boundary conditions can be described as (5) [42].

$$\begin{aligned} &\rho(s_{start})=\rho_{start}, \quad \rho(s_{end})=\rho_{end} \\ &\frac{d\rho}{ds}(s_{start})=\tan\Delta\theta_{start}, \quad \frac{d\rho}{ds}(s_{end})=0 \end{aligned} \tag{5}$$

In (5), the boundary conditions s_{start}, ρ_{start} and $\Delta\theta_{start}$ for the current position are easily achieved. A set of coefficients a, b, c in (4) can be obtained if the offset at the end point ρ_{end} is specified. Because a set of a, b, c represents a path candidate described by (4), we can set a finite number for ρ_{end} to obtain a set of path candidates. The length of the path candidates depends on the arc length Δs between the current point and end point on the center line.

2.2.3. Coordinates conversion of path candidate points

Path candidates are generated in the $s-\rho$ coordinate system, but path planning results must be mapped into a Cartesian coordinate system to convey to the maneuvering system. Path candidate points in the Cartesian coordinate system can be represented with respect to the arc length of the center line as (6) [43].

$$\frac{dx}{ds}=A\cos\theta \qquad \frac{dy}{ds}=A\sin\theta \qquad \frac{d\theta}{ds}=A\kappa \tag{6}$$

where κ is defined as the curvature of the path candidate points, and can be calculated as (7).

$$\kappa=\frac{B}{A}\left[\kappa_0+\frac{(1-\rho\kappa_0)\left(\frac{d^2\rho}{ds^2}\right)+\kappa_0\left(\frac{d\rho}{ds}\right)^2}{A^2}\right] \tag{7}$$

In (6) and (7), A and B are defined as:

$$A=\sqrt{\left(\frac{d\rho}{ds}\right)^2+(1-\rho\kappa_0)^2} \qquad B=sgn(1-\rho\kappa_0) \tag{8}$$

The Cartesian coordinates (x, y) and heading angle θ of the path candidate points can be calculated from (6)–(8).

2.3. Path selection

Path selection aims to find the optimal path among all generated path candidates. A cost function f is used to rank paths numerically and pick the lowest-cost path. Considering comfortability and energy-efficiency, in our method acceleration is defined as a constant value for one step of path planning. In this case, the cost function is determined by path candidate r_i and the corresponding acceleration $a(r_i)$, which depends on the vehicle accelerator. The path selection process can be expressed by (8).

$$J=\min_{r_i,a} f(r_i,a(r_i)) \tag{9}$$

where r_i is a path candidate with index i, which is defined as $0, 1, 2, \ldots, n-1$ from left to right. n is the total number of path candidates.

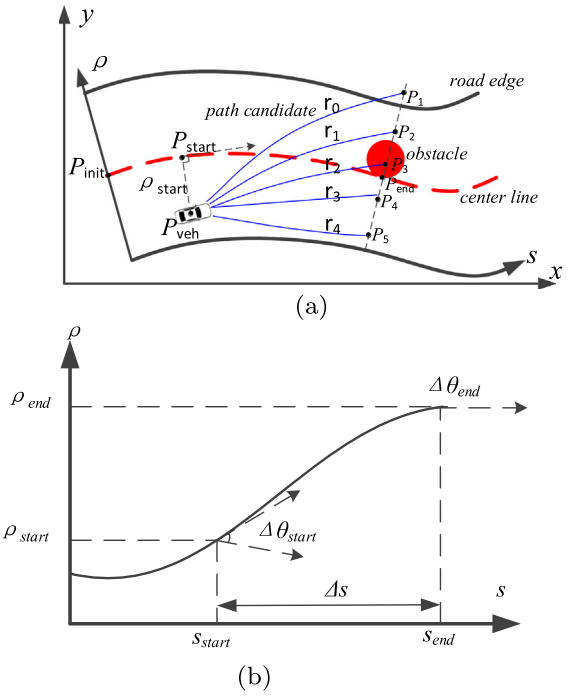

Fig. 3. Path generation in the $s-\rho$ coordinate system. (a) Path candidates in the $s-\rho$ coordinate system, (b) boundary conditions in the $s-\rho$ coordinate system.

The path selection for a skilled human driver comes from experience with respect to safety and comfortability. In this study, to avoid static and dynamic obstacles, the cost function for path selection is designed considering static safety, comfortability, and dynamic safety.

2.3.1. Cost function for static safety

Road edges, lane lines, and static obstacles are potential safety dangers. Static obstacles are blockages such as stones, construction signs and parked vehicles, on roads, whose information can be obtained from maps or detected by vehicle sensors. Because any object with a planar geometrical shape can be surrounded by its circumcircle, in this study a planar circle is used to describe static obstacles and planning vehicles which are referred to as "planners" in this paper.

To prevent collisions and maintain safe conditions, every path candidate should be checked for whether or not it crosses road edges, lane lines, or obstacles. Fig. 4(a) shows an example of path candidates on a single lane road. The collision check results are shown in Fig. 4(c). If a path crosses obstacles or road edges, the result of the collision check is 1. Otherwise, the result of the collision check is 0.

For multi-lane roads, the static safety function is more complicated. As shown in Fig. 4(b), a Chinese or American road with four lanes (including two traveling lanes and two opposing lanes) is taken as an example. The vehicle is traveling in the left lane of the traveling lanes. The collision check for path collisions with obstacles or road edges is defined as 1. According to traffic regulations, vehicles cannot cross opposing lanes, except for evading accidents. However, vehicles can usually cross traveling lanes for lane changing. In typical conditions, the vehicle continues traveling in its current lane. Therefore, we penalize vehicles that cross into the opposing lane (double lines) by assigning them a higher collision check (0.5), and penalize vehicles that cross traveling lanes (dashed line) by assigning them a lower collision check (0.2). If a path does not cross any obstacle, road edge, or lane line, the collision check is defined as 0. If a path crosses multiple types of lanes, road edges or obstacles, the collision check is defined as the highest value involved. The collision checks for a multi-lane road are shown in Fig. 4(d).

The results of the collision check are the most important information in terms of path selection, but may lead to dangerous accidents if other information is not used. Two examples are shown in Fig. 4(a) and (b). If the path selection criterion is the shortest path with a collision check of 0, the central path r_7 may be selected as the optimal path in Fig. 4(a), and r_4 may be selected as the optimal path in Fig. 4(b). In fact, as shown in r_7 in Fig. 4(a) and r_4 in Fig. 4(b), these paths pass very close to an obstacle. It is not safe for a driver to choose a path adjacent to an obstacle, because such a path may increase the probability of collision caused by control errors and occluded obstacles. Therefore, it is necessary to use a function to assess the risk of every path candidate. In this paper, a method using discrete Gaussian convolution combined with collision checks is used to determine the risk associated with every path candidate.

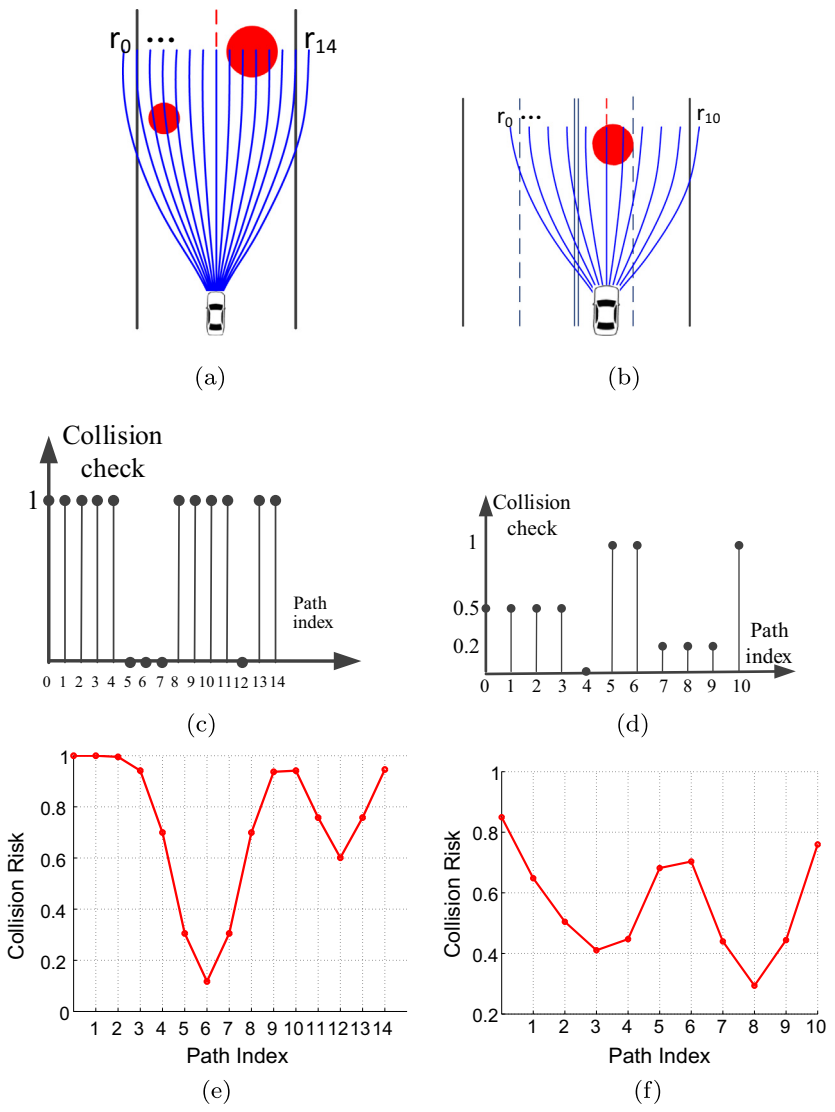

Fig. 4. Collision risk and static safety cost. (a) Path candidates on a single lane road, (b) path candidates on a multi-lane road - black line: road edge, blue line: path candidate, red circle: obstacle, dash red line: center line, dash black line: traveling lane line, double black lines: opposing lane line, (c) collision check for single lane road, (d) collision check for multi-lane road, (e) collision risk for single lane road, (f) collision risk for multi-lane road. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

$$f_s(r_i) = \sum_{k=-N}^{N} g_i[k] R[k+i] \tag{10}$$

where $f_s(r_i)$ is defined as the static safety cost function of path candidate r_i. $g_i[k]$ is the discrete inverted Gaussian function, which is defined as follows.

$$g_i[k] = \frac{1}{\sqrt{2\pi}\sigma} e^{\frac{-(k-i)^2}{2\sigma^2}} \tag{11}$$

where $2N+1$ is the length of convolution, which is equal to the number of path candidates in our method. σ is the standard deviation of collision risk, which is a parameter for determining the effective range of collision detections. The collision check is defined as 1 if i is out of range of the path index. Fig. 4(c) and (d) show the convolution results for single lane and multi-lane roads. Fig. 4(a) shows that r_7 is the shortest path, but the collision risk distribution in Fig. 4(c) indicates that the risk value of r_7 is not the lowest, because the path passes too close to an obstacle; instead r_6 is the safest path. Similarly, Fig. 4(d) shows that r_8 is the optimal path, which means that the vehicle must change lanes in order to avoid an obstacle.

2.3.2. Cost function for comfortability

The smoothness and continuity of a path are the two primary factors affecting driving comfortability. An unsmooth path may make occupants feel discomfort or even cause slippage of the wheels, which may decrease the stability of vehicle motion. Since the smoothness of a path is bound up with the curvature, the integration of the squared curvature along the length of a path is selected as the smoothness cost function, which is described as (12).

$$f_{sm}(r_i) = \int \kappa_i^2(s)ds \tag{12}$$

where $f_{sm}(r_i)$ is the smoothness cost function for r_i. $\kappa_i(s)$ is the curvature of r_i at a position with arc length s.

The cost of static safety is only considered for path selection in Fig. 5(a). According to the static safety cost, the shape of the path shows a sharp turn that converges to the road center. This is shown by the yellow path in Fig. 5(d). In Fig. 5(b), collision risk depends only on the smoothness cost. The selected path, shown in red in Fig. 6(d), is smoother than the yellow path, but it passes close to a road edge. The curve shown in Fig. 5(c) presents a distribution that considers the costs of both static safety and smoothness. The path selection results are shown in green in Fig. 5(d); the selected path is more reasonable than either the yellow or red paths.

The smoothness factor is collected at the current planning step, but it cannot prevent the generation of a path that is significantly different from a path in a previous step. If the difference between the path in the current step and the path in a previous step is too large, it will lead to an abrupt change. To avoid this case, path consistency must be considered. In our paper, the integral of the difference in heading angles between points on the previous path and points on the current path along the overlapped interval is used as the criterion for similarity, and is shown in Fig. 6. To eliminate the influence of vehicle speed with respect to consistency, the similarity criterion is divided by the difference in arc lengths between the start and end of the overlap. The consistency cost function of the ith path candidate $f_{co}(r_i)$ is shown by (13).

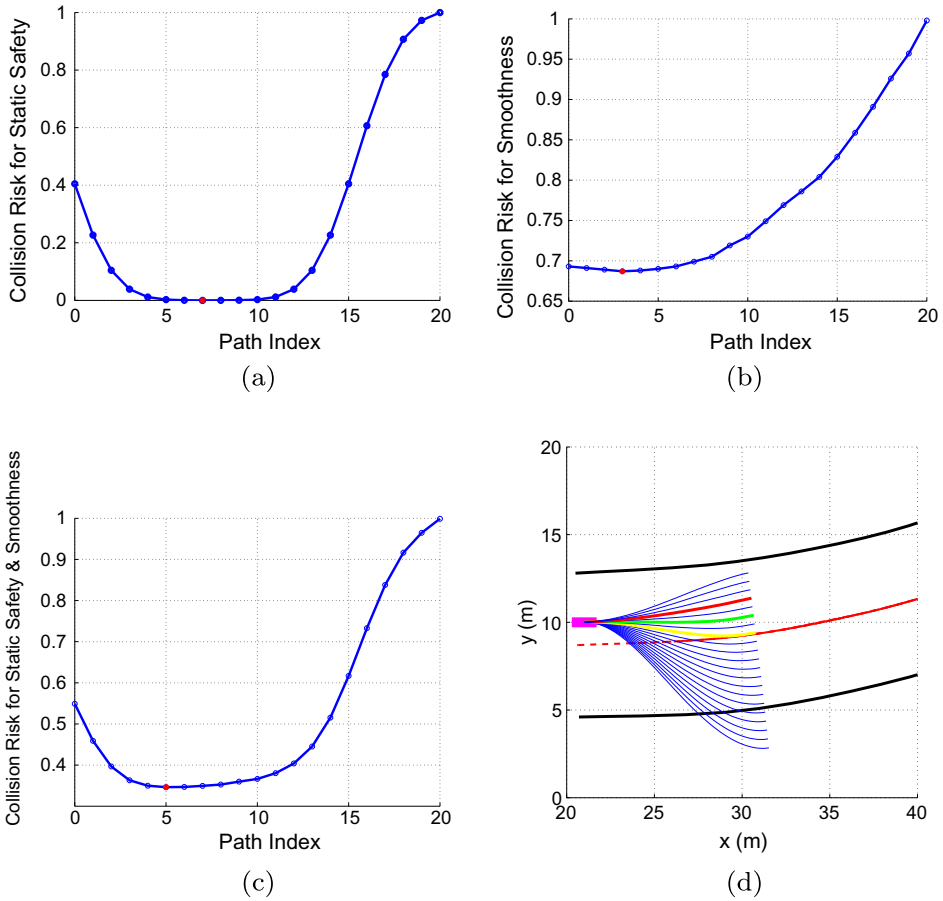

Fig. 5. Path selection considering static safety and smoothness. (a) Collision risk for static safety, (b) collision risk for smoothness, (c) collision risk for both safety and smoothness, (d) path selection - blue line: path candidates, yellow line: selection only by static safety cost, red line: selection only by smoothness cost, green line: selection combining both costs. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

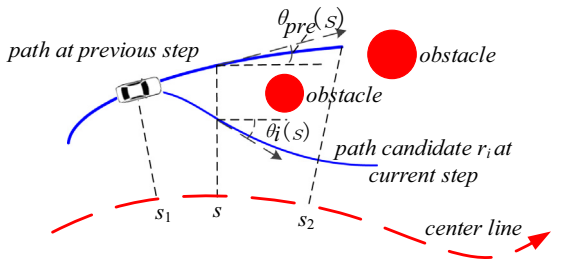

Fig. 6. Similarity between the previous path and the current path - $\theta_{pre}(s)$: heading angle at the position of arc length s on the previous selected path, $\theta_i(s)$: heading angle at the position of arc length s on the current path candidates, heading angle: the angle between the directions of x-axis and the tangent direction of the line at the current point.

$$f_{co}(r_i) = \frac{1}{s_2 - s_1} \int_{s_1}^{s_2} \Delta\theta_i(s) ds \tag{13}$$

where $\Delta\theta_i(s) = |\theta_{pre}(s) - \theta_i(s)|, \theta_{pre}(s), \theta_i(s) \in [0, \pi]$. s_1 and s_2 are the arc length of the center line at the starting point and end point of the overlap, respectively. $\theta_{pre}(s)$ and $\theta_i(s)$ are the heading angles for the arch length s positions on the previous and current path candidates, respectively.

The comfortability cost function f_c is defined as the weighted sum of the smoothness and consistency costs, and can be defined as follows.

$$f_c(r_i) = a f_{sm}(r_i) + b f_{co}(r_i) \tag{14}$$

where a and b are the weights used to balance the smoothness and consistency costs.

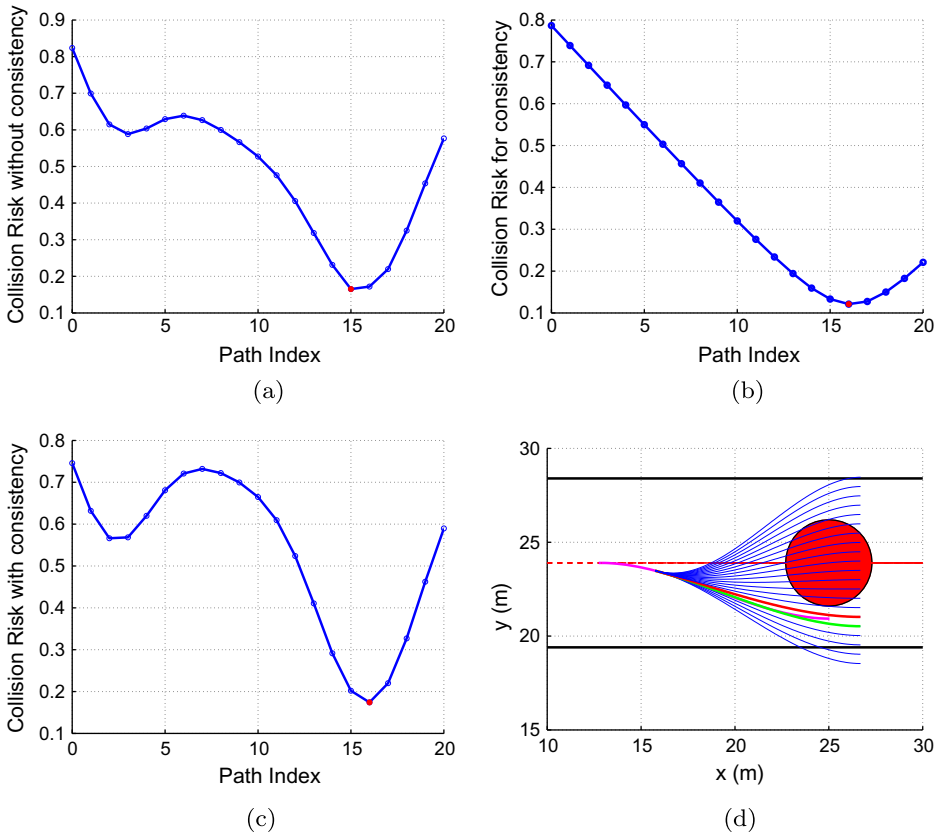

Fig. 7. Path selection with or without consistency. (a) Collision risk without consistency, (b) collision risk for consistency, (c) collision risk with consistency, (d) path selection - blue line: path candidates, dashed red line: center line, purple line: selected path at previous step, red line: selected path at current step without consistency, green line: selected path at current step with consistency. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

The purple path in Fig. 7(d) is the selected path from the previous step. The collision risk and selection result for the current step are shown by the curve in Fig. 7(a) and red path in Fig. 7(d), where the consistency cost is not taken into account. The selected path for the current step is close to the obstacle, and deviates from the selected path in the previous step. The collision risk in the current step only considers consistency costs, as shown in Fig. 7(b). The collision risk and selection results in the current step (where consistency costs are also considered) are shown by the curve in Fig. 7(c) and green path in Fig. 7(d). The green path is safer than the red one. In addition, the green path is similar to the path in the previous step, which means that the vehicle can follow it without excessive energy and control effort.

2.3.3. Cost function for dynamic safety

An optimal path can be achieved for a vehicle traveling on roads without moving obstacles by considering f_s and f_c. However, if moving obstacles (such as other vehicles) are present on roads, the selected path, which accounts only for static obstacles, may not be optimal. Because our algorithm is executed every 67 ms, moving obstacles can appear to be constant speed objects over this short duration. Therefore, a moving obstacle is defined as a plane circle with a constant speed and a trajectory along the road (meaning that the moving direction is the same as the tangent direction for the center line).

An example of a moving obstacle is shown in Fig. 8. The vehicle may select r_2 as the optimal path, according to the static safety and smoothness costs. When a moving obstacle is traveling along its trajectory it may arrive at position p_{o2} at the same time as the vehicle, which will lead to a collision. To prevent a collision, the planner must decelerate and replan the path. In this case, planner acceleration should be considered. In this paper, we design a cost function for dynamic safety considering the acceleration of the planner.

The paths that do not cross static obstacles or road edges are likely to be selected as optimal paths, which are referred to as "available paths" in this paper. All possible collision points with the moving obstacle for every available path (shown as the purple points in Fig. 9(b)), can be obtained by finding the intersections between the trajectory of the moving obstacle and the available paths. The collision range L_c shown in Fig. 9(b) usually equals the sum of the radii of the moving obstacle and planner. During time $t(r_i)$, the moving obstacle moved from P_{o1} to P_{o2}. If the planner intends to avoid a collision, two schemes can be executed: overtaking and following. Because driving at high speeds is not recommended, we design the method to avoid collision using the latter scheme. Thus, collision time $t(r_i)$ can be achieved by (15)

$$v_0 t(r_i) = \Delta s(r_i) - L_c \tag{15}$$

where v_0 is the current planner speed. $\Delta s(r_i)$ is the arc length between the starting point and collision point along r_i. $t(r_i)$ is the travel time along r_i from the starting point to the collision point, which is also the potential collision time.

If the distance the vehicle follows a moving obstacle is defined as L_f, the planner can travel distance $\Delta s(r_i) - L_f$ within $t(r_i)$ to avoid collision. According to (15) and the acceleration formula in physics, the appropriate acceleration $a(r_i)$ for path candidate r_i can be calculated by (16).

$$a(r_i) = 2(L_c - L_f)\frac{v_0^2}{(\Delta s(r_i) - L_c)^2} \tag{16}$$

where

$$L_f = \begin{cases} L_0 & L_0 >= \Delta s(r_i) \\ \Delta s(r_i) & \texttt{others} \end{cases} \tag{17}$$

The current vehicle speed can be calculated as the initial speed and the acceleration. In reality, the calculation of vehicle speed is restricted by some other factors, such as the curvature of the selected path, the width of the road and posted speed limits. Therefore, the vehicle speed limitation is shown by (18).

$$v_{limit}(r_i) = \min[v_\kappa(r_i), v_r(r_i), v_{sign}] \tag{18}$$

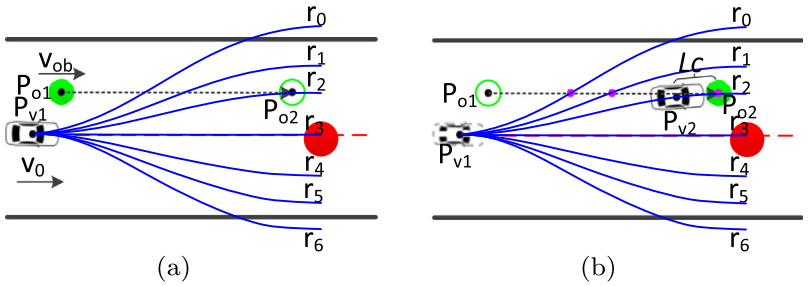

Fig. 8. Example of a moving obstacle on the road. Red circle: static obstacle. Green circle: moving obstacle. (a) Step t, (b) Step $t+1$. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

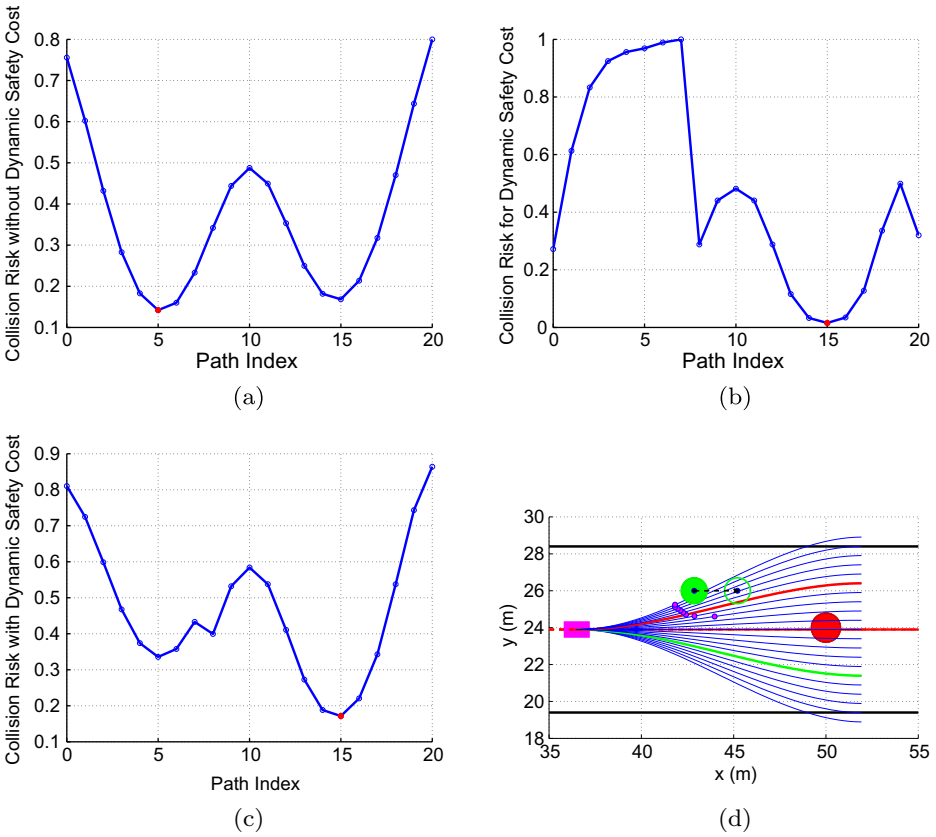

Fig. 9. Path selection with or without dynamic safety cost. (a) Collision risk, not considering dynamic safety cost, (b) collision risk considering only dynamic safety cost, (c) collision risk considering dynamic safety cost and other costs, (d) path selection - red filled circle: static obstacle, green filled circle: current position of moving obstacle, green circle: prospective position of moving obstacle, purple rectangle: current vehicle position, purple dot: possible collision position, blue line: path candidates, red dotted line: center line, red line: selected path, not considering dynamic safety cost, green line: selected path considering dynamic safety cost. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

where v_{sign} is the value shown on the speed limit sign. $v_\kappa(r_i)$ and $v_r(r_i)$ shown by (19) and (20) are speed limits considering curvature and risk, respectively.

$$v_\kappa(r_i) = \sqrt{\frac{|a_l|_{max}}{\max \kappa(r_i)}} \tag{19}$$

$$v_r(r_i) = (1 - k_{safe} f_s^2(r_i))\, v_{curve} \tag{20}$$

where $|a_l|_{max}$ is the limit of lateral acceleration, which is a design factor of the target vehicle speed. $\max \kappa(r_i)$ is the maximum curvature of r_i. k_{safe} is the safety gain for speed adjustment, which is a heuristic design parameter and is experimentally chosen according to the driving style. v_{curve} is the reference speed for the path, which is also an experimental parameter. In this paper, $|a_l|_{max}$, k_{safe}, and v_{curve} are set as 5000, 0.8, and 50, respectively. According to the speed limit shown by (18), the acceleration for r_i is limited by (21).

$$a(r_i) \leqslant \frac{v_{limit}^2(r_i) - v_0^2}{2(\Delta s(r_i) - L_f)} \tag{21}$$

To ensure that the accelerations occurring in successive planning steps are similar, a Gaussian filtering method is used to smooth the acceleration values. This is done using the values from past planning steps [44]. Because the planner accelerates uniformly during a planning step, the vehicle speed $v(r_i, t)$ can be easily calculated after the acceleration is obtained.

In autonomous driving, acceleration is an important parameter. Therefore, we design the cost function for dynamic safety considering acceleration and travel distance, as described by (22).

$$f_d(r_i, a(r_i)) = |a(r_i)|(\Delta s(r_i) - L_f) \tag{22}$$

Fig. 9(a) shows collision risk, where dynamic safety costs are not considered. The path selection result r_5 shown by the red path in Fig. 9(d) passes through the moving obstacle at the prospective position, meaning that a collision will occur. Fig. 9(b) shows collision risk only considering dynamic safety, where the risks of r_0 to r_7 are greater than the risks of other candidates, because the vehicle will decelerate and have greater acceleration. Fig. 9(c) shows collision risk where dynamic safety cost is considered combined with other factors. The path selection result r_{15} is shown by the green path in Fig. 9(d), where the selected path avoids both static and moving obstacles.

2.3.4. Total cost function for path selection

The total cost function shown by (23) is defined as the weighted sum of the three cost functions.

$$f(r_i, a(r_i)) = w_s f_s(r_i) + w_c f_c(r_i) + w_d f_d(r_i, a(r_i)) \tag{23}$$

where f is the total cost function. f_s,f_c, and f_d are the cost functions for static safety, comfortability, and dynamic safety, respectively. w_s, w_c, and w_d are the weights of the three cost functions. To set the three weights, f_s,f_c, and f_d are normalized to values between 0 and 1. The values of the three weights are determined by driving style. A higher w_s may cause a safer but slower path, while a higher w_c may cause a shorter path. A higher w_d may cause the vehicle to maintain its original speed and overtake the moving obstacle. In all simulations in this paper, w_s, w_c, and w_d are set to 0.5, 0.2, and 0.3, respectively. An optimal path with an appropriate acceleration and speed can be achieved by finding the minimum of the total cost function.

3. Simulation results

The simulations include two parts: the avoidance of static obstacles and the avoidance of moving obstacles. The map and static and moving obstacles are predesigned manually. The road edges are described by a cubic spline, and the obstacles are described by circles, as in the planner. For aesthetics, the planner is drawn as a rectangle in this paper.

In our simulations, the dashed blue lines and double yellow lines indicate traveling lane lines and opposing lane lines on a multi-lane road, respectively. The dashed red lines indicate the center lines of these roads. The solid red circles represent static obstacles on these roads. The purple rectangles and blue lines show the vehicles and path candidates, respectively. The solid green lines indicate the selected paths, and the passing trajectories of the planner are shown as purple lines. The moving obstacles and their trajectories are shown as green circles and dashed green lines, respectively. Figs. 10–16 (a)–(d) show planning moments, overall trajectories, and the acceleration and speed of the planner, respectively, where s, a, and v are the shift, acceleration, and speed. It should be noted that the accelerations shown in this paper are total values taking into account many factors, such as braking or throttling and resistance from the ground and air. Vehicle speed is limited to within 13.89 m/s (50 km/h) for most roads except the road in Fig. 13, where the vehicle speed limitation is 8.33 m/s (30 km/h).

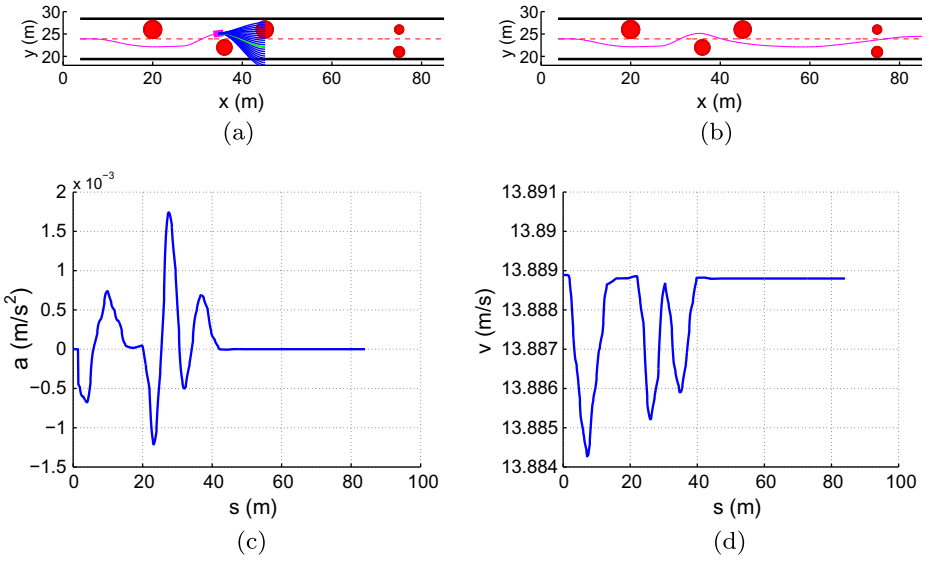

Fig. 10. Dynamic path planning results for a straight road with multiple sequent obstacles. (a) The vehicle avoids two sequent obstacles, (b) the trajectory of the vehicle passing along the road, (c) acceleration curve, (d) speed curve.

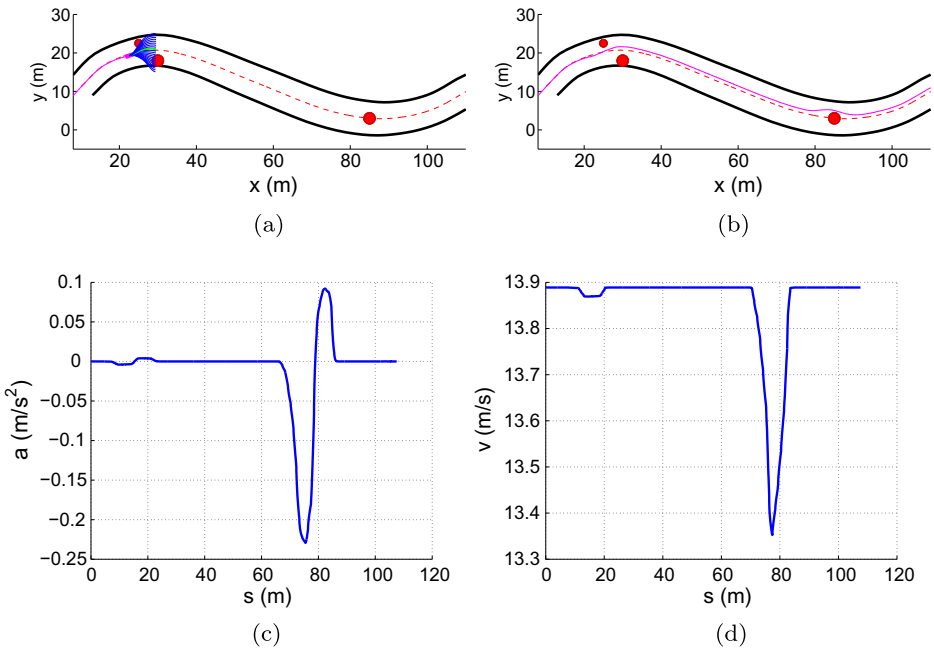

Fig. 11. Dynamic path planning results for an "S" shape road with sequent obstacles. (a) Moment at which the vehicle avoids the top obstacles (b) vehicle trajectory along the road, (c) acceleration curve, (d) speed curve.

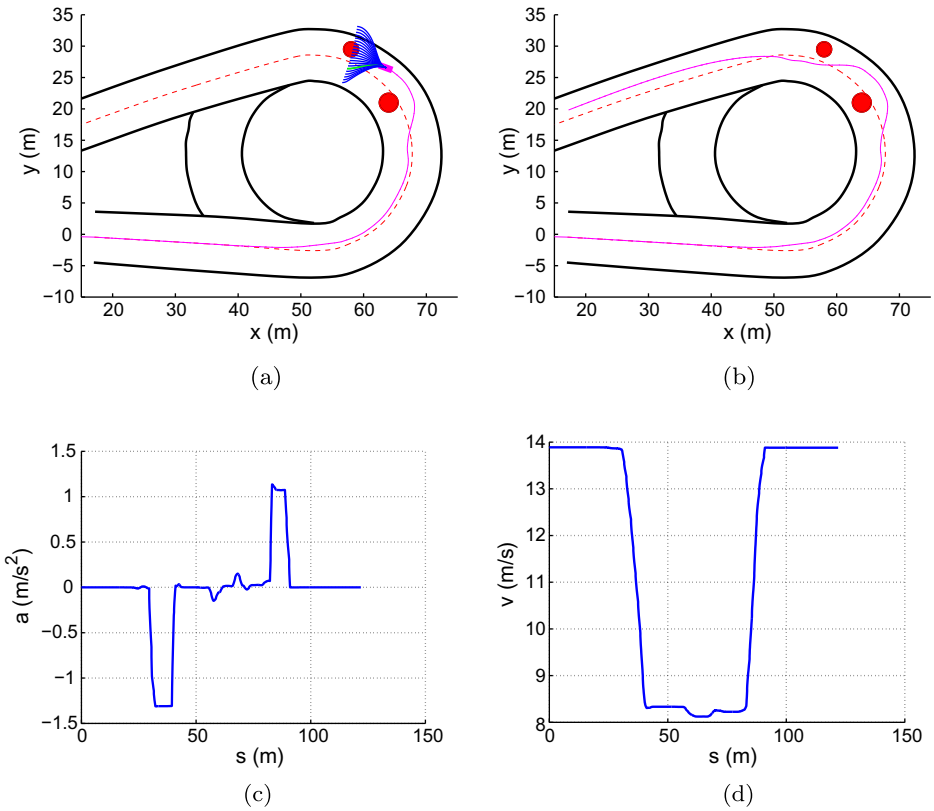

Fig. 12. Dynamic path planning results for a circular road with some obstacles. (a) Moment when vehicle avoids the second obstacle, (b) vehicle trajectory along the road, (c) acceleration curve, (d) speed curve.

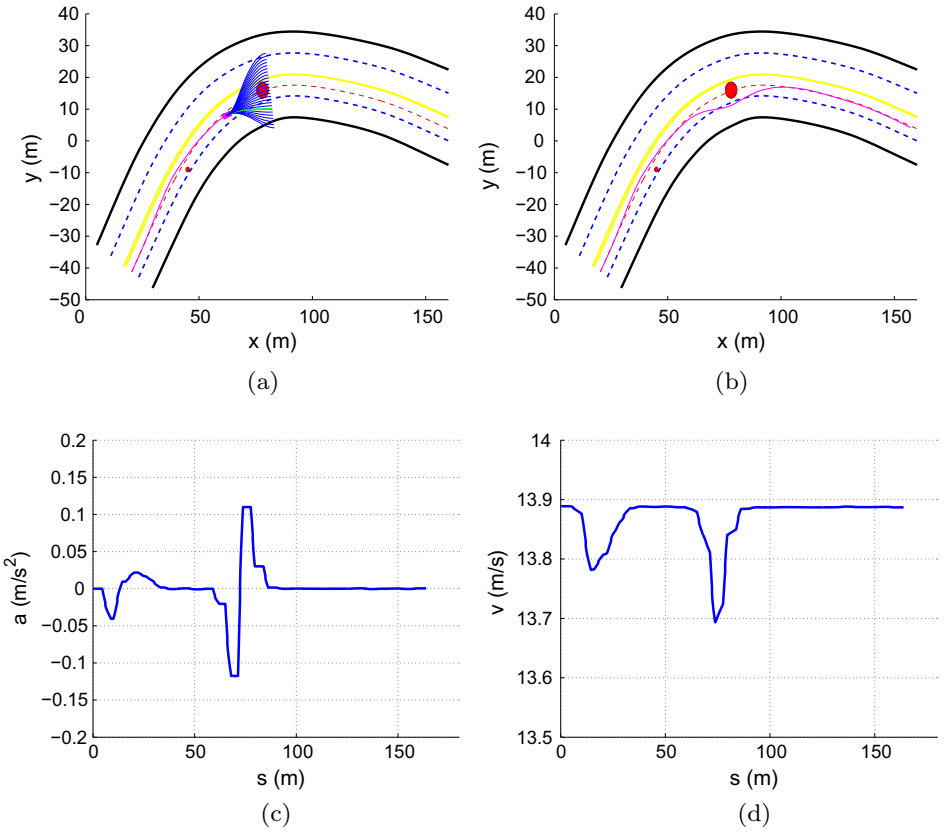

Fig. 13. Dynamic path planning results for multi-lane road with obstacles in the current lane. (a) Moment at which vehicle avoids larger obstacle, (b) trajectory of the vehicle passing through the road, (c) acceleration curve, (d) speed curve.

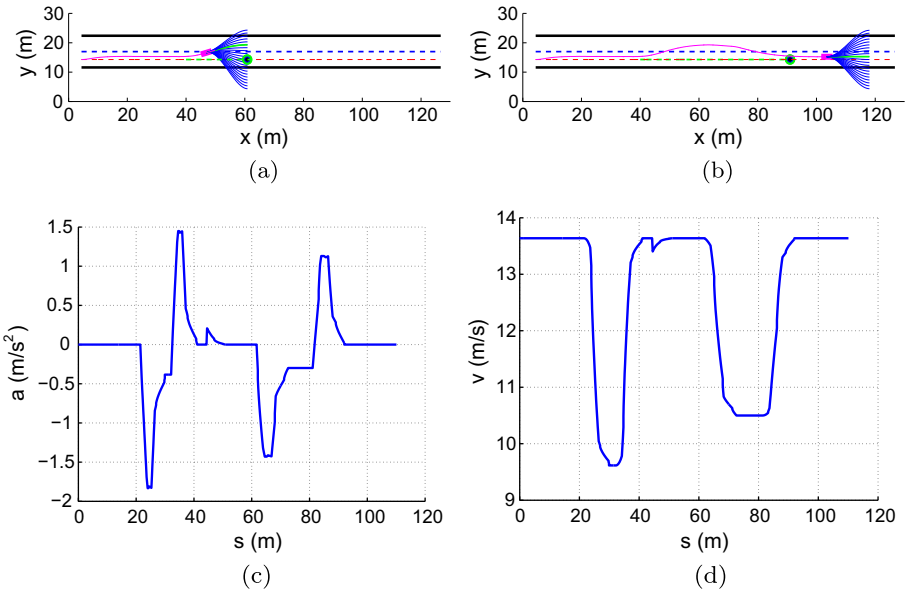

Fig. 14. Overtaking a moving obstacle. (a) Moment at beginning of overtaking the moving obstacle, (b) moment after overtaking moving obstacle, (c) acceleration curve, (d) speed curve.

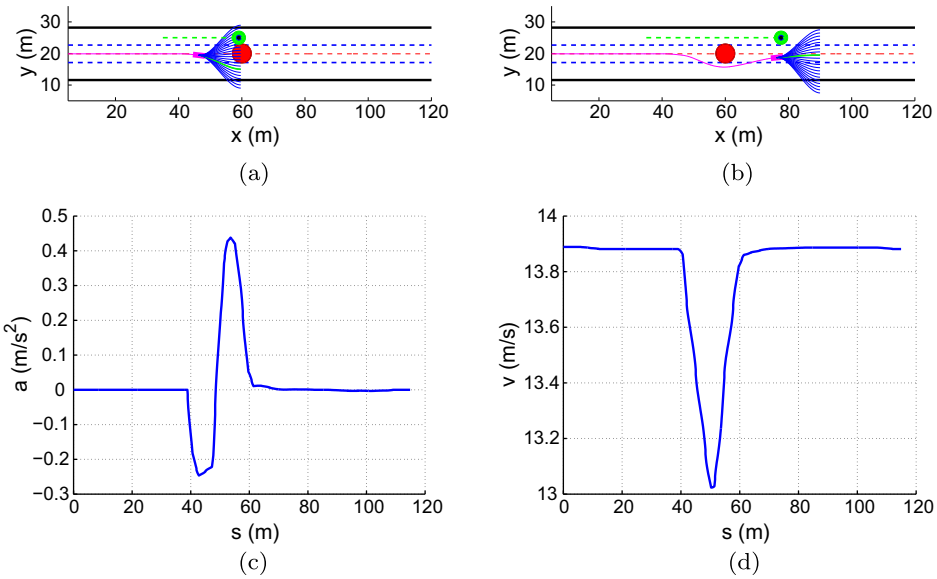

Fig. 15. Avoidance of moving and static obstacles by choosing another lane. (a) Moment detecting simultaneous moving and static obstacles, (b) moment after avoiding obstacles, (c) acceleration curve, (d) speed curve.

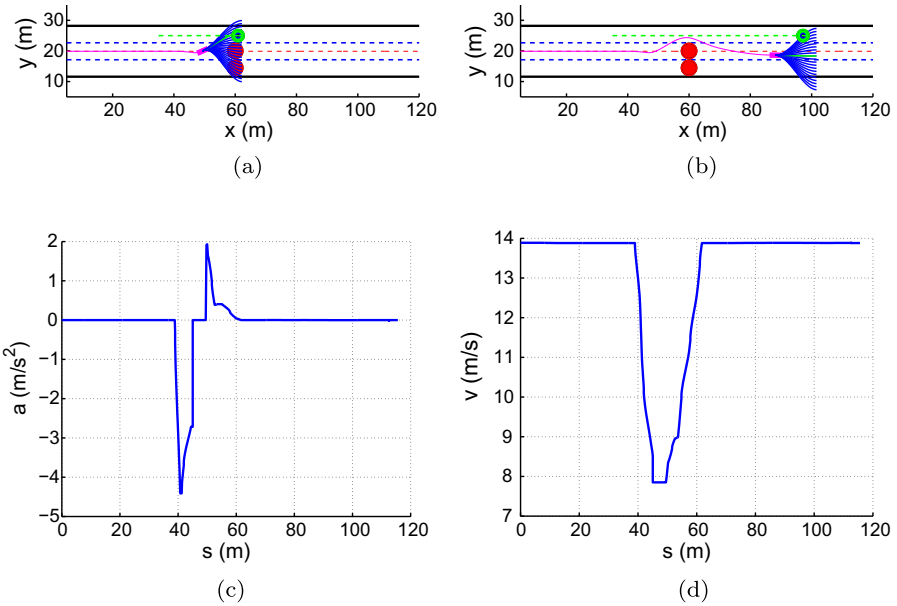

Fig. 16. Avoiding moving and static obstacles by following the moving obstacle. (a) Moment detecting the simultaneous moving and static obstacles, (b) moment after avoiding obstacles, (c) acceleration curve, (d) speed curve.

3.1. Avoidance of static obstacles

3.1.1. Straight road with multiple sequent obstacles

Fig. 10 shows a vehicle plan for traveling along a road with five sequent obstacles. Fig. 10(a) indicates a moment when the vehicle avoids two sequent obstacles. The vehicle turns left to avoid the obstacle on the right, travels several meters, and chooses the righthand path to avoid the obstacle on the left. As shown in Fig. 10(c) and (d), the speed decreases when selecting a path with a greater curvature and increases after avoiding an obstacle. Meanwhile, the vehicle returns to the center. Fig. 10(b) shows the overall trajectory, which is smooth and consecutive because of the smoothness and consistency costs used in the proposed method. This allows the vehicle to easily travel the road without abrupt steering.

3.1.2. "S" shape road

Fig. 11(a) shows the moment when the vehicle avoids the obstacles on the top of the "S" shape road, where two obstacles are present. The vehicle selects a path in the middle of the candidates to avoid the obstacles. At the same time, the speed and acceleration change slightly, as shown in Fig. 11(c) and (d). A safe and smooth curve in Fig. 11(b) shows the whole trajectory passing through the "S" shape road. When traveling on the slope of the "S" shape road, the vehicle does not choose the center line as the optimal path, but instead chooses the path on the lefthand side of the "S" shape road, because of smoothness costs.

3.1.3. Circular road

Fig. 12(a) shows the moment at which the vehicle avoids obstacles on a circular road. The two obstacles are close to one another, which indicates difficult maneuvering. Our method generates a safe and smooth trajectory. The speed decreases slightly as the vehicle avoids the obstacles for safety, as shown in Fig. 12(c) and (d). As shown in Fig. 12(b), the vehicle does not select the center line when entering the circle, but rather a path on the left side of the center line, because the selected path is smoother and sufficiently safe for travel. Meanwhile, the acceleration becomes negative until the speed decreases to the limited speed (8.33 m/s). When the vehicle travels out of the circle, it accelerates, and the speed returns to 13.89 m/s.

3.1.4. Multi-lane road

As shown in Fig. 13, a multi-lane road with two traveling lanes and two opposing lanes is designed for the simulation. When the vehicle detects a large obstacle, as shown in Fig. 13(a), the path candidates for the current lane are dangerous to follow. In terms of the collision rates for the different lane lines, the vehicle chooses to change traveling lanes to avoid the obstacle instead of crossing into the opposing lane, because the cost of crossing into the opposing lane is higher than the cost of changing traveling lanes. However, when avoiding a small obstacle, because of smoothness costs, the vehicle selects a path on the left of the obstacle without changing lanes. As shown in Fig. 13(c) and (d), the speed when avoiding a larger obstacle decreases more than that when avoiding a small obstacle, for safety reasons. The overall trajectory is shown in Fig. 13(b). The vehicle can continue traveling in the current lane and makes a valid decision when avoiding obstacles.

3.2. Avoidance of moving obstacles

3.2.1. Overtaking moving obstacles

As shown in Fig. 14, a road with two lanes is designed to test overtaking performance. The planner travels in the right lane at a speed of 13.89 m/s. An obstacle moves at 8.33 m/s in front of the planner. Fig. 14(a) shows the moment at which the vehicle starts to overtake the moving obstacle. If the planner is still traveling in the current lane, it must decelerate to avoid a collision. In contrast, changing speeds will increase the dynamic safety costs, and the costs of the paths in the current lane are higher than those in the other lane. Thus, the planner selects a smooth and safe path in the right lane (shown as the green path) and changes lanes. Then, the planner travels in the left lane until it overtakes the obstacle. After overtaking, the planner returns to the original lane, as shown in Fig. 14(b), because the paths in the original lane have lower costs. In the overtaking process, the planner slightly reduces its speed when changing lanes for safety, as shown in Fig. 14(c) and (d).

3.2.2. Avoidance of moving and static obstacles simultaneously

Challenging driving scenarios occur when static and moving obstacles emerge simultaneously. As shown in Fig. 15(a), the planner travels on the central lane of the road. The static obstacle lies on the current lane, and the moving obstacle moves on the left lane. The planner has two choices to avoid collision: choose the left lane, with significant deceleration caused by following the moving obstacle, or choose the right lane. Because the path candidates in the left lane have higher dynamic safety costs than those in the right lane, the planner selects a path in the right lane, shown as the green path. After collision avoidance, the planner returns to the central lane, as shown in Fig. 15(b). The speed and acceleration of the planner are shown in Fig. 15(c) and (d).

A more challenging design scenario, in which there is an additional static obstacle in the right lane, is shown in Fig. 16. Fig. 16(a) shows that the central lane and right lane are blocked by the obstacles. The left lane is the only lane that allows avoidance. However, there is a moving obstacle in the left lane at this moment. In this case, considering total costs, the planner selects a path in the left lane (the green path), and decelerates to a speed lower than that of the moving obstacle. The acceleration and speed are shown in Fig. 16(c) and (d).

All trajectories in Figs. 10–16 are smooth, continuous, and do not involve passing close to obstacles. This indicates that the proposed method can achieve a safe and comfortable path for avoiding both static and moving obstacles in various challenging scenarios. Moreover, the method can provide appropriate acceleration in order to manipulate vehicle throttling and braking. The appropriate speed is also given, to ensure that the vehicle travels at a safe speed.

3.3. Operation time

We tested our algorithm on a computer with an Intel Core i5-450M CPU (dual-core, 2.4 GHz) and 4GB RAM. Table 1 shows the operation time for every stage in a single planning step. The path generation time depends on the number of generated path candidates. In our simulations, 21 path candidates are generated. Because our method uses the path selection cost func-

Table 1
Operation time of the proposed approach.

Stage	Center line construction	Path generation	Path selection	Total
Time (ms)	4	48	5	57

tion instead of iteration, the total time is 57 ms for a planning step. The path planning algorithm can be executed more than 15 times per second, which meets the requirements of real-time systems.

4. Conclusions

This paper presents a dynamic path planning method for autonomous driving on various roads with avoidance of static and moving obstacles. The proposed method first constructs a center line using the global route. A set of path candidates are generated in the $s-\rho$ coordinate system. To select an optimal path, a novel cost function is designed that considers static safety, comfortability, and dynamic costs. This function prevents the autonomous vehicle from traveling unnecessarily close to static obstacles, allows the vehicle to avoid moving obstacles, and ensures comfortable driving.

Some challenging scenarios (including various roads with static and moving obstacles) were designed to test the developed method. The simulation results show that the proposed method can identify a safe and comfortable path (with appropriate acceleration and speed) for guiding the vehicle and avoiding obstacles. Although only one moving obstacle is considered in this paper, the proposed method could easily be extended in future work to deal with multiple moving obstacles. Meanwhile, the weights of the three cost functions in (23) are currently predefined in this paper, and their optimal values could be found in the future using machine learning methods.

References

[1] J. Ni, J. Hu, Dynamics control of autonomous vehicle at driving limits and experiment on an autonomous formula racing car, Mech. Syst. Signal Process. 90 (2017) 154–174.
[2] H. Zhang, J. Wang, Active steering actuator fault detection for an automatically-steered electric ground vehicle, IEEE Trans. Veh. Technol. 66 (5) (2017) 3685–3702.
[3] C. Hu, H. Jing, R. Wang, F. Yan, M. Chadli, Robust h output-feedback control for path following of autonomous ground vehicles, Mech. Syst. Signal Process. 70 (2015) 414–427.
[4] U. Ozguner, C. Stiller, K. Redmill, Systems for safety and autonomous behavior in cars: the Darpa grand challenge experience, Proc. – IEEE 95 (2) (2007) 397.
[5] C. Katrakazas, M. Quddus, W.-H. Chen, L. Deka, Real-time motion planning methods for autonomous on-road driving: state-of-the-art and future research directions, Transport. Res. Part C: Emerg. Technol. 60 (2015) 416–442.
[6] M. Buehler, K. Iagnemma, S. Singh, The DARPA Urban Challenge: Autonomous Vehicles in City Traffic, vol. 56, 2009.
[7] B. Siciliano, O. Khatib, F. Groen, The Darpa Urban Challenge: Autonomous Vehicles in City Traffic, Springer-Verlag, Berlin Heidelberg 56.
[8] D.A. Pomerleau, Alvinn: An Autonomous Land Vehicle in a Neural Network, Tech. Rep., DTIC Document, 1989.
[9] S. Le Vine, A. Zolfaghari, J. Polak, Autonomous cars: the tension between occupant experience and intersection capacity, Transport. Res. Part C: Emerg. Technol. 52 (2015) 1–14.
[10] H. Zhang, G. Zhang, J. Wang, H∞ observer design for LPV systems with uncertain measurements on scheduling variables: application to an electric ground vehicle, IEEE/ASME Trans. Mechatron. 21 (3) (2016) 1659–1670.
[11] S. Yang, Y. Cao, Z. Peng, G. Wen, K. Guo, Distributed formation control of nonholonomic autonomous vehicle via RBF neural network, Mech. Syst. Signal Process. 87 (2016) 81–95.
[12] M. Zhu, H. Chen, G. Xiong, A model predictive speed tracking control approach for autonomous ground vehicles, Mech. Syst. Signal Process. 87 (2017) 138–152.
[13] C. Chen, A. Seff, A. Kornhauser, J. Xiao, Deepdriving: learning affordance for direct perception in autonomous driving, in: IEEE International Conference on Computer Vision, 2015, pp. 2722–2730.
[14] C. Hu, H. Jing, R. Wang, F. Yan, M. Chadli, Robust h∞ h∞ mathcontainer loading mathjax output-feedback control for path following of autonomous ground vehicles, Mech. Syst. Signal Process. s70-71 (2016) 414–427.
[15] S. Zhang, W. Deng, Q. Zhao, H. Sun, B. Litkouhi, Dynamic trajectory planning for vehicle autonomous driving, in: IEEE International Conference on Systems, Man, and Cybernetics, 2013, pp. 4161–4166.
[16] X. Li, Z. Sun, D. Cao, D. Liu, H. He, Development of a new integrated local trajectory planning and tracking control framework for autonomous ground vehicles, Mech. Syst. Signal Process. 87 (2017) 118–137.
[17] O. Montiel, U. Orozco-Rosas, R. Seplveda, Path planning for mobile robots using bacterial potential field for avoiding static and dynamic obstacles, Exp. Syst. Appl. 42 (12) (2015) 5177–5191.
[18] J. Ji, A. Khajepour, W. Melek, Y. Huang, Path planning and tracking for vehicle collision avoidance based on model predictive control with multi-constraints, IEEE Trans. Veh. Technol. 66 (2) (2017) 952–964.
[19] W. Xu, J. Pan, J. Wei, J.M. Dolan, Motion planning under uncertainty for on-road autonomous driving, in: IEEE International Conference on Robotics and Automation, 2014, pp. 2507–2512.
[20] D. Ferguson, A. Stentz, The field D∗ algorithm for improved path planning and replanning in uniform and non-uniform cost environments, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, Tech. Rep. CMU-RI-TR-05-19.
[21] M. Pivtoraiko, R.A. Knepper, A. Kelly, Differentially constrained mobile robot motion planning in state lattices, J. Field Robot. 26 (3) (2009) 308–333.
[22] Z. Wang, S. Zlatanova, P.V. Oosterom, Path planning for first responders in the presence of moving obstacles with uncertain boundaries, IEEE Trans. Intell. Transport. Syst. 99 (2017) 1–11.
[23] L. Zuo, Q. Guo, X. Xu, H. Fu, A hierarchical path planning approach based on a∗ and least-squares policy iteration for mobile robots, Neurocomputing 170 (C) (2015) 257–266.
[24] A. Shum, K. Morris, A. Khajepour, Direction-dependent optimal path planning for autonomous vehicles, Robot. Autonom. Syst. 70 (2015) 202–214.
[25] J. Barraquand, B. Langlois, J.-C. Latombe, Numerical potential field techniques for robot path planning, IEEE Trans. Syst., Man Cybernet. 22 (2) (1992) 224–241.
[26] H.M. Choset, Principles of Robot Motion: Theory, Algorithms, and Implementation, MIT press, 2005.

[27] S. Waydo, R.M. Murray, Vehicle motion planning using stream functions, IEEE International Conference on Robotics and Automation, Vol. 2, 2003, pp. 2484–2491.
[28] R. Daily, D.M. Bevly, Harmonic potential field path planning for high speed vehicles, in: American Control Conference, 2008, pp. 4609–4614.
[29] S.M. LaValle, Rapidly-Exploring Random Trees A New Tool for Path Planning.
[30] K.J. Obermeyer, P. Oberlin, S. Darbha, Sampling-based path planning for a visual reconnaissance unmanned air vehicle, J. Guid. Control Dynam. 35 (2) (2015) 619–631.
[31] R. Cui, Y. Li, W. Yan, Mutual information-based multi-AUV path planning for scalar field sampling using multidimensional RRT∗, IEEE Trans. Syst. Man Cybernet. Syst. 46 (7) (2016) 993–1004.
[32] D. Devaurs, T. Simon, J. Corts, Optimal path planning in complex cost spaces with sampling-based algorithms, IEEE Trans. Autom. Sci. Eng. 13 (2) (2016) 415–424.
[33] Y. Kuwata, S. Karaman, J. Teo, E. Frazzoli, J.P. How, G. Fiore, Real-time motion planning with applications to autonomous urban driving, IEEE Trans. Control Syst. Technol. 17 (5) (2009) 1105–1118.
[34] S. Karaman, E. Frazzoli, Sampling-based algorithms for optimal motion planning, Int. J. Robot. Res. 30 (7) (2011) 846–894.
[35] M. Werling, J. Ziegler, S. Kammel, S. Thrun, Optimal trajectory generation for dynamic street scenarios in a Frenet frame, in: IEEE International Conference on Robotics and Automation, 2010, pp. 987–993.
[36] 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, J. Field Robot. 25 (9) (2008) 569–597.
[37] K. Chu, M. Lee, M. Sunwoo, Local path planning for off-road autonomous driving with avoidance of static obstacles, IEEE Trans. Intell. Transp. Syst. 13 (4) (2012) 1599–1616.
[38] M.D. Phung, H.Q. Cong, T.H. Dinh, Q. Ha, Enhanced discrete particle swarm optimization path planning for UAV vision-based surface inspection, Autom. Constr. 81 (2017) 25–33.
[39] H. Wang, J. Kearney, K. Atkinson, Arc-length parameterized spline curves for real-time simulation, in: 5th International Conference on Curves and Surfaces, 2002.
[40] I.N. Bronshtein, K.A. Semendyayev, G. Musiol, Handbook of Mathematics, Springer Science & Business Media, 2013.
[41] H. Wang, J. Kearney, K. Atkinson, Robust and efficient computation of the closest point on a spline curve, in: Proceedings of the 5th International Conference on Curves and Surfaces, 2002, pp. 397–406.
[42] B. Siciliano, L. Sciavicco, L. Villani, G. Oriolo, Robotics: Modelling, Planning and Control, Springer Science & Business Media, 2010.
[43] T.D. Barfoot, C.M. Clark, Motion planning for formations of mobile robots, Robot. Autonom. Syst. 46 (2) (2004) 65–78.
[44] J.-M. Geusebroek, A.W. Smeulders, J. Van de Weijer, Fast anisotropic gauss filtering, IEEE Trans. Image Process. 12 (8) (2003) 938–943.

Xuemin Hu received the B.Sc. degree in Biomedical Engineering from Huazhong University of Science and Technology and the Ph.D. degree in Signal and Information Processing from Wuhan University in 2007 and in 2012, respectively. From July 2012 to now, he is an assistant professor of School of Computer Science and Information Engineering, Hubei University, Wuhan, China. He was a visiting scholar of the University of Rhode Island, Kingston, RI, US from November 2015 to May 2016. His areas of interest include path planning, autonomous driving and computer vision.

Long Chen received the B.Sc. degree in communication engineering and the Ph.D. degree in signal and information processing from Wuhan University, Wuhan, China, in 2007 and in 2013, respectively. From 2008 to 2013, he was in charge of environmental perception system for autonomous vehicle SmartV-II with the Intelligent Vehicle Group, Wuhan University. He is currently an Associate Professor with the School of Data and Computer Science, Sun Yat-sen University, Zhuhai, China. His areas of interest include computer vision and unmanned autonomous vehicles.

Bo Tang received the B.S. degree from the Department of Information Physics, Central South University, Changsha, China, in 2007, and the M.S. degree from the Institute of Electronics, Chinese Academy of Science, Beijing, China, in 2010. In 2016, he received his Ph.D. degree in Electrical Engineering from University of Rhode Island. Currently, he is currently an Assistant Professor in the Department of Electrical and Computer Engineering, Mississippi State University, MS, USA. His research interests include statistical machine learning, computational intelligence, computer vision, and robotics.

Dongpu Cao received the Ph.D. degree from Concordia University, Canada, in 2008. He is currently a Senior Lecturer at Advanced Vehicle Engineering Center, Cranfield University, UK. His research focuses on vehicle dynamics, control and intelligence, where he has contributed more than 90 publications and 1 US patent. He received the ASME AVTT?2010 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, IEEE TRANSACTIONS ON INDUSTRIAL ELECTRONICS and ASME JOURNAL OF DYNAMIC SYSTEMS, MEASUREMENT, AND CONTROL. He has been a Guest Editor for IEEE/ASME TRANSACTIONS ON MECHATRONICS, VEHICLE SYSTEM DYNAMICS, and IEEE TRANSACTIONS ON HUMAN-MACHINE SYSTEMS. He serves on the SAE International Vehicle Dynamics Standards Committee and a few ASME, SAE, IEEE technical committees.

Haibo He received the B.S. and M.S. degrees in electrical engineering from Huazhong University of Science and Technology, China, in 1999 and 2002, respectively, and the Ph.D. degree in electrical engineering from Ohio University in 2006. He is currently the Robert Haas Endowed Chair Professor in Electrical Engineering at the University of Rhode Island. From 2006 to 2009, he was an Assistant Professor at the Department of Electrical and Computer Engineering, Stevens Institute of Technology. His current research interests include adaptive dynamic programming, computational intelligence, self-adaptive systems, and various applications.