2012 International Conference on Intelligent Systems Design and Engineering Application

A Two-level Path Planning Method for On-road Autonomous Driving

Liang Ma, Jing Yang, Meng Zhang
Institute of Artificial Intelligence and Robotics, Xi'an Jiaotong University, Xi'an, Shaanxi, 710049, P.R. China
lm@aiar.xjtu.edu.cn

***Abstract*—We present a real-time path planning method for on-road autonomous driving. This method utilizes a two-level hierarchy consisting of on-road behavior planning and online path generating to achieve feasible paths for autonomous vehicle in dynamic on-road environment which is characterized by lots of structured information and flat terrain. An on-road behavior planning technique is used to transform the uncertain path planning problem into several determined sub-ones. Then, the online path planning algorithm is applied to solve these sub-problems. This method contains three major components: anytime behavioral replanning, online path planning and mitigation strategy for time lag due to an agent system with asynchronous multi-sensor. We evaluate the proposed method on the KUAFU-1, an autonomous vehicle prototype modified by ourselves. Experimental results show that the proposed method can successfully provide real-time feasible paths in various on-road traffic sceneries.**

Keywords- path planning; behavioral planning; time lag

I. Introduction

A. Motivation

Path planning for an agile autonomous vehicle in a dynamic environment is a challenging problem, especially when the vehicle is required to use its full maneuvering capabilities [1]. We focus on the problem of path planning for on-road autonomous driving in this paper. On-road environment is characterized by lots of structured traveling information and flat terrain, and this somewhat alleviates difficulties and complexities of the path planning. Our aim is to propose a simple, feasible path planning method for autonomous vehicle driving in urban environments.

B. Related Work

The task of path planner can be stated as: taking into account the environmental model as well as the vehicle's dynamics and constraints, and then providing a feasible reference path to guide the vehicle approaching a goal location in the desired lane.

There is a rich literature on path planning. In general, these works can be roughly divided into two categories: geometric methods and graph search methods. In geometric methods, a sequence of geometric primitives such as line, arcs [2], Clothoids [3], Cubic spirals [4], Bezier [5], B-splines, and so on, are used to generate a feasible path, while methods belonging to the category of graph search apply some widely-used search algorithms including A* [6], anytime variations of A* [7], D* [8], PRM [9], and RRT (Rapidly-exploring random trees) [10] to generate paths.

Our path planning method builds on the previous work belonging to the first category. It consists of two levels: anytime behavior replanning and local smooth curve generation. By anytime behavior replanning, we transform the uncertain on-road path planning into several sub-problems with determined and specified goals. These sub-problems are then solved by the online path planning algorithm. Moreover, we develop an efficient mitigation strategy for time lag due to latency between a asynchronous multi-sensor system and the motion planning system.

II. Local Smooth Path Planning

During on-road driving, the goal location issued from the behavioral planning is a location within a road lane. Path planning is then the problem of determining a feasible path that moves the vehicle from an initial location to the goal location in the desired lane. Our path planner has two main steps: 1) path generator, and 2) path evaluation.

A. Bezier Path Generator

The goal issued from the behavior planner entails a desired lane and a desired position within that lane. With this goal, the path generator generates a set of curves as candidate paths to track the desired lane as well as towards the goal location. We choose cubic Bezier curves as the path generator because Bezier curve has the desired properties of smoothness and controllability.

The cubic Bezier curve is a parametric curve as follows.

$$B(t) = P_0(1-t)^3 + 3P_1 t(1-t)^2 + 3P_2 t^2(1-t) + P_3 t^3, t \in [0,1] \quad (1)$$

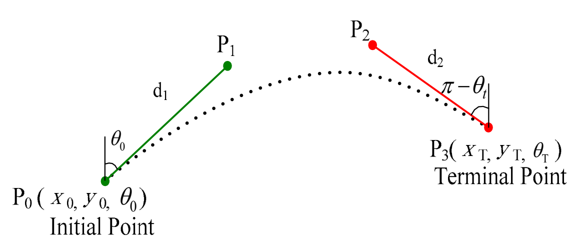

Figure 1. The cubic Bezier curve is shown by dashed line.

As shown in Figure.1, the shape of a cubic Bezier curve is controlled by four points, P0, P1, P2, and P3. The curve starts at P0 going towards P1 and arrives at P3 coming from the direction of P2. P1 and P2 are there to provide

This work is supported by the National Natural Science Foundation of China under Grant No.90920301.

directional information. More specifically, the distance between P0 and P1 determines 'how long' the curve moves into direction P2 before turning towards P3.

In our case, the P0 and P3 are determined by the current location of the vehicle and the goal location issued from the behavior planner. We then generate the candidate curves by setting different distances between P0 and P1, P2 and P3.

B. Path Evaluation

The optimal path is chosen from candidate curves by an evaluation module, which consists of two levels.

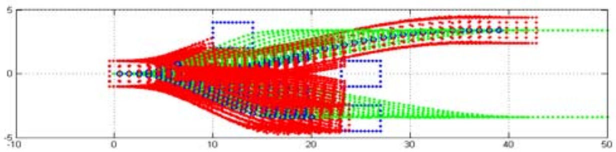

Figure 2. Collision check of candidate paths

The first level is a collision check method to choose collision-free paths. Traditionally, this is done through covering the vehicle and obstacles by multiple circles [11]. This approach will inevitably lose certain feasible driving regions. To overcome this limitation, vehicles and obstacles are represented by rectangular boxes which just encircle their contours. These boxes are then discretized into regularly spaced point sets for numerical computation. Consider point set A as all the possible point sets of vehicles that traverse along the candidate paths (as shown red point sets in Figure 2), and point set B as all the point sets of detected obstacles at the very moment (as shown blue point sets in Figure 2). The collision check problem is then converted to a nearest points search problem between two point sets. We calculate the minimum distance $\boldsymbol{d_{min}}$ between A and B using the classical K-D Tree[12] method, and paths with $\boldsymbol{d_{min}}$ smaller than $\boldsymbol{d_{safe}}$ will be considered unqualified and rejected by the selector.

The second level of the evaluation module is a function to choose the optimal path. We define a cost function to evaluate each candidate path. The path with minimum cost will be chosen as the optimal path. Our function is defined as:

$$J = w_1 S + w_3 \sum \kappa^2 + w_2 \sum d_{center}^2 \qquad (w_1 + w_3 + w_2 = 1) \tag{2}$$

where w_1 w_2 w_3 respectively denotes the weight for length, smoothness, and offset from the central line of the lane. κ denotes curvature of the path, S denotes the arc length of path, $\boldsymbol{d_{center}}$ denotes the distance between path and the centerline of lane.

Usually, the cost calculation of a path is calculated with all points of the path. In order to speed up the calculation, we sample points in the curve with equal arc length, and then evaluate the cost of the curve by these sampling points.

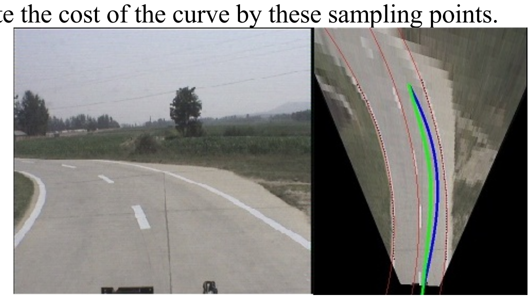

Figure 3. Bezier curve generated in lane keeping of big curvature lane

Due to limitation of the cubic Bezier curves, curvature of the planned path may be hard to conform to the large curvature of the lane. Thus we introduce a centerline constrain in the cost function in Equation (2) to alleviate this problem. Figure.3 shows an example of the role of the central line constraint, where the blue curve denotes the path generated without the central line constraint, and the green curve denotes the path generated with the central line constraint.

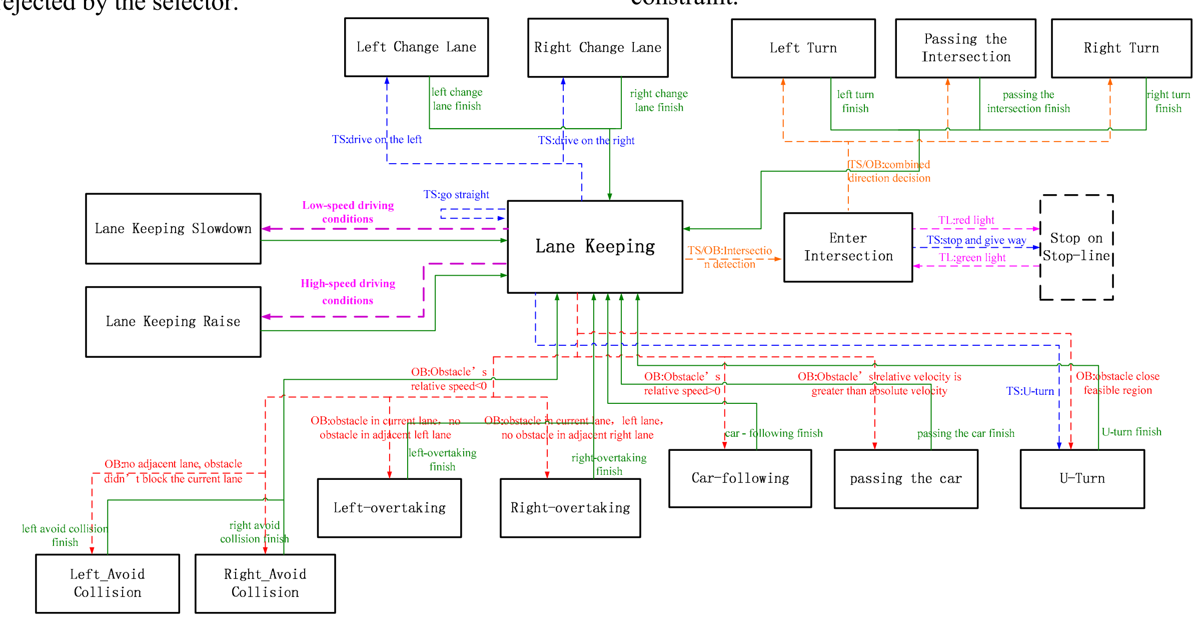

Figure 4. Schematic diagram of the behavior planner

III. Behavioral Planning

The behavioral planning is a sub-system in upper level of path planning. It plays a role of decision making by taking both driving mission and environmental information into account. When a given driving mission is performed, vehicle's behavior can be regarded as a sequence of driving behavior consisting of a set of driving maneuvers. In the particular context of on-road driving, the set of driving behavior includes lane keeping, lane change, intersection handling, overtaking, passing, U-turn and so on. Each driving behavior defines a motion goal for the local path planner. Choosing the proper maneuver sequence and implementing the chosen on-road maneuver are two main tasks of behavioral planning.

Figure.4 shows the schematic diagram of our behavior planner. By this behavior planner, we transform the uncertain problem of on-road path planning into several sub-problems with determined and specified local goals. For instance, in lane-keeping, the local goal is selected as a point that belongs to the centerline of current lane; in lane-change, it is chosen as a location in the centerline of adjacent lane; in overtaking, the local goal is within the feasible driving region close to target vehicle; in intersection handling, the local goal is thrown to the selected exit; in U-turn, the local goal consists of two sub-goals, which are determined by dividing the path into two segments. In case of dynamic obstacles on road, the issue becomes more complicated. We need to find more than one local goal in the feasible driving region close to moving obstacles so as to guarantee a safe and feasible path generation. Moreover, when the distance of the vehicle to the moving obstacles is less than a certain threshold value, fast re-planning should be done to produce a new path which can guarantee the vehicle to pass moving obstacles.

The converting of driving behaviors is also shown in Figure.4. How to choose an orderly set of driving behaviors to complete a driving mission is decided by two aspects, namely driving mission and environmental information that influences driving behavior decision. A correct driving behavior must take into account the environmental information, which mainly includes road, traffic sign and obstacle. At an intersection, for example, exits that lead to destination are firstly chosen according to a driving mission, then exits with sufficient space to pass are picked in terms of obstacle information. Finally the permitted exit is decided according to traffic sign.

IV. Mitigation Strategies for Time Lag

Autonomous driving is performed by an environment-in-the-loop control system. Time lag between the environment perception and the vehicle motion affects the performance of the system. In particular, the time latency due to environment perception and the path planning may cause the vehicle deviating from the planned path.

We assume that the sensed environmental data is captured at point A at time t_0, and the path planner output a path from A to the goal point B at time t_1. The planned path AB is shown as the blue curve in Figure.5. However, the vehicle has already moved to A' at time t_1 .The trajectory of the vehicle from A to A' (shown as the dashed line in Figure.5) is formed by tracking the path planned before t_0. The position shift d_{shift} of the vehicle from the planned path AB appears. This position shift can be ignored only when three conditions are satisfied with: 1) the time latency $\Delta t=t_1-t_0$ is small, 2) the vehicle moves with a slow speed, and 3) the path tracking algorithm is accurate enough. In on-road autonomous driving, this position shift should be taken into account.

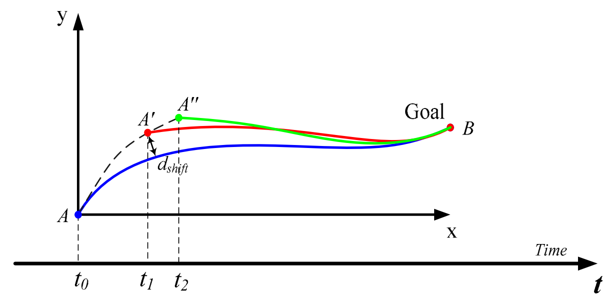

Figure 5. Illustration of the mitigation strategy for the time lag

We address this problem by two mitigation strategies: 1) increasing the frequency of environment perception such as detection of lane and moving obstacles, and 2) replanning a path from the vehicle's position A' to the goal point B. The first strategy is straightforward, and it depends on the hardware configuration of the environment perception sub-system.

In the replanning, we use a path planner similar to that presented in Sec.II. The only modification is that we replace the central line constraint in Equation (2) with the planned path AB. The benefit of this modification is that we need not do environmental perception with the sensed data at time t1. The time lag due to environmental data processing can be skipped.

To further mitigate the effects of time lag due to the replanning, we need to consider the position of the vehicle at the time when the replanned path A'B starts to be tracked. As shown in Figure.5, let A" denotes the position of the vehicle at time t2 when the path A'B is produced. The reference path for the path tracking should starts from A", whose position can be measured by the localization system of the vehicle. The actual trajectory of the vehicle may thus become A" B, as shown the green curve in Figure.5.

V. Experiment and Result

In order to verify the feasibility and robustness of the proposed path planning method, we have performed simulative experiment and real on-road driving experiment respectively. Experimental results are shown in Figure.6 and Figure.7. Figure.6 shows the simulative results of lane keeping, lane change and intersection handling. It is shown in this figure that multiple local goals (red stars) are thrown. The candidate paths are then generated by the Berizer path generator, as shown the green curves. The optimal path is picked from these candidates.

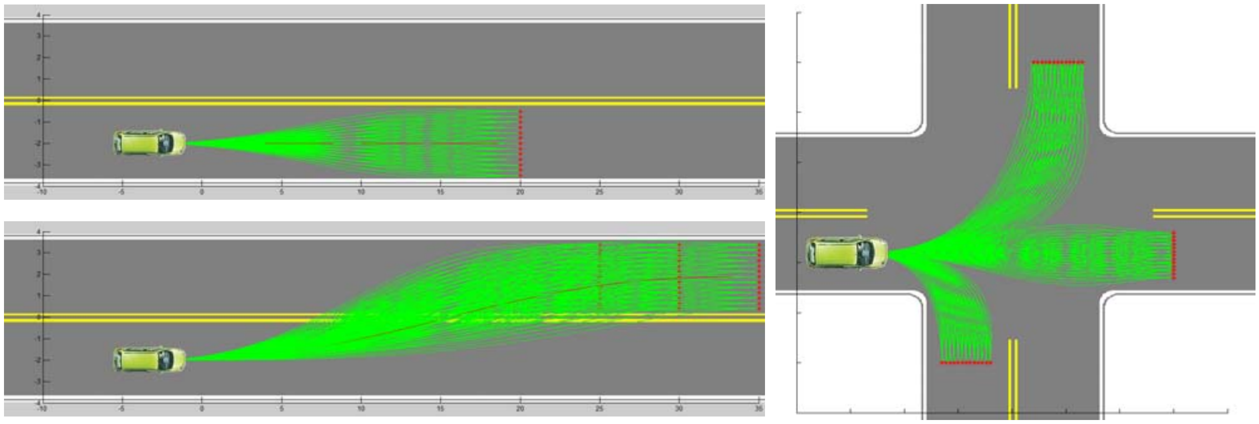

Figure 6. Simulative result of lane keeping, lane change and intersection

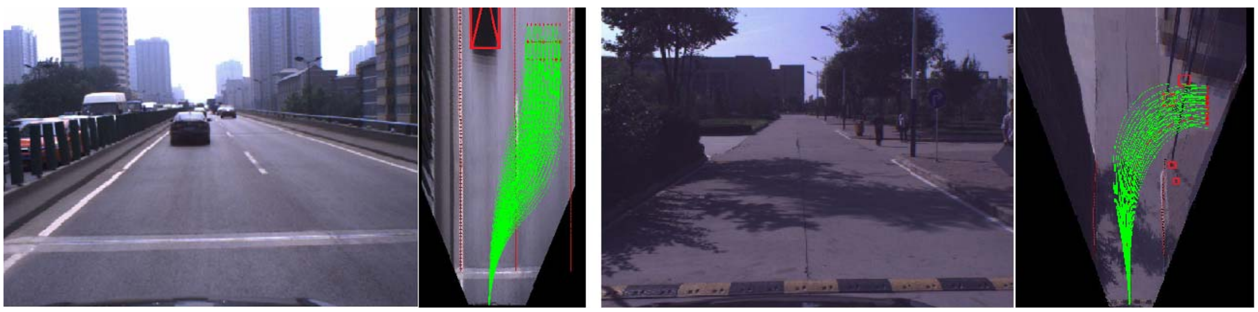

Figure 7. Real on-road driving experimental result of overtaking and intersection handling

Figure.7 shows the real on-road experimental results of overtaking and intersection handling. The left side of each experimental result demonstrates the actual environmental setting. In the experiment, multiple local goals are thrown to feasible driving region adjacent to obstacles (red boxes) which are chosen, and candidate obstacle-free paths (green curve) are generated. Such a complicated maneuver is composed of two or three frames' continuous planning.

VI. CONCLUSIONS

This paper presents a two-level path planning method for on-road autonomous driving. We address the on-road path planning in three major components: anytime behavioral replanning, multi-paths generation, mitigation strategy for time lag. We test the proposed method on the KUAFU-1, an autonomous vehicle prototype modified by ourselves. Experimental results show that the method can successfully provide real-time feasible paths in various on-road traffic sceneries. Our future work involves extending the method to the path planning under complex off-road environments.

REFERENCES

[1] E. Frazzoli, M. A. Dahleh, and E. Feron, "Real-time motion planning for agile autonomous vehicles," AIAA Journal of Guidance and Control, vol. 25, no. 1, pp. 116–129, 2002.

[2] Dubins, L. E. 1957. "On curves of minimal length with a constraint on average curvature and with prescribed initial and terminal positions and tangents." American Journal of Mathematics 79:497–516.

[3] Y. Kanayama and N. Miyake. "Trajectory generation for mobile robots," In Proceedings of the International Symposium on Robotics Research, pages 16–23, 1985.

[4] Y. Kanayama and B.I. Hartman. "Smooth local path planning for autonomous vehicles," In Proceedings of the International Conference on Robotics and Automation, volume 3, pages 1265–1270, 1989.

[5] K. Komoriya and K. Tanie. "Trajectory design and control of a wheel-type mobile robot using b-spline curve," In Proceedings of the IEEE-RSJ International Conference on Intelligent Robots and Systems, pages 398–405, 1989.

[6] P.E. Hart, N.J. Nilsson, and B. Raphael. "A formal basis for the heuristic determination of minimum-cost paths," IEEE Transactions of Systems Science and Cypernetics, 4(2), 1968.

[7] R. Zhou and E. Hansen. "Multiple sequence alignment using anytime A*," In Proceedings of the 18th National Conference on Artificial Intelligence, 2002.

[8] M. Likhachev, D. Ferguson, G. Gordon, A. Stentz, and S. Thrun. "Anytime dynamic A*: An anytime, replanning algorithm," In Proceedings of the International Conference on Automated Planning and Scheduling, June 2005.

[9] L.E. Kavraki and J.C. Latombe. "Randomized preprocessing of configuration space for fast path planning," In Proceedings of IEEE International Conference on Robotics, pages 2138-2145, 1994.

[10] La Valle SM, Kuffner J.J, Jr. "Randomized kinodynamic planning," International Journal of Robotics Research, vol. 20 pp: 378-400. 2001,

[11] Tetsuya Uchiki, Toshiaki Ohashi and Mario Tokoro. "Collision detection in motion simulation," Computer & Graphics.285-293.1983

[12] Moore A w. "An introductory tutorial on KD-Trees," Technical Reporl No. 209. Computer Laboratory. University of Cambridge. London. l-20. 199l.