2015 IEEE Intelligent Vehicles Symposium (IV)
June 28 - July 1, 2015. COEX, Seoul, Korea

A Constrained VFH Algorithm for Motion Planning of Autonomous Vehicles

Panrang Qu, Jianru Xue, Liang Ma, Chao Ma

Lab of Visual Cognitive Computing and Intelligent Vehicle
Xi'an Jiaotong University, Xi'an 710049, China
Email: jrxue@mail.xjtu.edu.cn

Abstract— **The Vector Field Histogram (VFH) is a classical motion planning algorithm which is widely used to handle the trajectory planning problem of mobile robots. However, the traditional VFH algorithm is rarely applied to autonomous vehicles due to the vehicle's well-known non-holonomic constraints, especially in urban environments. To address this problem, we propose a constrained VFH algorithm which takes both kinematic and dynamic constraints of the vehicle into consideration. The goal is achieved via two contributions that concern both kinematic and dynamic constraints of the vehicle. First, we develop a new active region for VFH to guarantee that all states within the region are reachable for the vehicle. Second, we improve the cost function to guide the search to favor feasible motion direction for the vehicle. The proposed algorithm is extensively tested in various simulated urban environments, and experimental results validate its efficiency.**

I. INTRODUCTION

Motion planning, producing a feasible motion state sequence which is used to guide the robot to the target location, is a key and essential technology for mobile robot [1], [2]. This fundamental research topic has received widespread attentions over three decades, and many motion planning algorithms, such as A*, D* [3] and Rapidly-exploring Random Tree (RRT) [4], were proposed. However, It is quite difficult to apply mentioned algorithms while taking into considerations of the ground vehicle's non-holonomic constraints.

In contrast, parametric curves, due to its simplicity and satisfying with the dynamic constraint, were favored by researchers for a long time. To name a few, the two-dimensional cubic Bezier curve is proposed by the Cornell team to represent trajectories [5], and Long Han et al. further present an improved trajectory planning algorithm dependent on Bezier curves [6]. With the rapid development of the autonomous vehicle, the capability of obstacle avoidance becomes critical, and this attracts increasing attention of researchers to classical motion planning algorithms. For example, the MIT's autonomous vehicle uses RRT to handle this problem [7]. Recently, L. Ma et al [8]. propose an efficient RRT variant for autonomous on-road driving. However, due to sampling randomly, RRT always generates different trajectoies even in the same scenario.

VFH was proposed by Johann Borenstein for motion planning of the mobile robot CARMEL [9]. Its main advantage is the robust motion planning capacity even in a cluttered obstacle scenario. It is also computationally efficient, and insensitive to mis-interpretation of obstacle information. Many enhanced variants are emerging. Iwan Ulrich et al. present an enhanced method called VFH+ by designing a cost function [10], and further propose VFH* to overcome major drawbacks of purely local motion planning algorithm [11], Dong Jie et al. come up with IVFH* for dynamic obstacles avoidance [12].

However, applying VFH algorithm to autonomous vehicles directly is always problematic due to the complex dynamics. In this paper, we propose a constrained VFH algorithm to solve the problem of motion planning for autonomous vehicles in urban environments. Three major problems are addressed.

- VFH is a kind of real-time motion planning algorithm which may results in vibrations of vehicle's heading angle.
- Unreachable states may be searched if applying the original VFH method to autonomous vehicles directly.
- The sparse location sequence generated by the original VFH algorithm cannot guide the autonomous vehicle toward the goal state smoothly.

We improve the original VFH algorithm by introducing non-holonomic constraints of the vehicle. The highlights of our improvements are demonstrated in two aspects: a reachable active region and a modified cost function. Compared with the original VFH algorithm, our algorithm can generate smoother trajectories which meet with the requirement of autonomous vehicles.

This paper is organized as following. In section II, a brief definition of motion planning is presented. In section III, a more detailed description for the constrained VFH algorithm is given. In section IV, some experimental results are shown. We end the paper with a conclusion in section V.

II. PROBLEM FORMULATION

Before detailed descriptions of our algorithm, we present the definition of motion planning in this section. It can be formally defined by a dynamic function [13]:

$$\dot{x}(t) = f(x(t), u(t)), x(0) = X_{initial}, x(T) = X_{goal} \quad (1)$$

where $x(t) \in X$, $X \subset \mathbb{R}^n$ denotes the state space, while $u(t) \in U$, $U \subset \mathbb{R}^m$ denotes the control input space. $X_{initial}$

This work was supported by the National Natural Science Foundation of China under Grants No. 91320301 and No. 61273252.

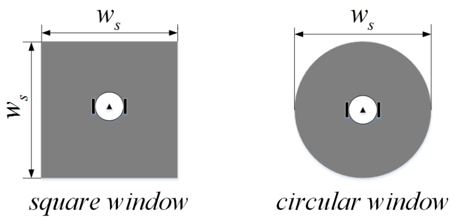


Fig. 1. Original active regions.

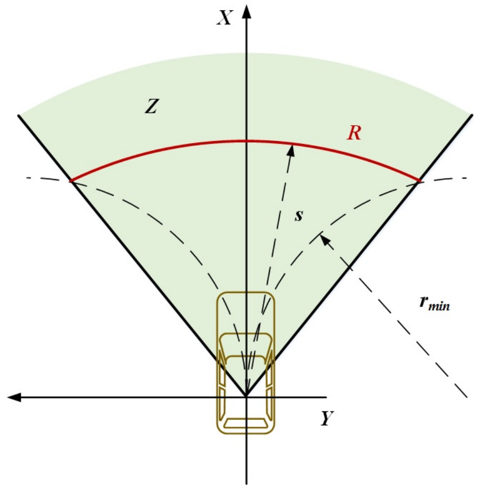


Fig. 2. Active region Z for autonomous vehicle.

and X_{goal} denote the initial state and the goal state of the autonomous vehicle, respectively.

Let X_{obs} and $X_{free} = X \setminus X_{obs}$ denote the obstacle occupied region and the obstacle free region, respectively. The goal of motion planning is to find an ordered pair sequence:

$$(x, u) = \{< x(t), u(t) > | x(t) \in X_{free}, u(t) \in U, \forall t \in [0, T]\} \tag{2}$$

where x is a feasible trajectory, and u is the corresponding control sequence, and $x(t)$ is a reachable state for the vehicle at time t, while $u(t)$ is the control input at time t.

III. ALGORITHM DESCRIPTION

In this section, the improved VFH for autonomous vehicles is presented. We first give a brief review of VFH, and then describe our algorithm in details. VFH can be implemented with a three-step procedure: 1) Polar histogram is generated based on obstacle vectors. 2) Find all candidate directions. 3) Compute the new motion direction.

A. Active Region

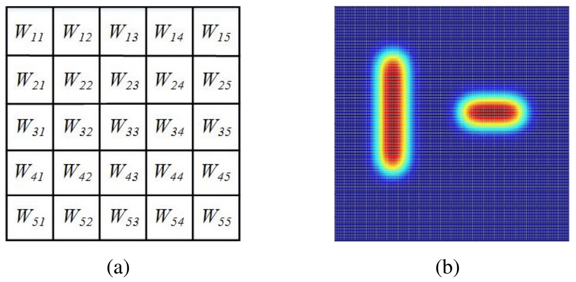
Active region, as shown in Fig. 1, is defined by a moving square window of $w_s \times w_s$ in [9], and a moving circular window of diameter w_s in [10]. It jointly represents the perceptual range and the feasible region of a mobile robot.

However, some states exist in the active regions are inaccessible for autonomous vehicles. The state of a vehicle is restrained from its maneuverability. Taking the non-holonomic constraints into account, we design a new active region for autonomous vehicles. In the new active region as shown in Fig. 2, r_{min} denotes the minimum steering radius of the vehicle. Usually, when the vehicle moves at low speed, r_{min} is invariable. The searching step is denoted as s, which is invariable in our method. We represent the active region in a polar coordinate system, which is then be defined as:

$$\begin{aligned} &Z = P(\rho, \varphi) \\ &\rho \in [0, \rho_{max}] \\ &\varphi \in [-\arcsin\frac{s}{2r_{min}}, \arcsin\frac{s}{2r_{min}}] \end{aligned} \tag{3}$$

where ρ denotes the distance, and ρ_{max} denotes the maximum active distance, while φ denotes the angle. The new position searched based on the proposed algorithm must be on the arc R, and it can be presented by an arc $P(s, \varphi)$ in the polar coordinate system.

Since taking into consideration non-holonomic constraints of the vehicle, all trajectory points on the arc R are reachable for the vehicle. Another important advantage of the new active region is the reduction of searching space, because this region is smaller than previous ones.



W_{11} W_{12} W_{13} W_{14} W_{15}
W_{21} W_{22} W_{23} W_{24} W_{25}
W_{31} W_{32} W_{33} W_{34} W_{35}
W_{41} W_{42} W_{43} W_{44} W_{45}
W_{51} W_{52} W_{53} W_{54} W_{55}
(a)
(b)

Fig. 3. Obstacle representation. (a) Filter mask. (b) Occupancy grids map.

B. Candidate Direction

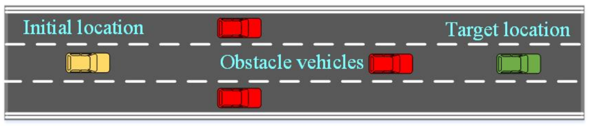
This section describes how to get the candidate directions of motion from the occupancy grid map and the proposed active region.

1) Obstacle grid:

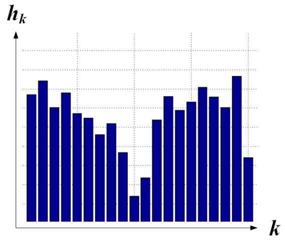
Occupancy grid map is a classical representation of obstacle information [14]. A binary value c corresponding to each grid indicates the confidence of obstacle: zero presents that the grid is obstacle-free while one denotes that the grid is occupied.

To model uncertainty of the location of obstacles, a discrete low-pass filter mask shown in Fig. 3(a) is used to smooth obstacle confidence. It means that the obstacle confidence of each grid is updated by $\sum c_{i,j} W_{i,j}$, where $\sum W_{i,j} = 1$. The new occupancy grid map is shown in Fig. 3(b), where warm-toned regions denote the more possibility of occupancy.

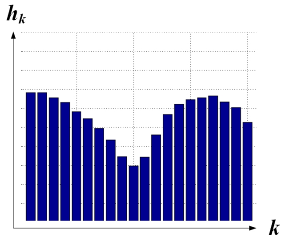
This filtered result is a probabilistic representation of obstacle information. Another advantage is that appropriate expansion of the obstacle occupied region can apparently enhance the safety of motion planning, since the safe distance between the autonomous vehicle and an actual obstacle is being expanded.



(a)



(b)



(c)

Fig. 4. Polar histogram. (a) The scenario. (b) Rough polar histogram. (c) Smooth polar histogram

2) Polar histogram:

The polar histogram is described in detail in [9]. Each gird occupied by obstacles inside the active region Z is represented as a vector, whose direction $\beta_{i,j}$ is defined by the direction from the vehicle center point (VCP) to the grid:

$$\beta_{i,j} = \arctan\left(\frac{y_j - y_0}{x_i - x_0}\right) \tag{4}$$

where (x_0, y_0) denotes the current location of the VCP, and (x_i, y_j) presents the location of the obstacle gird $C_{i,j}$. The magnitude $m_{i,j}$ of the obstacle grid $C_{i,j}$ is defined by:

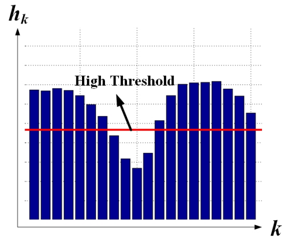
$$m_{i,j} = c_{i,j}^2 \frac{a}{d_{i,j}^2} \tag{5}$$

where $c_{i,j}$ denotes the occupied confidence of the obstacle grid $C_{i,j}$, and $d_{i,j}$ denotes the distance form obstacle grid $C_{i,j}$ to the VCP. For $m_{i,j}$ to be the square of $c_{i,j}$ for the nearest obstacle grid to the VCP, the parameters a is chosen according to:

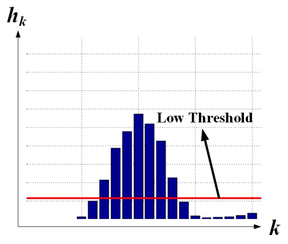
$$\frac{a}{d_{min}^2} = 1 \tag{6}$$

where d_{min} denotes distance from the nearest obstacle grid to the VCP. The magnitude $m_{i,j}$ is proportional to the square of $c_{i,j}$, which express that high $c_{i,j}$ values represent real obstacles while low $c_{i,j}$ values denote mis-interpretation [9]. Meanwhile, it is inversely proportional to the square of distance $d_{i,j}$, thus closer obstacle grids generate larger vector magnitudes while farther obstacle grids produce smaller ones.

Fig. 4 illustrates the procedure of the polar histogram and the filtered result. Fig. 4(a) shows a traffic scenario. The rough polar histogram H_p shown in Fig. 4(b) is built dependent on obstacle grids. We select a suitable angular resolution δ so that the value $K = \frac{2\arcsin\frac{s}{2r_{min}}}{\delta}$ is an integer. The active region Z is divided into K sectors, each sector k relates to a discrete direction $-\arcsin\frac{s}{2r_{min}} + k\delta$, and an obstacle density $h_k = \sum m_{i,j}$ which is defined by the magnitude sum of all obstacle vectors within it. Referring to the original VFH algorithm, a low-pass filter is used to smooth the rough polar histogram H_p caused by the discrete nature of the histogram grid, and a smooth polar histogram H shown in Fig. 4(c) is produced.



(a)



(b)

Fig. 5. Two-hierarchical threshold. (a) High level threshold. (b) Low level threshold.

3) Threshold:

A threshold is used to determine candidate directions based on the polar histogram H. However, selecting an appropriate threshold is a critical problem for the autonomous vehicle [9], since performances are largely dependent on the threshold. If the threshold is too large, the autonomous vehicle may neglect actual obstacles and a collision may occur, while selecting a smaller one may make some feasible candidate directions excluded, and the vehicle may lack of capability of passing through narrow passages.

In our algorithm, a two-level threshold based on multi-sensors information is adopted. In brief, if information, which consists of lanes, GPS and obstacles, calls for passing through a narrow passage, the threshold is temporarily raised as shown in Fig. 5(a); while in normal condition the threshold keep a low level as shown in Fig. 5(b).

The above processes generate a set of candidate directions of motion. In Fig. 5, the discrete direction of a sector whose obstacle density smaller than the threshold is a candidate direction.

C. Cost Function

The polar histogram with threshold shows two situations: obstacle blocked and obstacle free. However, it is unknown that which direction is the best candidate for the future motion.

The original VFH algorithm determines the new motion direction based on the size of the valley whose direction is closest to k_t, which is defined by the direction from the current location to the target location [9]. While the VFH+ method ameliorates this approach by designing a cost function to get an optimal direction among all candidate directions.The cost function is defined by $g(c_k)$ [10]:

$$g(c_k) = \mu_1 \cdot \Delta(c_k, k_t) + \mu_2 \cdot \Delta(c_k, \frac{\theta_n}{\alpha}) + \mu_3 \cdot \Delta(c_k, c_p) \tag{7}$$

where Δ is the calculation of absolute difference between two parameters. The first term of the equation Eq. 7 denotes the cost resulted from the difference between the k-*th* candidate direction c_k and the direction k_t. The second term presents the cost associated with the difference between the k-th candidate direction c_k and the robot's heading angle $\frac{\theta_n}{\delta}$. And the third term presents the cost resulted from the difference between the k-th candidate direction c_k and the previously selected direction c_p.

For an autonomous vehicle, the cost function $g(c_k)$ is unrealizable and inadaptable due to two reasons. First, because of the complex dynamics, the heading angle of the searched location is not equal to the selected candidate direction and is generally unknown. Second, both the vehicle's width and the smoothness of the selected direction sequence are not taken into account.

Motivated by the cost function $g(c_k)$ created by VFH+, we propose a new cost function $g(c_n)$ for the autonomous vehicle as follows:

$$g(c_n) = \begin{cases} inf, & \exists c \in [c_n^{(-2)}, c_n^{(-1)}, c_n^{(1)}, c_n^{(2)}], c \notin Set \\ \frac{1}{G(c_n)} g_0(c_n), & \forall c \in [c_n^{(-2)}, c_n^{(-1)}, c_n^{(1)}, c_n^{(2)}], c \in Set \end{cases} \tag{8}$$

where:

$$G(c_n) = \frac{1}{\sqrt{2\pi}\sigma} exp^{-\frac{(c_n - \hat{c}_{n-1})^2}{2\sigma^2}} \tag{9}$$

$$g_0(c_n) = \mu_1 \cdot \Delta(c_n, k_t)^2 + \mu_2 \cdot \Delta(\frac{\tilde{\theta}_{n+1}}{\alpha}, \frac{\theta_T}{\alpha}) \tag{10}$$

The discrete direction c_n denotes a candidate direction in the n-th planning step, Set denotes the candidate direction collection, discrete directions $c_n^{(-2)}$, $c_n^{(-1)}$, $c_n^{(1)}$ and $c_n^{(2)}$ are neighboring directions of the sector whose direction is c_n. And k_t denotes the direction form current location to target location, while $\hat{c}_{n-1}$ is defined by the previously selected direction. $\tilde{\theta}_{n+1}$ denotes the estimation of heading angle for the next searched location while θ_T presents the heading angle of the target pose. Parameters μ_1, μ_2 and σ are adjustable for better performances.

The improved cost function has three advantages: the consideration of the vehicle width, the achievement to the target pose, and the smoothness of the searched location sequence.

1) Vehicle width:

The cost function $g(c_n)$ takes vehicle width into account by examining the candidate direction c_n whether its neighbor sectors discrete directions are also in the candidate direction collection Set. In our implementation, the angular resolution $\delta = 5°$ and the searching step length $s = 5m$. So the arc $l = \delta s$ of a sector is equal to $0.436m$, while our vehicle's width $w = 2m$.

$$\delta s < w \tag{11}$$

Obviously, it is generally dangerous for the vehicle to move along a candidate direction without examining its neighboring sectors. And moving along a candidate direction whose four neighboring sectors obstacle densities are smaller than the threshold is always safe, since this enhanced sector's arc $l = 5\delta s$ which equals $2.18m$ is bigger than vehicle width.

2) Achievement to target pose:

The target pose (x_T, y_T, θ_T), including the target location (x_T, y_T) and the target heading angle θ_T, is the goal state of planning. The cost function $g(c_n)$ is proportional to $g_0(c_n)$.

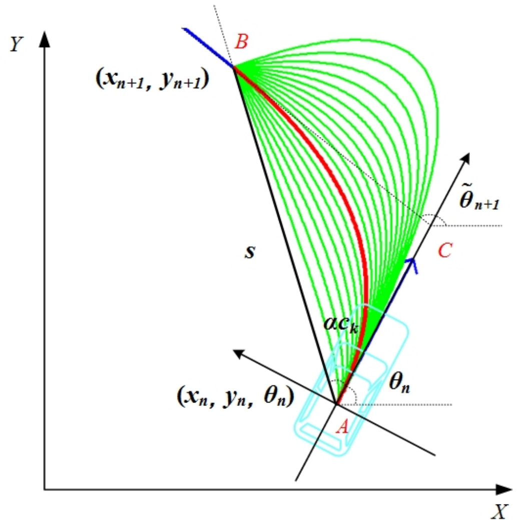


Fig. 6. Calculation of heading angle for projected location.

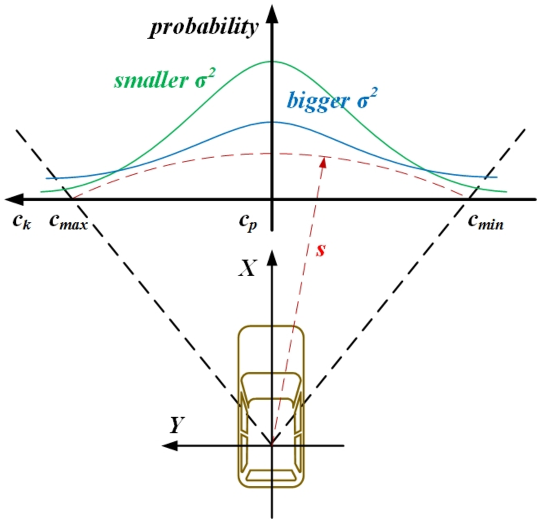


Fig. 7. Smoothness of the selected direction sequence.

The first term of the function $g_0(c_n)$, which represents the cost resulted from the difference square of a candidate direction c_n and the direction k_t, is an important part, because the autonomous vehicle is also a goal-oriented robot. The smaller this term is, the more probable the candidate direction will guide the vehicle toward target location. And this term results that the trajectory generated based on the proposed algorithm reaches the target location (x_T, y_T).

The second term of the function $g_0(c_n)$ is associated with the difference square of the target heading angle θ_T and the estimation of heading angle for the next searched location $\tilde{\theta}_{n+1}$, the smaller this term is, the more the heading angle of the next searched location is closer to the target heading angle θ_T. Estimation $\tilde{\theta}_{n+1}$ can be obtained based on Eq. 12, and this process is shown in Fig. 6. In detail, under the condition that the next location (x_{n+1}, y_{n+1}) and current vehicle pose (x_n, y_n, θ_n) are known. The estimation of the heading angle θ_{n+1} can be acquired by finding the curve

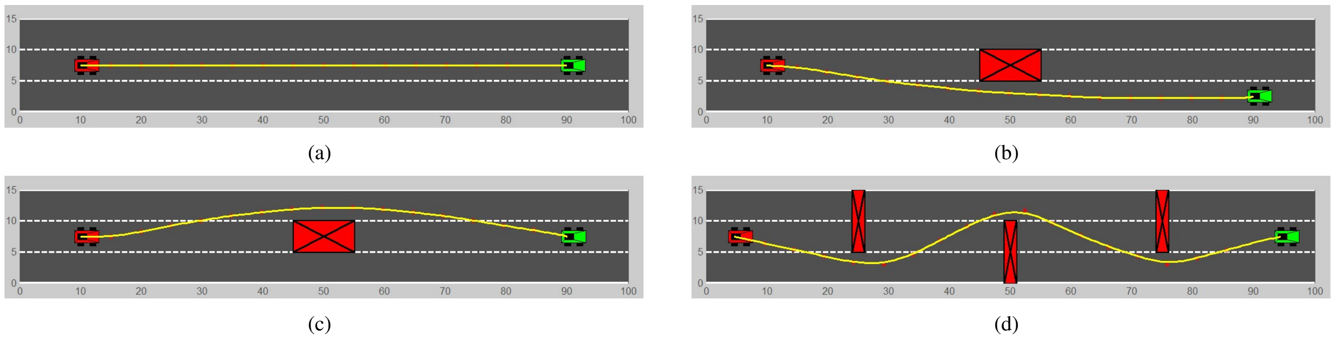


Fig. 8. Verification in structured environments. (a) Lane-keeping. (b) Lane-change. (c) Overtaking. (d) Multi-obstacle avoidance.

whose maximum curvature is minimum among a cluster of 2-order Bezier curves generated based on the above condition.

$$\tilde{\theta}_{n+1} = \arg\min Kappa_{max}(\theta_{n+1}) \tag{12}$$

where $Kappa_{max}$ denotes the maximum curvature of a 2-order Bezier curve.

3) Smoothness of trajectory:

The cost function $g(c_n)$ is inversely proportional to $G(c_n)$ which is a discrete Gaussian function with mean value $\hat{c}_{n-1}$ and variance value σ^2. As shown in Fig. 7, it represents the probability of being selected as motion direction for each discrete direction. In short, the closer to the previously selected direction $\hat{c}_{n-1}$, the more possible the candidate direction c_n is selected as final motion direction, while the further to the previously selected direction $\hat{c}_{n-1}$, the less possible the candidate direction c_n is selected as final motion direction. It simulates a kind of memory to the previously selected direction. An important benefit of this improvement is that it satisfies with the dynamics easily even in high speed cases.

In brief, our algorithm takes vehicle's width into account by examining neighboring sectors of the candidate direction to ensure safety. The function $g_0(c_n)$ is responsible for the achievement to the target pose, while $G(c_n)$ guides the vehicle moving along the direction which is closer to the previously selected direction, and it is characterized by a mechanical memory, with its help, a smooth trajectory will be generated.

It is the ratio of the coefficient μ_1 and μ_1 contributes to achievement to the target location or the target heading angle, but not their absolute values. To guarantee the achievement to the target location preferentially, the condition $\mu_1 > \mu_2$ must be satisfied. Our algorithm adopts a three-level of σ^2 to adapt to various scenarios. For example, when the vehicle turns right or left at an intersection, the high-level σ^2 allows it to make a big turn, and the middle-level σ^2 is useful to make a smooth turn when the vehicle conducts lane-changing and overtake, otherwise lane-keeping needs a low-level σ^2. We can get these scenario information from fusion data consisting of GPS, lanes [15] and obstacles.

D. B-spine Curve

Based on the selected direction and the searching step length, we can get a trajectory point. Repeat this process, a parse location set is generated. However, the set is of shortage of direction and velocity information to guide the vehicle moving to the target pose. Thus, a dense state set including location, direction and velocity is required.

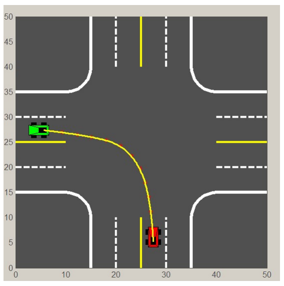
This can be achieved by a two-step procedure. First, the B-spline curve, which is generated based on the location set, is used to obtain a dense smooth trajectory x, the direction and curvature of each point on the trajectory x can be obtained during this process. Second, combining the trajectory curvature curve with the vehicle controller capability, an expectation velocity curve can be generated. Hereto, the control sequence u will be obtained indirectly.

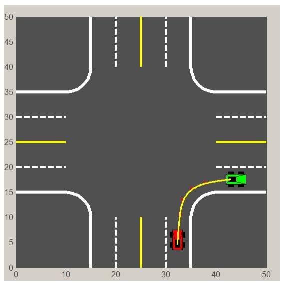
IV. EXPERIMENTAL RESULTS

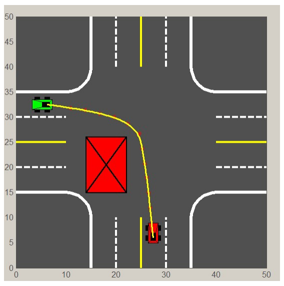
In this section, we evaluate performances of the proposed algorithm in various urban scenarios. Both structured and unstructured environments are selected to validate the motion planning functionality of the proposed algorithm.

For structured environments, four kinds of common traffic scenarios, including lane-keeping, lane-change, overtaking and multi-obstacle avoidance, are presented. Fig. 8(a) illustrates the result of the proposed approach for lane-keeping scene, in which the poses of red and green vehicles denote the initial state and the goal state, respectively, and the yellow line which approximates a straight line denotes the trajectory generated by our method. Fig. 8(b) and Fig. 8(c) respectively show the results of the algorithm for lane-change and overtaking scenes. And our method performs well in the multi-obstacle avoidance scene shown in Fig. 8(d), which is a big challenge for parametric curves.

For unstructured environments, a set of intersection scenes, including turning-left, turning-right, single-obstacle avoidance, multi-obstacles avoidance, round-island and cone-guidance are selected to evaluate the proposed algorithm. Fig. 9(a) presents the result of our method for an intersection turning-left scene, because of the reachable active region and the cost function proposed in this paper, the trajectory is smoother and satisfies the non-holonomic constraints well. Fig. 9(b) presents the planned trajectory for an intersection turning-right scene, which is more difficult due to bigger


(a)


(b)


(c)

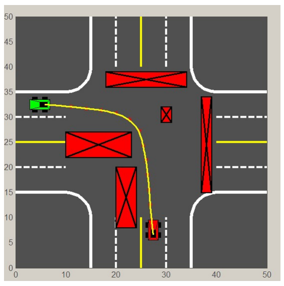
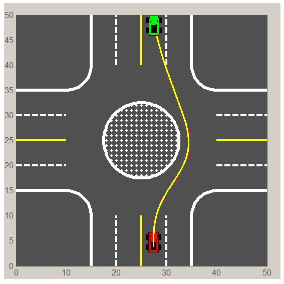

(d)

Fig. 9. Verification in unstructured environments. (a) Turning-left. (b) Turning-right. (c) Single-obstacle avoidance. (d) Multi-obstacle avoidance.


(a)

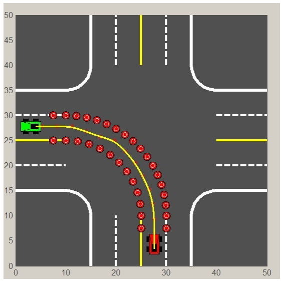

(b)

Fig. 10. Verification in structured environments. (a) Round-island. (b) Cone-guidance.

curvature. Fig. 9(c) and Fig. 9(d) present the results of the proposed method for single-obstacle avoidance and multi-obstacle avoidance scenes in an intersection. They show the enhanced obstacle avoidance capacity of our algorithm. And Fig. 10(a) presents the result of our method for a round-island scene while Fig. 10(b) is planning result for a cone-guidance scene.

The proposed algorithm is efficient due to two reasons. First, the polar histogram H is structured around the VCP, and is independent of the selected direction, the algorithm can be accelerated by matrix operations among several look-up tables [9]. Among these tables, two invariable tables are used to store the $\beta_{i,j}$ and $d_{i,j}$ respectively, and a variable one is used to store the $c_{i,j}$. Another reason is that the active region proposed in this paper is smaller than original ones, so a fewer number of obstacle grids are taken in account.

V. CONCLUSION

This paper presents a constrained VFH algorithm to improve the performance of autonomous vehicle in various environments, and several improvements are made. Firstly, a reachable active region for the autonomous vehicle is designed. Secondly, a discrete low-pass filter mask is used to enhance the safety of motion planning. The last and most important improvement is an improved cost function. Experiments were conducted by applications of the proposed algorithm to the autonomous vehicle in various urban scenarios. Experimental results demonstrate that the proposed VFH algorithm can generate smoother and collision-free trajectories efficiently.

In the future, the maximum active distance ρ_{max} would be modulated adaptively based on more abundant perceptual information. Since constant ρ_{max} which equals maximum perceptual distance in this paper results in failure accidentally.

REFERENCES

[1] S. M. LaValle, *Planning Algorithms*. Cambridge University Presss, 2006.

[2] T. M. Howard, M. Pivtoraiko, R. A. Knepper, and A. Kelly, "Model-predictive motion planning," *IEEE Robotics and Automation Magazine*, 2014.

[3] A. Stentz, "Optimal and efficient path planning for partially-known environments," in *1994 IEEE International Conference on Robotics and Automation*, 1994, pp. 3310–3317.

[4] S. M. LaValle and J. J. Kuffner, "Randomized kinodynamic planning," *The International Journal of Robotics Research*, vol. 20, no. 5, pp. 378–400, 2001.

[5] I. Miller, S. Lupashin, N. Zych, P. Moran, B. Schimpf, A. Nathan, and E. Garcia, "Cornell university's 2005 DARPA grand challenge entry," *Journal of Field Robotics*, vol. 23, no. 8, pp. 625–652, 2006.

[6] L. Han, H. Yashiro, H. T. N. Nejad, Q. H. Do, and S. Mita, "Bezier curve based path planning for autonomous vehicle in urban environment," in *2010 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2010, pp. 1036–1042.

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

[8] L. Ma, J. Xue, K. Kawabata, J. Zhu, C. Ma, and N. Zheng, "Efficient sampling-based motion planning for on-road autonomous driving," *IEEE Transactions on Intelligent Transportation Systems*, 2014.

[9] J. Borenstein and Y. Koren, "The vector field histogram-fast obstacle avoidance for mobile robots," *IEEE Transactions on Robotics and Automation*, vol. 7, no. 3, pp. 278–288, 1991.

[10] I. Ulrich and J. Borenstein, "VFH+: Reliable obstacle avoidance for fast mobile robots," in *1998 IEEE International Conference on Robotics and Automation*. IEEE, 1998, pp. 1572–1577.

[11] I. Ulrich and J. Borenstein, "VFH*: Local obstacle avoidance with look-ahead verification," in *2000 IEEE International Conference on Robotics and Automation*. IEEE, 2000, pp. 2505–2511.

[12] D. Jiea, M. Xueming, and P. Kaixiang, "IVFH*: Real-time dynamic obstacle avoidance for mobile robots," in *2010 11th International Conference on Control Automation Robotics and Vision (ICARCV)*. IEEE, 2010, pp. 844–847.

[13] S. Karaman and E. Frazzoli, "Optimal kinodynamic motion planning using incremental sampling-based methods," in *2010 49th IEEE Conference on Decision and Control (CDC)*. IEEE, 2010, pp. 7681–7687.

[14] A. Elfes, "Using occupancy grids for mobile robot perception and navigation," *Computer*, vol. 22, no. 6, pp. 46–57, 1989.

[15] D. Cui, J. Xue, S. Du, and N. Zheng, "Real-time global localization of intelligent road vehicles in lane-level via lane marking detection and shape registration," in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2014)*. IEEE, 2014, pp. 4958–4964.