Combing VFH with bezier for motion planning of an autonomous vehicle

Feng Ye¹, Jing Yang², Chao Ma³ and Haijun Rong⁴

¹School of Electronic Information Engineering, Xi’an Jiaotong University, Xi’an, China, yf87953246@stu.xjtu.edu.cn
²School of Electronic Information Engineering, Xi’an Jiaotong University, Xi’an, China, jasmine_yang@126.com
³School of Electronic Information Engineering, Xi’an Jiaotong University, Xi’an, China, Machao0919@stu.xjtu.edu.cn
⁴School of Aerospace, Xi’an Jiaotong University, Xi’an, China, hjrong@xjtu.edu.cn

Abstract. Vector Field Histogram (VFH) is a method for mobile robot obstacle avoidance. However, due to the nonholonomic constraints of the vehicle, the algorithm is seldom applied to autonomous vehicles. Especially when we expect the vehicle to reach target location in a certain direction, the algorithm is often unsatisfactory. Fortunately, the Bezier Curve is defined by the states of the starting point and the target point. We can use this feature to make the vehicle in the expected direction. Therefore, we propose an algorithm to combine the Bezier Curve with the VFH algorithm, to search for the collision-free states with the VFH search method, and to select the optimal trajectory point with the Bezier Curve as the reference line. This means that we will improve the cost function in the VFH algorithm by comparing the distance between candidate directions and reference line. Finally, select the closest direction to the reference line to be the optimal motion direction.

1. Introduction

Motion planning is an important part of autonomous vehicle research. It is problem is to generate continuous motion that connects start state and target state, while avoiding collision with known obstacles [1]. The research in robot motion planning can be traced back to the late 60’s. Over so many years, many motion planning algorithms were proposed, such as A*, D* [2] and RRT [3]. However, it is quite difficult to take the vehicle’s non-holonomic constraints into considerations.

In contrast, the parameter curve is simpler and it can satisfy the dynamic constraints well. Bezier Curve is a typical example. Cornell team proposed the two-dimensional cubic Bezier curve to represent trajectories. Long Han present a trajectory planning algorithm based on Bezier curve [4]. For intelligent vehicles, the ability of avoiding obstacles becomes very important, so more and more researcher focus on the study of obstacle avoidance algorithm. For example, the MIT’s intelligent vehicle uses RRT as trajectory planning to avoid obstacle. However, it is still difficult for intelligent vehicles to reach an expected state in complex environment.

The Bezier Curve earliest applied to automotive body design by Pierre Bezier in 1962. It is a typical method of generating trajectories by analytic method. The advantage of Bezier Curve is that it can be generated fast and it is smooth enough which is very important for intelligent vehicle when there is no obstacle or the obstacles are sparse. However, when the curve is generated, only the pose and curvature constrains of the endpoints are taken into account, the obstacles information are not taken into account. Therefore, the method itself does not have the obstacle avoidance function. What’s more,
the Bezier curve does not have the ability of continuous turning, so this method cannot solve motion planning problem in the complex environment.

VFH algorithm was proposed by Johann Borenstein for the mobile robot’s motion planning [5]. It is a method based on obstacle grid which used to represent environment. VFH algorithm has strong ability of avoiding obstacles and its planning space is continuous. After that, many improved versions are emerging. Iwan Ulrich et al. proposed VFH+ algorithm and VFH* algorithm which have large influence. Its main contribution is to add the cost function to the original VFH algorithm [6]. Dong Jie et al. proposed IVFH* for dynamic obstacles avoidance [7]. However, the vehicle is an non-holonomic constrains system, some states of active region in the original algorithm are unreachable for the vehicle. What’s more, we often hope that the vehicle reaches the target point with the heading we want, but the VFH algorithm performs not very well.

Through the description of VFH algorithm and Bezier Curve. We can see that the Bezier Curve is an analytic method, VFH algorithm is based on the sampling pattern and generates a trajectory consisting of discrete points. So in this paper, we propose an algorithm that combines VFH algorithm with Bezier Curve. It is also a combination of discrete method and analytic method. We combine the advantages of both, so that it can be applied to the intelligent vehicle’s motion planning to adapt to complex traffic scenes.

2. Problem Formulation

This section we describe the definition of motion planning. As shown in Fig 1, we present the state space of system as compact sets $X \subset R^n$ and the control space as $U \subset R^m$. Then we have [8]

$$x(t) = f(x(t), u(t)), x(0) = X_{init}, x(t_f) = X_{target}$$

Where $x(t) \in X, u(t) \in U$ and $X_{init}$ is the initial state, $X_{target}$ is the target state. $X_{obs} \subset X$ denotes that these states collide with obstacles, so collision-free state can be defined as $X_{free} \subset X \setminus X_{obs}$. The motion planning task is to find an state sequence that satisfies the following condition:

$$x(t) \in X_{free}, \forall t \in [0, t_f]$$

$$x(0) = X_{init}$$

$$x(t_f) = X_{target}$$

$$u(t) \in U$$

Where $x$ is called the trajectory and $x(t)$ is the feasible state for our intelligent vehicle and $u$ is called the control input sequence and $u(t)$ is also feasible for the vehicle.

Figure 1. Motion planning diagram
3. Algorithm Description
In this section, the algorithm based on Bezier curve and VFH algorithm is presented. We first make a brief introduction about Bezier Curve and VFH algorithm, and then describe our algorithm in details. Our algorithm is implemented in the following steps: Firstly, the Bezier curve clusters are generated according to the start point and target point, and then select the smoothest Bezier curve as the reference line. Secondly, we use VFH algorithm to find all the candidate directions. Thirdly, we select the direction closest to the reference line as the new motion direction and determine whether the vehicle to reach the end. If not, we will return to step two until the vehicle reaches the end point.

3.1. Third-Order Bezier Curve
Low-order Bezier Curve is simple, but the planning ability is bad. High-order Bezier Curve has strong planning ability, but the control point selecting is lack of basis and difficulty to control. To sum up, we choose the four control points of the third-order Bezier Curve planning.

![Third-order Bezier Curve](image)

**Figure 2. Third-order Bezier Curve**

Given four control points \( P_0(x_0, y_0, \theta_0) \), \( P_1(x_1, y_1) \), \( P_2(x_2, y_2) \) and \( P_3(x_3, y_3, \theta_3) \) as shown in Fig.2, we can get the X and Y coordinates of third-order Bezier Curve:

\[
x(t) = \sum_{i=0}^{3} b_{i,3}(t)x_i
\]

\[
= x_0(1-t)^3 + 3x_1t(1-t)^2 + 3x_2t^2(1-t) + x_3t^3, t \in [0,1]
\]

\[
y(t) = \sum_{i=0}^{3} b_{i,3}(t)y_i
\]

\[
= y_0(1-t)^3 + 3y_1t(1-t)^2 + 3y_2t^2(1-t) + y_3t^3, t \in [0,1]
\]

(1), (2) Can be organized into third-order polynomials with respect to the parameter \( t \):

\[
x(t) = [(x_3 - x_0) + 3(x_1 - x_2)]t^3 + 3(x_0 - 2x_1 + x_2)t^2
\]

\[
+ 3(x_1 - x_0)t + x_0, t \in [0,1]
\]

\[
y(t) = [(y_3 - y_0) + 3(y_1 - y_2)]t^3 + 3(y_0 - 2y_1 + y_2)t^2
\]

\[
+ 3(y_1 - y_0)t + y_0, t \in [0,1]
\]
For convenience of expression, (3), and (4) is reduced to the following form:

\[ x(t) = a_1 t^3 + a_2 t^2 + a_3 t + a_0, t \in [0, 1] \]  
\[ y(t) = b_1 t^3 + b_2 t^2 + b_3 t + b_0, t \in [0, 1] \]

In (5), (6):

\[ a_1 = (x_3 - x_0) + 3(x_1 - x_2), \quad a_2 = 3(x_0 - 2x_1 + x_2) \]
\[ a_3 = 3(x_1 - x_0), \quad a_0 = x_0 \]
\[ b_2 = (y_3 - y_0) + 3(y_1 - y_2), \quad b_3 = 3(y_0 - 2y_1 + y_2) \]
\[ b_1 = 3(y_1 - y_0), \quad b_0 = y_0 \]

The third-order Bezier Curve is shown in Fig. 2. Where \( P_0, P_1, P_2 \) and \( P_3 \) correspond to the curve four control point. \( \theta_0 \) is the inclination angle of the straight line of the line segment \( P_0P_1 \), \( \theta_2 \) is the inclination angle of the straight line of the line segment \( P_2P_3 \). In the following, we will briefly introduce some important features of the Bezier Curve:

First, Bezier Curves are guaranteed to pass through the first and last control points. If the first control point corresponds to the starting position of the vehicle, the last control point corresponds to the vehicle’s target position, then the Bezier Curve can ensure that the trajectory over the initial position and ultimately reach the target position. Second, Bezier Curves endpoints \( P_0 \) and \( P_3 \) are tangent to edges \( P_0P_1 \) and \( P_2P_3 \) of control polygon \( P_0P_1P_2P_3 \) respectively. If the tangent of the Bezier Curve and the heading of the motion trajectory are linked, then the Bezier Curve can satisfy the heading of the motion trajectory at the starting state \( P_0 \) and target state \( P_3 \). Third, the third-order Bezier Curves are not only self-continuous, but also their first derivative and second derivative are continuous, which fully meet the requirements of trajectory continuity in restricted vehicle systems. Fourth, it is convenient to convert the kinematic constraints into curve constraints, thus adding the vehicle kinematic constraints to the motion planning process.

From Fig. 2, we can see that control point \( P_1 \) must be tangent at point \( P_0 \), control point \( P_2 \) must be tangent at point \( P_3 \). But the distance between \( P_0 \) and \( P_1 \) can be changed, and the distance between \( P_2 \) and \( P_3 \) can also be changed. So by changing the location of the control point \( P_1 \) and \( P_2 \), we can get the curves cluster connecting \( P_0 \) and \( P_3 \).

After obtaining the cluster of curves, we need to further select the smoothest curve. The smoothest curves can be abstracted into optimization problems:

\[ \text{tra}_{\text{optimal}} = \arg \min \sum_{i} \kappa^2_{i} \]  

Where \( \text{tra}_{\text{optimal}} \) denotes the smoothest curve, \( \sum_{i} \kappa^2_{i} \) denotes the sum of square curvature at all points on the i-th curve if we use point sets to represent a curve.

Next we derive the curvature calculation formula. According to the definition of curvature:

\[ \kappa(t) = \frac{[x'(t)y''(t) - x''(t)y'(t)]}{[x'(t)^2 + y'(t)^2]^{3/2}} \]

so the curvature of start point \( P_0 \) is:

\[ \kappa_{P_0} = \frac{[x'(0)y''(0) - x''(0)y'(0)]}{[x'(0)^2 + y'(0)^2]^{3/2}} \]
\[
\kappa(0) = \frac{\left| x'(0)y'(0) - x'(0)y'(0) \right|}{\left| x'(0)^2 + y'(0)^2 \right|^{3/2}} \\
= \frac{2(x_0y_1 + x_1y_2 + x_2y_0 - x_0y_2 - x_1y_0 - x_2y_1)}{3[(x_1-x_0)^2 + (y_1-y_0)^2]^{3/2}} \\
= \frac{2}{3} \frac{(y_3-y_0)\cos\theta_1 - (x_3-x_0)\sin\theta_1 + d_2\sin(\theta_1-\theta_2)}{d_1^2}
\]

the curvature of target point \( P_3 \) is:

\[
\kappa(1) = \frac{\left| x'(1)y'(1) - x'(1)y'(1) \right|}{\left| x'(1)^2 + y'(1)^2 \right|^{3/2}} \\
= \frac{2(x_1y_2 + x_2y_3 + x_3y_1 - x_2y_3 - x_3y_2 - x_1y_3)}{3[(x_2-x_1)^2 + (y_2-y_1)^2]^{3/2}} \\
= \frac{2}{3} \frac{(x_1-x_0)\sin\theta_2 - (y_1-y_0)\cos\theta_2 + d_1\sin(\theta_1-\theta_2)}{d_2^2}
\]

so the relationship between \( d_1 \) and \( d_2 \) is:

\[
d_2 = \frac{(x_1-x_0)\sin\theta_2 - (y_1-y_0)\cos\theta_2 + \frac{3}{2} \kappa(0)d_1^2}{\sin(\theta_1-\theta_2)}
\]

At this point, we can calculate the sum of square curvature of each curve and select the smoothest curve as the reference line.

3.2. Combining VFH Algorithm with Bezier Curve

We use VFH algorithm to search collision-free region and reference line to select the optimal motion direction. So we will introduce something about VFH to show how we find the collision-free region. This section includes the following contents: First, the definition of active region. Second, find collision-free region. Third, select the optimal motion direction. Fourth, smooth the trajectory.

First, we will introduce the active region. Whether the rectangular active region of the VFH algorithm or the circular active region of the VFH+ algorithm cannot satisfy the vehicle kinematics constraints. Because some of the status points in the active region are unreachable for the vehicle.

Taking into account the kinematics constraints of the vehicle, we use a new active region which suitable for the vehicle. As shown in Fig. 3, the new active region is defined by a sector window in [9].

**Figure 3.** New active region

The sector active region \( Z \) can be expressed as formulas:
\[ Z = P(\rho, \varphi) \]  

\( P \) represents a polar coordinate system in which the center of the rear axle of the vehicle is the origin of the coordinates and the heading angle direction is 0°. \( \rho \) denotes the distance range of the active area, and \( \rho \in [0, \rho_{\text{max}}], \rho_{\text{max}} \) denotes the maximum active distance, \( \varphi \) denotes the angular range of the active region, and has

\[ \varphi \in [-\arcsin \frac{s}{2r_{\text{min}}}, \arcsin \frac{s}{2r_{\text{min}}} ] \]  

In (14), \( s \) represents the search step, that is, the distance traveled by the vehicle every time. \( r_{\text{min}} \) represents the inherent minimum turning radius of the vehicle, which can be approximated as a constant in the case where the vehicle speed is not particularly high. The trajectory point must be on the arc \( R = P(s, \varphi) \), obviously, all the points on the arc \( R \) are reachable to the vehicle.

By comparison we can see that the sector active region is smaller than the original active region. So it considers fewer obstacle grids, making the algorithm more efficient.

Second, we need to find the collision-free region. In the VFH algorithm, a two-dimensional grid is used to represent the environment [10]. Each grid contains a probability value. This probability value reflects the presence of obstacles in the grid credibility, the higher the probability value, indicating the presence of obstacles here the greater the possibility.

In order to search for the collision-free region, first, we need to represent the grid occupied by obstacles inside the sector active region as a vector whose direction depends on the relative position of the grid and the vehicle center point (VCP) [5]:

\[ \beta_{i,j} = \arctan( \frac{y_i - y_0}{x_i - x_0} ) \]  

where \( (x_0, y_0) \) represents the current position of the VCP, and \( (x_i, y_i) \) represents the position of the obstacle grid \( C_{i,j} \). The magnitude \( m_{i,j} \) of the obstacle grid \( C_{i,j} \) is defined by:

\[ m_{i,j} = (c_{i,j})^2(a - bd_{i,j}) \]  

In (16), \( a, b \) are positive constants and chosen such that \( a - b d_{\text{max}} = 0 \) where \( d_{\text{max}} = \rho_{\text{max}}, c_{i,j} \) denotes the certainty value of the obstacle grid \( C_{i,j}, d_{i,j} \) represents the distance between obstacle grid \( C_{i,j} \) and VCP.

**Figure 4.** Dividing the region into several sectors
Next, we divide the active region into several sectors and count the obstacle density for each sector, we can get the initial polar histogram. As shown in Fig. 4, selecting an angular resolution $\alpha$ divides the vehicle’s active region into $n = \frac{360}{\alpha}$ sectors. In order to ensure that $n$ is a positive integer, angular resolution $\alpha$ is often taken as a divisor of 360. Corresponding to each sector, there is an index $k$ and an obstacle density $h_k$, where $k$ is related to the discrete direction $\rho = k \cdot \alpha$ of the sector and defined as (17):

$$k = \text{INT} \left( \frac{\beta_{ji}}{\alpha} \right)$$

(17)

$\text{INT}$ stands for rounding. The obstacle density $h_k$ of the sector $k$ is defined as (18):

$$h_k = \sum_{C_{ji}} m_{ni}$$

(18)

By calculating $h_k$, we can get the initial polar histogram. However, due to the discrete nature of the obstacle grid, the initial histogram is uneven, which may affect the determination of the final motion direction. So we introduce a low-pass filter to smooth the initial polar histogram.

Finally, we use a threshold to determine candidate directions based on the smoothed polar histogram [11]. If the obstacle density of sector lowers than the threshold, we regard the sector as candidate directions. The sector is also the collision-free region.

Third, we use the cost function to find the optimal motion direction. Through the threshold we can see from the polar histogram which direction is collision-free? This section we will describe how to use the reference line (Bezier Curve) to select the best direction.

When we select the best direction, we should take the width of vehicle into account [9]. Assuming that the search step of the vehicle is 8 meters and the angle of the active region is 60 degrees. If the active region is divided equally into 20 sectors, the arc length corresponding to each sector is about 0.42 meters. The width of vehicle is about 2 meters, so the arc length of five sectors to meet the vehicle’s width. This require that the two sectors of the selected sector, which are adjacent to the left, and two sectors(five sectors in total) that are adjacent to the right are also collision-free. We use $C_n \in C_{\text{free}}$ to represents that the n-th sector is collision-free. If the n-th sector satisfies the requirement of vehicle’s width, we can express it as

$$\{C_{n-2}, C_{n-1}, C_n, C_{n+1}, C_{n+2}\} \in C_{\text{free}}$$

(19)

For each candidate direction that satisfies (17), we can calculate the shortest distance between this direction and reference line through (18):

$$d_{i,\text{min}} = \min \{d_{i,1}, d_{i,2}, \ldots, d_{i,n}\}, i = 1, 2, \ldots, m$$

(20)

This is also our cost function. We assume that the reference line is represented by $n$ points, and the number of sectors satisfying (19) is $m$ . $i$ denotes the i-th sector satisfying (19), $d_{i,n}$ denotes the distance between the i-th sector and the n-th point of the reference line, $d_{i,\text{min}}$ denotes the shortest distance between the i-th sector and the reference line.

Finally, we can calculate which sector is closest to the reference line by (21):

$$k = \arg \min_i (d_{i,\text{min}}), k = 1, 2, 3, \ldots, m$$

(21)

And the k-th sector is our optimal motion direction.
Fourth, we need to smooth the trajectory. In general, the trajectory of the vehicle is represented by a set of points. The trajectory points we obtain from procedure three is sparse and discrete. So we use the B-spline curve based on the trajectory points to smooth the trajectory [12]. The basis function of third-order B-spline curve is: sequence We use the trajectory point obtained from procedure 3 as the control point of B-spline curve. Then we can obtain a dense smooth trajectory. This trajectory is feasible for vehicle.

4. Experiment Result

In this section, we will show the performance of proposed algorithm in different traffic scenarios. There are two kinds of environment, structured environment that the road shape is regular and unstructured environment which has irregular shape of the road and unclear edge.

The pictures in Fig. 5 show the performance of our algorithm in structured environment. Fig. 5(a) shows that when the traffic scenario is very easy without any obstacles, the vehicle always runs straight in a lane and always remains in the middle of the lane. The position of the yellow vehicle denotes the initial state and the position of the blue vehicle denotes the target state. In Fig. 5(b), there is an obstacle in the lane, so the vehicle change the lane to right then change to the original lane when exceed the obstacle. We can see that when the vehicle reach the target state, the vehicle’s heading satisfies our requirement which we expect to be parallel to the lane line. This avoids causing shock. Fig. 5(c) shows a complex scenario with multiple obstacles. Our algorithm performs well in this scene and the heading of the target state is what we want.

![Figure 5](image-url)

Figure 5. (a)Lane-keeping. (b)One obstacle avoidance. (c)Multi-obstacles avoidance.
5. Conclusion

This paper presents a algorithm combined VFH algorithm with Bezier Curve to enhance the ability of intelligent vehicles to adapt to complex environments. Our algorithm has advantages from VFH algorithm and Bezier Curve. VFH algorithm has a strong ability to avoid obstacles which can find collision-free trajectories in complex multi-obstacles environment. Bezier Curves are not only self-continuous, but also their first derivative and second derivative are continuous, which fully meet
the requirements of trajectory continuity of the constrained vehicle system. So the trajectories generated by our algorithm are continuous and have strong ability to avoid multi-obstacles.

6. Acknowledgment
This work is funded in part by National Nature Science Foundation of China (Grant No. 61403300), 973 Program (No. 2015CB351703), National Science Council of ShaanXi Province (Grant No. 2014JM8337) and the Fundamental Research Funds for the Central Universities.

7. References
[1] Howard T M, Pivtoraiko M, Knepper R A, et al. Model-Predictive Motion Planning[J], IEEE ROBOTICS & AUTOMATION MAGAZINE, 2014.
[2] Ferguson D, Stentz A. Field D*: An Interpolation-based Path Planner and Replanner[C]/International symposium on Robotics Research (ISRR), 2005.
[3] L. Ma, J. Xue, K. Kawabata, J. Zhu, C. Ma, N. Zheng, “Efficient Sampling-based Motion Planning for On-road Autonomous Driving,” IEEE Transactions on Intelligent Transportation Systems, 2014.
[4] Han L, Yashiro H, Nejad H T N, et al. Bezier curve based path planning for autonomous vehicle in urban environment[C], Intelligent Vehicles Symposium (IV), 2010 IEEE. IEEE, 2010: 1036-1042.
[5] Borenstein J, Koren Y. The vector field histogram-fast obstacle avoidance for mobile robots [J], Robotics and Automation, IEEE Transactions on, 1991, 7(3): 278-288.
[6] Ulrich I, Borenstein J. VFH+: Reliable obstacle avoidance for fast mobile robots[C], Robotics and Automation, 1998. Proceedings. 1998 IEEE International Conference on. IEEE, 1998, 2: 1572-1577.
[7] Jiea D, Xueming M, Kaixiang P. IVFH*: Real-time dynamic obstacle avoidance for mobile robots[C], Control Automation Robotics & Vision (ICARCV), 2010 11th International Conference on. IEEE, 2010: 844-847.
[8] Karaman S, Frazzoli E. Optimal kinodynamic motion planning using incremental sampling-based methods[C], Decision and Control (CDC), 2010 49th IEEE Conference on. IEEE, 2010: 7681-7687.
[9] P. Qu, J. Xue, L. Ma, C. Ma, “A Constrained VFH Algorithm for Motion Planning of Intelligent Vehicles” IEEE Transactions on Intelligent Transportation Systems, 2014.
[10] Elfes A. Using occupancy grids for mobile robot perception and navigation [J], Computer, 1989, 22(6): 46-57.
[11] Theodor Nitulescu, Stefan Talu, Applications of descriptive geometry and computer aided design in engineering graphics, Cluj-Napoca, Romania, Risoprint Publishing house, 2001, ISBN 973-656-102-X.
[12] Stefan Talu, Cristina Racoea, Axonometric representations with applications in technique, Cluj-Napoca, Romania, MEGA Publishing house, 2007. ISBN 978-973-7867-98-8.