Vehicle path planning with maximizing safe margin for driving using Lagrange multipliers

Quoc Huy Do, Hossein Tehrani Nick Nejad, Keisuke Yoneda, Sakai Ryohei and Seiichi Mita

Abstract—We propose a path planning method for autonomous vehicle in cluttered environment with narrow passages. Different from traditional methods, we use a learning approach based on RBF kernel SVM to maximize the safety margin for driving. We use the Lagrange multipliers of SVM dual model to find most critical points in map and generate optimized hyperplane for path. The method is implemented on autonomous vehicle for outdoor parking and compared to well-known method in autonomous vehicle literatures. The experiments prove that the method is able to generate smooth and safe path in shorter time compared to other methods.

I. INTRODUCTION

Recent years, autonomous vehicles have become an increasingly important research topic in various civilian operations. A critical part of vehicle autonomy consists of a path planning and navigation that enables car to safely maneuver through different type of environments. General path planning methods have been studied in the literatures and it becomes common knowledge in the different fields from computer science, artificial intelligence to autonomous systems. Different methods from simplest to more complicated path planning methods involving different constraints (such as time or non-holonomic constraints) have been proposed in detail by many authors [1]-[4].

Intrinsically, a path planning task is the problem of looking for a feasible navigation from start point to goal point. The optimal paths can maximize the obstacle clearance, minimize steering (maximize smoothness), minimize gear change or brake or travelling distance and time. Dijkstra’s shortest path algorithm and A* are common solutions to find shortest path based on decision criteria [1], [3],[4]. In 2007 DARPA Challenge, CMU team (best ranking team) applied a path planning method based on Anytime Dynamic A* algorithm [5]. In their method, two kinds of path sets are prepared including smooth trajectories and sharp trajectories and choose an adequate path considering optimization criteria. Their approach generates good results in practice for urban driving. Other common methods in autonomous driving literature, such as Rapidly-exploring Random Trees (RRTs) or Probabilistic Roadmap (PRM), generate nodes randomly to construct paths which satisfy vehicle kinematic or dynamic requirements though they do not consider path clearance [2], [6], [7]. In 2007 DARPA Urban Challenge the MIT team used an algorithm based on the RRT to develop a path planner [8]. This method is effective for obstacle avoidance and quickly discovers the environment and finds a path to the goal root in the random steps though the path is not always guaranteed to be an optimal and safe one. Stanford team used a mixture of approaches in their overall system [9]. Based on the DARPA supplied data, initial trajectories for each waypoint are planned. The final path is selected considering different objective such as the closeness to the center of the road, distance to obstacles and vehicular motion constraints computed from a vehicle model. DARPA urban challenge teams developed path planning algorithms which are effective for route maps, such as urban roads and highways. However, in cluttered, narrow passages or highly dynamic environments, these methods are not suitable since they do not guarantee the maximum clearance and safety for the vehicle path. In terms of safety, path planning, proposed methods tend to maximize the distance between the vehicle and obstacles on the map and generate maximum clearance path for vehicle. Typical approach such as Voronoi diagram [10], [11] or the potential field [4], [12] are already proposed in the literature. Bhattacharya and Gavriloova proposed a clearance-based shortest path planner based on a Voronoi diagram [11]. In their method, all the edges in the Voronoi diagram which have a clearance less than the minimum clearance required will be removed, and then Dijkstra’s algorithm is applied to determine the shortest path in the roadmap. However, these methods have limitation for size of map and do not guarantee to generate optimized path to the goal position. Moreover, the above mentioned methods cannot find and cluster the critical points in the map within a given path. The main advantages and disadvantage of the typical path planning methods can be sum up in table 1.

<table>
<thead>
<tr>
<th>Method</th>
<th>Advantage</th>
<th>Disadvantage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dijkstra</td>
<td>Optimal shortest path</td>
<td>Slow, no clearance guarantee.</td>
</tr>
<tr>
<td>A*</td>
<td>Fast with good heuristic.</td>
<td>The result path tends to stay close to obstacles.</td>
</tr>
<tr>
<td>RRT</td>
<td>Fast</td>
<td>No shortest or clearance guarantee.</td>
</tr>
<tr>
<td>Potential Field</td>
<td>Path clearance is guaranteed</td>
<td>Have local optimal</td>
</tr>
<tr>
<td>Voronoi Diagram</td>
<td>Path clearance is guaranteed</td>
<td>The result path is not smooth or does not satisfy vehicle dynamic constraints</td>
</tr>
</tbody>
</table>

* This work was supported by Research Center for Smart Vehicles of Toyota Technological Institute and DENSO CORPORATION.

Quoc Huy Do, Keisuke Yoneda, Ryohi Sakai and Seichi Mita are with the Toyota Technological Institute, Nagoya, Aichi 468-8511 Japan. (e-mail: [sd89601, yoneda, srl11412, smita]@toyota-ti.ac.jp).
H. Tehrani Niknejad is with Corporate R&D Div.3, DENSO CORPORATION (e-mail: hossein_tehrani@denso.co.jp).
In safety margin maximization scheme, the SVM shows excellent performance in obtaining the wide path clearance [13]. Miura [14] proposed the subsequent non-linear separating surface generated from the SVM as an agent path. This was the first discovery of the relationship between path planning and SVM, but due to a lack of suitable preprocessor, that paper shows that direct use of SVM cannot succeed every time. Inheriting intrinsically the excellent smoothness of RBF, we propose a path planner based on RBF kernel SVM which is able to produce path in its intact smoothness rather than those produced by other the approaches. On the other hand, with the SVM optimized large margin, the planned path also shares the merit of a maximum clearance for the vehicle passage to handle mapping and localization uncertainty. This paper is an extension and improvement of previous work [15] from local path planning to more global path planning. The proposed method doesn’t require full and accurate information about the environment and the complexity of the algorithm is not affected by the shape of the obstacles. It finds the most critical data points in the map considering Lagrange multiplier of SVM dual problem. By finding critical points in the map, the size of the processing data points for path planning is enormously reduced and the desired paths can be rapidly generated through hyperplane that divides the critical points.

The structure of this paper is as follows: Section II describes map discovering and clustering the map data points. Section III presents the SVM and path planning hyperplane. Experimental results are presented in Section IV.

II. CLUSTERING DATA POINTS

A. Map discovering

In order to apply SVM, we divide the obstacles’ data points along the road from start point to the goal point into two separated groups. In this paper, we applied the Bidirectional Rapidly-exploring Random Tree (Bi-RRT) and the hybrid A* method. The hybrid A* introduced by Dolgov [4] is applied to the 3D kinematic state space using an heuristic to decrease the search space. The Bi-RRT can quickly discover the map and ensure to avoid the local minima phenomenon which may happen in complicated map [16]. In this step, no more constraints except collision avoidance were considered so that Bi-RRT can quickly return a path. The guiding path has no safety or smoothness property.

B. Selecting and clustering obstacles points

For the unknown or partially known environment, the obstacles’ shapes, sizes and postures are unknown. In practical path planning application, selecting data point (obstacles’ points) from the map for consideration is an important step. However, in conventional path planning methods, this step is often ignored and the shapes of the obstacles are not considered. The Voronoi diagram or risk potential field based method needs to collect all the data points in the map to generate the “skeleton map” for path searching algorithm [4],[11],[12] as shown in Fig.1 (a). The advantage of using the generated Bi-RRT guiding path is that we do not need to consider all the data points in the global map to process. After generating the guiding path, we select data points on the borders (left and right side) along this path as labeled data points to input as training data sets for SVM method. Fig. 1 (b) illustrates an example of the Bi-RRT path and input data sets. The obstacle borders points are divided in two left (orange points) and right (blue points) group. It is obvious that the case of the two data sets overlap each other will not happen because we never try to go through the obstacle.

C. Geometric margin (path corridor)

In this section, we define the notion of a geometric margin or safe corridor for generating paths. The geometric margin definitions connect the path planning with classification methods such as SVM. In classification, we are trying to maximize the margin between different classes of data. Similarly, in path planning, we are trying to find the safest corridor through map for car movement.

For a given hyperplane in 2D, we denote by \( p_- (p_+) \) the closest point to the hyperplane among the left (right) data points. Here, \( p_- \) refer to the obstacle points with left label and \( p_+ \) refer to the obstacle points with right label. From simple geometric considerations the margin of a hyperplane \( f \) with respect to a dataset \( X \) is:

\[
m_D(f) = \frac{1}{2} \bar{w}^T (p_+ - p_-)
\]

where \( \bar{w} \) denotes the vector normal to the decision boundary, \( \bar{w} \) is a unit vector in the direction of \( w \). We assume that \( p_+ \) and \( p_- \) are equidistant from the decision boundary i.e.

\[
f(p_+) = w^T p_+ + b = a
\]

\[
f(p_-) = w^T p_- + b = a
\]

for some constant \( a > 0 \) and \( b \) the bias. To make the geometric margin meaningful we fix the value of the decision
function at the points closest to the hyperplane, and set $a = 1$. Adding (2) and (3) then dividing by $\|w\|$, we have:

$$m_0(f) = \frac{1}{\|w\|}$$  \hspace{1cm} (4)

SVM generates hyperplane which maximizes the margin. We will explain in next section how to maximize the margin of path planning hyperplane and generate safe corridor for vehicle path planning through the data points.

III. SVM AND HYPERPLANE

Based on statistical learning theory, an SVM based on maximizing the margin [17] has been well developed for the classification and regression learning problem. Assume there is one set of data point samples, $\{p_1, p_2, \ldots, p_n\}$, acquired from the 2D space $X, X \in R^2$, in which $p_i$ is a data point in $X$. As a supervised learning scheme, data point sample $p_i$ in $X$ have been labeled in advance as $y_i \in \{-1,1\}$. Here, -1 refers to the obstacle points with left label and +1 refer to the obstacle points with right label.

The theory of SVM classification, learning with the labeled samples, gives then an optimized solution to produce a decision boundary $f(p) = 0$. Fig. 2 shows an example of a linear case, $f(p) = w^T p + b$, where $w$ denotes the vector normal to the decision boundary and $b$ the bias. The decision boundary tends to separate the whole set of sample data points into two categories according to their labels, i.e., it places sample data points with different labels on either side, $f(p) > 0$ and $f(p) < 0$, respectively. Though there are methods to generate a variant decision boundary, the boundary generated by the SVM classifier shares the merit of having large safety margins [16]. The margin, as the defined width of the maximal clearance that the classifier can achieve for separation from the obstacle, can be illustrated as the street clearance along the decision boundary (Fig. 2). Intuitively, with a larger margin, there is wider clearance for the vehicle to pass through.

For non-straight paths, a non-linear SVM is more practical. Employing the kernel technique [18] for high dimensional similarity manipulation, the linear SVM can be extended to a non-linear SVM. The idea is that a non-linear problem can be solved linearly when mapping the discriminant feature space into a high dimensional reproducing kernel Hilbert space (RKHS) $H$. The kernel function

$$k(p_i,p_j) = \varphi(p_i)^T \varphi(p_j)$$  \hspace{1cm} (5)

It manipulates the similarity of the feature map, $\varphi(p): R^2 \rightarrow R^H$. With the kernel trick, the quadratic optimization problem of SVM can be abbreviated eventually as follows:

$$\max_\alpha \sum_{i=1}^{n} a_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} a_i a_j y_i y_j k(p_i,p_j)$$  \hspace{1cm} (6)

Subject to $0 \leq a_i \leq \lambda, \forall i$

where $\alpha = [\alpha_1, \alpha_2, \ldots, \alpha_n]^T$ denotes a set of Lagrange multipliers, and $\lambda$ denotes a scaling factor to control the trade-off between the maximum margin and the minimum total remaining error of the trained classifier. From the KKT conditions, only those sample points conforming to the KKT complementarities, having non-zero $\alpha_i$ and referred to as support vectors (SVs), contribute to the solution. The SVs can be categorized more specifically as the margin SVs and non-margin SVs [18]. We are able to generate the hyperplane using SVs as shown in (7). For hyperplane which maximize the safety margin $f(p) = 0$.

$$f(p) = \sum_{\forall i} a_i y_i k(p_i,p) + b$$  \hspace{1cm} (7)

The Lagrange multipliers have important rule to guarantee the safety of the generated path for vehicle. For any data point $p_i$, the corresponding Lagrange multiplier $\alpha_i$ tell us the impact or importance of the data point in path planning. $\alpha_i = 0$ means the data point $p_i$ is out of margin and it is not necessary to take care about it in path. Point $p_j$ which $\alpha_j > 0$, means it is a critical point in the geometric margin and has impact on the path direction and orientation.

We are able to calculate the maximum geometric margin or safe path corridor through the following Lagrange multipliers:

$$w = \sum_{i=1}^{n} \sum_{j=1}^{n} a_i \alpha_j y_i y_j \varphi(p_i)^T \varphi(p_j)$$  \hspace{1cm} (8)

A. Gaussian RBF kernel

Several types of kernel function are frequently employed to solve classification problems, for example, the linear, polynomial, and radial basis function (RBF) kernels [19]. Due
to the non-holonomic constraints for car movement, the RBF kernel is particularly re-examined for its use to this purpose. In general, the RBF kernel, also known as the Gaussian kernel, is given as follows:

$$k(p_i, p_j) = \exp(-\delta \| p_i - p_j \|^2)$$  \hspace{1cm} (9)

If the distance between $p_i$ and $p_j$ is much larger than $\frac{1}{\sqrt{\delta}}$, this expression tends to zero so that for a fixed $p_i$, it is localized to an area around $p_i$. When $\delta$ is small (bottom right panel in Fig. 3) a given data point $p$ has a non-zero kernel value relative to any data points in the set of SVs. Therefore, the whole set of SVs affects the value of the discriminant function $f(p)$ at $p$, resulting in a smooth geometric margin boundary. When $\delta$ is increased (top left panel in Fig. 3) the locality of the SVs expansion increases, leading to greater curvature of the geometric margin boundary. When $\delta$ is large the value of the $f(p)$ is essentially constant outside the close proximity of the region where the data points are concentrated (see top left panel in Fig. 3). In this regime of the parameter the classifier is clearly over-fitting the data points and we have redundant hyperplanes.

**B. Path extraction from hyperplanes and SVs**

With RBF kernel, SVM generates safe and smooth path for the vehicle. After solving the dual problem in (6) as described in [18], we achieve the hyperplane and the SVs. The hyperplane (path) equation based on the SVs and Lagrange multiplier is shown in the following

$$f(p) = \sum_{\text{SV}} \alpha_i y_i \exp \left( -\delta \| p_i - p_j \|^2 \right) + b$$  \hspace{1cm} (10)

The hyperplane is in the featured space and it is implicitly in the original 2D space and we use an approximation method to acquire a path nearest to the hyperplane. Furthermore, the path should be smooth and feasible for non-holonomic movement of vehicle. From the start point, we generate the path’s control points based on the vehicle dynamic model to track along the hyperplane in (10) as shown in Fig. 4. The control points are selected to minimize the following cost function,

$$C = \alpha D(p) + \beta (\Delta \phi)$$  \hspace{1cm} (11)

where $p = (x, y)$ are the vertices of the control point $p$, $D(p)$ is the distance from a point $p$ to the hyperplane.

$$D(p) = \frac{|w . p + b|}{\|w\|}$$  \hspace{1cm} (12)

$w$ and $b$ are hyperplane margin and bias achieved from the training process. $\Delta \phi$ is the change in the tangential angle at the point $p_i$

$$\Delta \phi = \left| \arctan \frac{y_{i-1} - y_{i-2}}{x_{i-1} - x_{i-2}} - \arctan \frac{y_{i-2} - y_{i-3}}{x_{i-2} - x_{i-3}} \right|$$  \hspace{1cm} (13)

$\Delta p = p_i - p_{i-1}$ is the displacement vector at point $p_i$, $\kappa_{max}$ is the maximum curvature of the path that the vehicle can follow and $\alpha, \beta$ are weights. The first term of the cost function effectively guides the vehicle away from obstacles in both narrow and wide passages and the second term expressed the smoothness of the path.

To generate the final smooth path, the B-spline curve is applied to interpolate detailed (smaller distance) path points. One of the advantages of B-spline is that we can modify a curve locally without changing the shape in a global way. In our approach, the curve with degree $n=3$ was considered which guarantee to provide a $C2$ continuous path but less computational complexity than other higher degree curves. The control points of the B-spline curve are the path control points calculated in previous step. An example of the globally planned path was illustrated in Fig. 5. The green and red points represent the support vectors on the border of the obstacles on the right and left side respectively.
IV. EXPERIMENTS

A. Experimental Condition

![Experiment environment](image)

Fig.6. Outdoor autonomous parking experiments to evaluate the proposed method

We have implemented the path planning algorithms in our autonomous vehicle platform. We did some experiments for outdoor autonomous parking and evaluated different path planning algorithms. We implement our method and compare with the Dolgov method [4]. Figure 6 (a) shows the experiments environment for autonomous parking with sizes 2.5 x 120 meters and Fig. 6 (b) depicts the corresponding risk potential map. In Fig.6 (b), the white area indicates safety area which has a clearance with nearest obstacle. We set a start point and three different goal points. At the goal point “A” and “B”, the vehicle needs backward-parking and at the point “C”, it needs a forward-parking. We perform three types of experiments to evaluate SVM method, compared to Dolgov method [4]. The Dolgov-like method uses Hybrid-State A* (H-A*) and then optimizes by Risk Potential Map (RP). To explore map and generate guided path, we use two well known method including Hybrid A* (H-A*) and Bidirectional RRT (Bi-RRT). All algorithms are implemented in Visual C++ on a 3.4Ghz CPU computer

B. Evaluation results

Figure 7 depicts a graph which summarized a computational time. The gray part shows the calculation time of first part of algorithms to generate a guide path from start to goal using Hybrid-State A* (H-A*) and Bidirectional RRT (BRRT). As BRRT is a randomized algorithm, we performed the same experiment 1,000 times and compared average characteristics.

<table>
<thead>
<tr>
<th>Scenario Method</th>
<th>Goal “A”</th>
<th>Goal “B”</th>
<th>Goal “C”</th>
</tr>
</thead>
<tbody>
<tr>
<td>BRRT+SVM</td>
<td>2.031[m]</td>
<td>1.971[m]</td>
<td>2.088[m]</td>
</tr>
<tr>
<td>H-A*+RP</td>
<td>2.240[m]</td>
<td>2.341[m]</td>
<td>2.302[m]</td>
</tr>
<tr>
<td>H-A*+SVM</td>
<td>2.234[m]</td>
<td>2.367[m]</td>
<td>2.341[m]</td>
</tr>
</tbody>
</table>

The white bar depicts the path optimization time in the second step using SVM and Dolgov method which optimizes risk potential map (RP) [4].

Table 2 and 3 show a number of gear transition and average safety margin for final path. Figure 10 depicts the final typical paths for different methods. We can easily extract the following conclusion.

- SVM is almost nine times faster than risk potential map for path optimization (Fig. 7).
- Hybrid State A* is able to build a path which has the minimum gear transition (Table 2).
- Bidirectional RRT (BRRT) fails to generate stable gear transition (Table 2).
- The average safety margin for all methods is almost same and there is no big difference between different methods (Table 3).
- The final path smoothness for H-A*+SVM and H-A*+RP is almost same and both are able to generate suitable path for car movement. (Fig. 8).

The evaluation results show that SVM is fast and effective method for path optimization compared to Dolgov method using the risk potential map. Risk potential map needs a margin value at each position in the map though SVM just consider support vectors which have positive Lagrange multipliers.
V. CONCLUSIONS

In this paper different from traditional approach such as Voronoi or potential map, we use learning approach to classify data points and maximizing the safety margin. We use RBF kernel SVM to classify data points in map and find the most critical points for path planning. We can reduce the size of large scale map by finding support vectors and also we are able to generate the optimized hyperplane which maximizes safety corridor for car movement. By reducing the size of map, this method is faster than other method and by using RBF kernel it generates smooth path for nonlomonic movement of car. The evaluation results show that SVM is fast and effective method for path optimization. The resulted path by using Bi-RRT may have less computation time but the hybrid A* paths have less number of gear changes. Our advantage is the computation time. However, the number of gear transition is more than previous system and this will affect the executing time. Thus, in the future work, we will focus on enhance of the Bi-RRT based method such that the resulted path will have smaller gear transition along the path.

REFERENCES